# Random Forest

## Basics and Its Application to Econometrics

Seulkichan Bang, Giryang Park

February 20, 2025

Applied Econometrics Reading Group

# Outline

# Decision Tree

- Observation(s): $\{(y_i, X_{1i}, ..., X_{pi})\}_{i=1}^{n}$
  - $y_i$ is response, continuous or categorical(qualitative)
  - $X_{1i}, ..., X_{pi}$ are predictors
- Our focus: to well predict the response, given the value of predictors
- Strategy: **Decision tree** divides the **predictor space**

$$\{(X_1, ..., X_p) \mid X_k \in [a_k, b_k], \forall k = 1, ..., p\} \subset \mathbb{R}^p$$

  and use the mean response for the prediction.
- Depending on the type of the response variable, we call it either **Regression Tree** or **Classification Tree**

# Decision Tree

- Example: We want to predict a baseball player's 'Salary', using his 'Years' and 'Hits'

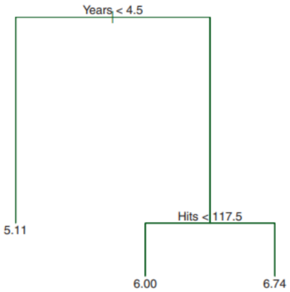



Figure 1: Regression Tree

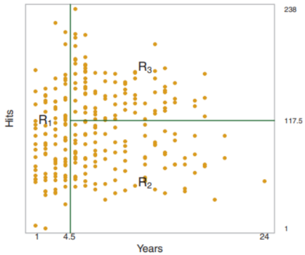



Figure 2: Covariate Partitioning

# Regression Tree

- Goal: Find $J$ disjoint regions(partitions) $R_1, ..., R_j$ such that $R_1 \cup ... \cup R_j =$**'predictor space'**, which minimize

$$\mathrm{RSS} = \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$$

  in training data, where $\hat{y}_{R_j}$ is mean response(salary) for $R_j$.
- $\hat{y}_{R_j}$ is also the prediction on a player's salary for test points that fall into the region $R_j$.

# Regression Tree

- Due to computational complexity, we adopt **recursive binary splitting**, which is a (i) top-down, (ii) greedy approach.
- At first (root node), we consider all the possible pairs of $(j, s)$ that yield:

$$R_1(j,s) = \{X | X_j < s\} \; and \; R_2(j,s) = \{X | X_j \geq s\}$$

- And we select a value of $(j, s)$ that minimize the RSS.
- Continue the process until a stopping criterion (e.g., maximum depth, minimum node size) is reached.

# Regression Tree

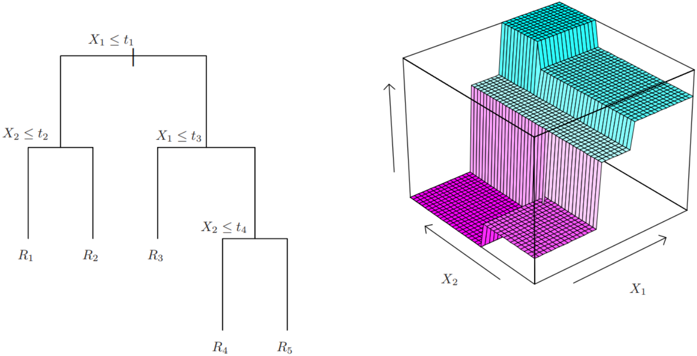



Figure: Example of Regression Tree

# Pruning

- Overfitting problem $\rightarrow$ RSS in test data?
- **Pruning**: grow a very large tree $T_0$ and then prune it to a subtree $T$ that leads to the lowest **test error rate**.
- **Cost complexity pruning**: For each value of a tuning parameter $\alpha(\geq 0)$, there exist a subtree $T \subset T_0$ that minimizes

$$\sum_{m=1}^{|T|} \sum_{i: x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T|.$$

- We can obtain a sequence of subtrees as a function of $\alpha$ and $\alpha$ will be selected using K-fold cross-validation.

# Pruning

**Building a Regression Tree**

1. Use recursive binary splitting to grow a large tree on the training data, stopping only when each terminal node has fewer than some minimum number of observations.
2. Apply cost complexity pruning to the large tree in order to obtain a sequence of best subtrees, as a function of $\alpha$.
3. Use K-fold cross-validation to choose $\alpha$. That is, divide the training observations into $K$ folds. For each $k = 1, \ldots, K$:
   a Repeat Steps 1 and 2 on all but $k$th fold of the training data.
   b Evaluate the mean squared prediction error on the data in the left-out $k$th fold, as a function of $\alpha$.

