PLSC 40601

Week 3: Trees and forests.

Molly Offer-Westort

Department of Political Science,
University of Chicago

Spring 2023

Housekeeping

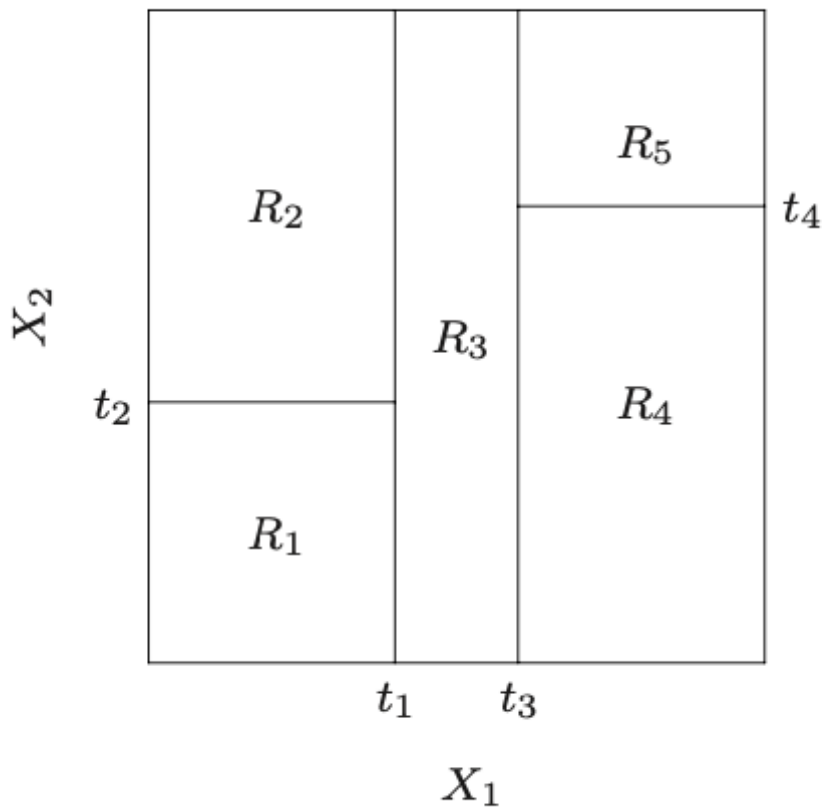
- ?

Trees

Regression Trees

- Suppose we have joint data, (Y, X_1, X_2).

- Our goal is to partition the data with the objective of prediction.



Hastie et al. (2009)

Regression Trees

- Our model is

$$f(\mathrm{X}) = \sum_{m=1}^{M} c_m \mathbb{1}\{\mathrm{X} \in R_m\}$$

- Our objective is

$$\sum_{i=1}^{N} \left(y_i - \hat{f}(\mathrm{x}_i)\right)^2$$

- With fixed regions R_m, how should we pick $\hat{c}_m$?

$$\hat{c}_m = \bar{y}_{\mathrm{x}_i \in R_m}$$

Regression Trees

- How do we pick partitions?
- A greedy approach:
  - splitting var j, split point s,

$$R_1(j,s) = \{X|X_j \leq s\} \text{ and } R_1(j,s) = \{X|X_j > s\}$$

  - Solve

$$\min_{j,s}\left[\min_{c_1}\sum_{i:x_{j[i]}\in R_1(j,s)}(y_i - c_1)^2 + \min_{c_2}\sum_{i:x_{j[i]}\in R_2(j,s)}(y_i - c_2)^2\right]$$

