Introduction to Statistical Machine Learning with Applications in Econometrics

Lecture 8: Tree-Based Methods (ISL ch. 8)

Instructor: Ma, Jun

Renmin University of China

November 23, 2022

Nonparametric regression

- Response Y and p different predictors $X = \left(X_1, X_2, ..., X_p\right)^\top$.
- Let Supp_X denote the set of all possible values (support) of X.
- Our training data consist of $\{(X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n)\}$, where $X_i = \left(X_{1,i}, X_{2,i}, ..., X_{p,i}\right)^\top$.
- $X_{j,i}$: the value of the j-th predictor, or input, for observation i, where $i = 1, 2, ..., n$ and $j = 1, 2, ..., p$.
- An unseen data point: (X_0, Y_0). We know that $f(X_0)$ is an optimal predictor of Y_0, where $f(x) = \text{E}[Y \mid X = x]$ since $f(X_0)$ minimizes the mean square prediction error

$$\text{E}\left[(Y_0 - f(X_0))^2 \mid X_0\right] \leq \text{E}\left[(Y_0 - g(X_0))^2 \mid X_0\right].$$

- The linear model approximates f by $f(x) \approx x^\top \beta$ for some optimal coefficients $\beta \in \mathbb{R}^p$.
- Nonlinear models use more flexible approximation.

- Nonparametric methods such as KNN directly estimate f. Only mild restrictions such as continuous differentiability are imposed. Misspecification bias is avoided.
- A conventional nonparametric regression method falls into one of the two categories:
  - local smoothing (averaging): KNN, local polynomial regression, ...
  - global smoothing: series regression, ...
- Any conventional nonparametric regression estimator $\hat{f}$ suffer from curse of dimensionality: the MSE $\int_{\mathcal{S}_X} \mathrm{E}\left[\left(\hat{f}(x) - f(x)\right)^2\right] \mathrm{d}x$ has a very slow best possible rate of convergence if p is large.
- Conventional nonparametric methods break down when $p \geq 4$.
- These methods have no selection among the predictors that are most useful for prediction.

Tree-based methods

- If the true model is sparse, i.e., $f(X)$ depends on only a small sub-vector of X even when p is large, a nonparametric procedure that selects variables may address curse of dimensionality.
- Tree-based methods are complicated algorithms that implicitly do variable selection.
- These involve segmenting Supp_X into a number of simple regions.
- The set of splitting rules used to segment Supp_X can be summarized in a decision tree.
- To make a prediction, we typically use the mean response value for the training observations in the region to which it belongs.
- They typically are not competitive with the best supervised learning approaches in terms of prediction accuracy.
- Multiple trees are combined: bagging and random forests.

Regression trees

- A partition $\mathcal{S}$ of Supp_X consists of J subsets $\mathcal{S} = \{S_1, S_2, ..., S_J\}$ with $\bigcup_{j=1}^{J} S_j = \text{Supp}_X$ and $S_i \cap S_j = \emptyset, \forall i \neq j$.
- Let $\hat{Y}_j$ be the average response in S_j:

$$\hat{Y}_j = \frac{\sum_{i=1}^{n} Y_i 1\left(X_i \in S_j\right)}{\sum_{i=1}^{n} 1\left(X_i \in S_j\right)}$$

- Predict Y_0 by

$$\hat{Y}_0 = \sum_{j=1}^{J} 1\left(X_0 \in S_j\right) \hat{Y}_j.$$

- The resulting estimator of $f\left(x\right)$ corresponding to the partition $\mathcal{S}$ is given by

$$\hat{f}_{\mathcal{S}}\left(x\right) = \sum_{j=1}^{J} 1\left(x \in S_j\right) \hat{Y}_j.$$

- We may try to choose the partition $\mathcal{S}$ that results in the best in-sample fit:

$$\min_{\mathcal{S} \in \{\text{all partitions}\}} \sum_{i=1}^{n} \left(Y_i - \hat{f}_{\mathcal{S}}\left(X_i\right)\right)^2.$$

