Propensity Score Weighting and Generalized Boosted Regression in R

Peter Sun

September 30, 2021

Contents

1 Load Packages

Stata DTA files can be imported directly into R using the `haven` and `sjlabelled` packages.

Propensity score weighting can be accomplished with base R. However, we need the `lmtest` and `sandwich` packages to estimate clustered covariance matrices in this example. Using these packages, we can obtain estimates and standard errors that are identical to Stata's `regress` program.

Generalized boosted regression requires the `gbm` package.

Finally, the `tidyverse` package can be optionally loaded for its convenient data manipulation and plotting functions.

```
library(tidyverse)
library(haven)
library(sjlabelled)
library(lmtest)
library(sandwich)
library(gbm)
```

2 Propensity Score Weighting

2.1 Description of Dataset

This dataset is from a study that investigates intergenerational dependence on welfare and its relation to child academic achievement.[1]

The dependent variable is `lwss97` or "letter-word identification" score, and the treatment condition is `kuse` or children who used Aid to Families With Dependent Children (AFDC). The covariates are:

- `male`: Child's Gender: Male (Reference: Female)
- `black`: Child's Race: African American (Reference: Other)
- `age97`: Child's Age in 1997
- `pcged97`: Caregiver's Education in 1997 (Years of Schooling)
- `mratio96`: Ratio of Family Income to Poverty Line in 1996

Additionally, `pcg_id` is a cluster variable that identifies children nested within families.

2.2 Load Data with Propensity Scores

```
d <- read_dta("data/chpt5_2_original.dta") %>%
  haven::zap_formats() %>%
  sjlabelled::remove_all_labels() %>%
  as_tibble()
head(d$ps)
```

```
## [1] 0.3096053 0.3064355 0.1935465 0.3243693 0.3178224 0.2095492
```

2.3 Estimate ATE and ATT Weights

Here we calculate separate weights for estimating the average treatment effect (ATE) and the average treatment effect for the treated (ATT).

For ATE, the weight estimates are calculated as follows for the treatment group:

[1] Hofferth, S., Stafford, F. P., Yeung, W. J., Duncan, G. J., Hill, M. S., Lepkowski, J., et al. (2001). *Panel study of income dynamics, 1968–1999: Supplemental files (computer file), ICPSR version.* Ann Arbor: University of Michigan Survey Research Center.

$$\omega = \frac{1}{\hat{e}(x)}$$

And for the control group:

$$\omega = \frac{1}{1 - \hat{e}(x)}$$

For ATT, the weight is 1 for a treated case. The weight for a comparison case is:

$$\omega = \frac{\hat{e}(x)}{1 - \hat{e}(x)}$$

```
d.weights <- d %>%
  mutate(ate_w = ifelse(kuse == 0, 1/(1-ps), 1/ps),
         att_w = ifelse(kuse == 0, ps/(1-ps), 1))
```

2.4 Outcome Analysis

2.4.1 Weighted Regression with ATE Weights

After creating the weights, use the `weights` argument in `lm()` to run a weighted outcome analysis and `lmtest::coeftest()` to control for clustering effects.

This analysis showed that children who used Aid to Families With Dependent Children (AFDC) had an average letter-word identification score that was 5.16 points lower than children who never used AFDC, $p < .01$.

```
m3 <- lm(lwss97 ~ kuse + male + black + age97 + pcged97 + mratio96,
         data = d.weights, weights = ate_w)
lmtest::coeftest(m3, vcov. = vcovCL(m3, cluster = d.weights$pcg_id))
```

```
##
## t test of coefficients:
##
##             Estimate Std. Error t value  Pr(>|t|)
## (Intercept) 84.19992    4.83032 17.4315 < 2.2e-16 ***
## kuse        -5.16399    1.42438 -3.6254 0.0003031 ***
## male        -1.62201    1.09186 -1.4855 0.1377180
## black       -2.49898    1.34670 -1.8556 0.0638009 .
## age97        0.73868    0.18075  4.0867 4.727e-05 ***
## pcged97      0.99264    0.35596  2.7886 0.0053938 **
## mratio96     1.13856    0.32220  3.5337 0.0004286 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2.4.2 Weighted Regression with ATT Weights

When considering only individuals assigned to the treatment condition, children who used AFDC had an average letter-word identification score that was 4.62 points lower than children who never used AFDC, $p < .01$.

