Lecture 10: Heterogeneous Treatment Effects

The application of machine learning in causal inference

Xi Chen

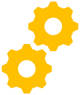
Rotterdam School of Management
Erasmus University Rotterdam

June 4, 2023

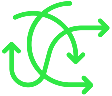
Outline

Outline

The power of machine learning



Predictive models automatically run adapted.



Dealing with seemingly complex problems.

Preventing overfitting by bias-efficiency tradeoff.

In recent years, an active area in causal inference is causal machine learning that applies ML methods to causal inference problems.

Some examples

- Using machine learning models to calculate propensity scores (Lee et al. 2010).
  - To alleviate the concerns over specification errors.
- Genetic matching (Diamond and Sekhon 2005).
  - To use genetic algorithm to automate the process of finding a good match.
- Selection of control variables for adjustment in RDD (Anastasopoulos 2019).
  - With an automatic LASSO procedure.

Some examples

- Using machine learning models to calculate propensity scores (Lee et al. 2010).
  - To alleviate the concerns over specification errors.
- Genetic matching (Diamond and Sekhon 2005).
  - To use genetic algorithm to automate the process of finding a good match.
- Selection of control variables for adjustment in RDD (Anastasopoulos 2019).
  - With an automatic LASSO procedure.

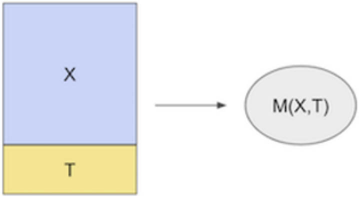
Some examples

- Using machine learning models to calculate propensity scores (Lee et al. 2010).
  - To alleviate the concerns over specification errors.
- Genetic matching (Diamond and Sekhon 2005).
  - To use genetic algorithm to automate the process of finding a good match.
- Selection of control variables for adjustment in RDD (Anastasopoulos 2019).
  - With an automatic LASSO procedure.

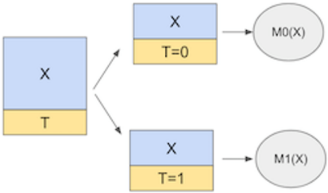
Meta learners for heterogeneous treatment effects

If we want to generalize the treatment effects, we need to know HTE function $\tau(x) = E\left(Y^1 - Y^0 \mid X_i = x\right)$

- S(ingle)-learner: fit a single ML model to $E(Y \mid D, X)$.
- T(two)-learner: fit two ML models to $E\left(Y^0 \mid D, X\right)$ and $E\left(Y^1 \mid D, X\right)$.



(a) S-learner



(b) T-learner

RSM Erasmus

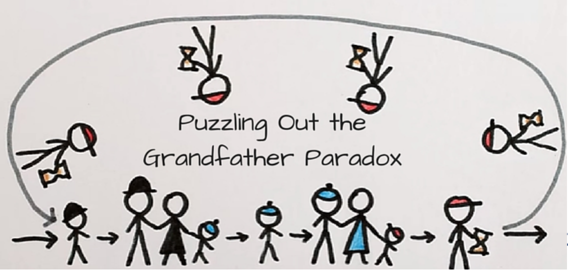
Outline

Back to the fundamental problem

Fact (The Fundamental Problem of Causal Inference)

For a unit, only one causal state can be realized, and the investigator can only observe the potential outcome from the realized causal state.

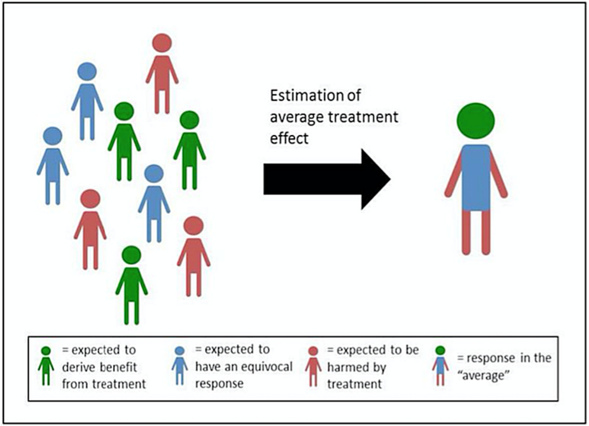
Implications: the individual treatment effects are inherently unknowable.



Back to the fundamental problem

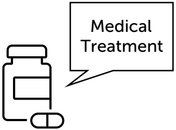
Solution 1: To assume ~~homogeneity~~ of units.
Solution 2: Potential outcome framework $\rightarrow$ ATE instead of ITE.