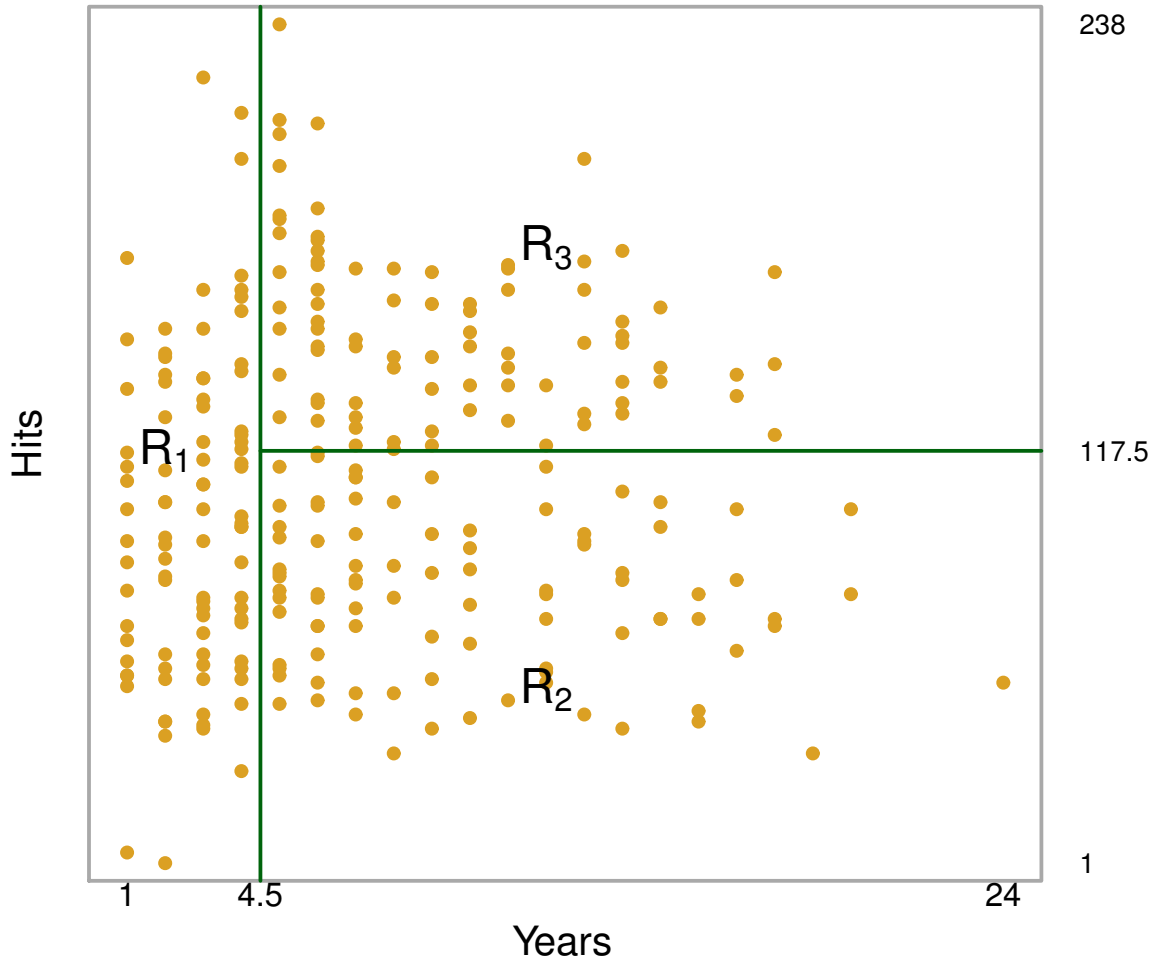
- If X is continuously distributed and $X_i \neq X_j, \forall i \neq j$, this problem has a trivial solution $J = n$ and $\{X_1, ..., X_n\} \cap S_j = \left\{X_j\right\}, \forall j$.
- We may choose $\mathcal{S} \in \{\text{all rectangle partitions}\}$. However, it is still computationally infeasible.
- Regression tree is a constrained and computationally feasible modification of this approach:
  - Partition Supp_X recursively by splitting subsets into halves. One new split at every step of the procedure.
  - Consider one predictor X_j at a time.
  - Use a simple binary rule at every step.
    - If X_j is quantitative, use the rule $X_j \geq c$ or $X_j < c$, for some cutoff point c.
    - If $X_j \in \{1, 2, ..., K\}$ is categorical, use the rule $X_j \in C$ or $X_j \notin C$.
    - Choose j and c (or C) to improve the in-sample fit at every step.

Example: the `Hitters` data

ISL Figure 8.1

- We use the `Hitters` data set to predict a baseball player's `Salary` based on
  - `Years` (the number of years that he has played in the major leagues);
  - `Hits` (the number of hits that he made in the previous year).

- The label $X_j < t_k$ indicates the left-hand branch emanating from that split, and the right-hand branch corresponds to $X_j \geq t_k$.
- The top split assigns observations having `Years` < 4.5 to the left branch. The predicted `salary` for these players is given by the mean response value for the players in the data set with `Years` < 4.5.
- Players with `Years` ≥ 4.5 are assigned to the right branch, and then that group is further subdivided by `Hits`.
- `Years` is the most important factor in determining `Salary`, and players with less experience earn lower salaries than more experienced players.
- Given that a player is less experienced, the number of `hits` that he made in the previous year seems to play little role in his `salary`.
- But among players who have been in the major leagues for five or more years, the number of `hits` made in the previous year does affect `salary`, and players who made more hits last year tend to have higher salaries.



ISL Figure 8.2

Terminology

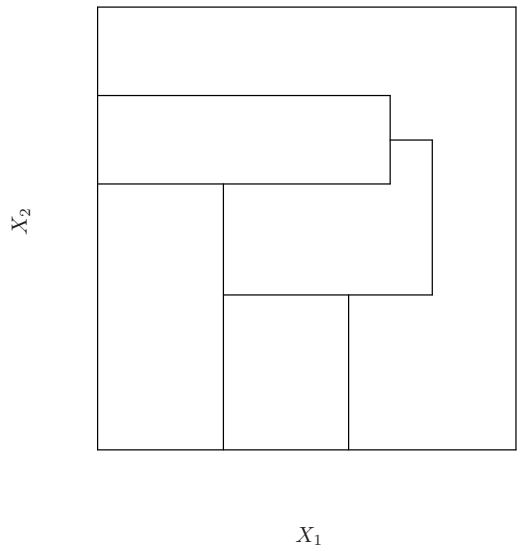
- A tree is composed of a series of nodes, which can be represented by a decision tree.
- Each node T is a subset of Supp_X. The root is just Supp_X.
- A splitting is a process of dividing a node into two child nodes.
- A termination rule is a stopping rule for the splitting process. To avoid overfitting, we require that a node can be split only if the number of observations lying in the node exceeds a threshold (e.g., five).
- A descendant of a node T is a subset of T that results from splitting T.
- A leaf is a terminal node with no descendants. All these leaves form a partition of Supp_X.
- A branch consists of a node and all its descendants.

