ECON42720 Causal Inference and Policy Evaluation

4 Matching and Re-weighting

Ben Elsner (UCD)

About this Lecture

Resources

As an **introduction**, I recommend Chapter 5 in Scott Cunningham's Mixtape

Slightly **more detailed coverage** can be found in

- Huntington-Klein's The Effect, Chapter 14
- Huber's Causal Analysis, Chapter 4

Many examples in this chapter, in particular the R codes, have been taken from The Effect or inspired by it.

Credits

Stephen Pettigrew produced some very instructive graphs on matching. You can find his slides on matching here. He has lots of interesting materials on causal inference on his website.

Starting Point: Conditional Independence

$$(Y^1, Y^0) \perp\!\!\!\perp D \mid X$$

For **causal identification**, we require the assumption that the **treatment** D is as good as **randomly assigned conditional on the covariates** X

Formally, this means that the potential outcomes are **conditionally independent** of the treatment assignment given the covariates

$$E[Y^1 \mid D = 1, X] = E[Y^1 \mid D = 0, X]$$
$$E[Y^0 \mid D = 1, X] = E[Y^0 \mid D = 0, X]$$

Conditional Independence and Selection on Observables

If CIA holds, we speak of **selection on observables**

- **Independence does not hold** in general
- But it holds in the **subpopulations** defined by the covariates X

The **groups defined by** X (think age, gender, neighbourhood, etc) determine the **treatment assignment**

- But **within each group**, who gets treated is **as good as random**

This is a **strong assumption!**

Example: Smoking and Lung Cancer

Does smoking cause lung cancer?

- Today we would say "yes, of course"
- But answering this question was far from clear in the 1950s
- There is a **strong correlation** between smoking and lung cancer, but is it causal?

(Potential) problem: confounders

- There could be genetic determinants of smoking and lung cancer
- There could be environmental factors that cause both smoking and lung cancer

We don't have **experimental evidence**

Example: Death Rates per 1,000

The following example from Cochran (1968) will illustrate what **selection on observables** and do for us

Smoking group	Canada	UK	US
Non-smokers	20.2	11.3	13.5
Cigarettes	20.5	14.1	13.5
Cigars/pipes	35.5	20.7	17.4

In all countries, the **highest death rates are for cigar and pipe smokers**

- Does this mean that smoking pipes and cigars is more dangerous than smoking cigarettes?
- And given the minor differences between cigarette smokers and non-smokers, are cigarettes harmless?

Smoking and Lung Cancer: Independence?

The **independence assumption** would imply that all three groups have the **same potential outcomes on average**

$$E[Y^1 \mid \text{Non-Smoker}] = E[Y^1 \mid \text{Cigarette}] = E[Y^1 \mid \text{Pipe}] = E[Y^1 \mid \text{Cigar}]$$
$$E[Y^0 \mid \text{Non-Smoker}] = E[Y^0 \mid \text{Cigarette}] = E[Y^0 \mid \text{Pipe}] = E[Y^0 \mid \text{Cigar}]$$

Suppose that the **independence assumption** holds

- This would/should also mean that observable characteristics X are similar between the groups
- I.e. the **covariates should be balanced** between groups

Are cigarette smokers similar to pipe and cigar smokers?

Let's ask Dall-E: show me a picture of a cigarette smoker and a cigar smoker

Age as a Confounder?

Smoking group	Canada	UK	US
Non-smokers	54.9	49.1	57.0
Cigarettes	50.5	49.8	53.2
Cigars/pipes	65.9	55.7	59.7

Clearly, **age affects what people smoke and also their death rates**

- Independence is violated: the **distribution of age** is different between the groups
- There may be other confounders, but let's focus on age for now

We have **covariate imbalance!**

Potential remedy: condition on age (**subclassification**)

Subclassification: Divide Age into Strata

	Death rates	# of Cigarette smokers	# of Pipe or cigar smokers
Age 20–40	20	65	10
Age 41–70	40	25	25
Age $\geq$ 71	60	10	65
Total		100	100

Subclassification: Divide Age into Strata

	Death rates	# of Cigarette smokers	# of Pipe or cigar smokers
Age 20–40	20	65	10
Age 41–70	40	25	25
Age $\geq$ 71	60	10	65
Total		100	100

The **death rate of cigarette smokers in the population** is:

$$20 \times \frac{65}{100} + 40 \times \frac{25}{100} + 60 \times \frac{10}{100} = 29$$

But: the **age distribution is (heavily) imbalanced** between the groups

Re-weighting: Age-Adjusted Death Rates

Let's **re-weight** the death rates of cigarette smokers by the **age distribution of pipe/cigar smokers**

	Death rates	# of Cigarette smokers	# of Pipe or cigar smokers
Age 20–40	20	65	10
Age 41–70	40	25	25
Age $\geq$ 71	60	10	65
Total		100	100

The **age-adjusted death rate of cigarette smokers** is:

$$20 \times \frac{10}{100} + 40 \times \frac{25}{100} + 60 \times \frac{65}{100} = 51$$

If **cigarette smokers** had the **same age distribution as pipe/cigar smokers**, their death rate would be 51

Age-Adjusted Death Rates

Cochran **computes age-adjusted death rates** (based on the population age distribution)

Smoking group	Canada	UK	US
Non-smokers	20.2	11.3	13.5
Cigarettes	29.5	14.8	21.2
Cigars/pipes	19.8	11.0	13.7

Here we **achieved balance on one covariate: age**

- The **age-adjusted death rates** are now more similar between the groups
- But there may be an **imbalance on other covariates** (SES, income, health, etc)

We need to **use a DAG** to identify **all confounders** and adjust for them

Identifying Assumptions

In presence of confounders X, we can **identify a causal effect under two assumptions**

