Meta-Learners for Estimating Heterogeneous Treatment Effects using Machine Learning

Jasjeet S. Sekhon
with Peter Bickel, Sören Künzel, and Bin Yu

UC Berkeley

May 3, 2017

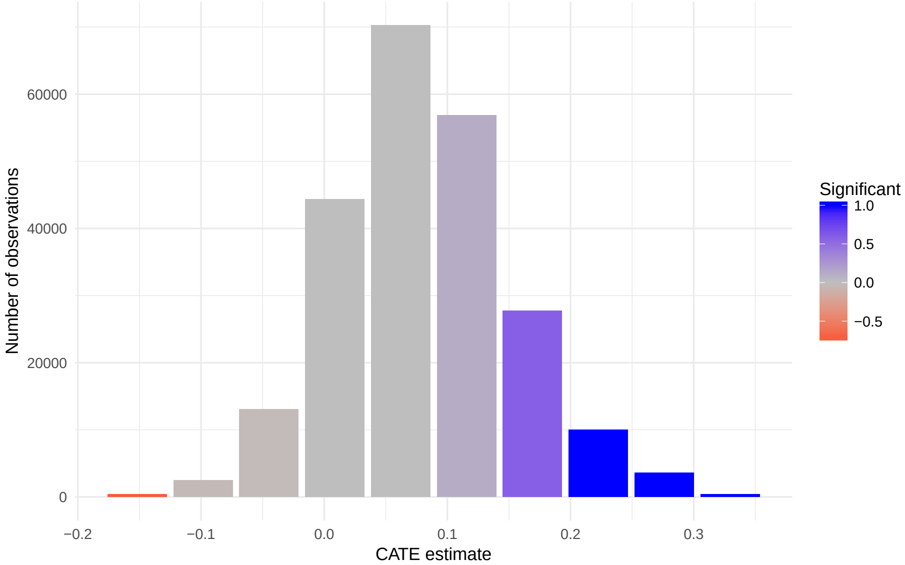
Heterogenous Data and Questions

- Measuring human activity has generated large datasets with granular data:
    - Individual voter files
    - Surveys linked to ancillary data
    - Browsing, search, and purchase data from online platforms
    - Administrative data: schools, criminal justice, IRS
- Big in size and breadth: wide datasets
- Data can be used for personalization of treatments, modeling behavior
- Many inferential issues: e.g., unknown sampling frames, heterogeneity, targeting optimal treatments

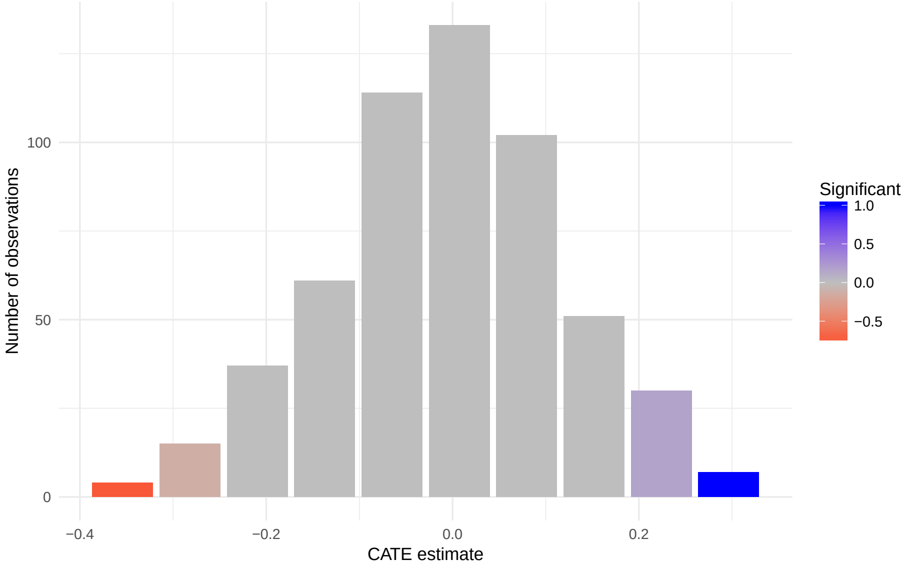
Prediction versus Causal Inference

- Causal Inference is like a prediction problem: but predicting something we don't directly observe and possibly cannot estimate well in a given sample
- ML algorithms are good at prediction, but have issues with causal inference:
  - Interventions imply counterfactuals: response schedule versus model prediction
  - Validation requires estimation in the case of causal inference
  - Identification problems not solved by large data
  - Predicting the outcome mistaken for predicting the causal effect
    - targeting based on the lagged outcome

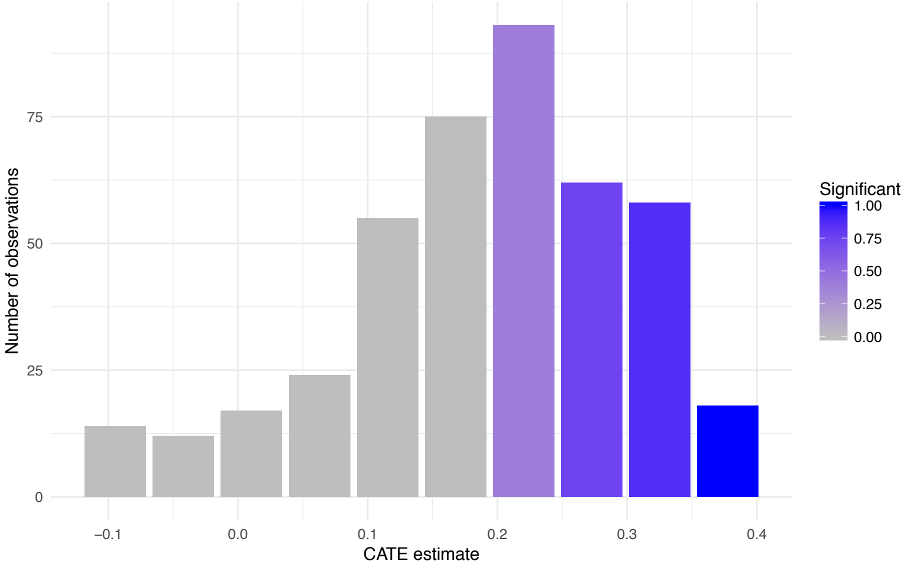
GOTV: Social pressure (Gerber, Green, Lairmer, 2008)



Persuasion: Abortion stigma (Broockman, Kalla, Sekhon, 2017)



Persuasion: Transphobia (Broockman, Kalla, 2015)



Conditional Average Treatment Effect (CATE)

Individual Treatment Effect (ITE): $D_i := Y_i(t) - Y_i(c)$

Let $\hat{\tau}_i$ be an estimator for D_i

$\tau(x_i)$ is the **CATE** for all units whose covariate vector is equal to x_i:

$$CATE := \tau(x_i) := \mathbb{E}\Big[D\Big|X = x_i\Big] = \mathbb{E}\Big[Y(t) - Y(c)\Big|X_i = x_i\Big]$$

Variance of Conditional Average Treatment Effect

$$\textit{CATE} := \tau(x_i) := \mathbb{E}\Big[D\Big|X = x_i\Big] = \mathbb{E}\Big[Y(t) - Y(c)\Big|X_i = x_i\Big]$$

Decompose the MSE at x_i:

$$\mathbb{E}\left[(D_i - \hat{\tau}_i)^2|X_i = x_i\right] = \underbrace{\mathbb{E}\left[(D_i - \tau(x_i))^2|X_i = x_i\right]}_{\text{Approximation Error}} + \underbrace{\mathbb{E}\left[(\tau(x_i) - \hat{\tau}_i)^2|X_i = x_i\right]}_{\text{Estimation Error}}$$

- Since we cannot estimate D_i, we estimate the CATE at x_i
- But the error for the CATE is not the same as the error for the ITE

Supplementary

How to estimate the CATE?