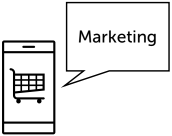
Problem: who's who?

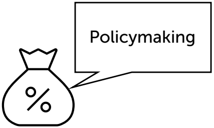


Two problems without HTEs

We identify ATE, but cannot predict how a particular person responds to the treatment...



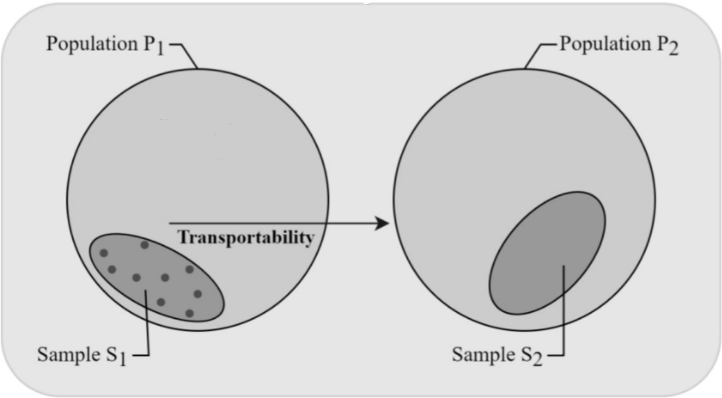




A treatment does not work for all, but may work for some.

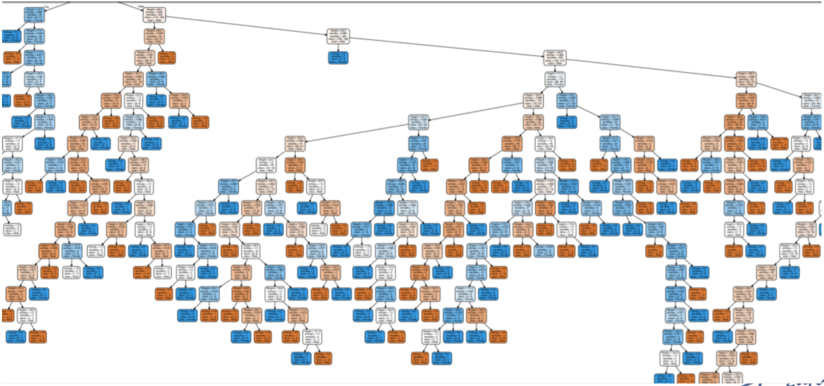
Predicting the treatment effects...

To predict the treatment effect for a new finite-population:



Background: data with rich features

We have accumulated and compiled data with rich sets of features...



Outline

The traditional approach

1. Specify a linear model:

$$Y = \alpha + \beta D + \varepsilon$$

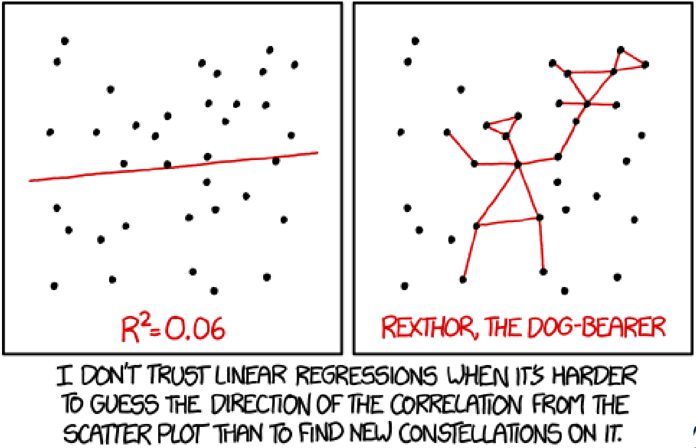
2. Adding interaction terms:

$$\begin{aligned} Y = \alpha + \beta D + & \underbrace{\lambda_1 DX_1 + \cdots + \lambda_K DX_K}_{\text{Two-way interactions}} \\ & + \underbrace{\lambda_{11} DX_1^2 + \cdots + \lambda_{KK} DX_K^2}_{\text{Three-way interactions}} \\ & + \cdots + \varepsilon \end{aligned}$$

