Building a Robot Judge: Data Science for Decision-Making

6. Machine Learning and Causal Inference

Learning Objectives

1. Implement and evaluate machine learning pipelines.
2. **Implement and evaluate causal inference designs.**
3. Understand how (not) to use data science tools (ML and CI) to support expert decision-making.

Zoom Private Chat Activity: Legal Briefs Dataset

- Let's say we have a dataset of legal briefs with text and metadata, everything you would expect from a case, including information on the actors (litigants, attorneys, judges) and the associated outcomes (e.g. who wins).
- Questions:
  1. A **machine learning task or question** that could be addressed with this dataset.
  2. A **causal inference task or question** that could be addressed with this dataset.
- Post your answers at this padlet: `https://padlet.com/eash44/eybycfi1130owbn0`
  - Read your classmates' answers – and "like" them liberally.

Outline

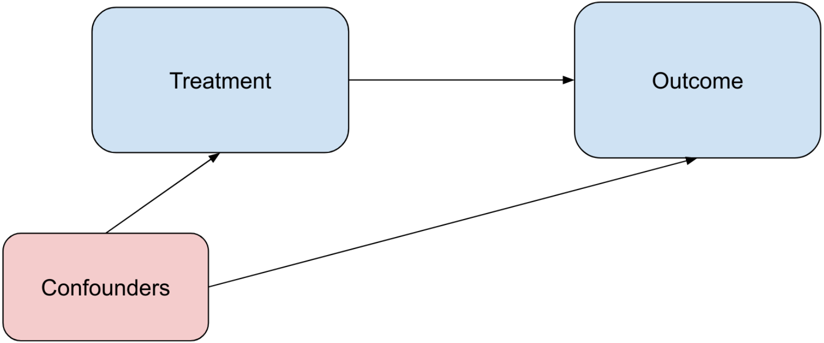
Matching

Double Machine Learning to Adjust for Confounders

Synthetic Control Method

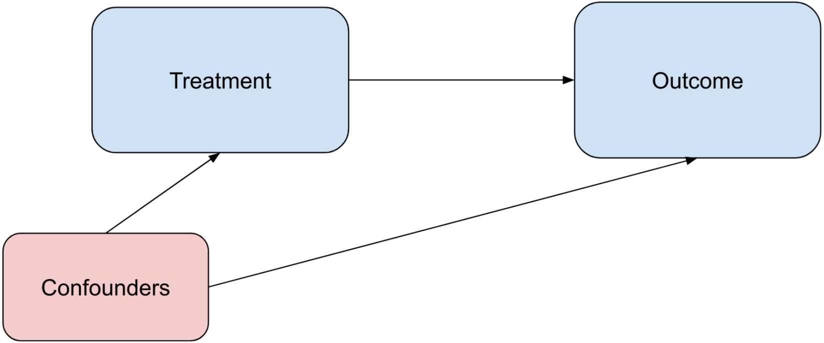
Machine Learning for Heterogenous Treatment Effects

Observed Confounders



- Recap from Week 2:
    - If the treated group and comparison group differ only by a set of observable characteristics, we can “control” or “adjust” for these variables to obtain causal estimates.

Observed Confounders



- Recap from Week 2:
    - If the treated group and comparison group differ only by a set of observable characteristics, we can "control" or "adjust" for these variables to obtain causal estimates.
- **Matching** is an alternative causal inference approach:
    - for each "treated" unit, find a matched "control" observation to compare to
    - (as opposed to including all observations in the dataset and adding covariates)

Propensity Score Matching (PSM)

$$Y = \alpha + \rho D + X'\beta + \epsilon$$

Propensity Score Matching (PSM)

$$Y = \alpha + \rho D + X'\beta + \epsilon$$

In the case of a binary treatment $D \in \{0,1\}$, the following is equivalent to perfectly adjusting for all observed confounders:

Propensity Score Matching (PSM)

$$Y = \alpha + \rho D + X'\beta + \epsilon$$

In the case of a binary treatment $D \in \{0,1\}$, the following is equivalent to perfectly adjusting for all observed confounders:

- Predict a cross-validated "propensity score" $\hat{D}(X) = \Pr(D = 1|X)$, the probability of treatment.
  - e.g., logistic regression, xgboost classifier.

Propensity Score Matching (PSM)

$$Y = \alpha + \rho D + X'\beta + \epsilon$$

In the case of a binary treatment $D \in \{0,1\}$, the following is equivalent to perfectly adjusting for all observed confounders:

- Predict a cross-validated "propensity score" $\hat{D}(X) = \Pr(D = 1|X)$, the probability of treatment.
  - e.g., logistic regression, xgboost classifier.
- Then adjust for $\hat{D}(X)$ in the regression.
  - in practice, include fixed effects for small bins of $\hat{D}(X)$
  - then all individuals are compared to other individuals with a similar propensity score.

Note: while PSM (adjusting for $\hat{D}(X)$) is sufficient to get unbiased $\hat{\rho}$ if X contains all confounders, including X as well in the regression might still shrink standard errors.

Doubly Robust Estimation

$$Y = \alpha + \rho D + X'\beta + \epsilon$$

Doubly Robust Estimation

$$Y = \alpha + \rho D + X'\beta + \epsilon$$

The following is also sufficient to identify a causal effect:

Doubly Robust Estimation

$$Y = \alpha + \rho D + X'\beta + \epsilon$$

The following is also sufficient to identify a causal effect:

- Learn an outcome regression $\hat{Y}(X)$, a cross-validated prediction of the outcome based on the observed confounders.
  - e.g., elastic net, xgboost regressor.
- If the prediction model $\hat{Y}(X)$ correctly learns the influence of all confounders on the outcome, then the regression

  $$Y = \alpha + \rho D + \gamma \hat{Y}(X) + \epsilon$$

  provides causal estimates.