Meta–learners

A meta–learner decomposes the problem of estimating the CATE into several sub–regression problems. The estimator which solve those sub–problems are called **base–learners**

- Flexibility to choose base–learners which work well in a particular setting
- Tuning can be done for each base–learner separately

How to estimate the CATE?

$$\begin{aligned}\tau(x) &= \mathbb{E}[Y(1) - Y(0)|X = x] \\ &= \mathbb{E}[Y(1)|X = x] - \mathbb{E}[Y(0)|X = x] \\ &= \mu_1(x) - \mu_0(x)\end{aligned}$$

How to estimate the CATE?

$$
\begin{aligned}
\tau(x) &= \mathbb{E}[Y(1) - Y(0)|X = x] \\
&= \mathbb{E}[Y(1)|X = x] - \mathbb{E}[Y(0)|X = x] \\
&= \mu_1(x) - \mu_0(x)
\end{aligned}
$$

T–learner

1.) Split the data into control and treatment group,

2.) Estimate the response functions separately,

$$
\begin{aligned}
\hat{\mu}_1(x) &= \hat{\mathbb{E}}[Y^{obs}|X = x, W = 1] \\
\hat{\mu}_0(x) &= \hat{\mathbb{E}}[Y^{obs}|X = x, W = 0],
\end{aligned}
$$

3.) $\hat{\tau}(x) := \hat{\mu}_1(x) - \hat{\mu}_0(x)$

How to estimate the CATE?

$$
\begin{aligned}
\tau(x) &= \mathbb{E}[Y(1) - Y(0)|X = x] \\
&= \mathbb{E}[Y(1)|X = x] - \mathbb{E}[Y(0)|X = x] \\
&= \mu_1(x) - \mu_0(x)
\end{aligned}
$$

T–learner

1.) Split the data into control and treatment group,

2.) Estimate the response functions separately,

$$
\begin{aligned}
\hat{\mu}_1(x) &= \hat{\mathbb{E}}[Y^{obs}|X = x, W = 1] \\
\hat{\mu}_0(x) &= \hat{\mathbb{E}}[Y^{obs}|X = x, W = 0],
\end{aligned}
$$

3.) $\hat{\tau}(x) := \hat{\mu}_1(x) - \hat{\mu}_0(x)$

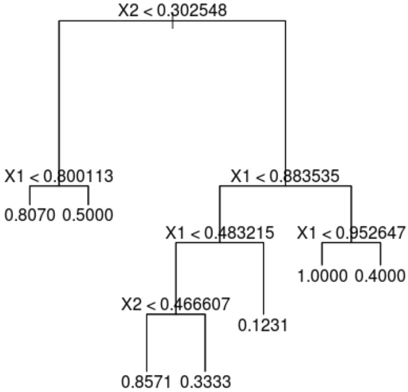
S–learner

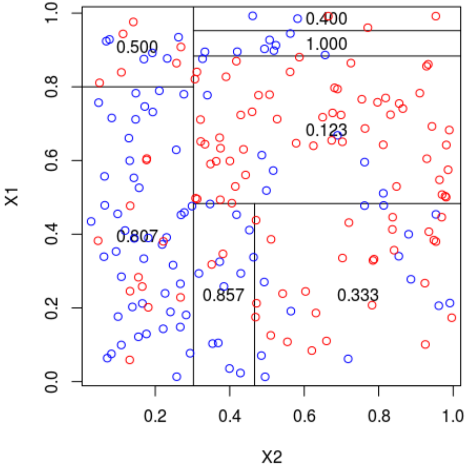
1.) Use the treatment assignment as a usual variable without giving it any special role and estimate

$$
\hat{\mu}(x, w) = \hat{\mathbb{E}}[Y^{obs}|X = x, W = w]
$$

2.) $\hat{\tau}(x) := \hat{\mu}(x, 1) - \hat{\mu}(x, 0)$

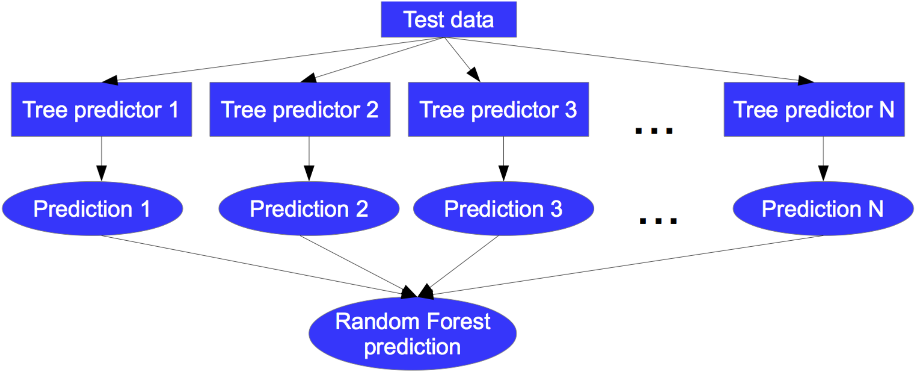
Regression Trees





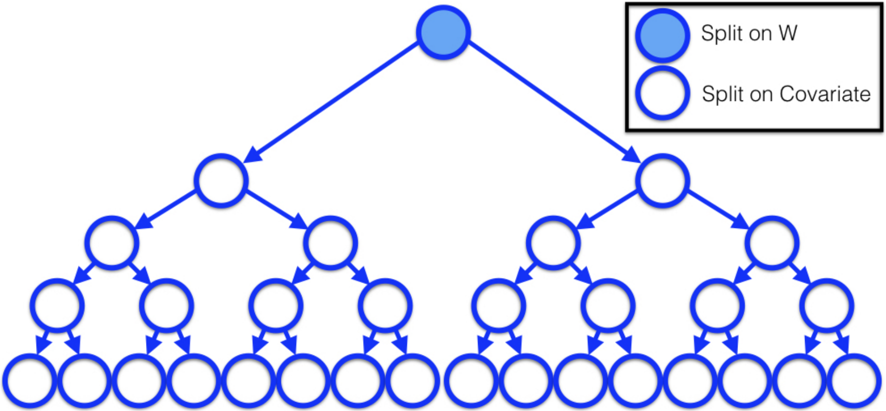
http://freakonometrics.hypotheses.org/1279

Random Forest = Many "random" Trees



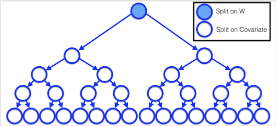
Supplementary

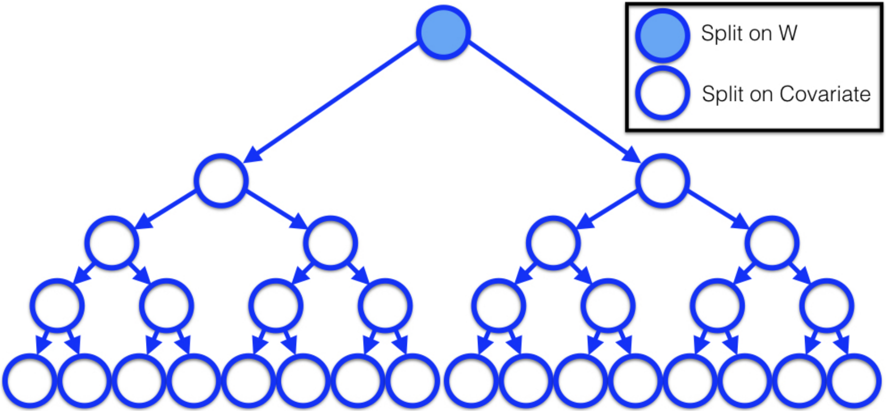
CATE $:= \hat{\tau}(x) = f(x, w = 1) - f(x, w = 0)$



