ECON 293/MGTECON 634: Machine Learning and Causal Inference

Susan Athey and Stefan Wager
Stanford University

Lecture 6: Heterogeneous Treatment Effects in Observational Studies

11 May 2018

The potential outcomes framework

For a set of i.i.d. subjects $i = 1, \ldots, n$, we observe a tuple (X_i, Y_i, W_i), comprised of

- A **feature vector** $X_i \in \mathbb{R}^p$,
- A **response** $Y_i \in \mathbb{R}$, and
- A **treatment assignment** $W_i \in \{0, 1\}$.

Following the **potential outcomes** framework (Neyman, 1923; Rubin, 1974), we posit the existence of quantities $Y_i^{(0)}$ and $Y_i^{(1)}$.

- These correspond to the response we **would have measured** given that the i-th subject received treatment ($W_i = 1$) or no treatment ($W_i = 0$).

The potential outcomes framework

For a set of i.i.d. subjects $i = 1, \ldots, n$, we observe a tuple (X_i, Y_i, W_i), comprised of

- A **feature vector** $X_i \in \mathbb{R}^p$,
- A **response** $Y_i \in \mathbb{R}$, and
- A **treatment assignment** $W_i \in \{0, 1\}$.

Our goal is to estimate the **conditional average treatment effect**

$$\tau(x) = \mathbb{E}\left[Y^{(1)} - Y^{(0)} \,\middle|\, X = x\right].$$

NB: In experiments, we only get to see $Y_i = Y_i^{(W_i)}$.

The potential outcomes framework

If we make no further assumptions, estimating $\tau(x)$ is not possible.

- We assume that we have measured enough features to achieve **unconfoundedness** (Rosenbaum and Rubin, 1983)

$$\left[\left\{Y_i^{(0)},\ Y_i^{(1)}\right\} \perp\!\!\!\perp W_i\right] \ \Big| \ X_i.$$

- When this assumption holds, methods based on matching or propensity score estimation are usually consistent.

Simple method: *k*-NN matching

Consider the k-**NN matching** estimator for $\tau(x)$:

$$\hat{\tau}\,(x) = \frac{1}{k}\sum_{\mathcal{S}_1(x)} Y_i - \frac{1}{k}\sum_{\mathcal{S}_0(x)} Y_i,$$

where $\mathcal{S}_{0/1}(x)$ is the set of k-nearest cases/controls to x. This is consistent given **unconfoundedness** and regularity conditions.

- **Pro:** Transparent asymptotics and good, robust performance when p is small.
- **Con:** Acute curse of dimensionality, even when $p = 20$ and $n = 20k$.

Simple method: k-NN matching

Consider the k-**NN matching** estimator for $\tau(x)$:

$$\hat{\tau}\left(x\right) = \frac{1}{k}\sum_{\mathcal{S}_1(x)} Y_i - \frac{1}{k}\sum_{\mathcal{S}_0(x)} Y_i,$$

where $\mathcal{S}_{0/1}(x)$ is the set of k-nearest cases/controls to x. This is consistent given **unconfoundedness** and regularity conditions.

Theorem. (Stone, 1977 + Rosenbaum and Rubin, 1983) Assume **unconfoundedness**, that conditional response functions are **Lipschitz**, and that we have **overlap**, i.e.,

$$\varepsilon \leq \mathbb{P}\left[W = 1 \,\middle|\, X = x\right] \leq 1 - \varepsilon \text{ for some } \varepsilon > 0.$$

Then, k-NN matching is **consistent**, provided that $k \to \infty$ and $k/n \to 0$.

Machine learning for HTE

Again assuming **unconfoundedness**,

$$\Big[\{Y_i(0),\ Y_i(1)\} \perp\!\!\!\perp W_i\Big] \quad \Big| \quad X_i,$$

we can also write the CATE function as

$$\begin{aligned}\tau(x) &= \mathbb{E}\left[Y_i(1) \,\middle|\, X_i = x\right] - \mathbb{E}\left[Y_i(0) \,\middle|\, X_i = x\right] \\ &= \mathbb{E}\left[Y_i \,\middle|\, X_i = x,\ W_i = 1\right] - \mathbb{E}\left[Y_i(0) \,\middle|\, X_i = x,\ W_i = 0\right] \\ &= \mu_{(1)}(x) - \mu_{(0)}(x).\end{aligned}$$

This representation is the starting point for several machine learning based HTE estimation strategies.

Machine learning for HTE

There are several **meta-learning** approaches for estimating HTEs via off-the-shelf machine learning tools.

The T-Learner fits separate models on the treated and controls.

1. Learn $\hat{\mu}_{(0)}(x)$ by predicting Y_i from X_i on the subset of observations with $W_i = 0$.
2. Learn $\hat{\mu}_{(1)}(x)$ by predicting Y_i from X_i on the subset of observations with $W_i = 1$.
3. Report $\hat{\tau}(x) = \hat{\mu}_{(1)}(x) - \hat{\mu}_{(0)}(x)$.

The S-Learner fits a single model to all the data.

1. Learn $\hat{\mu}(z)$ by predicting Y_i from $Z_i := (X_i, W_i)$ on all the data.
2. Report $\hat{\tau}(x) = \hat{\mu}((x, 1)) - \hat{\mu}((x, 0))$.

How robust are these methods to regularization bias?

Machine learning for HTE

There are several **meta-learning** approaches for estimating HTEs via off-the-shelf machine learning tools.

The X-Learner imputes unobserved outcomes, and uses them to learn the HTE.

1. Learn $\hat{e}(x)$ by predicting W_i from X_i.
2. Learn $\hat{\mu}_{(0)}(x)$ by predicting Y_i from X_i on the subset of observations with $W_i = 0$.
3. Define $\Delta_i(1) = Y_i - \hat{\mu}_{(0)}(X_i)$, and learn $\hat{\tau}_{(1)}(x)$ by predicting $\Delta_i(1)$ from X_i on those observations with $W_i = 1$.
4. Learn $\hat{\tau}_{(0)}(x)$ by swapping the roles of treated/controls.
5. Report $\hat{\tau}(x) = \hat{e}(x)\hat{\tau}_{(0)}(x) + (1 - \hat{e}(x))\hat{\tau}_{(1)}(x)$.

