Propensity Score Weighting

using machine learning

Young Geun Kim
ygeunkim.github.io

2019711358, Department of Statistics

03 Dec, 2020

Introduction

Reviewed Paper

Estimation

Reviewed and apply Lee et al. (2010): estimate propensity score using

- Logistic regression: `glm()`
- Random forests: `randomForest::randomForest()`
- SVM (Pirracchio et al., 2014): `e1071::svm()`

Evaluation

- Average standardized absolute mean distance
- Emprical distribution of IPTW
- IPW and SIPW

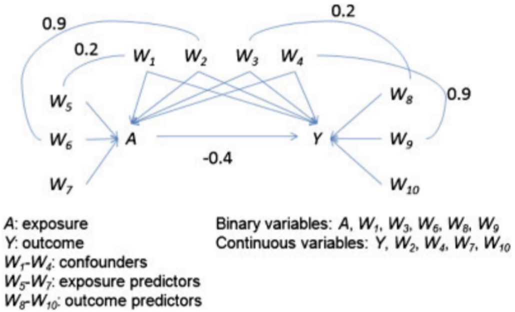
Custom Package

```r
# remotes::install_github("ygeunkim/propensityml")
library(propensityml)
```

Simulation study

Simulation setting by Setoguchi et al. (2008):

- 10 covariates: confounders, exposure predictors, outcome predictors
- Treatment, (true) treatment probability
- Continuous outcome



Correlation Matrix

of covariates:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0.2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0.9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0.9 & 0 \\ 0.2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.9 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0.2 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0.9 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Scenarios

1. Additivity and linearity:

$$P(Z = 1 \mid X_i) = \frac{1}{1 + \exp(-(\beta_0 + \beta_1 X_1 + \cdots + \beta_7 X_7))}$$

2. Moderate non-linearity: *3 quadratic term*

$$P(Z = 1 \mid X_i) = \frac{1}{1 + \exp(-(\beta_0 + \beta_1 X_1 + \cdots + \beta_7 X_7 + \beta_2 X_2^2))}$$

3. Moderate non-linearity: *10 two-way interaction terms*
4. Moderate non-additivity and non-linearity: *10 two-way interaction terms and 3 quadratic terms*

Here,

$$(\beta_0, \beta_1, \ldots, \beta_7)^T = (0, 0.8, -0.25, 0.6, -0.4, -0.8, -0.5, 0.7)^T$$

Outcome

As in Figure @ref(fig:setoguchifig),

$$Y = \alpha_0 + \alpha_1 X_1 + \cdots + \alpha_4 X_4 + \alpha_5 X_8 + \cdots + \alpha_7 X_{10} + \gamma Z$$

where

- $(\alpha_0, \alpha_1, \ldots, \alpha_7)^T = (-3.85, 0.3, -0.36, -73, -0.2, 0.71, -0.19, 0.26)^T$
- $\gamma = -0.4$: True effect

Function to reproduce Setoguchi et al. (2008)

```
sim_outcome(n = 1000, covmat = build_covariate()) %>%
  glimpse(width = 50)
#> Rows: 1,000
#> Columns: 13
#> $ w1           <fct> 0, 1, 1, 1, 0, 1, 1, 1,...
#> $ w2           <dbl> -0.2801, 0.3065, 0.6329...
#> $ w3           <fct> 0, 0, 0, 1, 1, 1, 1, 1,...
#> $ w4           <dbl> 1.6575, -1.4404, -1.939...
#> $ w5           <fct> 1, 1, 1, 0, 0, 1, 0, 0,...
#> $ w6           <fct> 0, 1, 1, 0, 0, 1, 1, 0,...
#> $ w7           <dbl> 0.4874, -0.0162, -0.155...
#> $ w8           <fct> 1, 1, 0, 0, 1, 0, 1, 1,...
#> $ w9           <fct> 1, 0, 0, 1, 1, 0, 1, 0,...
#> $ w10          <dbl> -0.3054, 0.5939, 0.4179...
#> $ exposure     <fct> 1, 1, 1, 1, 1, 0, 1, 1,...
#> $ y            <dbl> -120.253, 0.942, -51.95...
#> $ exposure_prob <dbl> 0.5000, 0.9072, 0.3465,...
```

Sample Sizes

Monte Carlo simulation

- For simulation, 1000 replicates

Sample size

- 1000
- 1500

Propensity Score Estimation

Covariate Balance

For example,

```
compute_balance(
  small_list[mcname == 1 & scenario == "A"],
  treatment = "exposure", trt_indicator = 1, outcome = "y",
  exclude = c("exposure_prob", "mcname", "scenario")
)
#>     variable   balance
#>  1:       w1  0.024098
#>  2:       w2 -0.059650
#>  3:       w3  0.022855
#>  4:       w4 -0.077968
#>  5:       w5  0.037466
#>  6:       w6 -0.092068
#>  7:       w7  0.069232
#>  8:       w8  0.039684
#>  9:       w9 -0.000728
#> 10:      w10  0.039634
```

Lee et al. (2010): under 0.2 is acceptable.

