Propensity Score Weighting

using machine learning

Young Geun Kim
ygeunkim.github.io

2019711358, Department of Statistics

08 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

My Own 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 (exposure), true propensity score
- Continuous outcome

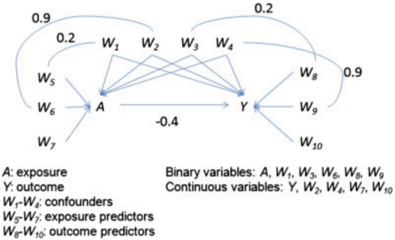

Figure 1: Simulation Data - Each W and A can be as X and Z in the course, respectively

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

True propensity score

Define $e(\mathbf{X}_i)$ for each scenario (`A`, `B`, `F`, `G`):

A Additivity and linearity:

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

B Moderate non-linearity: *3 quadratic term*

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

F Moderate non-linearity: *10 two-way interaction terms*

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

True Parameters

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

Outcome

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

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,...
```

Simulation and Evaluation

Monte Carlo simulation

- For simulation, 1000 replicates
- Sample size: 1000

```
doMC::registerDoMC(cores = 4)
mc_list <- mc_setoguchi(
  N = 1000, n_dat = 1000, scenario = scen,
  parallel = TRUE
)
```

Columns that indicate MC and Scenario: `mcname`, `scenario`

```
mc_list[, .N, .(mcname, scenario)]
#>       mcname scenario    N
#>    1:      1        A 1000
#>    2:      2        A 1000
#>    3:      3        A 1000
#>    4:      4        A 1000
#>    5:      5        A 1000
#>   ---
#> 3996:    996        G 1000
#> 3997:    997        G 1000
#> 3998:    998        G 1000
#> 3999:    999        G 1000
#> 4000:   1000        G 1000
```

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 <-
  mc_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
  )
```

Covariate Balance: ASAM

Scenarios	Model			
	Logistic	RF	SVM (Linear)	SVM (Radial)
A	0.011	0.011	0.009	0.010
B	0.033	0.030	0.041	0.042
F	0.035	0.033	0.041	0.041
G	0.076	0.075	0.080	0.080

- Under 0.2 is acceptable (Lee et al., 2010)
- All are OK.

Effect estimator

Estimation of Treatment Effect

- Inverse probability of treatment weighing (IPTW):

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

- Weight 1 vs $\frac{\hat{e}_i}{1-\hat{e}_i}$:

$$Z_i - \frac{\hat{e}_i(1 - Z_i)}{1 - \hat{e}_i}$$

Evaluation

- Empirical distribution
  - Histogram or boxplot
  - Bias: difference between true effect ($\gamma = -0.4$)
  - Standard deviation
  - 95% Confidence interval

Average Treatment Effect

Estimators

- Inverse probability weighting (IPW): $\hat{\Delta}_{IPW}$
- Stabilized inverse probability weighting (SIPW): $\hat{\Delta}_{SIPW}$

Performance

- If PSs are good: ATE can be estimated as the difference of the weighted means

Inverse Probability of Treatment Weighing

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

Empirical Distribution of Propensity Scores

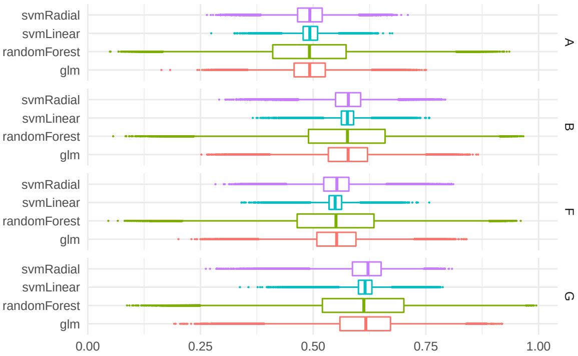

Figure 2: Propensity Scores

Comments about Propensity Scores

What method leads to more extreme PS, i.e. close to 0 or 1?

1. Random forest
2. Logistic regression
3. SVM (radial kernel)
4. SVM (linear kernel)

Empirical Distribution of IPTW

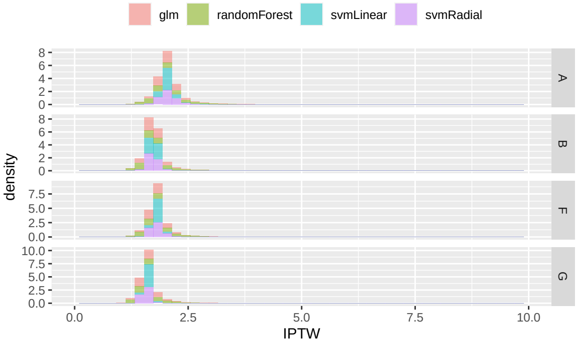

Figure 3: Empirical Distribution of IPTW

Performance Metric of IPTW

Metric	Scenarios	Model			
		Logistic regression	Random forests	SVM (Linear)	SVM (Radial)
bias	A	2.400	2.532	2.429	2.403
	B	2.400	2.537	2.379	2.400
	F	2.400	2.533	2.381	2.399
	G	2.399	2.540	2.285	2.400
estimate	A	0.000	0.006	0.001	-0.004
	B	0.000	-0.025	0.003	0.001
	F	0.000	-0.015	0.001	0.009
	G	0.003	-0.039	0.043	-0.005
mse	A	4.205	5.083	4.318	4.181
	B	4.337	5.325	4.219	4.268
	F	4.276	5.197	4.169	4.212
	G	4.524	5.759	3.953	4.398
sd	A	2.011	2.218	2.039	2.006
	B	2.044	2.277	2.014	2.026
	F	2.029	2.247	2.002	2.011
	G	2.088	2.372	1.938	2.059

