Final_Report

Group_7

2020/11/30

Overview

In this project, we are going to use 5 different ways to estimate propensity scores and two algorithms to estimate ATE. They are:

- Logistic Regression + Full Matching(propensity score distance measurement)
- L1 Logistic Reg + Full Matching(propensity score distance measurement)
- L2 Logistic Reg + Full Matching(propensity score distance measurement)
- CART + Full Matching(propensity score distance measurement)
- Boosting Stumps + Full Matching(propensity score distance measurement)
- Logistic Regression + Weighted Regression

Step 0.1 Load Required Packages

```
packages.used <- c("grDevices","glmnet","rpart","gbm","MatchIt","readr","dplyr","reshape2","tidyverse","smotefamily","knitr")
# check packages that need to be installed.
#packages.needed <- setdiff(packages.used,
#                           intersect(installed.packages()[,1],
#                                     packages.used))
# install additional packages
#if(length(packages.needed) > 0){
#  install.packages(packages.needed, dependencies = TRUE)
#}
library(smotefamily)
library(grDevices)
library(glmnet)
library(rpart)
library(gbm)
library(MatchIt)
library(readr)
library(dplyr)
library(reshape2)
library(knitr)
#library(tidyverse)
```

Step 0.2 Import Data

```
path = '../data/'
highdim = read_csv(paste0(path, 'highDim_dataset.csv')) #2000  187

lowdim = read_csv(paste0(path, 'lowDim_dataset.csv')) #475  24
```

1. Propensity Score Estimation

We define the propensity score as:

$$e(x) = Pr(T = 1|X = x)$$

We assume that:

$$0 < e(x) < 1$$

for all x, here we denote X as the covariates of p-dimensional vector of pre-treatment variables.

These following histograms indicate the change of different methods for propensity score estimations without and with oversampling by SMOTE on low dimensional data. All histograms show that treatment group and control group have enough propensity scores overlapped, which indicates that both datasets are qualified to perform Propensity Score Matching and Weighted Regression.

1.1 Without Oversampling for Imbalanced Classification

1.1.1 Estimate by Logistic Regression

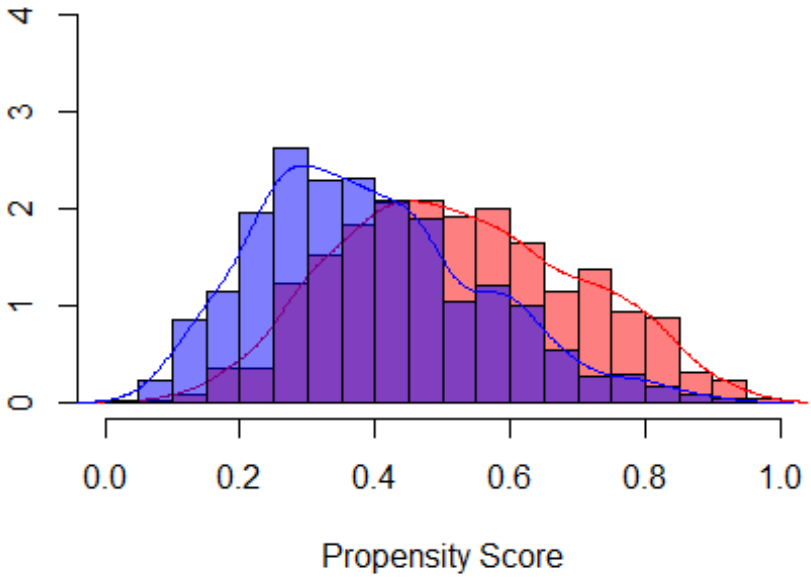
The logistic Regression model represents the class conditional probabilities through a linear function of the predictors:

$$logit[Pr(T = 1|X)] = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$$

$$Pr(T = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p)}}$$

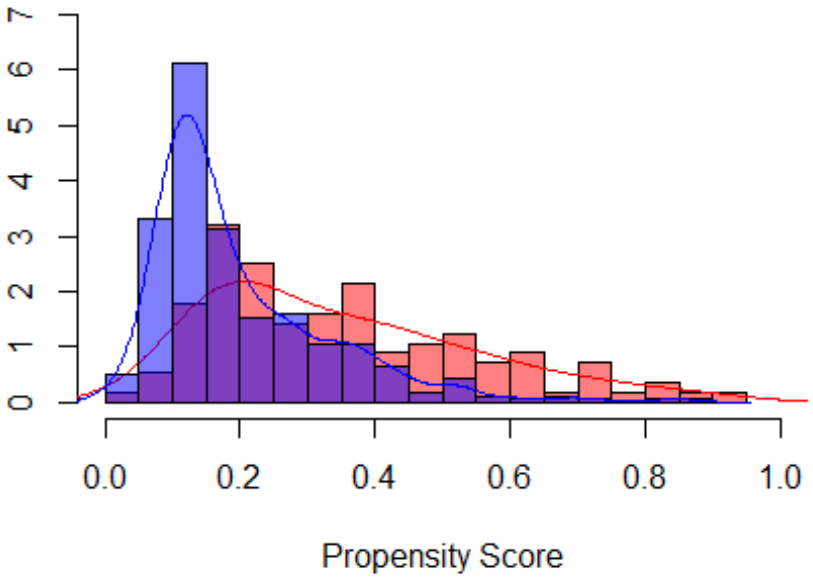
High dimensional data:

```
## Processing time of propensity score estimation by Logistic Regression for high d
imensional data is 0.7914579 seconds.
```



Low dimensional data:

```
## Processing time of propensity score estimation by Logistic Regression for low di
mensional data is 0.03500485 seconds.
```



1.1.2 Estimate by L1 Penalized Logistic Regression

Regularization term is introduced to decrease the model variance in the loss function Q in order to avoid overfitting of logistic regression model. For both L1 and L2 Penalized Logistic Regression, we modifying the loss function with a penalty term which effectively shrinks the estimates of the coefficients.

