ML Section 1.2: Decision Tree Learning

Adam Hawley

February 27, 2019

Contents

1 Intro **1**

2 Decision Tree Learning Algorithm **2**
2.1 Feature Choice 2
2.2 Furter Remarks 2
2.3 Properties of Decision Tree Learning 3

3 Pruning **3**

4 Numerical Features **3**

5 Missing Feature Values **3**

6 Constructive Induction (Think automated feature transformation or selection) **4**
6.1 Types 4
6.2 Feature Selection 4
6.2.1 Filter Methods 4
6.2.2 Wrapper Methods 4

1 Intro

- Training examples are represented as feature-value vectors.
- Each feature denotes some property of an example.
- Feature values can be continuous, but will be discretised beforehand.

2 Decision Tree Learning Algorithm

1. If training examples at the root node are perfectly classified, then stop.
2. Choose feature to test at root node.
3. A child of the root node is created for each value of the root feature.
4. Training examples are sorted to the children according to feature test.
5. Repeat steps 1-5 for each child (viewing it as the root of the new subtree).

2.1 Feature Choice

Choose the feature which is most useful for classifying examples. The quality measure for choosing a feature is the *information gain* — measuring how well a given feature separates the training examples to their categories.

When deciding which feature is best we do the following:

- Let S be the set of training examples, p^+ be the proportion of examples of class C1 and p^- be the proportion of examples of class C2.
- **Entropy** measures the *impurity* of S:
  - `E(S) = (-p^+*log(p^+)) + (-p^-*log(p^-))`
- Let F be a feature, and let S_v be the elements in S with F = v.
- Gain(S,F) is expected reduction in entropy due to sorting on F:
  - `Gain(S,F) = E(S) - \Sigma_(V\subset Values(F) ((|S_v|/|S|)E(S_v)`
  - *Best* feature is the feature F with the highest Gain(S,F)

SEE LECTURE FOR (HELPFUL EXAMPLE)

2.2 Furter Remarks

- Numerical features are discretised, i.e.feature values are denoted by intervals.
- After a decision tree is generated, it is simplified (*pruned*) to avoid **overfitting** of the training data.
- Bias of decision tree learner: the simpler tree that fits the training data is the better one.

2.3 Properties of Decision Tree Learning

Decision tree learning (DTL) searches a complete hypothesis space (all finite discrete-value functions can be expressed as decision trees). DTL also maintains only a single current hypothesis. DTL never revisits choices once they are made (no backtracking). It is also non-incremental (off-line) — this means that it takes all of the training data in batch form rather than one entry at a time.

3 Pruning

The goal of pruning is to avoid overfitting the training data. This can be done by dividing training data into a training and validation set. To prune a decision node, remove the subtree located at the node, making it a leaf node and assign it the most common classification of the training examples affiliated with this node. Nodes are removed if the pruned tree performs no worse on the validation data. Pruning order is guided by maximum reduction in error.

4 Numerical Features

What if features are numerical, e.g. number of As in sequence?

- Discretise the values by introducing intervals.
- Sort numerical feature values and pick interval bounds that are between changes in category values.
- Each bound c defines a boolean feature (true if feature value $> c$, and false otherwise).

5 Missing Feature Values

Solution 1 When a feature F is evaluated, assign missing values of F the most common value in the training examples of the respective category.

Soultion 2 Compute probabilities of feature values based on occurences in training data, and distribute examples with unknown feature values to the children nodes according to these probabilities.

6 Constructive Induction (Think automated feature transformation or selection)

6.1 Types

Feature Selection Select a (minimal) subset of features according to some criteria so that the learning performance (speed and quality) can be improved.

Feature Transformation

- Extracting a set of new features from the original features through some functional mapping.
- Discovering missing information about the relationships between features and augmenting the space of features by inferring or creating additional features.

6.2 Feature Selection

6.2.1 Filter Methods

- Filter features according to specific criteria
- *Examples*: Select top n features according to information gain, frequency, entropy etc.
- Run a DTL on training data and remove all features that are not mentioned in the resulting decision tree.

6.2.2 Wrapper Methods

- Evaluate the performance of the learner on a given feature subset.
- **Backward Elimination**: start with full set of features and greedily remove features one by one maximising accuracy increase at each step.
- **Forward Selection**: start with empty feature set and greedily add features.