Classification by Decision Tree Induction

Submited by:

Soumadip Biswas (10IT60R12)   Jyotirmay Dewangan (10IT60R04)

School of Information Technology
Indian Institute of Technology Kharagpur

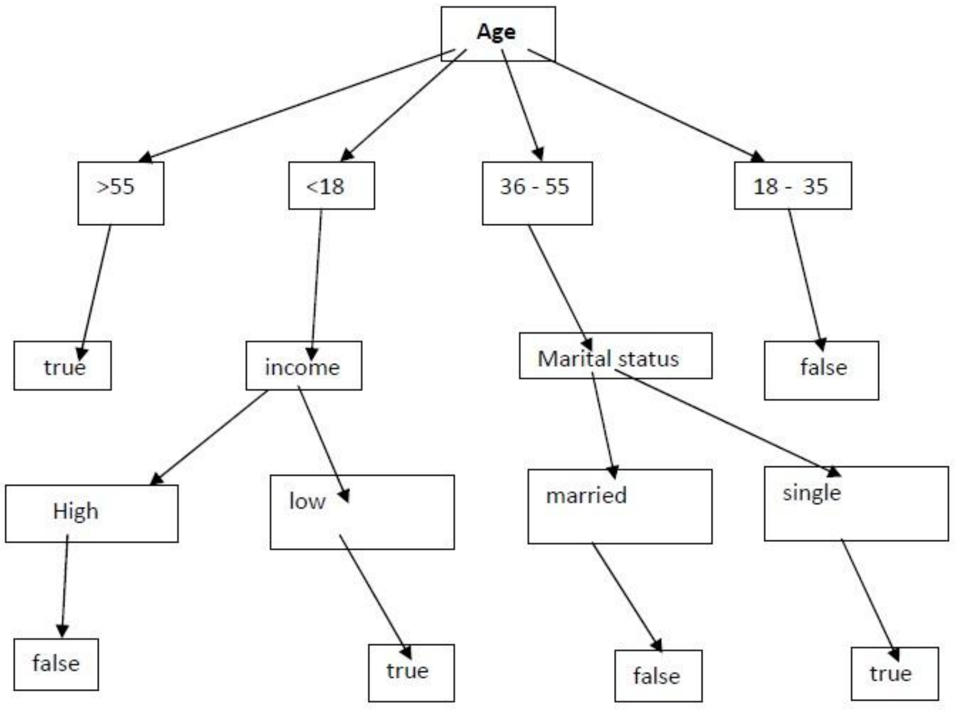
Abstract:

Decision tree algorithm has been successfully used in different systems in capturing knowledge. The main task performed in these systems is using inductive methods to the given values of attributes of an unknown object to determine appropriate classification according to decision tree rules. We examine the decision tree learning algorithm ID3 and implement this algorithm using Matlab.

Introduction:

A decision tree is a tree in which each branch node represents a choice between a number of alternatives, and each leaf node represents a decision. Decision tree are commonly used for gaining information for the purpose of decision -making. Decision tree starts with a root node on which it is for users to take actions. From this node, users split each node recursively according to decision tree learning algorithm. The final result is a decision tree in which each branch represents a possible scenario of decision and its outcome.

A decision tree is a Fow-chart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and leaf nodes represent classes or class distributions. The topmost node in a tree is the root node.



Example of **Decision tree**

Decision tree induction

ID3 algorithm is a decision tree learning algorithm based on information entropy proposed by Quinlan in 1986 and its predecessor is CLS algorithm. Quinlan introduces Shannon's information theory into the decision tree algorithm. The core of ID3 algorithm is selecting attributes from all levels of decision tree nodes. using information gain as attribute selection criteria. each selecting an attribute with the largest information gain to make decision tree nodes. Establishing branches by the different values of the node, building the decision tree nodes and branches recursively according to the instances of various branches, until a certain subset of the instances belonging to the same ategory.

Algorithm:

Generate decision tree. Generate a decision tree from the training tuples of data Partition *D*.

Input:

- Data partition, *D*, which is a set of training tuples and their associated class labels.
- *Attribute list*, the set of candidate attributes.
- *Attribute selection method*, a procedure to determine the splitting criterion that "best" partitions the data tuples into individual classes. This criterion consists of a *splitting attribute* and, possibly, either a *split point* or *splitting subset*.

