

Causal_Inference_IPTW

Steph Low

2024-10-25

Goal: evaluate the impact of National Supported Work (NSW) Demonstration, which is a labor training program on post-intervention income levels. The interest is in estimating the causal effect of this training program on income using Inverse Probability Treatment Weighting

The Lalonde data set has 614 subjects and 10 variables:

- age : age in years
- educ : years of schooling
- black : indicator variable for blacks
- hispan : indicator variable for Hispanics
- married : indicator variable for marital status
- nodegree : indicator variable for high school diploma
- re74 : real earnings in 1974
- re75 : real earnings in 1975
- re78 : real earnings in 1978 (the outcome variable; post-intervention income)
- treat : indicator variable for treatment status

Potential confounding variables are age, educ, black, hispan, married, nodegree, re74, and re75.

Data Preparation

First I need to prepare the data.

```
# Load data
data(lalonde)

# view data
# View(LaLonde)

# convert indicator variables to numeric and prep data
age <- lalonde$age
educ <- lalonde$educ
re74 <- lalonde$re74
re75 <- lalonde$re75
married <- lalonde$married
nodegree <- lalonde$nodegree
black <- as.numeric(lalonde$race=="black") # 1 for black, 0 otherwise
hispan <- as.numeric(lalonde$race=="hispan") # 1 for hispanic, 0 otherwise
treatment <- lalonde$treat
outcome <- lalonde$re78

# create a dataset with these variables for simplicity
data <- cbind(age, educ, re74, re75, married, nodegree, black, hispan, treatment, outcome)
data <- data.frame(data)

# get covariates
xvars <- c("age", "educ", "black", "hispan", "married", "nodegree", "re74", "re75")
```

Propensity Score Estimation

Next I want to fit a propensity score model. I'll estimate the propensity score using logistic regression where the outcome is treatment, including the 8 confounding variables in the model as predictors, with no interaction terms or non-linear terms (such as squared terms). Then I'll use the propensity scores to obtain the IPTW weights for each subject and create a Table One with SMD. In this case, since I'm working with weighted data, I will ignore the standard deviations of treated and control subjects, but the SMD is reliable.

```
# fit a propensity score model using logistic regression
psmodel <- glm(treatment~age+educ+re74+re75+married+nodegree+black+hispan
               , family=binomial(link = "logit")
               , data=data
               )

# show model summary
summary(psmodel)
```

```
##
## Call:
## glm(formula = treatment ~ age + educ + re74 + re75 + married +
##      nodegree + black + hispan, family = binomial(link = "logit"),
##      data = data)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.729e+00  1.017e+00  -4.649 3.33e-06 ***
## age          1.578e-02  1.358e-02   1.162  0.24521
## educ         1.613e-01  6.513e-02   2.477  0.01325 *
## re74        -7.178e-05  2.875e-05  -2.497  0.01253 *
## re75         5.345e-05  4.635e-05   1.153  0.24884
## married     -8.321e-01  2.903e-01  -2.866  0.00415 **
## nodegree     7.073e-01  3.377e-01   2.095  0.03620 *
## black        3.065e+00  2.865e-01  10.699 < 2e-16 ***
## hispan       9.836e-01  4.257e-01   2.311  0.02084 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 751.49  on 613  degrees of freedom
## Residual deviance: 487.84  on 605  degrees of freedom
## AIC: 505.84
##
## Number of Fisher Scoring iterations: 5
```

```
# get the value of the propensity score for each subject
ps <- predict(psmodel, type="response")

# calculate the IPTW weights: 1/ps for treated, 1/(1-ps) for control
weights <- ifelse(treatment==1,1/(ps), 1/(1-ps))

# find the min and max weights
min(weights)
```

```
## [1] 1.009163
```

```
max(weights)
```

```
## [1] 40.07729
```

```
# apply the weights to the data
weighted_data<-svydesign(ids=~1, data=data, weights=~weights)

# weighted table 1
weightedtable1 <- svyCreateTableOne(vars=xvars
                                   , strata = "treatment"
                                   , data = weighted_data
                                   , test = FALSE)

# show weighted table with SMD
print(weightedtable1, smd=TRUE)
```

```
##              Stratified by treatment
##              0              1              SMD
## n              616.00              553.63
## age (mean (SD))  27.10 (10.80)  25.57 (6.53)  0.172
## educ (mean (SD))  10.29 (2.74)  10.61 (2.05)  0.132
## black (mean (SD))   0.40 (0.49)   0.45 (0.50)  0.101
## hispan (mean (SD))  0.12 (0.32)   0.12 (0.33)  0.014
## married (mean (SD)) 0.41 (0.49)   0.31 (0.47)  0.197
## nodegree (mean (SD)) 0.62 (0.48)   0.57 (0.50)  0.112
## re74 (mean (SD))  4552.74 (6337.09) 2932.18 (5709.42) 0.269
## re75 (mean (SD))  2172.04 (3160.14) 1658.07 (3072.89) 0.165
```

CI and Truncation

Next I want use IPTW to find the estimate and 95% confidence interval for the average causal effect. This can be done using a Marginal Structural Model, which is used to model average causal effects. This confidence interval turns out to be (-1562, 2012).

Then I want to truncate the weights at the 1st and 99th percentiles and see how that impacts the estimate and 95% confidence interval.

```
# fit a marginal structural model
msm = svyglm(outcome~treatment
              , design = svydesign(~1, data=data, weights=~weights))

# find the confidence interval
confint(msm)
```

```
##              2.5 %   97.5 %
## (Intercept)  5705.529 7140.149
## treatment   -1562.856 2012.208
```

```
# truncate at 1st and 99th percentiles
truncated_model <- ipwpoint(exposure = treatment
                           , family = "binomial"
                           , link = "logit"
                           , denominator = ~age + educ + re74 + re75 + married + nodegree + black + hispan
                           , data=data
                           , trunc = 0.01
                           )
summary(truncated_model$weights.trunc)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  1.012   1.052   1.170   1.818   1.623  12.631
```

```
# add truncated weights to data
data$weights <- truncated_model$weights.trunc

# fit a MSM
truncated_msm <- svyglm(outcome~treatment
                        , design = svydesign(~1, data=data, weights=~weights) )
```