Causal Machine Learning

Causal Forest

Anthony Strittmatter

References

- Athey and Imbens (2016): "Recursive Partitioning for Heterogeneous Causal Effects", Proceedings in the National Academy of Sciences, 113 (27), pp. 7353-7360, download.

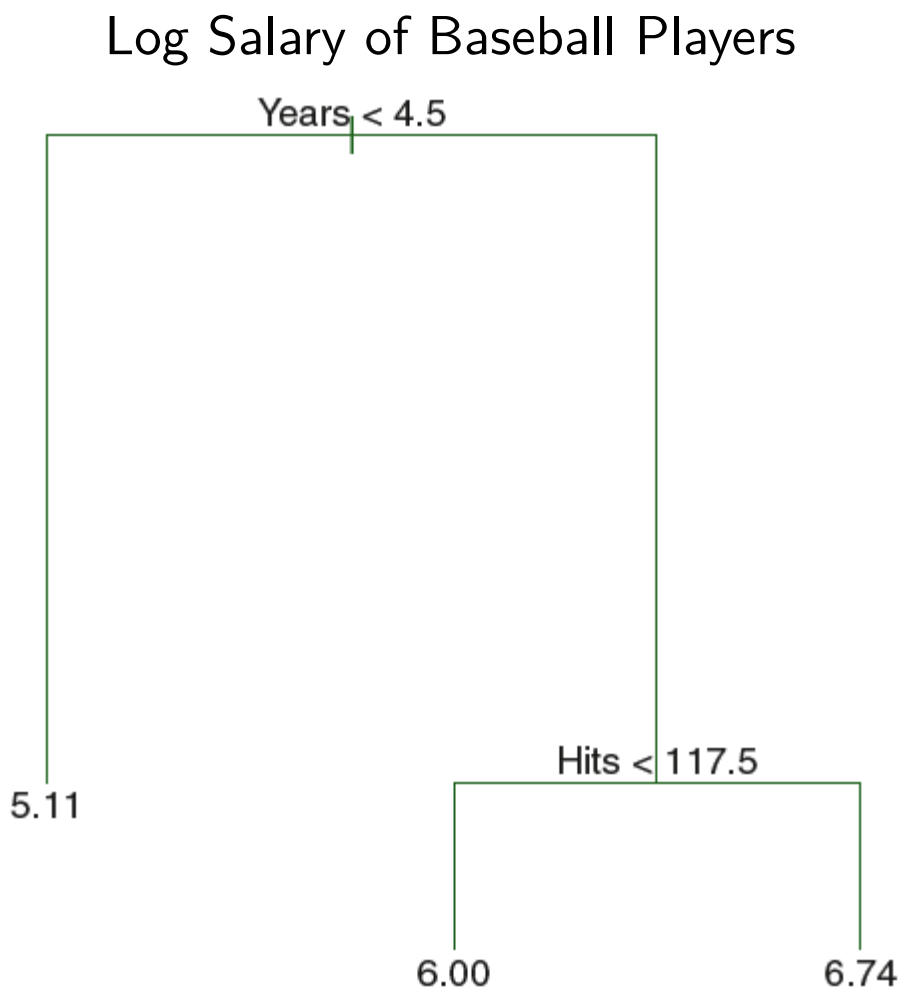
- Wager and Athey (2018): "Estimation and Inference of Heterogeneous Treatment Effects using Random Forests", Journal of the American Statistical Association, 113 (523), pp. 1228-1242, download.