Doubly Robust Estimation

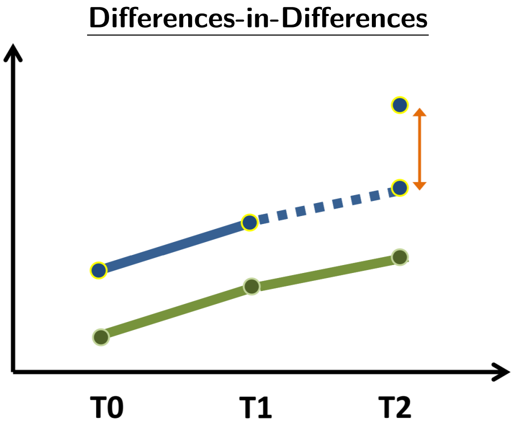
$$Y = \alpha + \rho D + X'\beta + \epsilon$$

The following is also sufficient to identify a causal effect:

- Learn an outcome regression $\hat{Y}(X)$, a cross-validated prediction of the outcome based on the observed confounders.
  - e.g., elastic net, xgboost regressor.
- If the prediction model $\hat{Y}(X)$ correctly learns the influence of all confounders on the outcome, then the regression

  $$Y = \alpha + \rho D + \gamma \hat{Y}(X) + \epsilon$$

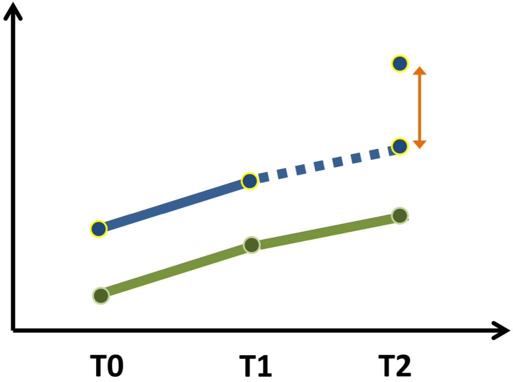
  provides causal estimates.
- Further, if you do **both** propensity score matching **and** adjustment for the predicted outcome, then only one of the models ($\hat{D}(X)$ or $\hat{Y}(X)$) has to be correct for $\hat{\rho}$ to be causally identified.



- use all untreated units as comparison groups for the treated units.
- Two-way fixed-effects regression:

$$Y_{jt} = \alpha_j + \alpha_t + \gamma D_{jt} + \varepsilon_{jt}$$

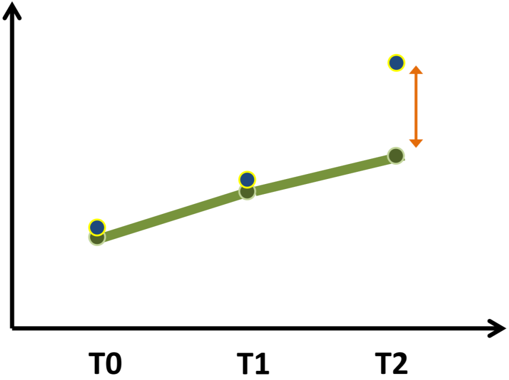
Differences-in-Differences



- use all untreated units as comparison groups for the treated units.
- Two-way fixed-effects regression:

$$Y_{jt} = \alpha_j + \alpha_t + \gamma \mathrm{D}_{jt} + \varepsilon_{jt}$$

Matched Differences-in-Differences



- for each treated unit j, search over comparison group and find unit j' with most similar pre-trends., then estimate

$$Y_{jt} - Y_{j't} = \alpha + \gamma \mathrm{D}_{jt} + \varepsilon_{jt}$$

Can use any distance metric / can match on covariates / can match on probability of treatment at the same time as j.

Outline

Matching

Double Machine Learning to Adjust for Confounders

Synthetic Control Method

Machine Learning for Heterogenous Treatment Effects

What if we have more control covariates than observations?

- the OLS estimator requires that the predictor matrix be full rank.
  - in particular, collinear predictors will break OLS
- with clustered standard errors, have to have more clusters than predictors.

What if we have more control covariates than observations?

- the OLS estimator requires that the predictor matrix be full rank.
    - in particular, collinear predictors will break OLS
- with clustered standard errors, have to have more clusters than predictors.
- Machine learning can help.

Selecting Controls with Double Lasso (1)

Selecting Controls with Double Lasso (1)

- Consider outcome variable Y and treatment variable D. We want to estimate β from
$$Y = \beta D + g(X) + \epsilon$$
  $g(X)$ is an **unknown** "nuisance function" summarizing the effect of all the confounders.

Selecting Controls with Double Lasso (1)

- Consider outcome variable Y and treatment variable D. We want to estimate β from

  $$Y = \beta D + g(X) + \epsilon$$

  $g(X)$ is an **unknown** “nuisance function” summarizing the effect of all the confounders.

  - X is a high-dimensional set of predictors – some are confounders, most are not.

Selecting Controls with Double Lasso (1)

- Consider outcome variable Y and treatment variable D. We want to estimate β from

  $$Y = \beta D + g(X) + \epsilon$$

  $g(X)$ is an **unknown** “nuisance function” summarizing the effect of all the confounders.
  - X is a high-dimensional set of predictors – some are confounders, most are not.
  - we will use **lasso** to select which predictors to include in our OLS regression.