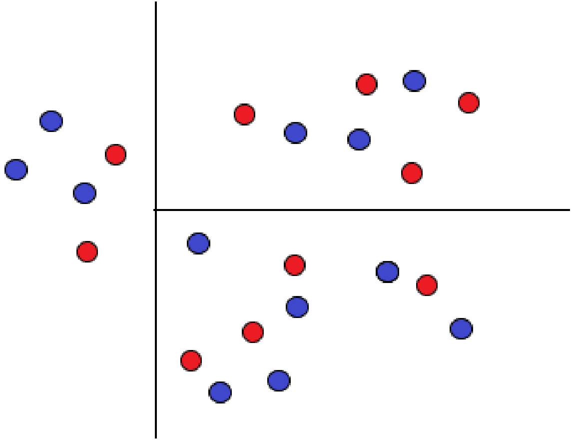
   Average the results for each value of $\alpha$, and pick $\alpha$ to minimize the average error.
4. Return the subtree from Step 2 that corresponds to the chosen value of $\alpha$.

# Classification Tree

- If the response is qualitative/categorical: error evaluation and prediction?
- For evaluation, instead of RSS, use one of the followings:
  1. (Classification error rate) $E = 1 - max_k(\hat{p}_{jk})$
  2. (Gini index) $G = \sum_{k=1}^{K} \hat{p}_{jk}(1 - \hat{p}_{jk})$
  3. (Entropy) $D = -\sum_{k=1}^{K} \hat{p}_{jk} \log \hat{p}_{jk})$

  where $\hat{p}_{jk}$ is the proportion of k-th class in the region $R_j$
- Prediction: Instead of mean response, predict with the most commonly occurring class in the region (majority voting)

# Classification Tree



Classification error rate of the right-bottom region is 1 - 0.6 = 0.4

# Why Decision Tree?

- Pros
  - Better performance for non-linear and complex relationship
  - Interpretability, Graphical representation
  - Can handle qualitative variables, interactions in the same way.
- Cons
  - Predictive accuracy not guaranteed always
  - Can be very non-robust

$\Rightarrow$ We want to improve predictive performance by **lowering the variance** of decision tree.

# Bagging

- **Bootstrap aggregation(bagging)** is a general-purpose procedure for reducing variance of statistical learning method.
- How to implement
  1. Generate $B$ different bootstrapped training data sets.
  2. Fit a model and compute $\hat{f}^{*b}(x)$ for $b = 1, ..., B$, then average them to obtain the predicted value.

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x)$$

  Here $\hat{f}^{*b}(x)$ is the prediction of $b$-th tree.

- Note: No pruning, $B$ is sufficiently large.

# Random Forest

- Motivation: Bagged trees can exhibit **high correlation**. Then averaging may not lead to large of a reduction in variance.
- **Random Forest** only considers randomly selected $m(< p)$ predictors from all the $p$ predictors for each split.

- If $m = p$, then it amounts to bagging.
- Convention is $m = \sqrt{p}$ for classification trees while $m = p/3$ for regression tree.
- Select small $m$ when there are a lot of correlated predictors.

# Out-Of-Bag Error

- A natural way to estimate the test error of a bagged model
- **Out-Of-Bag observation**: Observations not used to fit a given bagged tree. About $1/3$ of the entire data are OOB observations for each tree, on average.
- **Out-Of-Bag error**:
  1. For each (training) observations, make prediction by averaging the trees, for which it is OOB. About $B/3$ trees are used.
  2. Compute overall OOB MSE (for a regression problem).

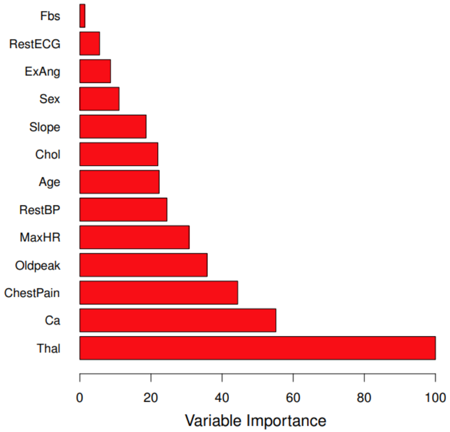
# Variable Importance

- Can we evaluate the importance of each predictor?
- Bagging improves prediction accuracy at the cost of interpretability.
- But still, one can obtain an overall summary of the importance of each predictor.

