Simple vs. Complex Bootstrap for Estimating Variance of Average Treatment Effect Using Propensity Score Matching

Objectives

The purpose of this project is to compare two methods of bootstrapping when using propensity-score matching to estimate the average treatment effect on an observational study. Bootstrapping is a method of calculating the variability of a treatment effect by sampling from observed data with replacement. Propensity-score matching is a method of reducing confounding on the treatment effect by ensuring that matched treatment-control pairs have similar baseline covariates. Propensity-score matching scores each observation based on the probability that they are in the treatment group given their initial covariates. Each observation in the treatment group is matched to the observation in the control group with the closest score.

When estimating the average treatment effect, the golden standard is a randomized control trial (RCT) due to its ability to almost guarantee a balance across baseline covariates between exposed and unexposed groups by randomizing treatment. This removes the potential for confounding and allows one to make a statement on the causal effect of the treatment on the outcome. In situations where one only has access to observational data, there is a potential for confounding from the covariates which reduces one's ability to comment on the causal relationships. Propensity scores are used to try to replicate an RCT in order to reduce confounding and understand the causal effect the treatment has on the outcome.

We are interested in seeing which bootstrap performs the best under different degrees of confounding of the covariates. We will generate highly, moderately and minimally confounded data and compare the results from the bootstraps with the true value abstracted from the generated data.

Statistical methods to be studied

Propensity-score matching is a method for estimating treatment effect and bootstrapping is used to estimate its variance. For propensity score matching, nearest neighbor methods were used to match treatment and control observations based on their baseline covariates. The nearest neighbor method matches each treated observation with the untreated observation with the closest score.

Bootstrapping can be used before or after propensity-score matching, known as complex or simple bootstrapping, respectively. A bootstrap is a resampling of observed data with replacement and each bootstrap sample contains the same number of observations as the original sample. By taking many bootstraps of our original data, it is possible to simulate multiple samples from a population. This allows one to estimate the variability in the average treatment effect in different samples of the same population.

We will study whether performing propensity score matching on each bootstrap sample predicts a better estimate of this variance compared to bootstrapping on already matched samples. The former accounts for the variability in propensity score matching on top of just the sampling variability (Fig. 1), hence will be higher than that of the later.

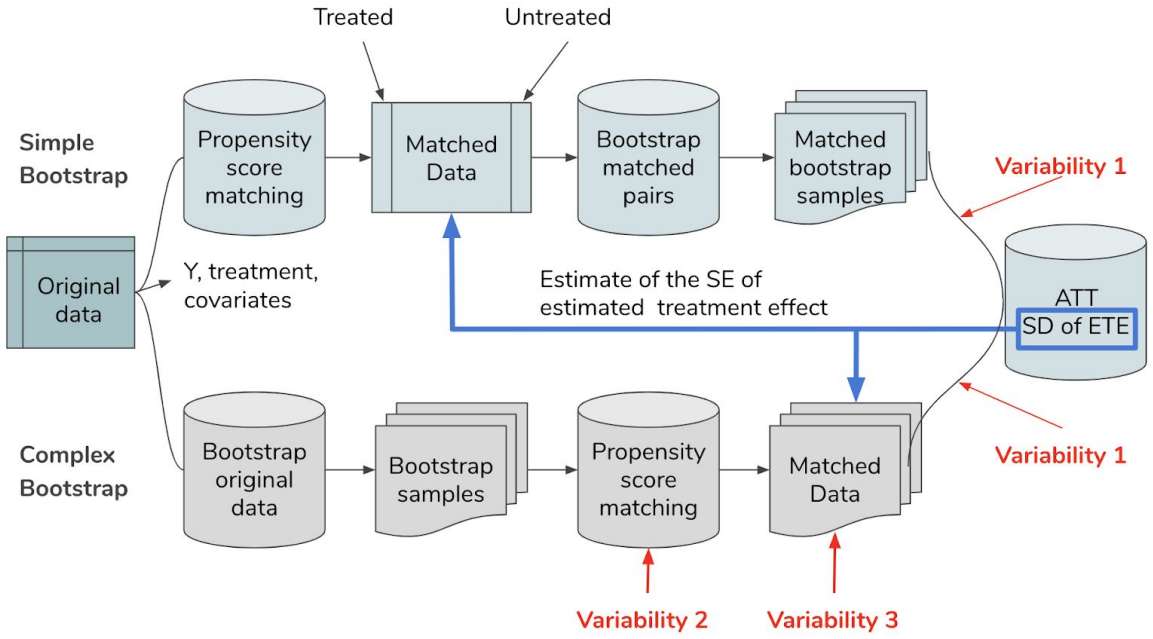

Figure 1. Simulation Procedure

Scenarios to be investigated

The goal of this simulation study is to compare the performance of the simple bootstrap and the complex bootstrap in estimating the sample variabilities of the estimated treatment effects using propensity-score matching.