Selecting Controls with Double Lasso (2)

- Data prep:
  - drop from X any potential colliders.
  - can add interactions and transformations, e.g. $x_7 x_9$, x_7^2.
  - standardize each variable in X to variance one

Selecting Controls with Double Lasso (2)

- Data prep:
  - drop from X any potential colliders.
  - can add interactions and transformations, e.g. $x_7 x_9$, x_7^2.
  - standardize each variable in X to variance one
- Train two lasso models, $Y \sim \text{Lasso}(X)$ and $D \sim \text{Lasso}(X)$:
  1. use CV grid search across the whole dataset to select best penalties λ_Y and λ_D.
  2. Run both lasso models with whole dataset, get subsets of non-zero predictors, X_Y and X_D

Selecting Controls with Double Lasso (2)

- Data prep:
  - drop from X any potential colliders.
  - can add interactions and transformations, e.g. $x_7 x_9$, x_7^2.
  - standardize each variable in X to variance one
- Train two lasso models, $Y \sim \text{Lasso}(X)$ and $D \sim \text{Lasso}(X)$:
  1. use CV grid search across the whole dataset to select best penalties λ_Y and λ_D.
  2. Run both lasso models with whole dataset, get subsets of non-zero predictors, X_Y and X_D
- Regress

$$Y = \beta D + X'_{YD}\gamma + \epsilon$$

where $X_{YD} = X_Y \cup X_D$ is the union of the lasso-selected covariates.

  - this is an optimal procedure if all confounders are contained in $X_Y \cup X_D$ (Belloni et al 2014).

Zoom Private Chat to Claudia: Which line(s) have a problem?

```
# python
param_grid = {'alpha':  [.01, .1, 1, 10]}
lasso = Lasso()
grid = GridSearchCV(lasso, param_grid)

01 grid.fit(X, Y)
02 alpha_Y = grid.best_params_['alpha']
03 lasso_Y = Lasso(alpha=alpha_Y)
04 lasso_Y.fit(X,Y)
05 selected_Y = lasso_Y.coef_ != 0

06 grid.fit(X, D)
07 alpha_D = grid.best_params_['alpha']
08 lasso_D = Lasso(alpha=alpha_D)
09 lasso_D.fit(X,D)
10 selected_D = lasso_D.coef_ != 0


11 X_YD = X[:,(selected_Y and selected_D)]

import statsmodels.api as sm
12 ols = sm.OLS(Y, np.hstack([D,X_YD]).fit()
```

What if $g(\cdot)$ is not linear?

- Lasso assumes that $g(X)$ is linear in X.
    - we somewhat relaxed that assumption by adding interactions and quadratic transformations.
    - but how do we know what interactions/transformations to add?

What if $g(\cdot)$ is not linear?

- Lasso assumes that $g(X)$ is linear in X.
  - we somewhat relaxed that assumption by adding interactions and quadratic transformations.
  - but how do we know what interactions/transformations to add?
- Can use a non-linear model, e.g. xgboost, to approximate and adjust for $g(X)$.
  $\rightarrow$ Double Machine Learning / Doubly Robust Estimation (Chernozhukov et al 2018)

Double ML: Setup

$$Y = \beta D + g(X) + \epsilon$$

- low-dimensional treatment D, high-dimensional set of (observed) confounders X.
  - OLS regression without adjusting for confounders will be biased for $\hat{\beta}$
  - can we just include them in the regression as linear covariates?
    - will not adjust correctly due to potential non-linearities.
    - will probably fail to converge due to high dimensionality / collinearity / overfitting

Double ML method

1. Learn Y given X, $\hat{Y}(X)$, using any ML method
2. Learn D given X, $\hat{D}(X)$, using any ML method

Double ML method

1. Learn Y given X, $\hat{Y}(X)$, using any ML method
2. Learn D given X, $\hat{D}(X)$, using any ML method
3. Form residuals $\tilde{Y} = Y - \hat{Y}(X)$ and $\tilde{D} = D - \hat{D}(X)$

Double ML method

1. Learn Y given X, $\hat{Y}(X)$, using any ML method
2. Learn D given X, $\hat{D}(X)$, using any ML method
3. Form residuals $\tilde{Y} = Y - \hat{Y}(X)$ and $\tilde{D} = D - \hat{D}(X)$
4. Regress $\tilde{Y}$ on $\tilde{D}$ to learn $\hat{\beta}$.

Double ML method

1. Learn Y given X, $\hat{Y}(X)$, using any ML method
2. Learn D given X, $\hat{D}(X)$, using any ML method
3. Form residuals $\tilde{Y} = Y - \hat{Y}(X)$ and $\tilde{D} = D - \hat{D}(X)$
4. Regress $\tilde{Y}$ on $\tilde{D}$ to learn $\hat{\beta}$.

Cross-Fitting: Split into samples A and B, 50% of data each, to prevent overfitting:

- Fit (1) and (2) on sample A, then predict (3) and regress (4) on sample B, to estimate $\hat{\beta}_A$
- vice versa: fit (1)/(2) on sample B, and predict/regress (3)/(4) on sample A, to learn a second estimate for $\hat{\beta}_B$.
- average them to get a more efficient estimator:$\hat{\beta} = \frac{1}{2}(\hat{\beta}_A + \hat{\beta}_B)$.

- Donohue and Levitt (2001) estimate

  $$y_{st} = \alpha_s + \alpha_t + \rho D_{st} + \epsilon_{st}$$

  where y_{st} is the crime rate in state s at time t and D_{st} is the historical abortion rate in s (16 years before t).
- Belloni et al (2014): learn functions $\hat{y}(X)$ and $\hat{D}(X)$ with X containing 284 variables, including interactions/polynomials/etc.
  - with lasso, include the union of selected X
  - with double ML, take cross-validated residuals

- Donohue and Levitt (2001) estimate

$$y_{st} = \alpha_s + \alpha_t + \rho D_{st} + \epsilon_{st}$$

where y_{st} is the crime rate in state s at time t and D_{st} is the historical abortion rate in s (16 years before t).

