Statistical Learning

https://github.com/ggorr/Machine-Learning/tree/master/ISLR

8
Tree-Based Methods

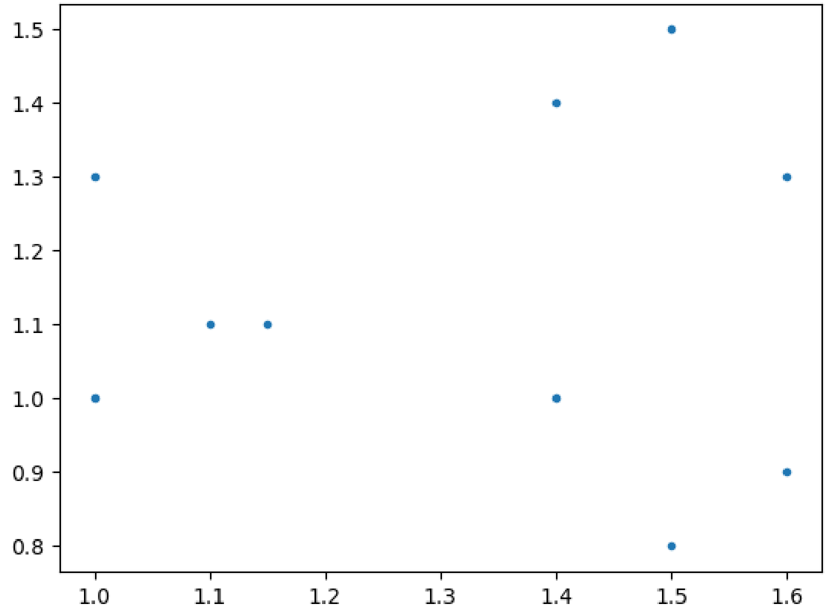
- 8.1 The Basics of Decision Trees
- 8.2 Bagging, Random Forests, Boosting
- 8.3 Lab: Decision Tree
- 8.4 Exercises

8.1 The Basics of Decision Trees

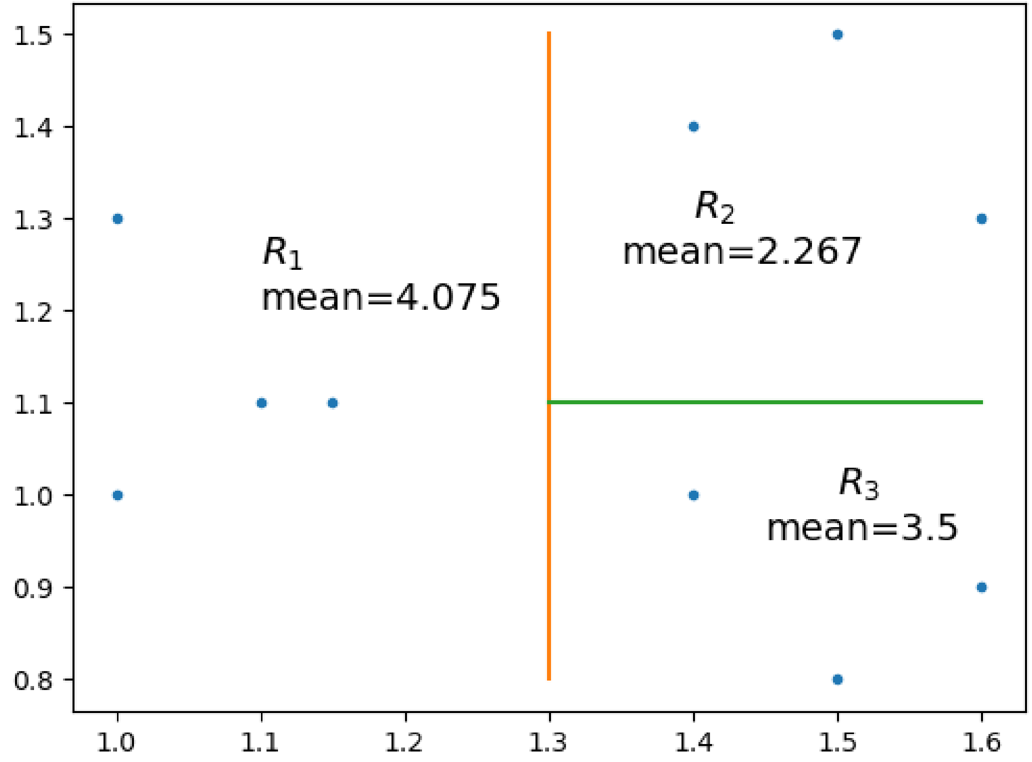
- 8.1.1 Regression Trees
- 8.1.2 Classification Trees
- 8.1.3 Trees Versus Linear Models
- 8.1.4 Advantages and Disadvantages of Trees