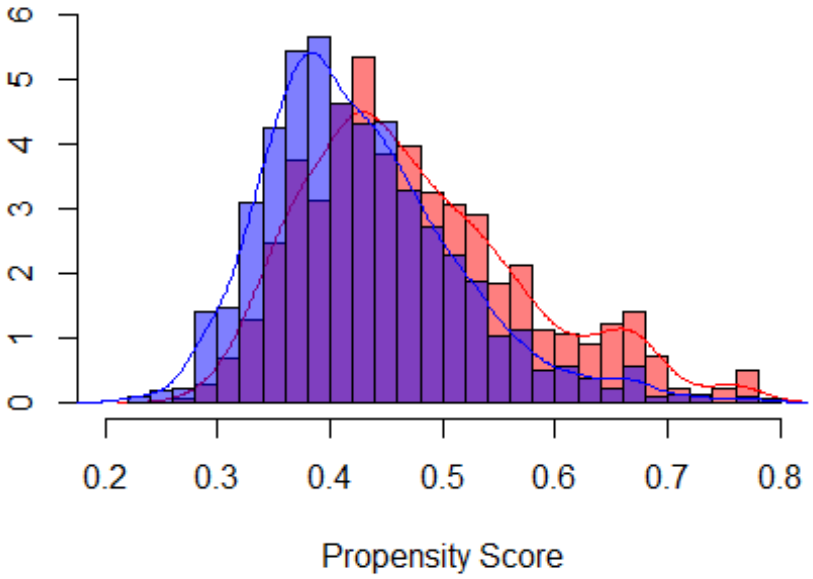
Lasso Regression (Least Absolute Shrinkage and Selection Operator) with L1 norm penalty term, adds "absolute value of magnitude" of coefficient as penalty term to the loss function.

$$Q = -\frac{1}{n}\sum_{i=1}^{n}[\, y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}) + log(1 + exp(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}))] + \lambda \sum_{j=1}^{p} |\, \beta_j|$$

where $Y \in \{0,1\}$

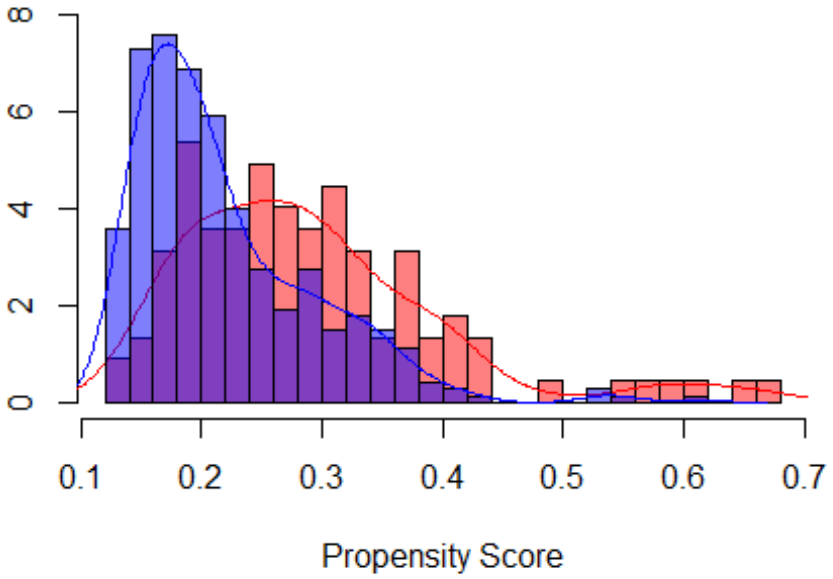
High Dimension Data

```
## Processing time of propensity score estimation by L1 Penalized Logistic Regressi
on for high dimensional data is 0.06709719 seconds.
```



Low Dimension Data

```
## Processing time of propensity score estimation by L1 Penalized Logistic Regressi
on for low dimensional data is 0.01479101 seconds.
```



