



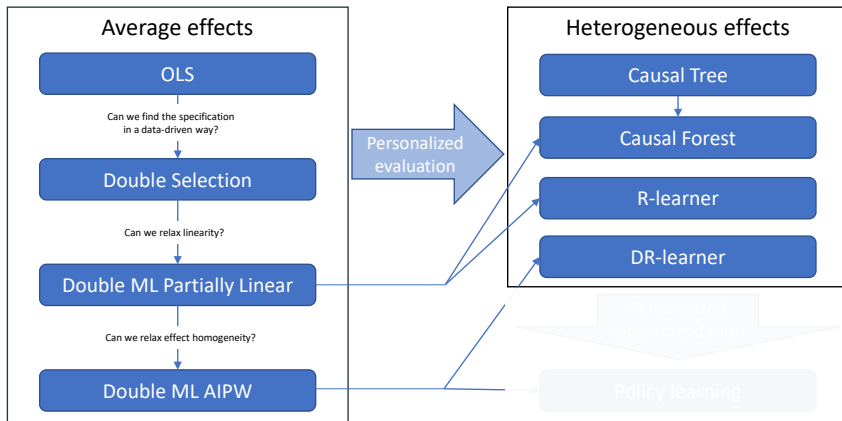
Causal Machine Learning

Policy learning

Michael Knaus

March 13, 2023

State of the journey



We focused on policy evaluation on an aggregate and personalized level

Plan of this morning

How to use Causal ML for decision making (policy recommendation)?

1. Conceptual framework
2. Offline policy learning with binary treatments
3. Offline policy learning with multiple treatments
4. Online policy learning (bandits)
5. Wrapping-up

Conceptual framework

From evaluation to recommendation

So far we focused on treatment **evaluation**: "What works (for whom)?"

Today we focus on treatment **recommendation**: "How to optimally treat (whom)?"

⇒ We are interested in **data-driven (personalized) treatment recommendations**

But isn't this settled given what we learned already about predicting effects? 🤔

Well, kind of, but not really...

Let's have a closer look

Conceptual framework: notation

Reminder:

$$W \in \{0, 1\}$$

Binary treatment indicator

$$Y(w)$$

Potential outcome (PO) under treatment w

$$X$$

Exogenous covariate(s)

$$\mu_w(x) = \mathbb{E}[Y(w) \mid X = x]$$

Conditional Average PO

$$\tau(x) = \mu_1(x) - \mu_0(x)$$

Conditional Average Treatment Effect (CATE)

New:

$$\pi(x) \in \{0, 1\}$$

Policy rule for x (conditional treatment choice)

$$Y(\pi(X))$$

PO under policy $\pi(X)$

$$Q(\pi) = \mathbb{E}[Y(\pi(X))]$$

Value function (expected outcome under policy)

Without loss of generality we assume throughout that a higher outcome is better

Example

Consider the case of two different assignment rules π^1 and π^2

i	$Y_i(0)$	$Y_i(1)$	π^1	π^2	$Y_i(\pi^1)$	$Y_i(\pi^2)$
1	$Y_1(0)$	$Y_1(1)$	0	0	$Y_1(0)$	$Y_1(0)$
2	$Y_2(0)$	$Y_2(1)$	1	0	$Y_2(1)$	$Y_2(0)$
3	$Y_3(0)$	$Y_3(1)$	0	1	$Y_3(0)$	$Y_3(1)$
4	$Y_4(0)$	$Y_4(1)$	1	1	$Y_4(1)$	$Y_4(1)$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots

The value functions of the policies take then the expectations over the two right columns: $Q(\pi^1) = \mathbb{E}[Y(\pi^1)]$ and $Q(\pi^2) = \mathbb{E}[Y(\pi^2)]$

Unreachable goals

In our dreams we would like to assign individuals to treatment with higher PO under treatment than without: $\pi^* = \mathbb{1}[Y(1) > Y(0)] = \mathbb{1}[Y(1) - Y(0) > 0]$

This is unfortunately not possible due to the fundamental problem of causal inference

However, we know from previous lectures that we could identify the CATE

This suggests the following optimal policy based on the CATE CEF:

$$\pi^*(x) = \mathbb{1}[\mathbb{E}[Y(1) - Y(0) \mid X = x] > 0] = \mathbb{1}[\tau(x) > 0] \quad (1)$$

Targeting CATE does not provide best policies

Manski (2004) calls the sample analogue of this rule the **CONDITIONAL EMPIRICAL SUCCESS** (CES) rule: $\hat{\pi}(x) = \mathbb{1}[\hat{\tau}(x) > 0]$

