Lecture 5. IPW and AIPW

Vinish Shrestha

Towson University

Motivation

Aggregated Estimator

Inverse Probability Weighting (IPW)

Propensity Score

Estimating the propensity score

Cross-fitting

Augmented Inverse Probability Weighting (AIPW)

Discussion

References

Motivation

Motivation

- We've seen the importance of independence assumption $(W_i \perp Y_i(0),\ Y_i(1))$
- However, independence assumption is too strict
- Treatment assignment may depend on baseline characteristics due to selection or treatment target
  - unconfoundedness assumption $(W_i \perp Y_i(0),\ Y_i(1) \mid X_i)$
- Last lecture we estimated ATE for each strata (defined by *highwork*)
- Then aggregated these conditional average treatment effect (CATE) estimates: aggregated estimate $\widehat{\tau}_{agg}$

Aggregated Estimator

Selection into treatment

- *Goal: To evaluate the effect of a tutoring program initiated following the first exam on grades at an introductory level course.*
- However, you are not able to completely randomize the treatment (unethical to force someone not to come)
- For simplicity, possible outcomes are **A** and **B** (GRADES)
- **Treatment Assignment Mechanism: You know that students who received B on their first exam are more likely to attend the tutoring session**

How do we proceed?

Second exam grade conditional on first exam grade

Table 1.	Grade in the 2nd exam $\mid$ 1st exam = A	
	A (2nd Exam)	B (2nd Exam)
Treat	5	2
Control	9	4

Table 2.	Grade in the 2nd exam $\mid$ 1st exam = B	
	A (2nd Exam)	B (2nd Exam)
Treat	15	5
Control	1	4

Aggregating CATE estimates

1. CATE estimate| FE = A:

$$\hat{\tau}_{FE=A} = \frac{5}{7} - \frac{9}{13} = 2.1\ pp$$

2. CATE estimate| FE = B:

$$\hat{\tau}_{FE=B} = \frac{15}{20} - \frac{1}{5} = 55\ pp$$

3. Aggregated estimate:

$$\hat{\tau}_{AGG} = \frac{20}{45} \times \hat{\tau}_{FE=A} - \frac{25}{45} \times \hat{\tau}_{FE=B} = 31.48\ pp.$$

Aggregating CATE estimates

- The first two are CATE estimates for groups receiving grades A and B in the first exam.
- Assumption: Once conditioned on first exam grade, treatment (attendance) is random.
- This enables valid within-group causal effect estimation.
- ATE estimate is formed by averaging CATEs with appropriate weights.
- Example with discrete feature space (grades A or B) shows that if variables influencing treatment are observed, ATE can be estimated by weighting CATEs (group-wise ATEs).

Aggregated Estimator

- The aggregated estimator is given as:

$$\hat{\tau}_{AGG} = \frac{\overbrace{n_A}^{\text{frac. A}}}{n} \underbrace{\left[\frac{1}{n_{A1}} \sum_{\substack{i \in A \\ \text{W} = 1}} Y_i - \frac{1}{n_{A0}} \sum_{\substack{i \in A \\ \text{W} = 0}} Y_i\right]}_{\text{difference in mean for A}} + \frac{\overbrace{n_B}^{\text{frac. B}}}{n} \underbrace{\left[\frac{1}{n_{B1}} \sum_{\substack{i \in B \\ \text{W} = 1}} Y_i - \frac{1}{n_{B0}} \sum_{\substack{i \in B \\ \text{W} = 0}} Y_i\right]}_{\text{difference in mean for B}} \tag{1}$$

$$\hat{\tau}_{AGG} = \frac{1}{n}\left[\frac{1}{\frac{n_{A1}}{n_A}} \sum_{\substack{i \in A \\ \text{W} = 1}} Y_i - \frac{1}{\frac{n_{A0}}{n_A}} \sum_{\substack{i \in A \\ \text{W} = 0}} Y_i\right] + \frac{1}{n}\left[\frac{1}{\frac{n_{B1}}{n_B}} \sum_{\substack{i \in B \\ \text{W} = 1}} Y_i - \frac{1}{\frac{n_{B0}}{n_B}} \sum_{\substack{i \in B \\ \text{W} = 0}} Y_i\right] \tag{2}$$

where,

1. $\frac{n_{A1}}{n_A}$: Represents the fraction of treated individuals who received A on the first exam.