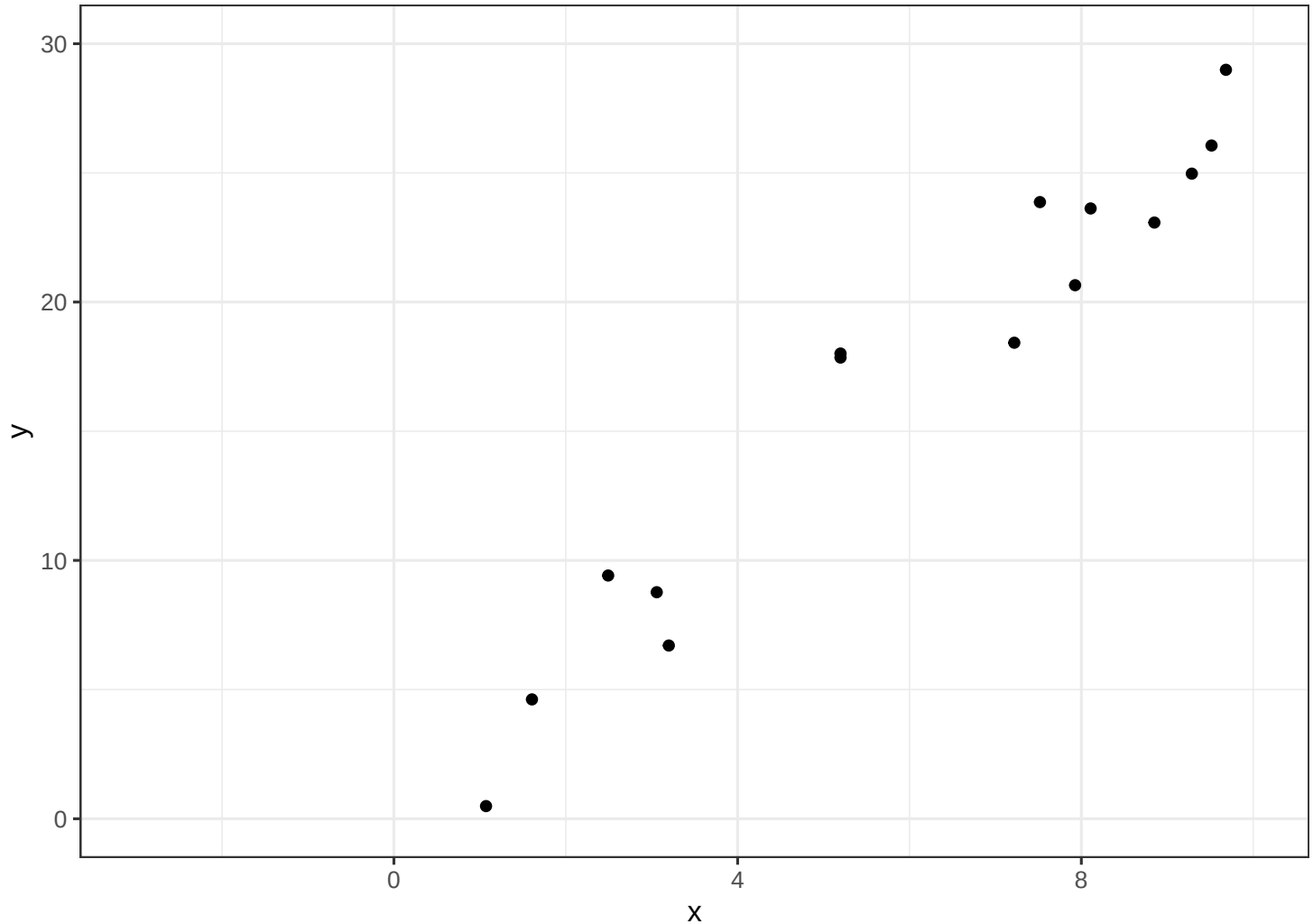
  - The inner minimization problem is again solved by averages.