The CES rule **seems to settle policy learning** as we have learned how to flexibly estimate CATEs/IATEs

BUT this intuition is at least partly misleading when it comes to estimation

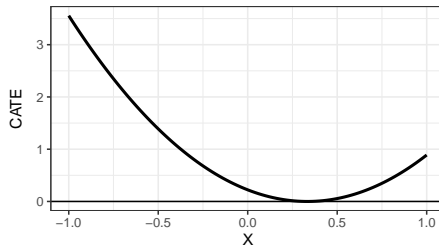
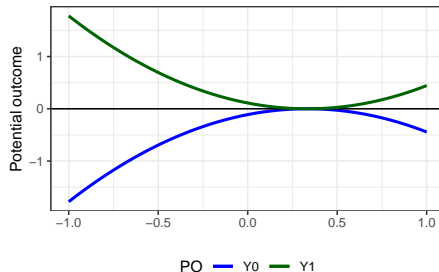
CATE estimation aims to minimize $\mathbb{E}[(\hat{\tau}(X) - \tau(X))^2] \Rightarrow$ approximate CEF of CATE as good as possible

BUT lower CATE MSE does not necessarily improve policy rules as Qian & Murphy (2011) powerfully demonstrate \Rightarrow next slides

Toy example: DGP

Qian & Murphy (2011) consider the following DGP:

- $X \sim \text{uniform}(-1, 1)$
 - $Y(1) = (X - 1/3)^2$
 - $Y(0) = -(X - 1/3)^2$
- $\Rightarrow \tau(X) = 2(X - 1/3)^2$



Toy example: oracle

We estimate CATEs with the **correct (quadratic) model**

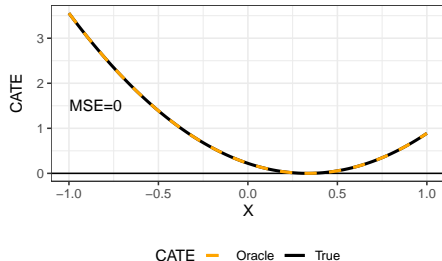
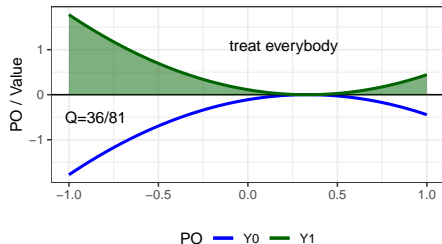
$$\Rightarrow \hat{\tau}^*(X) = \tau(X)$$

$$\Rightarrow \text{MSE}(\hat{\tau}^*(X)) = 0$$

\Rightarrow All estimated CATEs non-negative

\Rightarrow Treat everybody: $\pi^*(X) = 1$

\Rightarrow Highest possible value function achieved: $Q(\pi^*) = 36/81$ (green area)



Toy example: linear fit

We estimate CATEs with a misspecified linear model

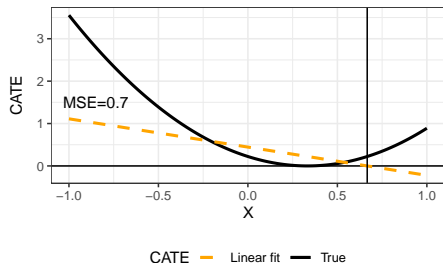
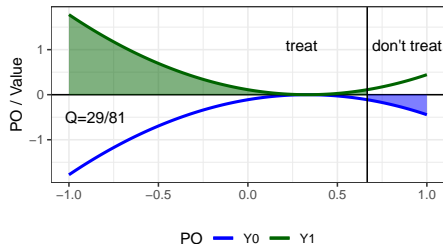
$$\Rightarrow \hat{\tau}^{lin}(X) = 4/9 - 2/3X$$

$$\Rightarrow \text{MSE}(\hat{\tau}^{lin}(X)) = 0.7$$

\Rightarrow Some CATEs erroneously negative

$$\Rightarrow \pi^{lin}(X) = \mathbb{1}[X < 2/3]$$

\Rightarrow Lower value function: $Q(\pi^{lin}) = 25/81$
(green + blue area)



Toy example: ATE

We estimate CATEs with a misspecified constant model (ATE)

