ECON 293/MGTECON 634:
Machine Learning and Causal Inference

Susan Athey and Stefan Wager
Stanford University

Lecture 4b: Heterogeneous Treatment Effect Estimation

27 April 2018

Motivation: Oracle analysis

	window shopper	bargain hunter	already convinced
no coupon	1/81	5/38	15/19
coupon	0/68	15/32	17/22

Results from a **randomized trial**. In each group, we show:

[number of purchases]/[number of customers].

This table would be possible if we knew customer types **a-priori**...

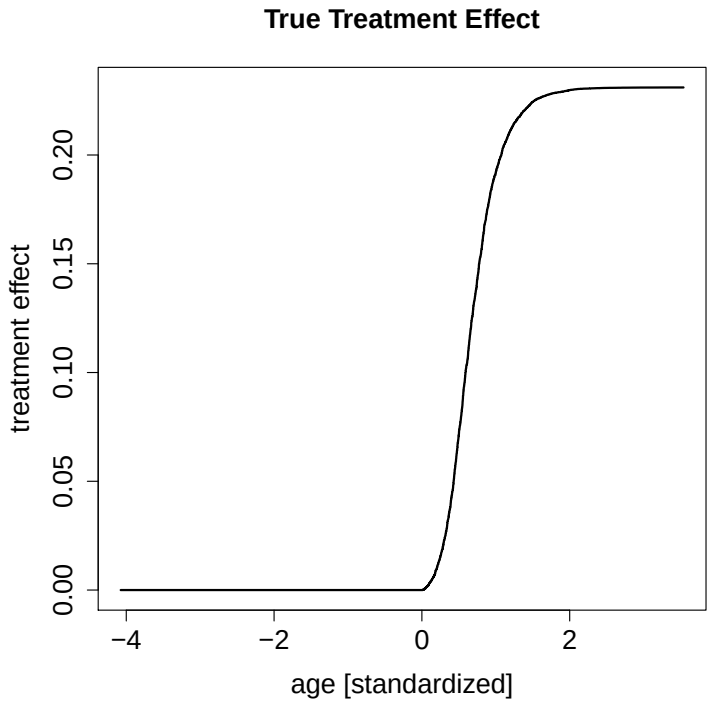
How can we model **treatment heterogeneity** when we don't know interesting customer types a-priori?

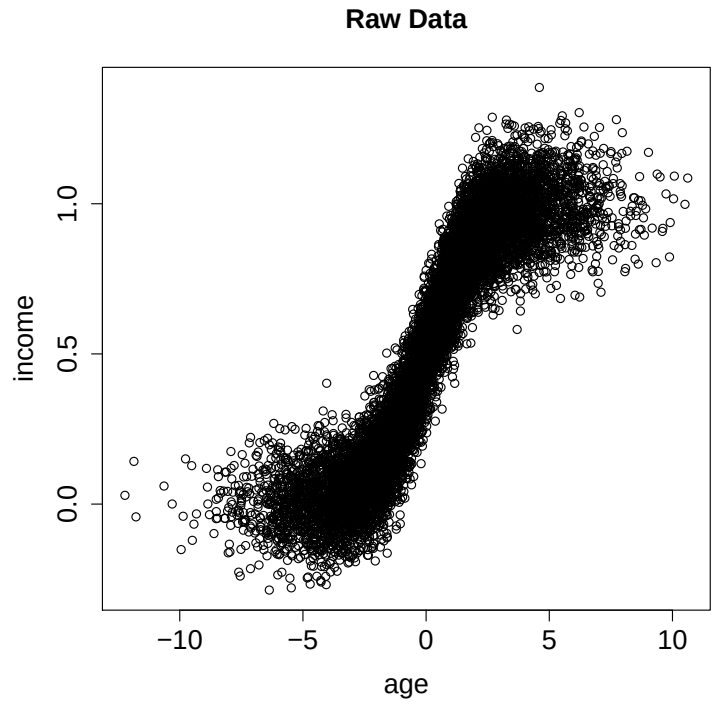
```
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)       0.039595   0.090401   0.438  0.66139
treatment         0.160037   0.130019   1.231  0.21837
age               0.006696   0.022174   0.302  0.76268
income           -0.080267   0.171391  -0.468  0.63955
treatment:age     0.039993   0.031633   1.264  0.20612
treatment:income  0.677001   0.246580   2.746  0.00604 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Results from running a **logistic regression** in `R` (with `glm`), on $n = 10,000$ observations. Which of the following must be true?

1. "Age" *is not* associated with treatment effect?
2. "Income" *is* associated with treatment effect?

NB: This is a simulated examples.





Neither (1) nor (2) is true. The treatment effect is a function of age only.

The potential outcomes framework

For a set of i.i.d. subjects $i = 1, ..., n$, 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)

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

- 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 $\tau(x)$:

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

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

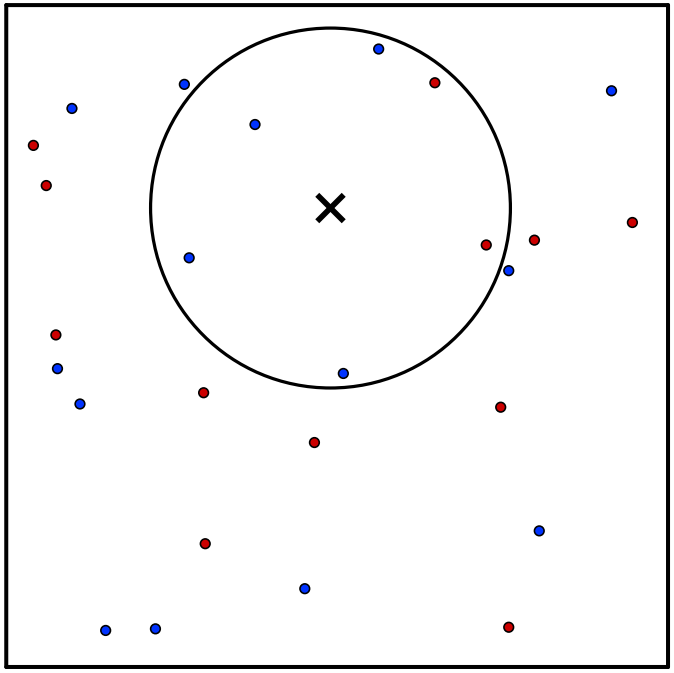
Theorem. (Stone, 1977 + Rosenbaum and Rubin, 1983) Assume **unconfoundedness**, that conditional response functions are **Lipschitz**, and that we have **overlap**, i.e.,

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

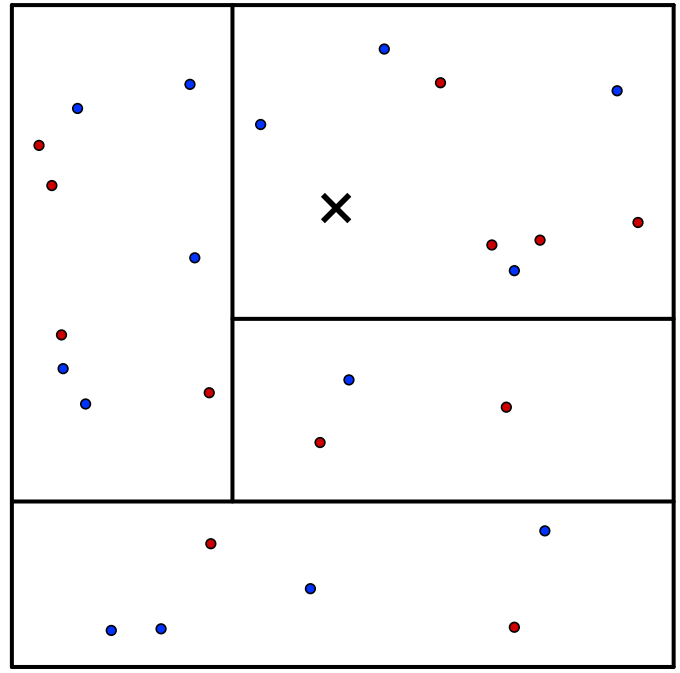
Then, k-NN matching is **consistent**, provided that $k \to \infty$ and $k/n \to 0$.

Making k-NN matching adaptive

A **causal tree** (Athey and Imbens, 2015) defines neighborhoods for matching based on **recursive partitioning** (Breiman, Friedman, Olshen, and Stone, 1984).



Euclidean neighborhood, for k-NN matching.



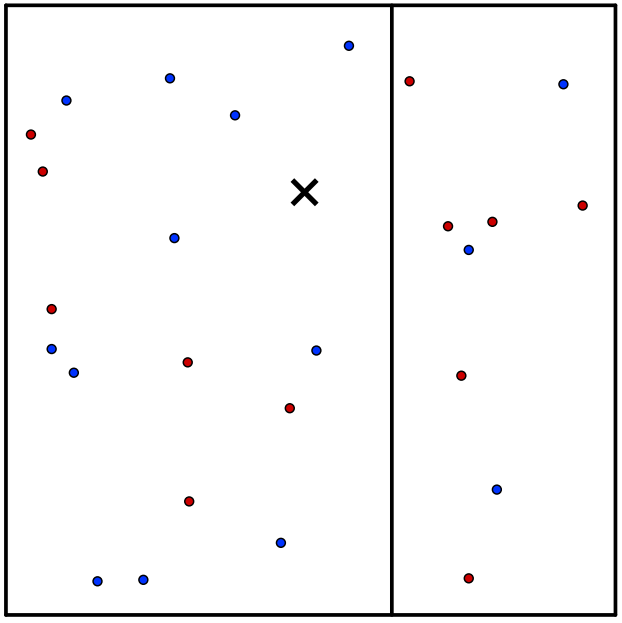
Tree-based neighborhood.

Interlude: How to place splits?

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 to minimize $\sum_i (y_i - \hat{y}_i)^2$.
- Equivalently, pick a split to **maximize the variance** $\widehat{\text{Var}}[\hat{y}_i]$.

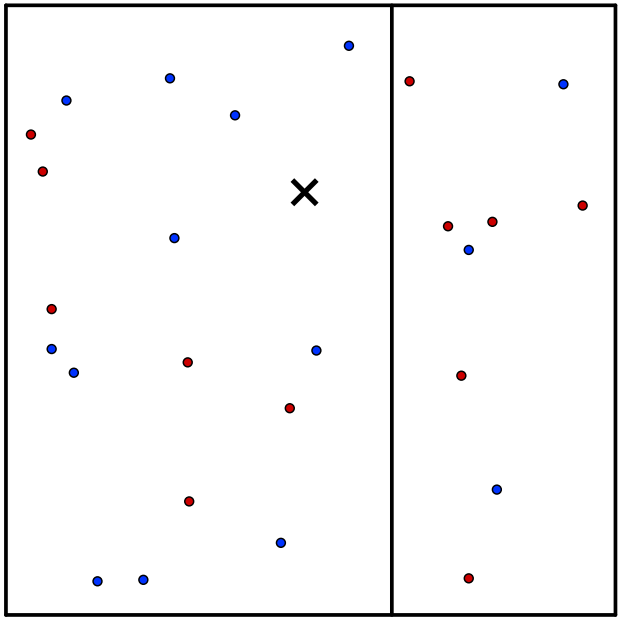


Interlude: How to place splits?

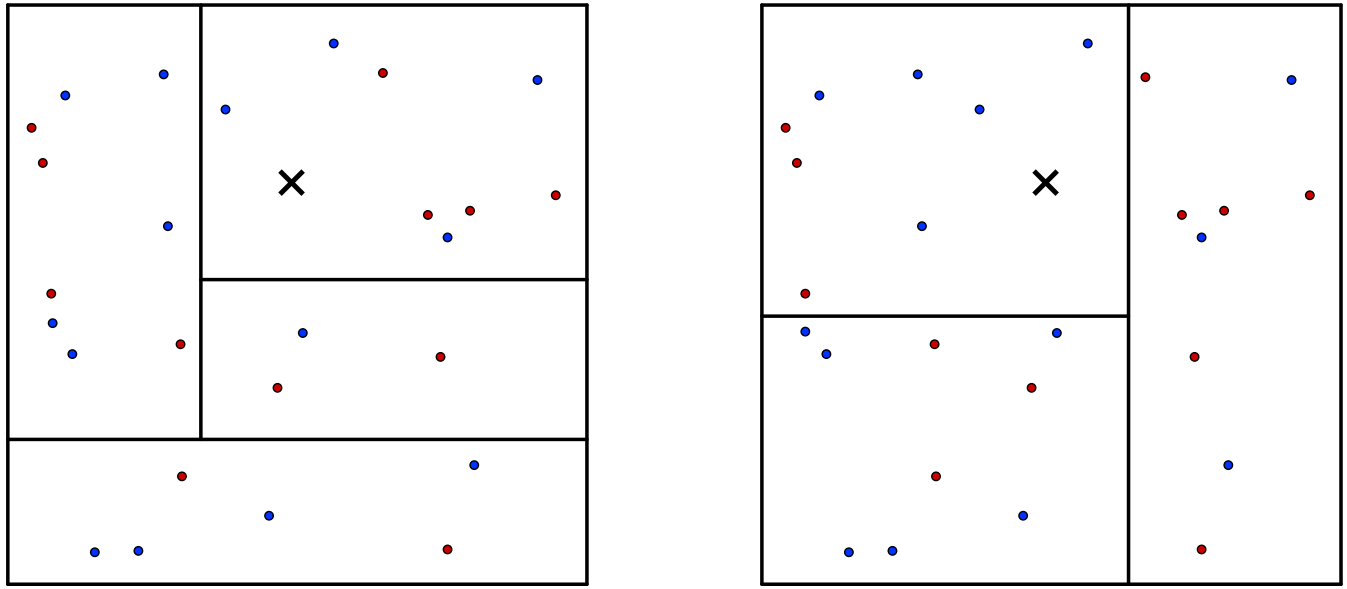
Trees recursively apply a **greedy splitting criterion**.

In the **treatment effect estimation case**, we use the max variance rule of Athey and Imbens (2015).

- Compute $\hat{\tau}$ in left/right leaf by considering data in each leaf as in a randomized experiment.