- Belloni et al (2014): learn functions $\hat{y}(X)$ and $\hat{D}(X)$ with X containing 284 variables, including interactions/polynomials/etc.
  - with lasso, include the union of selected X
  - with double ML, take cross-validated residuals

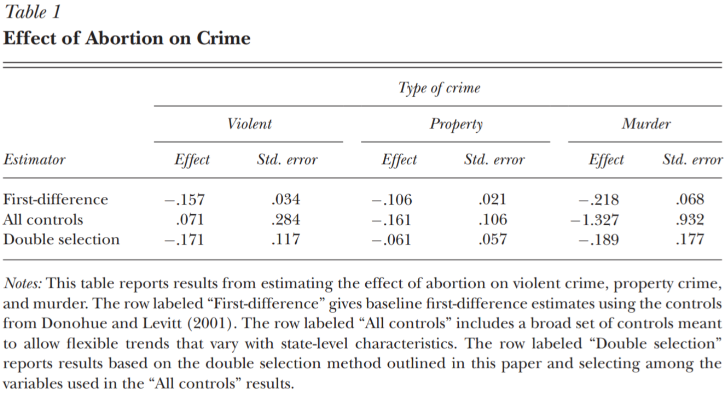


Table 1

Effect of Abortion on Crime

	Type of crime					
	Violent		*Property*		*Murder*	
Estimator	*Effect*	*Std. error*	*Effect*	*Std. error*	*Effect*	*Std. error*
First-difference	−.157	.034	−.106	.021	−.218	.068
All controls	.071	.284	−.161	.106	−1.327	.932
Double selection	−.171	.117	−.061	.057	−.189	.177

Notes: This table reports results from estimating the effect of abortion on violent crime, property crime, and murder. The row labeled "First-difference" gives baseline first-difference estimates using the controls from Donohue and Levitt (2001). The row labeled "All controls" includes a broad set of controls meant to allow flexible trends that vary with state-level characteristics. The row labeled "Double selection" reports results based on the double selection method outlined in this paper and selecting among the variables used in the "All controls" results.

Outline

Matching

Double Machine Learning to Adjust for Confounders

Synthetic Control Method

Machine Learning for Heterogenous Treatment Effects

Synthetic Control

- with matched differnces-in-differences, we matched each treated unit to a single similar control unit.
- **synthetic control**: construct a synthetic "match" from a weighted average of other individuals (based on covariates).

Synthetic Control

- with matched differnces-in-differences, we matched each treated unit to a single similar control unit.
- **synthetic control**: construct a synthetic "match" from a weighted average of other individuals (based on covariates).
- Statistically comparable to fixed effects or matching, but **powered up with ML.**

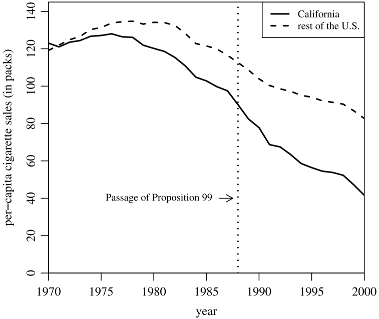
Example: Tobacco Laws in CA

In 1988, California passed anti-tobacco legislation (Proposition 99)

- Increased tax by $0.25/pack
- Extra tax revenues earmarked to health budget
- Funded anti-smoking campaigns
- Clean-air signs in closed spaces

```
pd.read_stata("http://fmwww.bc.edu/repec/bocode/s/synth_smoking.dta")
```

Trends in cigarette sales: Parallel trends fails



Rest of US is not a good comparison group for California

- Trends start diverging in 1970s, before the reform
- Parallel trends assumption fails ⇒ Cannot apply diff-in-diff

Source: Abadie, Diamond and Hainmueller (2010)

Synthetic Control Setup

- *Dataset:*
    - $j \in \{1, ... J+1\}$ *units,* $t = 1, 2, ..., T$ *periods.*
    - Outcome Y_{jt} (e.g. smoking), characteristics X_j

Synthetic Control Setup

- *Dataset:*
  - $j \in \{1, ...J+1\}$ *units,* $t = 1, 2, ..., T$ *periods.*
  - Outcome Y_{jt} (e.g. smoking), characteristics X_j
- Treatment:
  - Unit 1 (e.g. California) is exposed to intervention in periods $T_0 + 1, ..., T$
- Control group:
  - Remaining J units (other states) are potential controls (“donor pool”)

Synthetic Control Setup

- *Dataset:*
  - $j \in \{1,...J+1\}$ *units,* $t = 1,2,...,T$ *periods.*
  - Outcome Y_{jt} (e.g. smoking), characteristics X_j
- Treatment:
  - Unit 1 (e.g. California) is exposed to intervention in periods $T_0 + 1,...,T$
- Control group:
  - Remaining J units (other states) are potential controls ("donor pool")
- Objective:
  - find combination of untreated units that best approximates treated unit

Formalization

- Define weights $w_2, ..., w_{J+1} \geq 0$ where $\sum_{i=2}^{J+1} w_i = 1$.