  $\Rightarrow$ Compute total amount of RSS/Gini index decrease due to splits over a given predictor, averaged over all $B$ trees.

# Variable Importance

Figure: Example of Variable Importance plot



# Outline

# Motivation

Integrating Machine Learning to Policy Evaluation

- ML methods typically excels at prediction, but they often lack well-established statistical properties
- Treatment effect estimation is much more challenging due to the lack of *ground truth* for causal parameters
- Moreover, introducing heterogeneity in treatment effect is critical agenda for causal inference (e.g., identifying sub-populations and suggesting optimal policies)

# Motivation

To do so, several studies shed light on Random Forest algorithms.

- Athey & Imbens (2016) suggests heterogeneity-capturing splitting scheme to build `causal trees`
- Wager & Athey (2018) ensembles trees to `causal forest`
- Athey et al. (2019) generalizes the method even more so as to handle general GMM case

# Motivation

To do so, several studies shed light on Random Forest algorithms.

- Athey & Imbens (2016) suggests heterogeneity-capturing splitting scheme to build `causal trees`
- Wager & Athey (2018) ensembles trees to `causal forest`
- Athey et al. (2019) generalizes the method even more so as to handle general GMM case

# Motivation

To do so, several studies shed light on Random Forest algorithms.

- Athey & Imbens (2016) suggests heterogeneity-capturing splitting scheme to build `causal trees`
- Wager & Athey (2018) ensembles trees to `causal forest`
- Athey et al. (2019) generalizes the method even more so as to handle general GMM case

# Wager & Athey (JASA, 2018)

- **Estimation and Inference of Heterogeneous Treatment Effects using Random Forests** (Journal of American Statistical Association, 2018)

---

Main Contributions

- Established desirable statistical properties for random forests, which yields tractable asymptotic theory and valid inference
- Suggested adaptive and nonparametric estimation procedure for heterogeneous treatment effect
- Instead of leaf-wise/subgroup-wise average treatment effect, addressed individual treatment effect $\tau(x)$

# Basic Setup

- Observed data: $(X_i, Y_i, W_i) \in [0,1]^d \times \mathbb{R} \times \{0,1\}$ where $X_i$ is a feature vector, $Y_i$ is a response and $W_i$ is a treatment indicator

- Following potential outcomes framework, our goal is to estimate the conditional treatment effect $\tau(x)$:

$$\tau(x) = \mathbb{E}[Y_i(1) - Y_i(0) \mid X_i = x]$$

- Without any further assumption, estimation of $\tau(x)$ is impossible
$\Rightarrow$ Assumes **unconfoundedness** as follows:

$$\{Y_i(0), Y_i(1)\} \perp W_i \mid X_i$$

i.e., nearby observations in covariate space are presumed to have come from a randomized experiment

# Regression Trees to Causal Trees

Standard regression tree predicts conditional outcome via

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

Analogously, a causal tree estimates treatment effect via[1]

$$\begin{aligned}\hat{\tau}(x) =& \frac{1}{|\{i : W_i = 1, X_i \in L(x)\}|} \sum Y_i \\ & - \frac{1}{|\{j : W_j = 0, X_j \in L(x)\}|} \sum Y_j\end{aligned}$$

[1] This strategy implicitly assumes the case where leaves are small enough so that $(Y_i, W_i)$ pairs corresponding to $i \in L(x)$ are regarded to have come from randomized experiment.

# Causal Trees to Causal Forest

**Causal forest** generates an ensemble of $B$ causal trees:

$$\hat{\tau}(x) = B^{-1} \sum_{b=1}^{B} \hat{\tau}_b(x)$$

where $\hat{\tau}_b(x)$ refers to an estimate from $b$-th tree

# Causal Trees to Causal Forest

The following results are established for causal forest:

1. Pointwise consistency of $\hat{\tau}(x)$ for $\tau(x)$
2. Asymptotic sampling dist'n (unbiased, asymptotically normal)
3. Consistency of infinitesimal jackknife estimator for AVAR

To do so, two distinct conditions are imposed.

