

IPW, P3 Propensity Score Estimation

Henan Xu

0. Introduction of Inverse Propensity Weighting

Propensity score weighting is an alternative to propensity score matching in casual inference. Its idea, in the context of this project, is to directly use propensity scores as inverse weights in calculating the ATE.

Individuals from the treatment group are weighted as $\frac{1}{\hat{e}_i}$, whereas individuals from the control group are weighted as $\frac{1}{1-\hat{e}_i}$, where \hat{e}_i is the estimated propensity score for individual i .

Such an approach addresses some of the disadvantages inherent to propensity score matching. First, it may be impossible to pair each treatment to a different control, or even to any control. Secondly, grouping may include too few controls at high propensities to yield reliable group mean differences. Lastly, groups can be so coarse that the controls and treatments in a group are not well-matched.

1. Setup

```
library(glmnet)
```

2. Load Data

```
data_low <- read.csv("../data/lowDim_dataset.csv")
data_high <- read.csv("../data/highDim_dataset.csv")

# Low Dimension
treatment_low <- data_low$A
y_low <- data_low$Y
x_low <- data_low[, -c(1,2)]

# High Dimension
treatment_high <- data_high$A
y_high <- data_high$Y
x_high <- data_high[, -c(1,2)]
```

3. Calculate Propensity Score with L2 Ridge regression

3.1 Low Dimension

```
set.seed(5243)
glm_low <- cv.glmnet(as.matrix(x_low), treatment_low, family = "binomial", alpha = 0)
```

```
ps_low <- predict(glm_low$glmnet.fit,
                  s = glm_low$lambda.min,
                  newx = as.matrix(x_low),
                  type = "response")
data_low$ps <- ps_low
```

3.2 High Dimension

```
set.seed(5243)
glm_high <- cv.glmnet(as.matrix(x_high), treatment_high, family = "binomial", alpha = 0)

ps_high <- predict(glm_high$glmnet.fit,
                   s = glm_high$lambda.min,
                   newx = as.matrix(x_high),
                   type = "response")
data_high$ps <- ps_high
```

4. Inverse Propensity Weighting

4.1 Calculating inverse propensity weights for low dimension

```
start_time_wt_low <- Sys.time()
data_low$inv_prop_weight <- ifelse(data_low$A == 1, 1/data_low$ps,
                                   1/(1 - data_low$ps))
end_time_wt_low <- Sys.time()
tm_wt_low <- end_time_wt_low - start_time_wt_low
```

4.2 Calculating inverse propensity weights for high dimension

```
start_time_wt_high <- Sys.time()
data_high$inv_prop_weight <- ifelse(data_high$A == 1,
                                   1/data_high$ps,
                                   1/(1 - data_high$ps))
end_time_wt_high <- Sys.time()
tm_wt_high <- end_time_wt_high - start_time_wt_high
```

5. ATE calculated with Inverse Propensity Weighting

5.1 Calculating ATE for low dimension

```
start_time_ate_low <- Sys.time()
data_low_treatment <- data_low[which(data_low$A == 1), ]
data_low_control <- data_low[which(data_low$A == 0), ]
ATE_low <- (sum(data_low_treatment$inv_prop_weight * data_low_treatment$Y) -
```

```

    sum(data_low_control$inv_prop_weight * data_low_control$Y)) /
    dim(data_low)[1]
cat("The ATE calculated by IPW for low dimensional data is", ATE_low, ".")

```

```
## The ATE calculated by IPW for low dimensional data is 2.032712 .
```

```

end_time_ate_low <- Sys.time()
tm_ate_low <- end_time_ate_low - start_time_ate_low

```

5.2 Calculating ATE for high dimension

```

start_time_ate_high <- Sys.time()
data_high_treatment <- data_high[which(data_high$A == 1), ]
data_high_control <- data_high[which(data_high$A == 0), ]
ATE_high <- (sum(data_high_treatment$inv_prop_weight * data_high_treatment$Y) -
    sum(data_high_control$inv_prop_weight * data_high_control$Y)) /
    dim(data_high)[1]
cat("The ATE calculated by IPW for high dimensional data is", ATE_high, ".")

```

```
## The ATE calculated by IPW for high dimensional data is -2.34272 .
```

```

end_time_ate_high <- Sys.time()
tm_ate_high <- end_time_ate_high - start_time_ate_high

```

6. Running time of the algorithm

```

tm_low <- tm_wt_low + tm_ate_low
tm_high <- tm_wt_high + tm_ate_high
cat("The running time of IPW for low dimensional data is", tm_low, "s.\n")

```

```
## The running time of IPW for low dimensional data is 0.004002094 s.
```

```
cat("The running time of IPW for high dimensional data is", tm_high, "s.")
```

```
## The running time of IPW for high dimensional data is 0.005999804 s.
```