2. $\frac{n_{A0}}{n_A}$: Represents the fraction of untreated individuals who received A on the first exam.

3. $\frac{n_{B1}}{n_B}$: Represents the fraction of treated individuals who received B on the first exam.

4. $\frac{n_{B0}}{n_B}$: Represents the fraction of untreated individuals who received B on the first exam.

Note that ..

- $\frac{n_{A1}}{n_A} = \hat{e}(X_i = A) \approx P(W_i = 1|X_i = A)$
- $\frac{n_{A0}}{n_A} = 1 - \hat{e}(X_i = A \approx 1 - P(W_i = 1|X_i = A))$

Also,

- $\frac{n_{B1}}{n_B} = \hat{e}(X_i = B) \approx P(W_i = 1|X_i = B)$
- $\frac{n_{B0}}{n_B} = 1 - \hat{e}(X_i = B) \approx 1 - P(W_i = 1|X_i = B)$

Probability of being treated conditional upon X

- $P(W_i = 1|X_i) = e(X_i)$ *oracle propensity score*
- re-write:

$$\hat{\tau}_{AGG} = \frac{1}{n}\left[\frac{1}{\hat{e}(X_i = A)} \sum_{\substack{i \in A \\ \text{W} = 1}} Y_i - \frac{1}{1 - \hat{e}(X_i = A)} \sum_{\substack{i \in A \\ \text{W} = 0}} Y_i\right] + \tag{3}$$

$$\frac{1}{n}\left[\frac{1}{\hat{e}(X_i = B)} \sum_{\substack{i \in B \\ \text{W} = 1}} Y_i - \frac{1}{1 - \hat{e}(X_i = B)} \sum_{\substack{i \in B \\ \text{W} = 0}} Y_i\right] \tag{4}$$

- We'll see that this can be written as the inverse probability weighted estimator.

Inverse Probability Weighting (IPW)

IPW

- written as:

$$\hat{\tau}_{IPW} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{Y_i.W_i}{\hat{e}(X_i)} - \frac{Y_i.(1-W_i)}{1-\hat{e}(X_i)}\right) \tag{5}$$

- propensity score, $e(X_i)$, does the balancing act

IPW

- Intuitively, observations with high propensity score within the treated group are weighted down
- Observations with higher propensity score in the control group are weighted more.

In this way, propensity score is used to balance the differences in covariates across the treatment and control groups. Note that the validity of $\hat{\tau}$ still hinges on the unconfoundedness assumption. Any inference that you make is only good if your assumption holds.

Propensity Score

Propensity score

- We saw that in discrete cases one can use aggregated estimator to apply unconfoundedness
    - strata specific ATE and average them
- As the number of covariates increases, this approach is prone to the *curse of dimensionality*
- If features are continuous, we won't be able to estimate ATE at each value of $x \in \chi$ due to lack of enough sample size

Propensity score

Propensity score: $e(x)$. The probability of being treated given a set of covariates Xs.

$$e(x) = P(W_i = 1 | X_i = x) \tag{6}$$

- Note that x (grade A) is the realized value of the covariate X (grade in the 1st exam)
- If unconfoundedness assumption holds, we can write the following:

$$W_i \perp \{Y_i(0),\ Y_i(1)\} |\ e(X_i) (\#eq:pconf) \tag{7}$$

- Instead of conditioning on multi-dimensional vector, we can just condition on $e(X_i)$

Propensity score

- In reality, we won’t often know the propensity score
- We need to estimate it!
- Can use logit or machine learning methods to estimate propensity score
- We’ll take a look at logit, lasso, and random forest approach

Estimating the propensity score

Generate data

```r
library(rsample) # for data splitting
library(caret) # for logistic regression modeling
library(vip) # Model interpretability


set.seed(194) # for replicability
```

Generate data

```r
# Generate simulated Data
n <- 2000 # number of obsevations
p <- 10 # number of covariates

fun_makedat  <- function(n, p) {
X <- matrix(rnorm(n * p), n, p) # data matrix
true_effect  <- 2.5

W <- rbinom(n, 1, 0.1 + 0.4 * (X[, 1] < 0) + 0.2 * (X[, 2] > 0)) # X[, 1]
prob  <- 0.1 + 0.4 * (X[, 1] < 0) + 0.2 * (X[, 2] > 0)  # oracle propensit

Y <- true_effect * W + 2 * X[, 2] + 4 * pmax(X[, 1], 0) + rnorm(n)
#plot(X[, 1], X[, 2], col = as.factor(W))
```

