EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

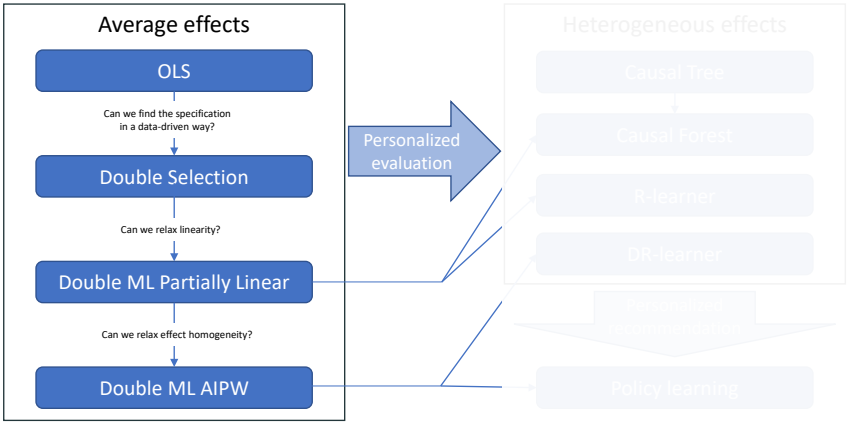
Causal Machine Learning

Predicting effects

Michael Knaus

March 13, 2023

Today we move away from average effects



Plan of this morning

1. The goal
2. The challenge
3. Causal trees/forests
4. Meta-learner

The goal

Effect heterogeneity

So far we focused either on constant effects or allowed for heterogeneous effects in broad groups like ATE, ATT or LATE

Such aggregate effect measures are the starting point of most analyses as they provide an excellent summary of the effectiveness of the treatment

$\Rightarrow$ Causal ML provided flexible tools for standard problems

However, often we want to go beyond average effects and

- estimate the effect in pre-specified subgroups (already standard)
- flexibly predict heterogeneous effects (only recently via Causal ML)

$\Rightarrow$ Causal ML can increase the scope of our analyses

Why is this interesting?

More comprehensive evaluation: who wins or loses and by how much?

This is useful along at least two dimensions:

- **Informs action:** More efficient allocation of public and private resources via targeting in the future

  $\Rightarrow$ Personalized policies, ads, medicine, ...
- **Understanding**: Heterogeneous effects can be suggestive for underlying mechanisms

The challenge

Predicting effects

Recall that we defined the Conditional Average Treatment Effect (CATE):
$\tau(X) = E[Y(1) - Y(0) \mid X] = E[\Delta \mid X]$

- What is the expected treatment effect in the subpopulation with characteristics X?

This is a conditional expectation function (CEF) and we know how to approximate them with machine learning 🥳

BUT the outcome is unobserved

$\Rightarrow$ Standard supervised ML cannot be applied directly 😞

$\Rightarrow$ Clever ideas needed

Captain obvious

The most straightforward idea is to use the identification result that we established for experiments and under IA3 (strong ignorability)

$$\tau(X) = E[Y(1) - Y(0) \mid X] = \underbrace{E[Y \mid W = 1, X]}_{\text{CEF in treated group}} - \underbrace{E[Y \mid W = 0, X = x]}_{\text{CEF in control group}}$$

$\Rightarrow$ We have two CEFs of observable outcomes

This suggests the following procedure:

1. Use ML estimator of your choice to fit model $\hat{m}(1, X)$ in treated
2. Use ML estimator of your choice to fit model $\hat{m}(0, X)$ in controls
3. Estimate CATE as $\hat{\tau}(x) = \hat{m}(1, x) - \hat{m}(0, x)$

This is the so-called T-LEARNER

Why is this not the best idea? (1/2)

The prediction problems do not know of joint goal to approximate a difference

$\Rightarrow \hat{m}(1,x)$ minimizes $MSE(\hat{m}(1,x)) = E[(\hat{m}(1,x) - m(1,x))^2]$