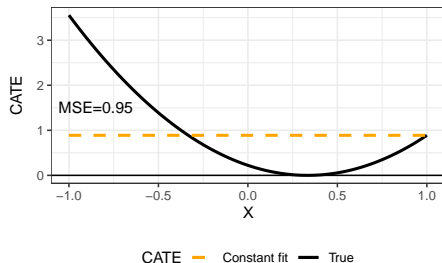
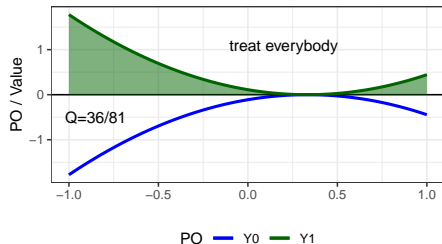
$$\Rightarrow \hat{\tau}^{ate}(X) = 8/9$$

$$\Rightarrow \text{MSE}(\hat{\tau}^{ate}(X)) = 0.95$$

\Rightarrow ATE positive

\Rightarrow Treat everybody: $\pi^{ate}(X) = 1$

\Rightarrow Highest possible value function achieved: $Q(\pi^{ate}) = 36/81$



Lesson learned

We would intuitively expect that the better we estimate heterogeneous effects, the better our resulting policy using the CES rule (at least I did)

However, lower MSE of CATE does not imply higher value function:

$$MSE(\hat{\tau}^{lin}(X)) < MSE(\hat{\tau}^{ate}(X)) \text{ but } Q(\pi^{lin}) < Q(\pi^{ate})$$

Methods that minimize MSE of CATEs focus on this only, but do not care about any downstream policy learning they might be used for

Remark: Note the resemblance to our discussion why MSE minimization in treated and controls separately is not the best strategy to minimize CATE MSE

A different objective function

For CATE estimation we $\min \mathbb{E}[(\hat{\tau}(X) - \tau(X))^2]$

Instead, the **objective function of policy learning** is to max the value function

$$\pi^* = \arg \max_{\pi} \mathbb{E}[Y(\pi(X))] = \arg \max_{\pi} Q(\pi) \quad (2)$$

or to minimize regret (difference to optimal policy)

$$\pi^* = \arg \min_{\pi} \underbrace{Q(\pi^*) - Q(\pi)}_{R(\pi)} \quad (3)$$

both are equivalent as $Q(\pi^*)$ is just a constant

A practical look at the objective function

The objective function can be represented in many different forms

Especially one has proven very **useful if we want to use ML for policy learning**

We **center the value function** around a benchmark policy that assigns treatments via a fair coin flip (50-50 chance of being treated, $\pi^{coin} \sim \text{Bernoulli}(0.5)$)

$$\pi^* = \arg \max_{\pi} Q(\pi) = \arg \max_{\pi} Q(\pi) - \underbrace{[0.5 \mathbb{E}[Y(1)] + 0.5 \mathbb{E}[Y(0)]]}_{Q(\pi^{coin})} \quad (4)$$

$$= \arg \max_{\pi} \underbrace{\mathbb{E}[|\tau(X)| \text{ sign}(\tau(X)) (2\pi(X) - 1)]}_{A(\pi)} \quad (5)$$

where $(2\pi(X) - 1) \in \{-1, 1\}$ is one if policy assigns treatment and minus one if not

Rewrite objective function

Suppressing dependence of π on X , we write

$$\begin{aligned}\pi^* &= \arg \max_{\pi} Q(\pi) = \arg \max_{\pi} \mathbb{E}[Y(\pi)] = \arg \max_{\pi} \mathbb{E}[Y(\pi)] - [0.5 \mathbb{E}[Y(1)] + 0.5 \mathbb{E}[Y(0)]] \\ &= \arg \max_{\pi} \mathbb{E}[\pi Y(1) + (1 - \pi)Y(0)] - 0.5 \mathbb{E}[Y(1)] - 0.5 \mathbb{E}[Y(0)] \\ &= \arg \max_{\pi} 2 \mathbb{E}[(\pi - 0.5)(Y(1) - Y(0))] \\ &= \arg \max_{\pi} \mathbb{E}[(2\pi - 1)(Y(1) - Y(0))] \\ &\stackrel{LIE}{=} \arg \max_{\pi} \mathbb{E}[(2\pi - 1)\tau(X)] \\ &= \arg \max_{\pi} \mathbb{E}[|\tau(X)| \operatorname{sign}(\tau(X)) (2\pi(X) - 1)]\end{aligned}$$

b/c all manipulations leave the maximum unchanged and $Y(\pi) = \pi Y(1) + (1 - \pi)Y(0)$

Rewrite objective function (bonus)