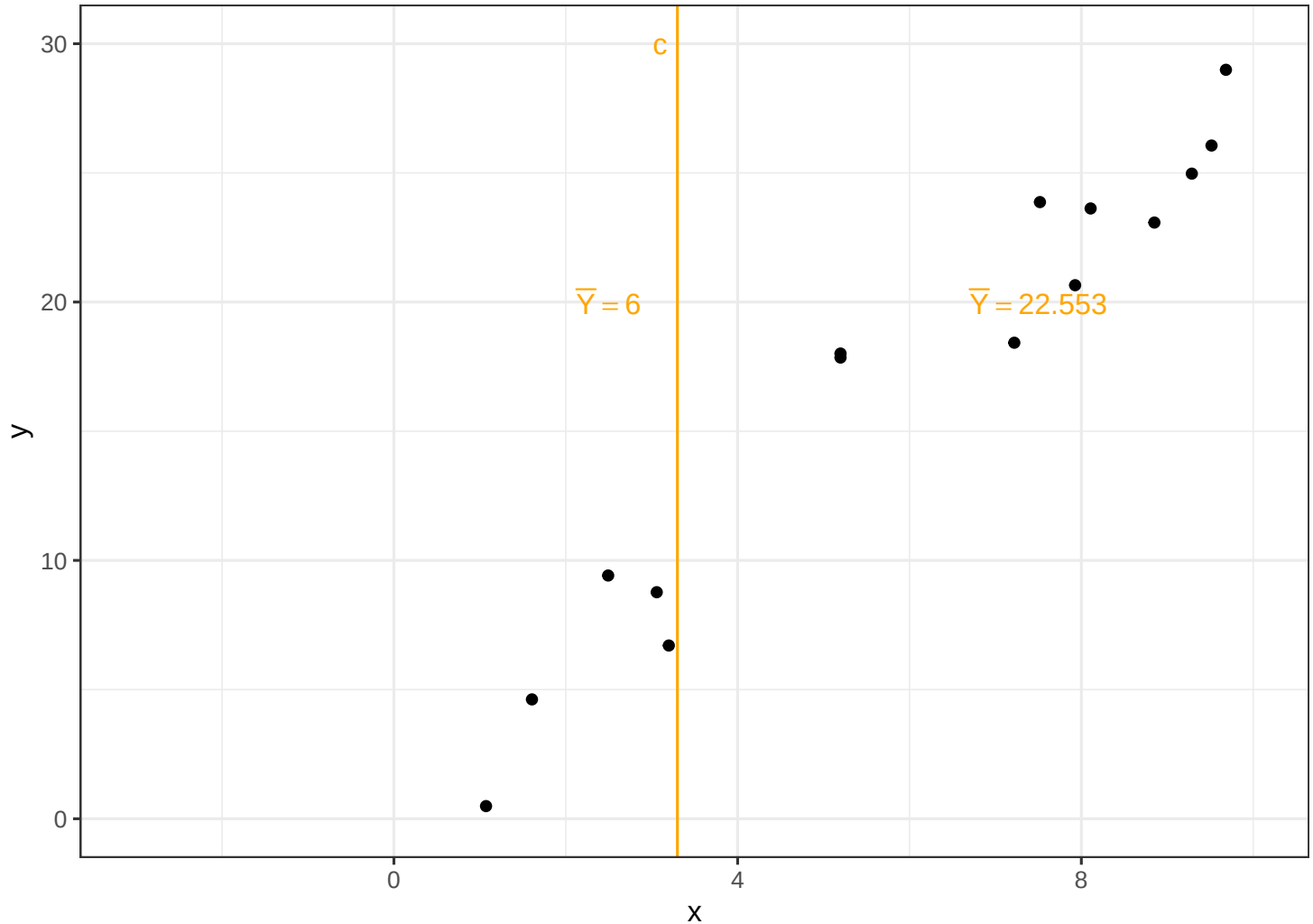
$$\hat{c}_1 = \bar{y}_{x_{j[i]}\in R_1(j,x)} \text{ and } \hat{c}_2 = \bar{y}_{x_{j[i]}\in R_2(j,x)}$$

  - Then pick s to solve the outer minimization problem for a given variable j.