- Split to **maximize the variance** of the estimates $\widehat{\text{Var}}[\hat{\tau}_i]$.



Which tree should you prefer?



Sometimes **random fluctuations** in the training data can **considerably alter** the fitted tree.

From trees to random forests (Breiman, 2001)

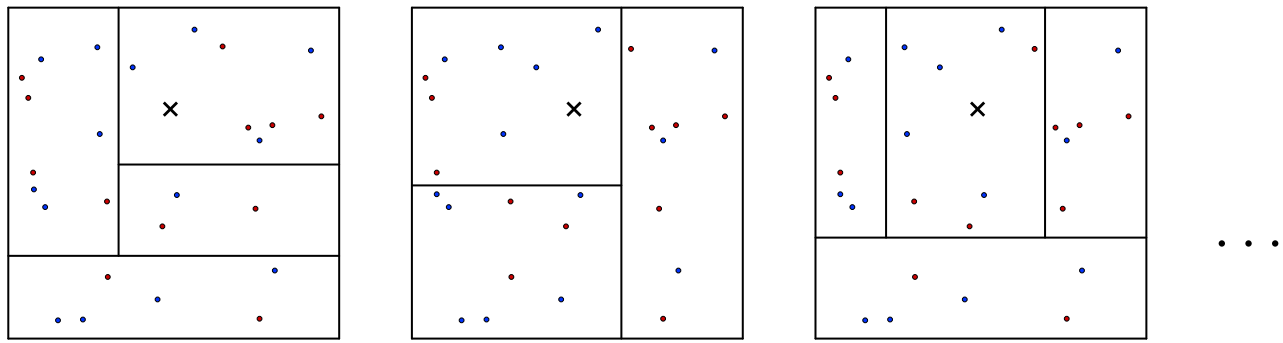
Suppose we have a training set $\{(X_i,\, Y_i,\, W_i)\}_{i=1}^n$, a test point x, and a tree predictor

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

Single trees are flexible and interpretable, but not always accurate.

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

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



From trees to random forests (Breiman, 2001)

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

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

Single trees are flexible and interpretable, but not always accurate.

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

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

We turn T into T^* by:

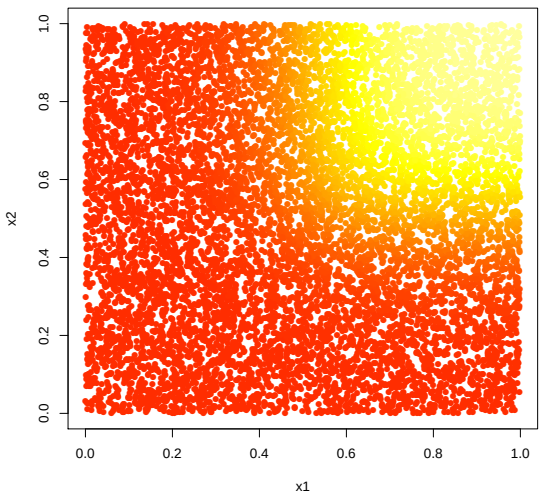
- Bagging / subsampling the training set (Breiman, 1996).
- Selecting the splitting variable at each step from m out of p randomly drawn features (Amit and Geman, 1997).
- **NB:** `grf` does something slightly more complicated; however, this is a **useful mental model** for causal forests in RCTs.

A first example

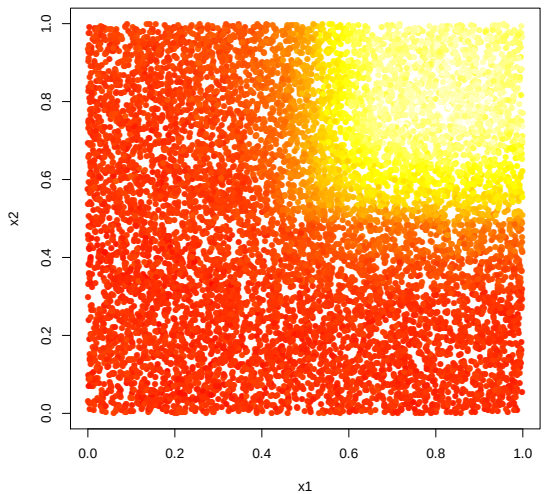
We have $n = 20$k observations whose features are distributed as $X \sim U([-1, 1]^p)$ with $p = 6$; treatment assignment is random. All **the signal is concentrated along two features**.

The plots below depict $\hat{\tau}(x)$ for 10k random test examples, projected into the 2 signal dimensions.

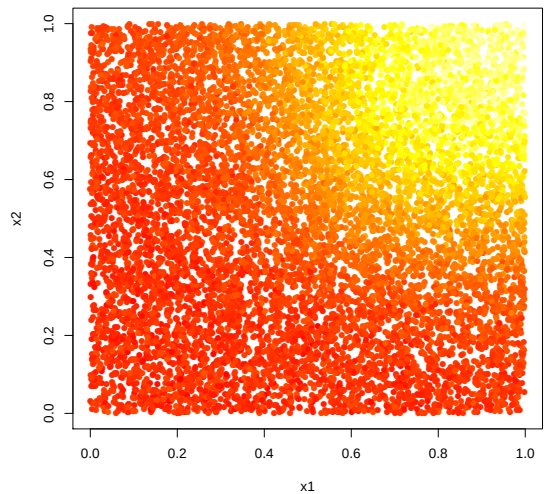
true effect $\tau(x)$



causal forest



k-NN estimate



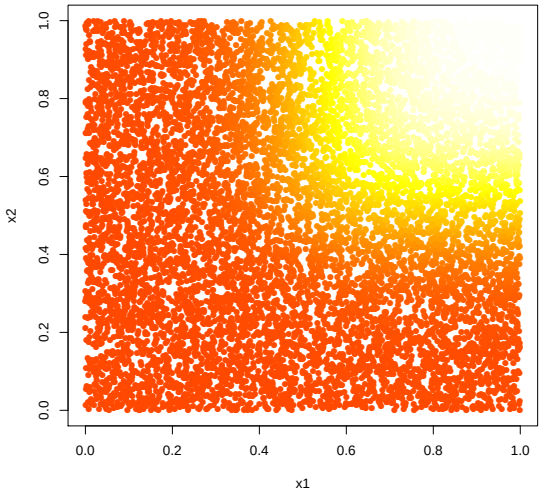
Software: `grf` for `R`.

A first example

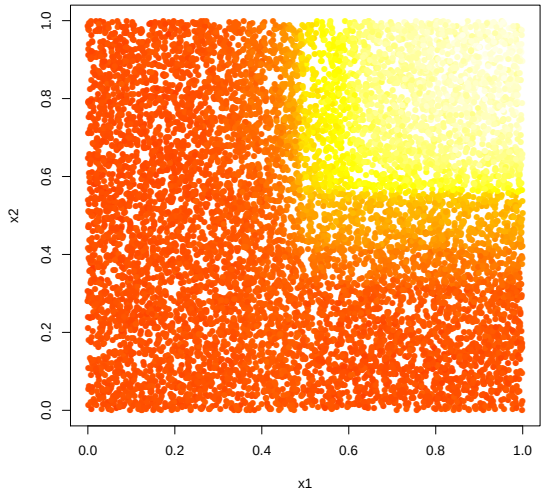
We have $n = 20$k observations whose features are distributed as $X \sim U([-1, 1]^p)$ with $p = 20$; treatment assignment is random. All **the signal is concentrated along two features**.