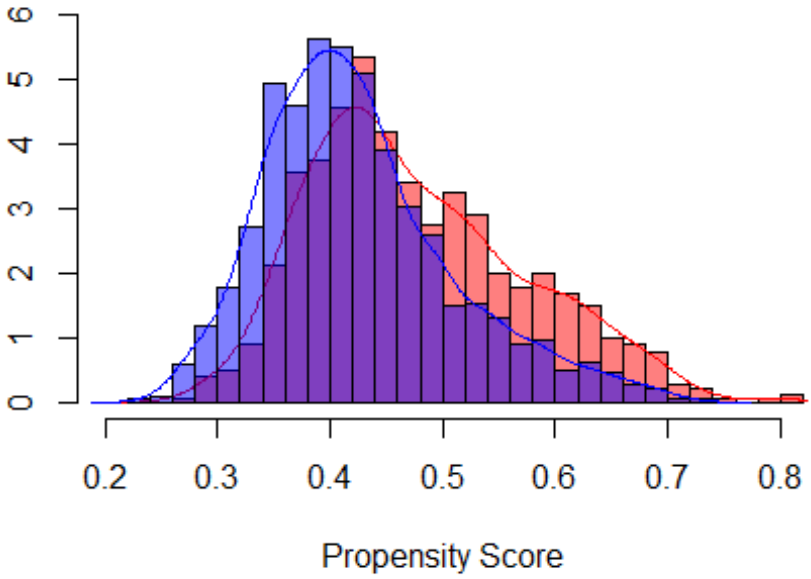
1.1.3 Estimate by L2 Penalized Logistic Regression

Ridge regression with L2 norm penalty term adds "squared magnitude" of coefficient as penalty term to the loss function.

$$Q = -\frac{1}{n}\sum_{i=1}^{n}[\, y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}) + log(1 + exp(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}))] + \lambda \sum_{j=1}^{p} {\beta_j}^2$$

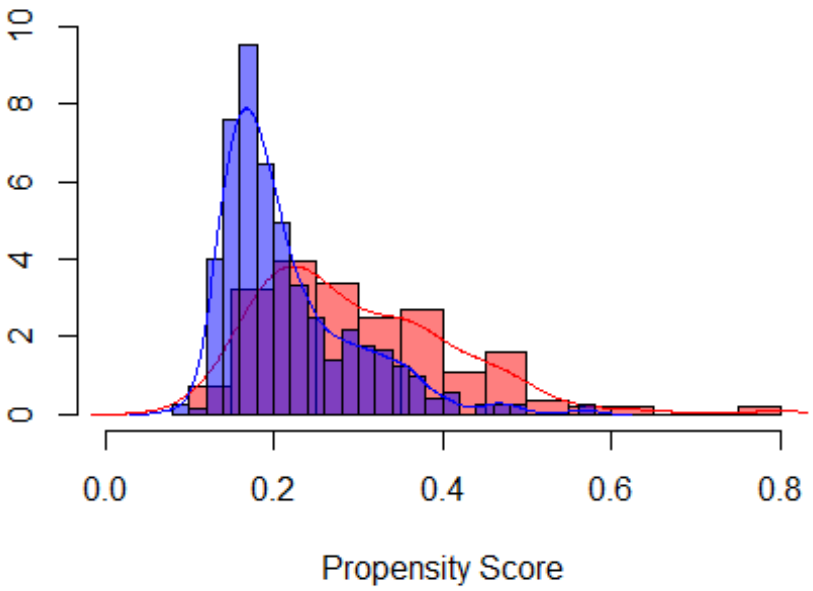
High Dimension Data

```
## Processing time of propensity score estimation by L2 Penalized Logistic Regressi
on for high dimensional data is 0.09896302 seconds.
```



Low Dimension Data

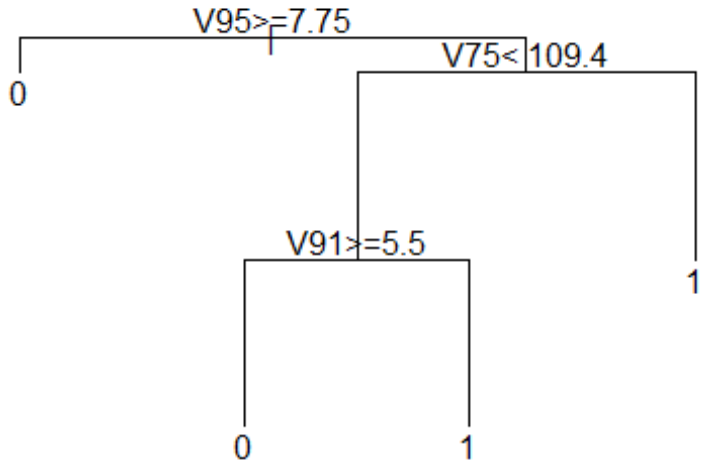
```
## Processing time of propensity score estimation by L2 Penalized Logistic Regressi
on for low dimensional data is 0.09896302 seconds.
```



