EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

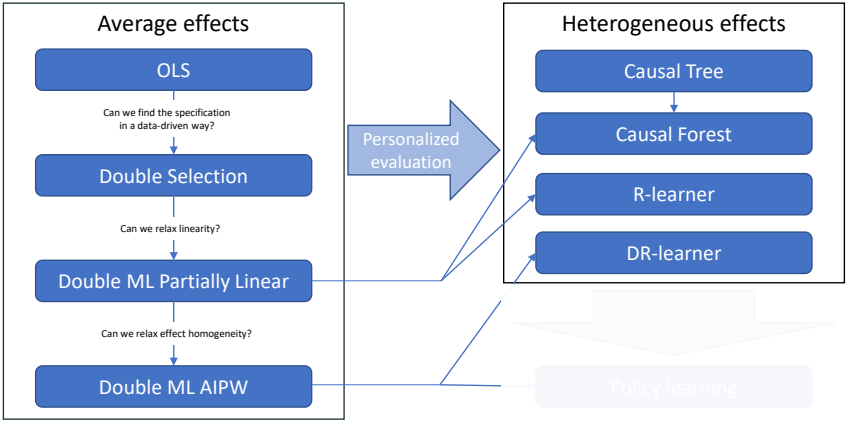
Causal Machine Learning

Policy learning

Michael Knaus

WiSe 25/26

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

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)?"

$\Rightarrow$ 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)
$\gamma_w(x) := \mathbb{E}[Y(w) \mid X = x]$	Conditional Average PO
$\tau(x) := \gamma_1(x) - \gamma_0(x)$	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 (average PO under policy $\pi(X)$)

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

Example: Policy

Consider a setting with a univariate $X \sim uniform(-1, 1)$

Then candidate/potential policies could be:

- $\pi^a(X) = \mathbb{1}[X > 1/2]$
- $\pi^b(X) = \mathbb{1}[X < -1/2]$
- $\pi^c(X) = \mathbb{1}[X < -1/2] + \mathbb{1}[X > 1/2]$

The PO under any policy $\pi(X)$ is then $Y(\pi(X)) = \pi(X)Y(1) + (1 - \pi(X))Y(0)$

Note that this has the flavor of consistency but with a hypothetical, not the real treatment

$\Rightarrow$ We stay in the PO dream world

Example: Value function

Consider the case of two 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 $Q(\pi^2) = \mathbb{E}[Y(\pi^2)]$

The value function asks "What would be the APO if we would have implemented the policy?"

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:

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

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

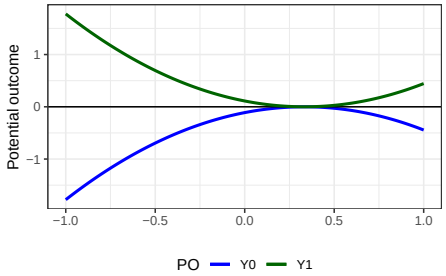
BUT lower CATE MSE does not necessarily improve policy rules as Qian & Murphy (2011) powerfully demonstrate $\Rightarrow$ next slides

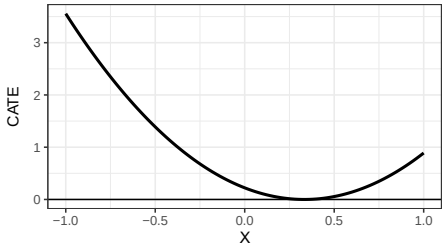
Toy example: DGP

Qian & Murphy (2011) consider the following DGP:

- $X \sim uniform(-1, 1)$
- $Y(1) = (X - 1/3)^2$
- $Y(0) = -(X - 1/3)^2$

$\Rightarrow \tau(X) = 2(X - 1/3)^2$





Toy example: oracle

We estimate CATEs with the correct (quadratic) model

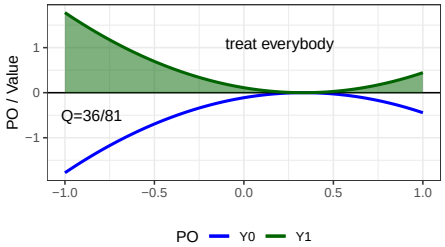
$\Rightarrow \hat{\tau}^*(X) = \tau(X)$

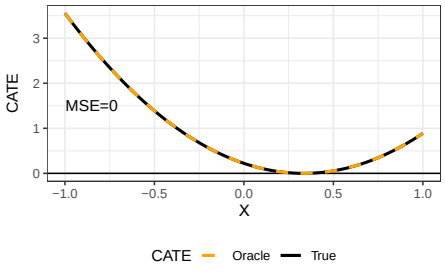
$\Rightarrow MSE(\hat{\tau}^*(X)) = 0$

$\Rightarrow$ 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

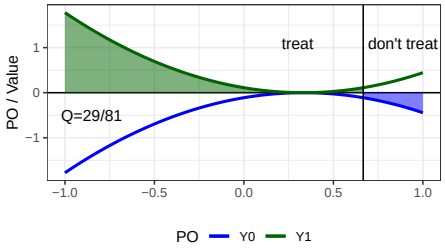
$\Rightarrow \hat{\tau}^{lin}(X) = 4/9 - 2/3X$

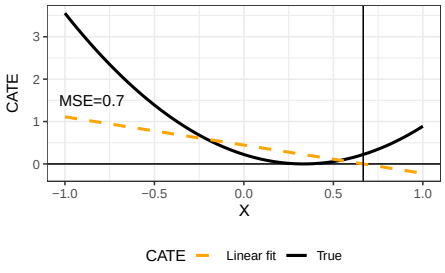
$\Rightarrow MSE(\hat{\tau}^{lin}(X)) = 0.7$

$\Rightarrow$ Some CATEs erroneously negative

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

$\Rightarrow$ Lower value function: $Q(\pi^{lin}) = 29/81$ (green + blue area)

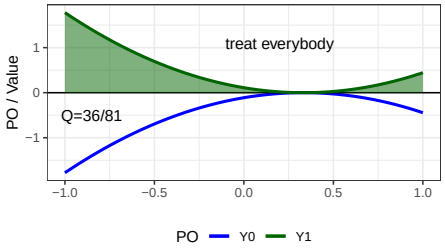


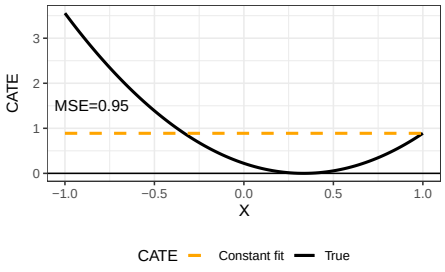


Toy example: ATE

We estimate CATEs with a misspecified constant model (ATE)

- $\Rightarrow \hat{\tau}^{ate}(X) = 8/9$
- $\Rightarrow MSE(\hat{\tau}^{ate}(X)) = 0.95$
- $\Rightarrow$ 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:

$MSE(\hat{\tau}^{lin}(X)) < MSE(\hat{\tau}^{ate}(X))$ 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

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

and the optimal value is $Q^* := Q(\pi^*)$

An equivalent way is to minimize regret (difference to optimal value)

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

Note that both are equivalent as Q^* 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, $\pi^{coin} \sim Bernoulli(0.5)$)

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

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

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

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

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

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

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

$\Rightarrow$ This objective function is zero if the policy always coincides with the indicator that CATE is positive

"Intuition"

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

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

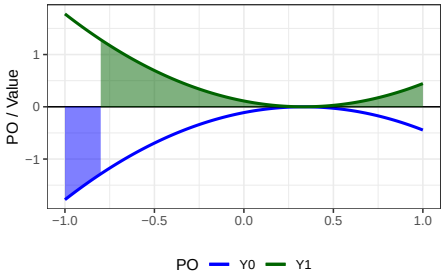
- If $sign(\tau(X))(2\pi(X) - 1) = 1$, i.e. if the policy picks the better treatment for X, we earn the absolute value of the CATE
- If $sign(\tau(X))(2\pi(X) - 1) = -1$, i.e. if the policy picks the worse treatment for X, we lose the absolute value of the CATE