1. **Conditional Independence**: $Y^0, Y^1 \perp D \mid X$
2. **Common Support**: $0 < P(D = 1 \mid X) < 1$ with probability one

Common support: for each stratum, we need some units that are treated and others that are control units

- We need **common support** to calculate the **weights for the adjustment**

Summary: Subclassification and Re-weighting

Treated and control units often differ in the **distribution of X (confounders)**

We can make **both groups** (somewhat) **comparable** by

1. dividing the sample into **strata based on X** (**subclassification**)
2. re-weighting the strata to **achieve balance on X** (**re-weighting**)

After re-weighting, both groups have the **same distribution of X by construction**

Causal Identification with Selection on Observables

Under **conditional independence and common support**, the following holds:

$$\begin{aligned} E\left[Y^1 - Y^0 \mid X\right] &= E\left[Y^1 - Y^0 \mid X, D = 1\right] \\ &= E\left[Y^1 \mid X, D = 1\right] - E\left[Y^0 \mid X, D = 0\right] \\ &= E\left[Y \mid X, D = 1\right] - E\left[Y \mid X, D = 0\right] \end{aligned}$$

The **estimator for the ATE** is as follows:

$$\widehat{\delta_{ATE}} = \int \Big(E\left[Y \mid X, D = 1\right] - E\left[Y \mid X, D = 0\right] \Big) d\Pr(X)$$

The Limits of Subclassification: The Curse of Dimensionality

In the example of **smoking and death rates**, we **adjusted for just one confounder**

- The hope was that, by slicing up age into three groups, achieve balance in treated and control groups
- We did achieve balance on age, but what about other confounders?
- Also, are three age groups enough or do we need more?

In practice, we have the **problem of a finite sample size**

- There are **limits to how many strata we can create**
- We cannot have an infinite number of groups defined by one variable (such as age)
- We cannot have an infinite number of variables to adjust for

This problem is known as the **curse of dimensionality**

The Limits of Subclassification: The Curse of Dimensionality

Let's say we have $k = 1, \ldots, K$ groups (for example defined by gender and age). We can calculate the ATT as

$$\widehat{\delta}_{ATT} = \sum_{k=1}^{K} \left(\overline{Y}^{1,k} - \overline{Y}^{0,k} \right) \times \left(\frac{N_T^k}{N_T} \right)$$

where $\overline{Y}^{1,k}$ and $\overline{Y}^{0,k}$ are the average outcomes in group k for treated and control units, and N_T^k is the number of treated units in group k.

In **large groups** (small K) we will easily find a **control unit for every treated unit**

As K increases and **groups get smaller**, we will have **more and more groups** that only contain **control or treated units but not both**

Possible Solution: Matching

Idea of matching:

- for each **treated unit**, find a **control unit** that is **similar on all confounders**
- **compare the outcomes** of **treated and control units**
- The **comparison** gives us an **estimate of the ATT**

Control units: **statistical twins** of treated units

It if also possible to have **multiple control units for each treated unit**

Statistical Twins?

Prince Charles

Male
Born in 1948
Raised in the UK
Married Twice
Lives in a castle
Wealthy and Famous

Ozzy Osbourne

Male
Born in 1948
Raised in the UK
Married Twice
Lives in a castle
Wealthy and Famous

Matching and the ATT: One Control Unit per Treated Unit

With one control unit for each treated unit, we **can calculate the ATT** as

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

- Y_i is the outcome for treated unit i
- $Y_{j(i)}$ is the outcome for the control unit $j(i)$

Matching and the ATT: Multiple Control Units per Treated Unit

Or if we find M matches for each treated unit, we can calculate the ATT as

$$\widehat{\delta}_{ATT} = \frac{1}{N_T} \sum_{D_i=1} \left(Y_i - \left[\frac{1}{M} \sum_{m=1}^{M} Y_{j_m(1)} \right] \right)$$

- $Y_{j_m(1)}$ is the outcome for the mth control unit matched to treated unit i

Matching and the ATE

We can also use **matching to estimate the ATE**. For this, we need to

- Find a similar control unit for each treated unit
- Find a similar treated unit for each control unit

The **estimator for the ATE** is as follows:

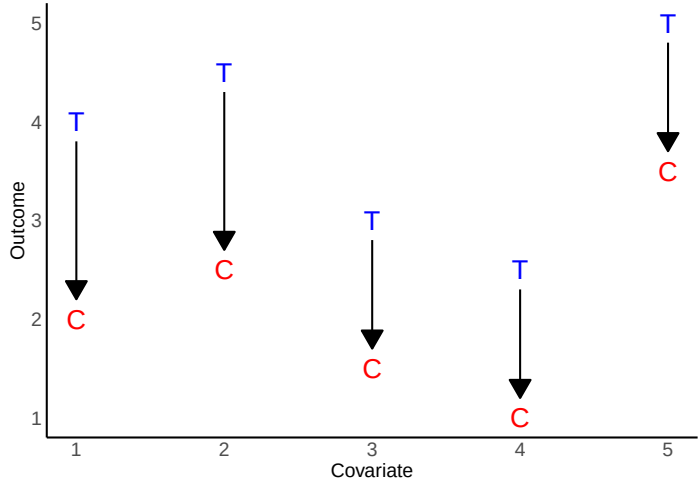
$$\widehat{\delta}_{ATE} = \frac{1}{N}\sum_{i=1}^{N}(2D_i - 1)\left[Y_i - \left(\frac{1}{M}\sum_{m=1}^{M} Y_{j_m(i)}\right)\right]$$

Exact Matching

Match each treated unit to a control unit that has **exactly the same covariate values**