```
m4 <- lm(lwss97 ~ kuse + male + black + age97 + pcged97 + mratio96,
         data = d.weights, weights = att_w)
lmtest::coeftest(m4, vcov. = vcovCL(m4, cluster = d.weights$pcg_id))
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value  Pr(>|t|)    
## (Intercept) 85.29467    5.05331 16.8790 < 2.2e-16 ***
## kuse        -4.62058    1.41182 -3.2728  0.001102 ** 
## male        -1.58995    1.14705 -1.3861  0.166020    
## black       -2.74605    1.41605 -1.9392  0.052756 .  
## age97        0.61577    0.20001  3.0787  0.002136 ** 
## pcged97      0.92698    0.36718  2.5246  0.011738 *  
## mratio96     1.26018    0.33556  3.7555  0.000183 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2.5 Check Imbalance

To assess balance before and after propensity score weighting, use logistic regression for dummy covariates and OLS regression for continuous covariates. Some examples are included below, and the full code can be found in Section 7.3.1 of the PSA-R code.

In model `c5` below, the treatment dummy variable is significant, meaning that there is no sufficient balance after the propensity score weighting.

To assess balance before propensity score weighting, remove the `weights` argument.

```r
c1 <- glm(male ~ kuse, family = quasibinomial, data = d.weights, weights = ate_w)
lmtest::coeftest(c1, vcov. = vcovCL(c1, cluster = d.weights$pcg_id))
```

```
##
## z test of coefficients:
##
##               Estimate Std. Error z value Pr(>|z|)  
## (Intercept)  0.142911   0.075538  1.8919   0.0585 .
## kuse        -0.060271   0.150143 -0.4014   0.6881  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
c2 <- glm(male ~ kuse, family = quasibinomial, data = d.weights, weights = att_w)
lmtest::coeftest(c2, vcov. = vcovCL(c2, cluster = d.weights$pcg_id))
```

```
##
## z test of coefficients:
##
##               Estimate Std. Error z value Pr(>|z|)  
## (Intercept)  0.152522   0.077410  1.9703   0.0488 *
## kuse        -0.079497   0.147480 -0.5390   0.5899  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
c5 <- lm(age97 ~ kuse, weights = ate_w, data = d.weights)
lmtest::coeftest(c5, vcov. = vcovCL(c5, cluster = d.weights$pcg_id))
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value  Pr(>|t|)    
## (Intercept)  6.51389    0.10900 59.7615 < 2.2e-16 ***
## kuse         0.61064    0.21883  2.7905  0.005362 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
c6 <- lm(age97 ~ kuse, weights = att_w, data = d.weights)
lmtest::coeftest(c6, vcov. = vcovCL(c6, cluster = d.weights$pcg_id))
```

```
##
## t test of coefficients:
##
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.55501    0.11499 57.0057  < 2e-16 ***
## kuse         0.56178    0.21939  2.5606  0.01059 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3 Estimate Propensity Scores Using Generalized Boosted Regression

3.1 Load Data and Sort

Generalized boosted regression (GBR) is an iterative method for creating propensity scores. Therefore, to create reproducible results, we need to use the `set.seed()` function in R.

After importing the data, missing data is deleted listwise, and the data is sorted in a random order. Note the use of the `set.seed()` function to create the same set of random numbers from the uniform distribution.

According to the `gbm` package vignette, if the data is sorted in a systematic way, then the data should be shuffled before running `gbm`.