$$\text{Synthetic Control Treatment Effect} = \underbrace{Y_{1t}}_{\text{treated}} - \underbrace{\sum_{j=2}^{J+1} w_j^* Y_{jt}}_{\text{synthetic}}$$

where $t \geq T_0$ is the post-intervention period

Formalization

- Define weights $w_2, \ldots, w_{J+1} \geq 0$ where $\sum_{i=2}^{J+1} w_i = 1$.

$$\text{Synthetic Control Treatment Effect} = \underbrace{Y_{1t}}_{\text{treated}} - \underbrace{\sum_{j=2}^{J+1} w_j^* Y_{jt}}_{\text{synthetic}}$$

where $t \geq T_0$ is the post-intervention period

- Synthetic control vector $\left(w_2^*, \ldots, w_{J+1}^*\right)$ is chosen to minimize

$$\sum_{m=1}^{k} v_m \left(x_{1m} - \sum_{j=2}^{J+1} w_j x_{jm} \right)^2$$

  - v_m = weight on m-th variable, chosen to minimize pre-reform MSE for Y, that is, match on pre-trends.
  - *(replacing this least-squares objective with a more general ML-type estimator would be a good student project)*

Synthetic California

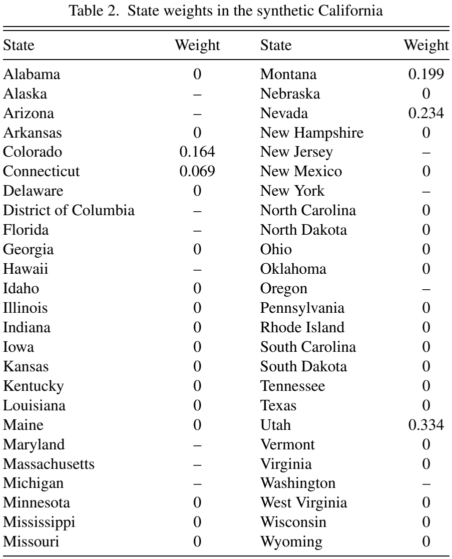
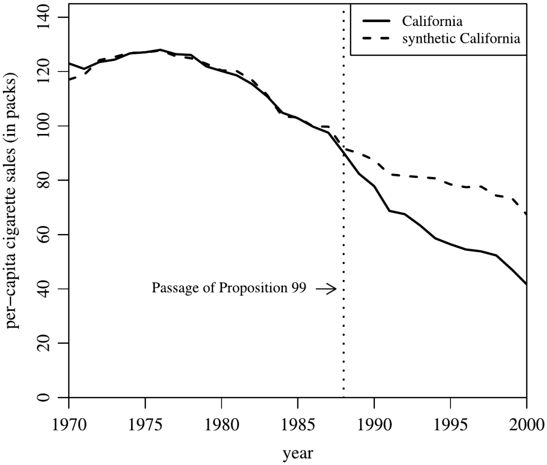


Table 2. State weights in the synthetic California

State	Weight	State	Weight
Alabama	0	Montana	0.199
Alaska	–	Nebraska	0
Arizona	–	Nevada	0.234
Arkansas	0	New Hampshire	0
Colorado	0.164	New Jersey	–
Connecticut	0.069	New Mexico	0
Delaware	0	New York	–
District of Columbia	–	North Carolina	0
Florida	–	North Dakota	0
Georgia	0	Ohio	0
Hawaii	–	Oklahoma	0
Idaho	0	Oregon	–
Illinois	0	Pennsylvania	0
Indiana	0	Rhode Island	0
Iowa	0	South Carolina	0
Kansas	0	South Dakota	0
Kentucky	0	Tennessee	0
Louisiana	0	Texas	0
Maine	0	Utah	0.334
Maryland	–	Vermont	0
Massachusetts	–	Virginia	0
Michigan	–	Washington	–
Minnesota	0	West Virginia	0
Mississippi	0	Wisconsin	0
Missouri	0	Wyoming	0



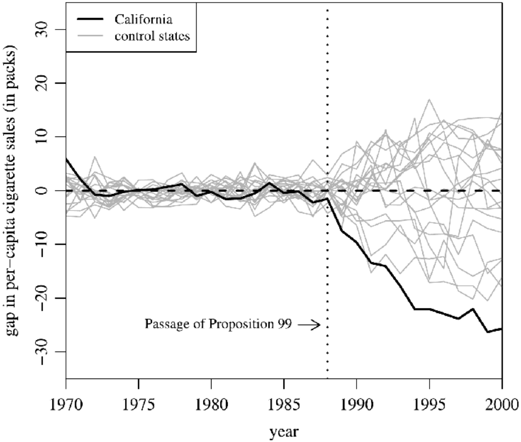
Source: Abadie, Diamond and Hainmueller (2010)

Inference

- Synthetic control does not give standard errors. Instead, use bootstrap approach:
    - Compare estimated synthetic control effect for california to distribution of placebo effects where treated unit is picked at random from donor pool.

Inference

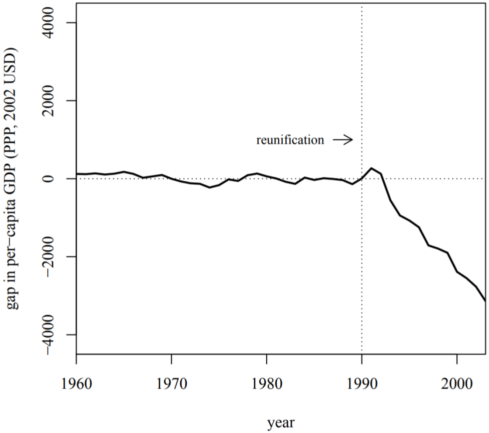
- Synthetic control does not give standard errors. Instead, use bootstrap approach:
  - Compare estimated synthetic control effect for california to distribution of placebo effects where treated unit is picked at random from donor pool.