Note that we could also cast it as minimization problem

$$\begin{aligned}\pi^* &= \arg \max_{\pi} \mathbb{E}[|\tau| \operatorname{sign}(\tau) (2\pi - 1)] \\&= \arg \max_{\pi} \mathbb{E}[|\tau| (\mathbb{1}[\tau > 0] - \mathbb{1}[\tau < 0]) (2\pi - 1)] \\&= \arg \max_{\pi} \mathbb{E}[|\tau| (2\mathbb{1}[\tau > 0] - 1) (2\pi - 1)] \\&= \arg \max_{\pi} \mathbb{E}[|\tau| (4\mathbb{1}[\tau > 0]\pi - 2\mathbb{1}[\tau > 0] - 2\pi)] \\&= \arg \max_{\pi} (-2) \mathbb{E}[|\tau| (\mathbb{1}[\tau > 0]^2 - 2\mathbb{1}[\tau > 0]\pi + \pi^2)] \\&= \arg \min_{\pi} \mathbb{E}[|\tau| (\mathbb{1}[\tau > 0] - \pi)^2]\end{aligned}$$

where we use that $\mathbb{1}[\tau > 0] = \mathbb{1}[\tau > 0]^2$ and $\pi = \pi^2$

⇒ This objective function is zero if the policy always coincides with the indicator that CATE is positive

"Intuition"

$A(\pi) = \mathbb{E}[|\tau(X)| \text{sign}(\tau(X)) (2\pi(X) - 1)]$ measures the **advantage** of a policy compared to random allocation

This helps us to understand **what drives our policy learning objective**:

- If $\text{sign}(\tau(X))(2\pi(X) - 1) = 1$, i.e. if the policy picks the better treatment for X , we **earn the absolute value of the CATE**
- If $\text{sign}(\tau(X))(2\pi(X) - 1) = -1$, i.e. if the policy picks the worse treatment for X , we **lose the absolute value of the CATE**

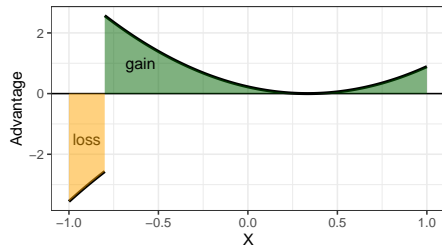
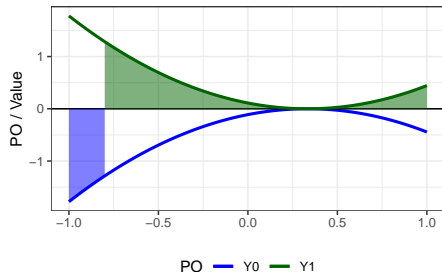
⇒ We need to **get it right for those with biggest CATEs**, those with CATEs close to zero are negligible

⇒ This shows the **difference to CATE MSE minimization**, where we need to find good approximations everywhere

Example (1/2)

Consider $\pi(X) = \mathbb{1}[X > -0.8]$

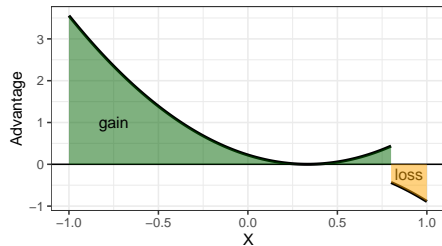
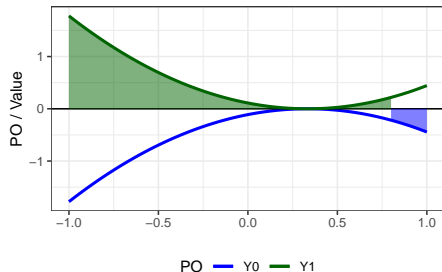
\Rightarrow We misclassify 10% and lose a lot of value because we misclassify those with highest CATE



Example (2/2)

Consider $\pi(X) = \mathbb{1}[X < 0.8]$

\Rightarrow We (again) misclassify 10% but lose not so much because they would have benefited not too much anyways



Offline policy learning with binary treatments

Identification as usual

Until now we operated in a world with known PO or at least known CATE functions

In reality we need to first identify them

Identification follows like in previous lectures from randomization or from strong ignorability in observational studies where we assume that X contains all confounders

This means, e.g, that the optimal policy is identified as

$$\pi^* = \arg \max_{\pi} \mathbb{E}[Y(\pi(X))] \quad (6)$$

$$\stackrel{LIE}{=} \arg \max_{\pi} \mathbb{E}[\mathbb{E}[Y(\pi(X)) \mid X]] \quad (7)$$

$$= \arg \max_{\pi} \mathbb{E}[\mathbb{E}[Y \mid W = \pi(X), X]] \quad (8)$$

Another question that we did not touch so far was about the **permissible policies**