CATE $:= \hat{\tau}(x) = f(x, w = 1) - f(x, w = 0)$

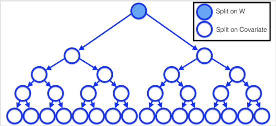
T-Learner

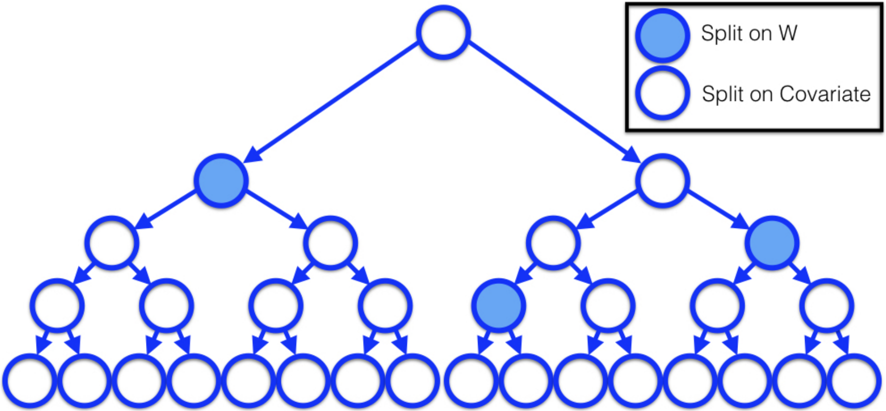




CATE $:= \hat{\tau}(x) = f(x, w = 1) - f(x, w = 0)$

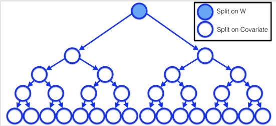
T-Learner



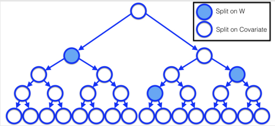


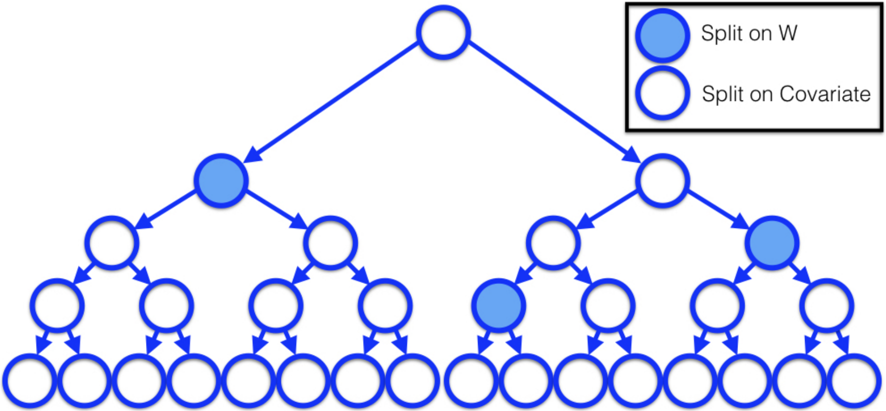
CATE $:= \hat{\tau}(x) = f(x, w = 1) - f(x, w = 0)$

T-Learner



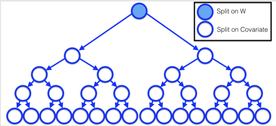
S-Learner



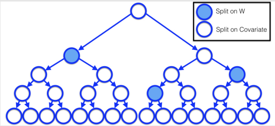


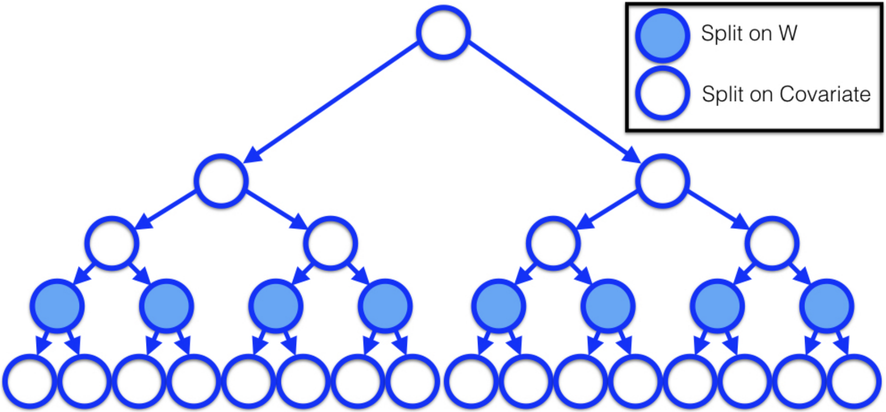
CATE $:= \hat{\tau}(x) = f(x, w=1) - f(x, w=0)$

T-Learner



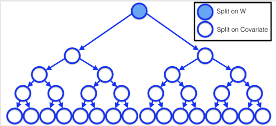
S-Learner



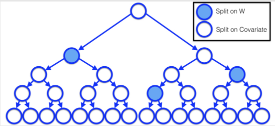


CATE $:= \hat{\tau}(x) = f(x, w = 1) - f(x, w = 0)$

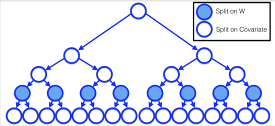
T-Learner

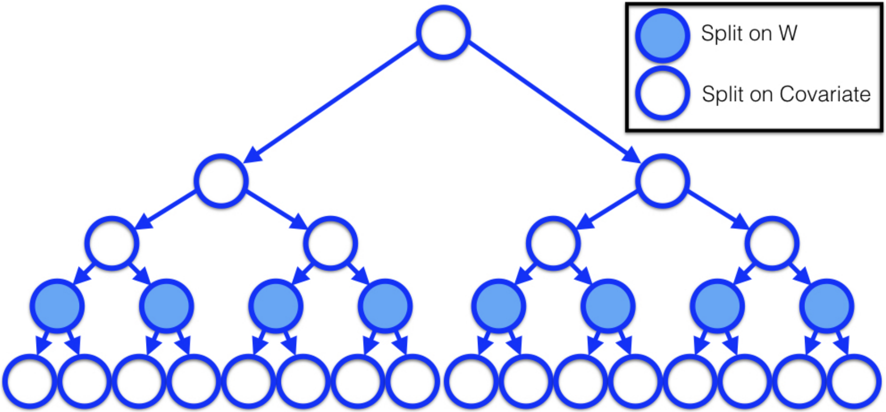


S-Learner



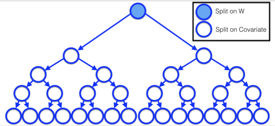
Causal Forest



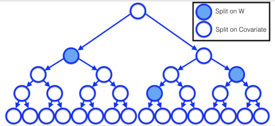


$$\text{CATE} := \hat{\tau}(x) = f(x, w = 1) - f(x, w = 0)$$

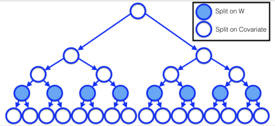
T-Learner



S-Learner



Causal Forest

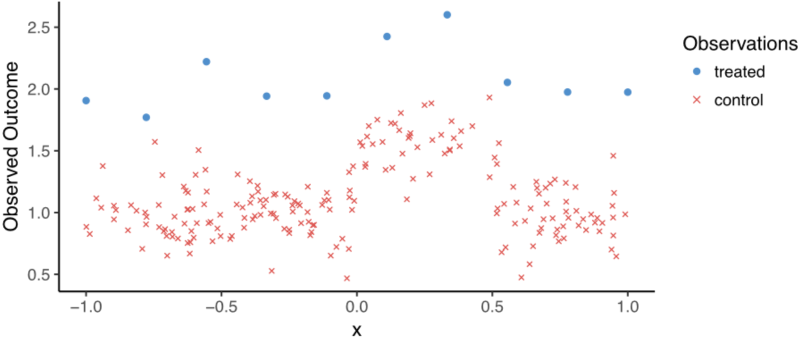


Honesty (Biau and Scornet, 2015; Scornet, 2015)

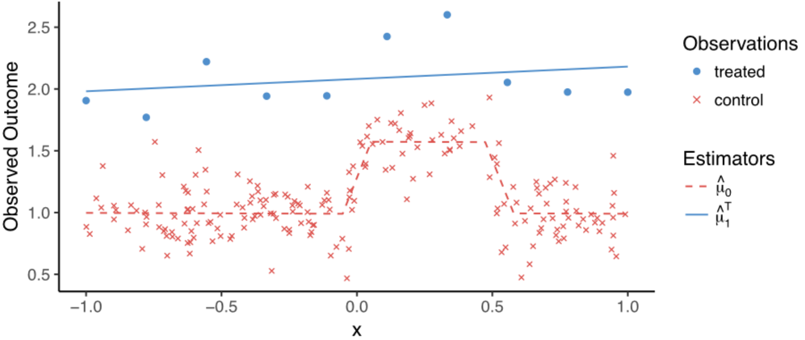
A tree estimator is **honest** iff the tree structure does not depend on the Y values used for leaf predictions:

- Purely random tree
- Wager and Athey (2017) definition of Causal Forest: Split the data and use half of it to span the tree

