Lab 6: The Bootstrap Method

Sidak Yntiso sgy210@nyu.edu (based on notes by C Samii)

March 09, 2020

Review

- Suppose a random sample of size N from a large population, P.

Review

- Suppose a random sample of size N from a large population, P.
- Bootstrap: the distribution of statistics computed on the samples drawn from $\{X_1, ..., X_N\}$ approximates the distribution of statistics computed on samples from P

Review

- Suppose a random sample of size N from a large population, P.
- Bootstrap: the distribution of statistics computed on the samples drawn from $\{X_1, ..., X_N\}$ approximates the distribution of statistics computed on samples from P
- For $D_i = 1$, weights are $\frac{1}{\hat{p}_i}$. $D_i = 0$, weights are $\frac{1}{1-\hat{p}_i}$

Review

- Suppose a random sample of size N from a large population, P.
- Bootstrap: the distribution of statistics computed on the samples drawn from $\{X_1, ..., X_N\}$ approximates the distribution of statistics computed on samples from P
- For $D_i = 1$, weights are $\frac{1}{\hat{p}_i}$. $D_i = 0$, weights are $\frac{1}{1-\hat{p}_i}$
- Virtually all empirical implementations are semiparametric in the sense that parametric propensity score estimation (using logit or probit) is combined with nonparametric treatment effect estimation (using weighting)

Review

- Suppose a random sample of size N from a large population, P.
- Bootstrap: the distribution of statistics computed on the samples drawn from $\{X_1, ..., X_N\}$ approximates the distribution of statistics computed on samples from P
- For $D_i = 1$, weights are $\frac{1}{\hat{p}_i}$. $D_i = 0$, weights are $\frac{1}{1-\hat{p}_i}$
- Virtually all empirical implementations are semiparametric in the sense that parametric propensity score estimation (using logit or probit) is combined with nonparametric treatment effect estimation (using weighting)
- Even when correctly specified, IPW produces imprecise point estimates in finite datasets if large weights exist (truncate weights in certain circumstances)

Goals

- Use the bootstrap method to estimate the standard error and confidence intervals of the IPTW difference in means

Goals

- Use the bootstrap method to estimate the standard error and confidence intervals of the IPTW difference in means
- Compare the sample, population and analytical distributions of the t-statistic

Goals

- Use the bootstrap method to estimate the standard error and confidence intervals of the IPTW difference in means
- Compare the sample, population and analytical distributions of the t-statistic
- Compute the sample, and populations distributions of other statistics

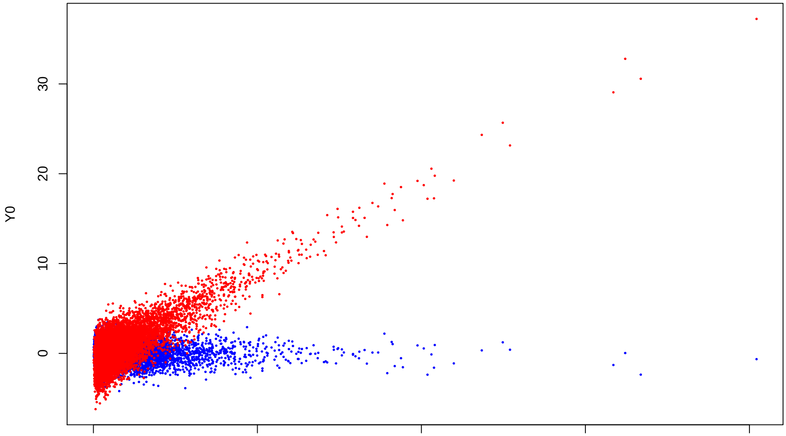
IPTW Example

```r
rm(list=ls())
library(estimatr)
# Make population data
rm(list=ls())
set.seed(11)
N.pop <- 10000
index <- 1:N.pop
X <- .5*exp(rnorm(N.pop))
Y0 <- rnorm(N.pop)
Y1 <- -1 + Y0 + 2*X + rnorm(N.pop)
e.X <- (1+exp(-X))^(-1)
D <- rbinom(N.pop, 1, e.X)
Y <- D*Y1 + (1-D)*Y0
rho <- mean(Y1-Y0)
rho
```

```
## [1] 0.6577518
```