```
# Correction: Randomly shuffle the data before fitting GBM model
set.seed(1000)
d2 <- read_dta("data/g3aca1.dta") %>%
  haven::zap_formats() %>%
  sjlabelled::remove_all_labels() %>%
  as_tibble() %>%
  select(intbl, ageyc, fmale, blck, whit, hisp, pcedu, ipovl, pcemft, fthr,
         dicsagg2, dicsint2, dccereg2, dccscom2, dccpros2, draggr2) %>%
  drop_na() %>%
  add_column(runif = runif(nrow(.))) %>%
  arrange(runif)
```

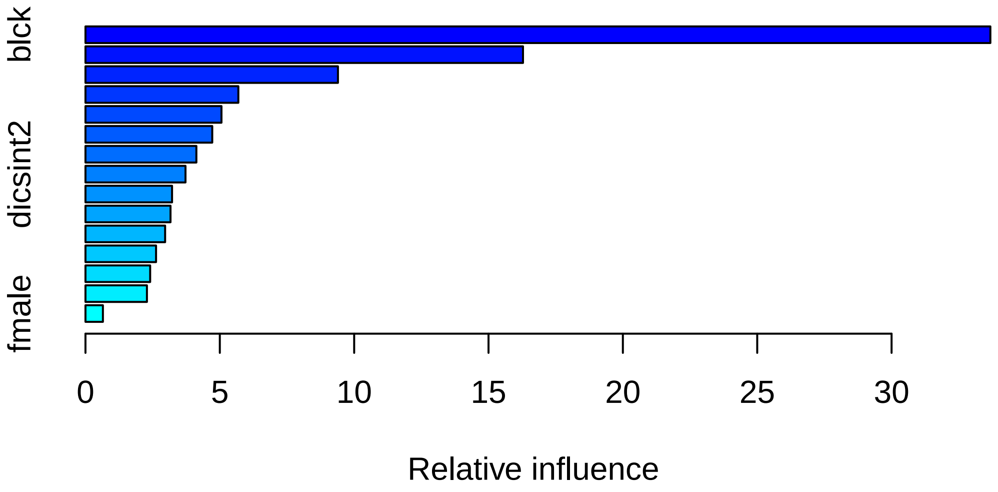
3.2 Fit Generalized Boosted Regression Model

The `gbm::gbm()` function has many arguments that can be fine-tuned. See `?gbm` for a detailed description of each argument.

A summary of the fitted model provides us with *relative influence*, which is the percentage of log likelihood explained by each input variable.

The GBM showed that `blck` had the strongest influence on the likelihood function (33.7%), followed by `ageyc` (16.3%) and `draggr2` (9.4%).

```
# Correction: Remove the column of random numbers from the formula
f5 <- as.formula(paste("intbl ~ ", paste(names(select(d2, -intbl, -runif)),
                                         collapse = " + ")))
set.seed(1000)
m5 <- gbm::gbm(formula = f5,
          data = d2,
          distribution = "bernoulli",
          n.trees = 1000, # number of trees to fit
          train.fraction = 0.8, # a random 80% subsample for estimation
          interaction.depth = 4, # allow all four-way interactions
          shrinkage = 0.0005) # small shrinkage to ensure smooth fit
summary(m5)
```



```
##                var    rel.inf
## blck          blck 33.6768868
## ageyc        ageyc 16.2834103
## draggr2    draggr2  9.3962151
## whit          whit  5.6895119
## ipovl        ipovl  5.0614691
## pcemft      pcemft  4.7179839
## pcedu        pcedu  4.1280063
## dicsint2  dicsint2  3.7223185
## dicsagg2  dicsagg2  3.2228567
## dccscom2  dccscom2  3.1648839
## dccereg2  dccereg2  2.9647178
## dccpros2  dccpros2  2.6269129
## hisp          hisp  2.4079157
## fthr          fthr  2.2874508
## fmale        fmale  0.6494603
```

3.3 Estimate Propensity Scores

After fitting the model, propensity scores can be generated by using the `predict.gbm()` function.