How robust are these methods to regularization bias?

Simulation Example: RCT

```
n = 4000; p = 10; treat.prob = 0.3
X = matrix(rnorm(n * p), n, p)
W = rbinom(n, 1, treat.prob)
TAU = 1/(1 + exp(-X[,3]))
Y = pmax(X[,1] + X[,2], 0) + W * TAU + rnorm(n)
```

Note in particular:

- This is a **randomized trial** with treatment fraction 0.3 (because treatment propensities don't depend on X).
- The treatment effect function is **simpler** than the main effect (which has interactions).

Simulation Example: T-learner

```
tf0 = regression_forest(X[W==0,], Y[W==0],
                        tune.parameters = TRUE)
tf1 = regression_forest(X[W==1,], Y[W==1],
                        tune.parameters = TRUE)
tf.preds.0 = predict(tf0, X.test)$predictions
tf.preds.1 = predict(tf1, X.test)$predictions
preds.tf = tf.preds.1 - tf.preds.0
```

Implement the T-learner via a **random forest**:

1. Learn $\hat{\mu}_{(0)}(x)$ by predicting Y_i from X_i on the subset of observations with $W_i = 0$.
2. Learn $\hat{\mu}_{(1)}(x)$ by predicting Y_i from X_i on the subset of observations with $W_i = 1$.
3. Report $\hat{\tau}(x) = \hat{\mu}_{(1)}(x) - \hat{\mu}_{(0)}(x)$.

Simulation Example: S-learner

```
sf = regression_forest(cbind(X, W), Y,
                       tune.parameters = TRUE)
pred.sf.0 = predict(sf, cbind(X.test, 0))$predictions
pred.sf.1 = predict(sf, cbind(X.test, 1))$predictions
preds.sf = pred.sf.1 - pred.sf.0
```

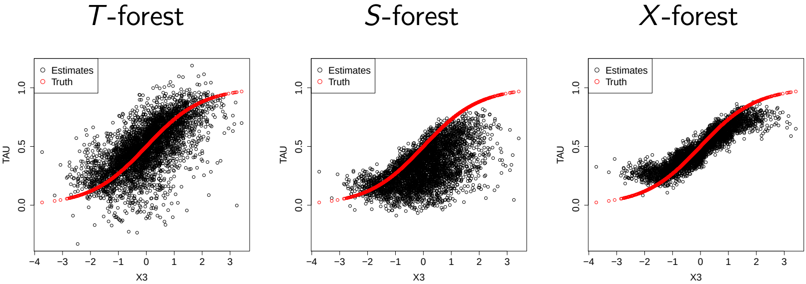
Implement the *S*-learner via a **random forest**:

1. Learn $\hat{\mu}(z)$ by predicting Y_i from $Z_i := (X_i, W_i)$ on all the data.
2. Report $\hat{\tau}(x) = \hat{\mu}((x, 1)) - \hat{\mu}((x, 0))$ on the test set.

Simulation Example: X-learner

```
tf0 = regression_forest(X[W==0,], Y[W==0],
                        tune.parameters = TRUE)
yhat0 = predict(tf0, X[W==1,])$predictions
xf1 = regression_forest(X[W==1,], Y[W==1]-yhat0,
                        tune.parameters = TRUE)
xf.preds.1 = predict(xf1, X.test)$predictions
tf1 = regression_forest(X[W==1,], Y[W==1],
                        tune.parameters = TRUE)
yhat1 = predict(tf1, X[W==0,])$predictions
xf0 = regression_forest(X[W==0,], yhat1-Y[W==0],
                        tune.parameters = TRUE)
xf.preds.0 = predict(xf0, X.test)$predictions
propf = regression_forest(X, W, tune.parameters = TRUE)
ehat.test = predict(propf, X.test)$predictions
preds.xf = (1 - ehat.test) * xf.preds.1 +
  ehat.test * xf.preds.0
```

Simulation Example: RCT



T-forest

S-forest

X-forest

The *T*- and *S*-learners have a hard time even approximating the treatment effect function.

- The *T*- and *S*-learners are only tuned to make accurate **predictions**, not to estimate **treatment effects**.
- In **randomized trials**, the *X*-construction can get at treatment effects directly.

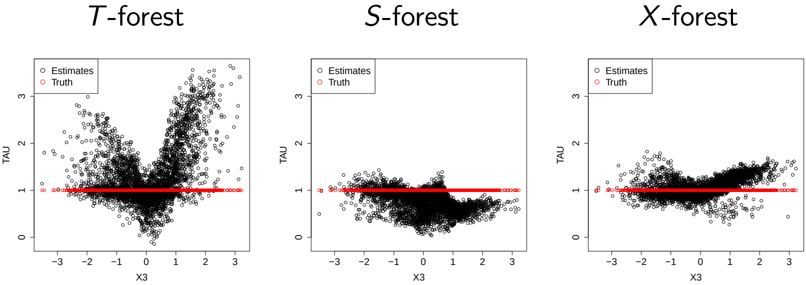
Simulation Example: Not an RCT

```
n = 4000; p = 10
X = matrix(rnorm(n * p), n, p)
W = rbinom(n, 1, 1 / (1 + exp(-X[,3])))
TAU = 1
Y = 2 * pmax(X[,1] + X[,2] + X[,3], 0) +
      W * TAU + rnorm(n)
```

Note in particular:

- This is **not** a randomized trial (because treatment propensities depend on X).
- The propensity function is **correlated** with the main effect.
- The treatment effect is **constant**.