Plot Results

```
plot(X, Y0, col="blue", pch=19, cex=.25,
     ylim=range(c(Y1,Y0)))
points(X, Y1, col="red", pch=19, cex=.25)
```



Sample Data

```
n.samp <- 500 # Draw a sample
samp.i <- sample(index, n.samp) # One case example
samp.data <- pop.data[samp.i,]
# Adjusted/unadjusted regression
summary(lm_robust(Y~D, data=samp.data))$coefficients[,1]
```

```
## (Intercept)           D
##   0.0151522   1.1000326
```

```
summary(lm_robust(Y~D+X, data=samp.data))$coefficients[,1]
```

```
## (Intercept)           D           X
##  -0.9644919   0.3193113   1.8262200
```

IPTW

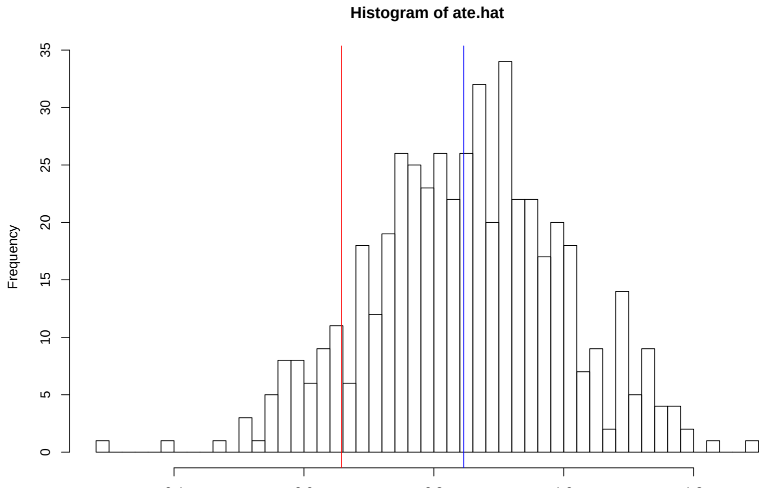
```
# Propensity scores
e.hat.X <- predict(glm(D~X, data=samp.data,
                       family="binomial"), type="response"
#Weights
samp.data$w <- samp.data$D*(1/e.hat.X) +
  (1-samp.data$D)*(1/(1-e.hat.X))
#Model
fit.ipsw.s <- lm_robust(Y~D, weights=samp.data$w,
                        data=samp.data)
```

Get bootstrap estimate

```r
n.boot <- 500
ate.hat <- rep(NA, n.boot)
t.out <- rep(NA, n.boot)
for (i in 1:n.boot){
  boot.index <- sample(samp.data$index, n.samp, replace=T)
  boot.data <- samp.data[match(boot.index,
                               samp.data$index),]
  e.hat.boot <- predict(glm(D~X, data=boot.data,
                            family="binomial"),
                        type="response")
  boot.data$w <- boot.data$D*(1/e.hat.boot) +
    (1-boot.data$D)*(1/(1-e.hat.boot))
  fit.ipsw.b <- lm_robust(Y~D, weights=boot.data$w,
                          data=boot.data)
  ate.hat[i] <- summary(fit.ipsw.b)$coefficients[2,1]
  t.out[i] <- summary(fit.ipsw.b)$coefficients[2,3]
}
```

Plot Bootstrap estimate

```r
hist(ate.hat, breaks=50)
abline(v=coef(fit.ipsw.s)[2], col="blue")
abline(v=rho, col="red")
```



Confidence Intervals

```
# Naive analytical asymptotic CI ignoring pscore estimatio
aaCI <- c(summary(fit.ipsw.s)$coefficient[2,5],
          summary(fit.ipsw.s)$coefficient[2,6])
# Bootstrap-b CI
bbCI <- quantile(ate.hat, c(0.025, .975))
# Bootstrap-t CI
btCI <- summary(fit.ipsw.s)$coefficient[2,2]*
  quantile(t.out, c(0.025, .975))
```

