Causal Inference Assignment Report

CS112 - Professor Diamond, A.

Ash Nguyen

2018/12/06

We begin with loading and cleaning the data.

```

#Loading the relevant libraries
library(foreign)
library(Matching)

#Cleaning data
Expe<-read.dta(”/Users/ash/Downloads/nsw_dw.dta”)
Controls<-read.dta(”/Users/ash/Downloads/cps_controls.dta”)
for (i in c(5,6,7,8)){
  Expe[,i]<-as.factor(Expe[,i])
}
for (i in c(5,6,7,8)){
  Controls[,i]<-as.factor(Controls[,i])
}
lapply(Expe,class) #check if the classes are correct
lapply(Controls,class)
```

1 Experimental Treatment Effect

To evaluate the treatment effect for the experimental data is straightforward because the units were properly randomized and we can assume a similar distribution across all observed and unobserved covariates for the control and the treatment units. In fact, we can check this assumption for the observed covariates by plotting several histograms for some covariates for both groups and realized that they are similar. For two covariates *age* and *education*, the results are plotted in Figure 1.

The treatment effect, calculated directly from the subtraction of the means of the treated from the mean of the control is \$1794.342. The 95% confidence interval for this estimate, using a simple linear regression and its standard error, is between \$1161.489 and \$2427.196.

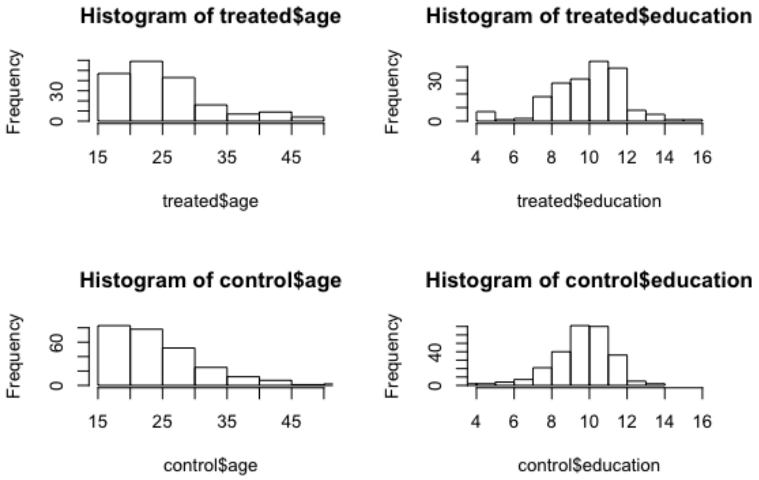

Figure 1: Histograms of *age* and *education* for treated and control units in experimental setting

```
###------Experimental Treatment Effect------###

#1.Simple treatment effect: difference in means
mean_control<-mean(Expe$re78[Expe$treat==0])
mean_experimental<-mean(Expe$re78[Expe$treat==1])
expe_effect<-mean_experimental-mean_control
cat("Experimental Treatment Effect:",expe_effect)

#2.Confidence interval by linear regression
linreg<-lm(re78~treat,data=Expe)
vals<-summary(linreg)
confi_95<-c(vals$coefficients[2,1]-1.96*vals$coefficients[2,2],
            vals$coefficients[2,1]+1.96*vals$coefficients[2,2])
cat("95% Confidence Interval:",confi_95)
```

```
[O]:
Experimental Treatment Effect: 1794.342
95% Confidence Interval: 553.9497 3034.735
```