3. Gather all λ's and mission accomplished.

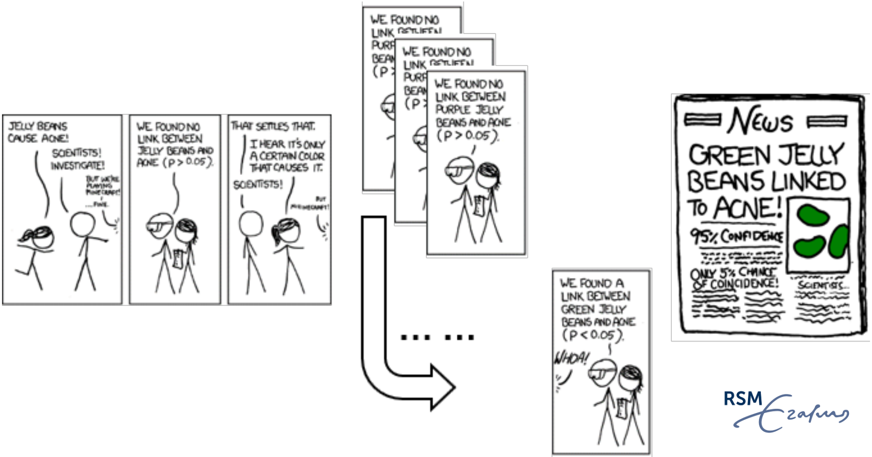
Problems of the traditional approach

It's parametric: linear, additive and separable.



Problems of the traditional approach

False discoveries from multiple testing, especially with **many features**...



RSM Erasmus

Naive applications of machine learning methods to HTEs[1]

S(ingle)-learner

1. estimate $\mu_d(x) = E(Y_i \mid D_i = d, X_i = x)$ using a single model.
2. compute $\hat{\tau}(x) = \hat{\mu}_1(x) - \hat{\mu}_1(x)$.

Example (S-learner)

LASSO with SVM to regularize over-fitting.

T(wo)-leaner

1. estimate $\mu_d(x) = E(Y_i \mid D_i = d, X_i)$ separately for $d = \{0, 1\}$.
2. compute $\hat{\tau}(x) = \hat{\mu}_1(x) - \hat{\mu}_0(x)$.

Example (T-learner)

Decision trees with regularization (tree depth).

[1] See Kunzel et al. (2019) for more details.

Naive applications of machine learning to HTEs

X-learner (X stands for "exchange")

1. estimate $\mu_d(x) = E(Y_i \mid D_i, X_i)$ separately for $D_i = 0$ or 1.
2. impute missing potential outcomes as an out-of-sample prediction (treatment $\hat{\mu}_1(x) \rightleftarrows$ control $\hat{\mu}_0(x)$).
3. impute the individual treatment effects $\tau_i(X_i)$ with observed outcomes Y_i^{obs} and imputed potential outcomes.
4. use the imputed ITEs $\hat{\tau}_i(X_i)$ as the response variable and use any supervised learning method with $\hat{\tau}_i(X_i) = f(X_i)$.

Naive applications of machine learning to HTEs

X-learner (X stands for "exchange")

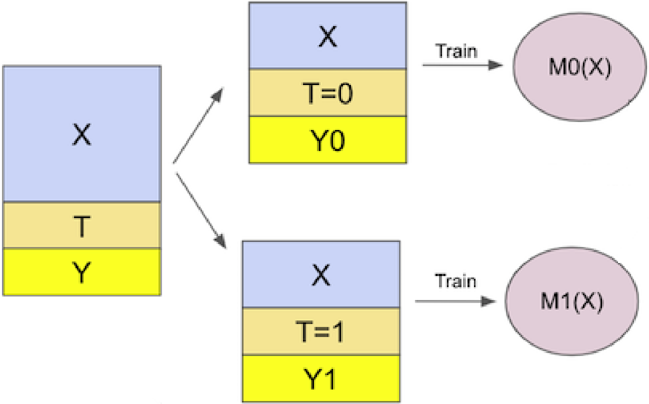
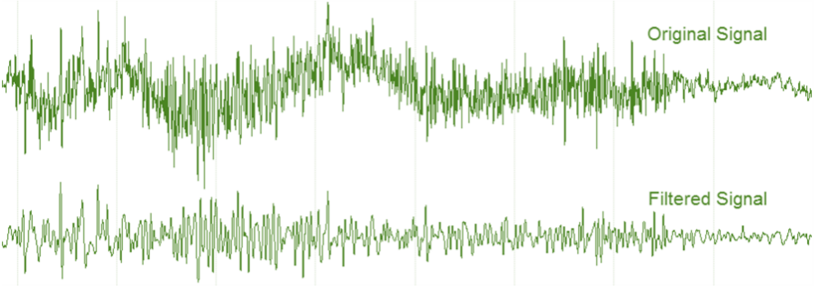


Figure: X-Learner

Problems with naive applications

HTEs require "good estimates" of **variance of ATE's**.
Naive applications: **no inference on the variance or second-moments**.