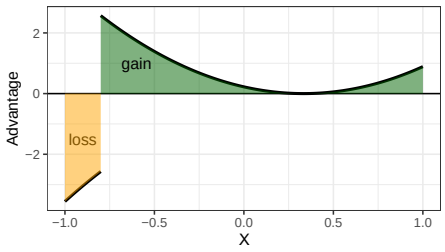
$\Rightarrow$ We need to get it right for those with biggest CATEs, those with CATEs close to zero are negligible

$\Rightarrow$ This shows the difference to CATE MSE minimization, where we need to find good approximations everywhere

Example (1/2)

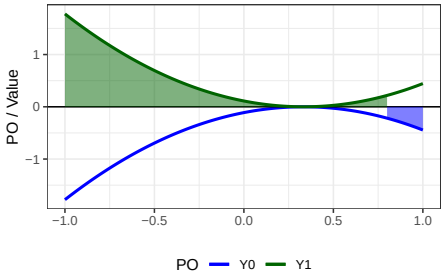
Consider $\pi(X) = \mathbb{1}[X > -0.8]$
$\Rightarrow$ We misclassify 10% and lose a lot of value because we misclassify those with highest CATE

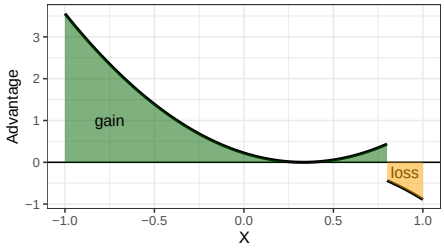




Example (2/2)

Consider $\pi(X) = \mathbb{1}[X < 0.8]$
$\Rightarrow$ We (again) misclassify 10% but lose less 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 the value function to be able to optimize it

Identification follows like in previous lectures from randomization or from strong ignorability assuming that X is a valid adjustment set and common support

Following the standard recipe, the value function is identified as

$$\begin{aligned} Q(\pi) &= \mathbb{E}[Y(\pi(X))] = \mathbb{E}[\pi(X)Y(1) + (1-\pi(X))Y(0)] \\ &= \mathbb{E}[\pi(X)m(1,X) + (1-\pi(X))m(0,X)] \end{aligned}$$

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

$$\begin{aligned} \pi^* &= \arg\max_{\pi} \mathbb{E}[Y(\pi(X))] \\ &= \arg\max_{\pi} \mathbb{E}[\pi(X)m(1,X) + (1-\pi(X))m(0,X)] \end{aligned}$$

Identification of value function

$$
\begin{aligned}
Q(\pi) &= \mathbb{E}[Y(\pi(X))] \\
&= \mathbb{E}[\pi(X)Y(1) + (1-\pi(X))Y(0)] \\
&\overset{LIE}{=} \mathbb{E}[\mathbb{E}[\pi(X)Y(1) + (1-\pi(X))Y(0) \mid X]] \\
&= \mathbb{E}[\pi(X)\,\mathbb{E}[Y(1) \mid X] + (1-\pi(X))\,\mathbb{E}[Y(0) \mid X]] \\
&= \mathbb{E}[\pi(X)\,\mathbb{E}[Y(1) \mid W=1, X] + (1-\pi(X))\,\mathbb{E}[Y(0) \mid W=0, X]] \\
&\overset{Cons}{=} \mathbb{E}[\pi(X)\,\mathbb{E}[Y \mid W=1, X] + (1-\pi(X))\,\mathbb{E}[Y \mid W=0, X]] \\
&= \mathbb{E}[\pi(X)m(1,X) + (1-\pi(X))m(0,X)]
\end{aligned}
$$

Candidate policies

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

$\Rightarrow$ 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 favoring correct classifications with larger absolute CATEs

Only problem is that we do not know the CATE

Athey and Wager (2021) use an old buddy for estimation, the pseudo-outcome

$$\tilde{Y}_{ATE} = \hat{m}(1,X) - \hat{m}(0,X) + \frac{W(Y - \hat{m}(1,X))}{\hat{e}(X)} - \frac{(1-W)(Y - \hat{m}(0,X))}{1 - \hat{e}(X)}$$