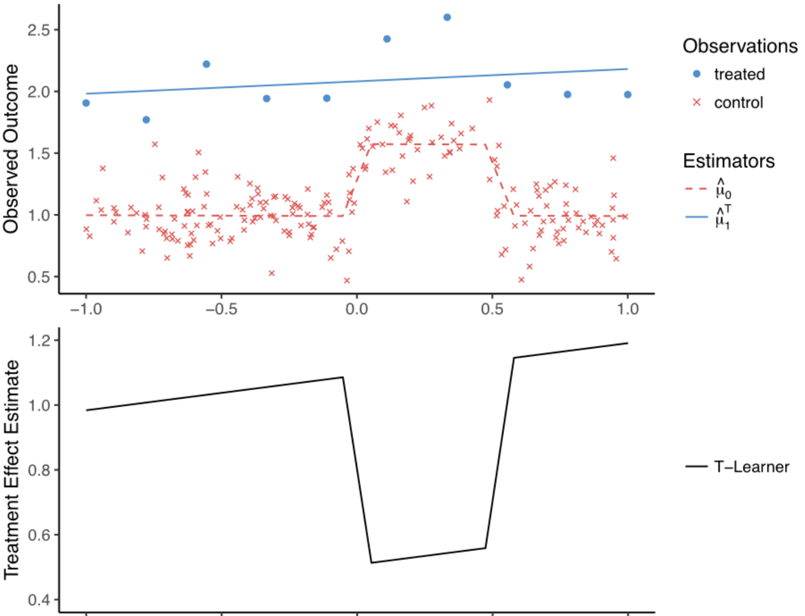
Motivating X



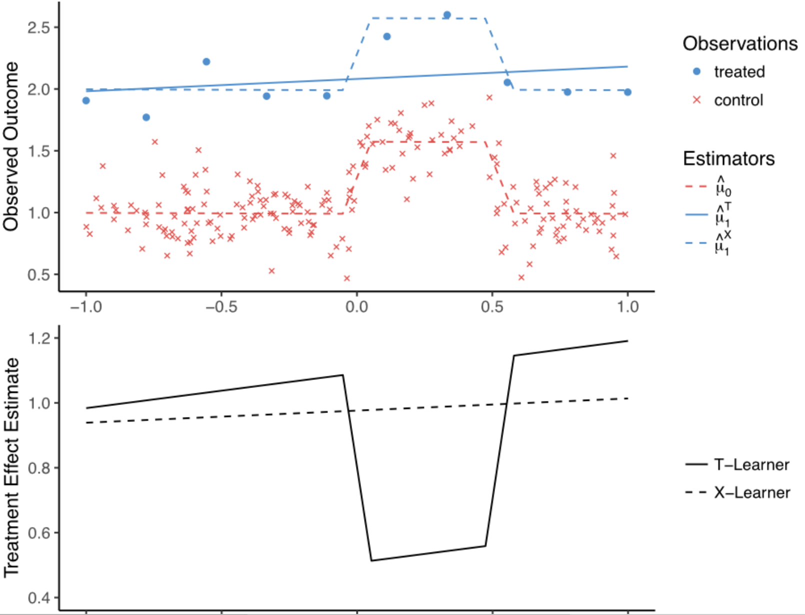
Motivating X



Motivating X



Motivating X



Formal defintion of the X–learner

$$\tau(x) = \mathbb{E}[Y(1) - Y(0)|X = x]$$
$$= \mathbb{E}[Y(1) - \mu_0(x)|X = x]$$

with $\mu_0(x) = \mathbb{E}[Y(0)|X = x]$.

X–learner

1.) Estimate the control response function,

$$\hat{\mu}_0(x) = \hat{\mathbb{E}}[Y(0)|X = x],$$

2.) Define the **pseudo residuals**,

$$\tilde{D}_i^1 := Y_i(1) - \hat{\mu}_0(X_i(1)),$$

3.) Estimate the CATE,

$$\hat{\tau}(x) = \hat{\mathbb{E}}[\tilde{D}^1|X = x].$$

X in algorithmic form

procedure $\text{X–}\textsc{Learner}(X, Y^{obs}, W)$

$\hat{\mu}_0 = M_1(Y^0 \sim X^0)$ ▷ Estimate response function

$\hat{\mu}_1 = M_2(Y^1 \sim X^1)$

$\tilde{D}_i^1 := Y_i^1 - \hat{\mu}_0(X_i^1)$ ▷ Compute pseudo residuals

$\tilde{D}_i^0 := \hat{\mu}_1(X_i^0) - Y_i^0$

$\hat{\tau}_1 = M_3(\tilde{D}^1 \sim X^1)$ ▷ Estimate CATE

$\hat{\tau}_0 = M_4(\tilde{D}^0 \sim X^0)$

$\hat{\tau}(x) = g(x)\hat{\tau}_0(x) + (1 - g(x))\hat{\tau}_1(x)$ ▷ Average

Algorithm 1: X–learner

Properties of the X–learner: Setup for Theory

A model for estimating the CATE

$$X \sim \lambda$$
$$W \sim \text{Bern}(e(X))$$
$$Y(0) = \mu_0(X) + \varepsilon(0)$$
$$Y(1) = \mu_1(X) + \varepsilon(1)$$

- This effect is in particular strong when μ_0 can be estimated very well
- Or when the error when estimating $\mu_0(x_i)$ is uncorrelated from the error when estimating $\mu_0(x_j)$ for $i \neq j$

Properties of the X–learner: Setup for Theory

A model for estimating the CATE

$$
\begin{aligned}
X &\sim \lambda \\
W &\sim \mathrm{Bern}(e(X)) \\
Y(0) &= \mu_0(X) + \varepsilon(0) \\
Y(1) &= \tau(X) + \mu_0(X) + \varepsilon(1)
\end{aligned}
$$

- If τ satisfies some regularity conditions (e.g. sparsity or smoothness), it can be directly exploited in the second base–learner
- This effect is in particular strong when μ_0 can be estimated very well
- Or when the error when estimating $\mu_0(x_i)$ is uncorrelated from the error when estimating $\mu_0(x_j)$ for $i \neq j$

Theorem 1

Theorem covers the case when estimating the base functions is not beneficial

Künzel, Sekhon, Bickel, Yu 2017

Assume we observe m control and n treatment units,

1.) Strong Ignorability holds: $(Y(0), Y(1)) \perp W|X \quad 0 < e(X) < 1$

2.) The treatment effect is linear, $\tau(x) = x^T\beta$

3.) There exists an estimator $\hat{\mu}_0$ with $\mathbb{E}[(\mu_0(x) - \hat{\mu}_0(x))^2] \leq C_x^0 m^{-a}$

Then the X-learner with $\hat{\mu}_0$ in the first stage, OLS in the second stage, achieves the parametric rate in n,

$$\mathbb{E}\left[\|\tau(x) - \hat{\tau}_X(x)\|^2\right] \leq C_x^1 m^{-a} + C_x^2 n^{-1}$$

If there are a many control units, such that $m \asymp n^{1/a}$, then

$$\mathbb{E}\left[\|\tau(x) - \hat{\tau}_X(x)\|^2\right] \leq 2C_x^1 n^{-1}$$

Theorem 2

Theorem covers the case when estimating the CATE function is not beneficial

Künzel, Sekhon, Bickel, Yu 2017

X–learner is minimax optimal for a class of estimators using KNN as the base leaner. Assume:

- Outcome functions are Lipschitz continuous
- CATE function has no simplification
- Features are uniformly distributed $[0, 1]^d$

The fastest possible rate of convergence for this class of problems is:

$$\mathcal{O}\left(\min(n_0, n_1)^{-\frac{1}{2+d}}\right)$$

- The speed of convergence is dominated by the size of the smaller assignment group
- In the worst case, there is nothing to learn from the other assignment group

Simulations: setup

1.) Simulate a 20–dimensional feature vector,

$$X_i \overset{iid}{\sim} \mathcal{N}(0, \Sigma).$$

with Σ being a correlation matrix with random off–diagonal elements between -0.2 and 0.2.

Simulations: setup

1.) Simulate a 20–dimensional feature vector,

$$X_i \overset{iid}{\sim} \mathcal{N}(0, \Sigma).$$

with Σ being a correlation matrix with random off–diagonal elements between -0.2 and 0.2.

2.) Create the potential outcomes according to

$$Y_i(1) = \mu_1(X_i) + \varepsilon_i(1)$$
$$Y_i(0) = \mu_0(X_i) + \varepsilon_i(0)$$

where $\varepsilon_i(1), \varepsilon_i(0) \overset{iid}{\sim} \mathcal{N}(0, 1)$.

Simulations: setup

1.) Simulate a 20–dimensional feature vector,

$$X_i \overset{iid}{\sim} \mathcal{N}(0, \Sigma).$$

with Σ being a correlation matrix with random off–diagonal elements between -0.2 and 0.2.