The plots below depict $\hat{\tau}(x)$ for 10k random test examples, projected into the 2 signal dimensions.

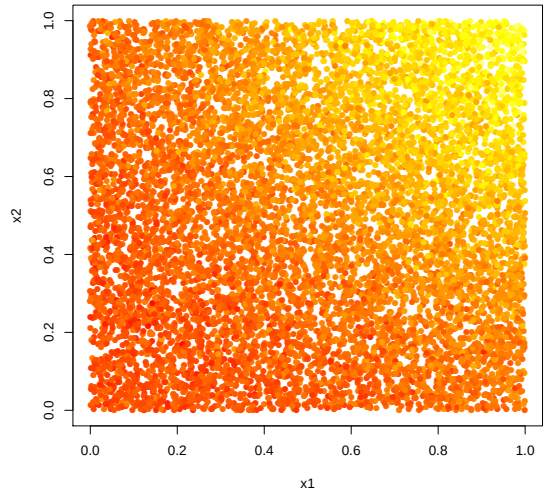
true effect $\tau(x)$



causal forest



k-NN estimate



Software: `grf` for `R`.

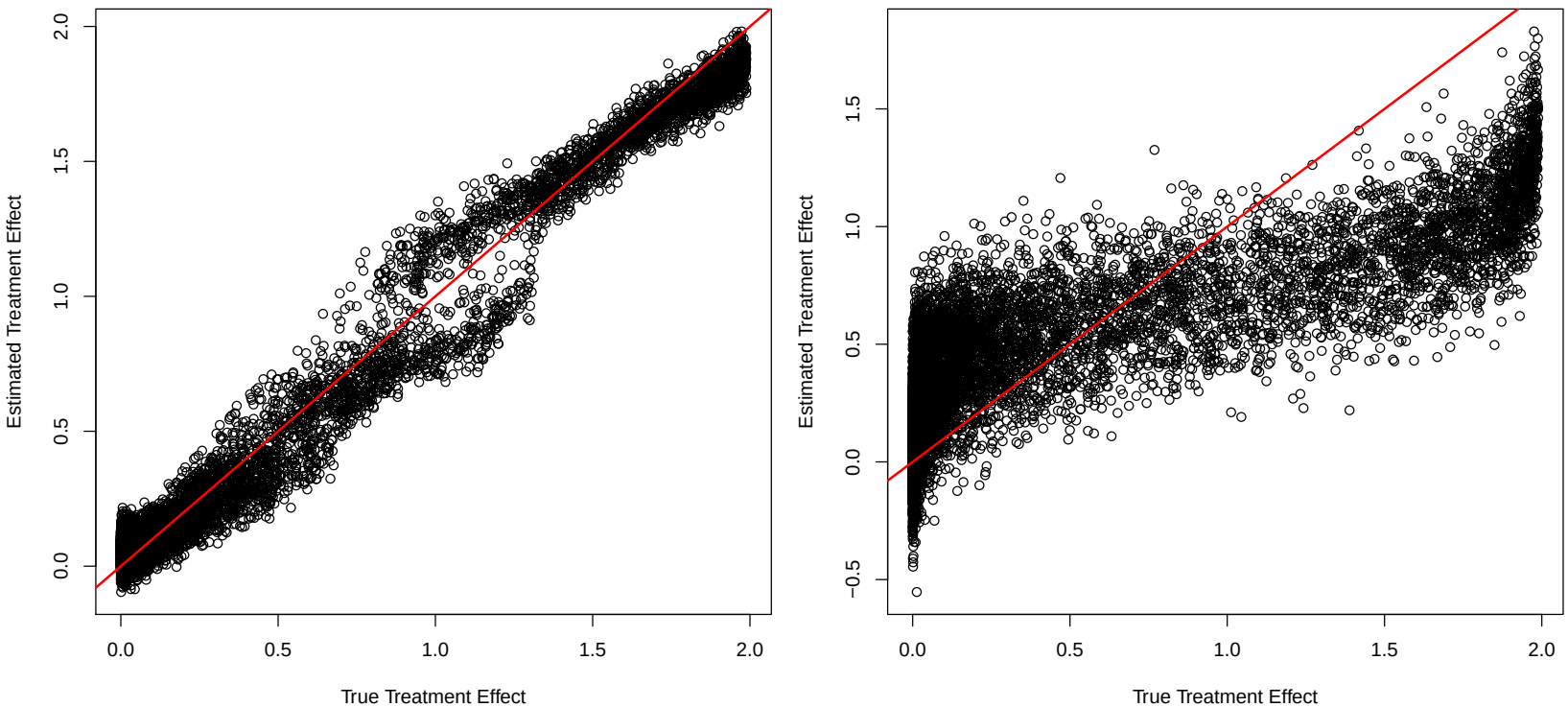
A first example

The causal forest dominates k-NN for both bias and variance.

With $p = 20$, the relative mean-squared error (MSE) for τ is

$$\frac{\text{MSE for } k\text{-NN (tuned on test set)}}{\text{MSE for forest (heuristically tuned)}} = 19.2.$$

causal forest **k-NN estimate**



For $p = 6$, the corresponding MSE ratio for τ is 2.2.

Avoiding overfitting via honest trees

Input: n training examples of the form $(X_i,\ Y_i,\ W_i)$ for causal trees, where X_i are features, Y_i is the response, and W_i is the treatment assignment.

1. Draw a random subsample of size $s \ll n$ from $\{1,\ ...,\ n\}$ without replacement, and then **divide it into two disjoint sets** of size $|\mathcal{I}| = \lfloor s/2 \rfloor$ and $|\mathcal{J}| = \lceil s/2 \rceil$.
2. Grow a tree via recursive partitioning, only looking at the $\mathcal{J}$-sample.
3. Estimate leaf-wise responses using only the $\mathcal{I}$-sample observations. Merge leaves if necessary.

In step 2, the splits are chosen by maximizing estimated treatment effect heterogeneity, as discussed earlier.

Application: General Social Survey

The General Social Survey is an extensive survey, collected since 1972, that seeks to measure demographics, political views, social attitudes, etc. of the U.S. population.

Of particular interest to us is a **randomized experiment**, for which we have data between 1986 and 2010.

- **Question A:** Are we spending too much, too little, or about the right amount on **welfare**?
- **Question B:** Are we spending too much, too little, or about the right amount on **assistance to the poor**?