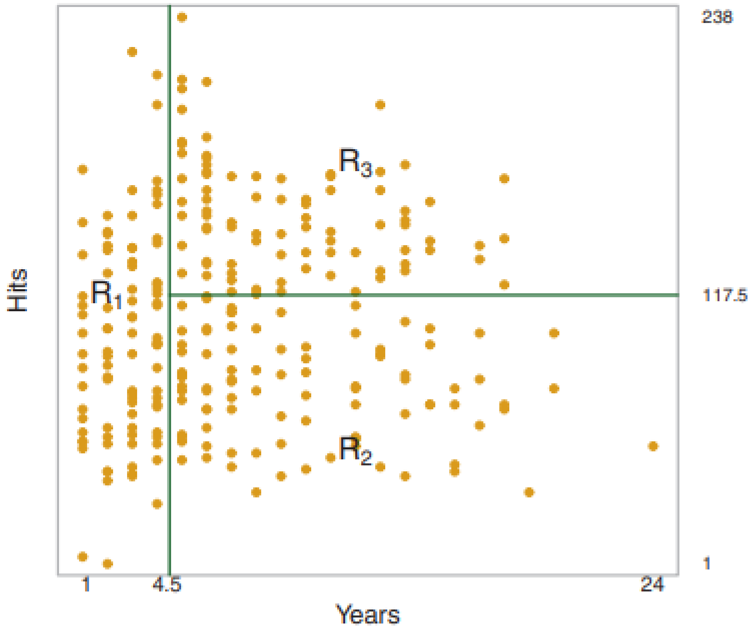
8.1.1 Regression Trees

x_1	1.1	1.0	1.0	1.15	1.5	1.6	1.4	1.4	1.5	1.6
x_2	1.1	1.3	1.0	1.1	1.5	1.3	1.4	1.0	0.8	0.9
y	4.0	4.2	4.0	4.1	2.7	1.8	2.3	3.1	3.9	3.5



- Means: 4.075, 2.267, 3.5





Hits
Years
R_1
R_2
R_3
238
117.5
1
1
4.5
24

Regression Tree

- Divide the predictor space into regions $R_1, \ldots, R_J$
- Prediction:

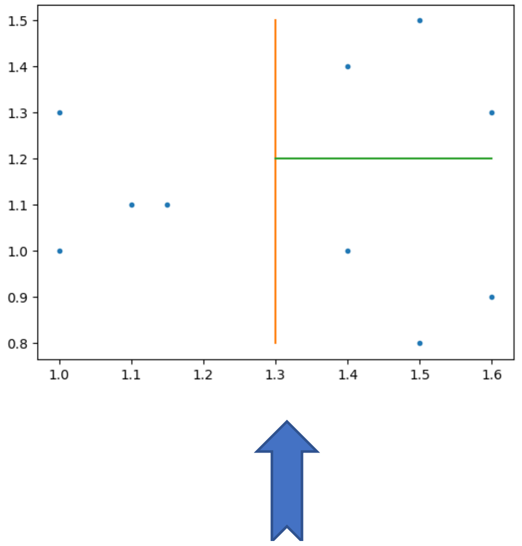
  The mean in the region
- Goal:

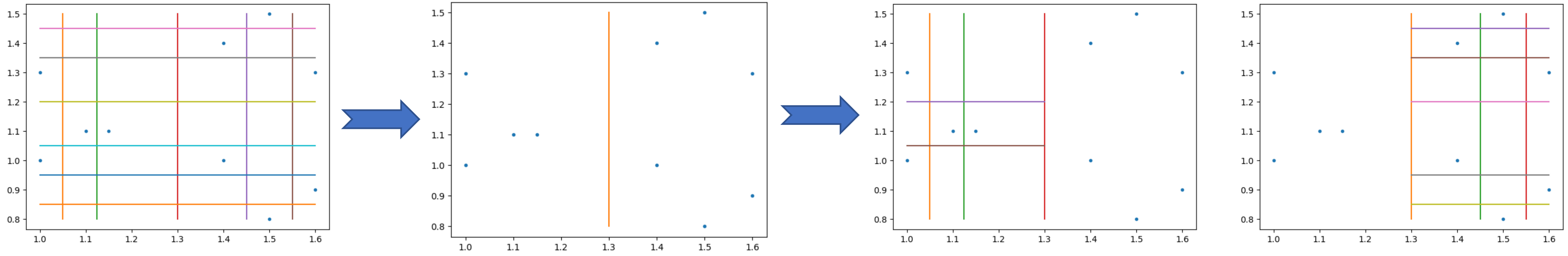
  Minimize the RSS

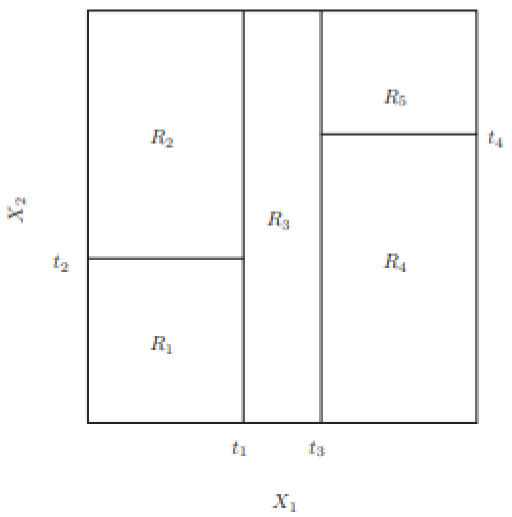
$$\sum_{j=1}^{J} \sum_{i:x_i \in R_j} \left(y_j - \hat{y}_{R_j} \right)^2$$

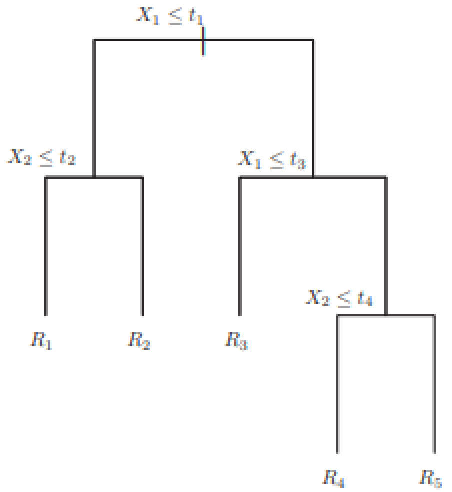
Finding Regions

- Let
$$R_1(j,s) = \{X|X_j < s\},\ R_2(j,s) = \{X|X_j \geq s\}$$
- Seek j and s that minimize the value
$$\sum_{i:x_i \in R_1(j,s)} \left(y_i - \hat{y}_{R_1}\right)^2 + \sum_{i:x_i \in R_2(j,s)} \left(y_i - \hat{y}_{R_2}\right)^2$$
- Repeat this process for $R_1(j,s)$ and $R_2(j,s)$