2.) Create the potential outcomes according to

$$Y_i(1) = \mu_1(X_i) + \varepsilon_i(1)$$
$$Y_i(0) = \mu_0(X_i) + \varepsilon_i(0)$$

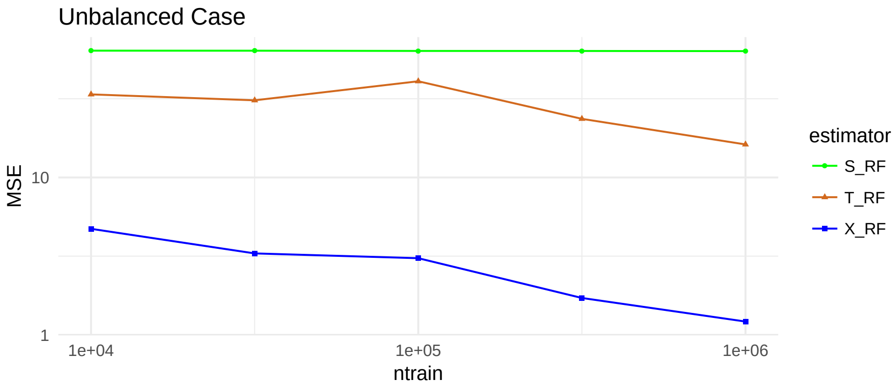
where $\varepsilon_i(1), \varepsilon_i(0) \overset{iid}{\sim} \mathcal{N}(0, 1)$.

3.) Simulate the treatment assignment according to

$$W_i \sim \text{Bern}(e(X_i))$$

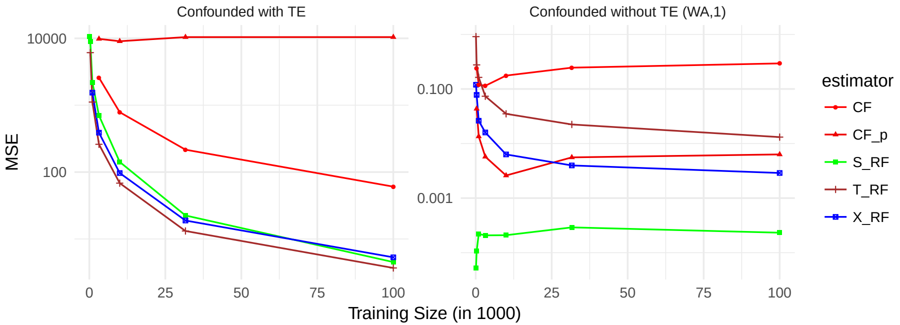
4.) Return $(X_i, W_i, Y(W_i))$.

The unbalanced case



$$\mu_0(x) = x^T\beta + 5 * 1(x1 > .5), \quad \text{with} \quad \beta \sim \text{Unif}\left([1,5]^d\right)$$
$$\mu_1(x) = \mu_1(x) + 8$$
$$e(x) = 0.01$$

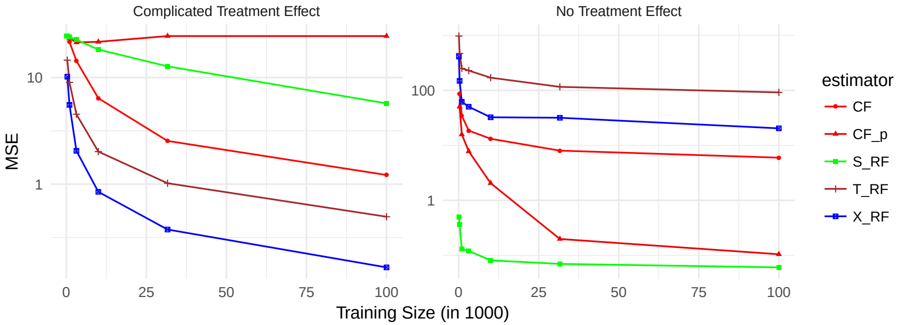
Resisting Confounding



$$\mu_1(x) = 2x_1 - 100x_2,$$
$$\mu_0(x) = 2x_1 + 2x_2,$$
$$e(x) = \max\left(.05, \min\left(.95, \frac{x_1}{2} + \frac{1}{4}\right)\right)$$

$$\mu_1(x) = 2x_1 - 1,$$
$$\mu_0(x) = 2x_1 - 1,$$
$$e(x) = \frac{1}{4}(1 + \beta_{2,4}(x_1))$$

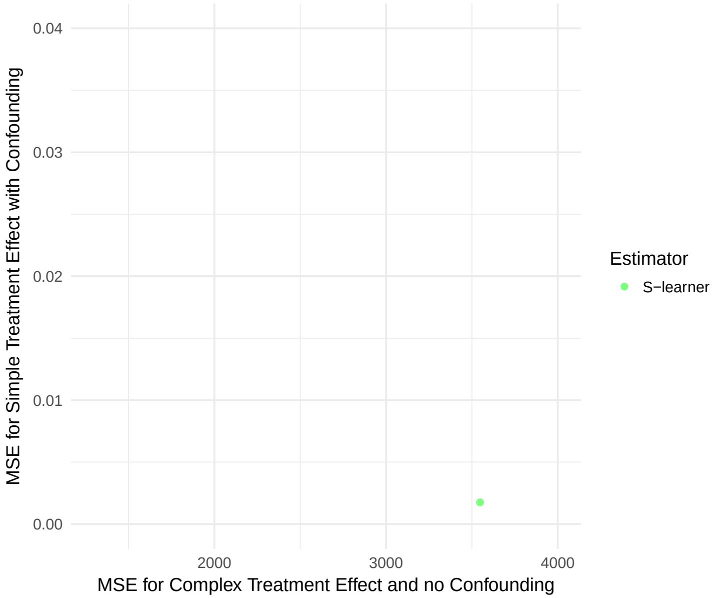
Complex versus Simple



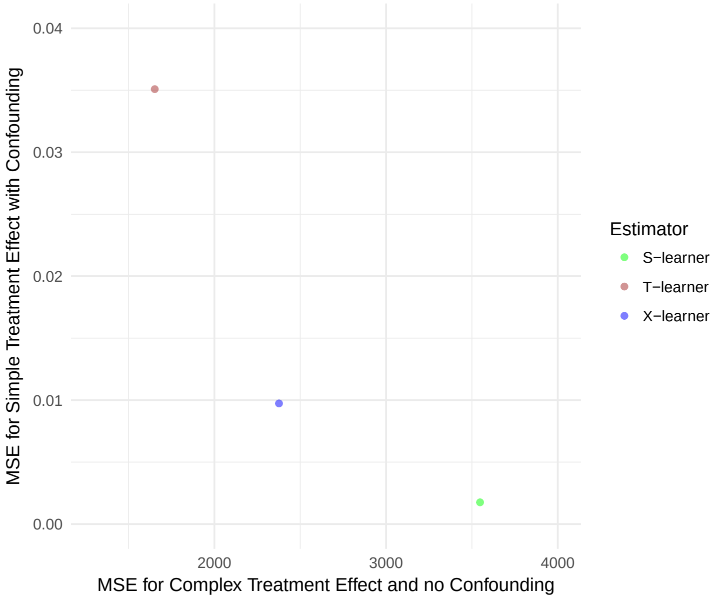
$$\mu_1(x) = x^T\beta_1, \text{ with } \beta_1 \sim \text{Unif}([1,30]^d)$$
$$\mu_0(x) = x^T\beta_0, \text{ with } \beta_0 \sim \text{Unif}([1,30]^d)$$
$$e(x) = .5$$

$$\mu_1(x) = x^T\beta, \text{ with } \beta \sim \text{Unif}([1,30]^d)$$
$$\mu_0(x) = \mu_1(x)$$
$$e(x) = .5$$