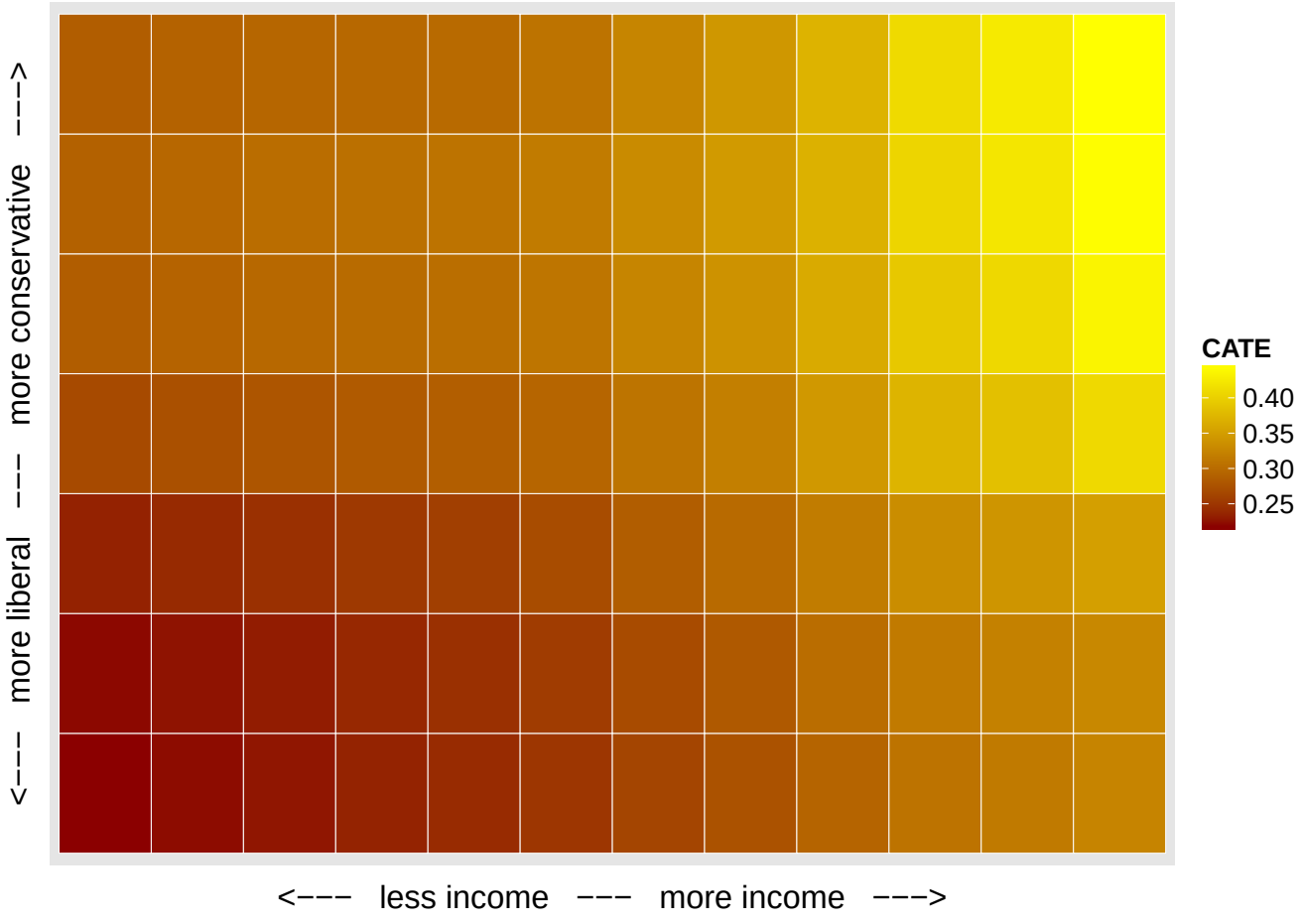
Treatment effect: how much less likely are people to answer **too much** to question B than to question A.

- We want to understand how the treatment effect depends on **covariates**: political views, income, age, hours worked, ...

NB: This dataset has also been analyzed by Green and Kern (2012) using Bayesian additive regression trees (Chipman, George, and McCulloch, 2010).

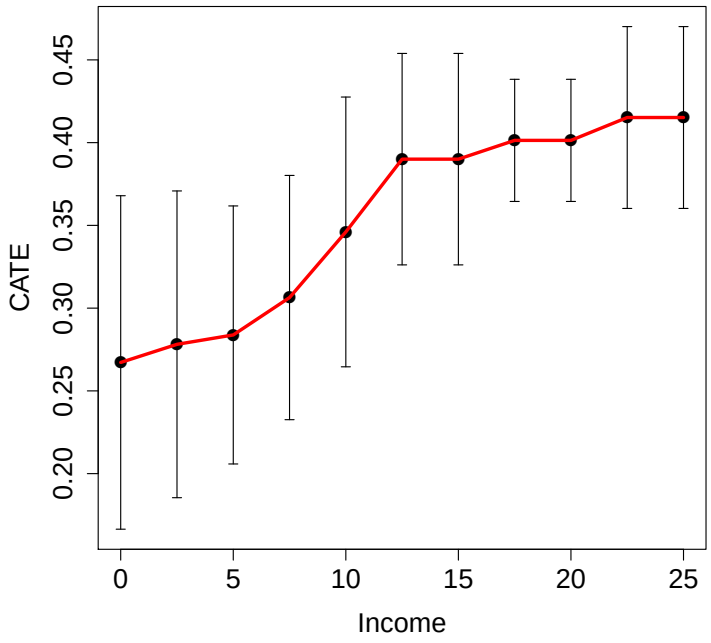
Application: General Social Survey

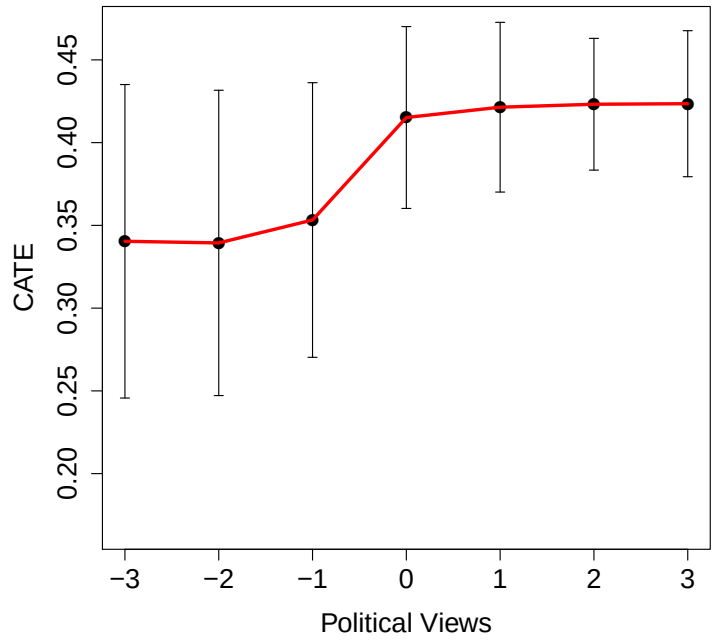
A causal forest analysis uncovers **strong treatment heterogeneity** ($n = 28,686$, $p = 12$).



