Bob Bell

11.27.2013

PH252D

R Assignment #4

2.1 Calculate the expected counterfactual outcome under each exposure level of a

> set.seed(252)

> W1<- as.integer(runif(n, 1,4) ) # lifestyle 1,2,3

> W2<- rbinom(n, size=1, prob= runif(n, 0.02, 0.7)) # gender

> A<- 1+ rbinom(n, size=6, prob=plogis(0.35 -0.3\*W1 +0.5\*(1-W2) )) #burpees

> U.Y<- rnorm(n, 0, sd=0.01)

> Y<- 30 +1.5\*W1 +3\*log(A)+.3\*(1-W2)\*A + U.Y # happiness

> # the counterfactuals

> Y.1<- 30 +1.5\*W1 +3\*log(1)+.3\*(1-W2)\*1 + U.Y #

> Y.2<- 30 +1.5\*W1 +3\*log(2)+.3\*(1-W2)\*2 + U.Y #

> Y.3<- 30 +1.5\*W1 +3\*log(3)+.3\*(1-W2)\*3 + U.Y #

> Y.4<- 30 +1.5\*W1 +3\*log(4)+.3\*(1-W2)\*4 + U.Y #

> Y.5<- 30 +1.5\*W1 +3\*log(5)+.3\*(1-W2)\*5 + U.Y #

> Y.6<- 30 +1.5\*W1 +3\*log(6)+.3\*(1-W2)\*6 + U.Y #

> Y.7<- 30 +1.5\*W1 +3\*log(7)+.3\*(1-W2)\*7 + U.Y #

>

> mean(Y.1)

[1] 33.2125

> mean(Y.2)

[1] 35.48273

> mean(Y.3)

[1] 36.88991

> mean(Y.4)

[1] 37.94374

> mean(Y.5)

[1] 38.80395

> mean(Y.6)

[1] 39.5417

> mean(Y.7)

[1] 40.19493

> Psi.F <- mean(Y.7-Y.1)

> Psi.F

[1] 6.982431

This means that for BART riders completing 7 burpees are – on average – 6.98 units happier than riders who only completed 1 burpee. The insight is that more burpees make life happier (at least for BART riders – not so sure about AC Transit or SF MUNI).

3. Import and explore data set

> ObsData <- read.csv("Rassign4.Fa2013.csv")

> View(ObsData)

> str(ObsData)

'data.frame': 5000 obs. of 4 variables:

$ W1: int 3 3 1 3 3 2 2 2 1 2 ...

$ W2: int 1 1 1 1 0 1 1 1 1 0 ...

$ A : int 2 3 3 1 4 5 3 5 4 5 ...

$ Y : num 36.6 37.8 34.8 34.5 39.8 ...

> n <- nrow(ObsData)

> summary(ObsData)

W1 W2 A Y

Min. :1.000 Min. :0.0000 Min. :1.000 Min. :31.49

1st Qu.:1.000 1st Qu.:0.0000 1st Qu.:3.000 1st Qu.:36.85

Median :2.000 Median :0.0000 Median :4.000 Median :37.83

Mean :2.023 Mean :0.3486 Mean :4.085 Mean :37.92

3rd Qu.:3.000 3rd Qu.:1.0000 3rd Qu.:5.000 3rd Qu.:39.32

Max. :3.000 Max. :1.0000 Max. :7.000 Max. :42.45

strataW1W2<- rep(NA, n)

strataW1W2[ ObsData$W1==1 & ObsData$W2==1] <- 11

strataW1W2[ ObsData$W1==2 & ObsData$W2==1] <- 21

strataW1W2[ ObsData$W1==3 & ObsData$W2==1] <- 31

strataW1W2[ ObsData$W1==1 & ObsData$W2==0] <- 10

strataW1W2[ ObsData$W1==2 & ObsData$W2==0] <- 20

strataW1W2[ ObsData$W1==3 & ObsData$W2==0] <- 30

# telling R that these are factors

strataW1W2<- as.factor(strataW1W2)

> table(ObsData$A, strataW1W2)

strataW1W2

10 11 20 21 30 31

1 2 6 7 20 25 40

2 40 50 56 80 114 132

3 130 124 174 156 281 223

4 266 175 348 172 324 145

5 318 127 320 117 249 63

6 232 57 156 28 99 12

7 69 10 36 5 11 1

At level a=1, strata 10, 11, 20 are sparse.

At level a=6, strata 31 is sparse.

At level a=7, strata 11, 21, 30, and 31 are sparse.

In short, though we have not violated positivity in a strict sense, the sparse cells will make it difficult for IPTW estimators later on, where the weights for the treatment assignment will potentially be a lot higher. Thus we have “near” violations of the positivity assumption.