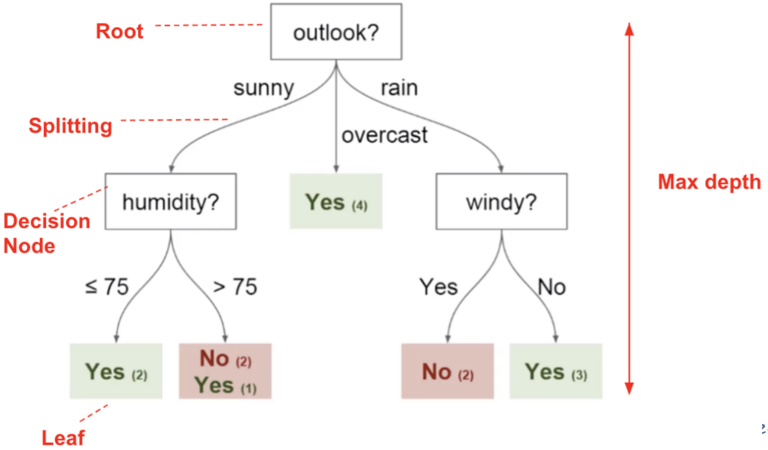


"Filtering the nuisance"

Outline

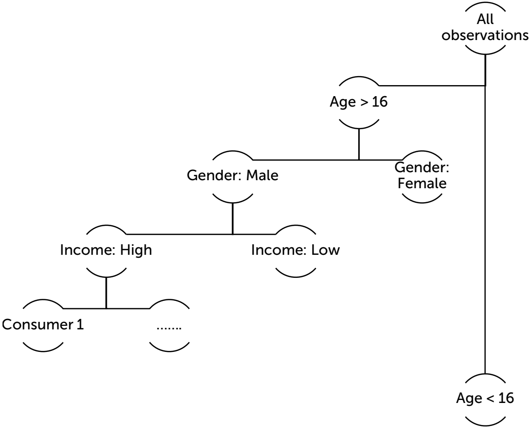
The ABCs of decision trees

Some terminologies with *a classifier of 365 days with weather conditions*:



Build a tree to predict purchases

1. Given some data (e.g. age, gender, and income), build a tree.
2. For a new case, check which leave the case is in.
3. Use $\bar{Y}$ of the leave as the predicted purchase.



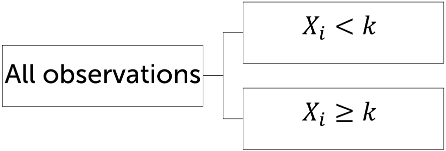
Tree building: how to make a partition

How to make a split?

- Choose a cutoff k to minimize a loss function
- Example, mean squared errors (MSE) with

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \overline{Y}_{j:j\in l(X_i|\Pi)}\right)^2,$$

with Π a partition and $l(\cdot)$ a leaf



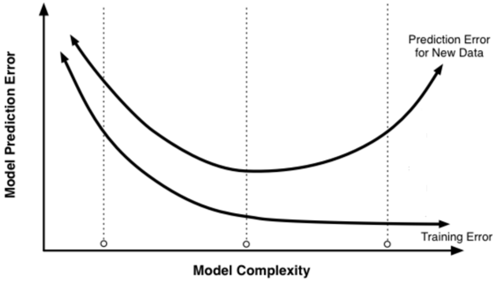
Choose k to minimize MSE

Tree building: when to stop splitting?

How many leaves to have (and/or the max depth)?

- With enough fine partitions $\mapsto$ 1 consumer in a leaf.
  - Perfect (in-sample) fit but uselessly high variance.

Regularization: to keep splits that improve MSE by at least c.



Bias-variance trade-off

Extending decision trees to causal trees

In a **decision tree**, the loss function is defined as MSE:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} \left(Y_i - \overline{Y}_{j:j \in l(X_i|\Pi)} \right)^2$$

Similarly, we can define a **causal tree** on treatment effects:

$$\text{MSE}_{\text{Causal}} = \frac{1}{N} \sum_{i=1}^{N} \left(\tau_i - \overline{\tau}_{j:j \in l(X_i|\Pi)} \right)^2$$

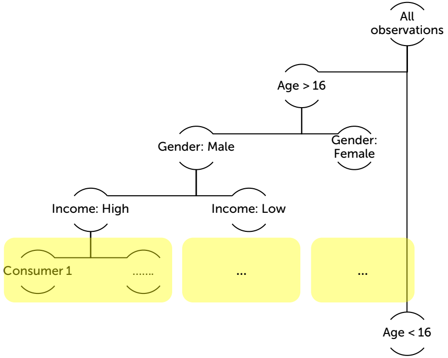
Fact (Challenge in defining causal $\text{MSE}_{\text{Causal}}$)

τ_i is unobserved, because of the fundamental problem of causal inference!

