Lecture 17

Classification Trees & Ensemble Methods: Bagging

Murat Unal

Johnson Graduate School of Management
Cornell University

10/28/2021

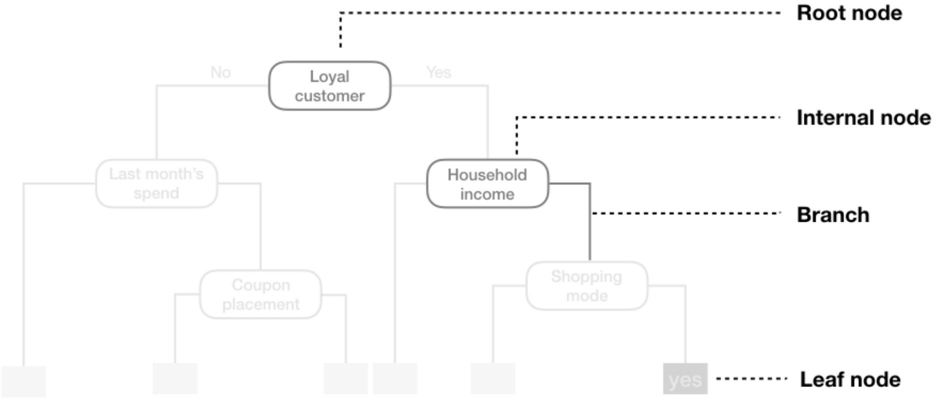
Agenda

Classification Trees

Application in R

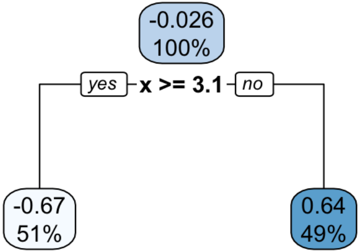
Ensemble Methods: Bagging

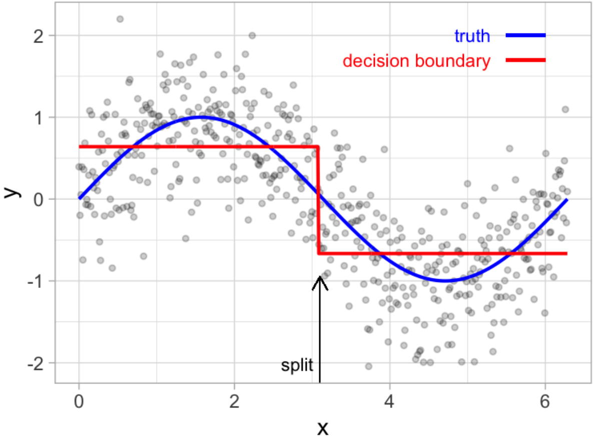
Decision tree



Source: HOML

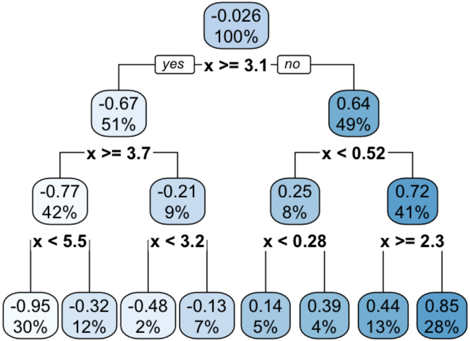
Decision tree with a single split

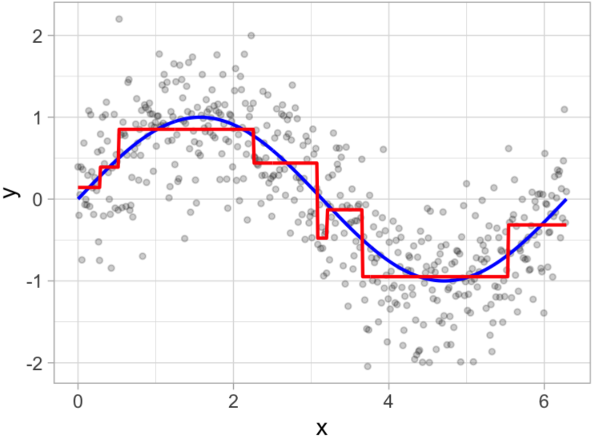




Source: HOML

Decision tree with depth=3





Source: HOML

Growing the tree

- Growing a tree consists of two main steps:
  1. Stratifying the feature space into J regions
  2. Making predictions $\hat{y}_{R_j}$ using the mean outcome of a given region R_j:

$$\hat{y}_{R_j} = \frac{1}{n_j} \sum_{i \in R_j} y$$

- For every observation that falls into the region R_j, we make the same prediction, which is simply the mean of the response values for the training observations in R_j.

