Decision Tree with the ID3 Algorithm

Chanda Tirtha , Chourasiya Shobhit , Jambigi Neetha

November 19, 2017

Abstract

The document describes approach for classifying data using decision tree using ID3 algorithm for a Car Evaluation Database.

1 Introduction

ID3 algorithm builds a decision tree from a fixed set of examples. Here we use this algorithm to classify car data. The resulting tree could be used to classify future samples. The leaf nodes of the decision tree contain the class name whereas a non-leaf node is a decision node. The decision node is an attribute test with each branch (to another decision tree) being a possible value of the attribute. ID3 uses information gain to help it decide which attribute goes into a decision node.

2 Implementation

2.1 Idea:

Building the decision tree using the ID3 algorithm:

1. We begin with the original set of attributes as the root node.
2. On each iteration of the algorithm, we evaluate every unused attribute of the remaining set and calculate the entropy and information gain of that attribute.
3. Then, we select the attribute which has the largest information gain value.
4. The set of remaining data and attributes is then split by the selected attribute to produce subsets of the data. The split is done recursively by passing the new data and attributes to the same function. The algorithm continues to recurse on each subset, considering only attributes never selected before.
5. The leaf nodes contain the classified instances.

2.2 Pseudo code

Call the build tree function recursively until the leaf nodes

```
Input : Full data set , Attributes list  and target attribute name.
FUNCTION build tree ( data, attributes, target ):
        Get the dataset for this iteration
        if (  all records in dataset have same classification ) :
                return classification
         if (  attributes is empty && entropyOfDataset != 0 ):
                return Majority(classification)
        else:
                best_attribute <-  chooseBestAttribute(dataset, attributes, target)
                for each value in best_attribute:
                        new_attributes <- remove the chosen attribute from attributes
                        data_for_subtree <- get data for the sub tree
                        subtree <- call build_tree() recursively for the new data &
                        new_attributes
```

The choose_best_attribute function returns the attribute with the maximum information gain amongst the input attribute list.

```
Input: data set , attributes list , target  attribute name

FUNCTION choose_best_attribute ( data, attributes, target ):
        For < all attributes >:
                new_information_gain <- Calculate information gain ( attribute )
                RETURN (attribute with the best Information gain)
```

- The formula for calculating Information gain used in function information gain

```
information_gain = Entropy ( full set ) – Entropy ( subset )

Entropy calculation:
        key_probability = Frequency of Key / total_data_entries
        entropy - = key_probability * ( log ( key_probability , 4))
```

3 Resultant Tree

The generated tree classifies the car data as follows: