2. Matching and weighting estimators

LPO 8852: Regression II

Sean P. Corcoran

Selection bias

Lecture 1 showed why the simple difference in means between treated and untreated cases does not identify the ATT (or ATE):

$$E(Y|D=1) - E(Y|D=0) =$$
$$E[Y(1)|D=1] - E[Y(0)|D=0] = ATT + \underbrace{E[Y(0)|D=1] - E[Y(0)|D=0]}_{\text{selection bias}}$$

Selection bias reflects differences in $Y(0)$ between the $D=1$ and $D=0$.

- Randomization of D eliminates selection bias!
- Regression can help under very strong conditions about potential outcomes.

Matching and weighting

Matching and weighting estimators construct comparison groups that are *balanced* on a set of observable variables. There are lots of ways to do this:

- Selecting specific matches
- Constructing a matched weighted sample
- Subclassification

Key assumption for causal interpretation: once we have conditioned on observables—by selecting matches, constructing weights, or stratifying—treatment assignment and potential outcomes are independent. This is the conditional independence assumption (CIA).

A note on weighted averages

What is a weighted average? Given a weight w_i for each observation i, the weighted average for Y is:

$$\frac{\sum_{i=1}^{n} w_i Y_i}{\sum_{i=1}^{n} w_i}$$

Weights are used for lots of reasons (Solon, Haider, & Wooldridge, 2015). In matching we may choose weights based on the values of confounders to eliminate differences in X between treated and untreated groups.

Example 1: private vs. public colleges, revisited

This is a stylized version of the private college example in Lecture 1:

		Private			Public			
	Student	Ivy	Leafy	Smart	All State	Tall State	Altered State	Earnings
A	1		Reject	Admit		Admit		110000
	2		Reject	Admit		Admit		100000
	3		Reject	Admit		Admit		110000
B	4	Admit			Admit		Admit	60000
	5	Admit			Admit		Admit	30000
C	6		Admit					115000
	7		Admit					75000
D	8	Reject			Admit	Admit		90000
	9	Reject			Admit	Admit		60000

Source: *Mastering Metrics* (2015). Shaded cell represents the student's chosen college, from those they were admitted to. Based on Dale & Krueger (2002).

Example 1

In the above table:

$$E[Y(1)|D=1] - E[Y(0)|D=0] = 92,000 - 72,500 = 19,500$$

$$= ATT + \underbrace{E[Y(0)|D=1] - E[Y(0)|D=0]}_{\text{selection bias}}$$

It is likely the treated group has a higher $Y(0)$ than the untreated group. This is suggested above by the higher mean earnings for students who applied and were admitted to private colleges (esp. groups A and C).

Example 1

What if we could create equivalent groups by <u>conditioning</u> on some X? For example, what if:

$$\underbrace{E[Y(0)|D=1,X]}_{\text{unobserved}} = \underbrace{E[Y(0)|D=0,X]}_{\text{observed!}}$$

In other words, there is no difference in potential outcomes $Y(0)$ between $D=0$ and $D=1$, once we condition on X. Then we could contrast the mean Y for each set of X and then average them.

In the private vs. public college example, assume there is no difference in $Y(0)$ conditional on application/admitted group A-D:

Example 1

	Student	Ivy	Leafy	Smart	All State	Tall State	Altered State	Earnings
A	1		R	A		A		110000
	2		R	A		A		100000
	3		R	A		A		110000
B	4	A			A		A	60000
	5	A			A		A	30000
C	6		A					115000
	7		A					75000
D	8	R			A	A		90000
	9	R			A	A		60000

Avg(Y|D=1, Group=A)=105,000
Avg(Y|D=0, Group=A)=110,000. Difference = 105,000 - 110,000 = -5,000

Avg(Y|D=1, Group=B)=60,000
Avg(Y|D=0, Group=B)=30,000. Difference = 60,000 - 30,000 = 30,000

Example 1

The simple average of the within-group differences (groups A and B) is:

$$(-5,000 + 30,000)/2 = \$12,500$$

A *weighted* average gives more weight to the group with more students:

$$(-5,000) * (3/5) + (30,000) * (2/5) = \$9,000$$

Another weighted average assigns weights to groups according to the number of *treated* students:

$$(-5,000) * (2/3) + (30,000) * (1/3) = \$6,666$$

Example 1

Weighted averages use the data more efficiently, and also generalize appropriately to the groups included in the calculation. Note groups C and D are either all treated (private college) or all untreated (public college). There is no **common support** here. This term will come up again.