2 Non-experimental Treatment Effect

After subtracting the experimental control units and adding the CPS data for the untreated, we have a large sample of units. Now, the data is non-experimental, since the untreated is chosen via means different than pure randomization. Using the same approach as above, we have the treatment effect for this non-experimental dataset is $-\$8497.516$ and its 95% confidence interval is $-\$-9893.077$ and $-\$7101.956$. This result is way off the experimental benchmark, but is to be expected. This shows that non-experimental data cannot be trusted for evaluating the true causal effect because it is radically different from the true causal effect.

```
###______Non-experimental Treatment Effect______###

#1.Creating a non-experimental dataset
Non_expe<-subset(Expe, treat==1)
#check if the variables are all in the same place
names(Controls)==names(Non_expe)
Non_expe<-rbind(Non_expe,Controls)
Non_expe<-Non_expe[,!(names(Non_expe)=="data_id")]

#2. Non-experimental treatment effect
nmean_control<-mean(Non_expe$re78[Non_expe$treat==0])
nmean_experimental<-mean(Non_expe$re78[Non_expe$treat==1])
n_expe_effect<-nmean_experimental-nmean_control
cat("Non-experimental Treatment Effect:",n_expe_effect)

n_linreg<-lm(re78~treat,data=Non_expe)
n_vals<-summary(n_linreg)
n_confi_95<-c(n_vals$coefficients[2,1]-1.96*n_vals$coefficients[2,2],
              n_vals$coefficients[2,1]+1.96*n_vals$coefficients[2,2])
cat("95% Confidence Interval:",n_confi_95)
```

```
 [O]:
Non-experimental Treatment Effect: -8497.516
95% Confidence Interval: -9893.077 -7101.956
```

3 Regular Matching Treatment Effect

Naturally the next step is to use some kind of matching to try to achieve the level of similarity between the treated and the control as the experimental data. We start with regular matching by propensity scores.

3.1 Propensity Scores Only

We first find the predicted propensity scores for both the treated and the control in the non-experimental data by a generalized linear model with logistic regression. This is of course not the unit true propensity scores, as we only estimate these scores using the observed covariates. Next, we will use the Match() function to try to find best matches for the treated units among the plethora of control units. The result will be checked via the MatchBalance() function, which tells us the level of similarity between the distributions of the covariates after matching.

```
#1.Calculating the propensity scores
prop_model<-glm(treat~.-re78,data=Non_expe,family=binomial)
scores<-predict(prop_model,type='response')
```

```

Y=Non_expe$re78
Tr=Non_expe$treat

#2.Match() using only propensity scores

#Match with propensity scores
prop_match<-Match(Y=Y,Tr=Tr,X=scores,M=1)
Matched<-data.frame(prop_match$mdata$Y,prop_match$mdata$Tr)
names(Matched)<-c('re78','treat')
MatchBalance(treat~.-re78,data=Non_expe,match.out=prop_match,nboots=100)
```

```
[O]:
Before Matching Minimum p.value: < 2.22e-16
Variable Name(s): age education black1 married1 nodegree1 re74 re75  Number(s): 1 2 3 5 6 7 8

After Matching Minimum p.value: < 2.22e-16
Variable Name(s): age education re74 re75  Number(s): 1 2 7 8
```

The balance check results are not very promising: even though some of the covariates have high p-values, there are still four variables that has extremely small p-values, which means the matched set does not resemble the treated units in these covariates. Nevertheless, we proceed to calculate the mean treatment effect.

```
#Matched treatment effect
m_mean_control<-mean(Matched$re78[Matched$treat==0])
m_mean_experimental<-mean(Matched$re78[Matched$treat==1])
m_expe_effect<-m_mean_experimental-m_mean_control
cat("Matched Treatment Effect:",m_expe_effect)

#Matched confidence interval
m_linreg<-lm(re78~treat,data=Matched)
m_vals<-summary(m_linreg)
m_confi_95<-c(m_vals$coefficients[2,1]-1.96*m_vals$coefficients[2,2],
              m_vals$coefficients[2,1]+1.96*m_vals$coefficients[2,2])
cat("95% Confidence Interval:",m_confi_95)
```

```
[O]:
Matched Treatment Effect: 824.574
95% Confidence Interval: 520.3935 1128.754
```

The results is certainly better than with no matching at all, but not as good as the experimental benchmark. This is understandable as in transforming the covariates into propensity scores, we looses important information about these covariates' distributions. Therefore, we will continue with regular matching, this time with an attempt to achieve balance on all covariates, not just propensity scores.

