Section 8

Matching and Weighting Estimators

Sooahn Shin

GOV 2003

Nov 4, 2021

Overview

- Logistics:
  - **Pset 7 released!** Due at 11:59 pm (ET) on Nov 10
- Today's topics:
  1. Matching estimators
  2. Weighting estimators

[Review] Observational studies

- **Identification**
  - Most common observational assumptions:
    - No unmeasured confounders: $D_i \perp\!\!\!\perp \big(Y_i(0), Y_i(1)\big) \mid \mathbf{X}_i$
    - Overlap/positivity: $0 < \mathbb{P}(D_i = 1 \mid \mathbf{X}_i = x) < 1$
  - Estimand:
    - ATE $= \mathbb{E}[Y_i(1) - Y_i(0)]$ (We identified this in Module 5)
    - ATT $= \mathbb{E}[Y_i(1) - Y_i(0) \mid D_i = 1]$
    - ATC $= \mathbb{E}[Y_i(1) - Y_i(0) \mid D_i = 0]$

[Review] Observational studies

- **Identification**
  - Most common observational assumptions:
    - No unmeasured confounders: $D_i \perp\!\!\!\perp \big(Y_i(0), Y_i(1)\big) \mid \mathbf{X}_i$
    - Overlap/positivity: $0 < \mathbb{P}(D_i = 1 \mid \mathbf{X}_i = x) < 1$
  - Estimand:
    - ATE $= \mathbb{E}[Y_i(1) - Y_i(0)]$ (We identified this in Module 5)
    - ATT $= \mathbb{E}[Y_i(1) - Y_i(0) \mid D_i = 1]$
    - ATC $= \mathbb{E}[Y_i(1) - Y_i(0) \mid D_i = 0]$
- **Estimation**
  - Regression estimators: $\widehat{\mu}_1(\mathbf{x})$ and $\widehat{\mu}_0(\mathbf{x})$ (Module 5)
  - Matching estimator (for ATT):

$$\widehat{\tau}_m = \frac{1}{n_1} \sum_{i=1}^{n} D_i \left(Y_i - \frac{1}{M} \sum_{j \in \mathcal{J}_M(i)} Y_j \right)$$

  - Weighting estimators:
    - Horvitz-Thompson estimator (= IPW estimator)
    - Hajek estimator (normalized weights)

Types of matching

- Exact matching: choose matches that have the same value of $\mathbf{X}_i$
    - Cf: Coarsened Exact Matching (CEM)

Types of matching

- Exact matching: choose matches that have the same value of $\mathbf{X}_i$
  - Cf: Coarsened Exact Matching (CEM)
- Mahalanobis distance matching: use distance metrics in case of high dimensional $\mathbf{X}_i$

Types of matching

- Exact matching: choose matches that have the same value of $\mathbf{X}_i$
  - Cf: Coarsened Exact Matching (CEM)
- Mahalanobis distance matching: use distance metrics in case of high dimensional $\mathbf{X}_i$
- Propensity score matching:

$$\big(Y_i(0), Y_i(1)\big) \perp\!\!\!\perp D_i \mid \mathbf{X}_i \Rightarrow \big(Y_i(0), Y_i(1)\big) \perp\!\!\!\perp D_i \mid \pi(\mathbf{X}_i)$$

  - This holds under **true** propensity score $\pi(\mathbf{X}_i)$.
  - We need to estimate it ($\widehat{\pi}(\mathbf{X}_i)$): e.g., using logistic regression (can add interactions) or machine learning.
  - Have to check if $\mathbf{X}_i$ is actually balanced.

Types of matching

- Other choices
  - Matching ratio: m-to-one matching
  - w/ or w/o replacement: consider the number of control units
  - Caliper: drop poor matches (estimand changes)

Types of matching

- Other choices
  - Matching ratio: m-to-one matching
  - w/ or w/o replacement: consider the number of control units
  - Caliper: drop poor matches (estimand changes)
- Algorithm:
  - Greedy algorithm: pair two units with the shortest distance, set them aside, and repeat $\rightsquigarrow$ depends on order and thus may not be optimal
  - Optimal matching:
    - $\mathbf{D}$: $n \times n$ matrix of pairwise distance or a cost matrix
    - Select n elements of $\mathbf{D}$ such that there is only one element in each row and one element in each column and the sum of pairwise distances is minimized $\rightsquigarrow$ linear sum assingment problem