R_5
R_2
t_4
X_2
R_3
t_2
R_4
R_1
t_1
t_3
X_1


$X_1 \le t_1$
$X_2 \le t_2$
$X_1 \le t_3$
$X_2 \le t_4$
R_1
R_2
R_3
R_4
R_5

Tree Pruning

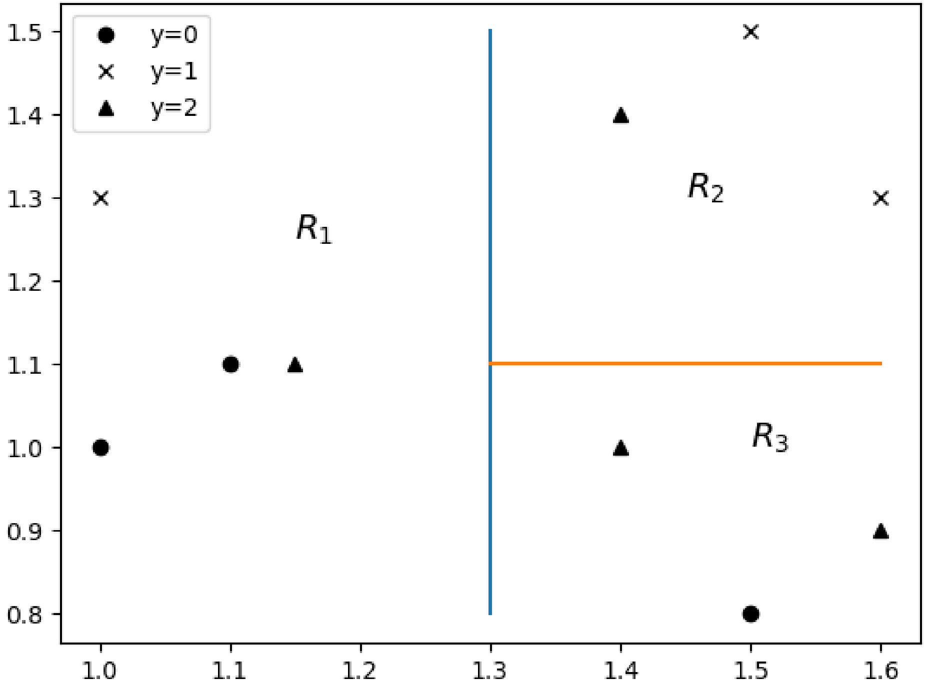
- Large(complex) tree
  - Overfitting
  - Example: n regions for n observations
- Finding small tree with low variance and low bias
- Strategy
  - Start with very large tree
  - Prune to obtain a subtree
- Algorithm
  - Cost complexity pruning

8.1.2 Classification Trees

- Classification tree
  - responses are qualitative
- Decision
  - the most commonly occurring class in each region

Example

x_1	1.1	1.0	1.0	1.15	1.5	1.6	1.4	1.4	1.5	1.6
x_2	1.1	1.3	1.0	1.1	1.5	1.3	1.4	1.0	0.8	0.9
y	0	1	0	2	1	1	2	2	0	2



RSS

- classification error rate

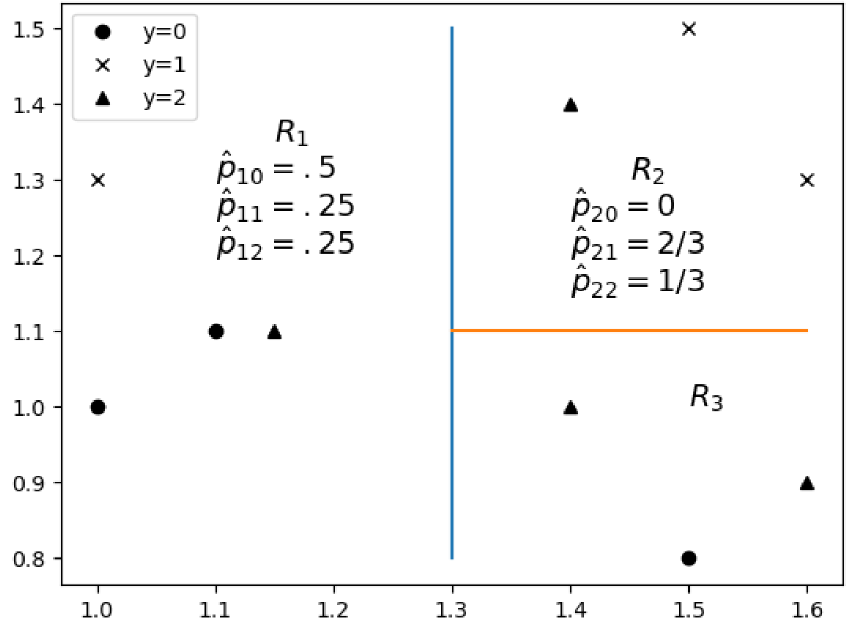
$$E = 1 - \max_k \hat{p}_{mk}$$

  - $\hat{p}_{mk}$: the proportion of the class k in the region R_m
  - not sufficiently sensitive
- Gini index