$\Rightarrow \hat{m}(0,x)$ minimizes $MSE(\hat{m}(0,x)) = E[(\hat{m}(0,x) - m(0,x))^2]$

BUT what they should aim to minimize is

$$\begin{aligned} MSE(\hat{\tau}(x)) &= E[(\hat{\tau}(x) - \tau(x))^2] \\ &= E[(\hat{m}(1,x) - \hat{m}(0,x) - (m(1,x) - m(0,x)))^2] \\ &= E[(\hat{m}(1,x) - m(1,x))^2] + E[(\hat{m}(0,x) - m(0,x))^2] \\ &\quad - 2E[(\hat{m}(1,x) - m(1,x))(\hat{m}(0,x) - m(0,x))] \\ &= MSE(\hat{m}(1,x)) + MSE(\hat{m}(0,x)) - 2MCE(\hat{m}(1,x), \hat{m}(0,x)) \end{aligned}$$

Lechner (2018) calls the additional term Mean Correlated Error (MCE)

Why is this not the best idea? (2/2)

The MCE tells us that positively correlated errors matter less

Example

$\hat{m}(1,x) = m(1,x) + 2$ and $\hat{m}(0,x) = m(0,x) + 2 \Rightarrow$ both make same error

$\Rightarrow MSE(\hat{m}(1,x)) = 4$ and $MSE(\hat{m}(0,x)) = 4$

But their CATE would be just on point: $MSE(\hat{\tau}(x)) = 4 + 4 - 2(2 \times 2) = 0$

On the other hand if errors go in different direction $\hat{m}(0,x) = m(0,x) - 2$

$\Rightarrow MSE(\hat{m}(1,x)) = 4$, $MSE(\hat{m}(0,x)) = 4$ but $MSE(\hat{\tau}(x)) = 16$

$\Rightarrow$ Not so clever yet, methods that are aware of the joint goal could provide improvements

Two strategies

Different ways to teach ML to target causal parameters:

1. **Modify** supervised ML methods to target causal effect estimation
   - Causal tree
   - Causal forest
2. **Combine** supervised ML methods to target causal effect estimation
   - R-learner
   - DR-learner

The first are method specific, the second are generic approaches/meta-learner

Many more methods, but too many to cover in one day

$\Rightarrow$ focus on those that build on familiar ideas

Causal trees/forests

Causal Trees (1/3)

Recall from the supervised ML lecture that one representation of the splitting criterion for regression trees is to maximize squared predictions $\max \sum_i \hat{m}^{tree}(X_i)^2$

This is very helpful if we want to model effect heterogeneity

There is no need to observe the outcome we are predicting

Adapted to the causal setting our splitting criterion becomes $\max \sum_i \hat{\tau}^{tree}(X_i)^2$ or any other of the outcomes-free representations we derived

It suffices to be able to calculate the average treatment effect in each candidate leaf

We know how to do that...

Causal Trees (2/3)

Parent node can be split along variable j at split point s in a left leaf $L(j, s) = \{X \mid X_j \leq s\}$ and a right leaf $R(j, s) = \{X \mid X_j > s\}$

We seek j and s that maximize the sum of squared leaf specific ATEs:

$$\max_{j,s} \left[\sum_{i:x_i \in L(j,s)} \hat{\tau}^2_{L(j,s)} + \sum_{i:x_i \in R(j,s)} \hat{\tau}^2_{R(j,s)} \right] \tag{1}$$

where we have different ways to calculate the leaf specific ATEs $\hat{\tau}_{L(j,s)}$ and $\hat{\tau}_{R(j,s)}$:

- In experiments, difference in outcome means between treated and controls
- Using, e.g. AIPW in case of confounding

Causal Trees (3/3)

This is the basic idea behind Athey and Imbens (2016)

The paper also extends the splitting criterion to anticipate honest estimation

They use an honest approach to guarantee valid inference after tree building

Cross-validation can be applied for pruning as in the standard case

Implemented in causalTree package for experiments

Causal Forest(s): a longer journey

The development of Causal Forests went through different stages:

- Causal Forest of Wager and Athey (2018) is an ensemble of Causal Trees (CT) and focused on the experimental setting
- Causal Forest of Athey, Tibshirani & Wager (2019) uses an approximation of the CT splitting rule for a binary random treatment but extends also to
  - observational settings
  - continuous treatments

The former might be considered as an interim technology

Athey et al. provide an excellent infrastructure for causal ML around the Causal Forest in the R package `grf`

Causal Forest and partially linear model

Today we think about Causal Forest (CF) in the following way:

CF estimates CATEs as a localized/individualized residual-on-residual regression

$$\hat{\tau}^{cf}(x) = \underset{\tau}{\arg\min} \left\{ \sum_{i=1}^{N} \alpha_i(x) \left[(Y_i - \hat{m}(X_i)) - \tau(x)(W_i - \hat{e}(X_i)) \right]^2 \right\}, \tag{2}$$

where $\alpha(x)$ are x-specific weights

This should look familiar, it estimates a localized version of the partially linear estimator

$\Rightarrow$ nuisance parameters $\hat{m}(X)$ and $\hat{e}(X)$ estimated in first step using cross-fitting or out-of-bag (a random forest specific way to ensure out-of-sample predictions)

Recall: Random Forest as weights

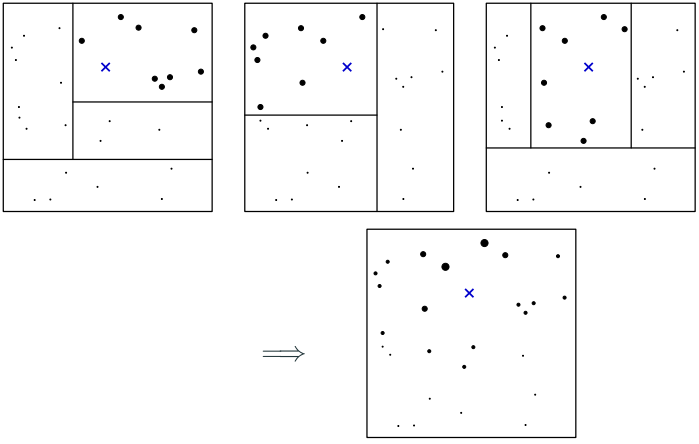


FIG. 1. *Illustration of the random forest weighting function. Each tree starts by giving equal (pos-itive) weight to the training examples in the same leaf as our test point x of interest, and zero weight to all the other training examples. Then the forest averages all these tree-based weightings, and effectively measures how often each training example falls into the same leaf as x.*

Source: Athey, Tibshirani & Wager (2019)

Splitting criterion uses influence function

What is the splitting criterion of causal forests that is applied in the tree building and eventually results in the weights for the RORR?

Causal forests use the following pseudo-outcome to place regression tree splits:

$$\rho = \left[\sum_i (W - \hat{e}(X))^2\right]^{-1} \left[(Y - \hat{m}(X)) - \hat{\tau}(W - \hat{e}(X))\right](W - \hat{e}(X))$$

This is the influence function of the partially linear estimator (see last week)

The splitting along the influence function is a general recipe to estimate heterogeneous parameters, e.g. in IV settings (Biewen & Kugler, 2021)

$\Rightarrow$ Generalized Random Forest

A note on statistical inference

Athey et al. (2019) show that the CATEs estimated using their CF are asymptotically normal and propose and implement an inference procedure

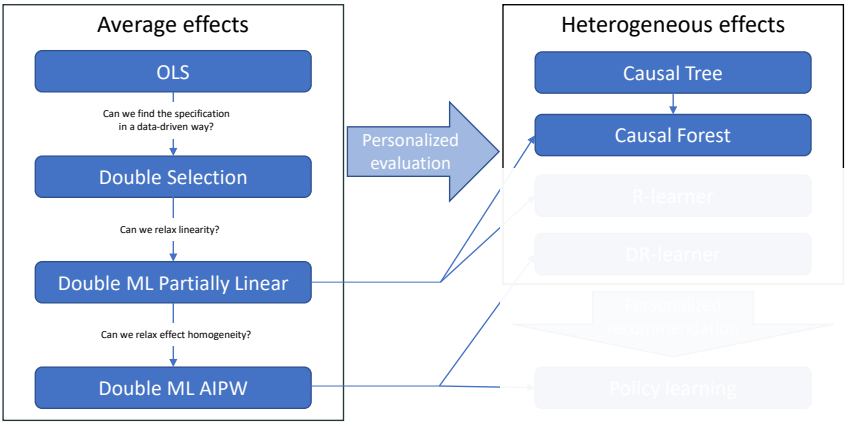
This is really nice and works when X is relatively small