Simulation Example: Not an RCT



T-forest

S-forest

X-forest

None of the *T*-, *S*-, or *X*-learners use **propensity scores** to guide treatment effect estimation.

- Makes methods vulnerable to confounding outside of RCTs.
- The *X*-learner does use the propensity score, but only in a minor role (for aggregation).

How can we **leverage good propensity score estimates** for accurate heterogeneous treatment effect estimation?

How can we **leverage good propensity score estimates** for accurate heterogeneous treatment effect estimation?

Outline:

- Review best practices for estimating **constant treatment effects** (i.e., via orthogonal moments).
- Apply this idea for orthogonalized HTE estimation with forests.
- Generalize this idea to loss-based **heterogeneous treatment effect** estimation.

Robinson's transformation and constant treatment effects

Suppose we assume a **constant treatment effect** τ, i.e.

$$\tau = \tau(x) = \mathbb{E}\left[Y_i(1) - Y_i(0) \,\middle|\, X_i = x\right] \text{ for all } x \in \mathcal{X}.$$

Given **unconfoundedness**, i.e.,

$$\left[\left\{Y_i^{(0)},\, Y_i^{(1)}\right\} \perp\!\!\!\perp W_i\right] \;\;\big|\;\; X_i,$$

we recover a **partially linear** model

$$\mathbb{E}\left[Y \,\middle|\, X = x,\, W = w\right] = \mu_{(0)}(x) + W\tau.$$

Our goal is to estimate τ. Note that this is not the same problem as estimating an **average treatment effect**, i.e., $\text{ATE} = \mathbb{E}\left[\tau(X)\right]$ for a potentially heterogeneous function $\tau(\cdot)$.

Robinson's transformation and constant treatment effects

Assume a **partially linear** model

$$\mathbb{E}\left[Y \,\middle|\, X = x,\, W = w\right] = \mu_{(0)}(x) + W\tau.$$

Robinson (1988) proposed the following estimator for τ:

1. Define the **propensity score** $e(x) = \mathbb{E}\left[W \,\middle|\, X = x\right]$, and estimate $\hat{e}(\cdot)$.
2. Define the **marginal response function** (i.e., marginalizing over W), $m(x) = \mathbb{E}\left[Y \,\middle|\, X = x\right]$, and estimate $\hat{m}(\cdot)$.
3. Define cross-fitted **residualized** treatments and responses $\widetilde{W}_i = W_i - \hat{e}^{(-i)}(X_i)$ and $\widetilde{Y}_i = Y_i - \hat{m}^{(-i)}(X_i)$.
4. Estimate $\hat{\tau} \leftarrow \text{OLS}(\widetilde{Y}_i \sim \widetilde{W}_i)$.

Theorem. Provided $\hat{e}(\cdot)$ and $\hat{m}(\cdot)$ are accurate enough, $\hat{\tau}$ has asymptotically optimal behavior.

Robinson's transformation and constant treatment effects

Theorem. Assume a **partially linear** model

$$\mathbb{E}\left[Y \,\middle|\, X = x,\, W = w\right] = \mu_{(0)}(x) + W\tau,$$

and that we estimate $\hat{\tau} \leftarrow \text{OLS}(\widetilde{Y}_i \sim \widetilde{W}_i)$ with **cross-fitting**.

Then, provided the regression adjustments are **accurate enough**,

$$\mathbb{E}\left[\left(\hat{e}(X) - e(X)\right)^2\right]^{\frac{1}{2}},\;\; \mathbb{E}\left[\left(\hat{m}(X) - m(X)\right)^2\right]^{\frac{1}{2}} = o\left(n^{-1/4}\right),$$

the resulting estimate $\hat{\tau}$ is $\sqrt{n}$-consistent, with

$$\sqrt{n}\left(\hat{\tau} - \tau\right) \Rightarrow \left(0,\, V\right).$$

Moreover, if $\text{Var}\left[Y \,\middle|\, X,\, W\right]$ is constant (i.e., under homoskedasticity), this estimator is **asymptotically efficient**.

Robinson's transformation and constant treatment effects

Theorem. Assume a **partially linear** model

$$\mathbb{E}\left[Y \,\middle|\, X = x,\, W = w\right] = \mu_{(0)}(x) + W\tau,$$

and that we estimate $\hat{\tau} \leftarrow \text{OLS}(\widetilde{Y}_i \sim \widetilde{W}_i)$ with **cross-fitting**.

Furthermore, even under heteroskedasticity, we can use standard heteroskedasticity-robust confidence intervals from the final OLS regression to build valid Gaussian **confidence intervals** for τ,

$$\tau \in \hat{\tau} \pm z_{1-\alpha/2}\,\hat{\sigma},$$

where $\hat{\sigma}$ is the error of the coefficient on $\widetilde{W}_i$ in the OLS.

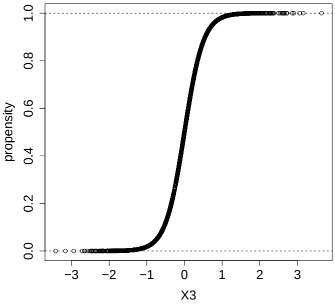
Constant vs average treatment effects

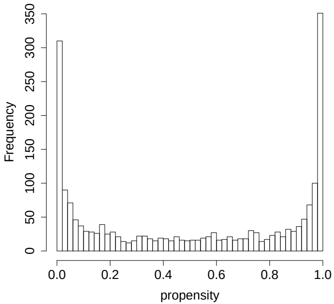
Estimating a **constant treatment effect** is not the same problem as estimating an **average treatment effect**, i.e., ATE $= \mathbb{E}\left[\tau(X)\right]$ for a potentially heterogeneous function $\tau(\cdot)$.

- To estimate an **average effect** we need reasonably accurate estimates of $\tau(x)$ everywhere.
- To estimate a **constant effect** we can opportunistically focus on areas with the most signal.