Example 1

Note in this example that neither the weighted nor unweighted average of groups A and B estimate the ATE or ATT for this population. This is due to the lack of common support.

- Without a counterfactual for the treated in group C, we can't estimate ATT (or ATE)
- Without a counterfactual for the untreated in group D, we can't estimate ATU (or ATE)

An illustration of the importance of being attentive to the population to which you are able to generalize with the data you have.

Example 1

Mastering Metrics explains how regression estimates are weighted averages of multiple matched comparisons. E.g., consider the regression:

$$Y_i = \alpha + \beta P_i + \gamma A_i + e_i$$

where $P_i = 1$ if the student attended a private college and $A_i = 1$ if the student was in group A (versus B). Groups C and D are excluded.

Using the Example 1 data, $\hat{\beta} = 10,000$. This is comparable to the averages found earlier, but is not identical to any of them. Regression effectively applies different weights. It estimates a **variance-weighted treatment effect**.

Example 1

The weighted averages in Example 1 can be characterized as:

- A **matching** approach: each treated case (private college attendee) is matched to one or more untreated case (public college attendee) with the same observable characteristic (application/admission group). Then the mean outcomes of the two matched groups are compared.

- A **subclassification** approach: cases are stratified according to some observable characteristic (application/admission group), mean differences are calculated within each group, and then averaged across groups (e.g., weighting by the number of treated cases).

Identifying assumption: conditional on application/admissions group, potential outcomes are balanced across treated and untreated cases. Treatment assignment is "as good as random."

Example 2: Catholic schools

Murnane & Willett (ch. 12) stratify the NELS sample by family income to estimate the effect of Catholic high school attendance on 12th grade math achievement:

Table 12.1 Descriptive statistics on annual family income, by stratum, overall and by type of high school attended, and average twelfth-grade mathematics achievement by income stratum and by high-school type ($n = 5{,}671$)

Stratum		Average Base-Year Annual Family Income *(1988 dollars, 15-point ordinal scale)*			Cell Frequencies		Average Mathematics Achievement (12th grade)		
Label	Income Range	Sample Variance	Sample Mean		Public	Catholic (% of stratum total)	Public	Catholic	Diff.
			Public	Catholic					
Hi_Inc	\$35,000 to \$74,999	0.24	11.38	11.42	1,969	344 *(14.87%)*	53.60	55.72	2.12***,†
Med_Inc	\$20,000 to \$34,999	0.22	9.65	9.73	1,745	177 *(9.21%)*	50.34	53.86	3.52***,†
Lo_Inc	≤\$19,999	3.06	6.33	6.77	1,365	71 *(4.94%)*	46.77	50.54	3.76***,†
							Weighted Average ATE		***3.01***
							Weighted Average ATT		***2.74***

~p <0.10; *p <0.05; **p <0.01; ***p <0.001
†One-sided test.

Example 2

Calculate the difference within each strata and then the weighted average of these differences across strata.

The ATE ($\hat{\beta}_{CATH} = 3.01$) uses *total* cell sizes as weights; ATT ($\hat{\beta}_{CATH} = 2.74$) uses counts of *treated* cases in each cell as weights. These estimates are smaller than the unconditional mean differences in math scores ($\hat{\beta}_{CATH} = 3.895$), suggesting upward bias.

Note income is a continuous variable. M&W created three income strata with the aim of (1) creating balance in family income within each strata; (2) maintaining common support.

Identifying assumption: conditional on income (strata), enrollment in Catholic school is "as good as random" (!).