You might see papers or presentations claiming that CF is so cool because it allows for high-dimensional X

Be aware that this is not in line with the theoretical analysis of the paper regarding inference

The results require low-dimensional variables and even there we may need a lot of observations until the ”asymptotics kick in”

Causal botanics unlocked



Simulation notebook: Causal Tree and Causal Forest

Application notebook: Predicting effects (part 1)

Causal Forest Fun Shinyapp

Meta-learner

What are Meta-learner?

Meta-learner combine multiple supervised ML steps in a pipeline that outputs predicted CATEs

The common ones require the following steps:

1. Estimate nuisance parameters using suitable ML method
2. Plug them into a clever minimization problem that targets CATE
3. Solve the minimization problem using suitable ML method
4. Predict CATE using the model learned in 3

Most popular ML methods are suitable and can be applied in steps 1, 3 and 4

Like for standard prediction methods, statistical inference is usually not available

R-learner: idea

Recall the partially linear model, but now allowing for x-specific treatment effects:

$$Y(w) = \tau(X)w + g(X) + U_{Y(w)};\ E[U_{Y(w)} \mid W, X] = 0 \tag{3}$$

$$\Rightarrow Y = W\tau(X) + g(X) + U_{Y(W)} \tag{4}$$

$$\Rightarrow Y - m(X) = \tau(X)(W - e(X)) + U_{Y(W)} \tag{5}$$

This motivates the R-learner of Nie and Wager (2020):

$$\hat{\tau}^{rl}(X) = \underset{\tau}{\arg\min} \sum_{i=1}^{N} \left(Y_i - \hat{m}(X_i) - \tau(X_i)(W_i - \hat{e}(X_i))\right)^2 \tag{6}$$

with cross-fitted high-quality nuisance parameters from first step

R-learner: linear ML methods

An interesting option arises if we model the CATE as linear function: $\tau(X) = X'\beta$

$$\hat{\beta}^{rl} = \underset{\beta}{\arg\min} \sum_{i=1}^{N} \big(Y_i - \hat{m}(X_i) - \underbrace{(W_i - \hat{e}(X_i))X_i'}_{=\tilde{X}_i'}\beta\big)^2 = \underset{\beta}{\arg\min} \sum_{i=1}^{N} \big(Y_i - \hat{m}(X_i) - \tilde{X}'\beta\big)^2 \quad (7)$$

where $\tilde{X} = (W - \hat{e}(X))X$ are modified/pseudo-**covariates**

and $\hat{\tau}^{rl}(x) = x\hat{\beta}^{rl} \neq \tilde{x}\hat{\beta}^{rl}$ is the estimated CATE for a specific x

$\Rightarrow$ All the shrinkage estimators (Lasso and friends) can be applied

Remark: The nuisance parameters can still be estimated with non-linear ML

R-learner: generic ML methods

We rewrite (6) differently if we are not willing to impose linearity of the CATE:

$$\hat{\tau}^{rl}(X) = \arg\min_{\tau} \sum_{i=1}^{N} (W_i - \hat{e}(X_i))^2 \left(\frac{Y_i - \hat{m}(X_i)}{W_i - \hat{e}(X_i)} - \tau(X_i) \right)^2 \tag{8}$$

Every supervised ML model that is capable of dealing with weighted minimization problems can be used (neural nets, random forest, boosting, ...) with