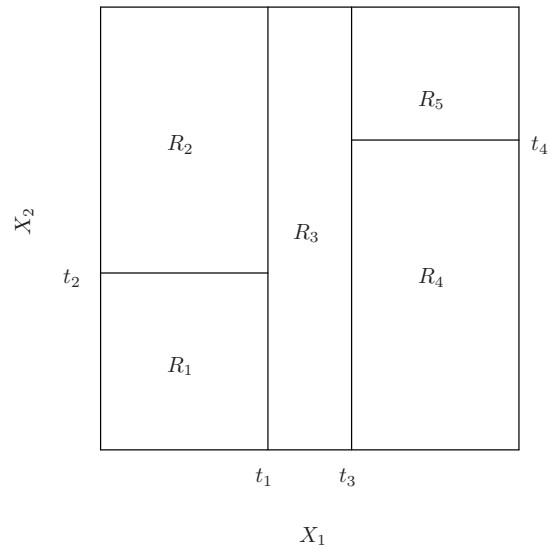
Algorithm

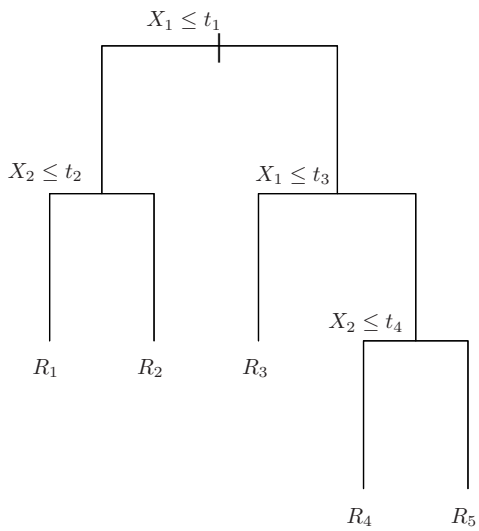
$$RSS_T(j,c) = \sum_{i=1}^{n} \left\{ 1\left(X_i \in T, X_{j,i} \geq c\right) \left(Y_i - \frac{\sum_{i=1}^{n} 1\left(X_i \in T, X_{j,i} \geq c\right) Y_i}{\sum_{i=1}^{n} 1\left(X_i \in T, X_{j,i} \geq c\right)} \right)^2 \right.$$
$$\left. + 1\left(X_i \in T, X_{j,i} < c\right) \left(Y_i - \frac{\sum_{i=1}^{n} 1\left(X_i \in T, X_{j,i} < c\right) Y_i}{\sum_{i=1}^{n} 1\left(X_i \in T, X_{j,i} < c\right)} \right)^2 \right\}.$$

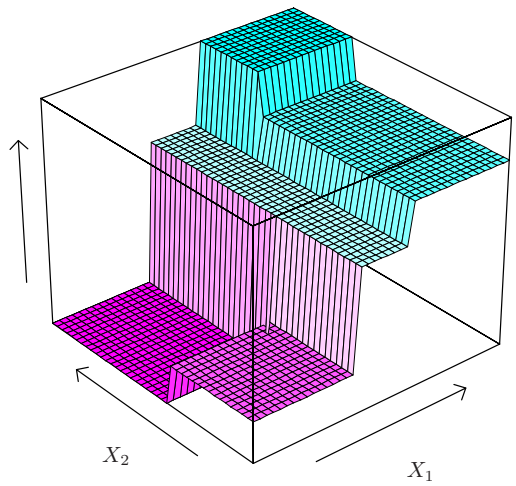
1. Split the root by solving $(j^*, c^*) = \text{argmin}_{j,c} RSS_{\text{Supp}_X}(j,c)$. Two resulting child nodes: $\left\{x \in \text{Supp}_X : x_{j^*} \geq c^*\right\}$ and $\left\{x \in \text{Supp}_X : x_{j^*} < c^*\right\}$ ($x = \left(x_1, ..., x_p\right)^\top$).
2. For each node T that does not meet the termination rule, split T by solving $\min_{j,c} RSS_T(j,c)$. This gives two further child nodes.
3. $\mathscr{L}(\mathscr{T})$ denotes the leaves of the resulting tree $\mathscr{T}$. The estimator:

$$\hat{f}_{\mathscr{T}}(x) = \sum_{T \in \mathscr{L}(\mathscr{T})} \hat{Y}_T 1(x \in T), \text{ where } \hat{Y}_T = \frac{\sum_{i=1}^{n} 1(X_i \in T) Y_i}{\sum_{i=1}^{n} 1(X_i \in T)}.$$









ISL Figure 8.3

- Note the variable selection feature of this algorithm. It is possible that the algorithm decides that one predictor is not useful for prediction and no splitting based on that predictor happens in such a case.
- The process may produce good predictions on the training set, but is likely to overfit the data.
- A large tree may have only a few observations in each leaf. This leads to high variance of $\hat{f}_{\mathcal{T}}(x)$.
- Combining terminal nodes may improve out-of-sample prediction accuracy (lower variance).