Extending decision trees to causal trees

Think about the idea of subclassification in Lecture 4.

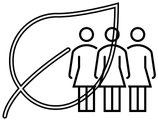
- For each sub-class, we can estimate the treatment effects!
- In a decision tree, **a sub-class = a leaf in a tree**.



Extending decision trees to causal trees

Each leaf is a subclass, and $\tau(X_i)$ is defined for a leaf $l(X_i \mid \Pi)$.
Summarize over all the leaves to obtain treatment effects.

Decision Trees
Y_i

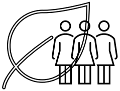


Causal Trees
τ_i

The ATE of a leaf: $\text{ATE}_l = \overline{Y}_l^1 - \overline{Y}_l^0$.

Extending decision trees to causal trees

Decision Trees
Y_i



Causal Trees
τ_i

Sample size requirement

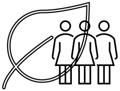
To estimate $I(X_i \mid \Pi)$ for each leaf, we need enough samples for each treatment condition within a leaf, i.e., restricting trees to have at least $2 \cdot k$ samples for any leaf.

The insight is from PO

We focus on ATE to “avoid” the fundamental problem.

Extending decision trees to causal trees

Decision Trees
Y_i



Causal Trees
τ_i

Sample size requirement

To estimate $I(X_i \mid \Pi)$ for each leaf, we need enough samples for each treatment condition within a leaf, i.e., restricting trees to have at least $2 \cdot k$ samples for any leaf.

The insight is from PO

We focus on ATE to "avoid" the fundamental problem.

How to build causal trees?

Naive application of the traditional MSE criterion:

$$\text{MSE}_0 = \frac{1}{N_l} \sum_{l=1}^{N_l} \Bigg(\underbrace{\tau_l}_{\text{ATE of a leaf}} - \underbrace{\overline{\tau}_l}_{\text{Average of ATEs of all leaves}} \Bigg)^2$$

Question: is this MSE adequate for our purpose?

MSE_0: the cross-leaf variance.
Minimize MSE_0?

But we want to find heterogeneity.
MSE_0 is against our objective!

How to build causal trees?

Reverse the sign of MSE_0

$$\mathrm{MSE}_1 = -\frac{1}{N_l}\sum_{l=1}^{N_l}\left(\underbrace{\tau_l}_{\text{ATE of a leaf}} - \underbrace{\overline{\tau}_l}_{\text{Average of ATEs of all leaves}}\right)^2$$

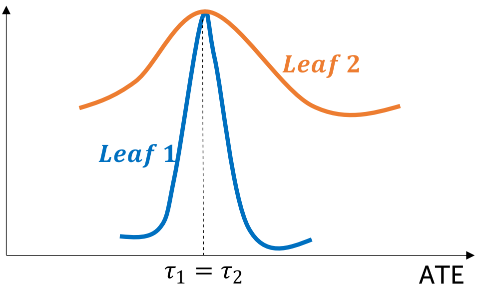
To minimize $\mathrm{MSE}_1 \Rightarrow$ to maximize the cross-leaf variance.
We intend to find heterogeneity of treatment effects $\tau(X_i)$.

Question: is MSE_1 adequate?

How to build causal trees?

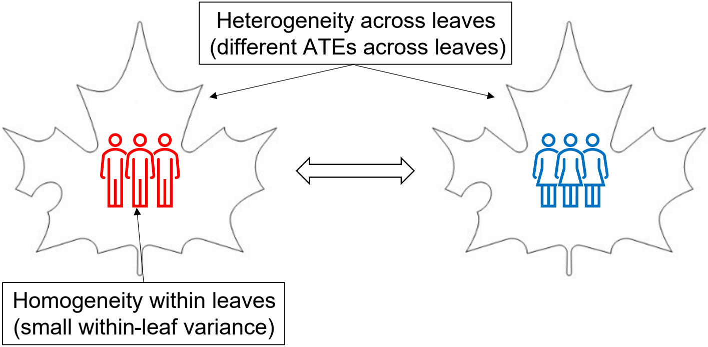
In a causal tree, suppose we have two leaves of the same ATE.