Visualizing Forest Predictions

We can visualize the output of a forest using a **partial dependence plot**. Here, we vary one feature, and set others to their median. ($n = 28,686$, $p = 12$)

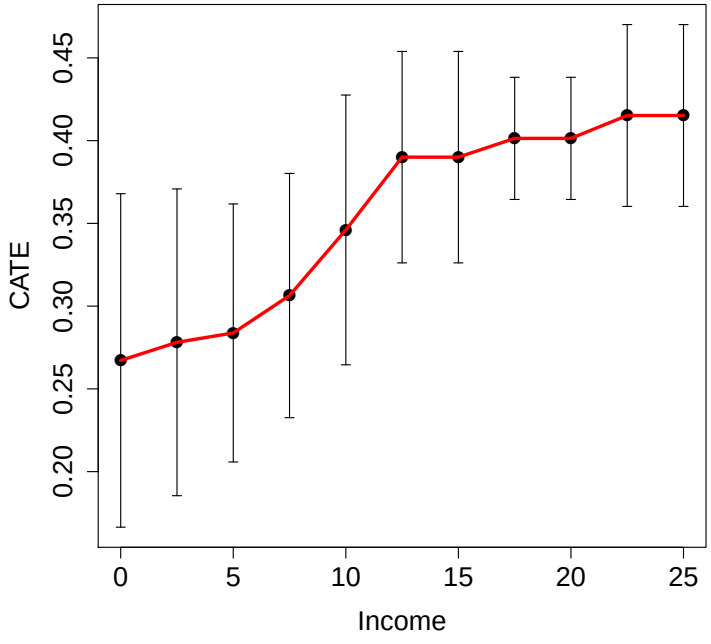


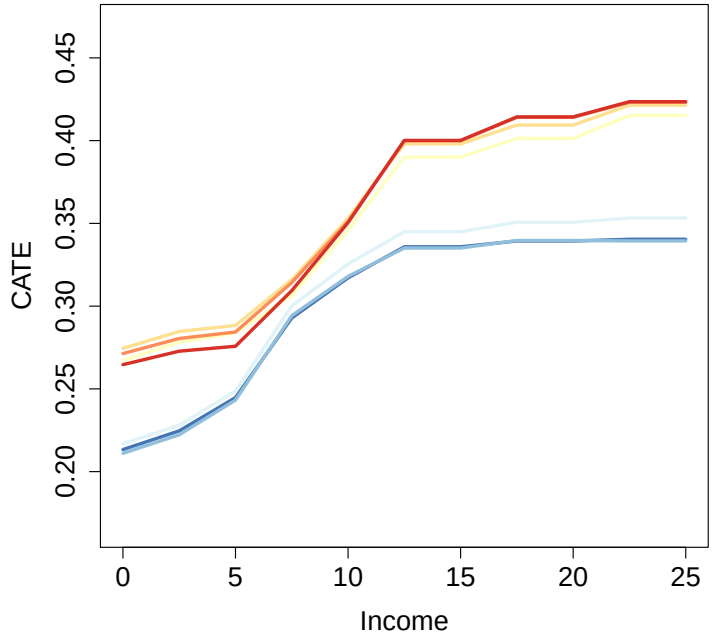


For more: Goldstein et al. "Peeking inside the black box." JCGS 24(1), 2015.

Visualizing Forest Predictions

We can visualize the output of a forest using a **partial dependence plot**. Here, we vary income, with political views set to 7 different levels. ($n = 28,686$, $p = 12$)



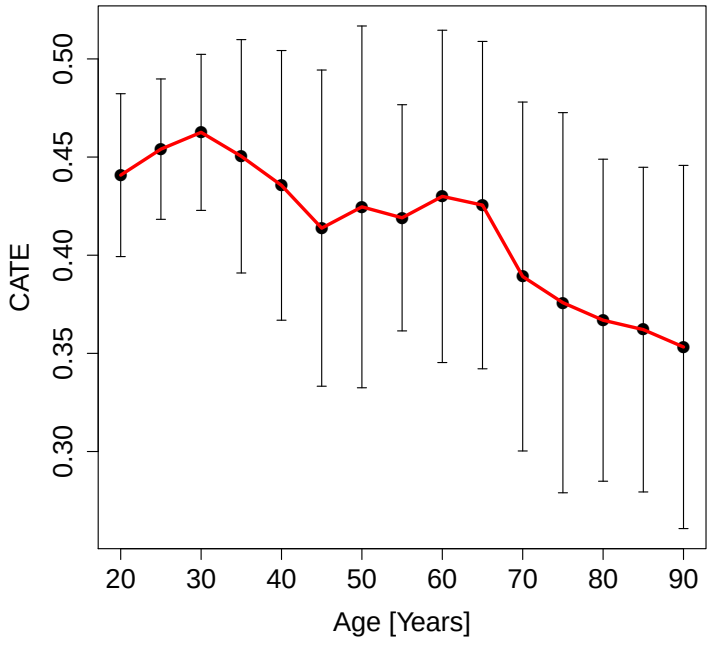


For more: Goldstein et al. "Peeking inside the black box." JCGS 24(1), 2015.

Visualizing Forest Predictions

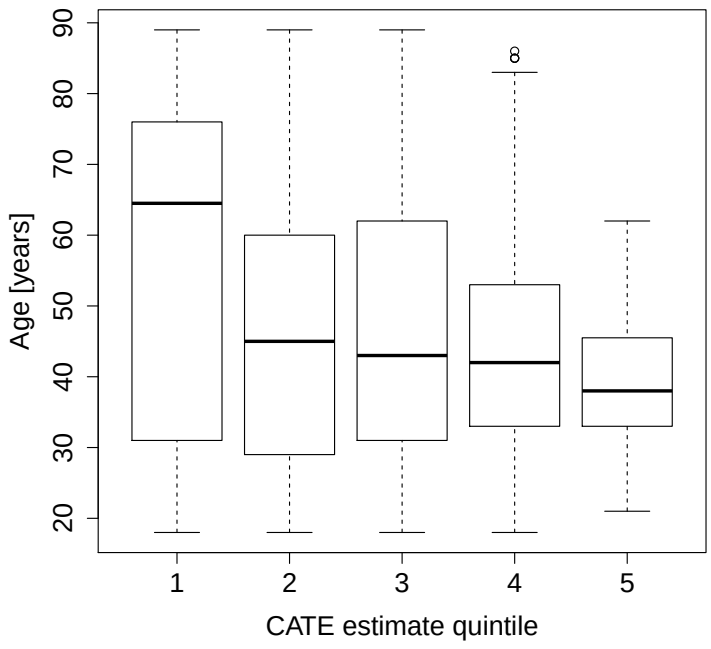
On the left, we show a a **partial dependence plot** with non-plotted covariates set to their **median**. We do a non-parametric regression of $\hat{\tau}(X)$ against age in various subgroups, **without fixing other covariates**.

