Lab 7: Matching

Sidak Yntiso sgy210@nyu.edu (based on notes by D Dimmery)

March 23, 2020

Matching Big Picture

- Zoom link: https://nyu.zoom.us/j/911298463
- MATCHING IS NOT AN IDENTIFICATION STRATEGY.

Matching Big Picture

- Zoom link: https://nyu.zoom.us/j/911298463
- MATCHING IS NOT AN IDENTIFICATION STRATEGY.

- Heckman, Ichimura, Smith and Todd (1998) decomposition:
- $B = \int_{S_{1X}} E[Y_0|X, D = 1]dF(X|D = 1) - \int_{S_{0X}} E[Y_0|X, D = 0]dF(X|D = 0)$,
  - S_{1X} is the support of X for treated units, $S_X = S_{1X} \cap S_{0X}$

Matching Big Picture

- Zoom link: https://nyu.zoom.us/j/911298463
- MATCHING IS NOT AN IDENTIFICATION STRATEGY.

- Heckman, Ichimura, Smith and Todd (1998) decomposition:
- $B = \int_{S_{1X}} E[Y_0|X, D = 1]dF(X|D = 1) - \int_{S_{0X}} E[Y_0|X, D = 0]dF(X|D = 0)$,
  - S_{1X} is the support of X for treated units, $S_X = S_{1X} \cap S_{0X}$

- $B = B_1 + B_2 + B_3$
  - $B_1 = \int_{S_{1X} \setminus S_X} E[Y_0|X, D = 1]dF(X|D = 1) - \int_{S_{0X} \setminus S_X} E[Y_0|X, D = 0]dF(X|D = 0)(X|D = 0)$,
    - where $S_{1X} \setminus S_X$ is the support of X only observed under D=1
  - $B_2 = \int_{S_X} E[Y_0|X, D = 0](dF(X|D = 1) - dF(X|D = 0))$
  - Matching addresses B_1 and B_2

Matching Big Picture

- Zoom link: https://nyu.zoom.us/j/911298463
- MATCHING IS NOT AN IDENTIFICATION STRATEGY.

- Heckman, Ichimura, Smith and Todd (1998) decomposition:
- $B = \int_{S_{1X}} E[Y_0|X, D = 1]dF(X|D = 1) - \int_{S_{0X}} E[Y_0|X, D = 0]dF(X|D = 0)$,
  - S_{1X} is the support of X for treated units, $S_X = S_{1X} \cap S_{0X}$

- $B = B_1 + B_2 + B_3$
  - $B_1 = \int_{S_{1X} \setminus S_X} E[Y_0|X, D = 1]dF(X|D = 1) - \int_{S_{0X} \setminus S_X} E[Y_0|X, D = 0]dF(X|D = 0)(X|D = 0)$,
    - where $S_{1X} \setminus S_X$ is the support of X only observed under D=1
  - $B_2 = \int_{S_X} E[Y_0|X, D = 0](dF(X|D = 1) - dF(X|D = 0))$
  - Matching addresses B_1 and B_2

- $B_3 = (\int_{S_X} dF(X|D = 1))\bar{B}_{S_X}$
- CIA requires an assumptions to control B_3.
- How could two identical units receive *different* treatments?

Given CIA . . . why not just put covariates in a regression?

Given CIA . . . why not just put covariates in a regression?

- Separating the procedures mean that you can address two types of confounding separately.

1. Different treatment groups may have different chances of getting treated.
2. Different treatment groups may have different baseline (control) potential outcomes.

- A design which addresses both of these options separately is called "doubly robust".
  - Double robustness means that we only have to get ONE of these right for consistent estimation.

Load packages

```
#install packages
#install.packages("MatchIt",type="source)
#install.packages("cem",repos="http://r.iq.harvard.edu", t

try(library('MatchIt'),silent=TRUE)
```

```
## Warning: package 'MatchIt' was built under R version 3.6
```

```
try(library('cem'),silent=TRUE)
```

```
## Warning: package 'cem' was built under R version 3.6.3

## Loading required package: tcltk

## Loading required package: lattice

##
## How to use CEM? Type vignette("cem")
```

Setup dataset

- Lalonde 1986 evaluates the returns a 1976 jobs training program (National Supported Work Demonstration)
- Outcome `re78` is retained earnings in 1978; treatment is the job training program (NT=185).