Adaptivity

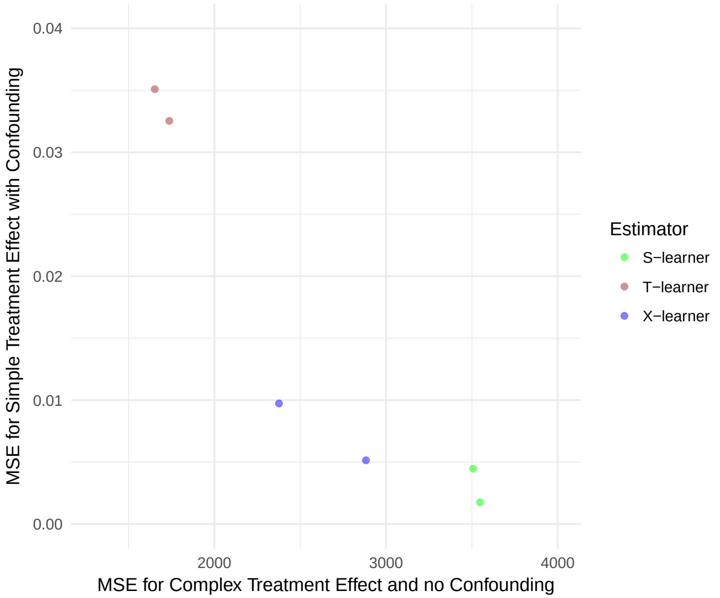

MSE for Simple Treatment Effect with Confounding
0.00
0.01
0.02
0.03
0.04
2000
3000
4000
MSE for Complex Treatment Effect and no Confounding
Estimator
S−learner

Adaptivity

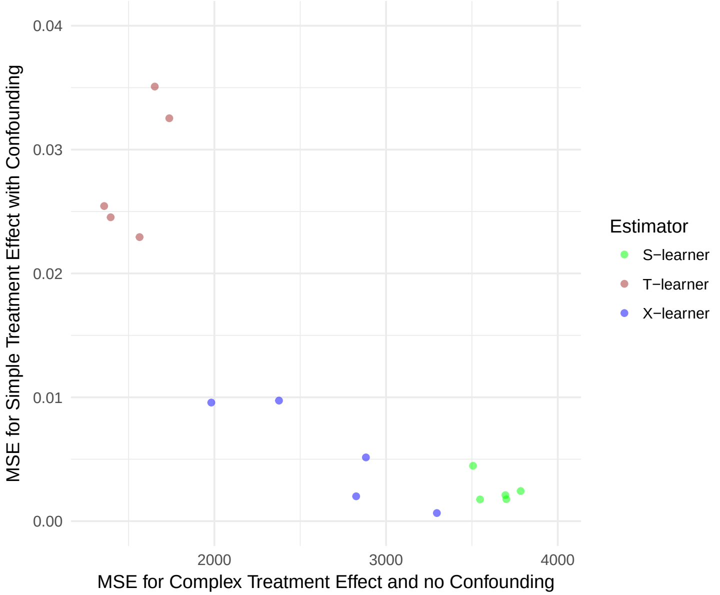


0.04
0.03
0.02
0.01
0.00
MSE for Simple Treatment Effect with Confounding
2000
3000
4000
MSE for Complex Treatment Effect and no Confounding
Estimator
S−learner
T−learner
X−learner

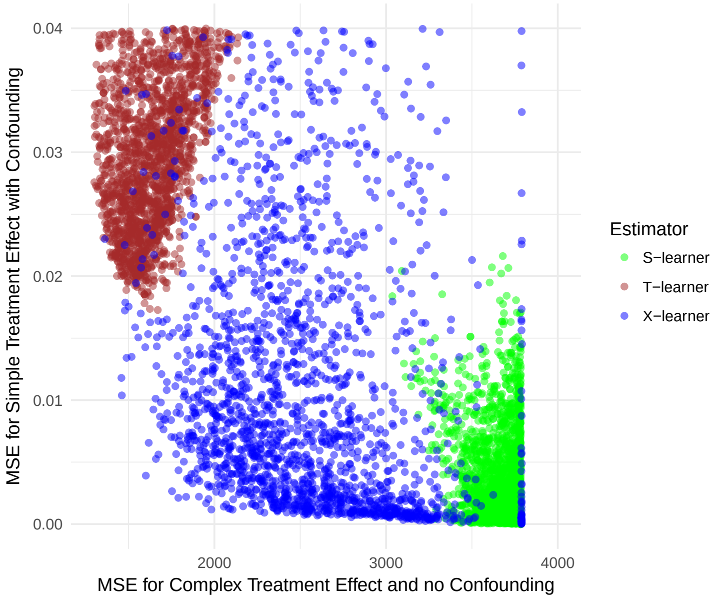
Adaptivity


0.04
0.03
0.02
0.01
0.00
MSE for Simple Treatment Effect with Confounding
2000
3000
4000
MSE for Complex Treatment Effect and no Confounding
Estimator
S−learner
T−learner
X−learner

Adaptivity


MSE for Simple Treatment Effect with Confounding
0.04
0.03
0.02
0.01
0.00
2000
3000
4000
MSE for Complex Treatment Effect and no Confounding
Estimator
S−learner
T−learner
X−learner

Adaptivity


0.04
0.03
0.02
0.01
0.00
MSE for Simple Treatment Effect with Confounding
2000
3000
4000
MSE for Complex Treatment Effect and no Confounding
Estimator
S−learner
T−learner
X−learner

Tuning

All meta–learners can be separated into several small regression problems, and we tune them separately using tuning methods which are specific for each of the learner

We have implemented a package combining the X–learner with honest Random Forests and it currently implements three tuning methods:

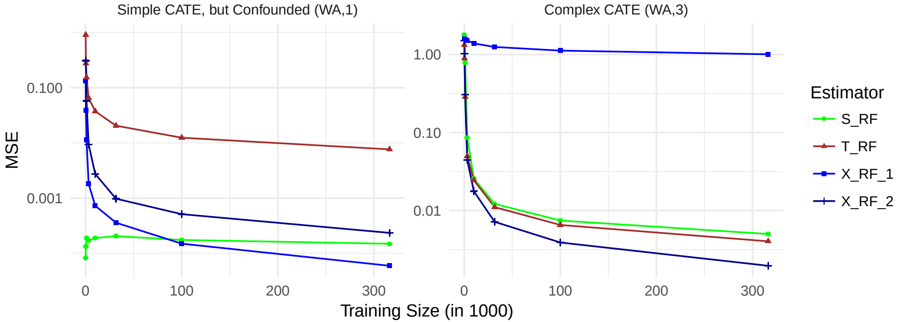
1.) Pre-specified tuning

2.) Gaussian Process

3.) Hyperband

Supplementary

Tuning Help



$$\mu_1(x) = 2x_1 - 1,$$
$$\mu_0(x) = 2x_1 - 1,$$
$$e(x) = \frac{1}{4}(1 + \beta_{2,4}(X_1))$$

$$\mu_1(x) = \zeta(X_1)\zeta(X_2),$$
$$\mu_0(x) = -\zeta(X_1)\zeta(X_2),$$
$$e(x) = 0.5,$$
$$\zeta(x) = \frac{2}{1 + e^{-12(x-1/2)}}$$

Confidence Intervals

Assume some regularity conditions on the distribution of (X_i, W_i, Y_i^{obs}), such as

1.) The components of the feature vectors X_i are independent

2.) The response functions μ_0 and μ_1 are Lipschitz continuous

Then a particular version of Causal Forest which satisfy honesty and some other criteria for the structure of the trees is

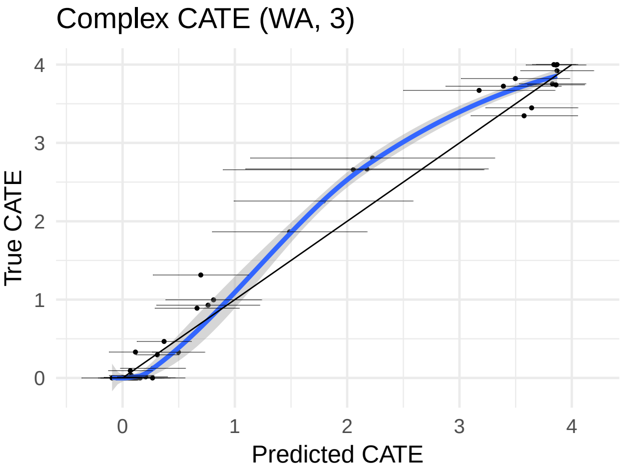
(i) asymptotically normal (Wager and Athey 2017)

(ii) and its variance can be estimated using the infinitesimal jackknife (Wager, Hastie and Efron 2014)

Confidence intervals for more than 5 dimensions turns out to be very difficult, and Wager and Athey report coverage in a sparse setting with only 12 dimensions of as low as 59%