Regression Trees

- For just one variable:


30
20
10
0
y
0
4
8
x


30
20
10
0
y
0
4
8
x
c
$\overline{Y} = 6$
$\overline{Y} = 22.553$

Elements of trees

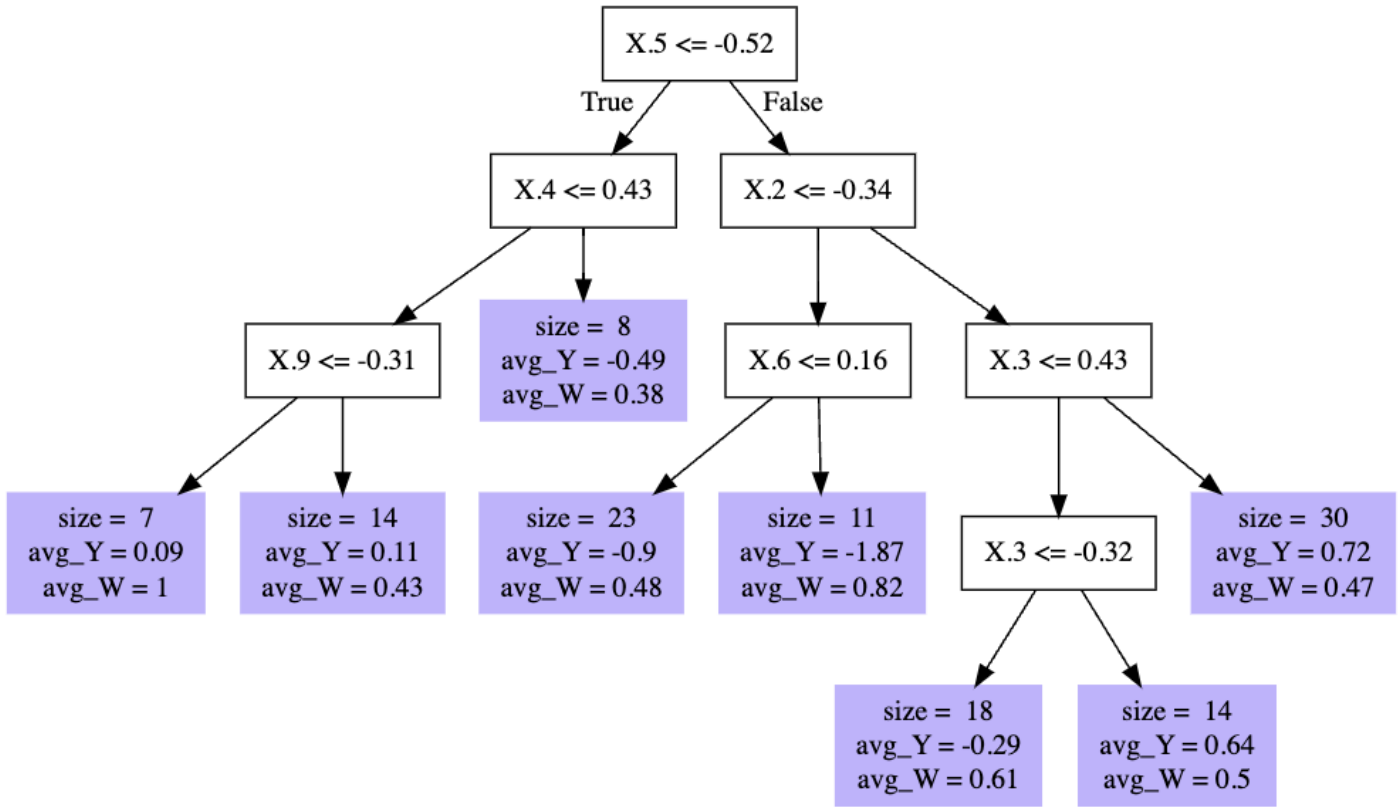
```
> n <- 500
> p <- 10
> X <- matrix(rnorm(n * p), n, p)
> W <- rbinom(n, 1, 0.5)
> Y <- pmax(X[, 1], 0) * W + X[, 2] +
+   pmin(X[, 3], 0) + rnorm(n)
> c.forest <- causal_forest(X, Y, W)
> tree <- get_tree(c.forest, 1)
> leaf.nodes <- get_leaf_node(tree, X[1:5, ])
```

