CSE 4020 – MACHINE LEARNIG LAB
FACULTY : PROF. SUGANYA G
DHRUBANKA DUTTA – 17BCE1019
L – 11,12 , 6.8.19
LINEAR REGRESSION

Decision Tree

1. Abstract

Decision tree is a powerful tool used in machine learning for classification of data. It is like a flowchart tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test and each leaf node holds a class label. Numerical values are allowed but only nominal and not continuous values.
Decision trees implicitly perform variable screening or feature selection. It requires relatively less effort for data preparation. Non linear relations do not affect the final result of the classification. The models built from decision trees are easy to understand and implement.
Place the best attribute of the data set at the root of the tree. Split the training set into subsets. Subsets should be made in such a way that each subset contains data with the same value for an attribute. Repeat these steps on each subset until you find leaf nodes in all the branches of the tree.

2. Introduction

Decision trees are used in real life applications where the order of selection of features matters and there is an impact of the previous choice on the next. Some examples are -

a) Selecting the best choice out of all possible alternatives.
b) Churn Analysis
c) Investment Solutions

3. Methodology

The following assumptions are made while using a decision tree:

- At the beginning, the whole training set is considered as the root.
- Feature values are preferred to be categorical. If the values are continuous then they are discretized prior to building the model.
- Records are distributed recursively on the basis of attribute values.
- Order to placing attributes as root or internal node of the tree is done by using some statistical approach.

Initially the entropy of all the attributes are calculated. The attribute with the highest entropy is designated as the root node. For the next attribute , the information gain is calculated

4. Data set

The data set comprises of three features – Taste, Temperature, Texture. These describe the different conditions of the food and based on these it has to be decided whether it is edible or not. A decision tree is used for this because the answer of different attributes has different weightage. Ex – It will be more important that the food is edible rather than it being hot or cold.

5. Algorithm

```
id3(examples, attributes)
  '''
  examples are the training examples.  attributes is a list of
  attributes that may be tested by the learned decison tree.  Returns
  a tree that correctly classifies the given examples. Assume that
  the targetAttribute, which is the attribute whose value is to be
  predicted by the tree, is a class variable.
  '''
  node = DecisionTreeNode(examples)
  # handle target attributes with arbitrary labels
  dictionary = summarizeExamples(examples, targetAttribute)
  for key in dictionary:
     if dictionary[key] == total number of examples
       node.label = key
       return node
  # test for number of examples to avoid overfitting
  if attributes is empty or number of examples < minimum allowed per branch:
    node.label = most common value in examples
    return node
  bestA = the attribute with the most information gain
  node.decision = bestA
  for each possible value v of bestA:
    subset = the subset of examples that have value v for bestA
    if subset is not empty:
      node.addBranch(id3(subset, targetAttribute, attributes-bestA))
  return node

infoGain(examples, attribute, entropyOfSet)
  gain = entropyOfSet
  for value in attributeValues(examples, attribute):
     sub = subset(examples, attribute, value)
     gain -=  (number in sub)/(total number of examples) * entropy(sub)
  return gain

entropy(examples)
  '''
  log2(x) = log(x)/log(2)
  '''
  result = 0
  # handle target attributes with arbitrary labels
  dictionary = summarizeExamples(examples, targetAttribute)
  for key in dictionary:
     proportion = dictionary[key]/total number of examples
     result -= proportion * log2(proportion)
  return result
```

6. Results and Analysis

Execution results and comparison of results

A) #Testing1

```
test_data = pd.Series({'Age':'<=30','Income':'High','Student':'Yes','Credit_Rating':'E'})
test_data
```

```
Age              <=30
Income           High
Student           Yes
Credit_Rating       E
dtype: object
```

```
#Predict
pred = predict(test_data,tree)
pred
```

```
'Yes'
```

B) #Testing2

```
test_data = pd.Series({'Age':'>40','Income':'Medium','Student':'No','Credit_Rating':'E'})
test_data
```

```
Age                 >40
Income           Medium
Student              No
Credit_Rating         E
dtype: object
```

```
#Predict
pred = predict(test_data,tree)
pred
```

```
'No'
```

7. Conclusion

The decision tree is a powerful tool for implementation of clasification models. It is ver accurate .

8. References

https://dataaspirant.com/2017/01/30/how-decision-tree-algorithm-works/
https://www.quora.com/In-what-real-world-applications-is-the-decision-tree-classifier-used