1.1.4 Estimate by Regression Trees (CART)

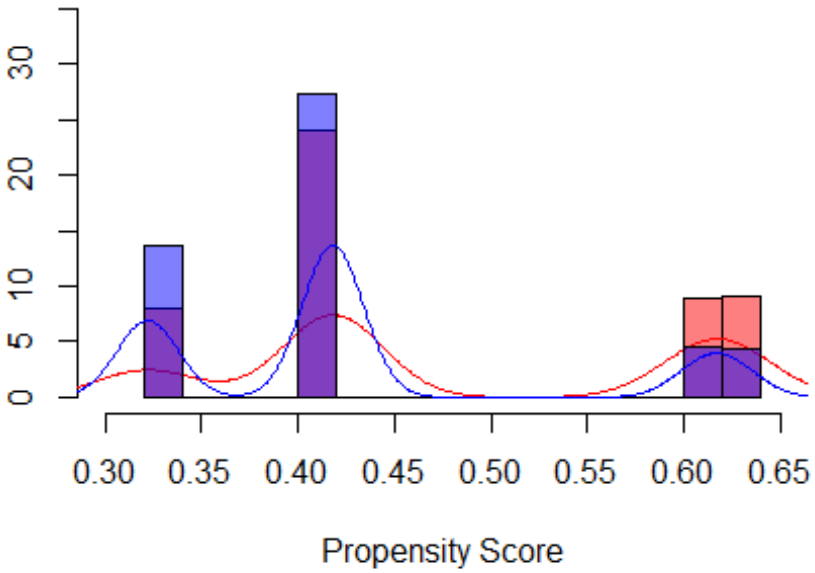
Classification and regression trees could use decision tree model and provide probability of class membership. We first split the space into two regions, and model the response by the mean of Y in each region. We choose the variable and split-point to achieve the best fit. Then one or both of these regions are split into two more regions, and this process in continued, until some stopping rule is applied. The corresponding regression model predicts Y with a constant c_m in region R_m, that is,

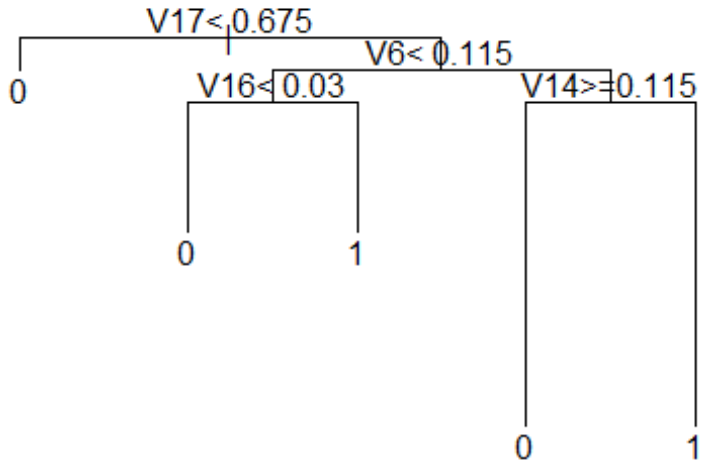
$$\widehat{f(x)} = \sum_{m=1}^{M} c_m \, I\{x \in R_m\}$$



High Dimension Data

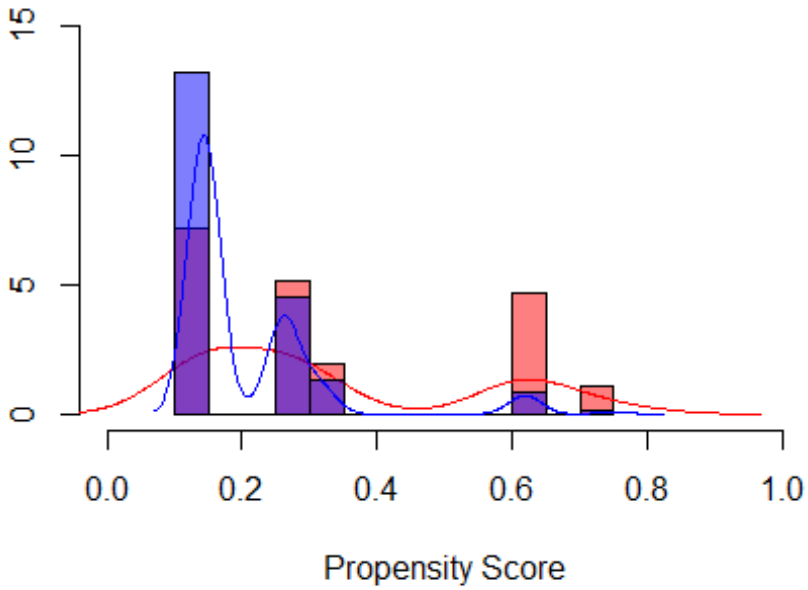
```
## Processing time of propensity score estimation by Regression Trees (CART) for hi
gh dimensional data is 2.565033 seconds.
```





Low Dimension Data

```
## Processing time of propensity score estimation by Regression Trees (CART) for lo
w dimensional data is 0.102999 seconds.
```