- ✓ **Subsampling**: Each causal tree is built using random subsamples of size $s$, where $s/n \ll 1$. For theoretical results, it is assumed that $s$ scales as $s \asymp n^{\beta}$ for some $\beta_{\text{min}} < \beta < 1$.
- ✓ **Honesty**: Information for selecting model structure (i.e., partitioning covariate space) cannot be the same with information for estimation given a model structure.

# Algorithms

**Algorithm** Double-Sample Trees

**Require:** $(X_i, Y_i)$ or $(X_i, Y_i, W_i)$, minimum leaf size $k$

1: Draw a random subsample of size $s$ without replacement, and 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, making splits using entire $\mathcal{J}$ sample (and $X$ or $W$ data from $\mathcal{I}$ sample), but without using $Y$ from $\mathcal{I}$-sample.

3: Estimate leaf-wise responses using only $\mathcal{I}$ sample.

Splitting criteria:

- Regression tree: Minimize RSS
- Causal tree: Maximize variance of $\hat{\tau}(x)$ for $\mathcal{I}$ sample

# Algorithms

---

**Algorithm** Propensity Trees

---

**Require:** $(X_i, Y_i, W_i)$, minimum leaf size $k$

1: Draw a random subsample of size $s$ without replacement.

2: Grow a classification tree using $(X_i, W_i)$ pairs from $\mathcal{I}$ sample, without using $Y$.

3: Estimate leaf-wise treatment effect $\tau(x)$.

---

Splitting criteria:

- Minimize Gini index (standard CART measure)

# Algorithms

Compute tree estimate $\hat{\tau}_b(x)$ using one of these algorithms, aggregate them to obtain final estimate $\hat{\tau}(x)$. And now ...

# Asymptotic Theory for Random Forests

Start with regression forest; want to estimate $\mu(x) = \mathbb{E}[Y|X = x]$

<u>Preliminaries</u>

(1) **Random-split**: Probability of each feature being selected for a split is bounded below by $\pi/d$ for $\pi \in (0, 1]$

(2) $\boldsymbol{\alpha}$**-regular**: Minimum fraction/size condition

(3) **Symmetry**: Output does not depend on training index

(4) Infinitesimal jackknife estimator of asymptotic variance:

$$\hat{V}_{ij}(x) = \frac{n-1}{n}\left(\frac{n}{n-s}\right)^2 \sum_{i=1}^{n} \text{cov}_*[\hat{\mu}_b^*(x), N_{ib}^*]$$

# Asymptotic Theory for Random Forests

For $n$ iid training samples $(X_i, Y_i)$, if the following conditions hold,

1. Feature density is bounded away from 0 and $\infty$
2. $\mu(x)$ and $\mathbb{E}[Y^2|X = x]$ are Lipschitz-continuous
3. $\mathrm{Var}[Y|X = x] > 0$ and bounded residual moments
4. Honesty, $\alpha$-regularity($\alpha \leq 0.2$), Symmetry, Random-split
5. Subsample size $s_n$ scales as

$$s_n \asymp n^{\beta} \text{ for some } \beta_{\min} := 1 - \left(1 + \frac{d}{\pi}\frac{\log(\alpha^{-1})}{\log((1-\alpha)^{-1})}\right)^{-1} < \beta < 1$$

the following theorem holds:

$$\frac{\hat{\mu}_n(x) - \mu(x)}{\sigma_n(x)} \Rightarrow N(0,1) \text{ for } \sigma_n(x) \to 0 \text{ and } \frac{\hat{V}_{ij}}{\sigma_n^2(x)} \to_p 1$$

# Rough Sketch of Proof

1. Asymptotic unbiasedness:
   Compute bound of bias using Lipschitz-continuity and honesty
2. Consistency and asymptotic normality:
   Certain conditions on Hájek projection of predictor guarantees the predictor be asymptotically normal
   $\rightarrow$ Those predictors are 1-incremental
   $\rightarrow$ From the properties of k-PNN, all honest and regular random-split trees are $\nu(s)$-incremental
   $\rightarrow$ Randomly subsampling $\nu$-incremental predictors recovers 1-incrementality

Refer to Section 3 and appendix for more technical details.

# Inference on Heterogeneous Treatment Effects

Expand previous results to heterogeneous causal effect estimation.

