Analysis

Joshua Hug & Matthew Martinez

12/12/2020

The following chunk loads, cleans the data and assigns to the environment the dataframe with the data as well as a vector with the name of all variables we are using.

```
#setwd("Current_location_of_file")
source("cleaning.R")
source("functions.R")
clean_and_load()
```

Regression Outcome

```
set.seed(11)
covs <- default_covs[c(-1,-2)]

# the regression estimate is done using regression bootstrap function

results <- Regression_bootstrap(z = USDA_new$LA1and10, y = USDA_new$lifeexp, x= USDA_new[covs])

results
```

```
##       Lins.est
## est 0.05388040
## se  0.03012437
```

```
# a confidence interval
c(results[1,1]-1.96*results[2,1], results[1,1]+1.96*results[2,1])
```

```
## [1] -0.005163357  0.112924160
```

Propensity Score

The following code calculates the propensity score using logistic regression and plots a histogram

```
library(ggplot2)
library(magrittr)

covs <- default_covs[c(-1,-2)]

x <- as.matrix(USDA_new[covs])
x <- scale(x)
pred <- USDA_new$LA1and10


# calculate the propensity score
```

```r
prop.logit <- glm(pred ~ x, family = "binomial")

prop_pred <- predict(prop.logit,USDA_new[covs],type = "response")

# assign to the original
USDA_new$prop_scores <- prop_pred

# Uncomment below for the propensity score histogram

# USDA_new %>% ggplot(aes(x=prop_scores))+
#   geom_histogram()+
#   labs(title = "propensity score histogram", x="propensity score value")
```

IPW

We use various truncation methods

```r
set.seed(11)
library(furrr)
```

```
## Loading required package: future
```

```r
covs <- default_covs[c(-1,-2)]

x <- as.matrix(USDA_new[covs])
x <- scale(x)
y <- USDA_new$lifeexp
z <- USDA_new$LA1and10

# a list of truncation levels used
trunc.list = list(trunc0 = c(0,1),
                  trunc.01 = c(0.01, 0.99),
                  trunc.05 = c(0.05, 0.95),
                  trunc.1 = c(0.1, 0.9))

# apply to various truncation levels
# I've used parallelization here through furrr

plan(multisession,workers=parallel::detectCores())


trunc.est_ipw = future_map(trunc.list,
                   function(t){
                     est = ipw.boot(z, y, x, truncpscore = t)
                     round(est, 3)
                   },.options = furrr_options(seed = TRUE))
trunc.est_ipw
```

```
## $trunc0
##         HT Hajek
## est 52.736 0.045
## se  23.223 0.785
##
## $trunc.01
```

```
##        HT  Hajek
## est 4.262 -0.173
## se  0.536  0.052
##
## $trunc.05
##        HT  Hajek
## est 0.490 -0.058
## se  0.239  0.031
##
## $trunc.1
##         HT Hajek
## est -2.521 0.032
## se   0.180 0.028
```

prop score stratification

```
set.seed(11)
covs <- default_covs[c(-1,-2)]

x <- USDA_new[covs]
x <- scale(x)
z <- USDA_new$LA1and10
y <- USDA_new$lifeexp

pscore <- glm(z ~ x, family = binomial)$fitted.values

n.strata = c(5, 10, 20, 50, 80)
strat.res = sapply(n.strata,
                   FUN = function(nn){
                     q.pscore = quantile(pscore, (1:(nn-1))/nn)
                     ps.strata = cut(pscore, breaks = c(0,q.pscore,1),
                                     labels = 1:nn)
                     Neyman_SRE(z, y, ps.strata)
                   })

rownames(strat.res) <- c("est", "se")
colnames(strat.res) <- n.strata
round(strat.res, 3)
```

```
##         5     10     20     50     80
## est 0.001 -0.042 -0.068 -0.073 -0.064
## se  0.038  0.039  0.042  0.050  0.050
```