Application 2: Effect of Reunification on West Germany GDP

Application 2: Effect of Reunification on West Germany GDP

Country	Weight	Country	Weight
Australia	0	Netherlands	0.11
Austria	0.47	New Zealand	0.11
Belgium	0	Norway	0
Canada	0	Portugal	0
Denmark	0	Spain	0
France	0	Sweden	0
Greece	0	Switzerland	0
Ireland	0	United Kingdom	0.17
Italy	0	United States	0
Japan	0		0.14



Practicalities

- In python, can use microsoft's SparseSC package.

```
# python
from numpy import hstack
from SparseSC import fit

# Let X be the features plus some of the targets
X = hstack([features, targets[:,:t])

# And let Y be the remaining targets
Y = targets[:,t:]

# fit the model:
sc = fit(X=X, Y=Y, model_type="full")
```

Summary: Synthetic Control

Advantages:

1. Works with a single treated unit.*
2. Makes explicit the contribution of each comparison unit to the synthetic control
3. Quantitative and qualitative ways to analyze similarities and differences of treatment and synthetic control
4. Formalizing how comparison units are chosen has nice properties for inference

Summary: Synthetic Control

Advantages:

1. Works with a single treated unit.*
2. Makes explicit the contribution of each comparison unit to the synthetic control
3. Quantitative and qualitative ways to analyze similarities and differences of treatment and synthetic control
4. Formalizing how comparison units are chosen has nice properties for inference

Limitations:

1. Still requires parallel trends / counterfactual assumption.
2. Could be idiosyncratic shocks to treated unit or comparison units
3. Cannot estimate effect of a single reform when multiple reforms passed at once

* With multiple treated units, use matched DD

Outline

Matching

Double Machine Learning to Adjust for Confounders

Synthetic Control Method

Machine Learning for Heterogenous Treatment Effects

Heterogeneous Treatment Effects

- Treatments don't affect every individual equally.
  - for example, effect of covid social distancing will depend on the age distribution.

Heterogeneous Treatment Effects

- Treatments don't affect every individual equally.
  - for example, effect of covid social distancing will depend on the age distribution.
- The simplest way to estimate these is to interact treatment with another covariate (the "**moderator**"):

$$Y_i = \beta_1 D_i + \beta_2 \text{Age}_i + \beta_3 D_i \text{Age}_i + \epsilon_i$$

  - here, β_3 summarizes heterogeneous impact by age ($\frac{\partial Y_i}{\partial D_i} = \beta_1 + \beta_3 \text{Age}_i$)

Activity: Brainstorming about Moderators

Revisit your customized causal graph:

- Add a new “bubble” with the header “Moderators”, and list some potential variables/characteristics that you expect to have a larger or smaller treatment effect.
- paste a link to your updated graph in the chat
- if you finish early, check out your classmate’s graphs.

Conditional Treatment Effects

- Consider the model

$$Y = \underbrace{\beta(X)}_{\text{CTE}} D + g(X) + \epsilon$$

  - the causal estimate $\beta(X)$ is a function of X.
  - this is the conditional treatment effect (CTE) – that is, the effect conditional on X.

Conditional Treatment Effects

- Consider the model
$$Y = \underbrace{\beta(X)}_{\text{CTE}} D + g(X) + \epsilon$$
  - the causal estimate $\beta(X)$ is a function of X.
  - this is the conditional treatment effect (CTE) – that is, the effect conditional on X.
- Can learn flexible representation of $\hat{\beta}(X)$ using machine learning.

T-Learner Method

- Residualize Y on the fixed effects and controls to get rid of $g(X)$:

$$Y = \beta(X)D + \epsilon$$

  - if D is randomly assigned (e.g. RCT), this is not necessary.
  - Note that the standard (non-conditional) treatment effect is then immediately obtainable by OLS: $\hat{\beta}_{OLS} = \text{Cov}(Y, D)/\text{Var}(D)$.

T-Learner Method

- Residualize Y on the fixed effects and controls to get rid of $g(X)$:

$$Y = \beta(X)D + \epsilon$$

  - if D is randomly assigned (e.g. RCT), this is not necessary.
  - Note that the standard (non-conditional) treatment effect is then immediately obtainable by OLS: $\hat{\beta}_{OLS} = \mathrm{Cov}(Y, D)/\mathrm{Var}(D)$.

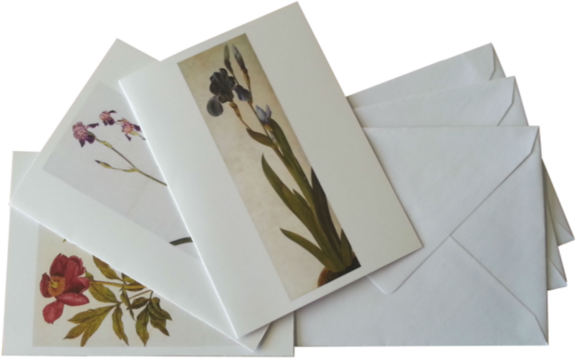
T-Learner Method:

- Using any machine learning method:
  - Learn $\mu_0(X) = \mathbb{E}(Y|X, D = 0)$
  - Learn $\mu_1(X) = \mathbb{E}(Y|X, D = 1)$
- Tune parameters in whole dataset using cross-validation.
- The conditional treatment effect estimate is $\hat{\beta}(X) = \mu_1(X) - \mu_0(X)$.

See Knaus, Lechner, and Strittmatter (2020) for a review of different methods/extensions.

