2. Matching estimators

LPO 8852: Regression II

Sean P. Corcoran

Selection bias

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

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

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

- Randomization of D would help!
- Regression can help under very strong conditions.

Matching

Matching estimators construct comparison groups that are *similar* according to a set of *matching variables*:

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

The assumption: once we have conditioned on these matching variables—by selecting matches, constructing weights, or stratifying—treatment assignment and potential outcomes are independent. (The conditional independence assumption).

Weighted average

What is a weighted average? Given a weight 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 (i.e., matching variables).

Example 1: re-weighting

Imagine a job training program that serves 100 people where the outcome of interest (Y) is employment.

	Treated ($D_i = 1$)	Untreated ($D_i = 0$)	Diff (Mean Y)
Men	60 $Y = 1$	350 $Y = 1$	
	20 $Y = 0$	150 $Y = 0$	0.05
Women	12 $Y = 1$	275 $Y = 1$	
	8 $Y = 0$	225 $Y = 0$	0.05
Total	100	1000	
Mean(Y)	**0.720**	**0.625**	**0.095**
Mean(Male)	0.800	0.500	

Source: Huntington-Klein ch. 14

Example 1: re-weighting

The simple difference in means is: $E(Y|D = 1) - E(Y|D = 0) = \mathbf{0.095}$. However, you'll notice that for both men and women the treatment effect is only 0.05.

Assuming that *conditional on gender, D* is independent of potential outcomes, then each gender group is like a "randomized trial." The true treatment effect is **0.05** (for both men and women).

The simple difference in means *overstates* the treatment effect because the treatment group is disproportionately male, and males have a higher $Y(0)$ in the untreated state. There is selection bias (of 0.045).

Example 1: re-weighting

Can we re-weight the untreated group so that it "looks like" the treated group?

	Treated	Untreated	Untreated weight
Men	80	500	0.16 (80/500)
Women	20	500	0.04 (20/500)

We want the 500 untreated men to represent the 80 in the treatment group, so each gets a weight of 0.16. We want the 500 untreated women to represent 20 in the treatment group, so each gets a weight of 0.04. Note the weights sum to $(0.16 * 500) + (0.04 * 500) = 100$

Example 1: re-weighting

Using the weights, what the is the proportion male in the untreated group? The proportion employed (Y)?

$$E_w(male|D = 0) = ((500 * 1 * 0.16) + (500 * 0 * 0.04))/100 = 0.80$$

$$E_w(Y|D = 0) = ((350 * 1 * 0.16) + (275 * 1 * 0.04))/100 = 0.67$$

Use this re-weighted mean to get the ATT:

$$ATT = 0.72 - 0.67 = 0.05$$

Note with the weights, the two samples are **balanced** on gender.

Example 1: re-weighting

To re-iterate:

- Gender was the only confounding factor here. Conditional on gender, treatment assignment was "as good as random."
- We adjusted for differences in gender between the two groups using weights.
- The weights were chosen based on the distribution of gender in the treatment group (for ATT).
- We could have chosen weights based on the distribution of gender *overall* for ATE

See the do-file *Lecture 2 weighting example* for this example in Stata.

Example 1b: subclassification

In this example we could alternatively use **subclassification**: grouping treated and untreated observations into strata, calculating differences within strata, and then weighting those differences to get a treatment effect estimate. The weights here are chosen based on the full sample (for ATE):

$$ATE = \underbrace{(0.75 - 0.70) * (580/1100)}_{\text{men}} + \underbrace{(0.60 - 0.55) * (520/1000)}_{\text{women}}$$

$$ATE = 0.05$$

Example 2: private vs. public colleges

		Private			Public			
		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: Angrist & Pischke *MM* (2015). Shaded cell represents the student's chosen college, from those they were admitted to. Based on Dale & Krueger (2002).