```
psb <- gbm::predict.gbm(m5, data = d2, type = "response") # Create ps
```

```
## Using 1000 trees...
```

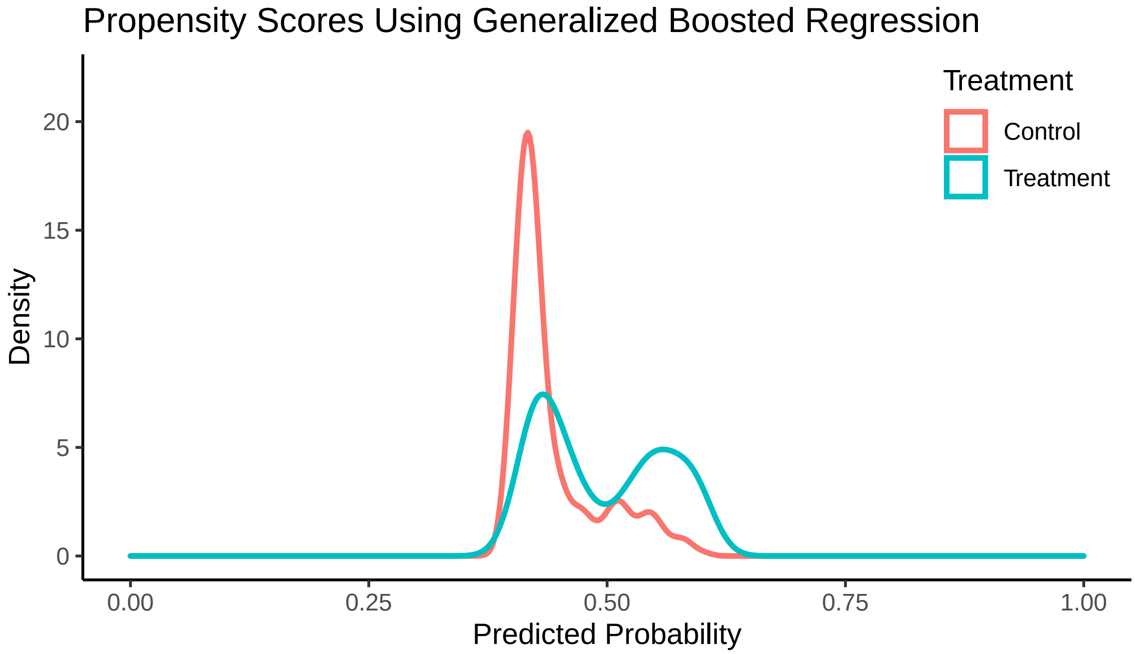
```
head(psb)
```

```
## [1] 0.4441531 0.4405950 0.5288983 0.5958331 0.3985049 0.5562076
```

3.4 Plot Propensity Score Distributions

As seen in the figure below, the propensity scores estimated by GBM has sufficient overlap between the control and treatment groups (i.e., “common support”), and the distributions are similar.

```
d2 %>%
  mutate(psb = psb,
         intbl = factor(intbl, labels = c("Control", "Treatment"))) %>%
  ggplot(aes(x = psb, color = intbl)) + theme_classic() +
  geom_density(size = 1) + xlim(0, 1) + ylim(0, 22) +
  labs(x = "Predicted Probability", y = "Density",
       title = "Propensity Scores Using Generalized Boosted Regression",
       color = "Treatment") +
  theme(legend.position = c(0.9, 0.85))
```



3.5 Summary Statistics of Propensity Scores

```
summary(psb)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.3913  0.4189  0.4383  0.4674  0.5202  0.6082
```

4 Bonus: Using the WeightIt Package

4.1 Estimate ATE and ATT Weights

```
# Load Packages
library(WeightIt)
library(cobalt)

# Estimate ATE and ATT weights and Check with Previous Results
ate_w2 <- WeightIt::get_w_from_ps(ps = d$ps,
                                  treat = d$kuse,
                                  estimand = "ATE")
table(ate_w2 == d.weights$ate_w)
```

```
##
## TRUE
## 1003
```

```
att_w2 <- WeightIt::get_w_from_ps(ps = d$ps,
                                  treat = d$kuse,
                                  estimand = "ATT")
table(att_w2 == (d.weights$ate_w * d.weights$ps))
```