Overview

Tree

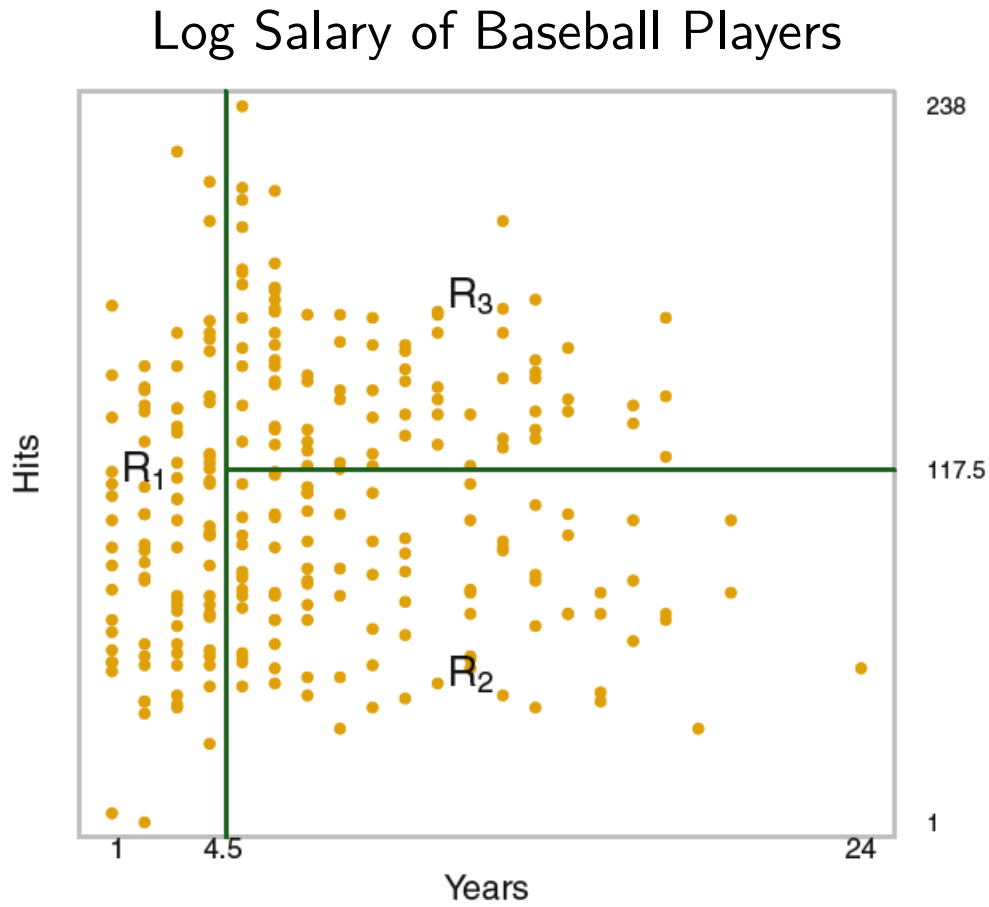
- Trees partition the sample into mutually exclusive groups l_j, which are called leaves.
- Let $\pi = \{l_1, ..., l_{\#(\pi)}\}$ be the terminal leaves of a specific tree or sample partition.
- Let $l_j \equiv l_j(x, \pi)$ be the respective leaf (for $j = 1, ..., \#(\pi)$).
- The leaf $l_j(x, \pi)$ of tree π is a function of the covariates X such that $x \in l_j$.
- Let $\#(\pi)$ be the number of terminal leaves in tree π.

Example: Shallow Tree



Source: James, Witten, Hastie, Tibshirani (2013)

Example: Shallow Tree (cont.)



Source: James, Witten, Hastie, Tibshirani (2013)

Recursive Partitioning

- Trees select the leaves with a top-down, greedy algorithm, which is called *recursive partitioning*.
- *Top-down* because we start with a root (tree without leaves) and successively add splits.
- *Greedy* because at each step of the tree building we add the split that improves the prediction power best (instead of looking ahead).

Tree Building Algorithm

(1) Start with the entire sample (root).

(2) For each predictor X_j and cut-point s define the pair of half-planes

$$I_1^{(j,s)} = \{X | X_j < s\} \text{ and } I_2^{(j,s)} = \{X | X_j \geq s\}.$$

- Calculate the mean outcomes $\bar{Y}_1$ and $\bar{Y}_2$ in each half-plane, respectively.
- Seek the covariate X_{j1}^* and the cut-point s_1^* that minimise

$$\sum_{i:X_i \in I_1^{(j,s)}} (Y_i - \bar{Y}_1)^2 + \sum_{i:X_i \in I_2^{(j,s)}} (Y_i - \bar{Y}_2)^2.$$

Tree Building Algorithm (cont.)

(2) For each predictor X_j and cut-point s define the triple of half-planes

$$I_1^{(j,s)} = \{X | X_{j1}^* < s_1^*, X_j < s\},\ I_2^{(j,s)} = \{X | X_{j1}^* < s_1^*, X_j \geq s\},\ \text{and}$$
$$I_3^{(j,s)} = \{X | X_{j1}^* \geq s_1^*\}$$