Doubly Robust

```
set.seed(11)
library(furrr)


covs <- default_covs[c(-1,-2)]

x <- as.matrix(USDA_new[covs])
```

```r
x <- scale(x)
y <- USDA_new$lifeexp
z <- USDA_new$LA1and10

plan(multisession,workers=parallel::detectCores())


trunc.list = list(trunc0 = c(0,1),
                  trunc.01 = c(0.01, 0.99),
                  trunc.05 = c(0.05, 0.95),
                  trunc.1 = c(0.1, 0.9))
trunc.est_dr = future_map(trunc.list,
                   function(t){
                     est = OS_ATE(z, y, x, truncpscore = t)
                     round(est, 3)
                   }, .options = furrr_options(seed = TRUE))
trunc.est_dr
```

```
## $trunc0
##       reg     HT Hajek    DR
## est 0.054 52.736 0.045 2.771
## se  0.033 22.928 0.756 1.723
##
## $trunc.01
##       reg    HT  Hajek    DR
## est 0.054 4.262 -0.173 0.091
## se  0.030 0.524  0.053 0.046
##
## $trunc.05
##       reg    HT  Hajek    DR
## est 0.054 0.490 -0.058 0.046
## se  0.030 0.239  0.031 0.031
##
## $trunc.1
##       reg     HT Hajek    DR
## est 0.054 -2.521 0.032 0.038
## se  0.029  0.185 0.027 0.029
```

Regression Tree

Regression Tree (this takes a very long time to run and so we don't include in knit)

```r
set.seed(11)

library(bartCause)
covs <- default_covs[c(-1,-2)]
x <- USDA_new[covs]
x <- scale(x)
z <- USDA_new$LA1and10
y <- USDA_new$lifeexp

model <- bartc(y,z,x)

summary(model)
```

Covariate balance check

covariate balance check

```
set.seed(11)
library(ggplot2)
library(dplyr)
library(tidyr)
library(furrr)
plan(multisession, workers = parallel::detectCores())

covs <- default_covs[c(-1, -2)]

x <- as.matrix(USDA_new[covs])
x <- scale(x)
y <- USDA_new$lifeexp
z <- USDA_new$LA1and10


## balance check BCHECK is now a list so need to format
Bcheck_all = future_map(
  1:dim(x)[2],
  .f = function(px) {
    OS_ATE(z, x[, px], x, truncpscore = c(0.1, 0.9))
  },
  .options = furrr_options(seed = TRUE)
)

asdf <- data.frame(Bcheck_all)
asdf$type <- c("est", "se")

# regression estimator

clean_reg <-
  asdf %>% pivot_longer(cols = !type) %>% slice(grep("reg", name))
Bcheck_reg <-
  matrix(
    c(
      clean_reg %>% filter(type == "est") %>% pull(value),
      clean_reg %>% filter(type == "se") %>% pull(value)
    ),
    nrow = 2,
    ncol = 7,
    byrow = T
  )

reg <- cov_balance_plot(title= "regression estimator", Bcheck_reg)

# HT estimator

clean_HT <-
  asdf %>% pivot_longer(cols = !type) %>% slice(grep("HT", name))
```

```r
Bcheck_HT <-
  matrix(
    c(
      clean_HT %>% filter(type == "est") %>% pull(value),
      clean_HT %>% filter(type == "se") %>% pull(value)
    ),
    nrow = 2,
    ncol = 7,
    byrow = T
  )

HT <- cov_balance_plot(title= "HT estimator", Bcheck_HT)


# Hajek estimator

clean_hj <-
  asdf %>% pivot_longer(cols = !type) %>% slice(grep("Hajek", name))

Bcheck_hj <-
  matrix(
    c(
      clean_hj %>% filter(type == "est") %>% pull(value),
      clean_hj %>% filter(type == "se") %>% pull(value)
    ),
    nrow = 2,
    ncol = 7,
    byrow = T
  )

Hajek <- cov_balance_plot(title= "Hajek estimator", Bcheck_hj)

# doubly robust


clean_dr <-
  asdf %>% pivot_longer(cols = !type) %>% slice(grep("DR", name))

Bcheck_dr <-
  matrix(
    c(
      clean_dr %>% filter(type == "est") %>% pull(value),
      clean_dr %>% filter(type == "se") %>% pull(value)
    ),
    nrow = 2,
    ncol = 7,
    byrow = T
  )

DR <- cov_balance_plot(title= "Doubly Robust estimator", Bcheck_dr)


#stratfied propensity score for 10 strata
```

```
pscore <- glm(z ~ x, family = binomial)$fitted.values
n <- 10
Bcheck_strat = sapply(1:dim(x)[2],
                FUN = function(px){
                  q.pscore = quantile(pscore, (1:(n-1))/n)
                  ps.strata = cut(pscore, breaks = c(0,q.pscore,1),
                                  labels = 1:n)
                  Neyman_SRE(z, x[, px], ps.strata)
                })

strat <- cov_balance_plot(title= "Stratified estimator", Bcheck_strat)
```

This prints the previous chunks plots

```
library(gridExtra)

grid.arrange(reg,HT,Hajek,strat, ncol = 2)
DR
```