1.1.5 Estimate by Boosting Stumps

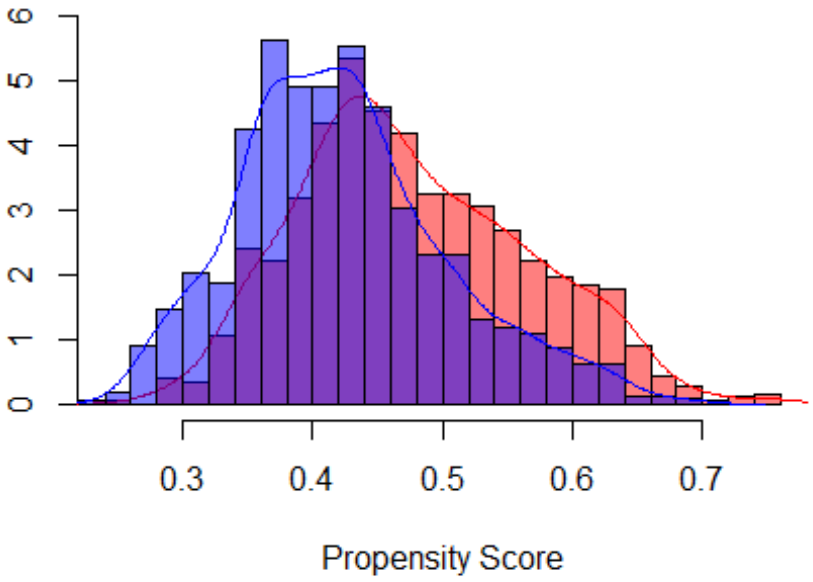
We can represent the boosting stumps model as an addictive model:

$$f_M(x) = \sum_{m=1}^{M} T\,(x; \Theta_m)$$

where $T(x;\theta)$ is the stump, Θ_m is the parameter of the tree stump, M is the number of tree stumps.

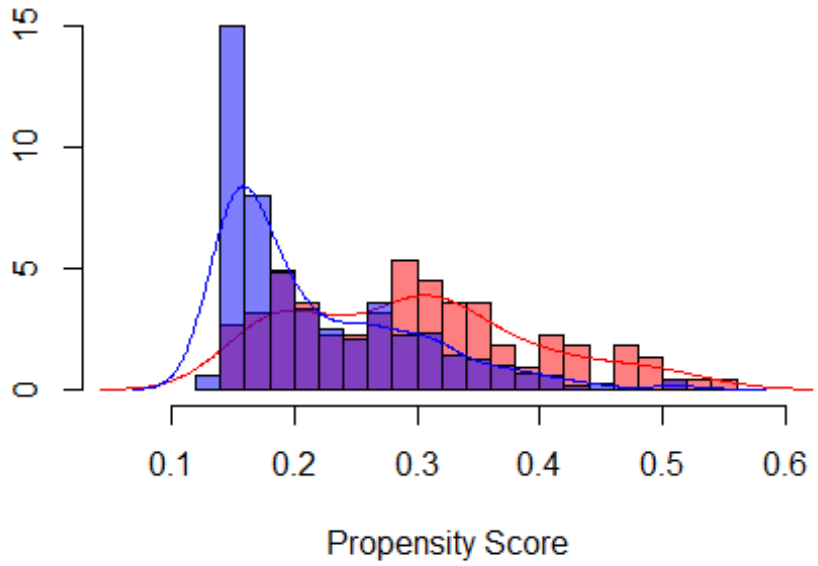
High Dimension Data

```
## Processing time of propensity score estimation by Boosting Stumps for high dimen
sional data is 6.628049 seconds.
```



Low Dimension Data

```
## Processing time of propensity score estimation by Boosting Stumps for low dimens
ional data is 0.133028 seconds.
```



1.2 Oversampling for Imbalanced Classification

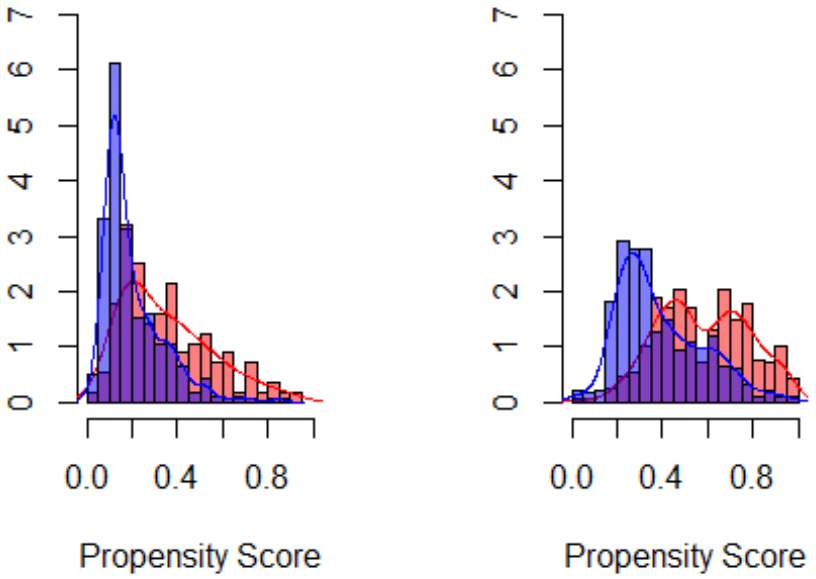
Since the treatment classification (A) ratio is $1103(0): 897(1) = 1.229654$ in high dimension data, and the treatment classification (A) ratio is $363(0): 112(1) = 3.241071$ in low dimension data. We decided to use Synthetic Minority Oversampling Technique to generate synthetic positive instances using SMOTE algorithm only on low dimension data. After oversampling, the treatment classification (A) ratio is $363(0): 336(1) = 1.080357$ in low dimension data.