Constant vs average treatment effects

```
n = 2000; p = 6; TAU = 0.3
X = matrix(rnorm(n * p), n, p)
pscore = 1 / (1 + exp(-4 * X[,3]))
W = rbinom(n, 1, pscore)
Y = log(1 + exp((X[,1] + X[,2]) / 3)) +
  TAU * W + rnorm(n)
```





Constant vs average treatment effects

```
rf.y = regression_forest(X, Y, tune.parameters = TRUE)
m.hat = predict(rf.y)$predictions
tY = Y - m.hat

lr.w = glm(W ~ X, family = binomial)
e.hat = predict(lr.w, type = "response")
tW = W - e.hat

ols.fit = lm(tY ~ tW)
tau.hat = coef(ols.fit)["tW"]
tau.se = sqrt(vcovHC(ols.fit)["tW", "tW"])
paste("95% CI:", round(tau.hat, 3),
      "+/-",  round(1.96 * tau.se, 3))
```

If we **know** that the treatment effect is constant, we can accurately estimate it, and get 95% CI for τ of $\mathbf{0.322} \pm \mathbf{0.145}$.

Constant vs average treatment effects

```
rf.y = regression_forest(X, Y, tune.parameters = TRUE)
m.hat = predict(rf.y)$predictions
lr.w = glm(W ~ X, family = binomial)
e.hat = predict(lr.w, type = "response")

cf = causal_forest(X, Y, W, Y.hat = m.hat, W.hat = e.hat,
                   tune.parameters = TRUE)
ate.hat = average_treatment_effect(cf,
                           target.sample = "all")
paste("95% CI:", round(ate.hat["estimate"], 3),
      "+/-",  round(1.96 * ate.hat["std.err"], 3))
```

If we **don't know** that the treatment effect is constant, it's harder to estimate it, and we get 95% CI for τ of **0.56** ± **0.346**.

- The `average_treatment_effect` function does **augmented inverse-propensity weighted** estimation (Lecture 4).

Constant vs average treatment effects

Here, we have 2 different choices:

- Assume a **constant effect** τ, in which case accurate estimation of τ is possible.
- Estimate an **average effect** $\mathbb{E}\left[\tau(X)\right]$ in a way that's robust to heterogeneity, at the cost of precision.

Constant vs average treatment effects

Here, we have 2 different choices:

- Assume a **constant effect** τ, in which case accurate estimation of τ is possible.
- Estimate an **average effect** $\mathbb{E}\left[\tau(X)\right]$ in a way that's robust to heterogeneity, at the cost of precision.

What about **Robinson's method** to get "$\hat{\tau}$", but **without assuming a constant effect** $\tau = \tau(x)$? In this case,

$$\sqrt{n}\left(\hat{\tau} - \tau_e\right) \Rightarrow \mathcal{N}\left(0,\ V\right), \quad \tau_e = \frac{\mathbb{E}\left[e(X)(1 - e(X))\tau(X)\right]}{\mathbb{E}\left[e(X)(1 - e(X))\right]}.$$

In other words, there are **two ways** to justify Robinson's method:

- **Assume** a constant effect.
- **Relax** the target of inference to τ_e.

The `average_treatment_effect` does Robinson's method if we set `target.sample = "overlap"`.

Robinson's method for HTE

Recall the **nearest neighbors** estimator for $\tau(x)$:

$$\hat{\tau}\left(x\right) = \frac{\sum_{\{i \in \mathcal{S}(x) : W_i = 1\}} Y_i}{\left|\{i \in \mathcal{S}(x) : W_i = 1\}\right|} - \frac{\sum_{\{i \in \mathcal{S}(x) : W_i = 0\}} Y_i}{\left|\{i \in \mathcal{S}(x) : W_i = 0\}\right|},$$

where $\mathcal{S}(x) = \{i : |X_i - x| \leq \delta_n\}$.

- The key assumption underlying nearest neighbors methods is that observations with X_i **"close"** to x have the same conditional average treatment effect as x.

If we want to estimate a constant treatment effect on observations near x, why not use **Robinson's method** for it?

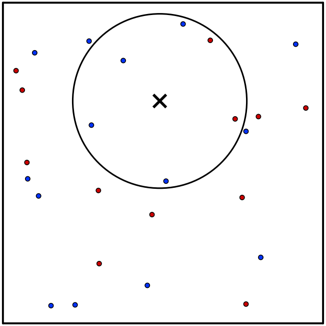
$$\hat{\tau}(x) \leftarrow \text{OLS}\left(\widetilde{Y}_i \sim \widetilde{W}_i,\ \texttt{subset:}\ W_i = 1\right),$$

where $\widetilde{W}_i = W_i - \hat{e}^{(-i)}(X_i)$, etc., rely on preliminary estimation.

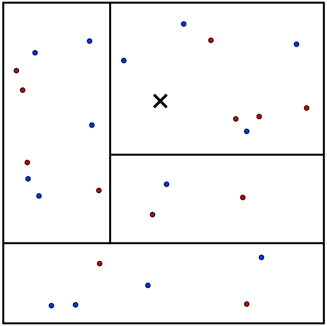
Robinson's method with forests

In order to present forest-based R-learning, we first review the **regression forest**. For now, we have data $(X_i,\ Y_i)$, want $\mu(x) = \mathbb{E}\left[Y \,\middle|\, X = x\right]$, and start with **neighborhood averaging**:

$$\hat{\mu}(x) = \frac{1}{|\mathcal{S}(x)|} \sum_{\{i : X_i \in \mathcal{S}(x)\}} Y_i.$$



k-NN neighborhood.



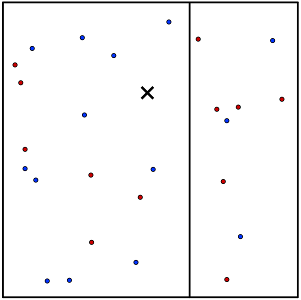
Tree-based neighborhood.