Example 2: private vs. public colleges

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 2: private vs. public colleges

What if we could create equivalent groups by conditioning 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 2: private vs. public colleges

		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 2: private vs. public colleges

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 individuals:

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

The weighted average uses the data more efficiently, and also generalizes 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 2: private vs. public colleges

Note in this case that neither the weighted nor unweighted average of groups A and B estimates the ATE or ATT. 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 2: private vs. public colleges

Angrist & Pischke *MM* (2015) explain 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). Students in groups C and D are excluded.

Using the Example 2 data, $\hat{\beta} = 10,000$. This is comparable to the averages on the previous slide, but not identical to either. Regression effectively applies different weights, but the idea is the same. (See *MM* for details).

We will return later to the differences between matching and regression.

Example 3: 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 3: Catholic schools

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

The ATE uses *total* cell sizes as weights; ATT uses counts of *treated* cases in each cell as weights. These 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 strata with the aim of (1) creating balance in family income within each strata; (2) maintaining common support.

Again, we are appealing to the conditional independence assumption. Conditional on income (strata), enrollment in Catholic school is "as good as random" (!).

Return of the "unobservables"

Example 3: Catholic schools

Can also 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***

$^{\cdot}p$ <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**.

Matching on a single variable

Examples 1-3 all created balance on a single variable (gender, sets of colleges, income). There are *lots* of ways to do this. When matching, there are a lot of choices to make:

1. What will the matching criteria be?
2. Will you *select* matches, or use weights to create a matched sample?
3. If selecting matches, how many?
4. If constructing a matched weighted sample, how will weights decay with distance?
5. What is the *worst* acceptable match?

Treatment effect estimation is usually the easy part. The hard part is finding the right matched comparison groups.

What will the matching criteria be?

The goal is to construct comparison groups that are "similar" on matching variables. What does "similar" mean?

- Exact matching
- Coarsened exact matching
- Distance matching (e.g. nearest neighbor)
- Propensity score matching (observations with similar *propensity* to be treated)

Select matches or use weights?

Selecting matches:

- Literally picking observations to be "in" or "out" based on some criteria.
- Usually if an observation is "in" it gets equal weight.
- Intuitively appealing and avoids situation where some observations get very large weights.

Constructing a matched weighted sample:

- Determine how close untreated observations are to treated observations.
- Weight based on similarity, or to make matched sample "look like" treated group.
- Has nice statistical properties and is less sensitive/noisy.

If selecting matches, how many?

- Nearest neighbor? k nearest neighbors? **Radius** matching (all neighbors within a given radius)?
- With replacement or without?

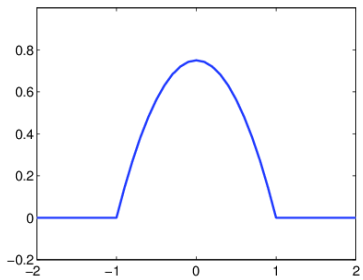
There is typically a **bias-variance tradeoff** in these decisions. More matches = larger sample size = less sampling variation. But more matches typically means "worse" matches, so more opportunity for bias.

With replacement = better matches. But matching with replacement may mean less variability. (The same matched observation may be used multiple times).

How will weights decay with distance?

Typically a distance measure or propensity score is used to construct weights. We often want "less similar" observations to receive less weight.

A **kernel function** can do this. For example, the Epanechnikov kernel is $K(x) = \frac{3}{4}(1 - x^2)$ for $-1 \leq x \leq 1$ and 0 otherwise:



where x is a standardized distance measure. Note the weight is largest when the distance is 0 and then decays as you move away from 0.

How will weights decay with distance?

Propensity scores are often used to construct **inverse probability weights** where each *treated* observation gets a weight of 1 divided by their probability of treatment (p_i) and each *untreated* observation gets a weight of 1 divided by their probability of non-treatment ($1 - p_i$).

As we will see, IPW makes the treated and untreated groups more similar:

- Treated observations with the biggest weights ($1/p_i$) are those that are more like the *untreated* group.
- Untreated observations with the biggest weights ($1/(1 - p_i)$) are those that are more like the *treated* group.