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

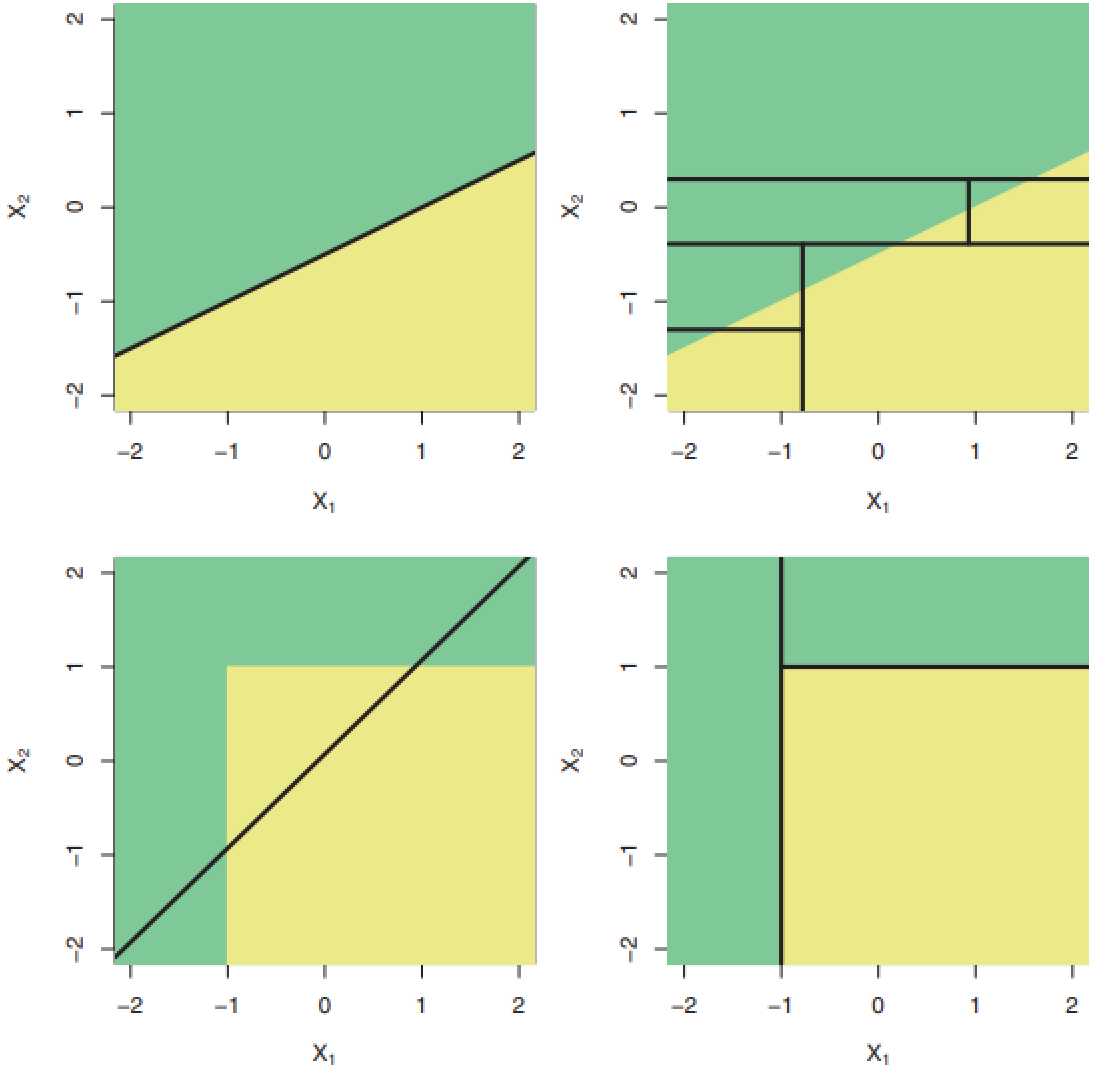
  - measure of node purity
- Cross-entropy