There are good reasons to think about this issue more carefully:

- **Practical:** In practice we often need policy rules that are **easy to communicate** and that respect **fairness or budget constraints**
- **Statistical:** Stoye (2009, 2012) notes that the regret of unrestricted learned policies can become larger than from random assignment

⇒ The literature agrees so far that we should aim to find the best policy within a **predefined class of policies** Π

Estimation (1/2)

Equation (4) suggests that policy learning boils down to a **weighted classification problem**

We want to **classify the sign of the CATE** while favouring correct classifications with larger absolute CATEs

Only problem is that **we do not know the CATE**

Athey and Wager (2021) recommend to use an old buddy for estimation, the **pseudo-outcome** $\tilde{Y}_{ATE} = \hat{m}(1, X) - \hat{m}(0, X) + \frac{W(Y - \hat{m}(1, X))}{\hat{e}(X)} - \frac{(1-W)(Y - \hat{m}(0, X))}{1 - \hat{e}(X)}$

⇒ We need to estimate the nuisance parameters or reuse \tilde{Y}_{ATE} of previous step

The resulting **weighted classification problem**

$$\hat{\pi} = \arg \max_{\pi \in \Pi} \left\{ \frac{1}{N} \sum_{i=1}^N |\tilde{Y}_{i,ATE}| \operatorname{sign}(\tilde{Y}_{i,ATE}) (2\pi(X_i) - 1) \right\} \quad (9)$$

can be solved by any method that

- classifies a binary outcome/response: $\operatorname{sign}(\tilde{Y}_{ATE})$...
- ... with weights $|\tilde{Y}_{ATE}|$...
- ... using covariates/predictors X

This is a **standard problem in supervised ML** (usually the second part next to regression)

Potential candidates are decision trees/forests, Logistic Lasso, Support Vector Machines, ...

Athey and Wager (2021) show that this procedure **minimizes regret faster** as the sample grows than **alternative procedures** that would only use outcome regression or inverse probability weighting

This requires the **familiar conditions** on the estimation of the nuisance parameters in \tilde{Y}_{ATE} as previously (high-quality, cross-fitting)

Again this is the result of the Neyman-orthogonality property that was already responsible for the nice properties of estimators based on \tilde{Y}_{ATE} previously

Simulation notebook: Offline policy learning

Application notebook: Offline policy learning

Offline policy learning with multiple treatments

Beyond binary treatments

In many settings, we have not only one treatment option, but multiple

Multiple treatment notation:

$$W \in \{0, \dots, T\}$$

$$Y(w)$$

$$Y = \sum_{w=0}^T \mathbb{1}[W = w]Y(w)$$

$$D(w) = \mathbb{1}[W = w]$$

$$e_w(x) = P[W = w \mid X = x]$$

$$\pi(x) \in \{0, \dots, T\}$$

$$Y(\pi(X))$$

$$Q(\pi) = \mathbb{E}[Y(\pi(X))]$$

Multiple treatment

PO under treatment w

Observed outcome

Indicator for being in treatment w

Probability of receiving w

Policy rule

PO under policy $\pi(X)$

Value function

A harder problem

Again we want to find the **value maximizing policy**: $\pi^* = \arg \max_{\pi} Q(\pi)$

However, with more than two options the **trick of classifying signs of CATEs fails** because it is now ambiguous what's the alternative treatment

Instead, we really have to find the maximum of $Q(\pi)$

Again we can use **ALPW scores** to make progress

Remember that the Conditional Average Potential Outcome is identified as

$$\mathbb{E}[Y(w) \mid X = x] = \mathbb{E} \left[\underbrace{m(w, x) + \frac{D(w)(Y - m(w, x))}{e_w(x)}}_{\equiv \Gamma(w, x)} \mid X = x \right] \quad (10)$$

Identification of optimal policy

The optimal policy under multiple treatments is identified as

$$\pi^* = \arg \max_{\pi} Q(\pi) = \arg \max_{\pi} \mathbb{E}[Y(\pi(X))] = \arg \max_{\pi} \mathbb{E} \left[\sum_{w=0}^T \mathbb{1}[\pi(X) = w] Y(w) \right] \quad (11)$$

$$\stackrel{LIE}{=} \arg \max_{\pi} \mathbb{E} \left[\sum_{w=0}^T \mathbb{E}[\mathbb{1}[\pi(X) = w] Y(w) \mid X] \right] \quad (12)$$

$$= \arg \max_{\pi} \mathbb{E} \left[\sum_{w=0}^T \mathbb{1}[\pi(X) = w] \mathbb{E}[Y(w) \mid X] \right] \quad (13)$$

$$= \arg \max_{\pi} \mathbb{E} \left[\sum_{w=0}^T \mathbb{1}[\pi(X) = w] \Gamma(w, X) \right] \quad (14)$$