- By definition of MSE_1, these two leaves are equivalent.
- But, the variances of leaf-specific ATEs are different.



Question: which leaf is better for prediction?

How to build causal trees?



To balance cross-leaf vs. within-leaf variance:

$$\mathrm{MSE}_2 = -\frac{1}{N_l}\sum_{l=1}^{N_l}\underbrace{\left(\tau_l - \overline{\tau}_l\right)^2}_{\text{Cross-leaf Variance}} + \frac{1}{N_l}\sum_{l=1}^{N_l}\left(\underbrace{\frac{S_1^2(l)}{N_1(l)} + \frac{S_0^2(l)}{N_0(l)}}_{\text{Within-leaf Variance}}\right)$$

Building causal trees: some notes

- The MSE_2 introduced here is not exactly the one in Athey and Imbens (2016), but the intuition is the same.
- The authors first extended the standard MSE for decision trees and then generalized it for causal trees.
- Here, we work "backwards" to understand the intuitions of building causal trees.
- For more details, please check Athey and Imbens (2016).

Building causal trees: some notes

- The MSE_2 introduced here is not exactly the one in Athey and Imbens (2016), but the intuition is the same.
- The authors first extended the standard MSE for decision trees and then generalized it for causal trees.
- Here, we work "backwards" to understand the intuitions of building causal trees.
- For more details, please check Athey and Imbens (2016).

Building causal trees: some notes

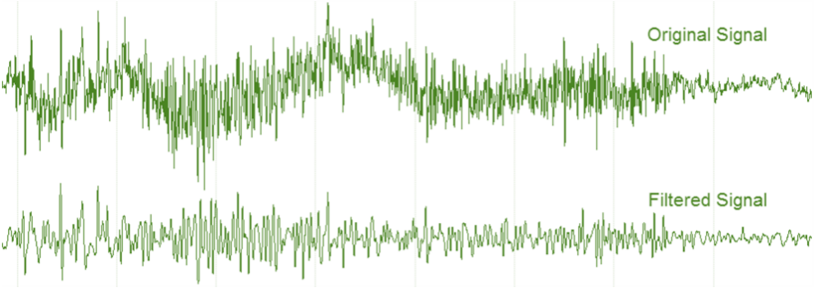
- The MSE_2 introduced here is not exactly the one in Athey and Imbens (2016), but the intuition is the same.
- The authors first extended the standard MSE for decision trees and then generalized it for causal trees.
- Here, we work "backwards" to understand the intuitions of building causal trees.
- For more details, please check Athey and Imbens (2016).

Building causal trees: some notes

- The MSE_2 introduced here is not exactly the one in Athey and Imbens (2016), but the intuition is the same.
- The authors first extended the standard MSE for decision trees and then generalized it for causal trees.
- Here, we work "backwards" to understand the intuitions of building causal trees.
- For more details, please check Athey and Imbens (2016).

How to build causal trees?

- Now we have a criterion to produce partitions/splits...
- Still need **an inference procedure** for the variance of ATE.
  - So we know how heterogeneous the ATE is.



"Filtering the nuisance"

The inference procedure: resampling

Sub-sampling: to sample a fraction of all observations (without replacement) to create a subsample.

Sub-sampling

Given a data $X = \{X_1, \cdots, X_N\}$, the size of a subsample S, and a statistic $T(X)$:

- Sample $X^* = \{X_1^*, \cdots, X_S^*\}$ from X without replacement.
- Calculate the statistic $T(X^*)$.
- Repeat many time (at most $\binom{N}{S}$ times).

Why sub-sampling?

Why sub-sampling instead of bootstrapping?

- Bootstrapping may not work here.
- A deterministic operation creates **"holes" in the distribution of the statistics** $T(X^*)$.
- **Causal trees**: to use MSE to determine splits (a minimization).

Example (The failure of bootstrapping)

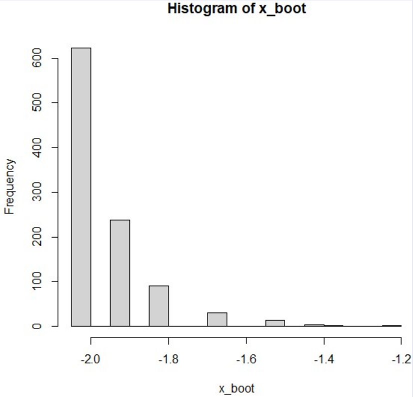
Suppose $X_1, \cdots, X_N \sim \text{Uniform}(0, 1)$, and a statistic: $T(X) = \min(X_1, \cdots, X_N)$. If you bootstrap, the test statistic would not converge to the true distribution.