The reason why we choose to use SMOTE is:

In full matching, reduce observations to similar pairs reduces bias allowing both groups to be equally represented and analyzed using statistical measures of significance to assess improvement for the treatment group. Even though using the propensity score for the match criteria would help dealing with imbalance data, we still want to try other oversampling techniques before full matching. We used Synthetic Minority Over-sampling Technique (SMOTE) on low dimensional data to create synthetic samples and used these generated samples to estimate the propensity scores.

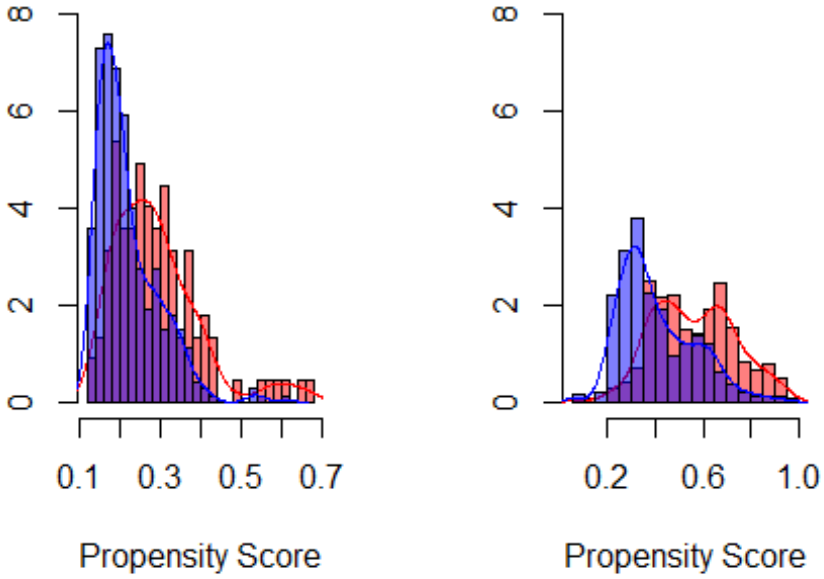
1.2.1 Estimate by Logistic Regression

```
## Processing time of propensity score estimation by Logistic Regression for balanc
ed low dimensional data is 0.03500295 seconds.
```



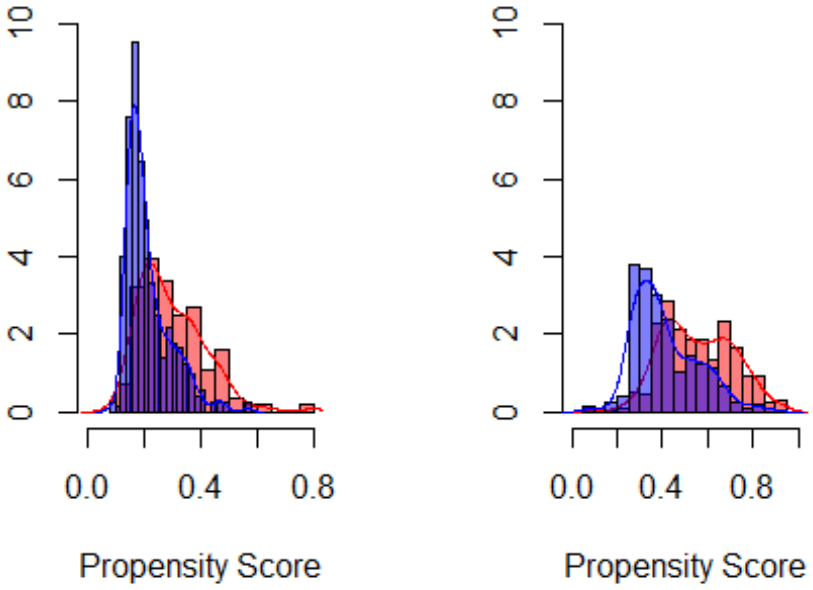
1.2.2 Estimate by L1 Penalized Logistic Regression

```
## Processing time of propensity score estimation by L1 Penalized Logistic Regressi
on for balanced low dimensional data is 0.01796579 seconds.
```