Types of matching

- Other choices
  - Matching ratio: m-to-one matching
  - w/ or w/o replacement: consider the number of control units
  - Caliper: drop poor matches (estimand changes)
- Algorithm:
  - Greedy algorithm: pair two units with the shortest distance, set them aside, and repeat $\rightsquigarrow$ depends on order and thus may not be optimal
  - Optimal matching:
    - $\mathbf{D}$: $n \times n$ matrix of pairwise distance or a cost matrix
    - Select n elements of $\mathbf{D}$ such that there is only one element in each row and one element in each column and the sum of pairwise distances is minimized $\rightsquigarrow$ linear sum assingment problem
- Assessing balance
  - standardized mean differences
  - Kolmogorov–Smirnov statistic (comparing distributions)

Matching estimators

- Workflow (in general):
  1. Check the balance before the matching
  2. Choose matching type (compute/estimate the distance/balancing score if necessary)
  3. Conduct matching (check the matched dataset)
  4. Check the balance after the matching
  5. Estimate ATT using matching estimator ($\widehat{\tau}_m$)
  6. Estimate the standard error
  - w/o replacement: cluster bootstrap
  - w/ replacement: use Abadie and Imbens (2006) estimator

Matching estimators

- Workflow (in general):
  1. Check the balance before the matching
  2. Choose matching type (compute/estimate the distance/balancing score if necessary)
  3. Conduct matching (check the matched dataset)
  4. Check the balance after the matching
  5. Estimate ATT using matching estimator ($\widehat{\tau}_m$)
  6. Estimate the standard error
  - w/o replacement: cluster bootstrap
  - w/ replacement: use Abadie and Imbens (2006) estimator
- Useful packages:
  - `cobalt`: for balance check (`bal.tab()` and `love.plot()`)
  - `MatchIt`: for matching
  - `Matching`: for matching + estimating
  - Machine learning packages for estimating $\widehat{\pi}(\mathbf{X}_i)$: e.g., `randomForest`
  - Optimal matching: `clue::solve_LSAP()`

Example: LaLonde dataset

- The effectiveness of a job trainining program (National Supported Work Demonstration; NSW) on wage increases.
- The federal government instituted a randomized evaluation of this program
- How well the result may be recovered when the experimental controls are replaced with a set of observational controls (Population Survey of Income Dynamics; PSID)?
- **Problem**: Imbalances between the experimental and observational data $\rightsquigarrow$ use matching

Example: LaLonde dataset

- Data:
  - Treated: 185 units from NSW
  - Control: 2490 units from PSID
  - Treatment: Participation in the job training program (`nsw`)
  - Outcome: 1978 earnings (in dollars; `re78`)
  - Pre-treatment covariates: age, race, marriage, past earnings, past employment

Example: Balance before matching

```
library(cobalt)
bal.tab(x = dat[,pretreat_covariates],
        treat = dat$nsw, continuous = "std", binary = "std")
```

Example: Balance before matching

```
## Balance Measures
##            Type Diff.Un
## age      Contin. -1.0094
## educ     Contin. -0.6805
## black     Binary  1.4816
## hisp      Binary  0.1288
## married   Binary -1.8453
## re74     Contin. -1.7178
## re75     Contin. -1.7744
## u74       Binary  1.6454
## u75       Binary  1.2309
##
## Sample sizes
##     Control Treated
## All    2490     185
```