and

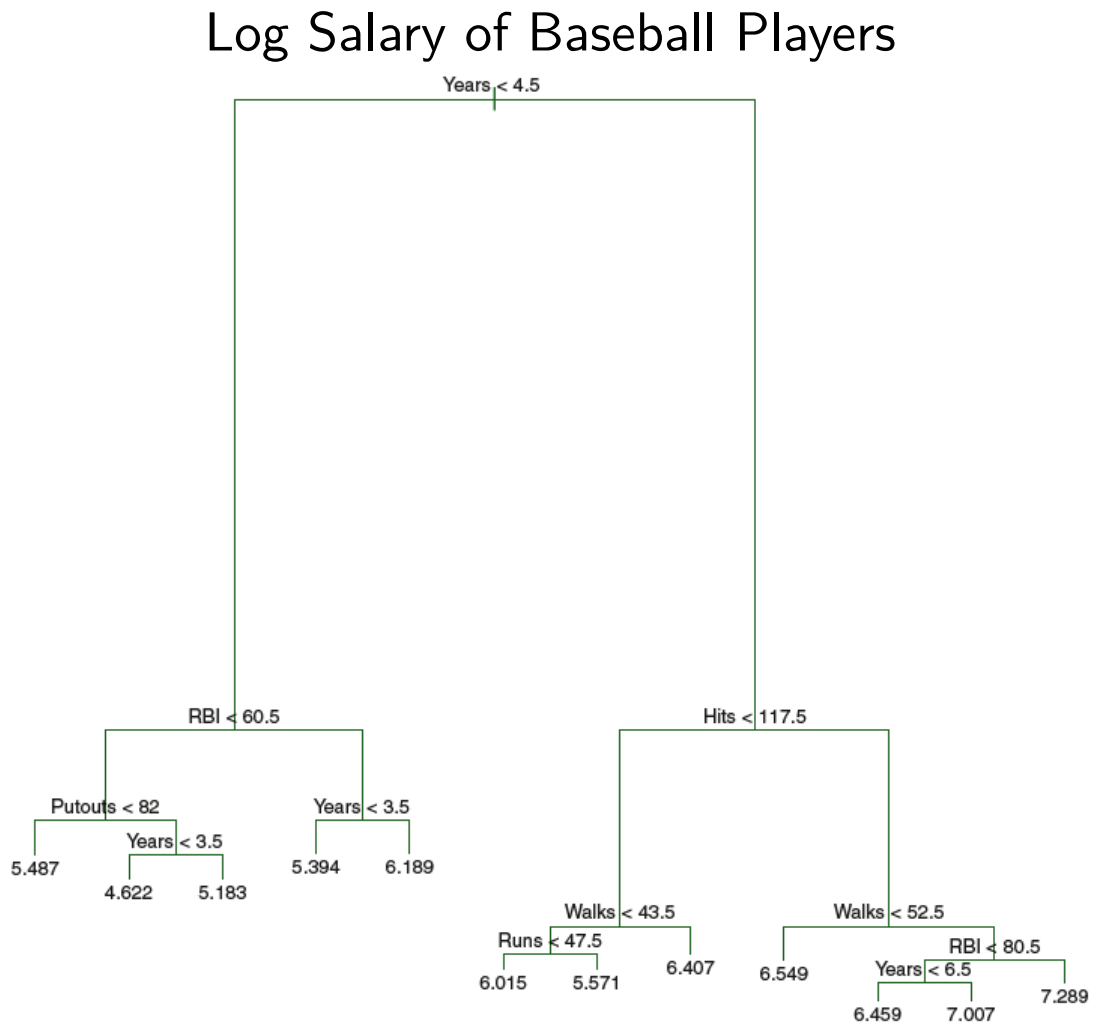
$$I_1^{(j,s)} = \{X | X_{j1}^* \geq s_1^*, X_j < s\},\ I_2^{(j,s)} = \{X | X_{j1}^* \geq s_1^*, X_j \geq s\},\ \text{and}$$
$$I_3^{(j,s)} = \{X | X_{j1}^* < s_1^*\}.$$

- Calculate the mean outcomes $\bar{Y}_1$, $\bar{Y}_2$, and $\bar{Y}_3$ in each half-plane, respectively.
- Seek the covariate X_{j2}^* and the cut-point s_2^* that minimise

$$\sum_{i:X_i \in I_1^{(j,s)}} (Y_i - \bar{Y}_1)^2 + \sum_{i:X_i \in I_2^{(j,s)}} (Y_i - \bar{Y}_2)^2 + \sum_{i:X_i \in I_3^{(j,s)}} (Y_i - \bar{Y}_3)^2.$$