Confidence Intervals

```
coef(fit.ipsw.s)[2]
```

```
##         D
## 0.8459996
```

```
aaCI
```

```
## [1] 0.5547534 1.1372458
```

```
bbCI
```

```
##      2.5%     97.5%
## 0.5649596 1.1341713
```

```
btCI
```

```
##      2.5%     97.5%
## 0.5687471 1.1125184
```

Examine actual sampling distribution

```r
n.iter <- 500
ate.hat.s <- rep(NA, n.iter)
t.out.s <- rep(NA, n.iter)

for(j in 1:n.iter){
  samp.i <- sample(index, n.samp)
  samp.data <- pop.data[samp.i,]
  e.hat.X <- predict(glm(D~X, data=samp.data,
                             family="binomial"),
                         type="response")
  samp.data$w <- samp.data$D*(1/e.hat.X) +
    (1-samp.data$D)*(1/(1-e.hat.X))
  fit.ipsw <- lm_robust(Y~D, weights=samp.data$w,
                             data=samp.data)
  ate.hat.s[j] <- summary(fit.ipsw)$coefficients[2,1]
  t.out.s[j] <- summary(fit.ipsw)$coefficients[2,3]
}
```

Examine actual sampling distribution

```r
mean(ate.hat.s) # True coef mean
```

```
## [1] 0.6805222
```

```r
coef(fit.ipsw.s)[2] # Estimate
```

```
##         D
## 0.8459996
```

```r
sd(ate.hat.s) # True sampling sd
```

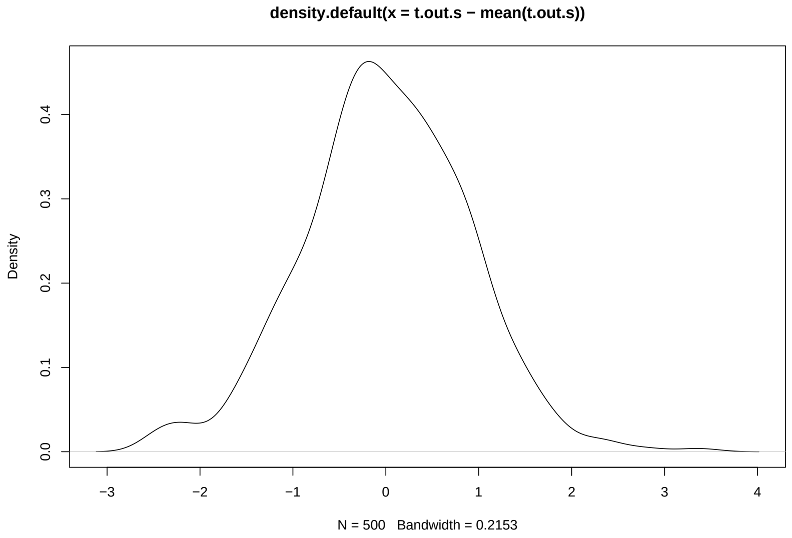
```
## [1] 0.1524192
```

```r
# Estimates
summary(fit.ipsw.s)$coefficient[2,2] #  Naive analytical
```

```
## [1] 0.1482366
```

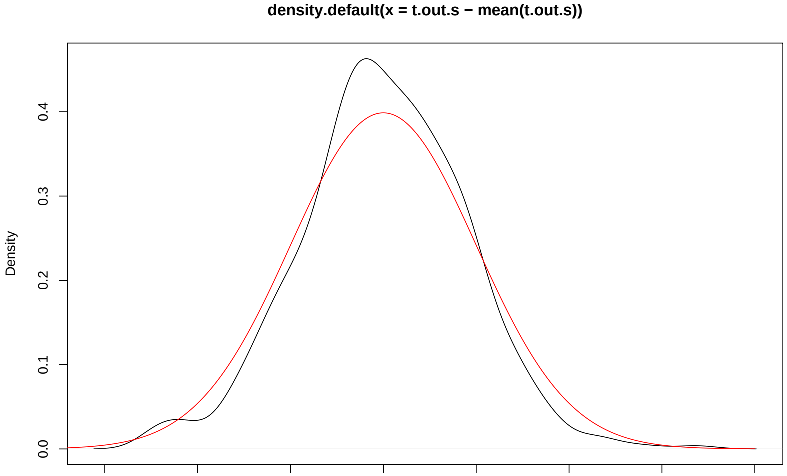
Plot Distributions

```r
plot(density(t.out.s-mean(t.out.s))) # True t stat dist
```