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

Estimation

The resulting weighted classification problem

$$\hat{\pi} = \underset{\pi \in \Pi}{\arg\max} \left\{ \frac{1}{N} \sum_{i=1}^{N} \underbrace{|\tilde{Y}_{i,ATE}|}_{\text{weight}} \underbrace{sign(\tilde{Y}_{i,ATE})}_{\text{to be classified}} \overbrace{(2\pi(X_i) - 1)}^{\text{function to be learned}} \right\} \tag{6}$$

can be solved by any method that

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

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

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

Properties

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 $\tilde{Y}_{ATE}$ (high-quality, cross-fitted)

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:

$W \in \{0, ..., T\}$	Multiple treatment
$Y(w)$	PO under treatment w
$Y = \sum_{w=0}^{T} \mathbb{1}[W = w]Y(w)$	Observed outcome
$D(w) = \mathbb{1}[W = w]$	Indicator for being in treatment w
$e_w(x) = P[W = w \mid X = x]$	Probability of receiving w
$\pi(x) \in \{0, ..., T\}$	Policy rule
$Y(\pi(X))$	PO under policy π(X)
$Q(\pi) = \mathbb{E}[Y(\pi(X))]$	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 AIPW scores to make progress

Remember that the Conditional Average Potential Outcome is identified as

$$\mathbb{E}[Y(w) \mid X = x] = \mathbb{E}\Bigg[\underbrace{m(w,x) + \frac{D(w)(Y - m(w,x))}{e_w(x)}}_{=: \Gamma_w} \Bigg| X = x\Bigg] \tag{7}$$

Identification of optimal policy

The optimal policy under multiple treatments is identified as

$$
\begin{aligned}
\pi^* &= \arg\max_{\pi} Q(\pi) = \arg\max_{\pi} \mathbb{E}[Y(\pi(X))] = \arg\max_{\pi} \mathbb{E}\left[\sum_{w=0}^{T} \mathbb{1}[\pi(X) = w] Y(w)\right] \\
&\overset{LIE}{=} \arg\max_{\pi} \mathbb{E}\left[\sum_{w=0}^{T} \mathbb{E}[\mathbb{1}[\pi(X) = w] Y(w) \mid X]\right] \\
&= \arg\max_{\pi} \mathbb{E}\left[\sum_{w=0}^{T} \mathbb{1}[\pi(X) = w]\, \mathbb{E}[Y(w) \mid X]\right] \\
&= \arg\max_{\pi} \mathbb{E}\left[\sum_{w=0}^{T} \mathbb{1}[\pi(X) = w]\, \mathbb{E}[\Gamma_w \mid X]\right] \qquad (8)
\end{aligned}
$$

Estimation of optimal policy (1/2)

Equation (8) motivates the estimation procedure of Zhou, Athey & Wager (2023):

Using estimated nuisance parameters as always, we calculate the observation and treatment specific AIPW pseudo-outcomes

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

and store them 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

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

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

$\Rightarrow$ 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 $\pi^2(X)$

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

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 $\pi^2(X)$

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

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 et al. (2023) propose to focus on decision trees as class of possible policies

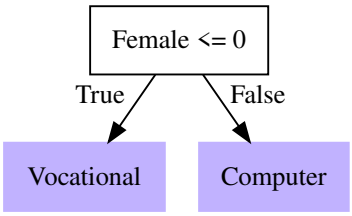
However, greedy splitting makes little sense in the multiple treatment setting

Zhou et al. (2023) provide a clever algorithm that searches through all possible splits for a fixed depth

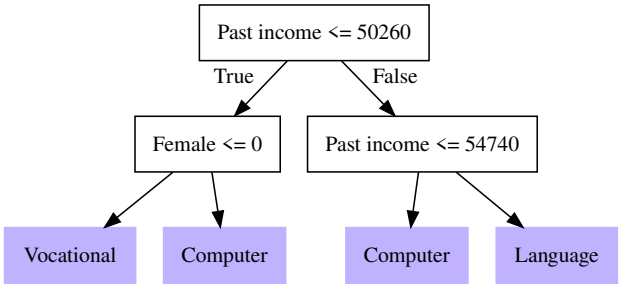
But still it can take ages to run

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

Discussion policy learning

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

See e.g. Ahrens et al. (2024) for some nice implementation

Loads of open questions

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

Ceterum censeo a fancy method alone is not a credible identification strategy

$\Rightarrow$ separate identification and estimation