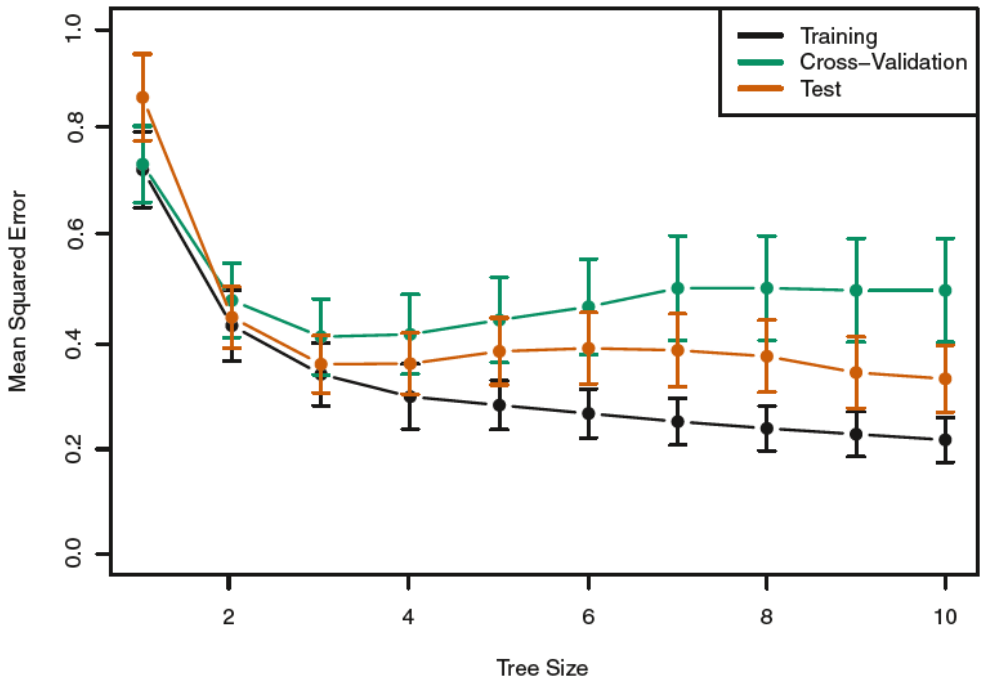
(3) Continue until some stopping rule is reached (e.g., max. tree size, min. terminal leave size, min. MSE gain) .

"Deep" Tree



Source: James, Witten, Hastie, Tibshirani (2013)

Selecting Optimal Tree Size



→ Pruning the complexity of trees can improve out-of-sample prediction power.

→ Select optimal tree size π with cross-validation.

Source: James, Witten, Hastie, Tibshirani (2013)

Complexity Pruning

(A) Use recursive partitioning to grow the deep tree π_0 in the training data.

(B) For each value of α a subtree $\pi \subseteq \pi_0$ minimises

$$\sum_{j=1}^{\#(\pi)} \sum_{i:X_i \in I_j} (Y_i - \bar{Y}_j)^2 + \alpha \#(\pi). \tag{1}$$

Obtain a sequence of best subtrees.

(C) Use cross-validation to choose α. Partition the sample in k folds. For each fold:

- (a) Repeat steps (A) and (B) excluding the kth-fold.
- (b) Evaluate the MSE using equation (1) in the kth-fold.
- (c) Average the MSE across the k folds for each value of α and select the α that minimises the average MSE.

(D) Return to the subtree from (B) with the selected value of α.

Prediction

- For the selected tree π^* use the estimation sample to predict

$$\widehat{Y}_i = \frac{1}{\sum_{j=1}^{\#(\pi^*)} \sum_{i=1}^{N} 1\{X_i \in I_j(x, \pi^*)\}} \sum_{j=1}^{\#(\pi^*)} \sum_{i=1}^{N} 1\{X_i \in I_j(x, \pi^*)\} \cdot Y_i.$$

- This is equivalent to the linear regression

$$\min_{\beta} \sum_{i=1}^{N} \left(Y_i - \sum_{j=1}^{\#(\pi^*)} 1\{X_i \in I_j(x, \pi^*)\} \beta_j \right)^2.$$

Advantages and Disadvantages of Trees

Advantages:

- Shallow trees are very easy to understand.
- Shallow trees can be displayed graphically in a nice way.
- Trees automatically handle interactions between covariates.
- It is not necessary to transform covariates as long as they have an order.

Disadvantages:

- Often trees are unstable and have a high variance.

Bootstrap Sampling

- We observe a sample $\{Y_i, X_i\}_{i=1}^N$ with size N
- Bootstrap algorithm:
  1. Draw randomly N observations with replacement from the original sample
  2. Estimate $\widehat{Y}_i^b$ in the "bootstrapped" sample b with a tree
  3. Repeat 1. and 2. B times
- We obtain B estimates $\widehat{Y}_i^1, \widehat{Y}_i^2, ..., \widehat{Y}_i^B$

Subsampling

- We observe a sample $\{Y_i, X_i\}_{i=1}^N$ with size N
- Subsampling algorithm:
  1. Draw randomly $M < N$ observations without replacement from the original sample
  2. Estimate $\widehat{Y}_i^s$ in the subsample s with a tree
  3. Repeat 1. and 2. S times
- We obtain S estimates $\widehat{Y}_i^1, \widehat{Y}_i^2, ..., \widehat{Y}_i^S$

Random Forests

- Build G deep trees π_g on different subsets of the data (subsampling or bootstrapping) and covariates.