Growing the tree

- The regions are chosen by minimizing the RSS across all J regions

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

  , where $\hat{y}_{R_j}$ is the mean response for the training observations within the jth box.

Growing the tree

- **Problem:** It is computationally infeasible to consider every possible partition of the feature space into J regions.
- **Solution:** Take a top-down, greedy approach that is known as recursive binary splitting.
  - recursive: start with the best split, then find the next best split
  - binary: each split creates two branches
  - greedy: the best split is made at that particular step, rather than looking ahead and picking a split that will lead to a better tree in some future step.

Pruning

- The tree building process can result in too many splits
- This will increase flexibility, hence lead to overfitting and reduce interpretability
- A smaller tree with fewer splits (that is, fewer regions) might lead to lower variance and better interpretation at the cost of a little bias

Pruning

- The solution lies in regularization and pruning the tree
- Grow a very large tree T_0, and then prune it back in order to obtain a subtree
- This is called cost complexity pruning

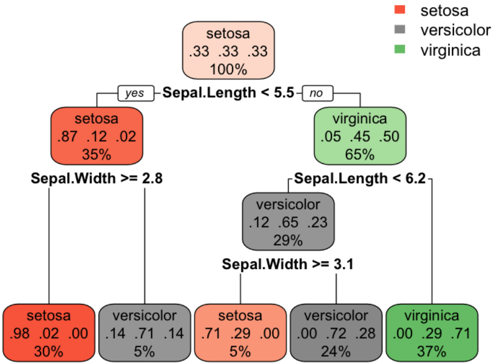
Pruning

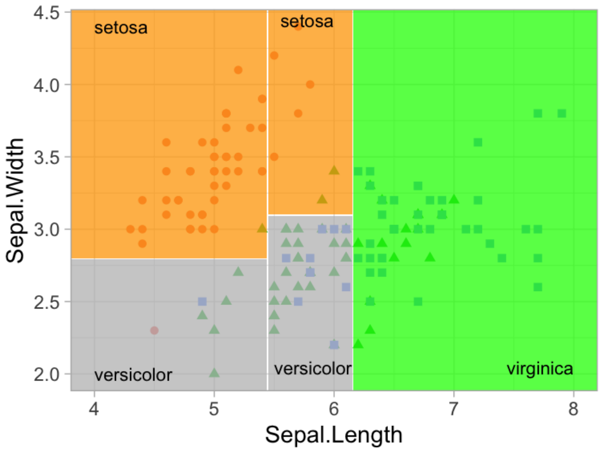
- Just like the Lasso, cost complexity pruning forces the tree to pay a price (penalty) α to become more complex
- We define complexity here as the number of regions $|T|$

$$\sum_{j=1}^{|T|} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2 + \alpha |T|$$

- For any value of α, we get a subtree $T \subset T_0$
- For $\alpha = 0$ we have T_0, as we increase α we start pruning the tree
- We choose $\hat{\alpha}$ via cross validation
- We then return to the full data set and obtain the subtree corresponding to $\hat{\alpha}$

Classification Tree





Source: HOML

Classification trees

- Similar to the regression case, we use recursive binary splitting to grow the tree
- What's different is that RSS can not be used for deciding the plits
- Here we use the Gini Index or Entropy
- Let $\hat{p}_{mk}$ represent the proportion of training observations in the mth region that belong to class k

Classification trees

- Gini Index is defined by :

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

- It is a measure of a node's purity. If a region is very homogeneous (that is, $\hat{p}_{mk} = 0$ or 1) then it is small.
- We aim to minimize the Gini Index

Classification trees

- Entropy is defined by :

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

- It is also a measure of a node's purity. If a region is very homogeneous (that is, $\hat{p}_{mk} = 0$ or 1) then it is small.
- We aim to minimize the Gini Index or Entropy

Ensemble methods

- We can overcome the weaknesses of a single tree by combining many individual trees.
- The following methods use trees as building blocks to construct more powerful prediction models.