Setup dataset

- Lalonde 1986 evaluates the returns a 1976 jobs training program (National Supported Work Demonstration)
- Outcome `re78` is retained earnings in 1978; treatment is the job training program (NT=185).

```
data(lalonde,package="MatchIt")
match.data <- subset(lalonde,treat==1)
#notice continuous covariates; subclassification difficult
covs="age+educ+black+hispan+married+nodegree+re74+re75"
base.mod <- lm(paste("re78 ~ treat+",covs,sep=""),lalonde)
coefs <- c(base=coef(base.mod)[2])
```

Estimates

```
coefs
```

```
## base.treat
##   1548.244
```

Covariate Balance

```
trt <- lalonde$treat==1
means <- apply(lalonde[,-1],2,function(x)
  tapply(x,trt,mean)) #estimate means by treat group
sds <- apply(lalonde[,-1],2,function(x)
  tapply(x,trt,sd)) #estimate sds by treat group
rownames(means)<-rownames(sds)<-c("Treated","Control")
t.p <- apply(lalonde[,-1],2,function(x)
  t.test(x[trt],x[!trt])$p.value) #ttest for covariate
```

View Initial Balance

```
round(t(rbind(means,sds,t.p)),3)
```

```
##              Treated  Control  Treated  Control   t.p
## age           28.030   25.816   10.787    7.155 0.003
## educ          10.235   10.346    2.855    2.011 0.585
## black          0.203    0.843    0.403    0.365 0.000
## hispan         0.142    0.059    0.350    0.237 0.001
## married        0.513    0.189    0.500    0.393 0.000
## nodegree       0.597    0.708    0.491    0.456 0.007
## re74        5619.237 2095.574 6788.751 4886.620 0.000
## re75        2466.484 1532.055 3291.996 3219.251 0.001
## re78        6984.170 6349.144 7294.162 7867.402 0.349
```

Exact

- http://gking.harvard.edu/matchit

```
em.match <- matchit(treat~age+educ+black+hispan+married+
                      nodegree+re74+re75,data=lalonde,
                    method='exact')
exact.data <- match.data(em.match) #N=25 observations
head(exact.data[order(exact.data$subclass),c(1,8:12)])
```

```
##          treat re74 re75       re78   weights subclass
## NSW12        1    0    0 17094.640 1.0000000        1
## PSID381      0    0    0 17941.080 0.9230769        1
## NSW25        1    0    0 11163.170 1.0000000        2
## PSID367      0    0    0  2281.610 0.4615385        2
## PSID411      0    0    0  5306.516 0.4615385        2
## NSW29        1    0    0 16218.040 1.0000000        3
```

Formula ATE

```
set.seed(11)
#randomly select treated and control units within subclass
exact.data$id <- paste(exact.data$subclass,
                       exact.data$treat)
rand.units <- unlist(
  lapply(unique(exact.data[,"id"]),function(x)
  sample(rownames(exact.data)[exact.data$id==x], 1) ))
exact.data_deduped<- exact.data[rand.units,]
#subtract treatment group means
diff.in.means =  function(treat,outcome,subclass,x) {
  outcome[treat==1&subclass==x] -
    outcome[treat==0&subclass==x]
}
```

Formula ATE 2

```
#ATE
em_ate = mean(unlist(lapply(
  unique(exact.data_deduped[,"subclass"]),
  function(x) diff.in.means(exact.data_deduped$treat,
                            exact.data_deduped$re78,
                            exact.data_deduped$subclass,x))
#variance
em_var = mean(unlist(lapply(
  unique(exact.data_deduped[,"subclass"]),
  function(x) (diff.in.means(exact.data_deduped$treat,
                             exact.data_deduped$re78,
                             exact.data_deduped$subclass,x)
                - em_ate)^2)))
em_ate; sqrt(em_var)
```

```
## [1] -1724.332
```

```
## [1] 5887.805
```

```
coefs <- c(coefs,exact.ate = em_ate)
```

Estimates

```
coefs
```

```
## base.treat  exact.ate
##   1548.244  -1724.332
```

Regression Model

What seems problematic with that approach?