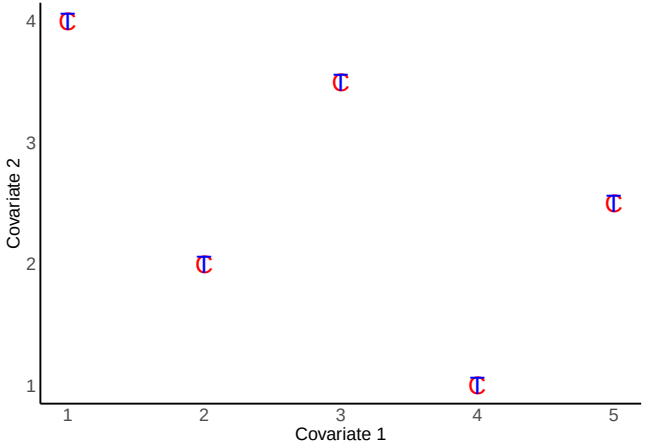
This is called **exact matching** and can be thought of as the **gold standard for matching**

Exact Matching with One Covariate



For each treated unit, we find a **control unit with the same covariate value**

Exact Matching with Two Covariates

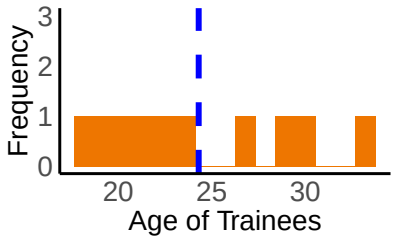


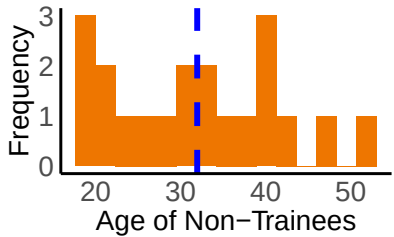
For each treated unit, we find a control unit with the **same values of covariates 1 and 2**

Example: Job Training Programme

Trainees			Non-Trainees		
Unit	Age	Earnings	Unit	Age	Earnings
1	18	9500	1	20	8500
2	29	12250	2	27	10075
3	24	11000	3	21	8725
4	27	11750	4	39	12775
5	33	13250	5	38	12550
6	22	10500	6	29	10525
7	19	9750	7	39	12775
8	20	10000	8	33	11425
9	21	10250	9	24	9400
10	30	12500	10	30	10750
			11	33	11425
			12	36	12100
			13	22	8950
			14	18	8050
			15	43	13675
			16	39	12775
			17	19	8275
			18	30	9000
			19	51	15475
			20	48	14800
Mean	24.3	$11,075	Mean	31.95	$11,101.25

Age Distribution of Trainees vs. Non-Trainees



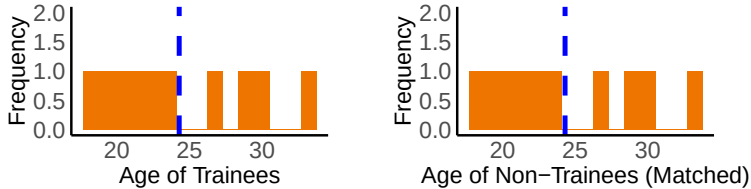


Clearly, the age distribution of trainees and non-trainees is different (mean 24.3 vs. 31.95)

Creating an (exactly) Matched Sample

Trainees			Non-Trainees			Matched Sample		
Unit	Age	Earnings	Unit	Age	Earnings	Unit	Age	Earnings
1	18	9500	1	20	8500	14	18	8050
2	29	12250	2	27	10075	6	29	10525
3	24	11000	3	21	8725	9	24	9400
4	27	11750	4	39	12775	8	27	10075
5	33	13250	5	38	12550	11	33	11425
6	22	10500	6	29	10525	13	22	8950
7	19	9750	7	39	12775	17	19	8275
8	20	10000	8	33	11425	1	20	8500
9	21	10250	9	24	9400	3	21	8725
10	30	12500	10	30	10750	10,18	30	9875
			11	33	11425			
			12	36	12100			
			13	22	8950			
			14	18	8050			
			15	43	13675			
			16	39	12775			
			17	19	8275			
			18	30	9000			
			19	51	15475			
			20	48	14800			
Mean	24.3	$11,075	Mean	31.95	$11,101.25	Mean	24.3	$9,380

Treated Sample vs. Matched Control Sample



With **exact matching**, the age distribution of **treated and matched control units are the same**

If age is the only confounder, we can **estimate the ATT** as

$$\text{ATT} = \frac{1}{N}\sum_{i=1}^{N}(Y_i - Y_{i'}) = 11,075 - 9,380 = 1,695$$

So the **estimated causal effect** of the **training programme** is 1,695 dollars

Approximate Matching

In most cases, we **cannot find a perfect match** for each treated unit

- Many **variables are continuous**
- We have many **covariates**
- . . . and **finite samples**

Approximate matching allows us to **match similar units**

Approximate Matching

There are two **main methods for approximate matching:**

1. **Distance Matching** $\rightarrow$ minimise distance in X
2. **Propensity Score Matching** $\rightarrow$ match on likelihood of being treated