What is the worse acceptable match?

How dissimilar will matches be allowed to be?

- Can choose a **caliper** or **bandwidth** for acceptable matches (in terms of the distance measure or propensity score).
- Note a kernel implies a bandwidth, since the weight is 0 beyond a certain distance.
- Exact matching requires exact matches (as the name implies!)
- Coarsened exact matching requires exact matches on the coarsened continuous variable(s).

Again this decision involves a bias-variance tradeoff. Methods do exist for choosing an optimal bandwidth based on some criteria.

Matching on multiple variables

When matching on *multiple* variables, we have all of the same decisions above to make. But we will need to reduce multiple differences into one dimension. Common approaches:

- Euclidean distance $||X_i - X_j|| = \sqrt{\sum_{m=1}^{k}(X_{mi} - X_{mj})^2}$, though variables are on different scales
- *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**—adjusts for any covariance between x's
- **Propensity scores**

With multiple matching variables, we can even combine criteria, like exact matching for one or more variables and distance matching for the others.

Mahalanobis distance

Take two observations (1 & 2) with X vectors of values X_1 and X_2. The Mahalanobis distance measure is:

$$d(X_1, X_2) = \sqrt{(X_1 - X_2)' C^{-1} (X_1 - X_2)}$$

Loosely, this is the sum of squared distances between values in X_1 and X_2 divided 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.

Propensity scores

Think of the **propensity score** as a "one-number summary" capturing the relationship between treatment and X: $P(X_i) = Pr(D_i = 1|X_i)$. It is the probability of treatment given X.

The propensity score is often estimated using a binary logistic model:

$$P(D_i|X_i) = \frac{1}{1 + e^{-X_i\beta}}$$

Taking the logit tranformation results in a linear function of X:

$$log\left(\frac{P}{1-P}\right) = X_i\beta$$

Curse of dimensionality, revisited

The curse of dimensionality comes up again when trying to match on multiple variables. The more matching variables you have, the less likely it is you will find a "close" match on any one variable. Getting a better match on one variable x_1 may entail a worse match on x_2.

Selecting matches in practice

Let's see some examples of matching when our aim is to *select specific matches* (i.e., we are not just creating weights for the purpose of re-weighting).

NOTE: there are lots of methods and decision points. It is easy to get lost in the weeds. But the objective is ultimately the same throughout:

- **Create balance so that you can appeal to the CIA**!

You also want common support. For example, if you are estimating the ATT, you want overlap between your treated group (their X's, or propensity scores) and your untreated group.

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.

Nearest neighbor matching

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

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

Stata's `teffects nnmatch`

Stata's `teffects` implements a wide array of treatment effect estimators using matching, weighting, regression adjustment, etc. `teffects nnmatch` uses 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 specify `ate` or `atet`, and `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)`. There are lots of other options.

See matching examples on Github using NHIS and simulated data.

Stata's `teffects nnmatch`

More on `teffects nnmatch`:

- The default distance metric is Mahalanobis.
- The default number of nearest neighbor matches is 1, but in the case of *exact* matching, `teffects` will use *all* available exact matches. Note for exact matching to work for ATT, there must be at least one exact match for every treated observation.
- Can include `nneighbor(#)` option to specify the number of neighbors. Note when there are *ties* for distance (or exact matches), `teffects` will take all of the ties as matches.
- Uses matching *with replacement*

Stata's `teffects nnmatch`

More on `teffects nnmatch`:

- Can choose a `caliper(#)` in the options to specify "how bad" the match can be.
- Can include the option `gen(`*`stubname`*`)` to have Stata create new variables with the observation numbers of nearest neighbor matches. Note a change in sort order will change observation numbers. You may wish to assign an id to your observations (and sort) to preserve sort order.