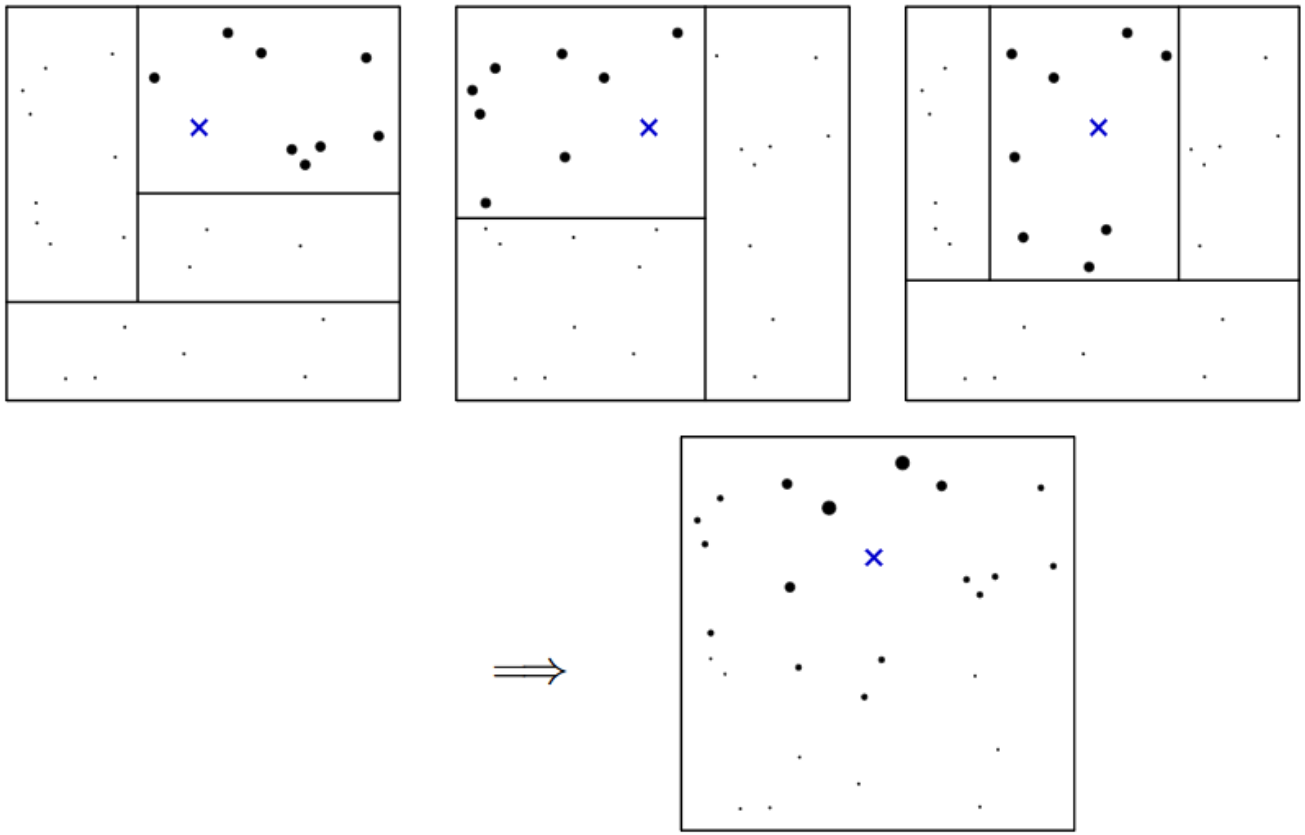
  $\rightarrow$ decorrelated trees

- These trees are overfitted in the sample and will have a high out-of-sample variance.
- To overcome this problem, we aggregate the trees

$$\widehat{Y}_i^{RF} = \frac{1}{G} \sum_{g=1}^{G} \widehat{Y}_i^{\pi_g}.$$

- This procedure is often called bootstrap-aggregation ("bagging").
- We loose interpretability but gain prediction power compared to (shallow) trees.
- Tuning parameters: Subsample size, number of covariates, number of trees, etc.

Random Forest: Weighted Representation



Source: Athey, Tibshirani, Wager (2018)

Random Forests: Weighted Representation

- Tree weights:

$$\alpha_{ig}(x) = \frac{1\{X_i \in l_j(x, \pi_g)\}}{\sum_{i=1}^{N} 1\{X_i \in l_j(x, \pi_g)\}}$$

- Forest weights:

$$\alpha_i(x) = \frac{1}{G} \sum_{i=1}^{G} \alpha_{ig}(x)$$

- Predicted outcome:

$$\widehat{\mu}(x) = \sum_{i=1}^{N} \alpha_i(x) Y_i \text{ and } \widehat{Y}_i = \widehat{\mu}(X_i)$$

Overview

Why are Heterogeneous Effects Interesting?