Glimpse data

```r
dat  <- fun_makedat(n = n, p = p)
head(dat)
```

```
##   W          Y          X1         X2         X3         X4          X5
## 1 1  3.2029041 -0.60932297  0.9356439 -0.2188877 -0.9595317 -0.47411361
## 2 0 -1.7942933 -0.11737977 -1.3312857  0.8847310  2.7649549 -0.03876519
## 3 0  0.7971548 -1.98031243  0.2309031 -0.8206863  1.3248576 -0.17210156
## 4 0  4.1376426  0.62555000 -0.1264872  0.4706030  1.3406212  0.89456575
## 5 0 -1.9165507  0.50837441 -1.0433613  0.2654401  1.2094630 -2.24436317
## 6 0 -0.6303680  0.08460611 -0.5654366 -0.3765896 -1.1452287 -0.68846197
##            X6          X7        X8         X9        X10 W_num prob
## 1 -0.21549924 -1.26340278 0.1013767  2.0759335  0.3676577     1  0.7
## 2  1.62228011 -0.45405847 0.2680015  0.4771143 -1.3440617     0  0.5
## 3 -1.29419853  0.30694694 0.5443953 -0.2278483 -1.4717929     0  0.7
## 4  0.06228866  0.50289767 0.1108830  0.5620414  0.6365266     0  0.1
```

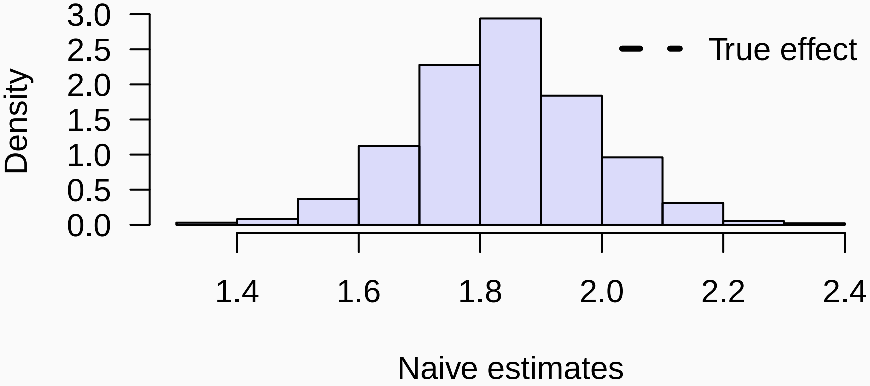
Let's start with the naive estimator

- $\hat{\tau}_{naive} = \frac{1}{n_{W=1}} \sum_{W_i=1} Y_i - \frac{1}{n_{W=0}} \sum_{W_i=0} Y_i$

```r
repl  <- 1000
store_naive  <- rep(0, repl)
store_oracle  <- rep(0, repl)

for(i in seq(repl)) {
    dat  <- fun_makedat(n = n, p = p)
    store_naive[i]  <- mean(dat$Y[dat$W == 1]) - mean(dat$Y[dat$W == 0])
}
```

Histogram of naive estimates

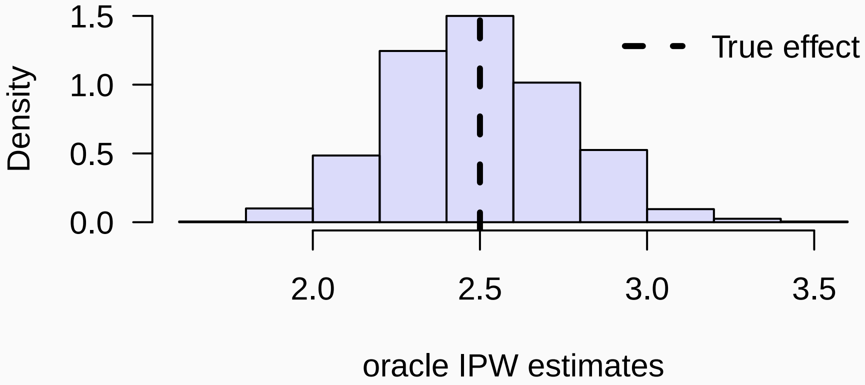


- This is quite off from the true effect of 2.5

Let’s then use the oracle propensity score and estimate IPW

```
for(i in seq(repl)) {
    dat  <- fun_makedat(n = n, p = p)
    Z  <- (dat$W_num * dat$Y / dat$prob) -  ((1 - dat$W_num) * dat$Y / (1-d
    store_oracle[i]  <- mean(Z)
}
```