Cost complexity pruning

- We grow a very large tree $\mathscr{T}_0$, and then prune it back in order to obtain a subtree.
- A subtree $\mathscr{T}'$ of $\mathscr{T}_0$ is obtained by deleting all the descendants of some node T and making T the terminal node. We denote $\mathscr{T}' \prec \mathscr{T}_0$.
- We penalize the complexity of $\mathscr{T}'$ by

$$\mathscr{T}^* (\alpha) = \underset{\mathscr{T}' \prec \mathscr{T}_0}{\operatorname{argmin}} \sum_{i=1} \left(Y_i - \hat{f}_{\mathscr{T}'} (X_i)\right)^2 + \alpha \left|\mathscr{L} (\mathscr{T}')\right|,$$

  where α is a nonnegative tuning parameter and $|\mathscr{L} (\mathscr{T}')|$ denotes the number of leaves in $\mathscr{L} (\mathscr{T}')$.
- The tuning parameter α can be chosen by cross validation.

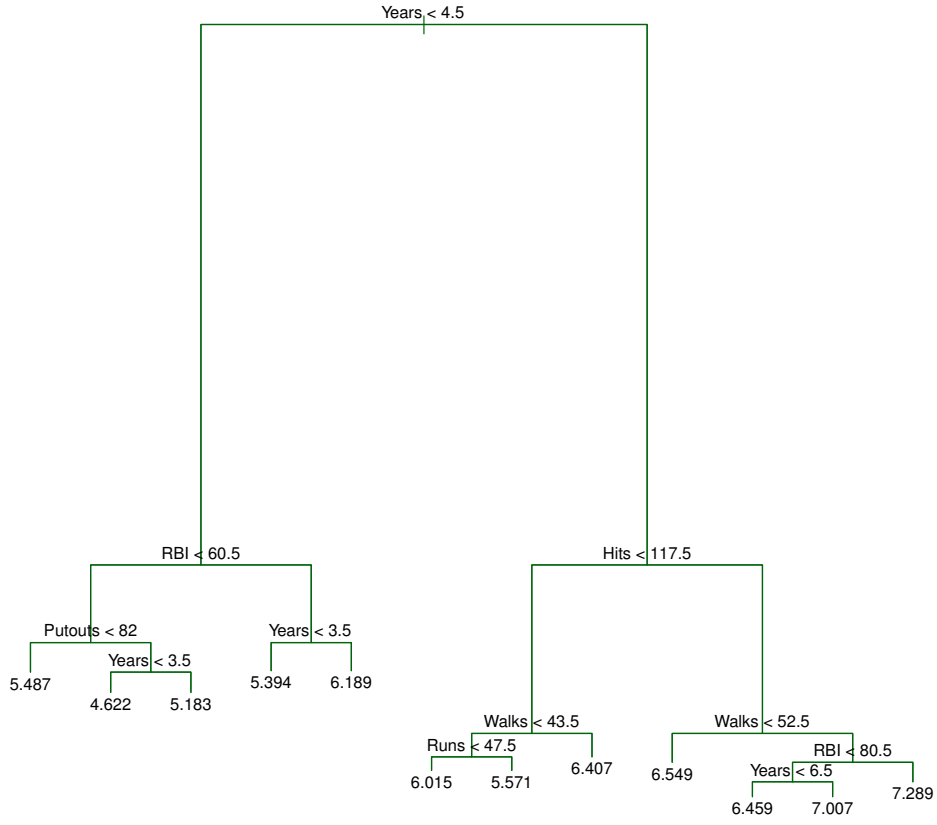
Cross validation for pruning

1. Split the sample randomly into K folds: $C_1, C_2, ..., C_K$ with $\bigcup_{k=1}^{K} C_k = \{1, 2, ..., n\}$.
2. Grow a tree $\mathscr{T}_j$ with the j-th fold held out. Select the optimal subtree $\mathscr{T}_j^* (\alpha)$.
3. Compute the test MSE

$$RSS_j (\alpha) = \sum_{i \in C_j} \left(Y_i - \hat{f}_{\mathscr{T}_j^* (\alpha)} (X_i) \right)^2$$