- For most treatments, there is no specific reasons to assume that the effects are homogeneous
- If the effects are heterogeneous, there might be some individuals who benefit from a treatment and others may experience disadvantages from the treatment
- Investigating heterogeneous treatment effects enables us to understand how a treatment works
- Most treatments have a target group (e.g. disadvantaged youths are the target group of the Job Corps program). If there are two treatments with the same average effect, they might be evaluated differently, when one treatment mainly affects the target group while the other treatment affects non-targets

$\Rightarrow$ **Can we use ML to identify the relevant dimensions of effect heterogeneity in a data-driven way?**

Relevant Parameters

- Individual Causal Effects:

$$\delta_i = Y_i^1 - Y_i^0$$

- Conditional Average Treatment Effects (CATEs):

$$\delta(x) = E[Y_i^1 - Y_i^0 | X_i = x] = E[\delta_i | X_i = x]$$

Why Can't we use Off-the-Shelf ML?

Out-of-Sample Mean-Squared-Error (MSE):

- Individual Causal Effect δ_i:

$$MSE_{\hat{\delta}} = E\left[(\hat{\delta}_i - \delta_i)^2\right] = \underbrace{E\left[(\hat{\delta}_i - E[\hat{\delta}_i])^2\right]}_{\text{Variance}} + \underbrace{E[\hat{\delta}_i - \delta_i]^2}_{\text{Squared Bias}}$$

- CATEs $\delta(x)$:

$$\begin{aligned} MSE_{\hat{\delta}(x)} &= E\left[(\hat{\delta}(x) - \delta(x))^2\right] \\ &= \underbrace{E\left[(\hat{\delta}(x) - E[\hat{\delta}(x)])^2\right]}_{\text{Variance}} + \underbrace{E[\hat{\delta}(x) - \delta(x)]^2}_{\text{Squared Bias}} \end{aligned}$$

$\rightarrow$ δ_i and $\delta(x)$ are unobservable

Overview

Adapting the ML Algorithm

- CATE: $\delta(x) = \mu_1(x) - \mu_0(x)$
- Criterion to estimate $\hat{\delta}(x)$:

$$\begin{aligned} MSE(\hat{\delta}(x)) =& E\left[(\hat{\delta}(x) - \delta(x))^2\right], \\ =& E\left[(\hat{\mu}_1(x) - \mu_1(x))^2\right] + E\left[(\hat{\mu}_0(x) - \mu_0(x))^2\right] \\ & - 2E\left[(\hat{\mu}_1(x) - \mu_1(x))\right] E\left[(\hat{\mu}_0(x) - \mu_0(x))\right], \\ =& MSE(\hat{\mu}_1(x)) + MSE(\hat{\mu}_0(x)) - 2MCE(\hat{\mu}_1(x), \hat{\mu}_0(x)), \end{aligned}$$

with mean-correlation-error
$MCE = E\left[(\hat{\mu}_1(x) - \mu_1(x))\right] E\left[(\hat{\mu}_0(x) - \mu_0(x))\right]$

$\rightarrow$ The main complication is the estimation of MCE

Reference: Lechner (2018)

Simple Solution

- Estimate $\hat{\mu}_1(x)$ in the sample with $D_i = 1$ and $\hat{\mu}_0(x)$ in the sample with $D_i = 0$ using two separate ML optimisation criteria
- For example, optimise π_d for the tree estimator using the criteria

$$\widehat{MSE}(\hat{\mu}_d(x)) = \frac{1}{N_X^d} \sum_{i=1}^{N} 1\{D_i = d\} 1\{X_i \in l_j(x, d, \pi)\} \cdot (\hat{\mu}_d(X_i) - Y_i)^2$$

with

$$\hat{\mu}_d(X_i) = \frac{1}{N_X^d} \sum_{i=1}^{N} 1\{D_i = d\} 1\{X_i \in l_j(x, d, \pi)\} \cdot Y_i$$

and $N_X^d = \sum_{i=1}^{N} 1\{D_i = d\} 1\{X_i \in l_j(x, d, \pi)\}$

- Estimate CATEs with $\hat{\delta}(x) = \hat{\mu}_1(x, \pi_1^*) - \hat{\mu}_0(x, \pi_0^*)$
- Implicit assumption: $MCE = const$