Output: A decision tree.

Method:

```
1. Create a node N.
2. If tuples in D are all of the same class, C then
a. Return N as a leaf node labelled with the class C.
3. If attribute list is empty then
4. Return N as a leaf node labelled with the majority class in D.
5. Apply Attribute selection method (D, attribute list) to find the "best" splitting
criterion.
6. Label node N with splitting criterion.
7. If splitting attribute is discrete-valued and multi-way splits allowed then
8. Attribute list ß attribute list - splitting attribute.
9. For each outcome j of splitting criterion
a. Let $D_j$ be the set of data tuples in D satisfying outcome j;
b. If $D_j$ is empty then
c. attach a leaf labelled with the majority class in D to node N;
d. Else attach the node returned by Generate decision tree ($D_j$, attribute list) to
node N;
10. End for
11. Return N;
```

Attribute selection measure:

An attribute selection measure is Heuristic for selecting the splittingcriteria. The attribute having best score for measure is chosen as the spliting attribute for given tuple. If splitting attribute is continuous valued or if we are restricted to binary trees then, respectively, either a split point or a splitting subset must also be determined as part of the splitting criterion.

Three popular selection measure

a. Information gain.
b. Gain ratio.
c. Gini index.

a. Information gain

- The information gain measure is used to select the test attribute at each node in the tree.
- The attribute with the highest information gain (or greatest entropy reduction) is chosen as the test attribute for the current node.
- This attribute minimizes the information needed to classify the samples in the resulting partitions.

info(D) is just the average amount of information needed to identify the class label of tuple in D.

$$info(D) = -\sum_{i=0}^{m} p_i \, log_2 p_i$$

Where, Pi is the probability that an arbitrary tuple in D belongs to C_i and is estimated by $|C_{i,\,D}|/\ |D|$

info$_A$(D) is the expected information required to classify a tuple from D base on partitioning by A.

$$info_A(D) = \sum_{j-1}^{v} \frac{|D_j|}{|D|} \, info(D_j)$$

Where $\frac{|D_j|}{|D|}$ acts as a weight of j_{th} partition.

Information gain is defined as the difference between the original information requirement and new requirement that is,

$$Gain(D) = \ info(D) - info_A\ (D)$$

Data Structure and Language to be used

Programming Tool to be used: MATLAB

Data Structure to be used to implement ID3 algorithm is MATLAB’s built-in multidimensional array class which is native to MATLAB and therefore can be handling more efficiently.

Result

Input data :

A	B	C	D	Class
1	3	0	1	0
1	3	0	2	0
2	3	0	1	1
3	2	0	1	1
3	1	1	1	1
3	1	1	2	0
2	1	1	2	1
1	2	0	1	0
1	1	1	1	1
3	2	1	1	1
1	2	1	2	1
2	2	0	2	1
2	3	1	1	1
3	2	0	2	0

Output model:

```
tran_file = input('Enter file name of traning data sheet :','s');
A_1 = xlsread(tran_file);
[row,col]=size(A_1)
for i=1:1:row
        if A_1( i,1)==1
                if A_1( i,3) == 0
                A_1(i,col+1) = 0;
                end
                if A_1( i,3) == 1
                A_1(i,col+1) = 1;
                end
        end
        if A_1( i,1)==2
        A_1(i,col+1) = 1;
        end
        if A_1( i,1)==3
                if A_1( i,4) == 1
                A_1(i,col+1) = 1;
                end
                if A_1( i,4) == 2
                A_1(i,col+1) = 0;
                end
        end
end
tran_file = input('Enter file name of output data sheet : ','s');
xlswrite(tran_file,A_1);
```

a) We applied training data set into output model ,and we found same class for each non conflicting tuple.
b) If we apply ID3 algorithm on well preprocessed data set it works fine.

Conclusion:

For implementing classification of data items we will be using ID3 algorithm as mentioned above and for implementation we are going to use the MATLAB tool.

References:

Jiawei Han and Micheline Kamber, Data Mining: Concepts and Techniques, 2nd edition, pg. 286-310.