4. IPTW for ATE

> library("nnet")

> gAW.reg<-multinom(A~ W1+W2, data=ObsData)

# weights: 28 (18 variable)

initial value 9729.550745

iter 10 value 8205.693797

iter 20 value 8032.285228

final value 8018.904637

converged

> gAW.reg

Call:

multinom(formula = A ~ W1 + W2, data = ObsData)

Coefficients:

(Intercept) W1 W2

2 3.069348 -0.4984121 -0.4822202

3 4.526567 -0.6840646 -0.8728118

4 5.760546 -1.0352705 -1.4050367

5 6.137191 -1.2529254 -1.8449198

6 6.136544 -1.5833200 -2.4323447

7 5.332889 -1.9769620 -2.8351344

Residual Deviance: 16037.81

AIC: 16073.81

> for (i in 1:7){

+ gAW[ObsData$A==i] <- gAW.pred[ObsData$A==i, as.character(i)]

+ }

> summary(gAW)

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.002224 0.176000 0.251200 0.228100 0.299400 0.341700

Some of these covariate combinations have little variability in burpee exposure.

> wt <- 1/gAW

> summary(wt)

Min. 1st Qu. Median Mean 3rd Qu. Max.

2.927 3.340 3.980 6.934 5.683 449.700

When we calculate the weights, we see how they vary tremendously, from under 3 to over 449. This “near” positivity violation means that at least one BART rider is upweighted almost 450 times.

> IPTW<- mean(wt\*as.numeric(ObsData$A==7)\*ObsData$Y) - mean(wt\*as.numeric(ObsData$A==1)\*ObsData$Y)

> IPTW

[1] 7.531834

7.5 units (as calculated with IPTW) is somewhat different that 6.98 units (as calculated by correctly specified model). Ideally, we would want a better estimate (with lower bias ).

> Stab.IPTW <- mean( wt\*as.numeric(ObsData$A==7)\*ObsData$Y)/mean( wt\*as.numeric(ObsData$A==7)) -mean( wt\*as.numeric(ObsData$A==1)\*ObsData$Y)/mean( wt\*as.numeric(ObsData$A==1))

> Stab.IPTW

[1] 6.855816

Here, we get a better estimate for stabilized IPTW estimand than we do the regular IPTW estimand. Minimizing the variance in the weights (with the Horvitz-Thompson estimator) helps us get a better estimand.

5. IPTW & Marginal Structural Models

IPTW for MSM parameter without stabilized weights

IPTW.msm<- glm(Y~A\*W1\*W2, weights=wt, data=ObsData )

> IPTW.msm

Call: glm(formula = Y ~ A \* W1 \* W2, data = ObsData, weights = wt)

Coefficients:

(Intercept) A W1 W2 A:W1 A:W2

30.23265 1.16872 1.40503 -0.23554 0.01895 -0.26149

W1:W2 A:W1:W2

0.08308 -0.01233

Degrees of Freedom: 4999 Total (i.e. Null); 4992 Residual

Null Deviance: 238500

Residual Deviance: 8979 AIC: 9116

Treating this MSM as the truth, we estimate that for every additional burpee, there is an increase in 1.16872 units ofhappiness.

6. Weight stabilization in IPTW for a MSM parameter

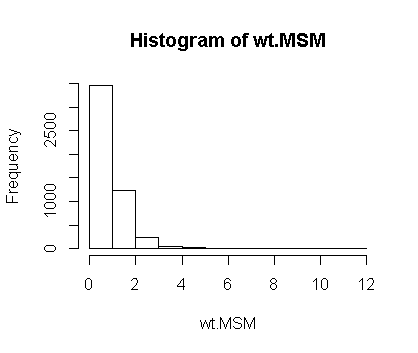
gA<- rep(NA, n)

for (i in 1:7){

gA[ObsData$A==i] <- mean(ObsData$A==i)

}

> hist(wt.MSM)



> summary(wt.MSM)

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.2904 0.7755 0.9505 0.9990 1.0460 11.8700

This is a much better distribution of the weights.

# 4. estimate the parameters of the MSM with IPTW

IPTW.msm.st<- glm(Y~A\*W1\*W2, weights=wt.MSM, data=ObsData )

> IPTW.msm.st$coef

(Intercept) A W1 W2 A:W1 A:W2

30.718230534 1.114328039 1.476906407 -0.024960390 0.005032942 -0.297991512

W1:W2 A:W1:W2

-0.002841273 0.002814627

Treating this MSM as the truth, we estimate that for every additional burpee, there is an increase in 1.1143 units of happiness.

Let’s compare this with what was computed above for IPTW.msm

Coefficients:

(Intercept) A W1 W2 A:W1 A:W2

30.23265 1.16872 1.40503 -0.23554 0.01895 -0.26149

W1:W2 A:W1:W2

0.08308 -0.01233

Degrees of Freedom: 4999 Total (i.e. Null); 4992 Residual

Null Deviance: 238500

Residual Deviance: 8979 AIC: 9116

Are the estimated parameters the same? No, not quite. Comparing the coefficients from both, A:W1:W2, W1:W2, A:W1, and W2 have significantly bigger coefficients. However, the coefficients for A in both MSM’s are relatively similar.