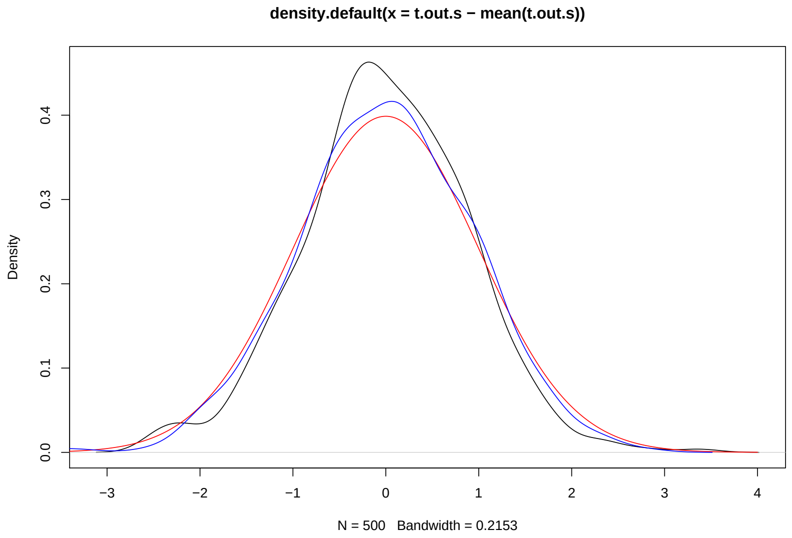


Plot Distributions

```
plot(density(t.out.s-mean(t.out.s))) # True t stat dist
points( seq(-4,4,.01), dt(seq(-4,4,.01), #Analytical
        df=fit.ipsw.s$df.residual),type="l", col="red")
```



Plot Distributions



Exercise 1

Use the pop.data and samp.data for the following questions.

Part A

What is the maximum value of X in the population data? What is the maximum in the sample data?

Part B

Using the non-parametric bootstrap, compute the standard deviation of the maximum of X in the population data.

Part C

Using the non-parametric bootstrap, compute the standard deviation of the maximum of X in the sample data. Does the sample boot-strapped standard deviation approximate the population boot-strapped standard deviation?

Exercise 2

Consider a population of 1000 units. Individual potential outcomes depend on treatment assignment and two stratifiying variables A, B:

$$Y_i(1) = 102 + 3a_i + 2b_i + 6(a_i \times b_i) + \nu_{i1}$$

$$Y_i(0) = 100 + 2a_i + b_i - 2(a_i \times b_i) + \nu_{i0}$$

Where A, B are independent uniform random variables with a minimum of 0.1 and maximum of 1, and $\nu_{i1}, \nu i0$ are independent normal random variables with an expectation of 0 and standard deviation 5. For each individual, y_i is equal to $D_i Y_i(1) + (1 - D_i) Y_i(0)$, where D_i is a Bernoulli distributed random variable.

Exercise 2

Part A

Compute the true ATE weighted by the given propensity scores (`prop.score`). You can use `lm_robust`.

Part B

Compute the unweighted ATE.

Part C

Compute the unweighted ATE conditioning on the observable covariates (A and B). Again, assume we do not know the propensity scores.

Part D

Using the observable covariates, estimate propensity scores for each unit (you can use a logistic regression). Compute the ATE weighted by the estimated propensity scores. How do these point estimates compare to the point estimates from Parts B and C?