Cagala et al 2021, Optimal Targeting in Fundraising: Setting

Figure 1: The gift consisting of the three folded cards and envelopes



Notes: The gifts consisted of three different folded cards showing flower motifs from paintings of Albrecht Dürer plus three envelopes.

- charity field experiment: 2345 warm-list donors, 17,425 cold-list donors
  - treament group (got a gift): 1180 warm-list, 2283 cold-list

The gifts work

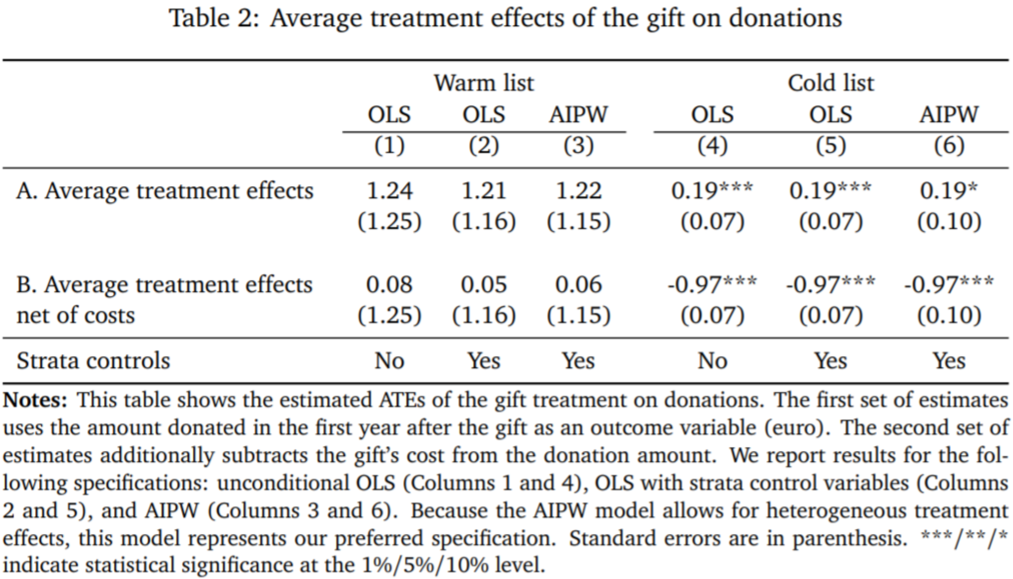


Table 2: Average treatment effects of the gift on donations

	Warm list			Cold list		
	OLS	OLS	AIPW	OLS	OLS	AIPW
	(1)	(2)	(3)	(4)	(5)	(6)
A. Average treatment effects	1.24	1.21	1.22	0.19***	0.19***	0.19*
	(1.25)	(1.16)	(1.15)	(0.07)	(0.07)	(0.10)
B. Average treatment effects net of costs	0.08	0.05	0.06	-0.97***	-0.97***	-0.97***
	(1.25)	(1.16)	(1.15)	(0.07)	(0.07)	(0.10)
Strata controls	No	Yes	Yes	No	Yes	Yes

Notes: This table shows the estimated ATEs of the gift treatment on donations. The first set of estimates uses the amount donated in the first year after the gift as an outcome variable (euro). The second set of estimates additionally subtracts the gift's cost from the donation amount. We report results for the following specifications: unconditional OLS (Columns 1 and 4), OLS with strata control variables (Columns 2 and 5), and AIPW (Columns 3 and 6). Because the AIPW model allows for heterogeneous treatment effects, this model represents our preferred specification. Standard errors are in parenthesis. ***/**/* indicate statistical significance at the 1%/5%/10% level.

Conditional Treatment Effects and Targeting

Cagala et al 2021

- data on donor characteristics X:
    - demographics, donor history
    - detailed info on neighborhood from Google Maps API, collected using the mailing address.

Conditional Treatment Effects and Targeting

Cagala et al 2021

- data on donor characteristics X:
  - demographics, donor history
  - detailed info on neighborhood from Google Maps API, collected using the mailing address.
- for each treatment group $D =$ gift or no gift, train a machine learning model on characteristics X to predict donations $\hat{Y}(X|D)$.

Conditional Treatment Effects and Targeting

Cagala et al 2021

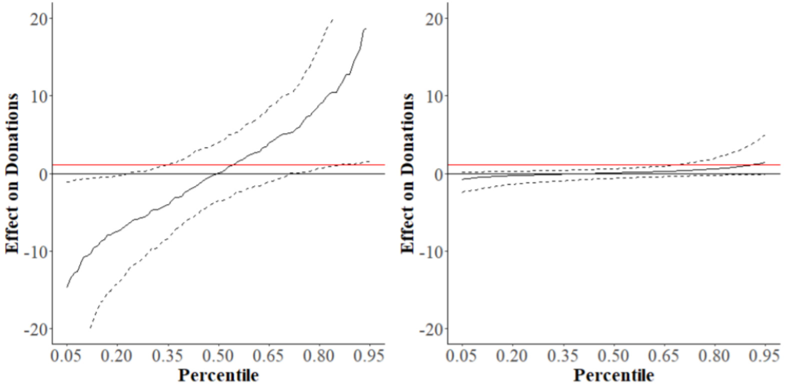
- data on donor characteristics X:
  - demographics, donor history
  - detailed info on neighborhood from Google Maps API, collected using the mailing address.
- for each treatment group $D =$ gift or no gift, train a machine learning model on characteristics X to predict donations $\hat{Y}(X|D)$.
- Optimal targeting rule is (roughly) ranking by $\hat{Y}$ and giving treatment to those for which

  $$\hat{Y}(X|1) - c > \hat{Y}(X|0)$$

  where c is the cost of the gift.