In total, 15 scenarios were investigated in the simulation study (table 1). To compare the performance of the simple bootstrap and the complex bootstrap in estimating the sample variabilities under different confounding relationships, we designed 3 different levels of confounding relationships, which were "weak", "relatively strong" and "strong". The proportion of the subjects exposed (or receiving treatment) was set to 0.1, 0.2, 0.3, 0.4, 0.5 under each confounding relationship. We conducted both simple and complex bootstrap methods under each scenario to attain the average treatment effect on the treated (ATT) and its variance, which can be used as the estimate of the true treatment effect and the variance of the estimated effect. The true treatment effect under each scenario was set to 1.

Table 1. Scenarios to be investigated

Scenario	π	confounding relationship	Scenario	π	confounding relationship	Scenario	π	confounding relationship
1	0.1		**6**	0.1		**11**	0.1	
2	0.2		**7**	0.2		**12**	0.2	
3	0.3	Weak	**8**	0.3	Relatively Strong	**13**	0.3	Strong
4	0.4		**9**	0.4		**14**	0.4	
5	0.5		**10**	0.5		**15**	0.5	

*π: the proportion of the subjects being exposed or receiving treatment

Methods for generating data

In this study we only considered the scenario with a continuous outcome, although a binary outcome would also be possible. How to generate a binary outcome is included in the R code for generating all the data in https://github.com/jaredgarfinkel/P8160_project1_group1.

Firstly, we created 10 baseline covariates (X_1 ~ X_{10}), where X_i ~ N(0,1), i = 1, 2, … 10. All of these covariates were set to have different effects on the exposure or outcome depending on the degree of confounding we were exploring. The data generated for the three different levels of confounded data is described below.

1. Strong confounding covariate

For a strong confounding relationship, we assumed that X_1 ~ X_7 affected the exposure status and X_4 ~ X_{10} affected the outcome. For each subject, the probability of being exposed (P_i) can be determined by using a logistic model:

$$logit(P_i) = \beta_{0treat} + \beta_w X_1 + \beta_M X_2 + \beta_S X_3 + \beta_W X_4 + \beta_M X_5 + \beta_S X_6 + \beta_{VS} X_7$$

The intercept of the model was used to determine the proportion of subjects being exposed in the data. The coefficients β_w, β_M, β_S, and β_{VS} were set to log(1.25), log(1.5), log(1.75) and log(2). These denote weak, mediate, strong and very strong effects on the exposure status. For each subject, the event of being exposed or unexposed follows a Bernoulli distribution with the parameter P_i. Accordingly, we can generate the exposure status for each subject from a Bernoulli distribution with the subject parameter P_i, where $P_i = \frac{exp[\ logit(P_i)]}{1+exp[\ logit(P_i)]}$. The continuous outcome was generated as: $Y = treat + \beta_W X_4 + \beta_M X_5 + \beta_S X_6 + \beta_{VS} X_7 + \beta_W X_8 + \beta_M X_9 + \beta_S X_{10} + \varepsilon_i$, using the same coefficients as above. ε_i represents noise, set to follow $\varepsilon \sim N(0,3)$.