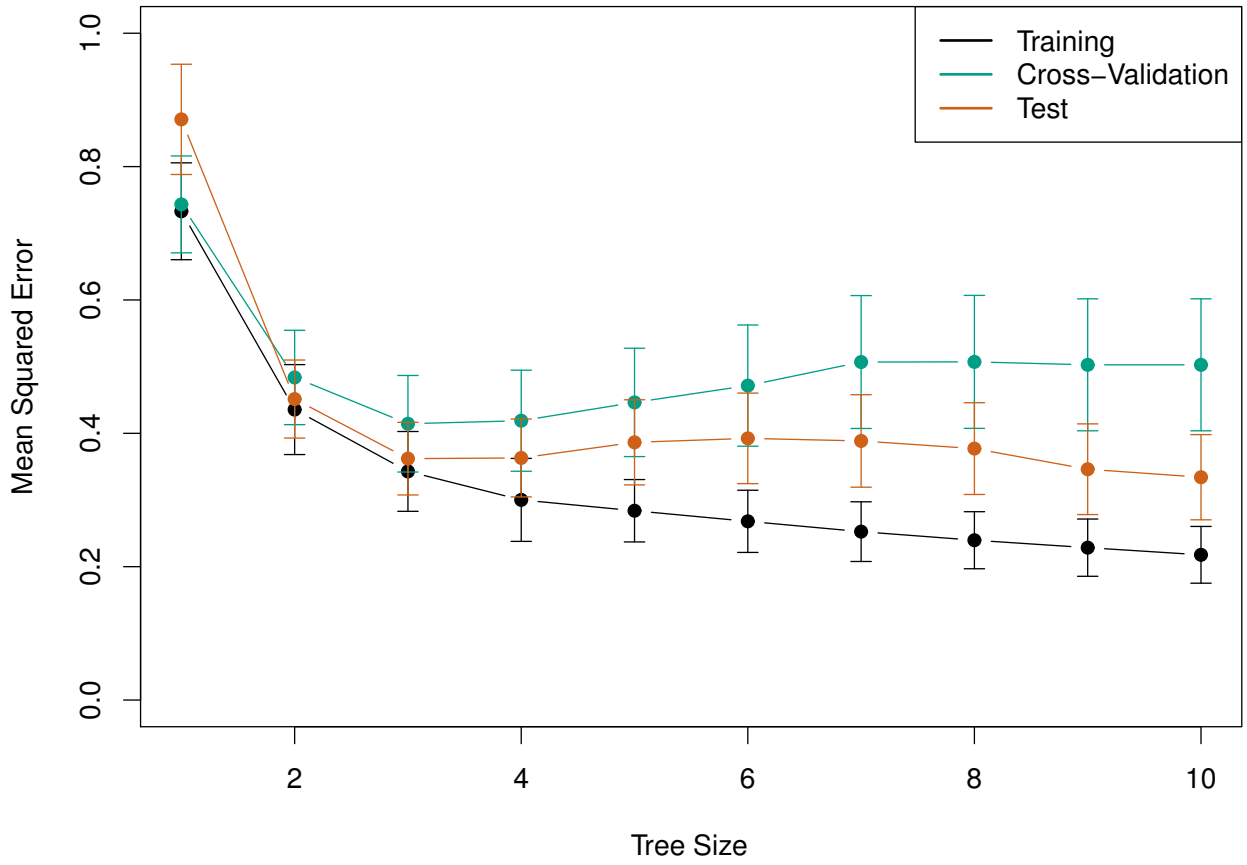
   for each $j = 1, 2, ..., K$. The cross-validated test MSE is $RSS_{CV} (\alpha) = K^{-1} \sum_{j=1}^{K} RSS_j (\alpha)$.
4. Select the tuning parameter α^* by $\alpha^* = \text{argmin}_{\alpha} RSS_{CV} (\alpha)$.
5. Use the tree $\mathscr{T}^* (\alpha^*)$ for out-of-sample prediction.

Example: the `Hitters` data



ISL Figure 8.4

- Divide the data into halves: 132 observations in the training set and 132 observations in the test set.
- Perform 6-fold cross validation with the training set.
- Figure 8.4: the unpruned tree.



ISL Figure 8.5

- The cross validation error is a reasonable approximation of the test error.

Classification trees

- Qualitative response $Y \in \{1, 2, ..., K\}$. We predict that each observation belongs to the most commonly occurring class of training observations in the region to which it belongs.
- A natural alternative to RSS is the misclassification error rate: the fraction of the training observations in that region that do not belong to the most common class.
- $\hat{p}_M(k)$ denotes the proportion of training observations in the region M: $\hat{p}_M(k) = n_M(k)/n_M$, where $n_M = \sum_{i=1}^{n} 1(X_i \in M)$ and $n_M(k) = \sum_{i=1}^{n} 1(X_i \in M, Y_i = k)$.
- Denote

$$\begin{aligned} T_+(j,c) &= T \cap \left\{x \in \text{Supp}_X : x_j \geq c\right\} \\ T_-(j,c) &= T \cap \left\{x \in \text{Supp}_X : x_j < c\right\} \\ E_M &= 1 - \max_{k=1,...,K} \hat{p}_M(k), \end{aligned}$$

- Note that

$$\begin{aligned} E_M &= 1 - \hat{p}_M\left(\underset{k=1,\ldots,K}{\operatorname{argmax}}\, \hat{p}_M(k)\right) \\ &= 1 - \frac{\sum_{i=1}^{n} 1\left(X_i \in M, Y_i = \underset{k=1,\ldots,K}{\operatorname{argmax}}\, \hat{p}_M(k)\right)}{\sum_{i=1}^{n} 1(X_i \in M)} \\ &= \frac{\sum_{i=1}^{n} 1\left(X_i \in M, Y_i \neq \underset{k=1,\ldots,K}{\operatorname{argmax}}\, \hat{p}_M(k)\right)}{\sum_{i=1}^{n} 1(X_i \in M)}. \end{aligned}$$