Effects are Heterogeneous

Figure 2: Sorted effects



(a) Warm list

(b) Cold list

Notes: This figure shows the heterogeneity of the effect of the gift on the donation amount. To that end, it sorts the estimated conditional average treatment effects by size and plots the size of the treatment effect in euro (vertical axis) against the percentiles of the effect size (horizontal axis). The red horizontal line represents the cost of the gift (1.16 euro). The solid line depicts the sorted effects. We report results between the 5 and 95 percentiles. The dashed lines report uniformly valid 95% confidence intervals, which build on a multiplier bootstrap and 500 replications.

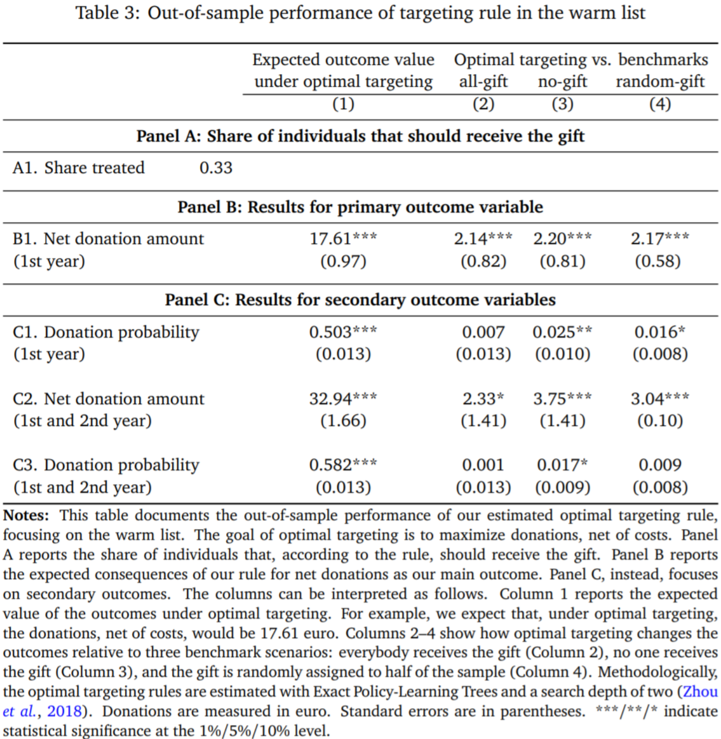


Table 3: Out-of-sample performance of targeting rule in the warm list

	Expected outcome value under optimal targeting	Optimal targeting vs. benchmarks		
		all-gift	no-gift	random-gift
	(1)	(2)	(3)	(4)
Panel A: Share of individuals that should receive the gift				
A1. Share treated	0.33			
Panel B: Results for primary outcome variable				
B1. Net donation amount	17.61***	2.14***	2.20***	2.17***
(1st year)	(0.97)	(0.82)	(0.81)	(0.58)
Panel C: Results for secondary outcome variables				
C1. Donation probability	0.503***	0.007	0.025**	0.016*
(1st year)	(0.013)	(0.013)	(0.010)	(0.008)
C2. Net donation amount	32.94***	2.33*	3.75***	3.04***
(1st and 2nd year)	(1.66)	(1.41)	(1.41)	(0.10)
C3. Donation probability	0.582***	0.001	0.017*	0.009
(1st and 2nd year)	(0.013)	(0.013)	(0.009)	(0.008)

Notes: This table documents the out-of-sample performance of our estimated optimal targeting rule, focusing on the warm list. The goal of optimal targeting is to maximize donations, net of costs. Panel A reports the share of individuals that, according to the rule, should receive the gift. Panel B reports the expected consequences of our rule for net donations as our main outcome. Panel C, instead, focuses on secondary outcomes. The columns can be interpreted as follows. Column 1 reports the expected value of the outcomes under optimal targeting. For example, we expect that, under optimal targeting, the donations, net of costs, would be 17.61 euro. Columns 2–4 show how optimal targeting changes the outcomes relative to three benchmark scenarios: everybody receives the gift (Column 2), no one receives the gift (Column 3), and the gift is randomly assigned to half of the sample (Column 4). Methodologically, the optimal targeting rules are estimated with Exact Policy-Learning Trees and a search depth of two (Zhou *et al.*, 2018). Donations are measured in euro. Standard errors are in parentheses. ***/**/* indicate statistical significance at the 1%/5%/10% level.

HW06 Homework Note

HW06 Homework Note

Two parts:

1. Complete a jupyter notebook on double ML / synthetic control / heterogenous treatment effects.

HW06 Homework Note

Two parts:

1. Complete a jupyter notebook on double ML / synthetic control / heterogenous treatment effects.
2. Peer review of response essays:
   - You will be randomly assigned two anonymized essays from one of your classmates.
   - Write one paragraph (5-10 sentences) about each essay, providing constructive feedback/suggestions. Identify at least one strength of the essay, and one area for improvement.
   - Follow the rubric on the homework assignments page.

Short Essay on Mullainathan Article (rest of class)

- Review “Biased algorithms are easier to fix than biased people” by Sendhil Mullainathan in *New York Times* (`bit.ly/nyt-bias`).
  - Think of another task where fixing biases in an algorithm is probably easier than fixing it in humans.
  - Can you think of the opposite case — a task where fixing biases in humans is easier than fixing biases in algorithms?
  - Has your attitude to this article changed at all since the first week of class?
- put your answers in a shared doc and paste a link here: `https://padlet.com/eash44/p6ypvf4uodlgu7jz`