Regression trees and forests: Review

Trees recursively apply a **greedy splitting criterion**.

In the **regression case**, the CART (Breiman et al., 1984) is standard.

- Compute $\hat{y}$ by averaging data in left/right leaf.
- Split minimizes $\sum_i (y_i - \hat{y}(X_i))^2$.
- Equivalently, pick a split to maximize the **weighted difference** $n_L n_R (\hat{y}_L - \hat{y}_R)^2$.



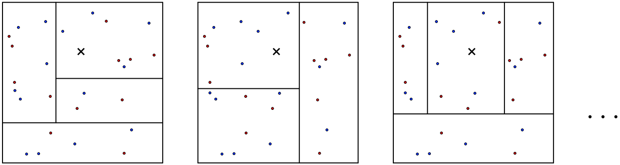
From trees to random forests (Breiman, 2001)

Suppose we have a training set $\{(X_i,\ Y_i)\}_{i=1}^n$, a test point x, and a tree predictor

$$\hat{\mu}(x) = T\left(x;\ \{(X_i,\ Y_i)\}_{i=1}^n\right).$$

Random forest idea: build and average many different trees T^*:

$$\hat{\mu}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b^*\left(x;\ \{(X_i,\ Y_i)\}_{i=1}^n\right).$$



From trees to random forests (Breiman, 2001)

Suppose we have a training set $\{(X_i,\ Y_i)\}_{i=1}^n$, a test point x, and a tree predictor

$$\hat{\mu}(x) = T\left(x;\ \{(X_i,\ Y_i)\}_{i=1}^n\right).$$

Random forest idea: build and average many different trees T^*:

$$\hat{\mu}(x) = \frac{1}{B} \sum_{b=1}^{B} T_b^*\left(x;\ \{(X_i,\ Y_i)\}_{i=1}^n\right).$$

We turn T into T^* by:

- Bagging / subsampling the training set (Breiman, 1996); this helps smooth over discontinuities (Bühlmann and Yu, 2002).
- Selecting the splitting variable at each step from m out of p randomly drawn features (Amit and Geman, 1997).

Aggregating causal estimates

For regression, natural to write a forest as an **average of trees**:

$$\hat{\mu}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b^*\left(x; \{(X_i,\ Y_i)\}_{i=1}^n\right).$$

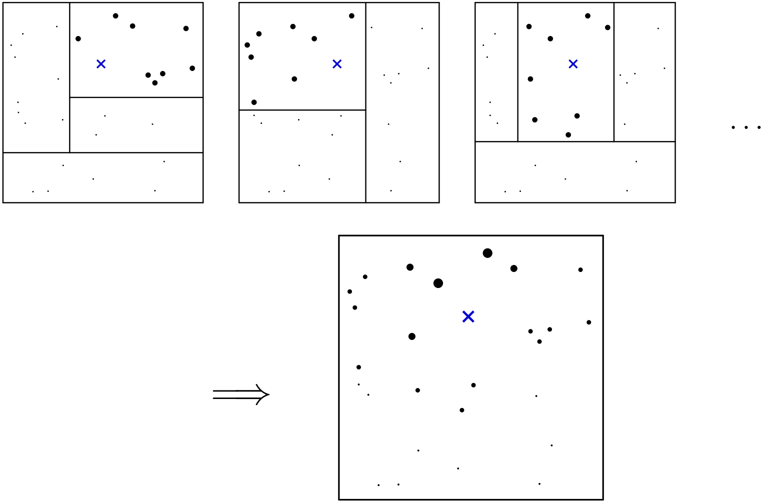
However, in causal forests, some leaves may be **highly variable**, and so averaging is undesirable.

A helpful alternative perspective is to view forests as weighting:

$$\hat{\mu}(x) = \frac{1}{B}\sum_{b=1}^{B}\sum_{i=1}^{n} Y_i \frac{1\left(Y_i \in L_b(x)\right)}{|L_b(x)|} = \sum_{i=1}^{n} Y_i \underbrace{\frac{1}{B}\sum_{b=1}^{B}\frac{1\left(Y_i \in L_b(x)\right)}{|L_b(x)|}}_{\alpha_i(x)}.$$

In other words, we understand random forests as a **data-adaptive "kernel"** with weights $\alpha_i(x)$.

The random forest kernel



Forests induce a kernel via **averaging tree-based neighborhoods**.

Aggregating causal estimates

Regression forests can also be understood as weighted estimators with a **forest kernel**,

$$\hat{\mu}\left(x\right) = \frac{1}{B}\sum_{b=1}^{B}\sum_{i=1}^{n} Y_i \frac{1\left(Y_i \in L_b(x)\right)}{\left|L_b(x)\right|} = \sum_{i=1}^{n} Y_i \underbrace{\frac{1}{B}\sum_{b=1}^{B}\frac{1\left(Y_i \in L_b(x)\right)}{\left|L_b(x)\right|}}_{\alpha_i(x)}.$$

This kernel-based approach naturally **extends** to the causal case. For a given test point x, we propose estimating $\tau(x)$ as follows:

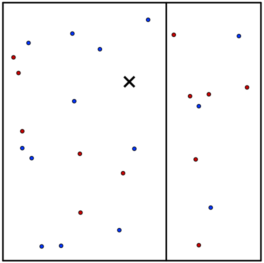
$$\hat{\tau}(x) \leftarrow \texttt{lm}\Bigg(\left(Y_i - \hat{m}^{(-i)}(X_i)\right) \sim \left(W_i - \hat{e}^{(-i)}(X_i)\right), \\ \texttt{weights} = \alpha_i(x)\Bigg).$$

Thus, forests provide us with a well-tuned, **data-adaptive kernel** for local estimation.

Recursive partitioning for causal effects

We now understand how to estimate constant treatment effects. How should this be reflected in a **splitting rule**?