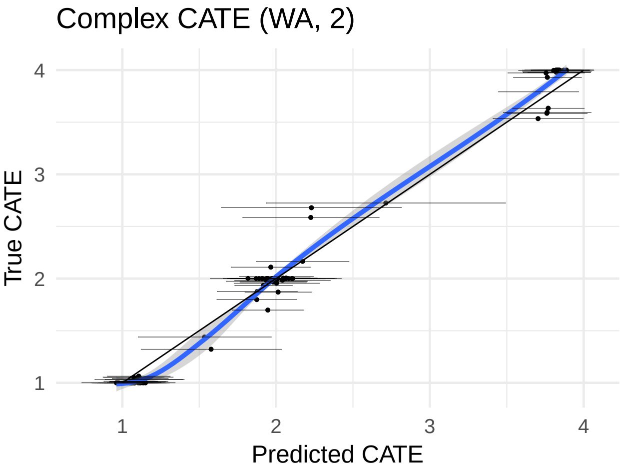
CF

Confidence Intervals



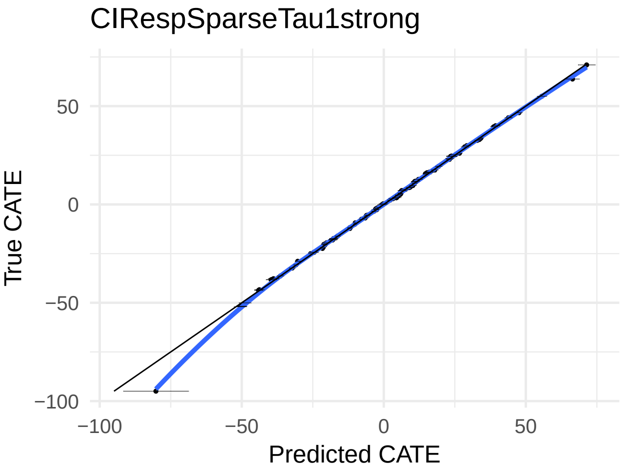
coverage $= 95.9\%$

Confidence Intervals



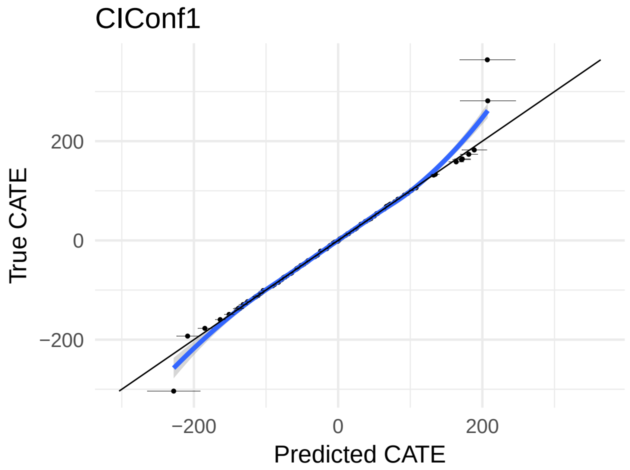
coverage = 96.3 %

Confidence Intervals



coverage = 93.3 %

Confidence Intervals

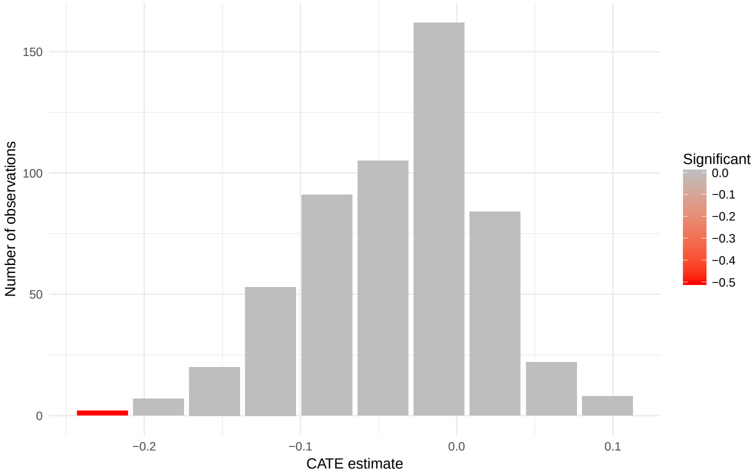


coverage = 95 %

Conclusion

- We expect more from our experiments than ever before
- We should protect the Type I error rate
- Power is a significant concern
- Somethings are easier to validate than others: experiments estimating average sample effects versus CATE
- Lots of observational data, massive push to use it: could be used to help estimate control outcomes
- Validation, validation, and validation

Persuasion: Abortion Policy (Broockman, Kalla, Sekhon, 2017)



Back to AbortionStigma

Individual Treatment Effects: Information Theory Bound

$Y_u \sim P = N(\mu, \sigma^2)$, and we want to predict a new Y_i.
Our expected risk with infinite data is:

$$\mathbb{E}(\mu - Y_i)^2 =$$

Individual Treatment Effects: Information Theory Bound

$Y_u \sim P = N(\mu, \sigma^2)$, and we want to predict a new Y_i.
Our expected risk with infinite data is:

$$\mathbb{E}(\mu - Y_i)^2 = \sigma^2 = \alpha$$

With one data point?

Individual Treatment Effects: Information Theory Bound

$Y_u \sim P = N(\mu, \sigma^2)$, and we want to predict a new Y_i.
Our expected risk with infinite data is:

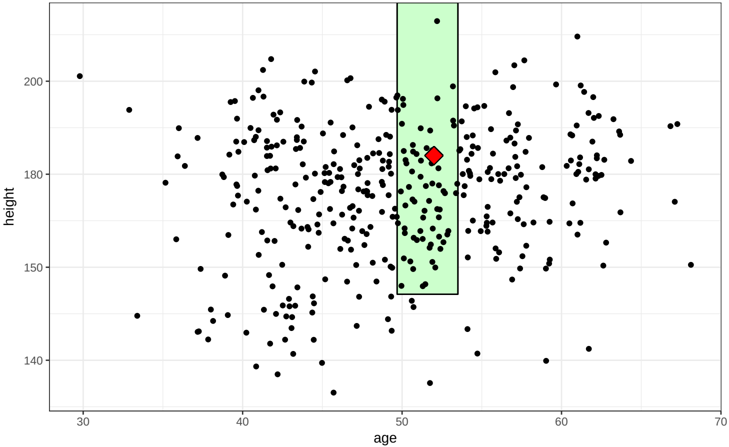
$$\mathbb{E}(\mu - Y_i)^2 = \sigma^2 = \alpha$$

With one data point?

$$\begin{aligned} E(Y_i - Y_u)^2 &= E(Y_i - \mu + Y_u - \mu)^2 \\ &= E(Y_i - \mu)^2 + E(Y_u - \mu)^2 \\ &= 2\sigma^2 \\ &= 2\alpha \end{aligned}$$

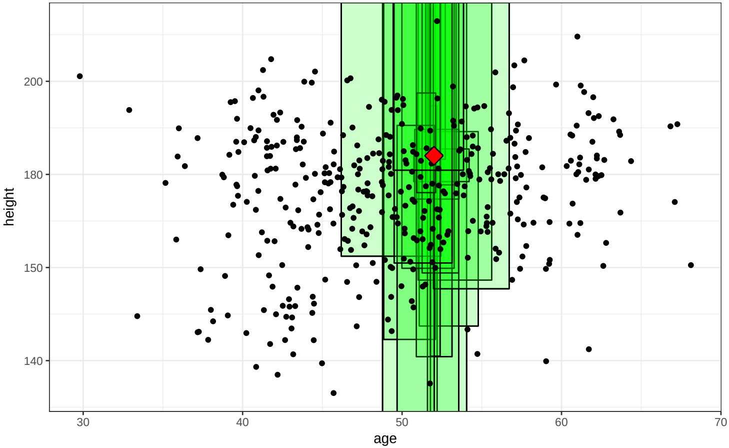
General results for Cover-Hart class, which is a convex cone (Gneiting, 2012)
Back to CATE