Elements of trees

```
> tree

GRF tree object
Number of training samples: 250
Variable splits:
(1) split_variable: X.5  split_value: -0.520294
  (2) split_variable: X.4  split_value: 0.426613
    (4) split_variable: X.9  split_value: -0.314623
      (8) * num_samples: 7  avg_Y: 0.09 avg_W: 1
      (9) * num_samples: 14  avg_Y: 0.11 avg_W: 0.43
    (5) * num_samples: 8  avg_Y: -0.49 avg_W: 0.38
  (3) split_variable: X.2  split_value: -0.344787
    (6) split_variable: X.6  split_value: 0.163222
      (10) * num_samples: 23  avg_Y: -0.9 avg_W: 0.48
      (11) * num_samples: 11  avg_Y: -1.87 avg_W: 0.82
    (7) split_variable: X.3  split_value: 0.428785
      (12) split_variable: X.3  split_value: -0.322743
        (14) * num_samples: 18  avg_Y: -0.29 avg_W: 0.61
        (15) * num_samples: 14  avg_Y: 0.64 avg_W: 0.5
      (13) * num_samples: 30  avg_Y: 0.72 avg_W: 0.47
```

A tree


X.5 <= -0.52
True
False
X.4 <= 0.43
X.2 <= -0.34
X.9 <= -0.31
size = 8
avg_Y = -0.49
avg_W = 0.38
X.6 <= 0.16
X.3 <= 0.43
size = 7
avg_Y = 0.09
avg_W = 1
size = 14
avg_Y = 0.11
avg_W = 0.43
size = 23
avg_Y = -0.9
avg_W = 0.48
size = 11
avg_Y = -1.87
avg_W = 0.82
X.3 <= -0.32
size = 30
avg_Y = 0.72
avg_W = 0.47
size = 18
avg_Y = -0.29
avg_W = 0.61
size = 14
avg_Y = 0.64
avg_W = 0.5

Elements of trees

- Node
- Split/branches
- Leaves

Trees for classification

- What about when Y is a category?
- For binary classification, minimize surrogate for classification error with split point s, $I(s) = \sum_{t=1}^{2} \gamma_t$

$$\gamma_t = 1 - \left[\bar{Y}_t^2(1 - \bar{Y}_t^2)^2\right]$$

- $I(s)$ measures the impurity of a partition. What happens if R_m has only 1's, or only 0s?
- Why impurity instead of classification error? Smooth function, easier to minimize. But there are other metrics we could use.
- We can extend these methods to more complex classification tasks.

Policy trees, or decision trees

- Policy trees solve:

$$\pi^* = \underset{\pi \in \Pi}{\operatorname{argmax}} \left[\frac{1}{N} \sum_{i=1}^{N} \Gamma_i \left(\pi(X_i) \right) \right]$$

$$\Gamma_i = \frac{1}{N} \sum_{i=1}^{N} \underbrace{\hat{\mathbb{E}}\left[Y_i | W_i = w, X_i \right]}_{\text{estimated outcome model}} +$$

$$\frac{1}{N} \sum_{i=1}^{N} \underbrace{\frac{\mathbb{1}\{W_i = w\} \left(Y_i - \hat{\mathbb{E}}\left[Y_i | W_i = w, X_i \right] \right)}{e_i(w; s)}}_{\text{residual bias correction}}$$

Tuning parameters

- How do we pick which splitting variables to use?
- How many splits to complete?
- Pruning?

Forests

Random forests...