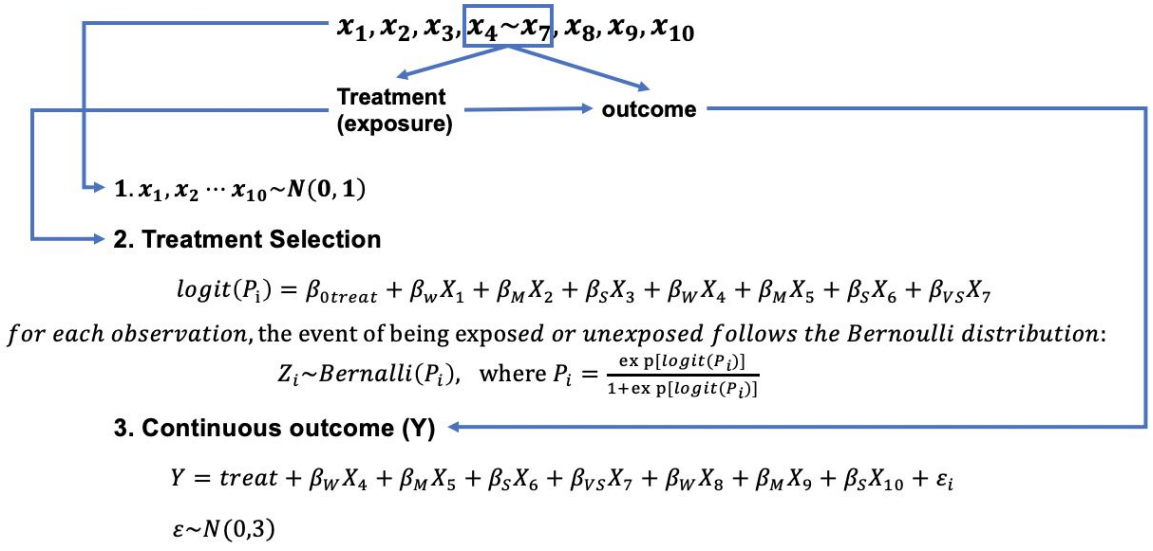

Figure 2. Basic steps for generating the strong confounding data

2. Relatively strong confounding covariate

For a relatively strong confounding relationship, we had covariate X_7 affect the exposure status and X_4 ~ X_{10} affect the outcome. For each subject, the probability of being exposed (P_i) can be determined by using a logistic model:

$$logit(P_i) = \beta_{0treat} + \beta_{VS}X_7$$

The coefficient β_{VS} was set to log(2) and intended to denote a very strong effect of X_7 on the exposure status. The remaining procedure for generating the exposure status and continuous outcome for each subject follows the same steps in generating strongly confounded data described in (1).

3. Weak confounding data

For a weak confounding relationship, we had covariate X_4 affect the exposure status and X_4 ~ X_{10} affect the outcome. For each subject, the probability of being exposed (P_i) can be determined by using a logistic model:

$$logit(P_i) = \beta_{0treat} + \beta_w X_4$$

The coefficient β_w was set to log(1.025) and intended to denote a weak effect of xX_4 on the exposure status. Since log(1.025) was close to zero, the effect of X_4 on the exposure status and outcome should be very weak. The remaining procedure for generating the exposure status and continuous outcome for each subject follows the same steps in generating strongly confounded data described in (1).

Performance measures

We are interested in comparing the simple bootstrap versus the complex bootstrap's ability to estimate the average treatment effect and the variability around this effect. The average treatment effect on the treated is the E[y_{i1} − y_{i0} | D = 1], meaning the average difference in the outcome variable for the exposed group and the unexposed group, given they are diseased. This can be calculated by:

$$ATT = 1/n \sum (y_{i1} - y_{i0})$$

where i = 1, 2, …., n represents the i^{th} matched pair, y_{i1} represents the outcome of the exposed observation from the i^{th} matched pair, and y_{i0} represents the outcome of the unexposed observation from the i^{th} matched pair.

The ATT slightly differs from the average treatment effect, which is E[y_{i1} − y_{i0}], which relies more heavily on equal covariates across the exposed and unexposed. In order to reach this balance the data ideally comes from a randomized controlled trial. Since here we are imitating an observational study, we solely consider estimating the ATT.

Bootstrapping is a common nonparametric method used for estimating the variance of an estimated value from a statistical model. We suspect both bootstrap methods will perform well in estimating the ATT. However, we believe there will be more of a discrepancy when estimating its variance, Var(ATT), or the variation in estimated ATT due to sampling

variability. This is the main performance measure of interest and we will focus on which bootstrapping method is closer to the true variance from the generated data.

Simulation results

Table 1, 2, and 3 below show the results of the estimated variance under all the different scenarios we were interested in exploring. The relative difference can be found by calculating absolute difference divided by the true variance. The simple bootstrap has a closer estimate of the variance in every scenario (Fig. 4, Fig. 5) and has very close estimates for the most part. The complex bootstrap, on the other hand, does not perform as well and performs worse as confounding becomes stronger. This may be a result of the confounding relationship being too high for propensity score matching method to effectively reduce as much confounding, such that there is a lot of variability in the propensity score matching and it is overestimating the variance. This indicates the simple bootstrap produces more reliable variances and should be used.

From Figure 3, for different treatment selection probabilities ranging from 0.1 to 0.5 and different levels of confounding, ATT estimated using the simple bootstrap versus the complex bootstrap vary in terms of proximity to the true value. The methods perform very similar to one another and there does not appear to be a trend in performance for different parameters. Figure 4 supports the lack in trend, therefore we would recommend using the simple bootstrap for predicting variance purposes although it does not necessarily perform superior for the estimate.