Why sub-sampling?

Run the example in R

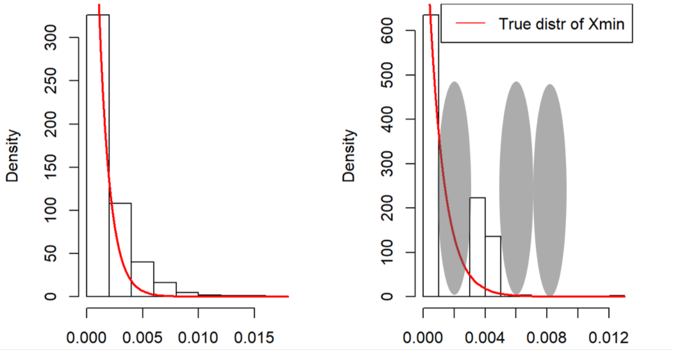
```
x <- rnorm(100)
x_boot <- rep(0,1000)

for (i in 1:1000) {
xs <-sample(x,100,replace = T)
x_boot[i] <- min(xs)
}
```

The distribution of X_{min}



Why sub-sampling?

Illustration: deterministic transformations create “holes” in the distribution.

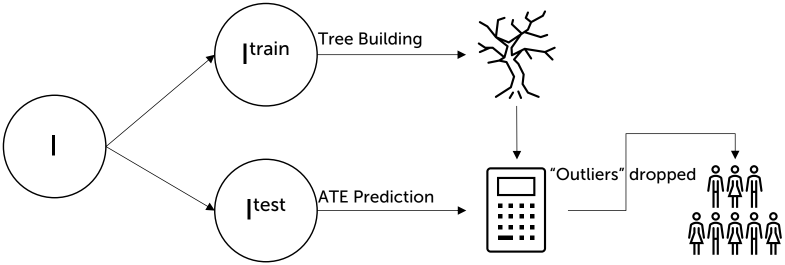


True Distribution Bootstrap Distribution

Building “honest trees”

In Wager and Athey (2018), a procedure to build “honest trees”.

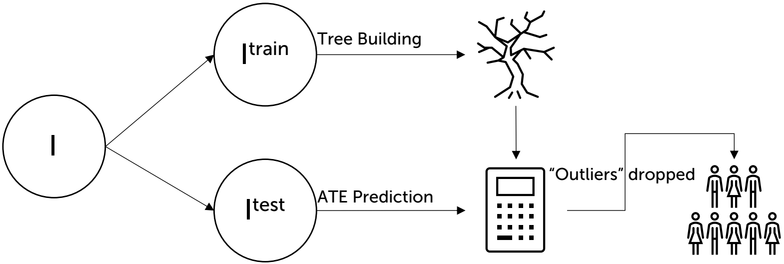
1. Split I sample (evenly) into a train set I^{train} and a test set I^{test}.
2. Build a tree with $I^{train}$and predict leaf-specific ATEs with I^{test}.



Building "honest trees"

The key insights into "honest trees"

- The splitting into $\{I^{\text{train}}, I^{\text{test}}\}$ is **cross-validation** from ML.
- The idea of **trimming** from CI: leaves in I^{train}that do not produce an ATE for I^{test} would be dropped.



The "dropping" reduces bias in HTEs.

Putting things together

Given data of a tuple $\{X_i, D_i, Y_i\}$ of N observations, the block size α, the minimum sample size per condition per leaf k, and the total number of repetition S, run the following:

A general procedure to causal forest

At a particular repetition s:

1. Draw a random subsample of size αN without replacement as I_s.
2. Split I_s to I_s^{Train} and I_s^{Test}.
3. Grow a tree T_s with I_s^{Train} using $\text{MSE}_{\text{Causal}}$ and restrict size of leaves $> k$.
4. Assign observations in I_s^{Test} with T_s and calculate $\tau_l^s\left(I_s^{\text{Test}}\right)$.
5. With full sample N, assign people with T_s, and calculate $\tau_i^s(N)$.
6. Repeat 1-5 S times and let $\tau_i = 1/S \sum_s \tau_i^s(N)$.