Regression Model

What seems problematic with that approach?

```
exact.mod <- lm(paste("re78 ~ treat+",covs,sep=""),
                exact.data,weights=exact.data$weights)
summary(exact.mod)$coefficients
```

```
##                Estimate Std. Error    t value   Pr(>|t|)
## (Intercept) -95247.991  39872.453 -2.3888169 0.02687363
## treat        -1106.311   2276.139 -0.4860472 0.63221648
## age           6512.273   2413.715  2.6980288 0.01383754
## educ         -2202.925   1440.236 -1.5295579 0.14178956
## nodegree      8258.394   6373.368  1.2957662 0.20981172
```

```
coefs <- c(coefs,exact.matchit=coef(exact.mod)[2])
```

Where the weights are

- $w_i = 1$ if treated
- $w_i = \frac{\#\text{ total control}}{\#\text{ total treated}} \frac{\#\text{ subclass treated}}{\#\text{ subclass control}}$ if control

Estimates

```
coefs
```

```
##           base.treat           exact.ate exact.matchit.tr
##             1548.244           -1724.332           -1106.
```

Nearest Neighbor

```
nn.match <- matchit(treat~age+educ+black+hispan+married+
                      nodegree+re74+re75,data=lalonde,
                    method='nearest',discard='control',
                    exact=c('nodegree','black'),
                    distance='GAMlogit')
```

```
## Loading required namespace: mgcv
```

```
nn.mod <- lm(paste("re78 ~ treat+",covs,sep=""),
             lalonde,weights=nn.match$weights)
coefs2 <- c(nn.matchit=coef(nn.mod)[2])
```

Estimates

```
coefs
```

```
##           base.treat           exact.ate exact.matchit.tr
##             1548.244           -1724.332           -1106.
```

```
coefs2
```

```
## nn.matchit.treat
##          1740.563
```

CEM

- CEM just creates bins along each covariate dimension (either pre-specified or automatic)
- Units lying in the same strata are then matched together
- Curse of dimensionality means that with lots of covariates, we'll only rarely have units in the same strata.
- What does that mean we're estimating? Is it the ATT?

CEM

- CEM just creates bins along each covariate dimension (either pre-specified or automatic)
- Units lying in the same strata are then matched together
- Curse of dimensionality means that with lots of covariates, we'll only rarely have units in the same strata.
- What does that mean we're estimating? Is it the ATT?

```
cem.match <- cem(treatment="treat",data=lalonde,
                 drop="re78")
cem.match #395 strata
```

```
##              G0  G1
## All         429 185
## Matched      78  68
## Unmatched 351 117
```

```
cem.mod <- lm(paste("re78 ~ treat+",covs,sep=""),
              lalonde,weights=cem.match$w)
coefs2<-c(coefs2,cem=coef(cem.mod)[2])
```

Estimates

```
coefs
```

```
##          base.treat           exact.ate exact.matchit.tr
##            1548.244           -1724.332           -1106.
```

```
coefs2
```

```
## nn.matchit.treat        cem.treat
##        1740.5631         744.2106
```

Tweaking CEM

```
cutpoints <- list(age=c(25,35),educ=c(6,12),
                  re74=c(100,5000),re75=c(100,5000))
cem.tweak.match <- cem(treatment="treat",
                       data=lalonde,
                       drop="re78",cutpoints=cutpoints)
cem.tweak.match
```

```
##              G0  G1
## All         429 185
## Matched     168 160
## Unmatched 261  25
```

```
cem.tweak.mod <- lm(paste("re78 ~ treat+",covs,sep=""),
                    lalonde,weights=cem.tweak.match$w)
coefs2<-c(coefs2,cem.tweak=coef(cem.tweak.mod)[2])
```

Estimates

```
coefs
```

```
##           base.treat            exact.ate exact.matchit.tr
##             1548.244            -1724.332            -1106.
```

```
coefs2
```

```
## nn.matchit.treat         cem.treat  cem.tweak.treat
##        1740.5631          744.2106          -451.7696
```