Estimation of optimal policy (1/2)

Equation (14) motivates the estimation procedure of Zhou, Athey & Wager (2018):

Using estimated nuisance parameters as always, we calculate the observation and treatment specific AIPW score

$$\hat{\Gamma}_{i,w} = \hat{m}(w, X) + \frac{\mathbb{1}(W = w)(Y - \hat{m}(w, X))}{\hat{e}_w(X)}$$

and store those in a $N \times T + 1$ matrix

$$\hat{\mathbf{\Gamma}} = \begin{bmatrix} \hat{\Gamma}_{1,0} & \cdots & \hat{\Gamma}_{1,T} \\ \vdots & \ddots & \vdots \\ \hat{\Gamma}_{N,0} & \cdots & \hat{\Gamma}_{N,T} \end{bmatrix}$$

Estimation of optimal policy (2/2)

The empirical optimization problem that we need to solve is

$$\hat{\pi} = \arg \max_{\pi \in \Pi} \left\{ \frac{1}{N} \sum_{i=1}^N \sum_{w=0}^T \mathbb{1}[\pi(X) = w] \hat{r}_{i,w} \right\}$$

This looks and is ugly as this is a so-called **non-convex optimization problem**

⇒ There is no fast/direct way of computing the estimated optimal policy

We basically need to check the value function of all possible policies

This can be a computational nightmare

Example (1/2)

Example: $W \in \{0, 1, 2\}$ with two candidate rules $\pi^1(X)$ and $\pi^2(X)$

$$\hat{\mathbf{r}} = \begin{bmatrix} \hat{r}_{1,0} & \hat{r}_{1,1} & \hat{r}_{1,2} \\ \hat{r}_{2,0} & \hat{r}_{2,1} & \hat{r}_{2,2} \\ \vdots & \vdots & \vdots \\ \hat{r}_{N,0} & \hat{r}_{N,1} & \hat{r}_{N,2} \end{bmatrix}; \pi^1(X) = \begin{bmatrix} 2 \\ 0 \\ \vdots \\ 1 \end{bmatrix}; \hat{Q}(\pi^1) = \begin{bmatrix} \hat{r}_{1,2} \\ \hat{r}_{2,0} \\ \vdots \\ \hat{r}_{N,1} \end{bmatrix}$$

Mean of $\hat{Q}(\pi^1)$ estimates the expected outcome under rule $\pi^1(X)$

Example (2/2)

Example: $W \in 0, 1, 2$ with two candidate rules $\pi^1(X)$ and $\pi^2(X)$

$$\hat{\mathbf{r}} = \begin{bmatrix} \hat{r}_{1,0} & \hat{r}_{1,1} & \hat{r}_{1,2} \\ \hat{r}_{2,0} & \hat{r}_{2,1} & \hat{r}_{2,2} \\ \vdots & \vdots & \vdots \\ \hat{r}_{N,0} & \hat{r}_{N,1} & \hat{r}_{N,2} \end{bmatrix}; \pi^2(X) = \begin{bmatrix} 2 \\ 2 \\ \vdots \\ 0 \end{bmatrix}; \hat{Q}(\pi^2) = \begin{bmatrix} \hat{r}_{1,2} \\ \hat{r}_{2,2} \\ \vdots \\ \hat{r}_{N,0} \end{bmatrix}$$

Mean of $\hat{Q}(\pi^2)$ estimates the expected outcome under rule $\pi^2(X)$

Compare $\hat{Q}(\pi^1)$ and $\hat{Q}(\pi^2)$ and choose policy with larger value

Optimal decision tree

Zhou, Athey & Wager (2018) propose to focus on decision trees as class of possible policies

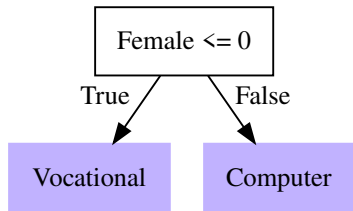
However, greedy splitting makes little sense in the multiple treatment setting

Zhou et al. (2018) provide a clever algorithm that searches through all possible splits for a fixed depth