Empirical Distribution of Weights for the Control Group

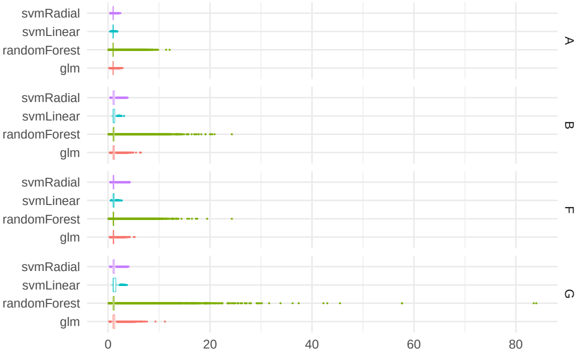

Figure 4: Weights for the Control Group

Comments about Weights

- Recall that extreme PS
  1. Random forest
  2. Logistic regression
  3. SVM (radial kernel)
  4. SVM (linear kernel)
- This result is same in the weight.

Performance Metric of Weights

Metric	Scenarios	Model			
		Logistic regression	Random forests	SVM (Linear)	SVM (Radial)
bias	A	1.385	1.448	1.399	1.388
	B	1.553	1.634	1.541	1.552
	F	1.502	1.576	1.492	1.497
	G	1.626	1.717	1.549	1.630
estimate	A	0.000	-0.063	-0.014	-0.003
	B	0.000	-0.081	0.012	0.000
	F	0.000	-0.074	0.010	0.005
	G	0.002	-0.089	0.079	-0.002
mse	A	1.151	1.385	1.165	1.139
	B	1.579	2.056	1.538	1.527
	F	1.435	1.814	1.402	1.385
	G	1.863	2.623	1.621	1.769
sd	A	0.996	1.127	1.008	0.991
	B	1.191	1.398	1.170	1.169
	F	1.129	1.307	1.111	1.105
	G	1.304	1.589	1.180	1.269

IPW

```
ipw_logit <-
  wt_logit %>%
  compute_ipw(
    treatment = "exposure", outcome = "y", weight = "iptw"
    mc_col = "mcname", sc_col = "scenario"
  )
```

Empirical Distribution of IPW

Figure 5: Empirical Distribution of IPW

Performance Metric of IPW

Metric	Scenarios	Model			
		Logistic regression	Random forests	SVM (Linear)	SVM (Radial)
bias	A	9.13	9.82	8.73	8.54
	B	4.22	4.24	6.00	5.96
	F	5.08	5.21	6.93	7.05
	G	9.86	9.72	13.43	13.15
estimate	A	-8.73	-9.42	-8.31	-8.12
	B	-3.41	-3.23	-5.41	-5.41
	F	-4.47	-4.51	-6.46	-6.59
	G	-9.46	-9.32	-13.03	-12.75
mse	A	78.56	92.65	74.85	70.61
	B	18.09	19.31	35.86	35.68
	F	25.87	28.32	47.50	49.37
	G	91.53	91.52	168.67	162.90

SIPW

```
sipw_logit <-
  wt_logit %>%
  compute_sipw(
    treatment = "exposure", outcome = "y", weight = "iptw",
    mc_col = "mcname", sc_col = "scenario"
  )
```

Empirical Distribution of IPW

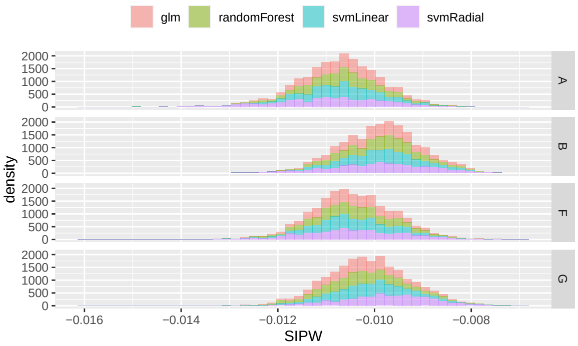

Figure 6: Empirical Distribution of IPW

Performance Metric of SIPW

Metric	Scenarios	Model			
		Logistic regression	Random forests	SVM (Linear)	SVM (Radial)
bias	A	0.411	0.411	0.411	0.411
	B	0.410	0.410	0.410	0.410
	F	0.410	0.410	0.410	0.410
	G	0.410	0.410	0.410	0.410
estimate	A	-0.011	-0.011	-0.011	-0.011
	B	-0.010	-0.010	-0.010	-0.010
	F	-0.010	-0.010	-0.010	-0.010
	G	-0.010	-0.010	-0.010	-0.010
mse	A	0.152	0.152	0.151	0.152
	B	0.152	0.152	0.152	0.152
	F	0.152	0.152	0.152	0.152
	G	0.152	0.152	0.152	0.152

Related Contents

About this project

Project repository

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

Project package

https://github.com/ygeunkim/propensityml

About the Machine

```
sessioninfo::session_info()[[1]]
#>  setting  value
#>  version  R version 4.0.3 (2020-10-10)
#>  os       macOS Catalina 10.15.7
#>  system   x86_64, darwin17.0
#>  ui       X11
#>  language (EN)
#>  collate  en_US.UTF-8
#>  ctype    en_US.UTF-8
#>  tz       Asia/Seoul
#>  date     2020-12-08
```

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.