```
##
## TRUE
## 1003
```

4.2 Estimate Propensity Scores using Generalized Boosted Regression

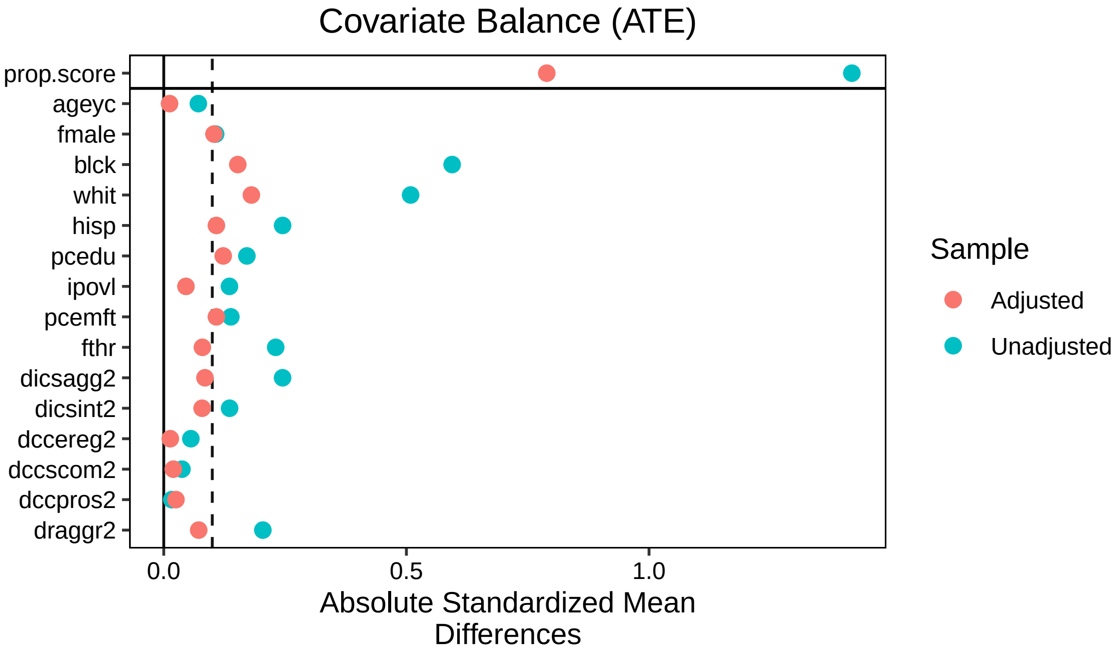
```
# Estimate PS with GBM
set.seed(1000)
m5.alt <- WeightIt::weightit(
  formula = f5,
  data = d2,
  method = "gbm",
  estimand = "ATE",
  distribution = "bernoulli",
  stop.method = "es.mean",
  n.trees = 10000, # different
  nTrain = 0.8 * nrow(d2), # different
  interaction.depth = 4,
  shrinkage = 0.0005
)
```

4.3 Check Imbalance with the Cobalt Package

The standardized mean difference (SMD) is a commonly used balance measure. It is calculated as the difference in means of a covariate across the treatment groups, divided by the standard deviation in the treated group. Stuart et al. (2013) recommend 0.1 or 0.25 as reasonable cut-offs for acceptable standardized biases.[2]

Using the more stringent 0.1 cut-off point, it can be seen below that balance has been achieved in most, but not all, of the covariates:

```
cobalt::love.plot(m5.alt, thresholds = c(m = .1), binary = "std", abs = T) +
  labs(title = "Covariate Balance (ATE)")
```



[2] Stuart, E. A., Lee, B. K., & Leacy, F. P. (2013). Prognostic score–based balance measures for propensity score methods in comparative effectiveness research. *Journal of Clinical Epidemiology*, 66(8 0), S84-S90.e1. https://doi.org/10.1016/j.jclinepi.2013.01.013