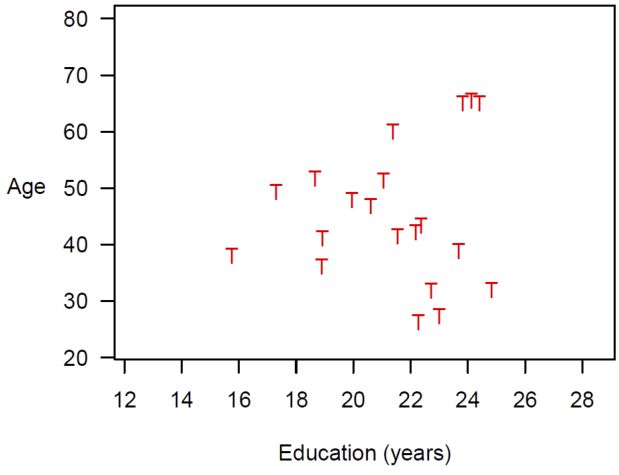
A third type of matching is **coarsened exact matching** (CEM)

It is also possible to **combine matching methods**

- Example: match exactly on some characteristics and approximately on others

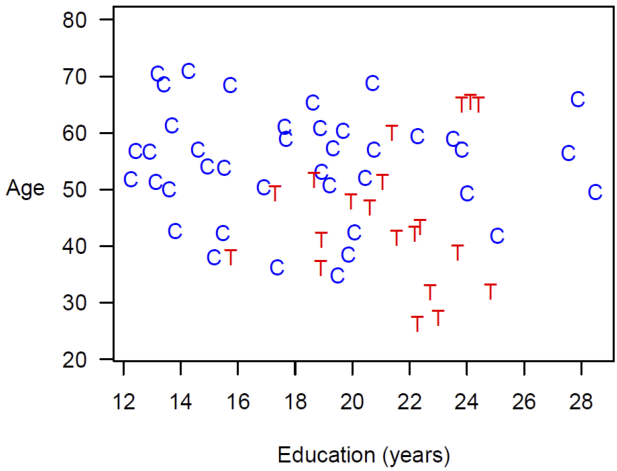
Mahalabnobis Distance Matching

Starting point: **treated and with covariates age and age at which they left education**



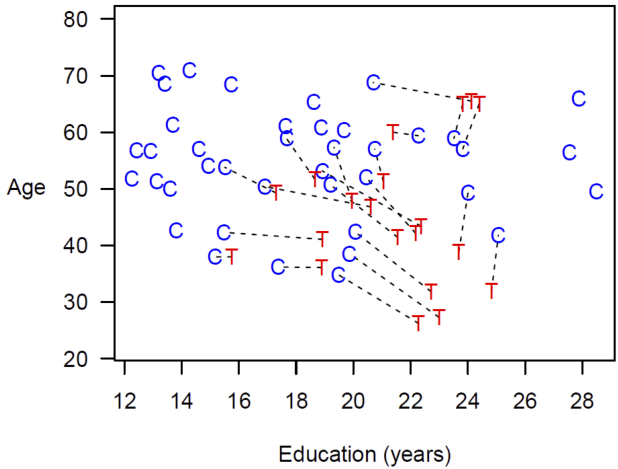
Mahalabnobis Distance Matching

Treated and control units are different w.r.t. education



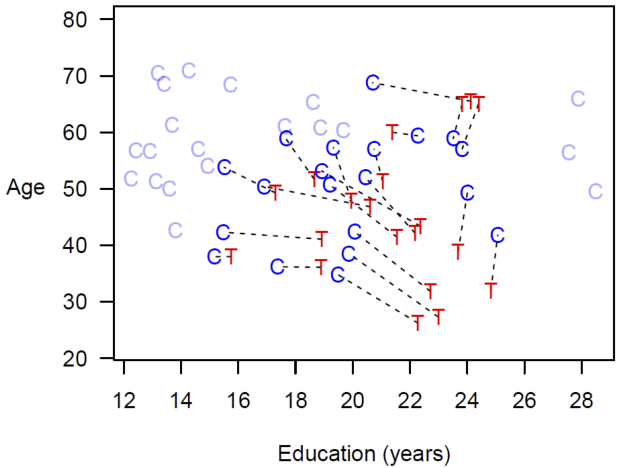
Mahalabnobis Distance Matching

For each treated unit, we **find the "closest" control unit** in terms of X



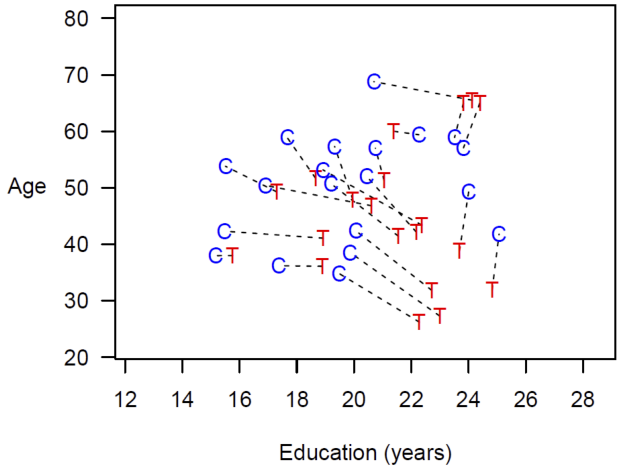
Mahalabnobis Distance Matching

Drop control units that are **not close enough** to any treated unit



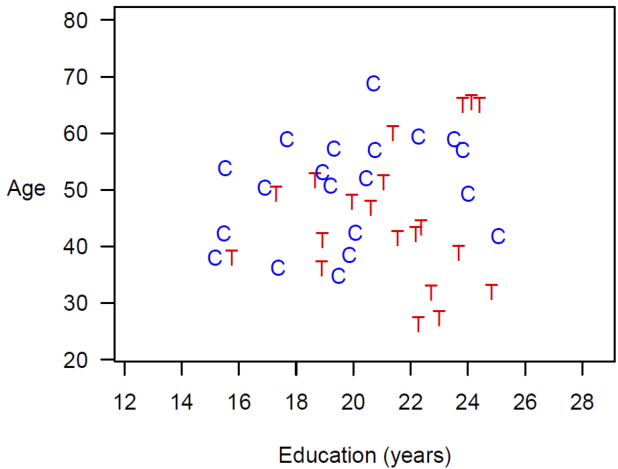
Mahalabnobis Distance Matching

Drop control units that are **not close enough** to any treated unit



Mahalabnobis Distance Matching

Our estimation sample:



Mahalabnobis Distance Matching with one Covariate

With **one covariate**, the distance is the Euclidean Distance

$$||X_i - X_j|| = \sqrt{(X_i - X_j)'(X_i - X_j)}$$
$$= \sqrt{\sum_{n=1}^{k}(X_{ni} - X_{nj})^2}$$

For each treated unit, we find the control unit with the **smallest distance** $||X_i - X_j||$

Mahalabnobis Distance Matching with multiple Covariates

With **multiple covariates** $1, \ldots, k$, we take into account the variance-covariance matrix $\widehat{\Sigma}_X$ of the covariates

$$||X_i - X_j|| = \sqrt{(X_i - X_j)' \widehat{\Sigma}_X^{-1} (X_i - X_j)}$$

As before, for each treated unit, we find the control unit with the **smallest distance** $||X_i - X_j||$

Purpose of weighting with $\widehat{\Sigma}_X^{-1}$:

- Covariates become **scale-invariant**
- All distances are **measured in terms of standard deviations**

Mahalanobis Distance Matching: Steps Involved

1. **Preprocess (Matching)**

- **Calculate the Distance** $||X_i - X_j|| = \sqrt{(X_i - X_j)' \widehat{\Sigma}_X^{-1} (X_i - X_j)}$
- **Match** each treated unit to the **nearest control unit**
- Prune control units if unused
- Prune matches if Distance>caliper (i.e. if they exceed a certain distance)

2. **Estimation**: calculate difference in means or run a regression

Other Distance Matching Methods

Nearest-neighbour Matching (NNM)

- Match with the nearest neighbour or the k **nearest neighbours** in terms of X
- Take the average of these neighbours as the counterfactual

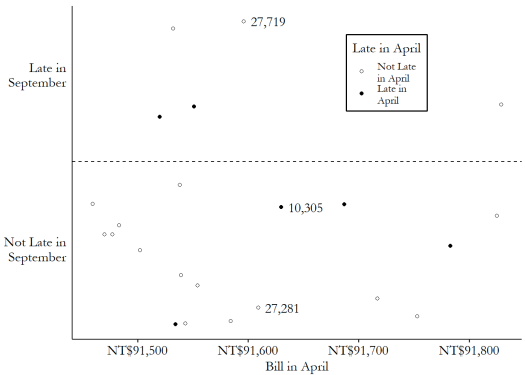
Radius Matching

- Match with **all control units** within a **certain radius of the treated unit**

Kernel Matching

- Match with **all control units** within a certain bandwidth of the treated unit
- **Weight the control units by their distance** to the treated unit

Distance Matching: Example



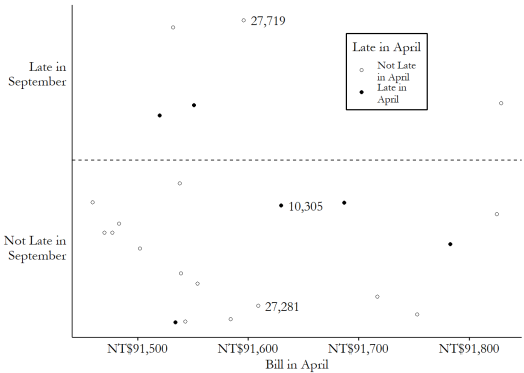
Example here: **credit card data**

Matching variables:

- Whether the person was billed in April or September
- The amount they were billed

The numbers (e.g. 10,305) just indicate the case number

Distance matching



Finding a match for 10,305

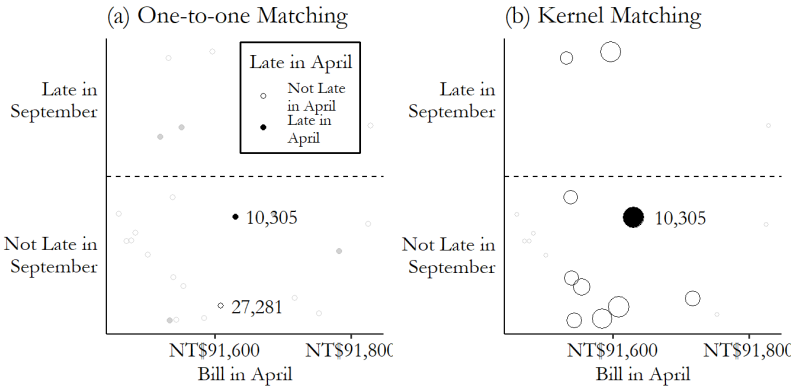
Match exactly on the month of billing

- Only look for a match among units billed in September

Find the **closest match in terms of the sum of the bill**

- Observation 27,281 fits the bill...

Kernel Matching



Left: only one control observation used as a match

Right: all control observations are used, but with different weights

- **weights decay with distance**

Kernel Matching

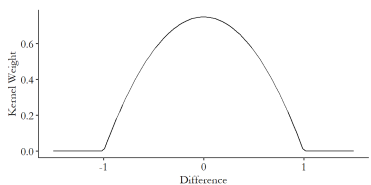
We want to create a weighted average by applying a kernel function

$$\bar{Y} = \frac{\sum_{i=1}^{n} w_i Y_i}{\sum_{i=1}^{n} w_i} = \frac{\sum_{i=1}^{n} K(X_i) Y_i}{\sum_{i=1}^{n} K(X_i)}$$

There are many Kernel functions; they are typically concave and assign the highest weight to the smallest distance.