Outline

Doubly robust estimator

A modified weighting estimator which is robust as long as the propensity score or the regression model is correctly specified.

$$
\begin{aligned}
\tau_{\text{DR}} = & \frac{1}{N} \sum_{i=1}^{N} \left[\frac{D_i Y_i}{e(X_i)} - \frac{D_i - e(X_i)}{e(X_i)} \underbrace{m_1(X_i)}_{\text{Fit } Y_i^1 \sim X_i} \right] \\
& - \frac{1}{N} \sum_{i=1}^{N} \left[\frac{(1 - D_i) Y_i}{1 - e(X_i)} + \frac{D_i - e(X_i)}{1 - e(X_i)} \underbrace{m_0(X_i)}_{\text{Fit } Y_i^0 \sim X_i} \right]
\end{aligned}
$$

Three conditional distributions,

$$
\begin{aligned}
e(X_i) &= E(D_i \mid X_i) \\
m_1(X_i) &= E(Y_i \mid X_i, D_i = 1) \\
m_0(X_i) &= E(Y_i \mid X_i, D_i = 0)
\end{aligned}
$$

RSM Erasmus

Technical notes

Why it's doubly robust?

The first term in τ_{DR} is (a similar procedure for the second term):

$$
\begin{aligned}
E\left[\frac{D_i Y_i}{e(X_i)} - \frac{D_i - e(X_i)}{e(X_i)} m_1(X_i)\right] &= E\left[\frac{D_i Y_i^1}{e(X_i)} - \frac{D_i - e(X_i)}{e(X_i)} m_1(X_i)\right] \\
&= E\left[Y_i^1 + \frac{D_i - e(X_i)}{e(X_i)}\left(Y_i^1 - m_1(X_i)\right)\right] \\
&= E\left(Y_i^1\right) + E\left[\frac{D_i - e(X_i)}{e(X_i)}\left(Y_i^1 - m_1(X_i)\right)\right]
\end{aligned}
$$

The second term is zero, either one of the conditions hold,

$$
\begin{cases}
e(X_i) & = E(D_i \mid X_i) \\
m_1(X_i) & = E(Y_i \mid X_i, D_i = 1)
\end{cases}
$$

In other word, $e(X_i)$ or $m_1(X_i)$ consistently estimates the conditional expectations.

Tuning the trees

A new loss function to tune the parameters of trees:

$$\widetilde{\tau}(\cdot) = \operatorname{argmin}_{\tau} \left(\frac{1}{N} \sum_{i=1}^{N} \left[\underbrace{(Y_i - m^*(X_i)) - (D_i - e^*(X_i))\,\tau(X_i)}_{\text{Doubly Robust ATE Estimators}} \right]^2 + \underbrace{\Lambda_N(\tau(\cdot))}_{\text{Regularization}} \right)$$

For more details, see Wager and Athey (2018).

Generalized with moment conditions

See more details in Athey et al. (2019).
A package "grf" for R, and the online resource is here:
`https://grf-labs.github.io/grf/index.html`
Applications:

- generalized weighting estimator
- instrumental variables
- treatment heterogeneity
- ...

References I

- Künzel, S. R., Sekhon, J. S., Bickel, P. J., & Yu, B. (2019). Metalearners for estimating heterogeneous treatment effects using machine learning. Proceedings of the national academy of sciences, 116(10), 4156-4165.
- Athey, S., & Imbens, G. (2016). Recursive partitioning for heterogeneous causal effects. Proceedings of the National Academy of Sciences, 113(27), 7353-7360.
- Wager, S., & Athey, S. (2018). Estimation and inference of heterogeneous treatment effects using random forests. Journal of the American Statistical Association, 113(523), 1228-1242.
- Davis, J., & Heller, S. B. (2017). Using causal forests to predict treatment heterogeneity: An application to summer jobs. American Economic Review, 107(5), 546-50.

References II

- Athey, S., Tibshirani, J., & Wager, S. (2019). Generalized random forests. Annals of Statistics, 47(2), 1148-1178.
- Nie, X., & Wager, S. (2021). Quasi-oracle estimation of heterogeneous treatment effects. Biometrika, 108(2), 299-319.
- Lee, B. K., Lessler, J., & Stuart, E. A. (2010). Improving propensity score weighting using machine learning. Statistics in medicine, 29(3), 337-346.
- Diamond, A., & Sekhon, J. (2005). Genetic matching for estimating causal effects: A new method of achieving balance in observational studies.
- Anastasopoulos, J. (2019). Principled estimation of regression discontinuity designs with covariates: a machine learning approach. arXiv preprint arXiv:1910.06381.