**HW 4.b**

2. Implement TMLE for G-Comp Estimand

set.seed (252)

FullData <- read.csv("Rassign4.Fa2013.csv")

ObsData <- FullData[FullData$A==1 | FullData$A==7,]

n <- nrow(ObsData)

n

[1] 232

A.binary <- as.numeric(ObsData$A==7)

table(A.binary, ObsData$A)

A.binary 1 7

0 100 0

1 0 132

ObsData <- data.frame(ObsData, A.binary)

W <- subset(ObsData, select=c(W1,W2))

# run the tmle package

out<- tmle(Y=ObsData$Y, A=ObsData$A.binary, W=W)

> summary(out)

Initial estimation of Q

Procedure: SuperLearner

Model:

Y ~ SL.glm\_All + SL.step\_All + SL.glm.interaction\_All

Coefficients:

SL.glm\_All 0

SL.step\_All 0.0001293811

SL.glm.interaction\_All 0.9998706

Estimation of g (treatment mechanism)

Procedure: SuperLearner

Model:

A ~ SL.glm\_All + SL.step\_All + SL.glm.interaction\_All

Coefficients:

SL.glm\_All 1

SL.step\_All 0

SL.glm.interaction\_All 0

Estimation of g.Z (intermediate variable assignment mechanism)

Procedure: No intermediate variable

Estimation of g.Delta (missingness mechanism)

Procedure: No missingness

Bounds on g: ( 0.025 0.975 )

Additive Effect

Parameter Estimate: 7.0027

Estimated Variance: 0.0032201

p-value: <2e-16

95% Conf Interval: (6.8915, 7.114)

> names(out)

[1] "estimates" "Qinit" "g" "g.Z"

[5] "g.Delta" "Qstar" "epsilon"

> out$epsilon

H0W H1W

-0.0007003250 0.0004966716

Parameter estimate of 7.00 is very close to 6.98. Actually, it is closer than IPTW estimates (regular and stabilized).

3. Evaluate finite sample performance

# ----------------

# generateData - function to generate the observed data + counterfactuals

# ----------------

generateData<- function(n){

W1<- as.integer(runif(n, 1,4) ) # lifestyle 1,2,3

W2<- rbinom(n, size=1, prob= runif(n, 0.02, 0.7)) # gender

A<- 1+ rbinom(n, size=6, prob=plogis(0.35 -0.3\*W1 +0.5\*(1-W2) )) #burpees

U.Y<- rnorm(n, 0, sd=0.01)

Y<- 30 +1.5\*W1 +3\*log(A)+.3\*(1-W2)\*A + U.Y # happiness

# the counterfactuals

Y.1<- 30 +1.5\*W1 +3\*log(1)+.3\*(1-W2)\*1 + U.Y #

Y.2<- 30 +1.5\*W1 +3\*log(2)+.3\*(1-W2)\*2 + U.Y #

Y.3<- 30 +1.5\*W1 +3\*log(3)+.3\*(1-W2)\*3 + U.Y #

Y.4<- 30 +1.5\*W1 +3\*log(4)+.3\*(1-W2)\*4 + U.Y #

Y.5<- 30 +1.5\*W1 +3\*log(5)+.3\*(1-W2)\*5 + U.Y #

Y.6<- 30 +1.5\*W1 +3\*log(6)+.3\*(1-W2)\*6 + U.Y #

Y.7<- 30 +1.5\*W1 +3\*log(7)+.3\*(1-W2)\*7 + U.Y #

data.frame(W1,W2,A,Y,Y.1, Y.2, Y.3, Y.4,Y.5, Y.6, Y.7)

}

set.seed(252)

R <- 500

estimates<- rep(NA, R)

for(r in 1:R){

# draw a new sample

FullData<- generateData(5000)

NewData <- FullData[FullData$A==1 | FullData$A==7,]

n <- nrow(NewData)

A.binary <- as.numeric(NewData$A==7)

NewData <- data.frame(NewData, A.binary)

# create a data frame of baseline covariates

W<- subset(NewData, select=c(W1,W2))

# run the tmle package

out<- tmle(Y=NewData$Y, A=NewData$A.binary, W=W)

estimates[r] <- out$estimates$ATE$psi

# print(r)

}

> mean(estimates)

[1] 6.993928

> var(estimates)

[1] 0.00344688

> sd <- sqrt(var(estimates))

> sd

[1] 0.05871013

> bias <- mean(estimates) - Psi.F

> bias

[1] 0.01149703

> hist(estimates)