$$D = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$$



8.1.3 Trees Versus Linear Models

- Classification Example
  - two classes green and yellow



8.1.4 Advantages and Disadvantages of Trees

- Trees are
  - easy to explain
  - similar to human decision-making
    - some people believe it!!!
  - not accurate relative to other regression

8.2 Bagging, Random Forests, Boosting

- 8.2.1 Bagging
- 8.2.2 Random Forests
- 8.2.3 Boosting

8.2.1 Bagging

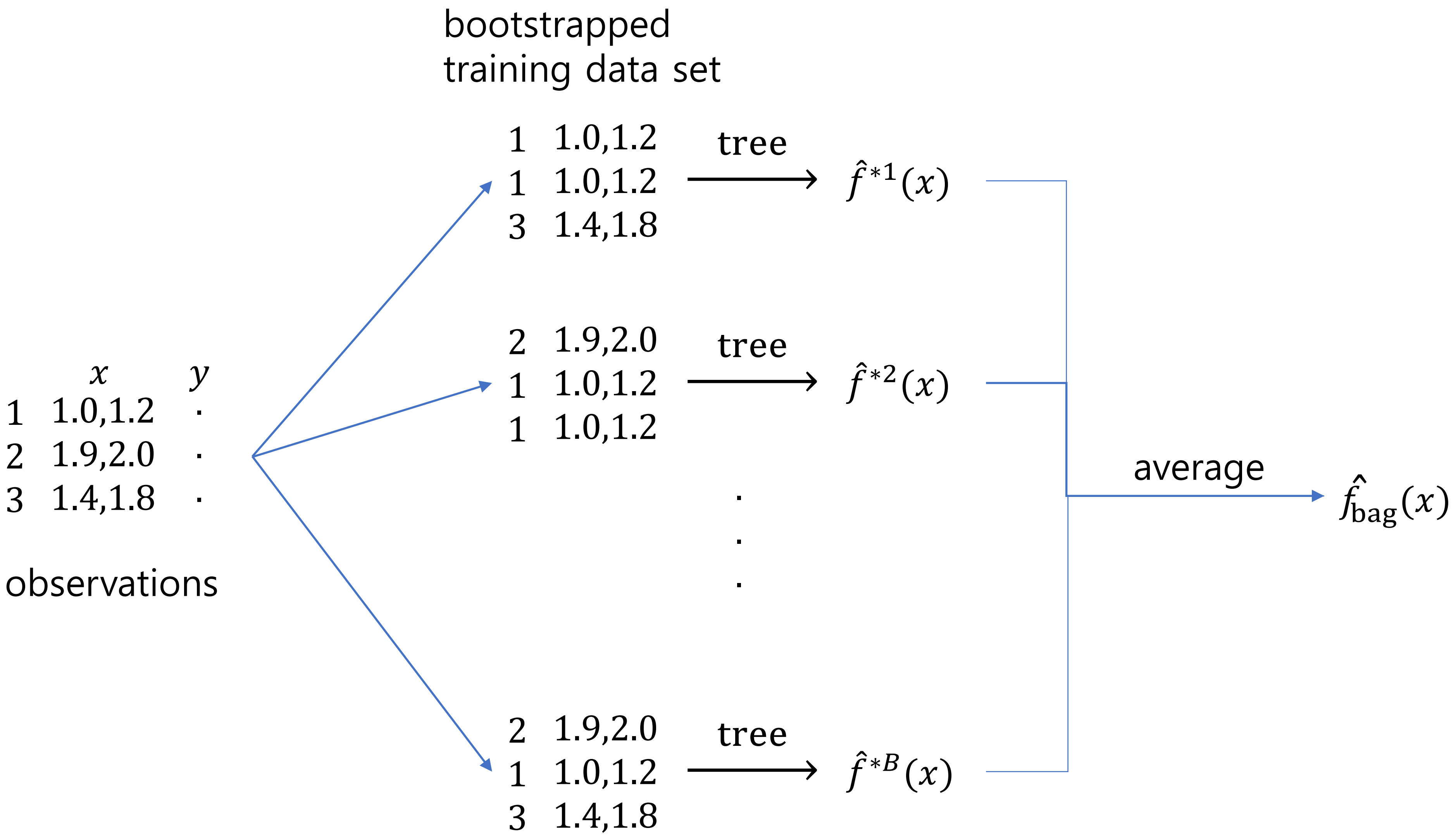
- Bootstrap aggregating = bagging
  - Motivation
    - Decision tree suffer from high variance
    - Averaging a set of observations reduces variance
  - Approach
    - Averaging estimates of bootstrapped training data sets
- Note
  - Bootstrap uses repeated samples with replacement