- Replace $SSR_T(j,c)$ by

$$\begin{aligned} E_T(j,c) &= \sum_{i=1}^{n} \left\{ 1(X_i \in T_+(j,c))\, 1\left(Y_i \neq \underset{k=1,\ldots,K}{\operatorname{argmax}}\, \hat{p}_{T_+(j,c)}(k)\right) \right. \\ &\quad \left. +1(X_i \in T_-(j,c))\, 1\left(Y_i \neq \underset{k=1,\ldots,K}{\operatorname{argmax}}\, \hat{p}_{T_-(j,c)}(k)\right) \right\} \\ &= n_{T_+(j,c)} E_{T_+(j,c)} + n_{T_-(j,c)} E_{T_-(j,c)}. \end{aligned}$$

- However misclassification error is not sufficiently sensitive for tree-growing, and in practice two other measures are preferable:

$$\text{Gini index: } G_M = \sum_{k=1}^{K} \hat{p}_M(k)\left(1 - \hat{p}_M(k)\right).$$

$$\text{Entropy: } D_M = -\sum_{k=1}^{K} \hat{p}_M(k) \log\left(\hat{p}_M(k)\right).$$

- $-\sum_{k=1}^{K} n_M(k) \log\left(\hat{p}_M(k)\right)$ is defined as the deviance in the book.
- Replace E_M by G_M or D_M. Note that

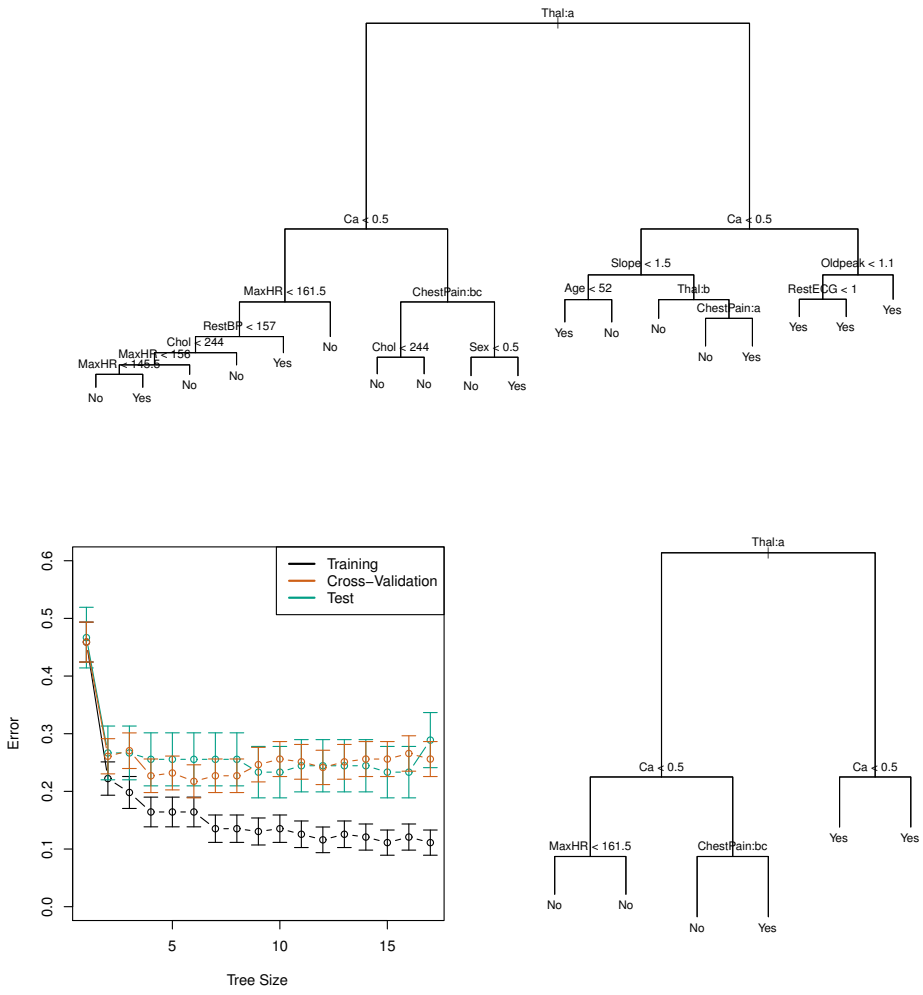
$$\begin{aligned} D_T(j,c) &= n_{T_+(j,c)} D_{T_+(j,c)} + n_{T_-(j,c)} D_{T_-(j,c)} \\ &= -\sum_{k=1}^{K} n_{T_+(j,c)}(k) \log\left(\hat{p}_{T_+(j,c)}(k)\right) \\ &\quad -\sum_{k=1}^{K} n_{T_-(j,c)}(k) \log\left(\hat{p}_{T_-(j,c)}(k)\right). \end{aligned}$$

- The Gini index and the entropy are very similar numerically.
- The Gini index or the entropy takes on a small value if all of the probabilities are close to zero or one.
- Both measure uncertainty or node purity: a small value indicates that a node contains predominantly observations from a single class.
- We predict the type of Y_0 given $X_0 = x$ as

$$\hat{f}_{\mathcal{T}^*(\alpha^*)}(x) = \underset{k=1,\ldots,K}{\operatorname{argmax}} \sum_{T \in \mathcal{L}(\mathcal{T}^*(\alpha^*))} 1(x \in T)\, \hat{p}_T(k).$$

Example: `heart` data

- These data contain a binary outcome `HD` for 303 patients who presented with chest pain.
- An outcome value of `Yes` indicates the presence of heart disease based on an angiographic test, while `No` means no heart disease.
- There are 13 predictors including `Age`, `Sex`, `Chol` (a cholesterol measurement), and other heart and lung function measurements.