Nearest Neighbour Fix

- Use nearest neighbour to approximate the MCE

$$\widehat{MCE}(\hat{\mu}_d(x)) = \frac{1}{N_X} \sum_{i=1}^{N} 1\{X_i \in I_j(x, d, \pi)\} \cdot \left(\hat{\mu}_1(X_i) - \tilde{Y}_i^1\right) \left(\hat{\mu}_0(X_i) - \tilde{Y}_i^0\right)$$

with

$$\tilde{Y}_i^d = \begin{cases} Y_i & \text{if } D_i = d \\ \tilde{Y}_i^{NN,1-d} & \text{if } D_i = 1 - d \end{cases}$$

and $N_X = \sum_{i=1}^{N} 1\{X_i \in I_j(x, d, \pi)\}$

- Select the splitting rule π_δ minimising
$\widehat{MSE}(\hat{\delta}(x)) = \widehat{MSE}(\hat{\mu}_1(x)) + \widehat{MSE}(\hat{\mu}_0(x)) - 2\widehat{MCE}(\hat{\mu}_1(x), \hat{\mu}_0(x))$
- Estimate CATEs with $\hat{\delta}(x) = \hat{\mu}_1(x, \pi_\delta^*) - \hat{\mu}_0(x, \pi_\delta^*)$

Reference: Lechner (2018)

Overview

Causal Tree

- S^{te} is the test sample and S^{tr} is the training sample
- Optimisation criteria for δ:

$$
\begin{aligned}
MSE_\delta(S^{te}, S^{tr}, \pi) =& \frac{1}{N^{te}} \sum_{i \in S^{te}} \left\{ (\hat{\delta}(X_i; S^{tr}, \pi) - \delta_i)^2 - \delta_i^2 \right\}, \\
=& \frac{1}{N^{te}} \sum_{i \in S^{te}} \left\{ \hat{\delta}^2(X_i; S^{tr}, \pi) - 2 \cdot \delta_i \cdot \hat{\delta}(X_i; S^{tr}, \pi) \right\}, \\
=& \frac{1}{N^{te}} \sum_{i \in S^{te}} \Big\{ \hat{\delta}^2(X_i; S^{tr}, \pi) \\
& -2 \cdot \hat{\delta}(X_i; S^{te}, \pi) \cdot \hat{\delta}(X_i; S^{tr}, \pi) \Big\},
\end{aligned}
$$

with $E_{S^{te}}[\delta_i | i \in S^{te} : i \in I_j(x, d, \pi)] = E[\hat{\delta}(X_i; S^{te}, \pi)]$

- The cross-validation analog is $\widehat{MSE}_\delta(S^{tr,CV}, S^{tr}, \pi)$

Reference: Athey and Imbens (2016)

Overview

Preliminaries

- Univariate OLS regression:

$$Y_i = \alpha + \delta D_i + u_i$$

with

$$\delta = \frac{Cov(Y, D)}{Var(D)}$$

- Variance estimator:

$$Var(D) = \frac{1}{N} \sum_{i=1}^{N} (D_i - \bar{D})^2$$

- Covariance estimator:

$$Cov(Y, D) = \frac{1}{N} \sum_{i=1}^{N} (Y_i - \bar{Y})(D_i - \bar{D})$$

Causal Forest

- Consider the random effects model $Y_i = D_i\delta(X_i) + \epsilon_i$ with

$$\delta(x) = Var(D_i|X_i = x)^{-1} Cov(Y_i, D_i|X_i = x)$$

- Estimator

$$\hat{\delta}(x) = \left(\sum_{i=1}^{N} \alpha_i(x)(D_i - \bar{D}_\alpha)^2 \right)^{-1} \sum_{i=1}^{N} \alpha_i(x)(D_i - \bar{D}_\alpha)(Y_i - \bar{Y}_\alpha)$$

with $\bar{D}_\alpha = \sum_{i=1}^{N} \alpha_i(x) D_i$ and $\bar{Y}_\alpha = \sum_{i=1}^{N} \alpha_i(x) Y_i$

Algorithm

- Optimal optimisation criteria:

$$\max \Delta(C_1, C_2) = \frac{N_{C_1} N_{C_2}}{N_P} \left(\hat{\delta}_{C_1} - \hat{\delta}_{C_2} \right)^2$$

  $\rightarrow$ Requires the estimation of $\hat{\delta}_{C_1}$ and $\hat{\delta}_{C_2}$ at each candidate split

- Computational efficient optimisation algorithm
  (1) **Labelling step:** Calculate $\hat{\delta}_P$, $A_P = Var_P(D_i)$, and the pseudo-outcome

$$p_i = A_P^{-1}(D_i - \bar{D}_P)(Y_i - \bar{Y}_P - (D_i - \bar{D}_P)\hat{\delta}_P)$$

  (2) **Regression step:**

$$\max \tilde{\Delta}(C_1, C_2) = \frac{1}{N_{C_1}} \left(\sum_{i: X_i \in C_1} p_i \right)^2 + \frac{1}{N_{C_2}} \left(\sum_{i: X_i \in C_2} p_i \right)^2$$