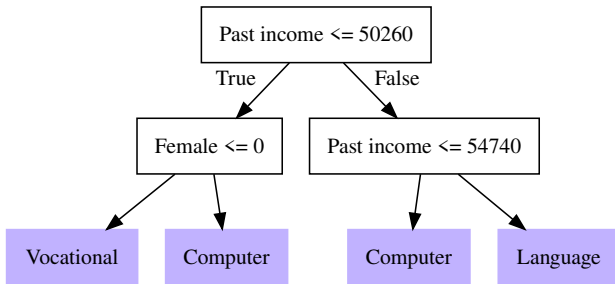
But still it can take ages to run

It is implemented in the R package `policytree`

Example



(a) Depth 1



(b) Depth 2

Source: [Knaus \(2022\)](#) check [replication notebook](#) if you want to see the full pipeline from ATEs to GATEs to IATEs to policy learning with multiple treatments

Policy learning is arguably the most valuable part of Causal ML

Previously we needed to derive policy recommendations based on the results of evaluation results in a more or less principled way

Policy learning promises at least to directly target the objective function of the policy maker and to optimize it in a data-driven way

Applications in social sciences are still in their infancies

⇒ Very exciting field

Loads of open questions

- How to properly incorporate budget or capacity constraints in the algorithm?
- How to tune the policy learner?
- Statistical inference?
- What is the outcome I really would like to optimize?
- Which variables are allowed? #Fairness
- Is assignment based on CATEs really so bad?
- ...

First steps in the applied literature, but even less experience than with the new evaluation methods

Online policy learning (bandits)

A dynamic setting

Our policy learning lived **so far** in the familiar world where we evaluate **historic data** and derive policy recommendations for the future (offline policy learning)

So-called **online policy learning** lives in a different setting

Instead of learning policies after treatment assignments took place, **we learn what works while assigning treatments**

This involves **algorithms** that are able to **actively interact/experiment with their environment**

For me this is the first step towards something that **might deserve the AI label** at some point

Tech companies vs. social science

Amazon, Facebook, Google, Spotify, ... use online PL to figure out which ads, playlists or other features of the user experience **maximize their returns**

Several features of their setting are **excellent for online learning**:

- Millions of users per day
- Real time data
- Real time outcomes (did the user click or not?)

These features are **rarely present in social sciences** and the literature of adapting the methods to productive use for us is in the beginning

Still (or for this reason) I would like to give you the idea behind these methods

Multi-armed bandits

Multiple treatments $W \in \{0, \dots, T\}$ (same framework as before)

But **units** $i = 0, \dots, N$ **arrive sequentially** to be assigned to treatment

Question: Which treatment to assign to unit $i + 1$? (no covariates for now)

Optimal treatment would be to assign those with **largest average potential outcome** $\mu_w = \mathbb{E}[Y(w)] \Rightarrow \pi_{i+1}^* = \arg \max_w \mathbb{E}[Y(w)] = \arg \max_w \mu_w$

Goal: **Minimize regret** for all units $\min \frac{1}{N} \sum_i (\underbrace{\mathbb{E}[Y_i(\pi_i^*)]}_{\text{optimal}} - \underbrace{\mathbb{E}[Y_i(W_i)]}_{\text{actual}})$

Remark: The name is derived from the problem gamblers face when deciding which of the one-armed bandits in the casino to play (**nice cartoon explanation**)

Two conflicting goals

We start out with randomly assigning each treatment

But if every treatment was assigned at least twice (i.e. we can estimate a variance), we have the chance to **balance two conflicting dimensions**:

- **Exploration:** we can assign units to treatments that we are uncertain about
⇒ use unit $i + 1$ to **explore what works best** (reduce uncertainty)
- **Exploitation:** we can likely decrease regret by assigning unit $i + 1$ to the treatment that is currently viewed as the best ⇒ **exploit what we know so far**

Two common strategies

Denote by $\hat{\mu}_{i,w}$ the estimated average PO using all units until i , by $\hat{\sigma}_{i,w}^2$ the variance of this estimate and by $\alpha > 0$ an appropriately chosen constant

UPPER CONFIDENCE BOUND (UCB) method:

$$W_{i+1} = \arg \max_w (\hat{\mu}_{i,w} + \alpha \hat{\sigma}_{i,w}) \quad (15)$$

⇒ choose the treatment with highest upper confidence bound for critical value α

Thompson sampling:

Draw for each treatment from a normal distribution $\tilde{\mu}_{i,w} \sim N(\hat{\mu}_{i,w}, \alpha^2 \hat{\sigma}_{i,w}^2)$

$$W_{i+1} = \arg \max_w \tilde{\mu}_{i,w} \quad (16)$$

Some comments

As the number of units increases and thus variance decreases, both methods assign more often the good treatments and ignore the bad ones