ISL Figure 8.6

- Cross-validation yields a tree with six terminal nodes.

- Some of the splits yield two terminal nodes that have the same predicted value.
- Regardless of the value of `RestECG`, a response value of `Yes` is predicted for those observations. The split is performed because it leads to increased node purity.
  - Right-hand leaf 9/9 `Yes`; left-hand leaf: 7/11 `Yes`.
  - Suppose that we have a test observation that belongs to the region given by that right-hand leaf. Then we can be pretty certain that its response value is `Yes`.
  - If a test observation belongs to the region given by the left-hand leaf, then its response value is probably `Yes`, but we are much less certain.

Advantages and disadvantages of trees

- Advantages:
  - Easy to explain to people. Trees can be displayed graphically, and are easily interpreted.
  - Trees more closely mirror human decision-making.
  - Trees can easily handle qualitative predictors without the need to create dummy variables.
- Disadvantages:
  - Not very useful for inference on marginal effects.
  - Small changes in the data can result in a very different tree: high variance
  - Trees generally do not have the same level of predictive accuracy as some of the other regression and classification approaches seen in this book.
  - By aggregating many decision trees, the predictive performance of trees can be substantially improved.

Bagging

- Averaging a set of observations reduces variance.
- Bagging is a general-purpose procedure for reducing the variance of a statistical learning method.
- We take repeated bootstrap samples from the training data set. We generate B different bootstrapped training data sets. We then train our method on the b-th bootstrapped training set in order to get $\hat{f}^{*b}(x)$. The bagging regression estimator is

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

- Pruning is not used since bagging reduces variance by averaging.
- For classification trees: for each test observation, we record the class predicted by each of the B trees, then the overall prediction is the most commonly occurring class among the B predictions.
- B is not a critical parameter with bagging; using a very large value of B will not lead to overfitting.

Out-of-Bag estimation of the test error

- There is a very straightforward way to estimate the test error of a bagged model.
- One can show that on average, each bootstrap sample contains around two-thirds of the observations. The remaining one-third of the observations are referred to as the out-of-bag (OOB) observations.
- We can predict the response for the i-th observation using each of the bootstrap samples in which that observation was OOB.
- Let $\mathcal{B}_i$ denote all bootstrap samples where the i-th observation is OOB. The OOB estimate of the test error is simply:

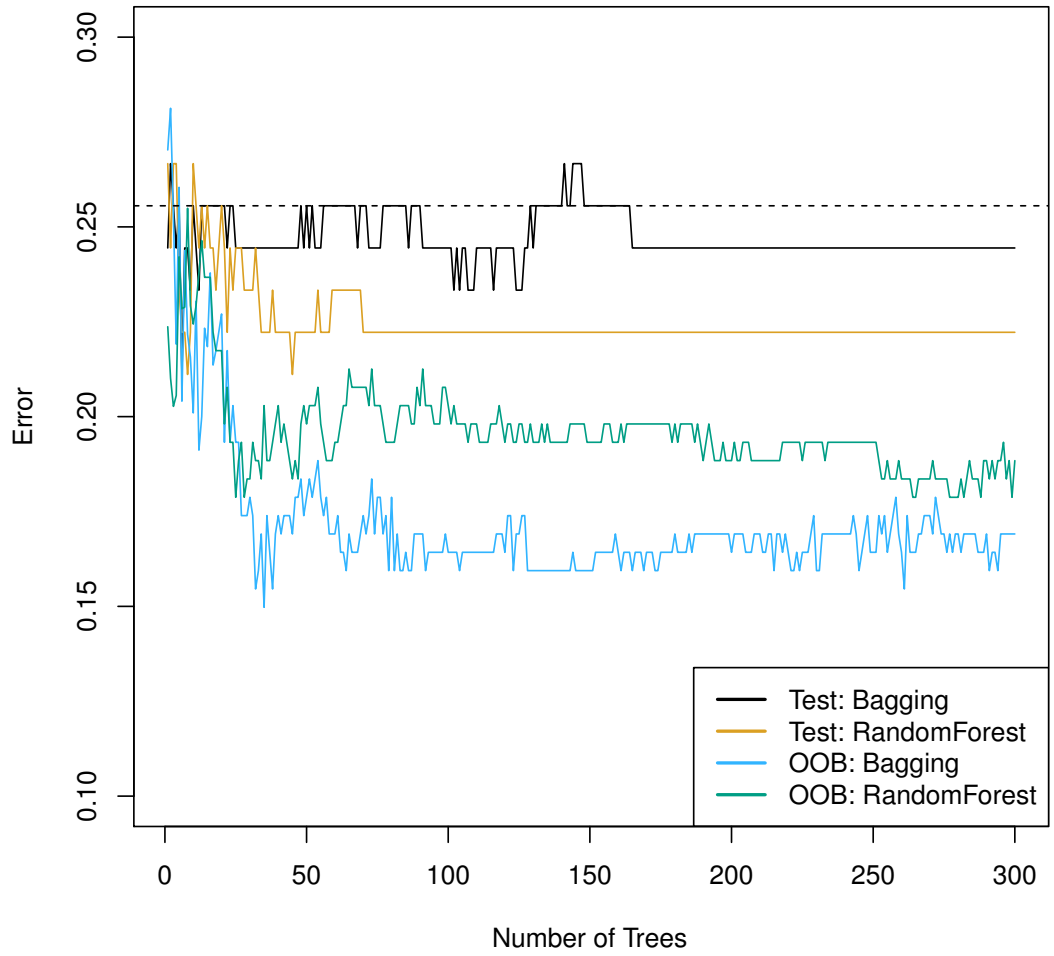
$$\frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \frac{1}{|\mathcal{B}_i|}\sum_{b\in\mathcal{B}_i}\hat{f}^{*b}\left(X_i\right)\right)^2,$$