Example 2

One could stratify on multiple covariates, as M&W do here with income and a measure of prior achievement (12 total cells):

Table 12.2 Sample frequencies and average twelfth-grade mathematics achievement, by high-school type, within 12 strata defined by the crossing of stratified versions of base-year annual family income and mathematics achievement ($n = 5{,}671$)

Stratum		Cell Frequencies		Average Mathematics Achievement (12th Grade)		
Base-Year Family Income	Base-Year Mathematics Achievement	Public	Catholic	Public	Catholic	Diff.
Hi_Inc	*Hi_Ach*	1,159	227	58.93	59.66	0.72
	MHi_Ach	432	73	49.18	50.71	1.53*,†
	MLo_Ach	321	38	42.75	44.23	1.48
	Lo_Ach	57	6	39.79	40.40	0.62
Med_Inc	*Hi_Ach*	790	93	57.42	59.42	2.00**,†
	MHi_Ach	469	49	47.95	50.14	2.19**,†
	MLo_Ach	390	33	41.92	44.56	2.64*,†
	Lo_Ach	96	2	37.94	39.77	1.83
Lo_Inc	*Hi_Ach*	405	36	56.12	56.59	0.47
	MHi_Ach	385	13	47.12	48.65	1.53
	MLo_Ach	433	21	40.99	41.70	0.71
	Lo_Ach	142	1	36.81	42.57	5.76
				Weighted Average ATE		***1.50***
				Weighted Average ATT		***1.31***

~$p<0.10$; *$p<0.05$; **$p<0.01$; ***$p<0.001$
†One-sided test.

Curse of dimensionality

Finer strata may provide a stronger argument for the conditional independence assumption that treatment group membership is unrelated to potential outcomes (within strata), but they make it more and more difficult to achieve common support—the **curse of dimensionality**.

Approaches to matching

There are many approaches to constructing matched comparison groups:

- Exact matching
- Coarsened exact matching
- Nearest neighbor/distance matching
- Propensity score matching

Exact matching

As the name suggests, **exact matching** entails pairing each treated observation with one or more untreated observations with the same X (one or more matching variables). Estimate the ATT with:

$$\widehat{ATT} = \frac{1}{N_T} \sum_{D_i=1} (Y_i - Y_{j(i)})$$

where $Y_{j(i)}$ represents the Y for the matched case(s) for treated observation i. If multiple exact matches are used, $Y_{j(i)}$ stands in for the average of these.

Note: this could also be done for untreated observations to estimate ATU.

Nearest neighbor matching

Nearest neighbor, approximate, or distance matching relaxes the need for an exact match and identifies "nearest neighbors" based on one or more matching variables X. Some distance measures:

- Euclidean distance
- *Normalized* Euclidean distance
- Mahalanobis distance

$\widehat{ATT}$ is the same as the previous slide, but the matched case(s) used for $Y_{j(i)}$ are based on distance (e.g., nearest neighbor) criteria.

Euclidean and normalized Euclidean distance

Suppose you have a vector X of k variables for two units i and j. The Euclidean distance between X_i and X_j is:

$$||X_i - X_j|| = \sqrt{\sum_{m=1}^{k} (X_{mi} - X_{mj})^2}$$

These X_m variables are likely on different scales. The *normalized* Euclidean distance scales each variable by its variance:

$$\sqrt{\sum_{m=1}^{k} \frac{(X_{mi} - X_{mj})^2}{\sigma_m^2}}$$

Mahalanobis distance

Suppose you have a vector X of k variables for two units i and j. The Mahalanobis distance between X_i and X_j is:

$$d(X_i, X_j) = \sqrt{(X_i - X_j)' C^{-1} (X_i - X_j)}$$

Loosely, this is the sum of squared distances between values in X_i and X_j normalized by the covariance. (C is the covariance matrix for the matching variables in X). If there is no covariance between the X, this reduces to the normalized Euclidean distance.

Why "take out" the covariance? Suppose there is some latent characteristic that shows up in multiple matching variables. If those multiple variables are used to calculate distance, we may be "double-counting" by using distance on all of those variables.

Stata's `teffects` commands

Stata's `teffects` commands implement a wide array of treatment effect estimators using matching, weighting, regression adjustment, etc.

- `teffects nnmatch`: exact and/or nearest neighbor matching
- `teffects psmatch`: propensity score matching
- `teffects ipw`: inverse probability weighting
- ...and others

The `teffects` manual on Stata is actually worth reading! See also my handout: *Stata commands for matching.*

Stata's `teffects nnmatch`

`teffects nnmatch` implements exact or nearest neighbor matching—or a combination of these.

`teffects nnmatch` *(y x) (t)*, *options*

y is the outcome, *x* are the matching variables, and *t* is the treatment indicator. In the options can use `ematch(`*vars*`)` to specify a list of variables on which you desire an exact match. For nearest neighbor matching you can specify the distance metric used, e.g., `metric(euclidean)`. Mahalanobis is the default.