Assignment of treatments is not related to potential outcomes \Rightarrow no identification issues

Larger α means more exploration and less exploitation (lower risk of missing a good treatment by chance)

Contextual bandits use "context" information X and the CAPO instead of average PO for assignment (see e.g. Dimakopoulou et al., 2019)

Many open questions regarding implementation in social science context (first projects started) \Rightarrow another exciting area for research

Simulation notebook: Multi-armed bandit

Further material on bandits

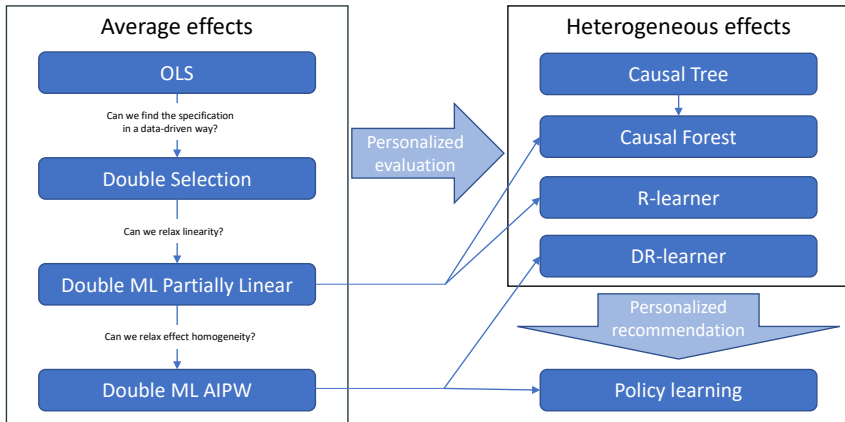
Introduction videos: [Multi-armed bandits](#), [UCB method](#) and [Thompson sampling](#)

More technical treatments: [Max Kasy teaching slides](#), "[Introduction to Multi-Armed Bandits](#)" by Aleksandrs Slivkins

Applications: [Covid testing](#) (Bastania et al., 2021), [refugee employment assistance](#) (Caria et al., 2021), [charitable giving](#) (Athey et al., 2022)

[Shiny app](#) to play with

End of the journey



Congrats! 🎉

Wrapping-up

Lessons learned (on a high level)

- Naive integration of ML in causal analysis can go wrong
 - We have to think about our target parameter
 - We should target the underlying objective function, e.g. #CATEvsPL
 - If we know what we want to do, we can leverage the power of ML ...
 - ... to make our lives easier (outsource what the machine can do better)
 - ... to get more out of the same data #nonparametricGATE #policylearning
- ⇒ More time to think about the important stuff (identification, interpretation, ...)
- ⇒ More fun in the estimation part (tuning a ML models is much more fun than specifying models yourself + (arguably) better science, at least for me)

BUT also no silver bullet: Methods help with estimation, not with identification

Key concepts

We have learned several key concepts for Causal ML:

- **Nuisance parameters**: Parameters (so far CEFs of observed variables) that are not of interest *per se* but help us to get our hands on the target parameter
- **Neyman-orthogonal scores**: Smart combination of **multiple** nuisance parameters such that these can be estimated using supervised ML (if high-quality and cross-fit)
- **Pseudo-outcomes**: May be used as outcomes in standard regressions to model/validate inherently unobservable causal quantities
- **Modified splitting criteria**: Teach regression trees and forests to model inherently unobservable causal quantities

That's it from my side

I hope this course was a nice add-on to the more classic econometrics courses

We are still at the beginning of understanding the fruitful integration of ML into economics/econometrics

However, the concepts you learned in this course should enable you to digest future developments in causal ML for policy evaluation/recommendation

The literature is currently exploding, so there are a lot of smaller and bigger topics to work on

More free resources

Collections:

- Collection "Machine Learning for Economists" by Dario Sansone
- Collection "Dive into Causal Machine Learning" of Alexander Quispe
- "Must-read recent papers and resources on Causal \cap ML"

Single resources (loosely ordered from introductory to advanced):

- "ML & Causal Inference: A Short Course" by Athey, Spiess and Wager
- grf package documentation
- DoubleML user guide
- "ML-based causal inference tutorial" by Golub Capital Social Impact Lab
- "Causal Inference for the Brave and True" Part II by Matheus Facure Alves
- Lecture notes of Stefan Wager
- Lecture notes of Christophe Gaillac and Jeremy L'Hour

Ceterum censeo a fancy method alone is not a credible
identification strategy
⇒ separate identification and estimation