Table 2. Sample and true variance of Bootstraps with weak confounding

Probability of Treatment Selection	True Variance	Simple Bootstrap			Complex Bootstrap		
		Estimated Variance	Absolute Difference	Relative Difference	Estimated Variance	Absolute Difference	Relative Difference
0.1	0.2391	0.2556	0.0165	0.0690	0.3175	0.0784	0.3279
0.2	0.1185	0.1252	0.0067	0.0565	0.1292	0.0106	0.0895
0.3	0.0795	0.0909	0.0113	0.1421	0.0936	0.0140	0.1761
0.4	0.0703	0.0715	0.0012	0.0171	0.0841	0.0138	0.1963
0.5	0.0613	0.0617	0.0005	0.0082	0.0840	0.0227	0.3703

Table 3. Sample and true variance of Bootstraps with relatively strong confounding

Probability of Treatment Selection	True Variance	Simple Bootstrap			Complex Bootstrap		
		Estimated Variance	Absolute Difference	Relative Difference	Estimated Variance	Absolute Difference	Relative Difference
0.1	0.2049	0.2238	0.0188	0.0918	0.3175	0.0784	0.3826
0.2	0.1110	0.1146	0.0037	0.0333	0.1292	0.0106	0.0955
0.3	0.0836	0.0848	0.0012	0.0144	0.0936	0.0140	0.1675
0.4	0.0674	0.0675	0.0001	0.0015	0.0841	0.0138	0.2047
0.5	0.2049	0.0666	0.0015	0.0073	0.0840	0.0227	0.1108

Table 4. Sample and true variance of Bootstraps with strong confounding

Probability of Treatment Selection	True Variance	Simple Bootstrap			Complex Bootstrap		
		Estimated Variance	Absolute Difference	Relative Difference	Estimated Variance	Absolute Difference	Relative Difference
0.1	0.1724	0.1911	0.0187	0.1085	0.2619	0.0896	0.5197
0.2	0.1149	0.1048	0.0101	0.0879	0.1470	0.0321	0.2974
0.3	0.0968	0.1009	0.0041	0.0424	0.1271	0.0303	0.3130
0.4	0.0800	0.0804	0.0005	0.0063	0.1175	0.0376	0.4700
0.5	0.0726	0.0888	0.0162	0.2231	0.1109	0.0383	0.5275