As before, we seek to proceed **greedily**, and seek to maximize the amount of signal expressed in each split.



For each candidate "left-right" split (L, R), we do the following:

- Compute $\hat{\tau}_L$ and $\hat{\tau}_R$ **assuming homogeneous leaf-effects**:

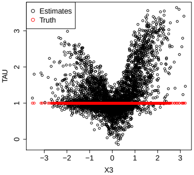
$$\hat{\tau}_L \leftarrow \texttt{lm}\left(\left(Y_i - \hat{m}^{(-i)}(X_i)\right) \sim \left(W_i - \hat{e}^{(-i)}(X_i)\right) : X_i \in L\right).$$

- Split to maximize the **weighted difference** $n_L n_R (\hat{\tau}_L - \hat{\tau}_R)^2$.
- In the **regression case**, this is equivalent to CART.

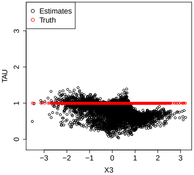
This is an instance of a **generalized random forest**.

Simulation example revisted: Not an RCT

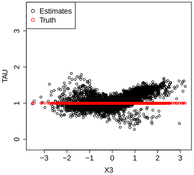
T-forest



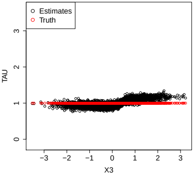
S-forest



X-forest



causal forest

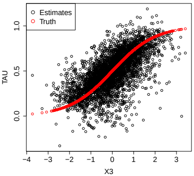


The ability of a causal forest to rely on a propensity score fit helps accuracy outside of RCTs.

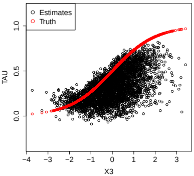
```
cf = causal_forest(X, Y, W, tune.parameters = TRUE)
preds.cf = predict(cf, X.test)$predictions
```

Simulation example revisted: RCT

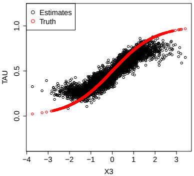
T-forest



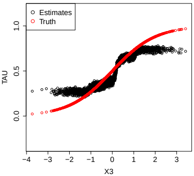
S-forest



X-forest



causal forest



In an RCT, both the *X*-forest and causal forest qualitatively fit the signal. Here, causal forest regularizes more aggressively, which helps slightly in RMSE $\sqrt{\mathbb{E}\left[(\hat{\tau}(X) - \tau(X))^2\right]}$.

	T-forest	*S*-forest	*X*-forest	causal forest
RMSE	0.173	0.246	0.087	0.074

Robinson's method for HTE: The general case

The fact that Robinson's method is consistent hinges on the fact that, with a **constant** treatment effect,

$$\mathbb{E}\left[Y \,\middle|\, X = x,\, W = w\right] = \mu_{(0)}(x) + W\tau,$$

we can also write τ as

$$\tau = \frac{\mathrm{Cov}\left[Y_i - m(X_i),\, W_i - e(X_i)\right]}{\mathrm{Var}\left[W_i - e(X_i)\right]}.$$

In a **non-parametric** setup, we can still apply the transformation conditionally:

$$\tau(x) = \frac{\mathrm{Cov}\left[Y_i - m(X_i),\, W_i - e(X_i) \,\middle|\, X_i = x\right]}{\mathrm{Var}\left[W_i - e(X_i) \,\middle|\, X_i = x\right]}.$$

An oracle estimator

In a **non-parametric** setup, apply the transformation conditionally:

$$\tau(x) = \frac{\operatorname{Cov}\left[Y_i - m(X_i),\, W_i - e(X_i) \,\middle|\, X_i = x\right]}{\operatorname{Var}\left[W_i - e(X_i) \,\middle|\, X_i = x\right]}$$

$$\implies \tau(\cdot) = \operatorname{argmin}_\tau \left\{ \mathbb{E}\left[\left(Y_i - m(X_i) - \tau(X_i)\left(W_i - e(X_i)\right)\right)^2 \right] \right\}.$$

If we knew $e(\cdot)$ and $m(\cdot)$, this suggests a natural **oracle learner**:

$$\tilde{\tau}(\cdot) = \operatorname{argmin}_\tau \left\{ \frac{1}{n} \sum_{i=1}^{n} \Big(\left(Y_i - m(X_i)\right) - \tau(X_i)\left(W_i - e(X_i)\right) \Big)^2 + \Lambda_n\left(\tau(\cdot)\right) \right\},$$

where $\Lambda_n(\cdot)$ is an appropriate **regularizer** (e.g., an L_1-penalty in high dimensions, or an RKHS-norm penalty non-parametrically).

Question: What about the **plug-in version** with $\hat{m}(\cdot)$ and $\hat{e}(\cdot)$?

Robinson's method for HTE: The general case

The previous argument suggests the following **two-step method**

1. Fit $\hat{m}(x)$ and $\hat{e}(x)$ via appropriate methods tuned for optimal **predictive accuracy**, then
2. Estimate treatment effects via a **cross-fit** plug-in estimator,

$$\hat{\tau}(\cdot) = \operatorname{argmin}_{\tau} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\left(Y_i - \hat{m}^{(-i)}(X_i) \right) - \left(W_i - \hat{e}^{(-i)}(X_i) \right) \tau(X_i) \right)^2 + \Lambda_n \left(\tau(\cdot) \right) \right\}.$$

We refer to this class of algorithms as "*R*-learning".

"Theorem:" For a large class of problems, $\hat{\tau}(\cdot)$ satisfies the same MSE **regret bounds** as the oracle $\tilde{\tau}(\cdot)$, provided $\hat{m}(\cdot)$ and $\hat{e}(\cdot)$ converge fast enough under squared error.