The averaging effect of Random Forest



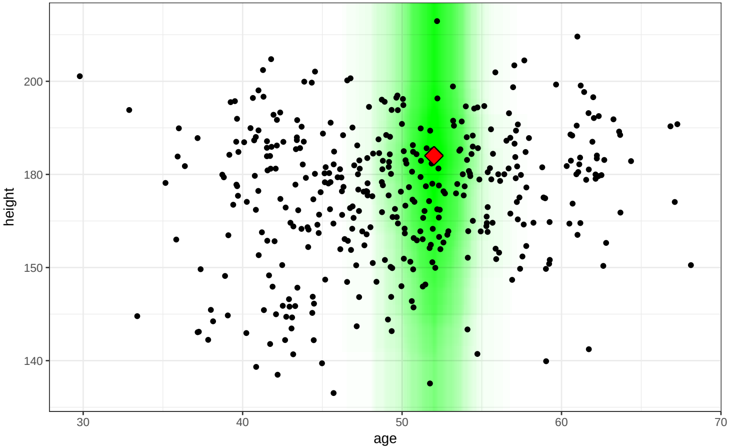
Back to RF

The averaging effect of Random Forest



Back to RF

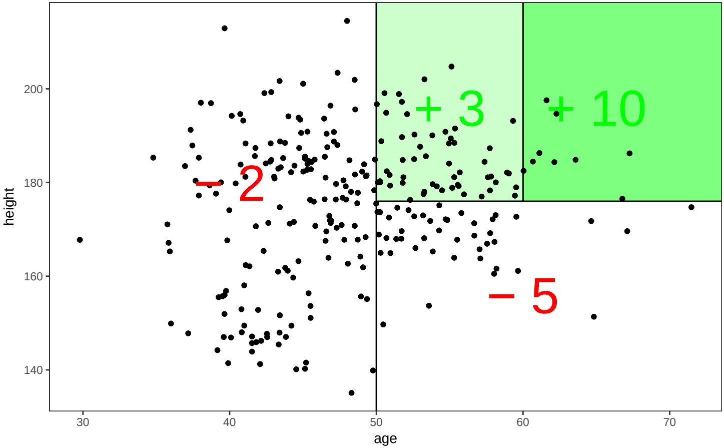
The averaging effect of Random Forest



Averaging leaves makes the weighing function of random forest smooth

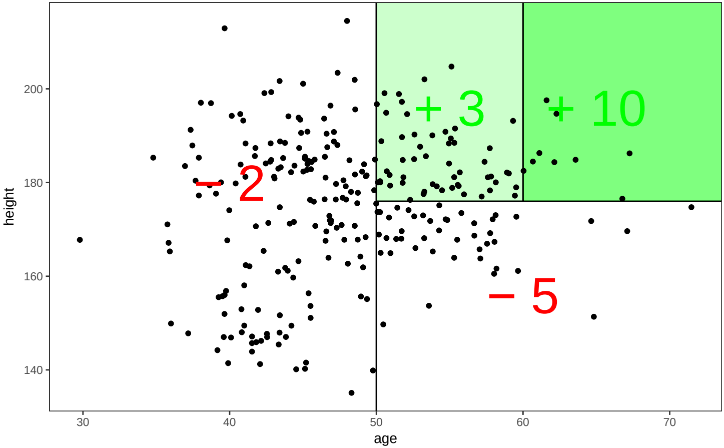
Back to RF

Honest versus adaptive fitting



Back to RF

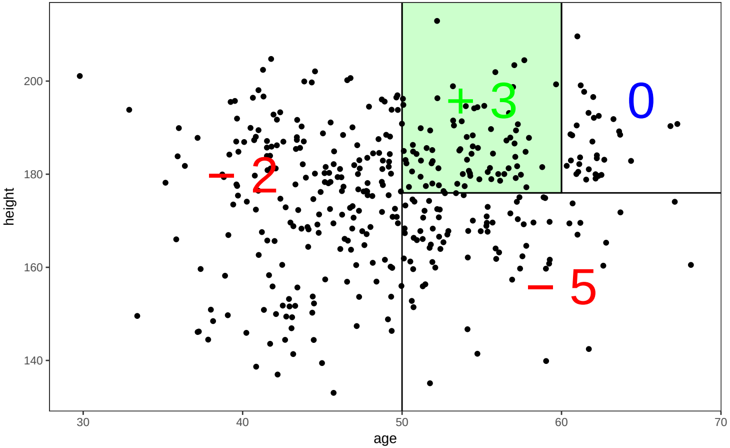
Honest versus adaptive fitting



Using the same data for the partitioning and the leaf estimates can lead to over-fitting

Back to RF

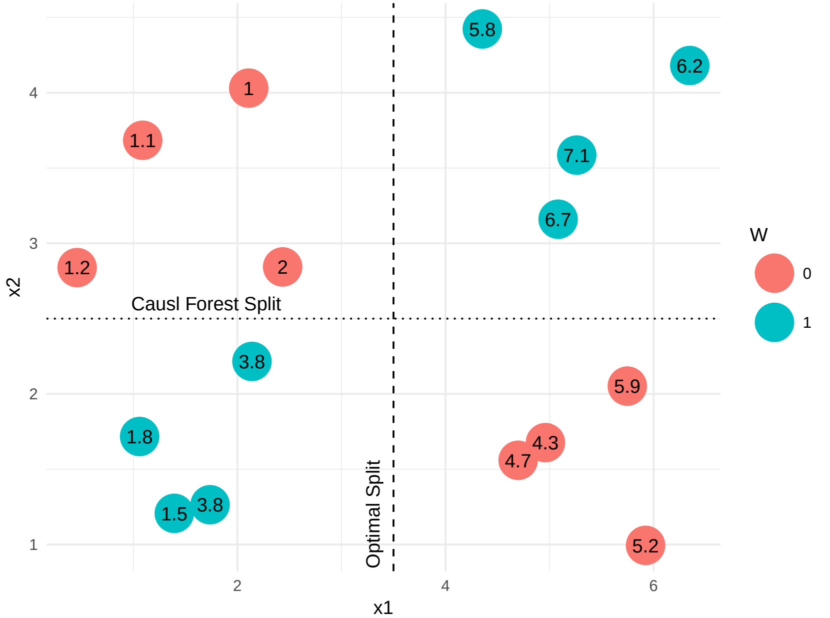
Honest versus adaptive fitting



Using the same data for the partitioning and the leaf estimates can lead to over-fitting

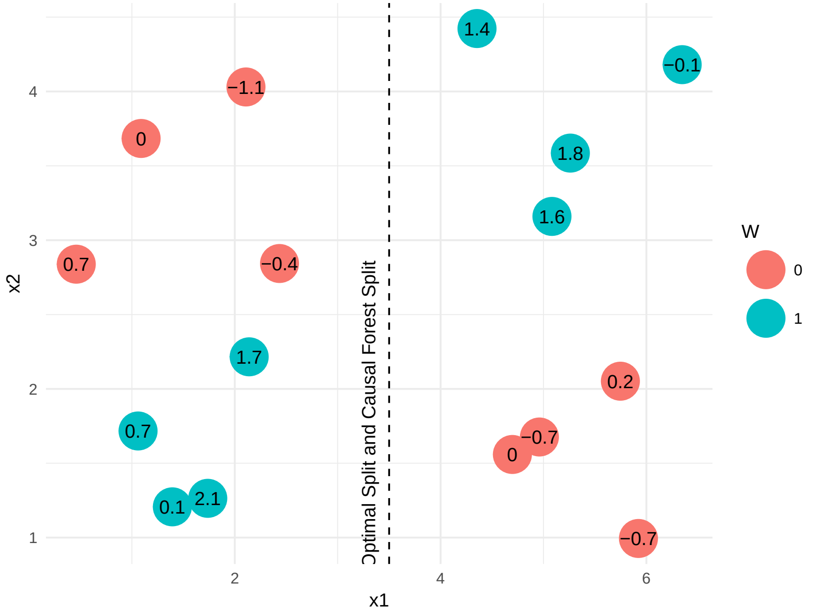
Back to RF

Causal Forest and "Confounding" back



Causal Forest and “Confounding”: after residualization

back



List of Hyperparameters

- **Ensemble–Strategy** — Specifies how the two estimators of the second stage should be aggregated
- **Relevant–Variable–Indices** — Indices of variables used as predictors
- **ntree** — Numbers of trees in the forest
- **mtry** — Numbers variables sampled at each node to be considered as possible splitting variables
- **min–node–size–spl** — Minimum node-size in the splitting set
- **min–node–size–ave** — Minimum node-size in the averaging set
- **splitratio** — Proportion of the training data used as the splitting set
- **replace** — Sample with or without replacement in the first stage
- **sample–fraction** — Fraction of samples at each bootstrap
- **middle–split** — Whether to split exactly between two observations or randomly anywhere between them

Back to Tuning