Histogram of IPW estimates using oracle propensity score

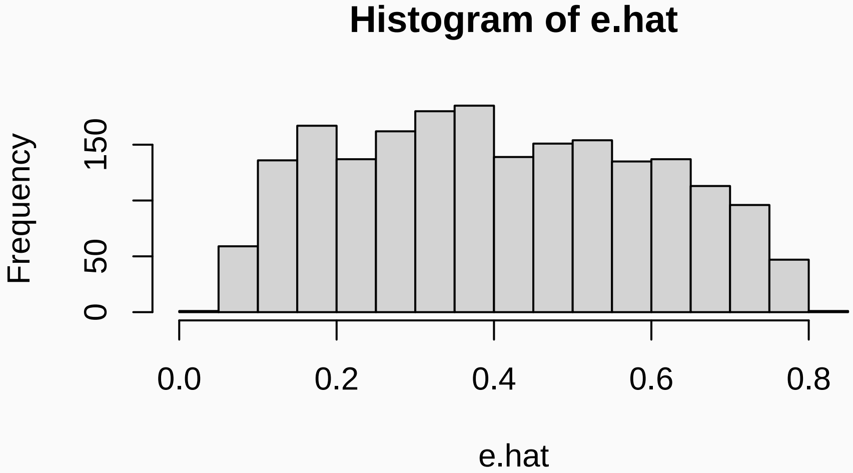


- The histogram of IPW estimates using oracle propensity score is centered around the true effect.

Estimating propensity score: logistic regression

```r
# declare covariates/features
covariates  <- c("X1", "X2", "X3", "X4", "X5", "X6", "X7", "X8", "X9", "X1
fmla  <- as.formula(paste0("~", paste0("bs(", covariates, ", df=3)", colla
W  <- dat[, "W"] # treatment
Y  <- dat[, "Y"] # outcome
XX  <- model.matrix(fmla, dat) # Xs or the features
logit  <- cv.glmnet(x = XX, y = W, family = "binomial")
e.hat  <- predict(logit, XX, s = "lambda.min", type = "response")
```

Histogram of estimated propensity score



Get estimate for IPW estimator

```r
fun_ipw  <- function(W, Y, e.hat){
    # @Arg W: treatment
    # @Arg Y: outcome
    # @Arg e.hat: estimated propensity score
    Z  <- (W * Y / e.hat) -  ((1 - W) * Y / (1-e.hat))
    ipw.est  <- mean(Z)
    ipw.se  <- sd(Z) / sqrt(length(Z))
    ipw.tstat  <- ipw.est / ipw.se
    ipw.results  <- c(estimate = ipw.est, std.error = ipw.se, t.stat = ipw
    return(ipw.results)
}
ipw.results  <- fun_ipw(W = dat$W_num, Y = dat$Y, e.hat = e.hat)
```

Get estimate for IPW estimator

```
##  estimate std.error    t.stat
## 2.7763989 0.2981542 9.3119561
```

- propensity score is estimated using logistic regression

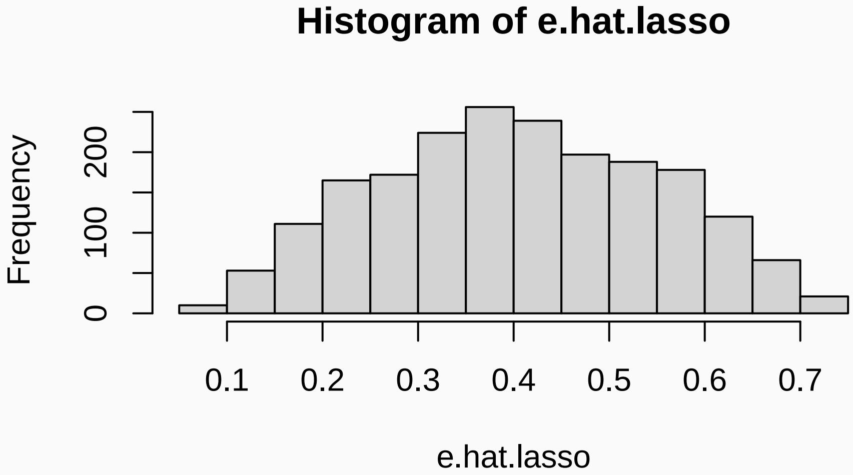
Estimate propensity score using lasso

```
lasso.mod  <- cv.glmnet(
    x = XX,
    y = dat$W_num,
    alpha = 1 # default uses 10 fold cross validation
)

e.hat.lasso  <- predict(lasso.mod, XX, s = lasso.mod$lambda.1se)
```