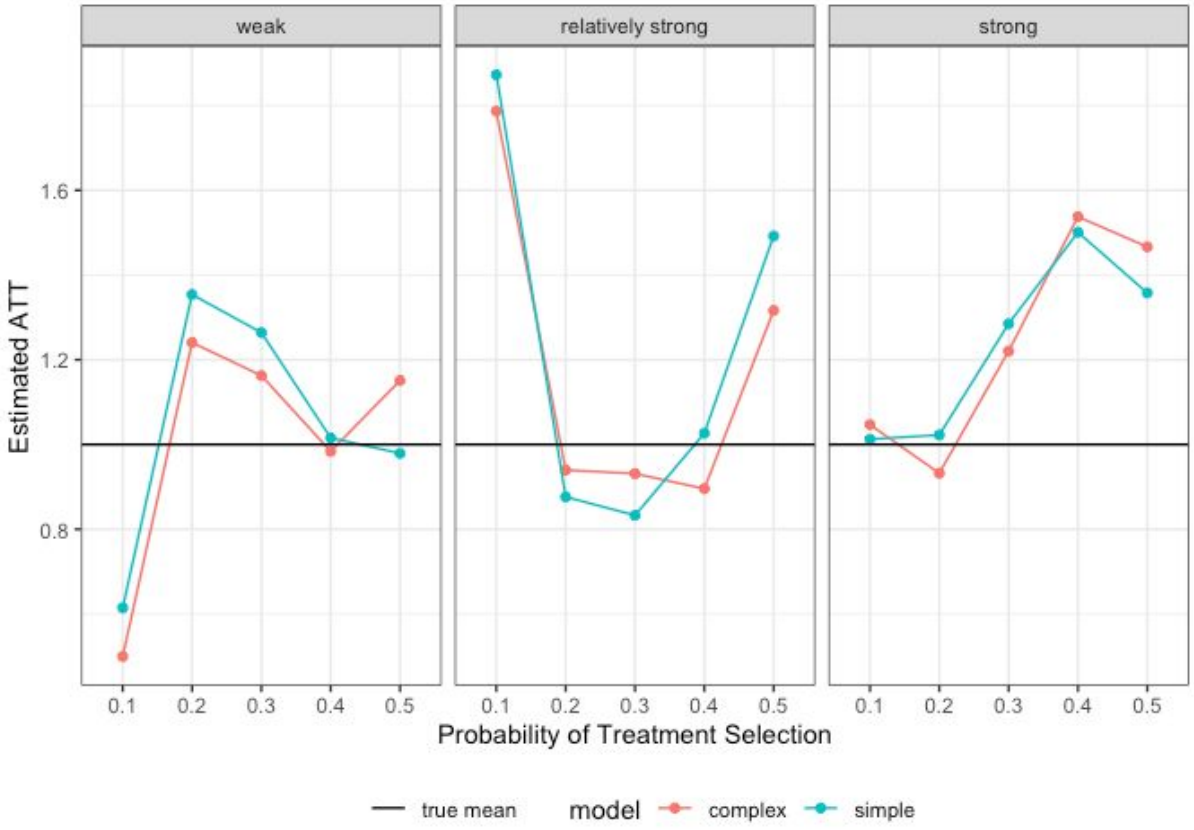

Figure 3. Estimated and True ATT

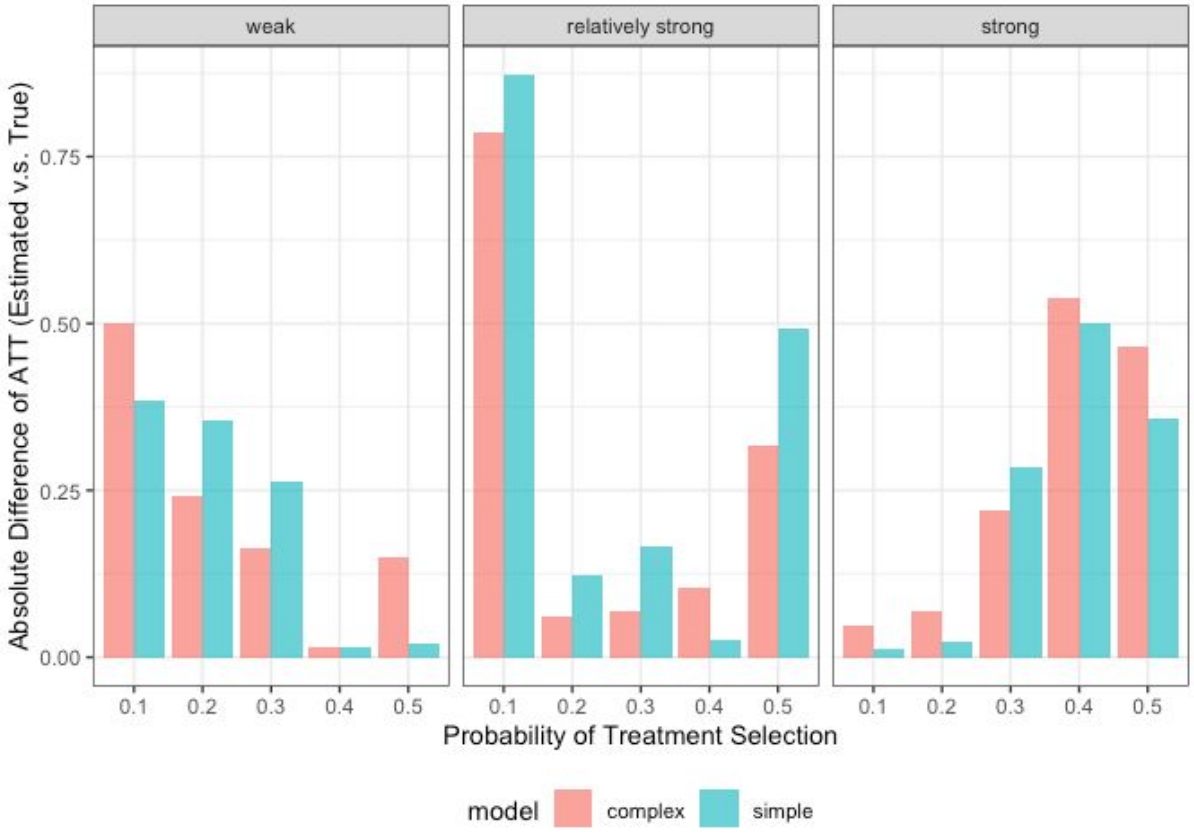

Figure 4. Absolute Difference of ATT (Estimated v.s. Ture)