Bagging in Regression

- find estimate $\hat{f}^{*b}(x)$ for b-th bootstrapped training data set
- averaging

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

  where B is the number of bootstrapped training data sets


bootstrapped
training data set
1 1.0,1.2
1 1.0,1.2
3 1.4,1.8
tree
$\hat{f}^{*1}(x)$
2 1.9,2.0
1 1.0,1.2
1 1.0,1.2
tree
$\hat{f}^{*2}(x)$
2 1.9,2.0
1 1.0,1.2
3 1.4,1.8
tree
$\hat{f}^{*B}(x)$
x
y
1 1.0,1.2
2 1.9,2.0
3 1.4,1.8
observations
average
$\hat{f}_{\mathrm{bag}}(x)$

Bagging with Decision Tree

- Complex tree
  - High variance
- Bagging
  - Averaging without pruning
  - Reduces variance

Bagging in Classification

- Majority vote

Out-of-bag Error Estimation

- Let S_b be the b-th bootstrapped training data set
- An observation x_i is said to be out-of-bag if $x_i \notin S_b$
  - not used in training $\hat{f}^{*b}(x_i)$
  - $\Pr(x_i \in S_b) = 1 - \left(\frac{n-1}{n}\right)^n \approx 1 - \frac{1}{e} \approx \frac{2}{3}$
- Test output
  - regression
    - $\hat{f}^{*}_{\text{test}}(x_i) = \text{average}\left\{\hat{f}^{*b}(x_i) \middle| x_i \notin S_b\right\}$
  - classification
    - $\hat{f}^{*}_{\text{test}}(x_i) = \text{vote}\left\{\hat{f}^{*b}(x_i) \middle| x_i \notin S_b\right\}$

Variable Importance Measures

- average of Gini indices

8.2.2 Random Forests

- Bagging
  - Strong predictors splits tree
  - All of the bagged trees will look quite similar to each other
  - Variance will not be decreased via average
- Random Forest
  - A sort of bagging
  - For each time a split in a tree, a random sample of m predictors are chosen from the full set of p predictors
  - $m \approx \sqrt{p}$

Random Forest Algorithm

- Choose a random sample from observations
  - Build tree
    - Choose m predictors
    - Split a branch
    - Choose another m predictors
    - Split a branch
    - and so on
  - Find the prediction function
- Repeat the process B times
- Average prediction functions

8.2.3 Boosting

- Boosting
  - A sort of decision tree
  - the trees are grown sequentially
    - each tree is grown using information from previously grown trees

Algorithm 8.2 *Boosting for Regression Trees*

1. Set $\hat{f}(x) = 0$ and $r_i = y_i$ for all i in the training set.

2. For $b = 1, 2, \ldots, B$, repeat:

   (a) Fit a tree $\hat{f}^b$ with d splits ($d+1$ terminal nodes) to the training data (X, r).

   (b) Update $\hat{f}$ by adding in a shrunken version of the new tree:

$$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x). \tag{8.10}$$

   (c) Update the residuals,

$$r_i \leftarrow r_i - \lambda \hat{f}^b(x_i). \tag{8.11}$$

3. Output the boosted model,

$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x). \tag{8.12}$$