Histogram

```
hist(e.hat.lasso)
```



Let’s get an estimate for IPW

```
# call the IPW function
ipw.results.lasso  <- fun_ipw(W = dat$W_num, Y = dat$Y, e.hat = e.hat.lass
print(ipw.results.lasso)
```

```
##   estimate  std.error     t.stat
##  2.4943523  0.2393982 10.4192588
```

Estimate propensity score using random forest

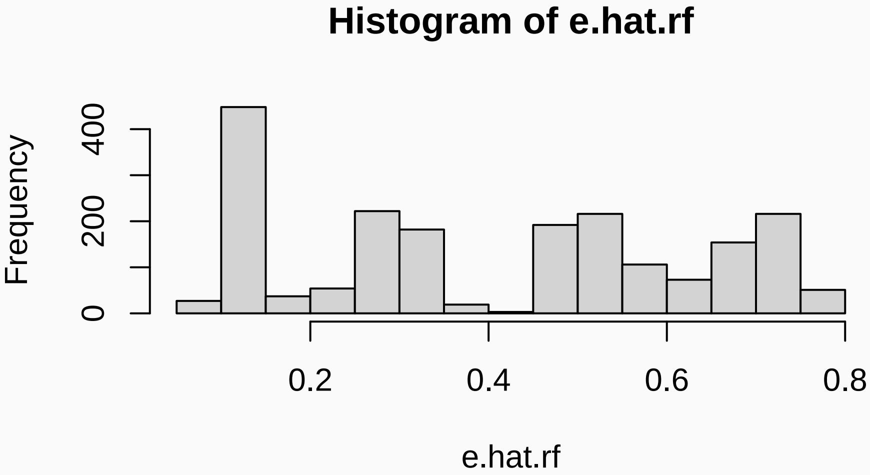
```r
library(grf) # lib for modeling

rf.mod  <-  regression_forest(as.matrix(XX),
                              dat$W_num,
                              honesty = TRUE,
                              num.trees = 10000,
                              tune.parameters = "all")

e.hat.rf  <- predict(rf.mod, XX)$predictions
```

Histogram

```
hist(e.hat.rf)
```



Let’s get an estimate for IPW

```
# call the IPW function
ipw.results.rf  <- fun_ipw(W = dat$W_num, Y = dat$Y, e.hat = e.hat.rf)
print(ipw.results.rf)
```

```
##   estimate  std.error     t.stat
##  2.3093979  0.2263953 10.2007316
```

Cross-fitting

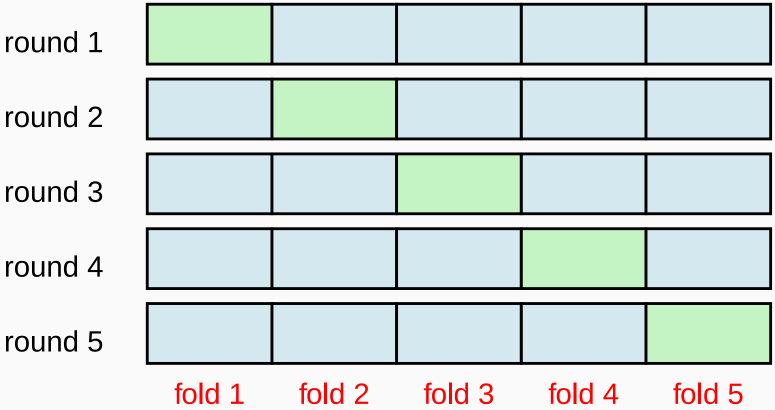
Cross-fitting for training and prediction

- Idea is that the same observation shouldn't be used for training the model as well as making predictions

A Generic Algorithm

1. Divide the data into K folds randomly.
2. Train the model using $-k$ folds (all folds except the k^{th} one).
3. Generate a fit of *fold* k on the model trained using $-k$ folds
4. Repeat steps 2 and 3 to generate fit for all K number of folds.