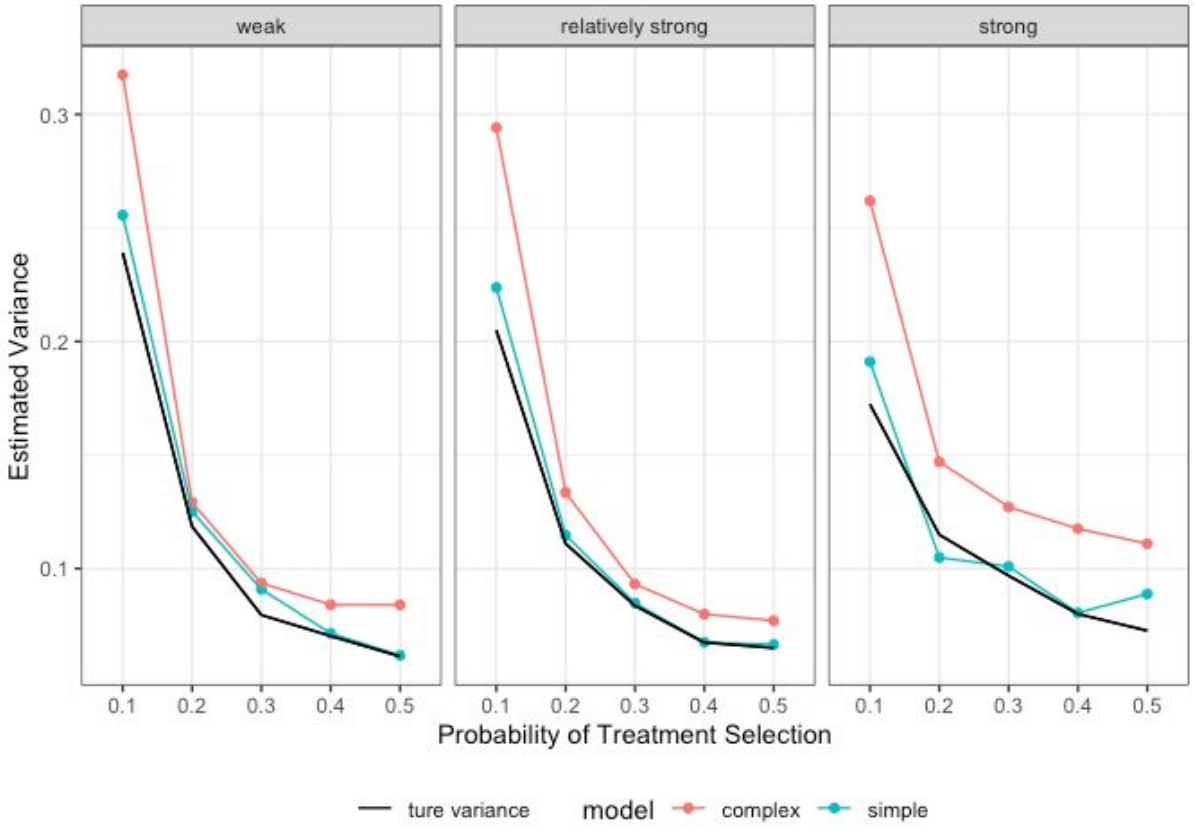

Figure 5. Estimated and True Variance

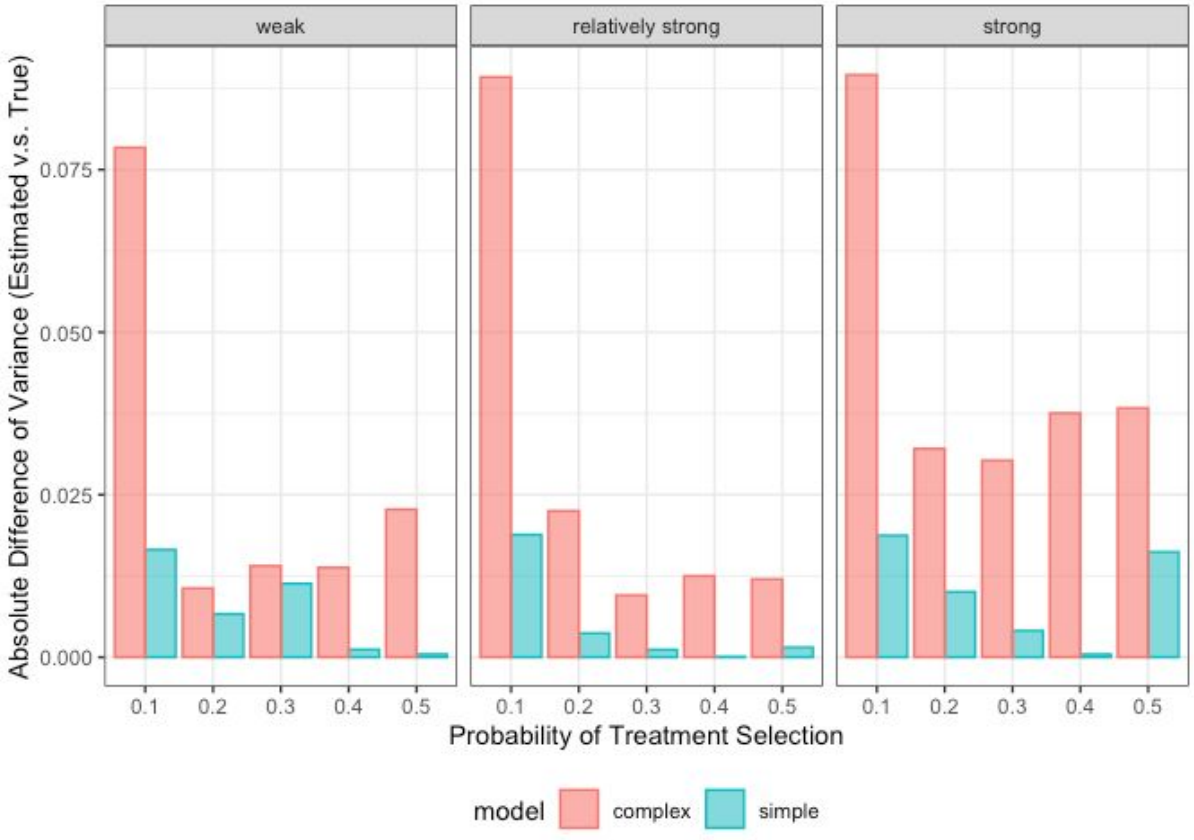

Figure 6. Absolute Difference of Variance (Estimated v.s. Ture)

Reference

[1] Austin, Peter C., and Dylan S. Small. "The use of bootstrapping when using propensity-score matching without replacement: a simulation study." *Statistics in medicine* 33.24 (2014): 4306-4319.

Appendix

Data Generation

```
## set values for all variables
N = 1000 # sample size = 1000

for(i in 1:10){
  assign(paste0("x.", i), rnorm(N))
} # create 10 variables


# define linear terms
beta.0.treat = log(0.5)  # log(pi/1-pi)
beta.effect = -log(3) # 1/3 => the proportion of log odds(of disease
#d vs not diseased) reduced by receiving the treatment
beta.low = log(1.25)
beta.med = log(1.5)
beta.high = log(1.75)
beta.v.high = log(2)


# treament selection
logit.treat <- beta.0.treat + beta.low*x.1 + beta.med*x.2 +
  beta.high*x.3 + beta.low*x.4 + beta.med*x.5 + beta.high*x.6 + beta.v.high*x.7

p.treat <- exp(logit.treat)/(1 + exp(logit.treat)) # probability of treatment selection

treat <- rbinom(N,1,p.treat) # assign treatment based on the calculated probability


# generate the continuous outcome
y <- 1*treat + beta.low*x.4 + beta.med*x.5 + beta.high*x.6 + beta.v.high* x.7
+ beta.low*x.8 + beta.med*x.9 + beta.high*x.10 + rnorm(N,0,3)


## Create the dataframe
df = tibble(x.1,x.2,x.3,x.4,x.5,x.6,x.7,x.8,x.9,x.10,y,treat) %>%
  select(y, treat, everything()) %>%
  mutate(treat = as.logical(treat == 1))
```