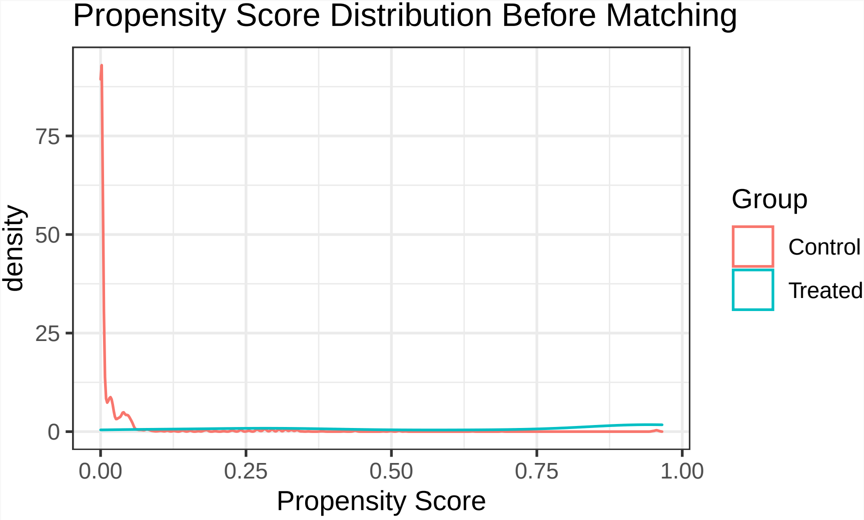
Example: Propensity score matching

```r
# Estimate propensity score using logistic regression
# Propensity score
pscores <- glm(nsw ~ age + I(age^2) + black + married + hisp + u74,
                   family = binomial(), data = dat)$fitted.values
# Conduct one-to-one nearest neighbor propensity score matching
library(Matching)
match_ps <- Match(Y=dat$re78, Tr=dat$nsw,
                   X=pscores, M=1, replace = TRUE, ties = FALSE)
summary(match_ps)
```

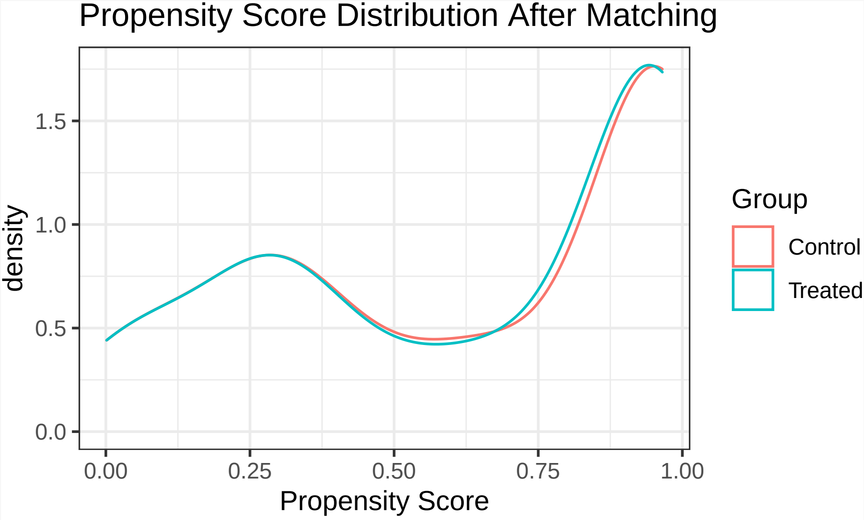
Example: Propensity score matching

```
##
## Estimate...  -778.31
## SE.........  833.23
## T-stat.....  -0.93409
## p.val......  0.35026
##
## Original number of observations..............  2675
## Original number of treated obs...............  185
## Matched number of observations...............  185
## Matched number of observations  (unweighted).  185
```