Mahalanobis Distance

- $(x - \mu)' V^{-1} (x - \mu)$
- In our case, μ corresponds to a given treated unit.
- Mahalanobis distance is a very common distance "metric".
- You can think about it as simple Euclidean distance in a warped feature space (warped according the the inverse variance-covariance matrix)

Mahalanobis Distance

- $(x-\mu)'V^{-1}(x-\mu)$
- In our case, μ corresponds to a given treated unit.
- Mahalanobis distance is a very common distance "metric".
- You can think about it as simple Euclidean distance in a warped feature space (warped according the the inverse variance-covariance matrix)

```
ctl.data <- subset(lalonde,treat==0)
V<-cov(lalonde[,-c(1,ncol(lalonde))])
mahal.dist <- apply(match.data[,-c(1,ncol(match.data))],1,
                    function(x) mahalanobis(
                      ctl.data[,-c(1,ncol(ctl.data))],x,V))
matches <- apply(mahal.dist,2,which.min)
N <- length(matches)
match.data <- rbind(match.data,ctl.data[matches,])
sort(table(apply(mahal.dist,2,which.min)))
```

```
##
##   1  17  23  59  72  95  96  97 112 127 150 158 168 177
```

Evaluate Balance

```
trt.factor <- rep(c("Treat","Control"),c(N,N))
means <- apply(match.data[,-1],2,function(x)
  tapply(x,trt.factor,mean)) #estimate means by treat grou
sds <- apply(match.data[,-1],2,function(x)
  tapply(x,trt.factor,sd)) #estimate sds by treat group
rownames(means)<-rownames(sds)<-c("Treated","Control")
t.p <- apply(match.data[,-1],2,function(x)
  t.test(x[1:N],x[{N+1}:{2*N}])$p.value) #ttest for covari
```

View Matched Balance

```
round(t(rbind(means,sds,t.p)),3)[-9,]
```

```
##              Treated  Control  Treated  Control   t.p
## age           25.546   25.816    8.745    7.155 0.745
## educ          10.443   10.346    1.841    2.011 0.628
## black          0.832    0.843    0.374    0.365 0.779
## hispan         0.059    0.059    0.237    0.237 1.000
## married        0.184    0.189    0.388    0.393 0.894
## nodegree       0.703    0.708    0.458    0.456 0.910
## re74        1871.365 2095.574 4213.141 4886.620 0.637
## re75        1141.974 1532.055 2428.479 3219.251 0.189
```

And Estimate ATT

```
mahal.match.mod <- lm(paste("re78 ~ treat+",covs,sep=""),
                      match.data)
coefs3 <- c(mahal.match=coef(mahal.match.mod)[2])
```

Estimates

```
coefs
```

```
##          base.treat           exact.ate exact.matchit.tr
##            1548.244           -1724.332           -1106.
```

```
coefs2
```

```
## nn.matchit.treat        cem.treat  cem.tweak.treat
##        1740.5631         744.2106         -451.7696
```

```
coefs3
```

```
## mahal.match.treat
##          417.8293
```