Example: The lasso

We run a simulation comparing **lasso-based** HTE estimators

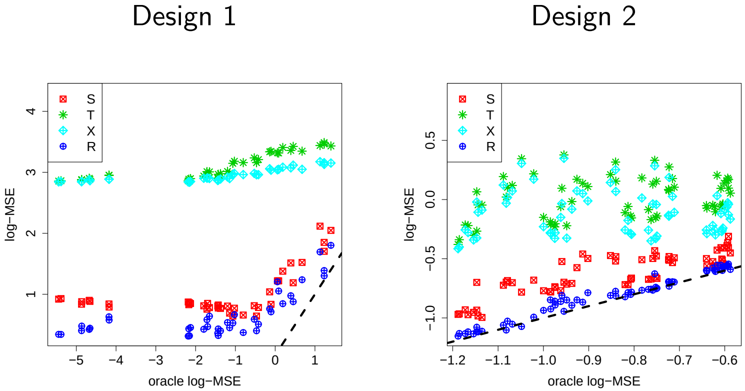
- S-lasso fits a **single model** (Imai and Ratkovic, 2013):

$$\text{argmin}\left\{\sum_{i=1}^{n}\left(Y_i - X_i b + (W_i - 1/2)X_i\delta\right)^2 + \lambda \left\|b\right\|_1 + \zeta \left\|\delta\right\|_1\right\}.$$

- T-lasso fits **two lassos** separately on the treated/controls.
- X-lasso of Künzel, Sekhon, Bickel and Yu (2017).
- R-lasso, the cross-fit plug-in version of the Robinson oracle.

All methods are fit by `glmnet` and tuned by **cross-validation**. The data-generating functions are non-parametric; we then run the lasso on a basis expansion.

Example: The lasso


Design 1

Design 2

We vary ambient dimension, sparsity, sample size, amount of overlap, and signal-to-noise ratio.

- **NB:** The quality of the *S*-learner is very sensitive to the class of methods used. The *S*-forest is terrible, but the *S*-lasso or *S*-boosting are at least somewhat stable.

The California GAIN Study

The California **Greater Avenues to Independence** (GAIN) program aims to reduce dependence on welfare and promote work among disadvantaged households.

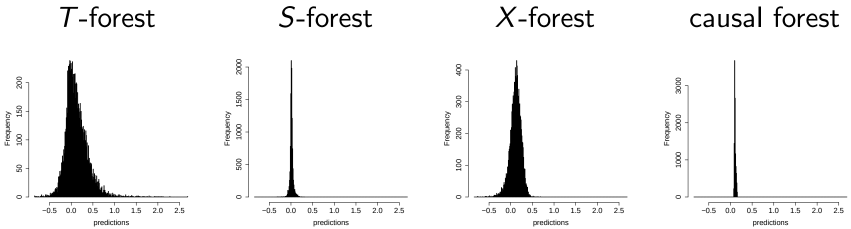
In 1988-1993, there was a **randomized evaluation** of GAIN; we want to use this to look for **heterogeneous treatment effects**. We have access to $p = 54$ covariates, including past income, demographics, etc.

Following Hotz, Imbens, and Klerman (2006), we focus on data from **Alameda**, **Los Angeles**, **Riverside** and **San Diego** counties.

Each county enrolled participants with a **different covariate mix**, and randomized subjects to treatment with **different probabilities**.

Once we remove county information, this is no longer a **randomized study**; however, Hotz et al present evidence that **unconfoundedness** still holds.

The California GAIN Study



T-forest

S-forest

X-forest

causal forest

The full dataset as 19,170 samples. We divided into a **training set** of size 8,000 for learning $\hat{\tau}(x)$, and a **test set** of size 11,170 for evaluation.

The above plot shows histograms for estimates $\hat{\tau}(X)$ on the test set. Which one is better?

Evaluating HTE estimators

Evaluating estimators $\hat{\tau}(\cdot)$ of $\tau(\cdot)$ is non-trivial. In contrast, suppose we have data $(X_i,\ Y_i)$, and want to evaluate the accuracy of $\hat{\mu}(x)$ as an estimator $\mu(x) = \mathbb{E}\left[Y \,\middle|\, X = x\right]$. If we have a **test set**, we can just look at prediction error,

$$\mathbb{E}\left[\sum_{\text{test}} \left(Y_i - \hat{\mu}(X_i)\right)^2\right] = \mathbb{E}\left[\sum_{\text{test}} \left(Y_i - \mu(X_i)\right)^2\right] + \mathbb{E}\left[\sum_{\text{test}} \left(\mu(X_i) - \hat{\mu}(X_i)\right)^2\right];$$

in expectation depends on the error of $\hat{\mu}(x)$ + irreducible error.

For treatment effect estimation, we'd want to compute

$$\sum_{\text{test}} \left(Y_i(1) - Y_i(0) - \hat{\tau}(X_i)\right)^2,$$

but of course can't do so with real data.

Evaluating HTE estimators

For treatment effect estimation, we'd want to compute

$$\sum_{\text{test}} \left(Y_i(1) - Y_i(0) - \hat{\tau}(X_i)\right)^2 ,$$

but of course can't do so with real data.

- In an observational study where both $m(\cdot)$ and $e(\cdot)$ are unknown, estimating the error of $\hat{\tau}(\cdot)$ (necessarilry?) requires estimating these nuisance components.
- However, when $e(x)$ is known, there are some "objective" evaluation methods; we'll discuss these now.

The GAIN study was **randomized by county**, and so we have access to the true propensity score; we will use this for evaluation (recall that county is masked during training).