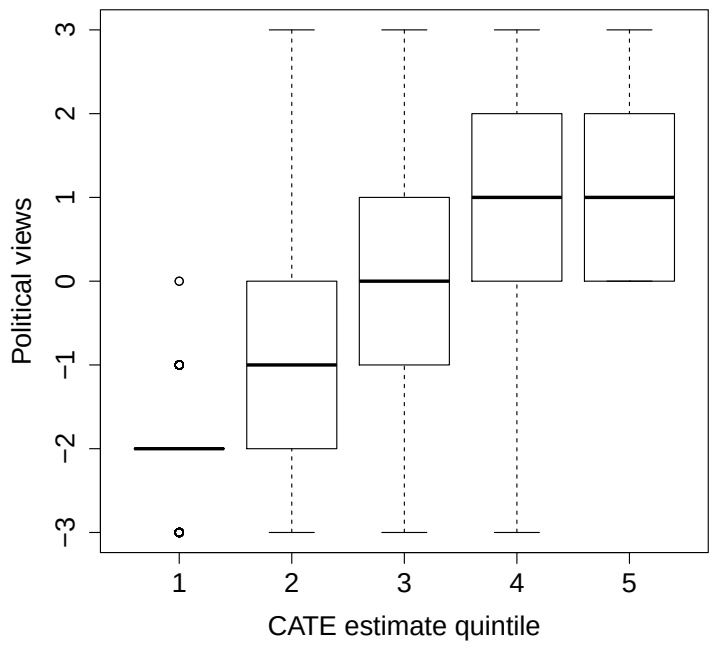
NB: In this dataset, the median person is rich (income in 25k+ bucket), moderate (political views = 0), and employed.



Visualizing Forest Predictions

Another useful **descriptive statistic** asks what typical samples with large/small predictions look like. This type of visualization can be helpful in guiding new questions.



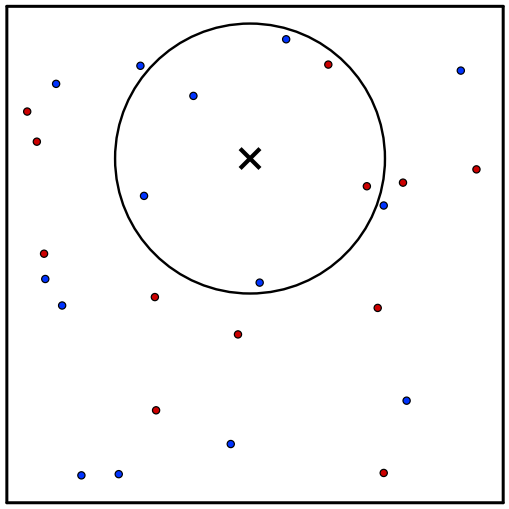


Nearest neighbor CATE estimation

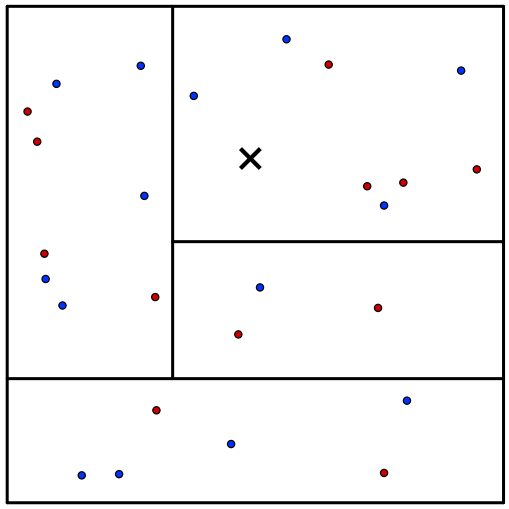
We have discussed k-**NN** as a paradigm for CATE estimation, and **causal forests** as an adaptive generalization of it. At a high level, our motivation is: If we assume **unconfoundedness**

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

along with **continuity** of the propensity score, then data in a neighborhood of x behaves as though it were from an RCT.



Euclidean neighborhood



Tree-based neighborhood

Regression-based CATE estimation

Again assuming **unconfoundedness**,

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

we can also write the CATE function as

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

Idea: Fit $\hat{\mu}_{(w)}(x)$ on observations with $W_i = w$, and then use the **difference in regression surfaces** to get $\hat{\tau}(x) = \hat{\mu}_{(1)}(x) - \hat{\mu}_{(0)}(x)$.

Penalized regression

For the rest of this lecture, we'll use models of the form

$$\mu_{(w)}(x) = \psi(x) \cdot \beta_{(0)},$$

where $\psi(\cdot)$ is a **basis expansion** of the original data. For example, suppose that x measures

$$(\text{income, age, gender}) \in \mathbb{R}_+ \times \mathbb{R}_+ \times \{\text{male, female, other}\}\,.$$

Then construct $\psi(\cdot)$ by making **all interactions** of gender with income, $(\text{income} - \overline{\text{income}})^2$, age, $(\text{age} - \overline{\text{age}})^2$, and $(\text{income} - \overline{\text{income}})(\text{age} - \overline{\text{age}})$, such that $\psi(x) \in \mathbb{R}^{18}$.

In other words, we represent x using a dictionary of **expressive** but not necessarily **interpretable** basis functions. You can keep adding basis functions until the dimension of $\psi(x)$ is much larger than n.

Penalized regression

For the rest of this lecture, we'll use models of the form

$$\mu_{(w)}(x) = \psi(x) \cdot \beta_{(0)},$$

where $\psi(\cdot)$ is a **basis expansion** of the original data. We then fit the regression using variants of the **lasso**,

$$\hat{\beta}_{(w)} = \text{argmin} \left\{ \sum_{\{i:W_i=w\}} \left(Y_i - \psi(X_i) \cdot \beta_{(w)}\right)^2 + \lambda \left\|\beta_{(w)}\right\|_1 \right\}.$$

NB: These parameter estimates $\hat{\beta}_{(w)}$ should not be directly interpreted; rather, they should be used to make predictions.

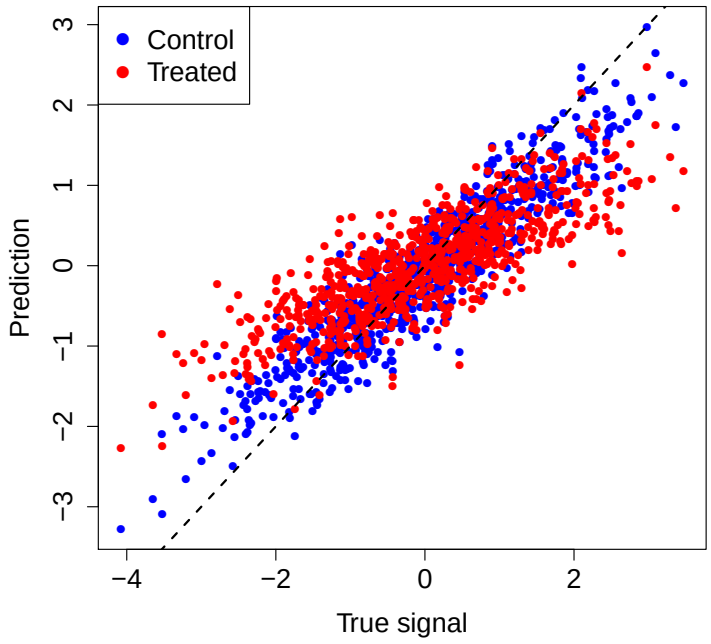
The **lasso** attempts to fit $\mu_{(w)}(x)$ using a large collection of **pre-specified** basis functions. **Boosting** and **neural networks** extend this idea, and also attempt to **learn the dictionary** $\psi(x)$.