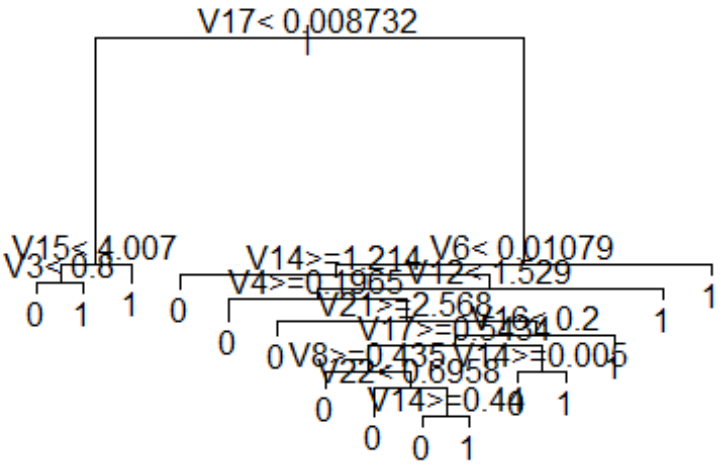


1.2.3 Estimate by L2 Penalized Logistic Regression

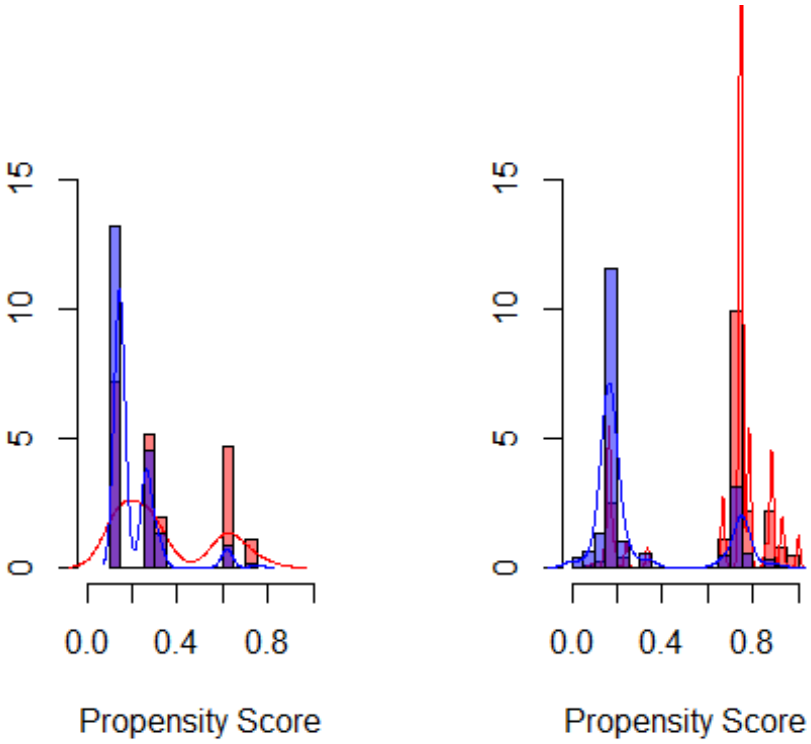
```
## Processing time of propensity score estimation by L2 Penalized Logistic Regressi
on for balanced low dimensional data is 0.01699805 seconds.
```



1.2.4 Estimate by Regression Trees (CART)

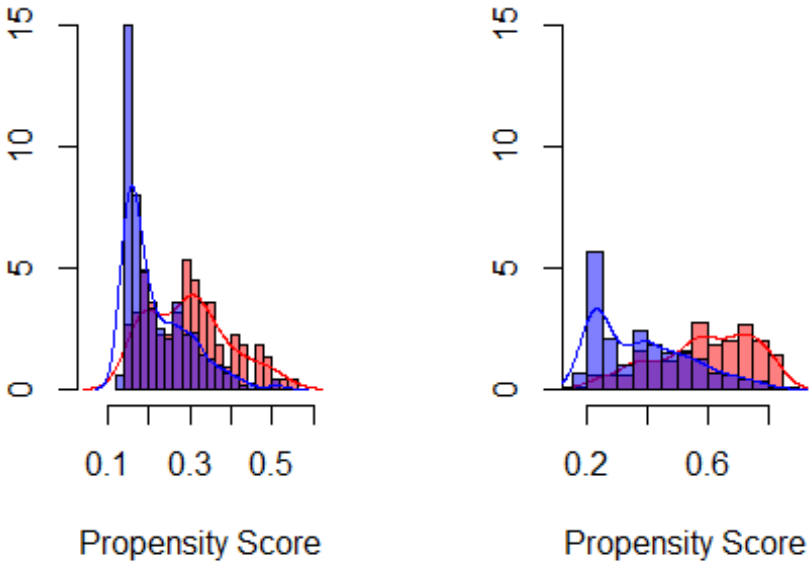



```
## Processing time of propensity score estimation by Regression Trees (CART) for ba
lanced low dimensional data is 0.165025 seconds.
```



1.2.5 Estimate by Boosting Stumps

```
## Processing time of propensity score estimation by Boosting Stumps for balanced l
ow dimensional data is 0.5192671 seconds.
```



2. Propensity Score Matching

After we got propensity score estimations based on different methods, we implemented Full Matching. First, we calculated distances of propensity scores, and then we obtained the matched sets where each matched set contains at least one treated individual and one control individual and these were formed in an optimal way. (s.t. Treated individuals who have many comparison individuals who are similar will be grouped with many comparison individuals). After that, we calculated subclass treatment effects for each matched set and then estimated overall ATE by an weighted average of the subclass effects where weights corresponding to the number of individuals in each subclass.