- weights $(W - \hat{e}(X))^2$
- pseudo-outcome $\frac{Y - \hat{m}(X)}{W - \hat{e}(X)}$
- the unmodified covariates

Rewrite R-learner

$$
\begin{aligned}
\hat{\tau}^{rl}(X) &= \arg\min_{\tau} \sum_{i=1}^{N} \left(Y_i - \hat{m}(X_i) - \tau(X_i)(W_i - \hat{e}(X_i))\right)^2 \\
&= \arg\min_{\tau} \sum_{i=1}^{N} \frac{(W_i - \hat{e}(X_i))^2}{(W_i - \hat{e}(X_i))^2} \left(Y_i - \hat{m}(X_i) - \tau(X_i)(W_i - \hat{e}(X_i))\right)^2 \\
&= \arg\min_{\tau} \sum_{i=1}^{N} (W_i - \hat{e}(X_i))^2 \left(\frac{Y_i - \hat{m}(X_i) - \tau(X_i)(W_i - \hat{e}(X_i))}{W_i - \hat{e}(X_i)}\right)^2 \\
&= \arg\min_{\tau} \sum_{i=1}^{N} (W_i - \hat{e}(X_i))^2 \left(\frac{Y_i - \hat{m}(X_i)}{W_i - \hat{e}(X_i)} - \tau(X_i)\right)^2
\end{aligned}
$$

DR-learner

Recall or note that

$$\tau(x) = \mathbb{E}\Bigg[\underbrace{m(1,x) - m(0,x) + \frac{(W - e(X))(Y - m(W,X))}{e(X)(1 - e(X))}}_{\tilde{Y}_{ATE}} \Bigg| X = x\Bigg]$$

$\Rightarrow \mathbb{E}[\tilde{Y}_{ATE} \mid X]$ is a CEF of a (admittedly fancy looking) random variable

and we know how to approximate CEFs of random variables

$\Rightarrow$ The DR-learner of Kennedy (2020) uses $\tilde{Y}_{ATE}$ in a generic ML problem

$$\hat{\tau}^{dr}(X) = \arg\min_{\tau} \sum_{i=1}^{N} \left(\tilde{Y}_{i,ATE} - \tau(X_i)\right)^2 \tag{9}$$

Discussion Meta-learner

Advantages:

- Very flexible
- Very individualized
- Predicted effects can be used for policy assignment (treat if $\hat{\tau}(x) > 0$)
- DR-learner naturally extends to multiple treatments

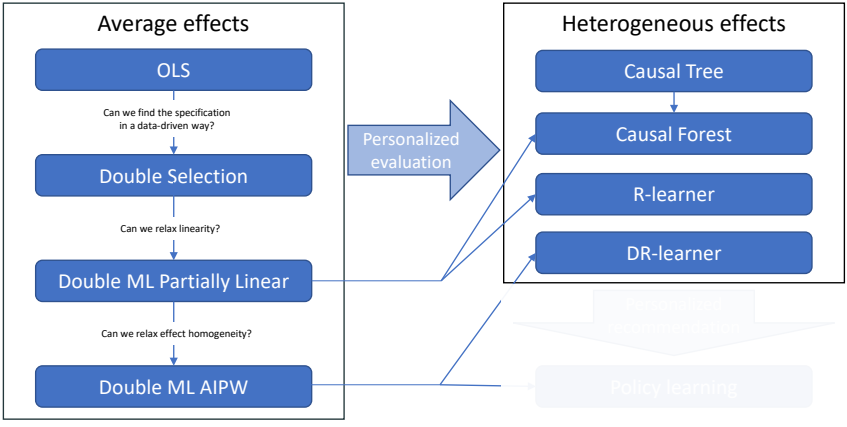
Disadvantages:

- No/little inference results (same problem as with standard ML)
- Hard to interpret (same problem as with standard ML)

Simulation notebook: Meta-learner

Application notebook: Predicting effects (part 2)

Heterogeneous effects unlocked



We could leverage concepts from previous lectures to make fast progress

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

$\Rightarrow$ separate identification and estimation