There are lots of other options!

Matching: objectives

Again, there are lots of approaches to creating matched comparison groups. However, there are a few basic principles:

1. You are appealing to the **conditional independence** assumption. So choose matching variables that make this plausible.
2. Given a choice of X, you want to see **balance** in your matched comparison groups. Ideally, you want to see balance in the full distributions of X, not just the means.
3. You want **common support**: there are treated and untreated cases for all values of your X.
4. You want **efficient** estimators (smaller standard errors). Use more of the data when possible, but there is a bias-efficiency tradeoff..

Staying honest with `teffects`

`teffects` will automatically give you a treatment effect estimate based on the procedure you request (e.g., `nnmatch`). The option `ate` or `atet` in the options will request the ATE or ATT, respectively.

A word of caution: matching often involves multiple iterations to obtain better balance. It is not good practice to allow estimates of the treatment effect to guide your decisions about matching!!

You can precede `teffects` with `quietly:` to suppress the output. It will do all of the necessary matching—allowing you to do balance diagnostics—without letting you "cheat" by seeing the ATE.

Alternative commands like `psmatch2` can perform matching without requesting a treatment effect estimate.

Stata's `tebalance summarize`

Can use `tebalance summarize` following `teffects nnmatch`:

```
. tebalance summarize
note: refitting the model using the generate() option

  Covariate balance summary
                                           Raw      Matched
                       ----------------------------------------
                       Number of obs =     200          168
                       Treated obs   =      84           84
                       Control obs   =     116           84
                       ----------------------------------------

  ------------------------------------------------------------
               |Standardized differences        Variance ratio
               |        Raw     Matched          Raw   Matched
  -------------+----------------------------------------------
           age |   .5124947    .0095797     .8829962  1.011965
          educ |   .1125516      .20222     1.038685   1.08452
  ------------------------------------------------------------
```

Note: the *standardized difference* is the difference in means between the treated and untreated groups, divided by the square root of a pooled variance. They can be interpreted in standard deviation units.

Stata's `tebalance summarize`

Try `tebalance summarize, baseline` following `teffects` to see baseline (pre-matching) differences in covariates in original units.

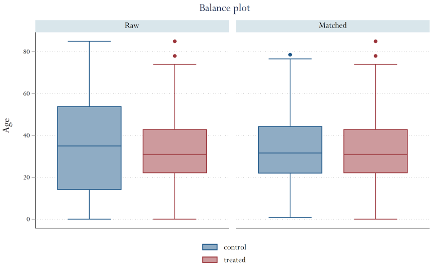
```
. tebalance summarize, baseline
note: refitting the model using the generate() option

  Covariate balance summary
                                           Raw      Matched
                       ----------------------------------------
                       Number of obs =     750          556
                       Treated obs   =     278          278
                       Control obs   =     472          278
                       ----------------------------------------

  ------------------------------------------------------------
               |        Means                 Variances
               |    Control     Treated      Control   Treated
  -------------+----------------------------------------------
           age |   27.49364     30.3705     41.56259  38.57342
  ------------------------------------------------------------
```

Stata's `tebalance summarize`

Note: when there are *multiple* nearest neighbor matches, they should be appropriately weighted so that the sum of the weights of one's neighbors equals one. (In other words, if one treatment observation has five matched untreated neighbors, they will each count as 1/5). Stata should do this automatically in `tebalance`.

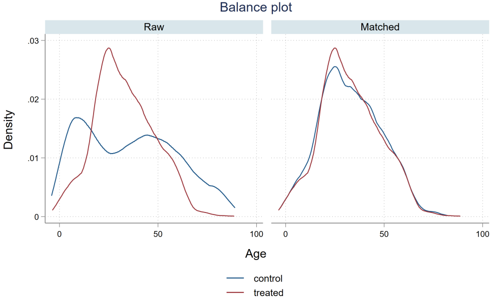
Stata's `tebalance box`

Can use `tebalance box` to get a fuller picture of the matched sample distributions:



Stata's `tebalance density`

Can use `tebalance density` to get a fuller picture of the matched sample distributions:



In-class example 1

See do-file: *Lecture 2 matching with simulated data*. In this example, potential earnings (y) are affected by age and education. There is also a treatment (*treat*) that is positively related to age and education. This do-file illustrates:

- Exact matching on one variable (age)
- Exact matching on multiple variables (age and education)
- Nearest neighbor matching (Euclidean and Mahalanobis)
- Balance checking
- Capturing the observation numbers of nearest neighbors (and distance to these neighbors)

Refining nearest neighbor matches

There are a number of things you can do to control the number and quality of nearest neighbor matches:

- Choose a **caliper** or bandwidth for acceptable matches, in terms of distance. The option `caliper(#)` sets the max distance allowed.
- Can choose the number of nearest neighbors desired with `nneighbor(#)` option (the default is 1). Note ties are used.
- Can take all neighbors within a given caliper (**radius** matching) or *k* nearest neighbors subject to being within the caliper.
- Can perform matching with or without replacement. (`teffects nnmatch` is with replacement).

Refining nearest neighbor matches

There is typically a **bias-variance tradeoff** in these decisions. More matches = larger sample size = less sampling variation. But more matches typically means “worse” matches, so more opportunity for bias.

Matching with replacement = “better” matches. But matching with replacement may mean less variability. I.e., the same observation may be used over and over again as the nearest neighbor.

Abadie & Imbens (2011) bias correction

When the conditional independence assumption holds, the only source of bias when matching comes from imbalance in the covariates (i.e., imperfect matches).

When there is imperfect matching, the treatment effect estimator is a combination of the "true" effect and differences in Y that are a byproduct of the imbalance in covariates.

Abadie & Imbens (2011) propose a consistent bias-corrected estimator. The idea here is that one can use OLS to estimate the relationship between Y and covariates X. The difference in (predicted) Y due to the differences in X (between the perfect and actual match) is used to adjust the treatment effect estimate. In `teffects`: use `biasadj(`*`varnames`*`)` option with *varnames* the list of continuous covariates.

Post-matching predictions

After `teffects nnmatch` you can create new variables that contain each unit's "potential outcomes" (`po`) and "treatment effect" (`te`). Obviously, we can't know these! These are imputed based on the matches.

- Need to specify which potential outcome condition you want (e.g., Y_{i0} or Y_{i1}). Let's call `po0` the potential outcome in the untreated state and `po1` the potential outcome in the treated state.
- For treated observations, `po0` is the mean outcome of their matched untreated observations. `te` is the difference between their actual y and this imputed counterfactual.
- For untreated observations, `po1` is the mean outcome of their matched treated observations. `te` is the difference between their actual y and this imputed counterfactual.

Using mahapick for Mahalanobis matching

FYI, an alternative command for identifying k nearest neighbors using Mahalanobis distance is `mahapick`. It automatically creates the list of matches and can output them to a file.

```
mahapick x1 x2 x3..., idvar(id) treated(treat)
nummatches(#) genfile(filename) score
```

The *x1, x2, x3...* are the matching variables, *id* is the unique observation ID, *treat* is the treatment indicator, and *filename* is where you want to save the resulting list of matches. `score` tells Stata to include the distance score in the output file.

As always with nearest neighbor matching, be aware of how ties are handled, and whether and how sort order matters.

Using psmatch2 for Mahalanobis matching

Another alternative for Mahalanobis matching is `psmatch2`, which is an older command used for propensity score matching. (More on this later).

```
psmatch2 treat , mahalanobis(x1 x2 x3...) neighbor(#)
```

The *x1, x2, x3...* are the matching variables, and *treat* is the treatment indicator. There are lots of options, including radius matching, matching *without* replacement, and more.

As always with nearest neighbor matching, be aware of how ties are handled, and whether and how sort order matters.

Coarsened exact matching

Iacus, King, and Porro (2012) introduced **coarsened exact matching**, in which exact matches are required on continuous variables that have been binned ("coarsened"). See the user-written Stata command `cem`. Ex:

```
cem x1 (#), treatment(treat) showbreaks
```

The option `(#)` is the number of cutpoints for variable *x1*. For example, `(#5)` will use 5 equally-spaced cutpoints. This can be omitted and `cem` will automatically coarsen the data based on a binning algorithm.

Coarsened exact matching

`cem` performs the coarsening and matching and creates weights, but does not estimate the treatment effect. You can do this yourself using the provided weights (*cem_weights*):

```
reg y treat [iweight=cem_weights]
```