where $|\mathcal{B}_i|$ denotes the number of elements in $\mathcal{B}_i$.

Random forests

- Bagged trees are highly correlated: often the same predictor is picked in the splits.
- Random forests provide an improvement over bagged trees. This reduces the variance when we average the trees.
- In each split, a random selection of m predictors is chosen as split candidates from the full set of p predictors. The split is allowed to use only one of those m predictors. Typically, $m \approx \sqrt{p}$. This step makes bagged trees less correlated.

Example: `heart` data

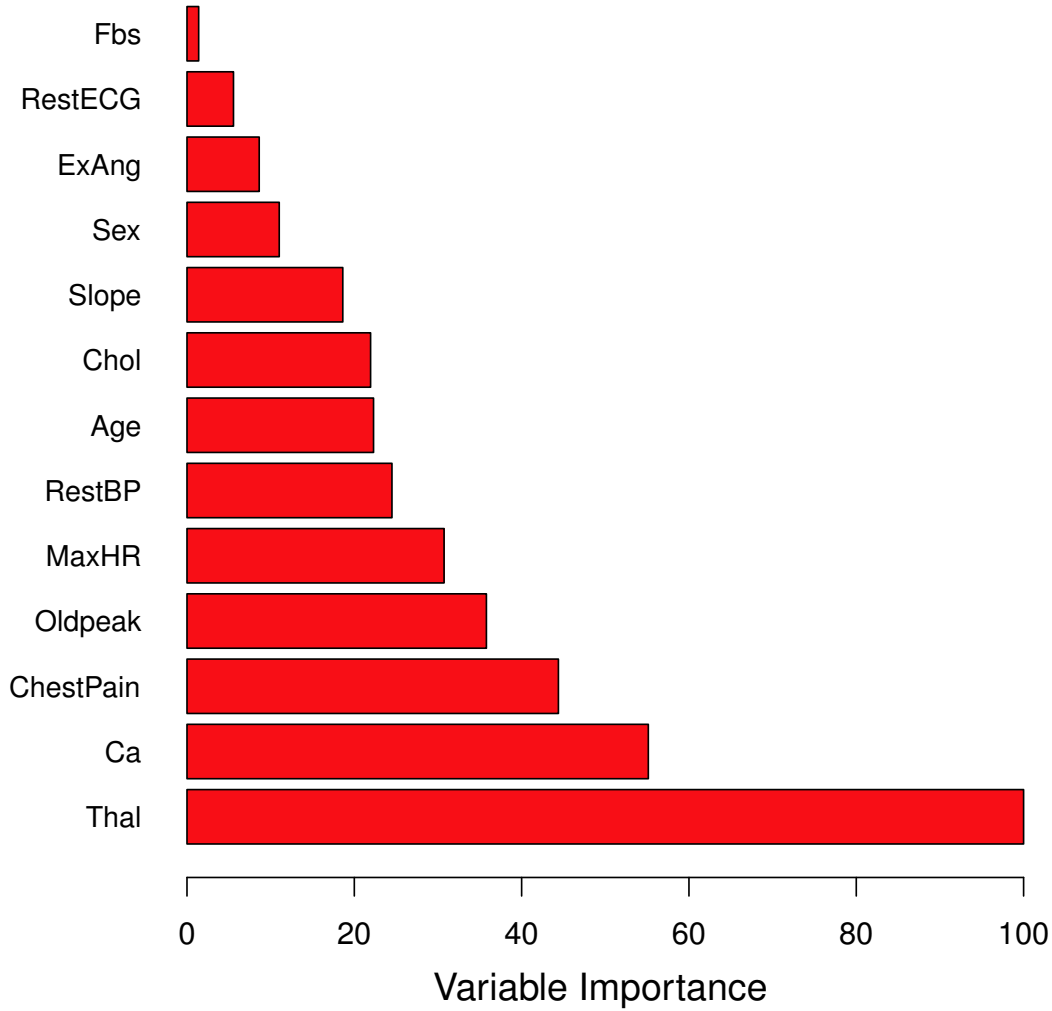


ISL Figure 8.8

Variable importance measures

- One can obtain an overall summary of the importance of each predictor using the RSS (for bagging regression trees) or the Gini index (for bagging classification trees).
- In the case of bagging regression trees, we can record the total amount that the RSS is decreased due to splits over a given predictor, then take the average over all B trees. A large value indicates an important predictor.
- For classification, we can add up the total amount that the Gini index is decreased by splits over a given predictor, then take the average over all B trees.

Example: `heart` data



ISL Figure 8.9