Example: Epanechnikov kernel

$$K(X) = \frac{3}{4}(1 - X^2)$$



K(X) is only defined between -1 and 1

Coarsened Exact Matching

Idea of CEM:

- Coarsen X (for example different age groups)
- Perform exact matching based on coarsened data

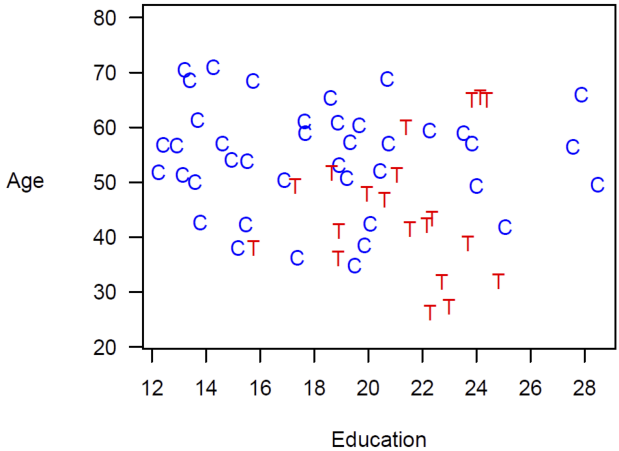
Advantage: **easy and fast**

Disadvantages:

- researcher **degrees of freedom** (categories are chosen by the researcher)
- curse of dimensionality (few categories: many but imprecise matches; many categories: few but more precise matches)

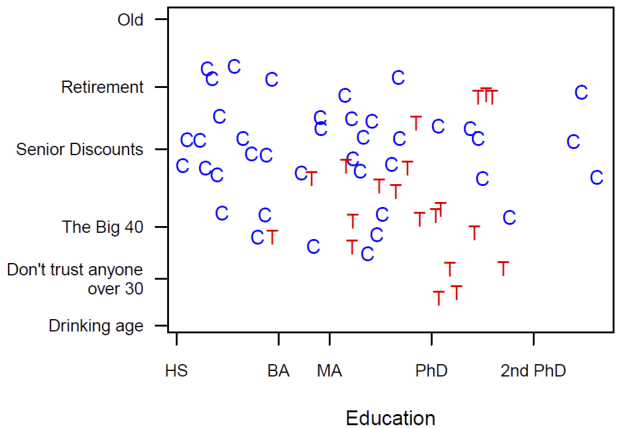
Coarsened Exact Matching

Starting point: same as before



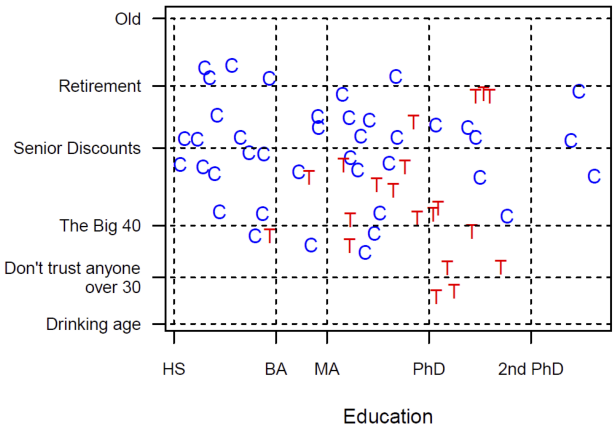
Coarsened exact matching

Coarsen: **divide variables into categories**



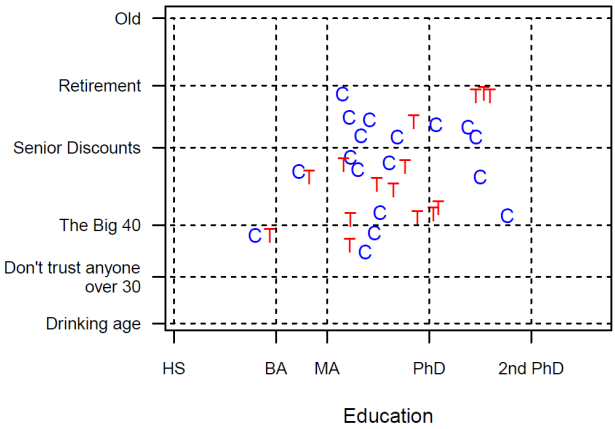
Coarsened exact matching

Now see **which cells contain treated and control units**



Coarsened exact matching

We **find matches within cells**



Coarsened exact matching

We **find matches within cells**

We need to **take a stand** regarding **matching within the cells**

- nearest neighbour or k nearest neighbours
- with or without replacement
- kernel distance function (usually not necessary)

References

Broockman, David E. 2013. Black Politicians Are More Intrinsically Motivated to Advance Blacks' Interests: A Field Experiment Manipulating Political Incentives. *American Journal of Political Science*, **57**(3), 521–536.

Cochran, W. G. 1968. The Effectiveness of Adjustment by Subclassification in Removing Bias in Observational Studies. *Biometrics*, **24**(2).

APPENDIX

Group Work

Re-weighting

Example: Using Broockman (2013) for R examples

Research question: are black politicians more likely to help black citizens even if the incentives are low?