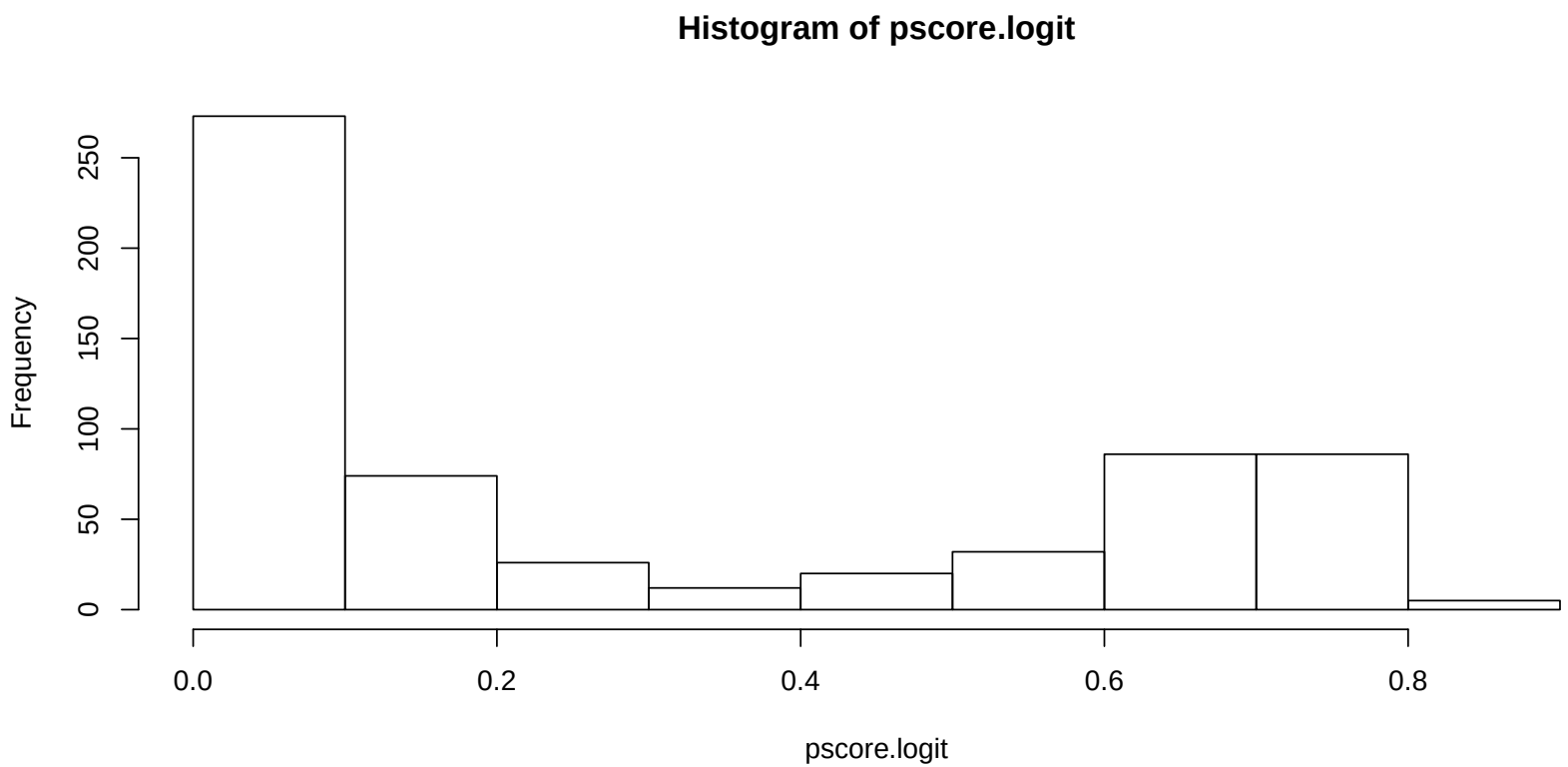
Fitting the Propensity Score

- First, estimate a model of the propensity score.
- (Typically just some logit)

Fitting the Propensity Score

- First, estimate a model of the propensity score.
- (Typically just some logit)

```
p.model <- glm(paste("treat~",covs,sep=""),
               lalonde,family="binomial")
pscore.logit <- predict(p.model,type="response")
hist(pscore.logit)
```



Estimate Model

- What do you want to estimate? This will change the appropriate weights.
- For ATT, sampling probability for treated units is 1.

Estimate Model

- What do you want to estimate? This will change the appropriate weights.
- For ATT, sampling probability for treated units is 1.

```
ipw.logit <- trt + (1-trt)/(1-pscore.logit)
ipw.logit.mod <- lm(paste("re78 ~ treat+",covs,sep=""),
                    lalonde,weights=ipw.logit)
##ATT estimates
coefs3 <- c(coefs3,
            ipw.logit=coef(ipw.logit.mod)[2])
```

Estimates

```
coefs
```

```
##          base.treat           exact.ate exact.matchit.tr
##            1548.244           -1724.332           -1106.
```

```
coefs2
```

```
## nn.matchit.treat         cem.treat  cem.tweak.treat
##        1740.5631          744.2106          -451.7696
```

```
coefs3
```

```
## mahal.match.treat   ipw.logit.treat
##          417.8293         1331.9846
```

Propensity Score matching

- We don't have to weight, though. We might match, instead.