Cross-fitting illustration



	fold 1	fold 2	fold 3	fold 4	fold 5
round 1	🟩	🟦	🟦	🟦	🟦
round 2	🟦	🟩	🟦	🟦	🟦
round 3	🟦	🟦	🟩	🟦	🟦
round 4	🟦	🟦	🟦	🟩	🟦
round 5	🟦	🟦	🟦	🟦	🟩

Estimating propensity score using cross-fitting

```
dat.e.hat  <- data.frame()
k  <- 10 # k-folds
fold  <- sample(1:k, n, replace = TRUE)

for(i in seq_along(1:k)){
    index  <- which(fold == i)

    lasso.mod  <- cv.glmnet(
            x = XX[-index, ],
            y = dat$W_num[-index],
            alpha = 1 # default uses 10 fold cross validation
    )
    e.hat.lasso  <- predict(lasso.mod, XX[index, ], s = lasso.mod$lambda.1
    dat.e.hat.new  <- data.frame(e.hat.lasso,index)
```

Estimate IPW using the cross-fitted propensity score estimates

```
# call the IPW function but propensity scores are croff-fitted
ipw.results.lasso2  <- fun_ipw(W = dat$W_num,
                               Y = dat$Y,
                               e.hat = e.hat.dat$e.hat.lasso)
print(ipw.results.lasso2)
```

```
##   estimate  std.error     t.stat
##  2.5475174  0.2515263 10.1282336
```

Augmented Inverse Probability Weighting (AIPW)

AIPW

- The other approach to estimate τ is to think of it from the conditional response approach.
- Write $\mu_w(x) = E[Y_i|\ X_i = x, W_i = w]$.

Then:

$\tau(x) = E[Y_i|\ X_i = x, W_i = 1] - E[Y_i|\ X_i = x, W_i = 0]$

- The consistent estimator is formed using the sample counterparts

$\hat{\tau}(x) = N^{-1} \sum_{i=1}^{N} \hat{\mu}_1(X_i) - \hat{\mu}_0(X_i)$

AIPW

- AIPW approach combines both IPW approach as well as regression outcome approach to estimate τ.

$$\hat{\tau}_{AIPW} = \frac{1}{N}\sum_{i=1}^{N}(\hat{\mu}_1(X_i) - \hat{\mu}_0(X_i) + \frac{(Y_i - \hat{\mu}_1(X_i)).W_i}{\hat{e}(X_i)} - \frac{(Y_i - \hat{\mu}_0(X_i)).(1-W_i)}{1-\hat{e}(X_i)})$$

ML approach using cross-fitting is used to estimate both $\hat{e}(x)$ and $\hat{\mu}_w(x)$.

- AIPW approach can be thought of:

1. Estimating ATE taking the difference across conditional responses.
2. Adjusting the residuals using weights given by the propensity score.

AIPW Advantage

1. $\hat{\tau}_{AIPW}$ is consistent as long as $\hat{e}(x)$ or $\hat{\mu}_w(x)$ is consistent.
   - This is because $E[(Y_i - \hat{\mu}_{W_i}(X_i)) \approx 0$.
2. $\hat{\tau}_{AIPW}$ is a good approximation to oracle $\hat{\tau}^*_{AIPW}$ as long as $\hat{\mu}(.)$ and $\hat{e}(.)$ are reasonably accurate.
   - *If one estimate is highly accurate, then it can compensate lack of accuracy on the other estimate. If both $\hat{\mu}(.)$ and $\hat{e}(.)$ are $\sqrt{n}$-consistent, then the following holds.*

$$\sqrt{n}(\hat{\tau}_{AIPW} - \hat{\tau}^*_{AIPW}) \to_p 0.$$

Discussion

Discussion

- We took a look at experimental setting
  1. treatment is completely randomized
  2. treatment is correlated with Xs
- We talked about assumptions
  1. independence, unconfoundedness
  2. SUTVA
  3. overlap
- Took a look at IPW and AIPW
- Discussed that instead of accounting for all Xs for unconfoundedness, one can feasibly account for propensity score $P(W_i = 1|Xs)$.

References

References i