Functions

```
# bootstrapping
boot_sample = function(df) {
  sample_frac(df, size = 1,replace = TRUE)
}
```

```r
# matching; the input is the dataframe with observations
#unmatched and the output is the dataframe with only matched observations.
match = function(df){
  mode_match = MatchIt::matchit(treat ~ x.1+x.2+x.3+x.4+x.5+x.6+x.7,
                                method = "nearest",
                                data = df,
                                ratio = 1,
                                caliper = 0.01
                                )
  return(match.data(mode_match))
}


# Estimation of average treatment effect of the treated
#(ATT)=sum(y exposed- y unexposed)/# of matched pairs
ATT = function(df){
  sum = df %>%
    group_by(treat) %>%
    summarise(sum = sum(y))
  sum_diff = sum[2,2]- sum[1,2]
  result = sum_diff/(nrow(df)/2)
  return(result$sum)
}


# define a function to do the simple bootstrap(by bootstrapping the index of each observation)
resample_index = function(df){
  df1 = df %>%
    filter(treat == 1) %>%
    arrange(distance)
  df1 = df1 %>%
    mutate(index = 1:nrow(df1))
  df2 = df %>%
    filter(treat == 0) %>%
    arrange(distance)
  df2 = df2 %>%
    mutate(index = 1:nrow(df2))
  selected_index = sample(c(1:(nrow(df)/2)), nrow(df)/2, replace = T) #resample the index
  df3 = rbind(df1[selected_index,],df2[selected_index,])
  return(df3)
}


#define a function to regenerate data for 1000 times
regenerate_df = function(seed){
    set.seed(seed)
    # generate data
    N = 1000 # sample size = 1000
    for(j in 1:10){
      assign(paste0("x.", j), rnorm(N))
      } # create 10 variables
    # define linear terms
    beta.0.treat = log(pi/(1 - pi)) # we choose 0.1 to 0.5 as pi
```

```
    beta.effect = -log(3) # 1/3 => the proportion of log odds(of diseased vs not diseased) reduced
    beta.low = log(1.25)
    beta.med = log(1.5)
    beta.high = log(1.75)
    beta.v.high = log(2)
    # treament selection
    logit.treat <- beta.0.treat + beta.low*x.1 + beta.med*x.2 + beta.high*x.3 +
      beta.low*x.4 + beta.med*x.5 + beta.high*x.6 + beta.v.high*x.7
    p.treat <- exp(logit.treat)/(1 + exp(logit.treat)) # probability of treatment selection
    treat <- rbinom(N,1,p.treat) # assign treatment based on the calculated probability
    y <- 1*treat + beta.low*x.4 + beta.med*x.5 + beta.high*x.6 + beta.v.high* x.7
    + beta.low*x.8 + beta.med*x.9 + beta.high*x.10 + rnorm(N,0,3)
    # create df
    df = tibble(x.1,x.2,x.3,x.4,x.5,x.6,x.7,x.8,x.9,x.10,y,treat) %>%
      select(y, treat, everything()) %>%
      mutate(treat = as.logical(treat == 1))
    return(df)
}
```

True Variance

```
generate = tibble(
  seed = 1:1000
) # create a dataframe to do 1000 independent experiments


# regenerate data
generate = generate %>%
  mutate(new_df = map(seed,regenerate_df))
# do the matching for each regenerated data
generate = generate %>%
  mutate(matched_df = map(new_df,match))
# calculate the treatment effect
generate = generate %>%
  mutate(ATT = map(matched_df, ATT))
# calculate the true effects & true variance
mean(as.numeric(generate$ATT))
sd(as.numeric(generate$ATT))
```

Simple Bootstrap

```
df_sim = match(df) # match the data first

bootstrap_sample_sim = tibble(
  strap_number = 1:1000,
  strap_sample = rerun(1000, resample_index(df_sim))
) # simple bootstrap
```

```r
# calculate the average treatment effect
att_sim = NULL
for (i in 1:nrow(bootstrap_sample_sim)) {
  att_sim[i]= ATT(bootstrap_sample_sim$strap_sample[[i]])
}

# add estimated treatment effect to the boostrap samples
bootstrap_sample_sim = cbind(bootstrap_sample_sim,att_sim)
```

Complex Bootstrap

```r
bootstrap_sample = tibble(
  strap_number = 1:1000,
  strap_sample = rerun(1000, boot_sample(df)),
  prs_df = map(strap_sample,match)
) # bootstrap first

att_complex = NULL
for (i in 1:nrow(bootstrap_sample)) {
  att_complex[i]= ATT(bootstrap_sample$prs_df[[i]])
}

# add estimated treatment effect to the boostrap samples
bootstrap_sample = cbind(bootstrap_sample,att_complex)
```

Compare the treatment effect and the standard error

```r
# the treatment effect and the standard error by simple bootstrap
a = c(mean(bootstrap_sample_sim$att_sim),
      sd(bootstrap_sample_sim$att_sim)) # mean, std
# the treatment effect and the standard error by complex bootstrap
b = c(mean(bootstrap_sample$att_complex),
      sd(bootstrap_sample$att_complex)) # mean, std

results = rbind(a,b)
rownames(results) = c("simple bootstrap",
                      "complex bootstrap")
colnames(results) = c("mean", "sd")

results
```