- Define a tree predictor or classifier as $h(\mathrm{x}, \Theta_k)$
- Θ_k are i.i.d. random vectors (putting the *random* in random forests)
- With predictors or classifiers $\{h(\mathrm{x}, \Theta_k),\ k = 1, \ldots, K\}$, combine across trees, where estimates produced from each tree are averaged for prediction problems, and treated as votes for classification problems.

Random forests…

- Different types of forests use different approaches to random vectors Θ_k.

- How to compare them?

Generalization error for forests

For numerical predictors:

$$PE^*_{\text{forest}} = \mathrm{E}_{\mathsf{X},Y}\left[(Y - \mathrm{E}_{\Theta}\left[h(\mathsf{X},\Theta)\right])^2\right]$$

$$PE^*_{\text{tree}} = \mathrm{E}_{\Theta}\left[\mathrm{E}_{\mathsf{X},Y}\left[(Y - h(\mathsf{X},\Theta))^2\right]\right]$$

$$PE^*_{\text{forest}} \leq \bar{\rho} PE^*_{\text{tree}}$$

where $\bar{\rho}$ is a weighted correlation between $Y - h(\mathsf{X},\Theta)$, $Y - h(\mathsf{X},\Theta')$ where Θ, Θ' are independent.
Implication: for good generalization error of a forest, we need low correlation error across Θ, and low error trees.

Some approaches to forests

- Adaptive reweighting of the training set (arcing), see Adaboost (**ada**ptive + **boost**ing) (Freund et al., 1996) (*not* random)

- Bagging

- Forest RI: Random input selection at each node. Don't prune. Fix number of features used.
  - Tuning parameter: number of features used.
  - Performs pretty well, even when number of features is small (1!)
  - If total number of features is small, can result in high correlation.

- Forest RC: Random combination of inputs selected at each node. Fix number of features used, combine them with random coefficients. Create a fixed number of combinations, and search over for the best.
  - Tuning parameters: number of features used, number of combinations of features.

Some approaches to forests

- Which approach works best depends on number of covariates, how correlated covariates are, how predictive covariates are.

- More (combinations of) features not always better.

Other qualities

In addition to low correlation across prediction error, and strength of trees, we might want methods that are

- Robust to outliers
- Computational speed
- Internal metrics of error, strength, correlation, variable importance
- Possibility to parallelize?

Internal metrics

"Out-of-bag" estimation (we've seen this before)

$$\hat{f}_i^{oob} = \frac{1}{|C^{-i}|} \sum_{b \in C^{-i}} \hat{f}^b(x_i)$$

C^{-i} is the set of bootstrap samples that do not contain i, $|C^{-i}|$ is the size of this set

Honesty.

Honesty

- Returning to (causal) inference…we might like to use these methods to get valid inference, potentially on causal targets.

- As we think about causal quantities, we'll move the target.