Specifically, if the following conditions are satisfied,

1. Lipschitz continuity of conditional mean functions
2. Overlap (i.e., $\varepsilon < \mathbb{E}[W = 1|X = x] < 1 - \varepsilon$)
3. Honesty, $\alpha$-regularity($\alpha \leq 0.2$), Symmetry, Random-split
4. Properly scaled subsample size $s_n$

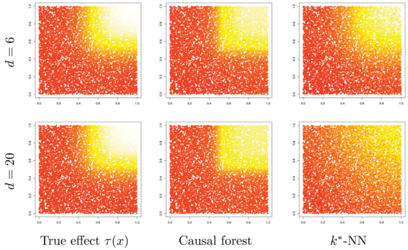
the following theorem holds:

$$\frac{\hat{\tau}(x) - \tau(x)}{\sqrt{\mathrm{Var}[\hat{\tau}(x)]}} \Rightarrow N(0, 1) \;\; \text{and} \;\; \frac{\hat{V}_{ij}}{\mathrm{Var}[\hat{\tau}(x)]} \rightarrow_p 1$$

# Simulation Results

Baseline[2]: Nonadaptive k nearest neighborhood formulated as

$$\hat{\tau}_{\mathrm{KNN}}(x) = \frac{1}{k} \sum_{i \in \mathcal{S}_1(x)} Y_i - \frac{1}{k} \sum_{i \in \mathcal{S}_0(x)} Y_i$$




- Bias effect in high-dimensional setting
- Boundary behavior

[2] Trees and forests can be regarded as *adaptive* nearest neighbor method where model is selected in a data-driven way. The neighbors of $x$ are data points that fall in the same leaf as it.

# R Packages

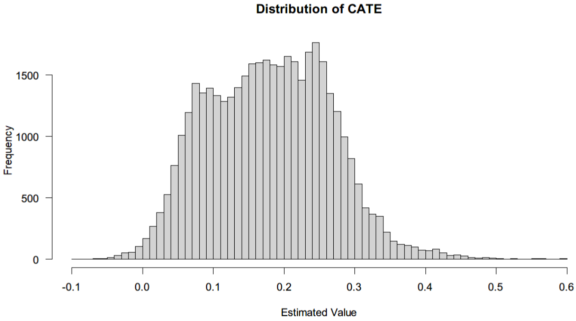
- For building individual trees, use `causalTree`
- For causal forest estimates, use `grf::causal_forest`
  - Refer to github or website for details on implementation

```
causal_forest(X, Y, W,
              num.trees, min.node.size,
              mtry, tune.parameters, ...)
```

- Various diagnostic tools are also available in `grf`

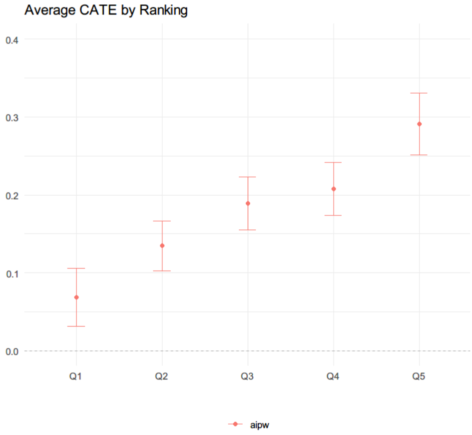
# Example: Marriage Premium Heterogeneity

**Marriage Wage Premium** explores treatment effect of marriage on earnings and labor market outcomes.



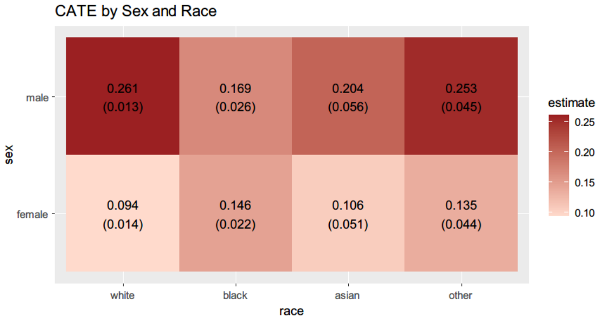
# Example: Marriage Premium Heterogeneity

One can make inference on data-driven subgroups



# Example: Marriage Premium Heterogeneity

... or observe heterogeneity across various features



# Reference


- An Introduction To Statistical Learning with Applications in R Chapter 8
- Wager, S., & Athey, S. (2018). Estimation and inference of heterogeneous treatment effects using random forests. Journal of the American Statistical Association, 113(523), 1228-1242.
- Wager, S., & Athey, S. (2021). Estimating Heterogeneous Treatment Effects in R. Online Causal Inference Seminar.