Logistic Regression

Evaluation

Average standardized absolute mean distance (ASAM)

- Covariate balancing: standardized mean differece, which is standardized by pooled sd
- Average the abs(covariate balancing) across all the covariates
- Lower: treatment and control groups are more similar w.r.t. the given covariates.

```
doMC::registerDoMC(cores = 8)
logit_asam <-
  small_list %>%
  compute_asam(
    treatment = "exposure", outcome = "y", exclude = "exposure_prob",
    formula = exposure ~ . - y - exposure_prob, method = "logit",
    mc_col = "mcname", sc_col = "scenario", parallel = TRUE
  )
```

ASAM for each model

Table 1: ASAM performance for small

Scenarios	Logistic regression	Random forests	SVM
A	0.011	0.011	0.010
B	0.032	0.029	0.042
F	0.034	0.033	0.041
G	0.077	0.074	0.081

Effect estimator

IPTW

- Estimating ATE: using inverse probability of treatment weighing (IPTW)

$$IPTW_i = \frac{Z_i}{\hat{e}_i} + \frac{1 - Z_i}{1 - \hat{e}_i}$$

Evaluation

- Empirical distribution
  - Histogram
  - Bias: difference between true effect ($\gamma = -0.4$)
  - Standard deviation
  - Confidence interval

How it works?

```
doMC::registerDoMC(cores = 8)
iptw_logit <-
  small_list %>%
  add_iptw(
    treatment = "exposure",
    formula = exposure ~ . - y - exposure_prob, method = "l
    mc_col = "mcname", sc_col = "scenario", parallel = TRUE
  )
```

Empirical Distribution

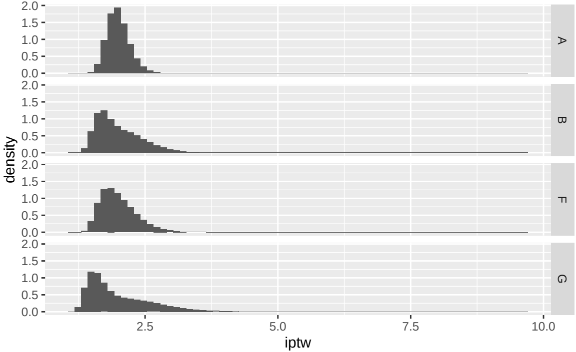


Figure 2: Empirical Distribution of IPTW

Table

```
iptw_logit[,
           .(
             estimate = mean(iptw),
             bias = mean(abs(iptw)) / .4,
             se = sd(iptw),
             lb = quantile(iptw, .25),
             ub = quantile(iptw, .75)
           ),
           by = scenario]
#>     scenario estimate bias    se   lb   ub
#> 1:         A        2    5 0.215 1.85 2.13
#> 2:         B        2    5 0.420 1.68 2.25
#> 3:         F        2    5 0.340 1.75 2.19
#> 4:         G        2    5 0.607 1.54 2.34
```

Other Models

Weighting

Methods

- Inverse probability weighting (IPW): weighted regression of outcome on treatment $\hat{\Delta}_{IPW}$
- Stabilized inverse probability weighting (SIPW): $\hat{\Delta}_{SIPW}$

```
doMC::registerDoMC(cores = 8)
ipw_logit <-
  small_list %>%
  compute_ipw(
    treatment = "exposure", outcome = "y",
    formula = exposure ~ . - y - exposure_prob, method = "logit",
    mc_col = "mcname", sc_col = "scenario", parallel = TRUE
  )
```

Our data

- weight of treatment: 1
- weight of control: $\frac{p_i}{1-p_i}$
- If $\hat{e}$ is proper
  - then two weights are similar
  - ATE estimate: difference of weighted means

Empirical Distribution of IPW

```
#> Warning: Removed 1 rows containing non-finite values (st
```

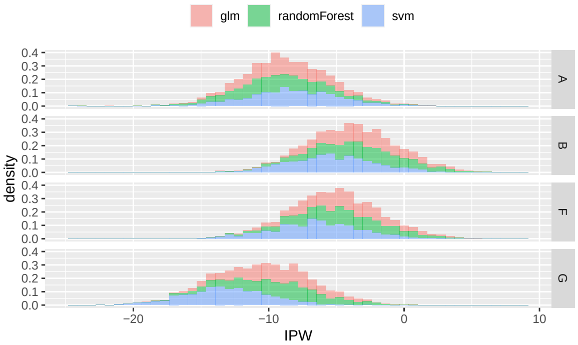


Figure 3: Empirical Distribution of IPW

Empirical Distribution of SIPW

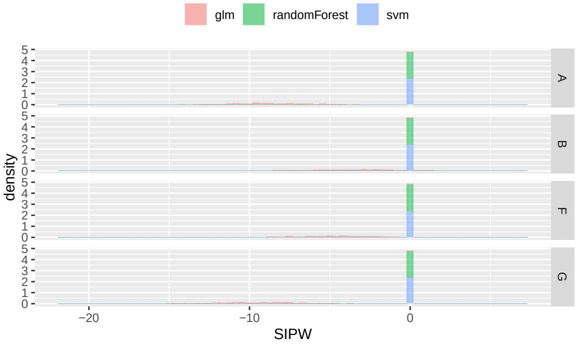


Figure 4: Empirical Distribution of SIPW

Related Contents

About this project

Project repository

https://github.com/ygeunkim/psweighting-ml

Project package

https://github.com/ygeunkim/propensityml

References I

Lee, B. K., Lessler, J., and Stuart, E. A. (2010). Improving propensity score weighting using machine learning. *Statistics in Medicine*, 29(3):337–346.

Pirracchio, R., Petersen, M. L., and van der Laan, M. (2014). Improving propensity score estimators' robustness to model misspecification using super learner. *American Journal of Epidemiology*, 181(2):108–119.

Setoguchi, S., Schneeweiss, S., Brookhart, M. A., Glynn, R. J., and Cook, E. F. (2008). Evaluating uses of data mining techniques in propensity score estimation: a simulation study. *Pharmacoepidemiology and Drug Safety*, 17(6):546–555.