The results of estimated ATEs are shown as following:

```
##                   Logistic        L1        L2      CART        BS
## highdim          -2.985818 -2.939771 -3.268819 -3.110418 -3.542559
## lowdim            2.855779  2.971267  3.094955  9.701434  2.870900
## balanced lowdim   2.386179  2.628000  2.408261  2.532557  2.268999
```

3. Weighted Regression

We will use the propensity score that estimated by logistic regression in part 1.1.1 and apply weighted regression to estimate ATE. Weighted least square estimation of the regression function:

$$Y_i = \alpha_0 + \tau * T_i + \alpha_1' * Z_i + \alpha_2' * (Z_i - \overline{Z}) * T_i + \epsilon_i$$

The weight w_i is:

$$w_i = \frac{T_i}{\widehat{e_i}} + \frac{1 - T_i}{1 - \widehat{e_i}}$$

where $\widehat{e_i}$ is the estimated propensity score for individual i. This weighting serves to weight both the treated and control groups up to the full sample The Z_i are a subset of the covariates X_i with sample average $\overline{Z}$. τ is an estimate for ATE

3.1 Estimate by Logistic Regression + Weighted Regression

High dimensional data:

First, we need to find Z_i which is the subset of the covariates X_i, we will select Z by estimating linear regressions:

$$Y_i = \beta_{k0} + \beta_{k1} * T_i + \beta_{k2} * X_{ik} + \epsilon_i$$

We calculate the t-statistic for the test of the null hypothesis that the slope coefficient β_{k2} is equal to zero in each of these regressions, and now select for Z all the covariates with a t-statistic larger in absolute value than 1.96. we will only keep covariates with t-values less than 1.96 or larger than 1.96. Therefore, the following covariates do not qualify:

```
##  [1] V1          V2          V4          V5          V8          V9
##  [7] V13         V15         V18         V28         V29         V34
## [13] V36         V39         V41         V42         V43         V44
```

```
## [19] V47        V49        V50        V52        V53        V54
## [25] V55        V57        V58        V59        V97        V151
## [31] weight.ATE
## 189 Levels: Y A V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 ... weigh
t.ATE
```

After deleting the above variables, ATE is estimated by following result, which is around -4.135:

```
## [...]
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     6.876e-01  4.061e-01   1.693  0.09058 .
## A              -4.135e+00  5.817e-01  -7.109 1.72e-12 ***
## [...]
```

Low dimensional data:

For low dimensional data, we follow the same steps, first is to find the subset:

```
## [1] V2  V4  V8  V9  V11 V14 V16 V20
## 26 Levels: Y A V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 ... weight
.ATE
```

After deleting the above variables, ATE is estimated by following result, which is around 2.788:

```
## [...]
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   10.94462    0.14003  78.158  < 2e-16 ***
## A              2.78831    0.21032  13.257  < 2e-16 ***
## [...]
```

Low dimensional balanced data:

For low dimensional data, we follow the same steps, first is to find the subset:

```
## [1] V4         V8         V9         V11        V14        V20        weight.ATE
## 26 Levels: Y V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17 ... weig
ht.ATE
```

After deleting the above variables, ATE is estimated by following result, which is around 2.929:

```
## [...]
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## A              2.92947    0.15824  18.513  < 2e-16 ***
## [...]
```

Summary

ATE results comparison

The following table shows estimated ATEs, we could see that Logistic Regression with Full Matching using propensity score distance measurement performs the best on both datasets. Regression tree with Full Matching using propensity score distance measurement is not well performed on low dimensional data, while the estimated ATE is very close to true ATE value after oversampling by SMOTE.

```
##                 Logistic_FullMatching L1_FullMatching L2_FullMatching
## highdim                     -2.985818       -2.939771       -3.268819
## lowdim                       2.855779        2.971267        3.094955
## balanced lowdim              2.386179        2.628000        2.408261
##                 CART_FullMatching BS_FullMatching Logistic_WeightedReg True_ATE
## highdim                  -3.110418       -3.542559             -4.13500     -3.0
## lowdim                    9.701434        2.870900              2.78831      2.5
## balanced lowdim           2.532557        2.268999              2.92947       NA
```

Propensity score estimation time

The following table shows each method's time of estimating propensity scores. We could see that L1 runs fastest, which followed closely by L2. The reason might be L1 and L2 penalized strictly on dataset, therefore, they will only choose covariates that most related with.

```
##                   Logistic         L1         L2       CART       BS
## highdim         0.39100190 0.03896785 0.05304599 1.48576000 3.217024
## lowdim          0.03999996 0.01000094 0.05304599 0.04700000 0.073946
## balanced lowdim 0.01700902 0.01003218 0.01000118 0.08700299 0.229990
```

ATE estimation time

The following table shows each method's time of estimating ATE. We could see that Weighted Regression performs much better than FullMatching. The reason might because the package MatchIt will be slower while processing large dataset. In the future, we could try different packages to see if the running time can be improved.

```
##                 Logistic_FullMatching L1_FullMatching L2_FullMatching
## highdim                     4.5569811        2.277723        2.646396
## lowdim                      1.3001111        0.355448        0.316005
## balanced lowdim             0.4705532        0.545321        0.566998
##                 CART_FullMatching BS_FullMatching Logistic_WeightedReg
## highdim                 2.8312519        2.157993                  0.49
## lowdim                  0.2040021        0.292038                  0.02
## balanced lowdim         0.3109958        0.245270                  0.02
```

References

https://github.com/TZstatsADS/ADS_Teaching/blob/master/Tutorials/wk10-overview-casual-inference-methods.pdf

Tolk, A., S., Diallo, .., Ryzhov, I., Yilmaz, L., Buckley, S., & Miller, J. (2014). BLENDING PROPENSITY SCORE MATCHING AND SYNTHETIC MINORITY OVER-SAMPLING TECHNIQUE FOR IMBALANCED CLASSIFICATION.