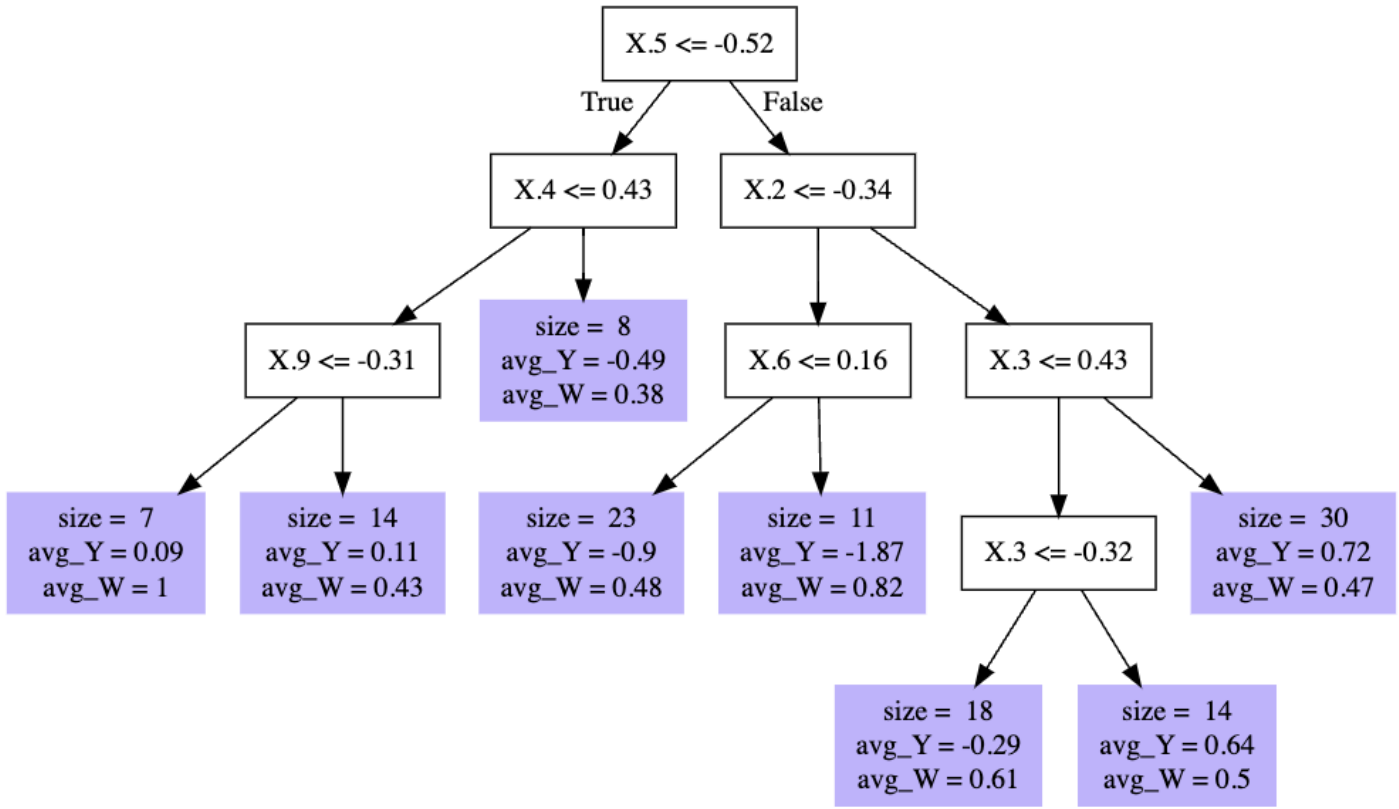
An honest tree algorithm

1. Split the sample into two folds.
2. Use the first fold to learn splits of the tree.
3. Estimate response within leaves using the second fold.

- This can result in some leaves being empty. Prune them?

- This procedure reduces bias relative to those proposed by Breiman (2001).

A tree


X.5 <= -0.52
True
False
X.4 <= 0.43
X.2 <= -0.34
X.9 <= -0.31
size = 8
avg_Y = -0.49
avg_W = 0.38
X.6 <= 0.16
X.3 <= 0.43
size = 7
avg_Y = 0.09
avg_W = 1
size = 14
avg_Y = 0.11
avg_W = 0.43
size = 23
avg_Y = -0.9
avg_W = 0.48
size = 11
avg_Y = -1.87
avg_W = 0.82
X.3 <= -0.32
size = 30
avg_Y = 0.72
avg_W = 0.47
size = 18
avg_Y = -0.29
avg_W = 0.61
size = 14
avg_Y = 0.64
avg_W = 0.5

References I

Breiman, L. (2001). Random forests. *Machine learning*, 45:5–32.

Freund, Y., Schapire, R. E., et al. (1996). Experiments with a new boosting algorithm. In *icml*, volume 96, pages 148–156. Citeseer.

Hastie, T., Tibshirani, R., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.