3.2 All Covariates And Propensity Scores

Here we match for all other covariates, including the propensity scores. We therefore have a much better balance.

```
X_all=cbind(scores,Non_expe$age,Non_expe$education,Non_expe$black,Non_expe$married,
          Non_expe$hispanic,Non_expe$nodegree,Non_expe$re74,Non_expe$re75)

all_match<-Match(Y=Y,Tr=Tr,X=X_all,M=1)
Matched_all<-data.frame(all_match$mdata$Y,all_match$mdata$Tr)
names(Matched_all)<-c('re78','treat')
MatchBalance(treat~.-re78,data=Non_expe,match.out=all_match,nboots=100)
```

```
[O]:
Before Matching Minimum p.value: < 2.22e-16
Variable Name(s): age education black1 married1 nodegree1 re74 re75  Number(s): 1 2 3 5 6 7 8

After Matching Minimum p.value: < 2.22e-16
Variable Name(s): re74  Number(s): 7
```

This is much better than the previous matching attempt. We then calculate the mean treatment effect and its confidence interval for this matched sample.

```
#Matched treatment effect for all covariates
m_mean_control_all<-mean(Matched_all$re78[Matched_all$treat==0])
m_mean_experimental_all<-mean(Matched_all$re78[Matched_all$treat==1])
m_expe_effect_all<-m_mean_experimental_all-m_mean_control_all
cat("Matched Treatment Effect:",m_expe_effect_all)

#Matched confidence interval for all covariates
m_linreg_all<-lm(re78~treat,data=Matched_all)
m_vals_all<-summary(m_linreg_all)
m_confi_95_all<-c(m_vals_all$coefficients[2,1]-1.96*m_vals_all$coefficients[2,2],
                  m_vals_all$coefficients[2,1]+1.96*m_vals_all$coefficients[2,2])
cat("95% Confidence Interval:",m_confi_95_all)
```

```
[O]:
Matched Treatment Effect: 1928.685
95% Confidence Interval: 669.8559 3187.514
```

This is a remarkably good results, which is very near to the experimental benchmark. Our matched sample achieve good balance across all covariates, so we know it is representative for the counterfactual of the treatment. However, this is by no means as good as the experimental data, since it does not control for unobserved covariates like randomization.

4 Genetic Matching Treatment Effect

Regular matching might provide decent balance but it favors achieving overall balance in general, so it might sacrifice some covariates in order to achieve a higher balance on the others. We do know one method that favors the worst-balanced covariate: genetic matching.

4.1 Propensity Scores Only

Using genetic matching for the estimated propensity scores, we will receive a slightly worst balance than regular matching with propensity scores only.

```
#1.GenMatch() using only propensity scores

gen_match<-GenMatch(Tr=Tr,X=scores,BalanceMatrix=scores,estimand="ATT", M=1,
                    pop.size=16, max.generations=10, wait.generations=1)
gen_out<-Match(Y=Y,Tr=Tr, X=scores, M=1,estimand='ATT',
               Weight.matrix=gen_match)

Matched_gen<-data.frame(gen_out$mdata$Y,gen_out$mdata$Tr)
names(Matched_gen)<-c('re78','treat')
MatchBalance(treat~.-re78, data=Non_expe,match.out=gen_out, nboots=500)
```

```
[O]:
Before Matching Minimum p.value: < 2.22e-16
Variable Name(s): age education black1 married1 nodegree1 re74 re75  Number(s): 1 2 3 5 6 7 8

After Matching Minimum p.value: < 2.22e-16
Variable Name(s): age education re74 re75  Number(s): 1 2 7 8
```