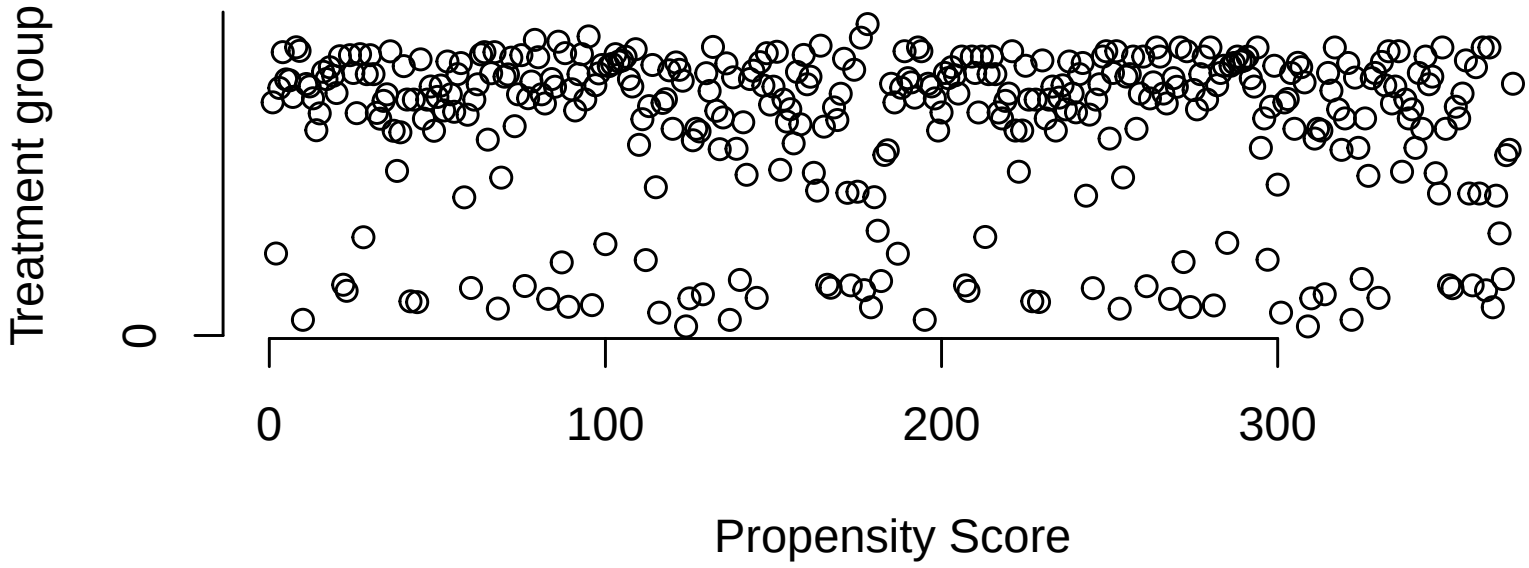
Propensity Score matching

- We don't have to weight, though. We might match, instead.

```
ctl.data <- subset(lalonde,treat==0)
pscore.logit.ctl<-pscore.logit[!trt]
pscore.logit.trt<-pscore.logit[trt]
match.data <- subset(lalonde,treat==1)
matches <- sapply(pscore.logit.trt,function(x)
  which.min(abs(pscore.logit.ctl-x)))
match.data <- rbind(match.data,ctl.data[matches,])
pm.logit.mod<-
  lm(paste("re78 ~ treat+",covs,sep=""),match.data)
```

Estimation and such

```
plot(c(pscore.logit.trt,pscore.logit.ctl[matches]),axes=F,
     ylab="Treatment group",xlab="Propensity Score")
axis(1); axis(2,c(0,1))
```




```
coefs3 <- c(coefs3,pmat.logit=coef(pm.logit.mod)[2])
```

Final Estimates

```
coefs
```

```
##          base.treat          exact.ate exact.matchit.tr
##            1548.244          -1724.332           -1106.
```

```
coefs2
```

```
## nn.matchit.treat        cem.treat  cem.tweak.treat
##        1740.5631         744.2106         -451.7696
```

```
coefs3
```

```
## mahal.match.treat   ipw.logit.treat  pmat.logit.treat
##          417.8293         1331.9846         1963.8733
```