Using `mahapick` for Mahalanobis matching

If your interest is in identifying k nearest neighbors using Mahalanobis distance and you want a less cumbersome way to save the list of specific matches—along with the actual distance score—try `mahapick`

`mahapick` *x1 x2 x3...*`, idvar(`*id*`) treated(`*treat*`) nmatches(#) 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

An alternative to `mahapick` is `psmatch2`, which is also used for propensity score matching.

`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.

Propensity scores

Rosenbaum & Rubin (1983) showed that if $Y(0), Y(1)$ are independent of D conditional on X, then they are also independent of D conditional on a **propensity score** constructed using X.

- Rather than stratifying or matching on all of the variables in X, it is sufficient to use the "one-number summary" of the relationship between treatment and X: $P(X_i) = Pr(D_i = 1|X_i)$
- $P(X_i)$ can be estimated using a logit, probit, or LPM regression from which one can obtain predicted probabilities $\widehat{P(X_i)}$. LPM is not advised if predicted probability falls outside of [0,1].

Stata also refers to the propensity score as the probability of treatment.

Propensity scores

The propensity score estimator for ATT can be written as:

$$E_{P(X)|D=1}\left(\underbrace{E[Y(1)|D=1,P(X)]}_{\text{treated}}-\underbrace{E[Y(0)|D=0,P(X)]}_{\text{untreated}}\right)$$

In theory, *for each propensity score* we calculate the difference in mean outcomes for the treated and untreated with that $P(X)$. We then take a weighted average of these over the different propensity score values. The subscript $_{P(X)|D=1}$ means we are taking a weighted average over the area of common support.

Compare logic to Example 2 where we averaged the group differences in earnings across two groups with common support (A and C), weighting as appropriate.

Propensity scores

In practice $P(X)$ takes on a continuum of values and thus stratifying on $P(X)$ itself—in the manner we did with subclassification—is not feasible.

Thus, we can do other things with the propensity score, including matching and re-weighting. Even when propensity scores are not used to estimate treatment effects, they can be useful diagnostic tools since they force you to think about balance between the treated and untreated groups, and the model of selection into treatment.

Stata's `teffects psmatch`

`teffects psmatch` can estimate propensity scores and produce ATT and ATE via nearest neighbor matching using propensity scores.

`teffects psmatch` *(y) (t x, tmodel), options*

Again y is the outcome, x are the covariates, and t is the treatment indicator. *tmodel* is the type of propensity score model you would like to estimate (e.g., logit, probit). In the options can specify `ate` or `atet` for the treatment effect estimation, the number of nearest neighbors, the caliper, etc.

Stata's `teffects psmatch`

Can obtain predicted propensity scores after `teffects psmatch` using the `predict` command. Requires the `gen()` option in the `teffects psmatch` command, which creates variables containing the index of the nearest neighbor(s):

`predict` *(newvar)*`, ps` *options*

Can also predict *potential outcomes* (`po`), individual treatment effects given potential outcomes (`te`), and distance to nearest neighbor (`distance`).

Using psmatch2 for propensity score matching

I also recommend the older user-written package `psmatch2`, which is useful for refining your propensity score model *before* requesting the ATT estimate. Alternatively, can "quietly" run `teffects` and then diagnose balance with `tebalance`. NOTE, however, that the treatment effect standard errors are incorrect in `psmatch2`. Use `teffects` for the final ATT calculation.

Using psmatch2 for propensity score matching

`psmatch2` creates several variables in your dataset: `_pscore`, `_treated`, `_support`, `_weight`, `_id`, `_n1`, `_nn`, `_pdif`

- `_pscore`: estimated $P(X)$
- `_treated`: flags observations Stata recognized as treated
- `_support`: flags observations on common support
- `_weight`: weight for matched controls (untreated obs only)
- `_id`: id number assigned for identifying matches
- `_n1`: id of nearest neighbor (treated obs only)
- `_nn`: number of matched neighbors
- `_pdif`: absolute value of diff between $P(X)$ and $P(X)$ of NN

As noted earlier, `teffects psmatch` can be augmented with options (and used with the `predict` command to get similar information)