This looks pretty similar to the propensity scores regular matching. This is because when we only match on propensity scores with an attempt to balance only on these scores (BalanceMatrix=scores), genetic matching will reduces into regular propensity scores matching because the weight of other covariates is now 0. The treatment effect for propensity scores matching with GenMatch() is better than the regular matching.

```
#Propensity scores genetic matching treatment effect
mean_control_gen<-mean(Matched_gen$re78[Matched_gen$treat==0])
mean_experimental_gen<-mean(Matched_gen$re78[Matched_gen$treat==1])
expe_effect_gen<-mean_experimental_gen-mean_control_gen
cat("Matched Treatment Effect:",expe_effect_gen)

#Propensity scores genetic matching confidence interval
linreg_gen<-lm(re78~treat,data=Matched_gen)
vals_gen<-summary(linreg_gen)
confi_95_gen<-c(vals_gen$coefficients[2,1]-1.96*vals_gen$coefficients[2,2],
                      vals_gen$coefficients[2,1]+1.96*vals_gen$coefficients[2,2])
cat("95% Confidence Interval:",confi_95_gen)
```

```
[O]:
Matched Treatment Effect: 1812.853
95% Confidence Interval: 1062.245 2563.461
```

4.2 All Covariates And Propensity Scores

Using all covariates and propensity scores for the genetic matching procedure, we get better balance than before.

```
gen_match_all<-GenMatch(Tr=Tr,X=X_all,BalanceMatrix=X_all,estimand="ATT", M=1,
                        pop.size=500, max.generations=20, wait.generations=5)
gen_out_all<-Match(Y=Y,Tr=Tr, X=X_all, M=1,estimand='ATT',
                  Weight.matrix=gen_match_all)

Matched_gen_all<-data.frame(gen_out_all$mdata$Y,gen_out_all$mdata$Tr)
names(Matched_gen_all)<-c('re78','treat')
MatchBalance(treat~.-re78, data=Non_expe,match.out=gen_out_all, nboots=500)
```

```
[O]:
Before Matching Minimum p.value: < 2.22e-16
Variable Name(s): age education black1 married1 nodegree1 re74 re75  Number(s): 1 2 3 5 6 7 8

After Matching Minimum p.value: 0.086
Variable Name(s): re74  Number(s): 7
```

The results are better than regular matching because genetic matching relies on genetic algorithms: it randomly mutates generations after generations of the units in the search space, trying to achieve the best possible solutions. Therefore, by random chance (and in this case the huge search space of 16177 treated and control units in the non-experimental data) it will have a smaller probability of achieving the best possible balance at the end, compared to a direct maximization method like regular matching. However, if we increase the pop.size, the max.generations and the wait.generations arguments, we will increasingly get better results as we allow the genetic algorithm to search longer for a bigger population. This, however, is also the disadvantage of genetic matching: computational resource expense.

This is also understandable because of the large search space problem we discussed earlier in 4.1. Since the potential pool of control units to choose from, pop.size=500 is too small (compared to 16774 units) for the genetic algorithm to deplete the search space in 20 generations. Even though we do not have enough computational resources to perform a bigger search (genetic matching with the above arguments took me 3 hours), we are highly confident that if we increase the allowed pop.size and the max.generations arguments, we will achieve even better balance than the regular matching with all covariates.

```r
#Genetic matching treatment effect
mean_control_genall<-mean(Matched_gen_all$re78[Matched_gen_all$treat==0])
mean_experimental_genall<-mean(Matched_gen_all$re78[Matched_gen_all$treat==1])
expe_effect_genall<-mean_experimental_genall-mean_control_genall
cat("Matched Treatment Effect:",expe_effect_genall)

#Genetic matching confidence interval
linreg_genall<-lm(re78~treat,data=Matched_gen_all)
vals_genall<-summary(linreg_genall)
confi_95_genall<-c(vals_genall$coefficients[2,1]-1.96*vals_genall$coefficients[2,2],
                  vals_genall$coefficients[2,1]+1.96*vals_genall$coefficients[2,2])
cat("95% Confidence Interval:",confi_95_genall)
```

```
[O]:
Matched Treatment Effect: 1786.5
95% Confidence Interval: 533.9335 3039.067
```

The results, of course, are not good because we did not achieve great balance with the matching procedures.

All the code used in this report can be found here:
https://gist.github.com/AshNguyen/d46d115373c511e3ca2bac44fba21072