Example: Propensity score matching



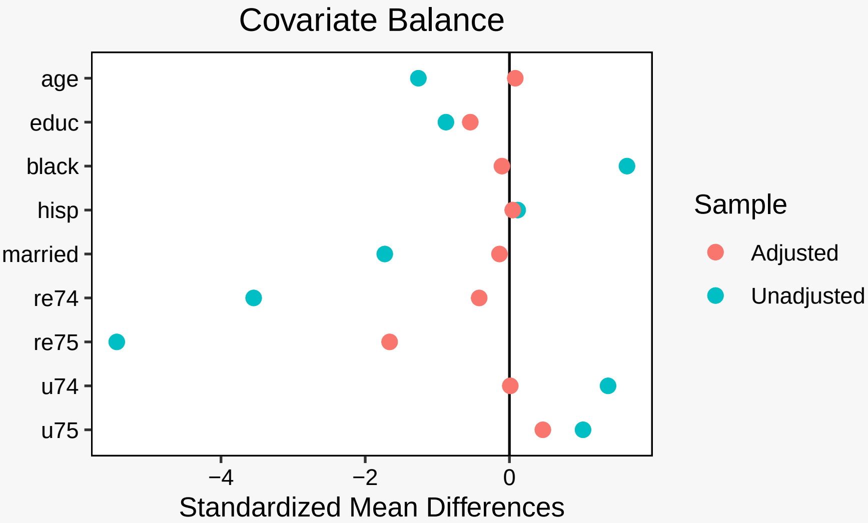
Example: Propensity score matching



Example: Propensity score matching

```r
library(cobalt)
love.plot(nsw ~ age+educ+black+hisp+married+re74+re75+u74+u75,
  data = dat,
  stats = "mean.diffs",
  weights = data.frame(Matched = get.w(match_ps)),
  method = c("matching"), binary = "std")
```

Example: Propensity score matching



Weighting estimators

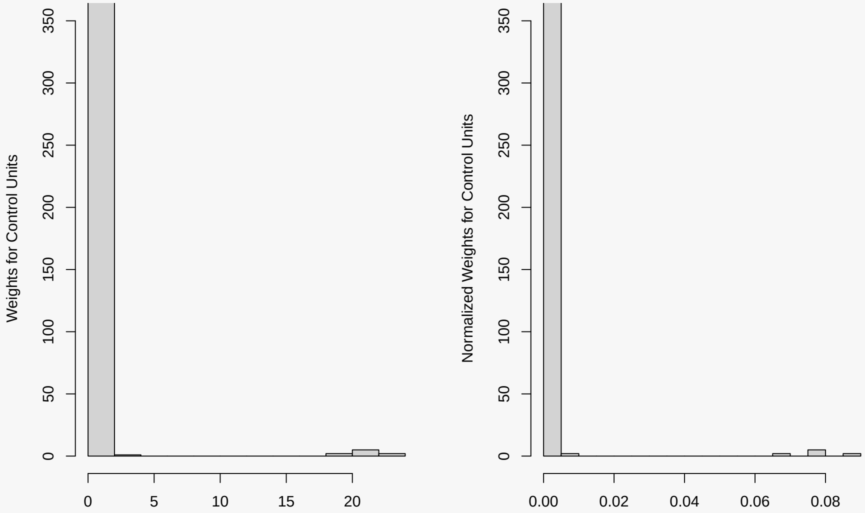
- Matching is actually a special case of a weighting estimator
- Horvitz-Thompson estimator: weight by inverse propensity score.

$$\widehat{\text{ATE}} = \widehat{\tau}_{ipw} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{D_i Y_i}{\widehat{\pi}(\mathbf{X}_i)} - \frac{(1-D_i)Y_i}{1-\widehat{\pi}(\mathbf{X}_i)}\right)$$

  - Would be unbiased if we knew the true propensity scores, $\pi(\mathbf{X}_i)$ (Pset 7 Q2 Bonus)
  - Under no unmeasured confounders, $\widehat{\tau}_{ipw} \overset{p}{\to} \tau$ (consistent)
  - Hajek estimator: normalizes the weights
- Potential of extreme weights due to lack of overlap: $\pi(\mathbf{X}_i)$ close to 0 or 1
  - Windsorizing: trim weights beyond 5th and 95th percentile

Example

```
# Generating propensity score weights for the ATT
W.out <- WeightIt::weightit(nsw ~ age + I(age^2) + black + married + hisp +
                    method = "ps", estimand = "ATT")
```



Estimating ATT with Weights

```
## append estimated weights
dat <- dat %>% mutate(weights = W.out$weights)
## ATT
att_ipw <- function(dat, indices = NULL) {
  if (is.null(indices)) indices <- 1:nrow(dat)
  dat <- dat[indices,]

  weights   <- dat %>% filter(treat == 0) %>% pull(weights)
  reweights <- weights / sum(weights)
  Y1 <- dat %>% filter(treat == 1) %>% pull(re78)
  Y0 <- dat %>% filter(treat == 0) %>% pull(re78)
  att_ht  <- (sum(Y1) - sum(Y0 * weights)) / nobs
  att_hjk <- mean(Y1) - sum(reweights * Y0)
  return(c(att_ht, att_hjk))
}
## Use bootsrap for estimating SE
```