	Riverside	Alameda	Los Angeles	San Diego
propensity	0.81	0.50	0.67	0.86

Transformed outcome validation

Recall that (this fact underlies consistency of the IPW estimator)

$$\mathbb{E}\left[\Gamma_i \,\middle|\, X_i = x\right] = \tau(x), \quad \Gamma_i = \frac{W_i Y_i}{e(X_i)} - \frac{(1 - W_i) Y_i}{1 - e(X_i)}.$$

It follows that we can use "transformed outcomes" Γ_i for evaluation:

$$\mathbb{E}\left[\sum_{\text{test}} \left(\Gamma_i - \hat{\tau}(X_i)\right)^2\right] = \mathbb{E}\left[\sum_{\text{test}} \left(\Gamma_i - \tau(X_i)\right)^2\right]$$

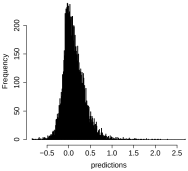
$$+ \,\mathbb{E}\left[\sum_{\text{test}} \left(\tau(X_i) - \hat{\tau}(X_i)\right)^2\right].$$

Let's try this! Concretely, report

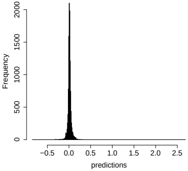
$$\widehat{L} = \frac{1}{|\text{test}|} \sum_{\text{test}} \left(\Gamma_i - \hat{\tau}(X_i)\right)^2.$$

Transformed outcome validation

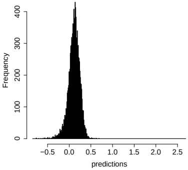
T-forest



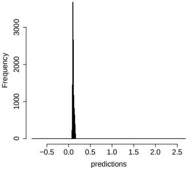
S-forest



X-forest



causal forest



The full dataset as 19,170 samples. We divided into a **training set** of size 8,000 for learning $\hat{\tau}(x)$, and a **test set** of size 11,170. We report $\widehat{L}$, along with a standard error estimate.

	T-forest	*S*-forest	*X*-forest	causal forest
error estimate	22.38	22.42	22.40	22.41
std err	1.75	1.73	1.73	1.73

Transformed outcome validation

We report $\widehat{L}$, along with a standard error estimate.

	T-forest	S-forest	X-forest	causal forest
error estimate	22.38	22.42	22.40	22.41
std err	1.75	1.73	1.73	1.73

All the numbers are very large, and very variable. What's going on?

$$\frac{1}{|\text{test}|}\sum_{\text{test}}\left(\Gamma_i - \hat{\tau}(X_i)\right)^2$$
$$= \underbrace{\frac{1}{|\text{test}|}\sum_{\text{test}}\Gamma_i^2}_{22.43} - \underbrace{\frac{1}{|\text{test}|}\sum_{\text{test}}\Gamma_i\hat{\tau}(X_i)}_{0.04} + \underbrace{\frac{1}{|\text{test}|}\sum_{\text{test}}\hat{\tau}(X_i)^2}_{0.01}.$$

We're mostly just measuring the "shared" component!

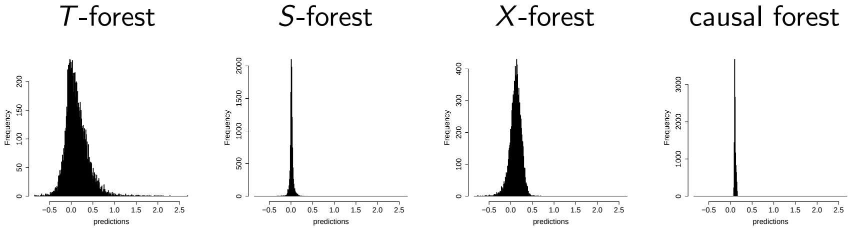
Transformed outcome model comparison

We can alleviate this problem by comparing two treatment effect estimates. Let $\hat{\tau}_0(x)$ be some **baseline** treatment effect estimator; then

$$\begin{aligned}\frac{1}{|\text{test}|}\sum_{\text{test}}\left(\Gamma_i - \hat{\tau}(X_i)\right)^2 - \frac{1}{|\text{test}|}\sum_{\text{test}}\left(\Gamma_i - \hat{\tau}_0(X_i)\right)^2 \\ = \frac{1}{|\text{test}|}\sum_{\text{test}}\left(-2\Gamma_i\left(\hat{\tau}(X_i) - \hat{\tau}_0(X_i)\right) + \hat{\tau}^2(X_i) - \hat{\tau}_0^2(X_i)\right).\end{aligned}$$

One simple choice is to use a **constant baseline** $\hat{\tau}_0(x) = \hat{\tau}_0$, obtained via Robinson's method (this would be the optimal estimator if the treatment effect were actually constant).

Transformed outcome model comparison



The full dataset as 19,170 samples. We divided into a **training set** of size 8,000 for learning $\hat{\tau}(x)$, and a **test set** of size 11,170. We report improvement over baseline, along with a s.e. estimate.

	T-forest	*S*-forest	*X*-forest	causal forest
error comparison	-0.024	0.019	-0.004	-0.001
std err	0.034	0.009	0.014	0.002

Better, but still too noisy to tell the difference!

Transformed outcome model comparison

On this dataset, we cannot measure improvement over a constant baseline in terms of MSE on $\tau(x)$.

	T-forest	S-forest	X-forest	causal forest
error comparison	-0.024	0.019	-0.004	-0.001
std err	0.034	0.009	0.014	0.002

Should we be disappointed? Try stratifying based on whether $\hat{\tau}(X_i)$ is smaller/larger than the median treatment effect estimate, and use county information to evaluate sub-group ATEs on the test set.

	small $\hat{\tau}(X_i)$	large $\hat{\tau}(X_i)$
subgroup ATE	0.241	0.117
std err	0.063	0.045

Finding good subgroups is easier than accurate $\tau(\cdot)$ estimation?

Estimating HTEs: Recap

Accurate estimation of heterogeneous treatment effects often requires large sample sizes, and methods are still evolving.

Some high-level thoughts:

- Meta-learners are helpful for focusing of treatment effects. Be skeptical of methods that don't purposefully regularize the CATE function estimate.
- Orthogonal moments matter for reducing confounding. In observational studies, be skeptical of methods that don't use propensity scores to reduce bias.
- Validation is hard, and it's important to keep the core scientific question in mind.