Methodology: audit study; sent emails to U.S. state legislators; asking them to help them sign up for unemployment benefits

Experimental variation:

- Sender with black vs. white name
- Sender lives in same district as legislator or far away

Matching: white and black legislators with similar characteristics

Broockman (2013) data preparation

We use the excellent `Matching` package in R. A great alternative is `MatchIt`

```
library(Matching)
library(causaldata)
library(tidyverse)

br <- causaldata::black_politicians

# Outcome
Y <- br %>%
    pull(responded)
# Treatment
D <- br %>%
    pull(leg_black)
# Matching variables
# Note select() is also in the Matching package, so we specify dplyr
X <- br %>%
    dplyr::select(medianhhincom, blackpercent, leg_democrat) %>%
    as.matrix()
```

Mahalanobis distance matching in R

```r
# Set weight=2 for Mahalanobis distance
M <- Match(Y, D, X, Weight = 2, caliper = 1)

# See treatment effect estimate
summary(M)
```

```
##
## Estimate...  -0.0073462
## AI SE......  0.072683
## T-stat.....  -0.10107
## p.val......  0.91949
##
## Original number of observations..............  5593
## Original number of treated obs...............  364
## Matched number of observations...............  363
## Matched number of observations  (unweighted).  405
##
## Caliper (SDs)........................................   1 1 1
## Number of obs dropped by 'exact' or 'caliper'  1
```

Mahalanobis distance matching in R

Previous slide: the estimate -0.007346 means that black legislators were 0.7 percentage points less likely to respond to emails

This effect is not statistically significant

Mahalanobis distance matching in R

```
# Get matched data for use elsewhere. Note that this approach will
# duplicate each observation for each time it was matched
matched_treated <- tibble(id = M$index.treated,
                          weight = M$weights)
matched_control <- tibble(id = M$index.control,
                          weight = M$weights)
matched_sets <- bind_rows(matched_treated,
                          matched_control)
# Simplify to one row per observation
matched_sets <- matched_sets %>%
                    group_by(id) %>%
                    summarize(weight = sum(weight))
# And bring back to data
matched_br <- br %>%
    mutate(id = row_number()) %>%
    left_join(matched_sets, by = 'id')
```

Mahalanobis distance matching in R

```
# OLS estimation based on matched sample
lm(responded~leg_black, data = matched_br, weights = weight)
```

```
##
## Call:
## lm(formula = responded ~ leg_black, data = matched_br, weights = weight)
##
## Coefficients:
## (Intercept)    leg_black
##    0.398531    -0.007346
```

We can see that the estimate is the same as with matching

Coarsened Exact Matching in R

Broockman performs CEM to make black and white legislators more comparable

We use the `cem` package here. Alternatively we could use the `method_cem` command of the `MatchIt` package.

```
library(cem); library(tidyverse)
br <- causaldata::black_politicians
```

Coarsened Exact Matching in R

```r
# Limit to just the relevant variables and omit missings
brcem <- br %>%
    dplyr::select(responded, leg_black, medianhhincom,
    blackpercent, leg_democrat) %>%
    na.omit() %>%
    as.data.frame() # Must be a data.frame, not a tibble

# Two ways to create breaks. Use quantiles to create quantile cuts or manually for evenly
# although you MUST do it yourself for binary variables). Be sure
# to include the "edges" (max and min values).
inc_bins <- quantile(brcem$medianhhincom, (0:6)/6)

create_even_breaks <- function(x, n) {
    minx <- min(x)
    maxx <- max(x)

    return(minx + ((0:n)/n)*(maxx-minx))
}
```

Coarsened Exact Matching in R

```
bp_bins <- create_even_breaks(brcem$blackpercent, 6)

# For binary, we specifically need two even bins
ld_bins <- create_even_breaks(brcem$leg_democrat,2)

# Make a list of bins
allbreaks <- list('medianhhincom' = inc_bins,
                  'blackpercent' = bp_bins,
                  'leg_democrat' = ld_bins)
```

Coarsened Exact Matching in R

```
# Match, being sure not to match on the outcome
# Note the baseline.group is the *treated* group
c <- cem(treatment = 'leg_black', data = brcem,
         baseline.group =  '1',
         drop = 'responded',
         cutpoints = allbreaks,
         keep.all = TRUE)
```

```
##
## Using 'leg_black'='1' as baseline group
```

```
# Get weights for other purposes
brcem <- brcem %>%
    mutate(cem_weight = c$w)
```

Coarsened Exact Matching in R

```r
# OLS estimation with weighted dataset
lm(responded~leg_black, data = brcem, weights = cem_weight)
```

```
##
## Call:
## lm(formula = responded ~ leg_black, data = brcem, weights = cem_weight)
##
## Coefficients:
## (Intercept)    leg_black
##     0.34680      0.02302
```

Coarsened Exact Matching in R

```
# Use the inbuilt ATT estimation command from cem
att(c, responded ~ leg_black, data = brcem)
```

```
##
##              G0  G1
## All        5229 364
## Matched    4491 338
## Unmatched   738  26
##
## Linear regression model on CEM matched data:
##
## SATT point estimate: 0.023020 (p.value=0.391783)
## 95% conf. interval: [-0.029659, 0.075699]
```

PSM in R

To perform PSM, we can use the `MatchIt` package. Here we estimate the propensity score for the LaLonde data

```
library("MatchIt")
library('marginaleffects')
data("lalonde")

# 1:1 NN PS matching w/o replacement
m.out1 <- matchit(treat ~ age + educ + race + married +
                    nodegree + re74 + re75, data = lalonde,
                  method = "nearest", distance = "glm")
```

