New-age classification
RandomForests in eight minutes

Ty Stanford

University of Adelaide

October 19, 2011
Outline

The data

Classification Trees

Example with Trees

RandomForests
We have data

\[ X_i = (X_{i1}, X_{i2}, \ldots, X_{ip}), \quad i = 1, \ldots, n \]

with known group membership in one of \( K \) classes \( G_1, G_2, \ldots, G_K \)

We need to create a decision rule if we get an observation \( X_{n+1} \) of unknown class. What group do we assign it to?
Briemann [1984] was the first literature to bring them into the statistician’s realm.
Aim of classification trees

- From the root node successively split the data at nodes

- for new observations $\mathbf{x}_{n+1}$ you simply traverse the tree down the nodes until you reach a leaf node - thus classify $\mathbf{x}_{n+1} \in \{G_1, G_2, \ldots, G_K\}$

There are two things that need to be considered:

1. Splitting rules (node impurity)
2. Determining the final leaf nodes (stopping rule)
Splitting rules

- Splitting is done on a feature/variable from the possible features (feature set).
- Say a splitting value $c_m$ for feature $v_m$ on node $m$ is used.
- Observations with values $x_{v_m}$ less than $c_m$ are classified to the left daughter node.
Node impurity

Before talking about the variable we use at each node to split the data...

We need to talk about node impurity

- a measure (or a function) of the misclassification at a node
- minimise
- unfortunately there is no one measure which is considered best (e.g. regression trees use least squares)
- this measure will help us to chose the best variable to use as the next split
The two favoured impurity measures $i_m(T)$ are

Gini: $i_m(T) = \sum_{j=1}^{K} \hat{p}_{mj} (1 - \hat{p}_{mj})$

Entropy: $i_m(T) = - \sum_{j=1}^{K} \hat{p}_{mj} \log_2 \hat{p}_{mj}$

where $\hat{p}_{mj}$ is the proportion of observations, $x_i$s, in group $j$ at node $m$. 
Create a tree

We now have the machinery to create a classification tree:

1. for each (as yet unused) feature, find the value of that variable that minimises the node impurity
2. then scan all features for the feature that has the smallest minimised node impurity
3. this feature and splitting value creates a left and right daughter node
4. repeat until all nodes have 'pure' data.
Classification Tree (Entropy impurity)
Classification Tree (Gini impurity)

Decision Tree: Gini

Gene1955 < 0.2585
Gene545 ≥ 0.1841
Gene153 < -0.2133

Log_2 Expression Ratio

Tree Decision Process
Random Forests (Briemann, 2001)

To understand the Random Forest algorithm, consider the 2 class problem.

- We build $N_T$ trees (to make a “forest”).
- Each of these $N_T$ trees are constructed as follows...
m observations selected from the total n observations WITH replacement

At each node, randomly select a (new) subset of q variables (where q << p) to inform a recursive split of the data
Classification with RandomForest

Each tree votes for what an input’s classification is, the class with the most votes wins
Pros/Cons

Advantages:

▶ It can handle multi-class data easily.
▶ Makes use of the framework of trees, but takes away some of the disadvantages of trees via the imposed randomness.
▶ Feature importance takes into account the sometimes complex relationships with other variables.
▶ There are other feature importance measures in-built that can be used.
Pros/Cons

Disadvantages:

▶ As we are fitting an entire forest we now no longer have an easily understandable/visualised method.
▶ Some of the literature claims it does not ‘over-fit’ the data but this is still debated.
▶ Selection of the training set may drastically alter the model.
Feature Selection

How do we get variable importance from the Random Forest algorithm?

▶ We have $N_T$ trees and we have $n_{OOB}$ 'Out-Of-the-Bag' (not sampled in the tree construction) observations for each tree.
▶ We aggregate the error of each tree with the OOB test sets.
▶ Now to test the $i$th feature's importance:
  ▶ Randomly permute the values of the $i$th feature in the OOB test sets.
  ▶ Now get the errors of the trees with these altered OOB test sets.
  ▶ The increase in error from the original OOB test error is an indication of the feature's importance.
Feature Selection

How do we get variable importance from the Random Forest algorithm?

- We have $N_T$ trees and we have $n_{OOB}$ ‘Out-Of-the-Bag’ (not sampled in the tree construction) observations for each tree.
Feature Selection

How do we get variable importance from the Random Forest algorithm?

- We have $N_T$ trees and we have $n_{OOB}$ ‘Out-Of-the-Bag’ (not sampled in the tree construction) observations for each tree.
- We aggregate the error of each tree with the OOB test sets.

Now to test the $i^{th}$ feature's importance:
- Randomly permute the values of the $i^{th}$ feature in the OOB test sets.
- Now get the errors of the trees with these altered OOB test sets.
- The increase in error from the original OOB test error is an indication of the feature's importance.
Feature Selection

How do we get variable importance from the Random Forest algorithm?

- We have $N_T$ trees and we have $n_{OOB}$ ‘Out-Of-the-Bag’ (not sampled in the tree construction) observations for each tree.
- We aggregate the error of each tree with the OOB test sets.
- Now to test the $i^{th}$ features importance:
How do we get variable importance from the Random Forest algorithm?

- We have $N_T$ trees and we have $n_{OOB}$ ‘Out-Of-the-Bag’ (not sampled in the tree construction) observations for each tree.
- We aggregate the error of each tree with the OOB test sets.
- Now to test the $i^{th}$ features importance:
  - Randomly permute the values of the $i^{th}$ feature in the OOB test sets.
Feature Selection

How do we get variable importance from the Random Forest algorithm?

- We have $N_T$ trees and we have $n_{OOB}$ ‘Out-Of-the-Bag’ (not sampled in the tree construction) observations for each tree.
- We aggregate the error of each tree with the OOB test sets.
- Now to test the $i^{th}$ features importance:
  - Randomly permute the values of the $i^{th}$ feature in the OOB test sets.
  - Now get the errors of the trees with these altered OOB test sets.
Feature Selection

How do we get variable importance from the Random Forest algorithm?

- We have $N_T$ trees and we have $n_{OOB}$ ‘Out-Of-the-Bag’ (not sampled in the tree construction) observations for each tree.
- We aggregate the error of each tree with the OOB test sets.
- Now to test the $i^{th}$ features importance:
  - Randomly permute the values of the $i^{th}$ feature in the OOB test sets.
  - Now get the errors of the trees with these altered OOB test sets.
  - The increase in error from the original OOB test error is an indication of the feature’s importance.
Feature Selection

$n_{OOB}$

$n_{OOB}$
Normalised spectra heatmap by group (Adam et al., 2002)