Penalized regression

Idea #1: Just use the lasso out of the box, and set $\hat{\tau}(x) = \psi(x) \cdot (\hat{\beta}_{(1)} - \hat{\beta}_{(0)})$, with

$$\hat{\beta}_{(w)} = \text{argmin} \left\{ \sum_{\{i:W_i=w\}} \left(Y_i - \psi(X_i) \cdot \beta_{(w)}\right)^2 + \lambda \left\|\beta_{(w)}\right\|_1 \right\}.$$

This is a **bad idea**, because of **regularization bias**. Below, run separate lassos, with 10% of samples treated.



Penalized regression

Idea #1: Just use the lasso out of the box, and set $\hat{\tau}(x) = \psi(x) \cdot (\hat{\beta}_{(1)} - \hat{\beta}_{(0)})$, with

$$\hat{\beta}_{(w)} = \text{argmin} \left\{ \sum_{\{i:W_i=w\}} \left(Y_i - \psi(X_i) \cdot \beta_{(w)}\right)^2 + \lambda \left\|\beta_{(w)}\right\|_1 \right\}.$$

This is a **bad idea**, because of **regularization bias**. Below, run separate lassos, with 10% of samples treated.

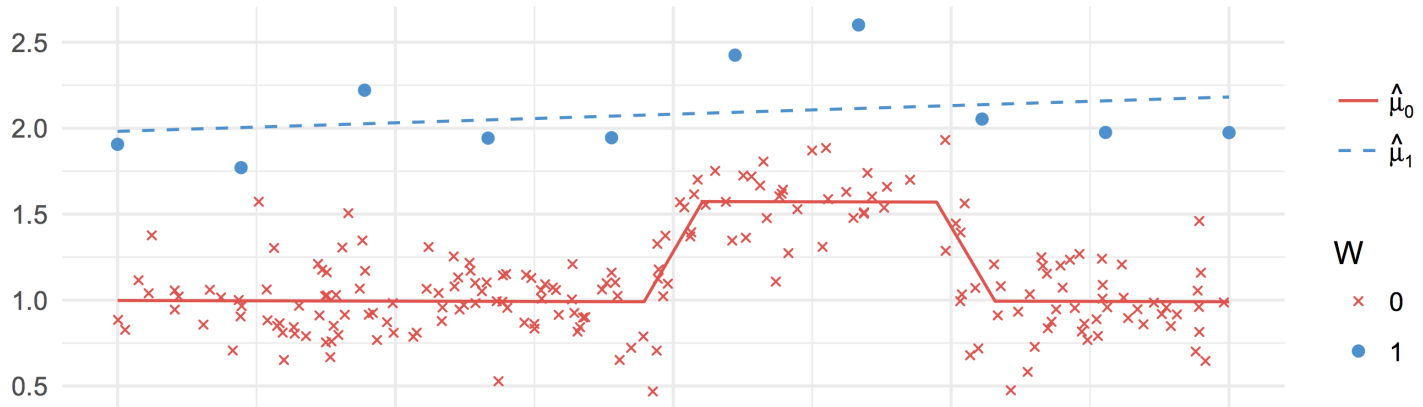


Image source: Künzel et al. "Meta-learners for HTE." arXiv, 2017.

Penalized regression

Idea #2: Use **interactions** to explicitly penalize treatment effects,

$$\left\{\hat{b}, \hat{\zeta}\right\} = \text{argmin} \left\{ \sum_{i=1}^{n} \left(Y_i - \psi(X_i) \cdot b - (W_i - 0.5)\, \psi(X_i) \cdot \zeta\right)^2 + \lambda_b \|b\|_1 \ + \lambda \|\zeta\|_1 \right\},$$

and then set $\hat{\tau}(x) = \hat{\zeta} \cdot \psi(x)$. Implicitly, this approach uses $\hat{\beta}_{(1)} = \hat{b} + \hat{\zeta}/2$ and $\hat{\beta}_{(0)} = \hat{b} - \hat{\zeta}/2$.

This transformation helps solve **regularization bias**, because $\hat{\beta}_{(0)}$ and $\hat{\beta}_{(1)}$ are now regularized towards each other in case of imbalanced sample sizes.

Penalized regression

Idea #3: If there are many more **controls** than **treated units**, do the following:

1. Fit $\hat{\mu}_{(0)}(x)$ via the lasso on the **controls**.
2. On the **treated** units, compute $\Delta_i = Y_i - \hat{\mu}_{(0)}(X_i)$.
3. Fit a lasso with **response** Δ_i on the **treated**.

Because we have so many control units, we attempt to predict the expected control counterfactual outcome for the treated units.

See Künzel et al. (2017) for more.

Lasso vs Random Forests

Both the lasso and random forests make predictions of the form:

$$\hat{\tau}(x) = \sum_j \beta_j T_j(x).$$

The **lasso** uses pre-determined basis functions $T_j = \psi_j$, but then learns weights β_j. **Random forests** use pre-determined weights $\beta_j = 1/B$, but learns the basis functions (via fitted trees).

- Main pro of lasso: You **can specify** the basis functions, which can increase power in high dimensions.
- Main pro of random forests: You **don't need to specify** the basis functions, which can increase robustness to unexpected signals.

More sophisticated methods, like **boosting** or **neural networks**, attempt to simultaneously learn both weights an basis functions.

Lasso vs Random Forests

Method 1: Random forest.
Method 2: Lasso on $\psi(X) = X$.
Method 3: Lasso on ψ via 3rd-order polynomials + interactions.

RCT with $n = 2,000$, $p = 10$, $X \sim U([-1, 1]^p)$, $\mathbb{P}[W_i = 1] = 0.5$,

$$\mathbb{E}\left[Y(w) \,\middle|\, X = x\right] = X_3 + X_4^3 + w * \tau(X), \quad \tau(X) = X_1 + X_2^2/2.$$

Note that the lasso used in method 3 can fit the signal, but method 2 is **misspecified**.

Results in terms of RMSE, i.e., $\sqrt{\mathbb{E}\left[(\hat{\tau}(X_i) - \tau(X_i))^2\right]}$.

Method 1: 0.063; **Method 2:** 0.152; **Method 3:** 0.030.

Lasso vs Random Forests

Method 1: Random forest.
Method 2: Lasso on $\psi(X) = X$.
Method 3: Lasso on ψ via 3rd-order polynomials + interactions.

RCT with $n = 2,000$, $p = 10$, $X \sim U([-1, 1]^p)$, $\mathbb{P}[W_i = 1] = 0.5$,

$$\mathbb{E}\left[Y(w) \,\middle|\, X = x\right] = X_3 + X_4^3 + w * \tau(X), \quad \tau(X) = 1/\left(1 + e^{-X_1}\right).$$

Both lassos, methods 2 and 3, are **misspecified**, although method #3 comes closer to being able to describe the signal.

Results in terms of RMSE, i.e., $\sqrt{\mathbb{E}\left[(\hat{\tau}(X_i) - \tau(X_i))^2\right]}$.

Method 1: 0.014; **Method 2:** 0.023; **Method 3:** 0.021.

Wrap up

We discussed several methods for estimating **CATE functions**, including

- k-NN / trees / random forests,
- difference in regression functions,
- penalized interactions, and
- modified response regression.

These methods will re-appear in subsequent lectures.

Question: What happened to the propensity score?