PSM in R

Checking balance after nearest neighbor matching

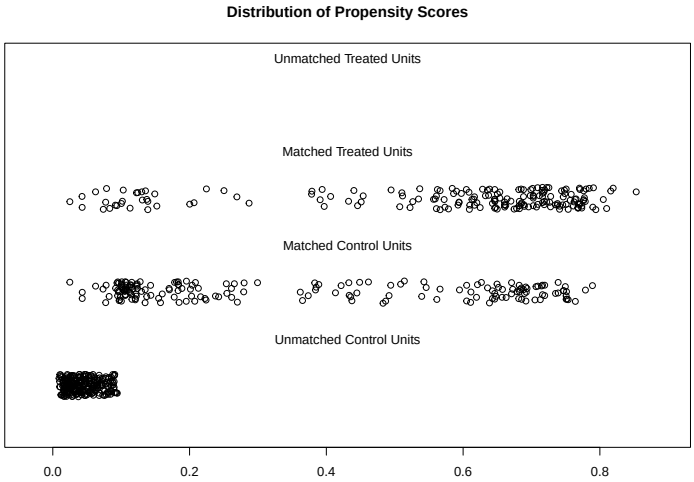
```
summary(m.out1, un = FALSE)
```

```
##
## Call:
## matchit(formula = treat ~ age + educ + race + married + nodegree +
##     re74 + re75, data = lalonde, method = "nearest", distance = "glm")
##
## Summary of Balance for Matched Data:
##            Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## distance          0.5774        0.3629          0.9739     0.7566    0.1321
## age              25.8162       25.3027          0.0718     0.4568    0.0847
## educ             10.3459       10.6054         -0.1290     0.5721    0.0239
## raceblack         0.8432        0.4703          1.0259          .    0.3730
## racehispan        0.0595        0.2162         -0.6629          .    0.1568
## racewhite         0.0973        0.3135         -0.7296          .    0.2162
## married           0.1892        0.2108         -0.0552          .    0.0216
## nodegree          0.7081        0.6378          0.1546          .    0.0703
## re74           2095.5737     2342.1076         -0.0505     1.3289    0.0469
## re75           1532.0553     1614.7451         -0.0257     1.4956    0.0452
##            eCDF Max Std. Pair Dist.
## distance     0.4216          0.9740
## age          0.2541          1.3938
```

PSM in R

We can also plot the distribution of propensity scores

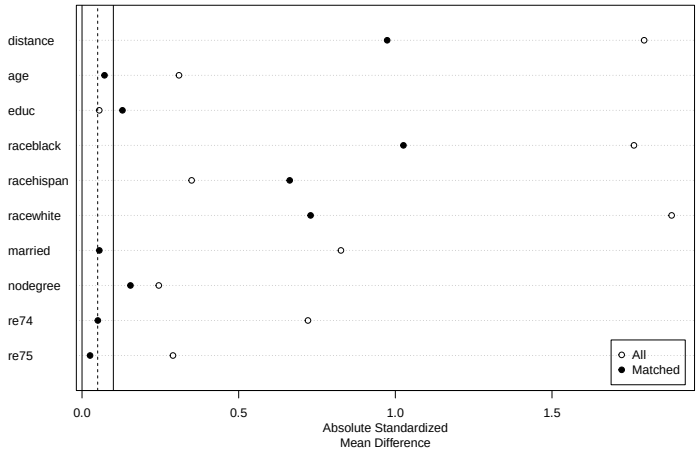
```
plot(m.out1, type = "jitter", interactive = FALSE)
```



PSM in R

Or how about this one...

```
plot(summary(m.out1))
```



PSM in R

```r
# Generate matched dataset
m.data <- match.data(m.out1)
# Run a regression on the matched dataset
fit <- lm(re78 ~ treat + age + educ + race + married + nodegree +
            re74 + re75, data = m.data, weights = weights)
summary(fit)
```

```
##
## Call:
## lm(formula = re78 ~ treat + age + educ + race + married + nodegree +
##     re74 + re75, data = m.data, weights = weights)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -8891  -5063  -1703   3422  53495
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.582e+03  3.296e+03  -0.783  0.43394
## treat        1.345e+03  7.898e+02   1.703  0.08945 .
## age          7.804e+00  4.292e+01   0.182  0.85581
```

PSM in R

```r
# Can also compute the ATT based on the interactions of the treatment
fit <- lm(re78 ~ treat * (age + educ + race + married + nodegree +
            re74 + re75), data = m.data, weights = weights)

avg_comparisons(fit,
                variables = "treat",
                vcov = ~subclass,
                newdata = subset(m.data, treat == 1),
                wts = "weights")
```

```
##
##   Term Contrast Estimate Std. Error    z Pr(>|z|)   S 2.5 % 97.5 %
##  treat    1 - 0     1121        752 1.49    0.136 2.9  -354   2596
##
## Columns: term, contrast, estimate, std.error, statistic, p.value, s.value, conf.low, co
```

PSM in R

Another option based on the `Matching` package; PSM done directly here

```
library("Matching")
attach (lalonde)
D <- treat
Y <- re78 # define outcome
X <- cbind( age , educ , nodegree , married , re74 , re75)
ps<- glm(D ~ X, family=binomial)$fitted
psmatching <- Match(Y=Y, Tr=D, X=ps , BiasAdjust = TRUE)
```

PSM in R

Another option based on the `Matching` package; PSM done directly here

```
summary(psmatching)
```

```
##
## Estimate...  597.88
## AI SE......  913.53
## T-stat.....  0.65447
## p.val......  0.51281
##
## Original number of observations..............  614
## Original number of treated obs...............  185
## Matched number of observations...............  185
## Matched number of observations  (unweighted).  289
```

benjamin.elsner@ucd.ie

www.benjaminelsner.com

Sign up for office hours

YouTube Channel

@ben_elsner

LinkedIn

Contact

Prof. Benjamin Elsner
University College Dublin
School of Economics
Newman Building, Office G206
benjamin.elsner@ucd.ie

Office hours: book on Calendly