  (3) Relabel child nodes to parent nodes and repeat (1) and (2) until stopping criteria of tree is reached
  (4) Calculate weights $\alpha_{ig}(x)$ and build forest by repeating (1)-(3) with different subsamples and covariates

Pseudo-Outcome

- Assume D_i is binary and randomly assigned and denote $p = \bar{D}_P = Pr(D_i = 1) = Pr(D_i = 1|C_j)$

$$
\begin{aligned}
E[p_i|C_j] =& E\left[\frac{D_i - p}{p(1-p)}(Y_i - \bar{Y}_P - (D_i - p)\hat{\delta}_P) \middle| C_j \right] \\
=& E\left[\frac{1}{p}(Y_i - \bar{Y}_P - (1-p)\hat{\delta}_P) \middle| C_j, D_i = 1 \right] p \\
& - E\left[\frac{1}{(1-p)}(Y_i - \bar{Y}_P + p\hat{\delta}_P) \middle| C_j, D_i = 0 \right] (1-p) \\
=& E\left[Y_i - \bar{Y}_P - (1-p)\hat{\delta}_P \middle| C_j, D_i = 1 \right] - E\left[Y_i - \bar{Y}_P + p\hat{\delta}_P \middle| C_j, D_i = 0 \right] \\
=& E\left[Y_i(1) - \bar{Y}_P \middle| C_j, D_i = 1 \right] - E\left[Y_i(0) - \bar{Y}_P \middle| C_j, D_i = 0 \right] - \hat{\delta}_P \\
=& E\left[Y_i(1) - Y_i(0) | C_j \right] - \hat{\delta}_P
\end{aligned}
$$

- Difference between approximated causal effect at child and parent node
- Pseudo-outcome is updated at each parent node

Local Centering

- Generalised causal forest is targeted to find maximum heterogeneity in pseudo-outcome
- But not specifically designed to account for selection into treatment (even though deep causal forests correct automatically to some extent for selection)
- Define centred variables

$$\tilde{Y}_i = Y_i - \hat{\mu}(X_i)$$

and

$$\tilde{D}_i = D_i - \hat{p}(X_i)$$

with $\hat{\mu}(x) = \hat{E}[Y_i|X_i = x]$ and $\hat{p}(x) = \hat{E}[D_i|X_i = x]$

- Apply generalised causal forest algorithm to $\tilde{Y}_i$ and $\tilde{D}_i$ instead of Y_i and D_i

$\rightarrow$ Orthogonalisation

Local Centering (cont.)

MSE from Simulation

Confounding	Heterogeneity	K	N	GCF	Centred GCF
No	No	10	800	0.85	0.87
No	No	10	1,600	0.58	0.59
No	No	20	800	0.92	0.93
No	No	20	1,600	0.52	0.52
Yes	No	10	800	1.12	0.27
Yes	No	10	1,600	0.80	0.20
Yes	No	20	800	1.17	0.17
Yes	No	20	1,600	0.95	0.11
Yes	Yes	10	800	1.92	0.91
Yes	Yes	10	1,600	1.51	0.62
Yes	Yes	20	800	1.92	0.93
Yes	Yes	20	1,600	1.55	0.57

Note: K is the number of covariates in the simulation

Source: Athey, Tibshirani, and Wager (2018)

Asymptotic Properties for Causal Forest

- Minimum subsample size S is scaled N^{β} with $\beta_{min} < \beta < 1$
- CATEs are consistent and asymptotically normal

$$(\hat{\delta}(x) - \delta(x))/\sqrt{Var(\delta(x))} \xrightarrow{d} N(0,1)$$

Reference: Wager and Athey (2018)

Inference

- Infinitesimal Jackknife
  - $Q_{ig} = 1$ when observation i is used to built tree g and $Q_{ig} = 0$ otherwise
  - Calculate the covariance $Cov(\hat{\delta}(x), Q_{ig})$ across all trees $g = 1, ..., G$
  - Variance estimator:

$$\hat{V}(x) = \frac{N-1}{N} \left(\frac{N}{N-S} \right)^2 \sum_{i=1}^{N} Cov(\hat{\delta}(x), Q_{ig})^2$$

  - $(N/(N-S)))^2$ is a finite sample correction for subsampling
  - $\hat{V}(x) \xrightarrow{p} Var(\delta(x))$

Reference: Wager and Athey (2018)

Advantages and Disadvantages of Causal Forest

Advantages:

- D_i can be binary or continuous
- Asymptotic properties for CATEs available
- Variance estimator available

Disadvantages:

- Best suited for experiments
- We have to assume that subsample sizes do not get too small, because otherwise asymptotic results break down