1. Bagging
2. Random forests
3. Boosting

Bagging

- Individual decision trees are non-robust, i.e. have high variability.
- Averaging across observations reduces variance.
- Bagging(Bootstrap aggregation) uses this idea by taking repeated samples from the (single) training data set and averaging the results.
- This reduces the variance of the individual trees.

Bagging

- Bootstrapping involves repeatedly drawing independent samples from our data set (Z) to create bootstrap data sets $(Z_1, Z_2 \cdot Z_B)$.
- This sample is performed with replacement, which means that the same observation can be sampled more than once
- And each bootstrap sample will have the same number of observations as the original data set.

Bagging

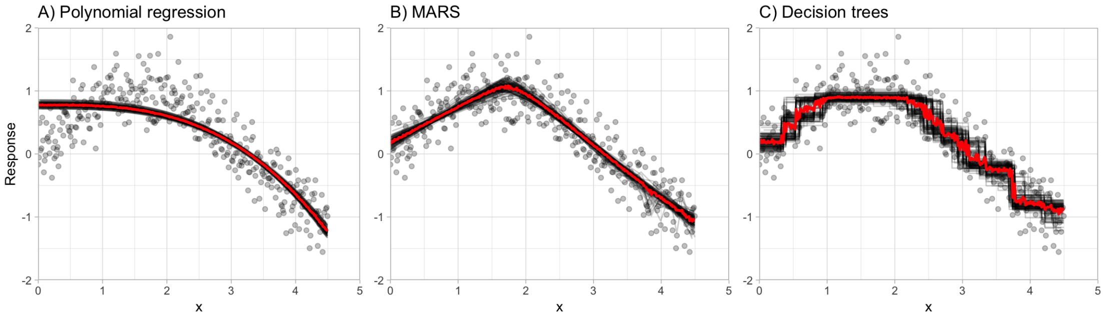
1. Create B different bootstrapped training data sets.
2. Train a decision tree $\hat{f}^b(x)$ on each of the samples.
3. Average all the predictions to obtain:

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

4. The final model is given by $\hat{f}_{bag}(x)$

Bagging

The effect of bagging 100 base learners. High variance models such as decision trees (C) benefit the most from the aggregation effect in bagging, whereas low variance models such as polynomial regression (A) show little improvement



Source: HOML

Bagging

- For regression trees: we typically apply bagging without pruning.
- We grow deep individual trees, which result in high variance and low bias.
- However, averaging ultimately reduces the variance.

Bagging

- For classification trees: for each test observation, we record the class predicted by each of the B trees, and take a **majority vote**, whereby the overall prediction is the most commonly occurring class among the B predictions.
- The number of trees is generally not critical with bagging. $B = 100$ has good performance.

Out-of-bag error estimation

- Bagging offers an easy way to obtain estimates of the test error without the need to do cross validation.
- For any bootstrapped sample, on average, each bagged tree makes use of around two-thirds of the observations.
- The remaining one-third of the observations not used to fit a given bagged tree are referred to as the out-of-bag (OOB) observations.

Out-of-bag error estimation

- We can predict the response for the ith observation using each of the trees in which that observation was OOB.
- This will yield around $B/3$ predictions for the ith observation, which we average.
- This is a useful alternative for CV, because when B and n are large, CV will be computationally intensive.

Variable importance measure

- We can obtain an overall summary of the importance of each feature using the RSS (for bagging regression trees) or the Gini index (for bagging classification trees).

Variable importance measure

- For bagged/RF regression trees, we record the total amount that the RSS is decreased due to splits over a given predictor, averaged over all B trees. A large value indicates an important predictor.
- Similarly, for bagged/RF classication trees, we add up the total amount that the Gini index is decreased by splits over a given predictor, averaged over all B trees.

References

- Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani (2017)
  An Introduction to Statistical Learning
  Springer.
  https://www.statlearning.com/

- Ed Rubin (2020)
  Economics 524 (424): Prediction and Machine-Learning in Econometrics
  Univ, of Oregon.

- Bradley Boehmke, Brandon Greenwell (2020)
  Hands-On Machine Learning with R
  Taylor & Francis Group
  https://bradleyboehmke.github.io/HOML/