Binary Decision Tree

Nail I. Khawaldah, Nathmi A. Ishtayeh
Al Zaraqa University- Computer Science .

ABSTRACT
In this paper, we discuss an important topic in machine learning, which now the popular method to get optimal output, and solve different problems.
Binary Decision Tree (BDT) is a method to classify and sort data to achieve purpose.

1 INTRODUCTION

1.1 Overview on binary decision tree

We can define a binary decision tree in many technical definitions, we may also use binary decision trees for many different types of processes or in different applications. In simple words we can submit the definition of a decision tree as follow: it is an analysis diagram, which can help decision makers to decide which is the best option between different options, by projecting possible outcomes. The decision tree, gives the decision maker an overview of the multiple stages by that will follow each possible decision. Each branch shows the probability of the outcome. Decision trees can be used when making a wide variety of choices.

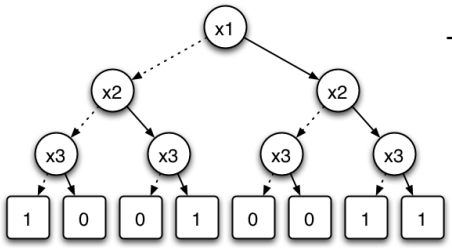
Assume we have boolean variables x making up the input to a function. At the root node we test one of the variables, say, x1 ; : : : ; xn , and we have two subtrees, one for the case where x1= 0 and one where x1= 1. Each of the two subtrees is now testing another variable, each with another two subtrees, and so on.
At the leaves we have either 0 or 1, which is the output of the function on the inputs that constitute the path from the root to the leaf.

A Binary Decision Tree (BDT) is a representation of a Boolean logic function as a decision tree . A BDT has internal nodes and leaves. Each internal nodes is shows as a circle, and represents a decision on the input variable that it is labeled with. Each internal node has exactly two outgoing edges, which correspond to the input variable taking a value of 0 (left edge, drawn as a dotted line) or 1 (right edge, drawn as a soild line) . Each leaf is shown as a rectangle , representing one of the two possible logic value : 0 or 1

A binary decision tree produced admirable success in many fields in different application such that: "**classification, Sorting** ."

1.2 Classification with BTD

A predictive model that uses a set of binary rules applied to calculate a target value can be used for classification (categorical variables) or regression (continuous variables) applications .Rules are developed using software available in many statistics packages. Different algorithms are used to determine the “best” split at a node.



x1	x2	x3	f
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

A decision tree is a classifier expressed as a recursive partition of the instance space. The decision tree consists of nodes that form a rooted tree, meaning it is a directed tree with a node called “root” that has no incoming edges. All other nodes have exactly one incoming edge. A node with outgoing edges is called an internal or test node. All other nodes are called leaves (also known as terminal or decision nodes). In a decision tree, each internal node splits the instance space into two or more sub-spaces according to a certain-discrete function of the input attributes values.

1.3 Advantages and Disadvantages of Decision Trees

Several advantages of the decision tree as a classification tool have been pointed out in the literature:
1.Decision trees are self–explanatory and when compacted they are also easy to follow. In other words if the decision tree has a reasonable number of leaves, it can be grasped by non–professional users. Furthermore decision trees can be converted to a set of rules. Thus, this representation is considered as comprehensible.
2.Decision trees can handle both nominal and numeric input attributes.
3.Decision tree representation is rich enough to represent any discrete–value classifier.
4. Decision trees are capable of handling datasets that may have errors.
5.Decision trees are capable of handling datasets that may have missing values.
6.Decision trees are considered to be a nonparametric method. This means that decision trees have no assumptions about the space distribution and the classifier structure.
On the other hand, decision trees have such disadvantages as:
1. Most of the algorithms (like ID3 and C4.5) require that the target attribute will have only discrete values.
2.As decision trees use the “divide and conquer” method, they tend to perform well if a few highly relevant attributes exist, but less so if many complex interactions are present.
3.The greedy characteristic of decision trees leads to another disadvantage that should be pointed out. This is its over–sensitivity to the training set,to irrelevant attributes and to noise (Quinlan, 1993).

1. TREE GROWING WITH CLASSIFICATION

2.1 Table creating

We first make a list of attributes that we can measure.
These attributes (for now) must be discrete .We then choose a target attribute that we want to predict.
Then create an experience table that lists what we have seen in the past.

2.2 Splitting Criteria

There are many criteria's can be used to apply on decision tree and I will mention some of the popular criteria's as follow:

- **Binary criteria**

The binary criteria are used for creating binary decision trees. These measures are based on division of the input attribute domain into two sub-domains.
Let β(aí; dom1(aí); dom2(aí); S) denote the binary criterion value for attribute aí over sample S when dom1(aí) and dom2(aí) are its corresponding subdomains. The value obtained for the optimal division of the attribute domain into two mutually exclusive and exhaustive sub-domains is used for comparing attributes.

- **Information Gain**

We now return to the problem of trying to determine the best attribute to choose for a particular node in a tree. The following measure calculates a numerical value for a given attribute, A, with respect to a set of examples, S. Note that the values of attribute A will range over a set of possibilities which we call Values(A), and that, for a particular value from that set, v, we write Sv for the set of examples which have value v for attribute A.
The information gain of attribute A, relative to a collection of examples, S, is calculated as:

$$Gain(S,A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

The information gain of an attribute can be seen as the expected reduction in entropy caused by knowing the value of attribute A.

- **Entropy**

Putting together a decision tree is all a matter of choosing which attribute to test at each node in the tree. We shall define a measure called information gain which will be used to decide which attribute to test at each node. Information gain is itself calculated using a measure called entropy, which we first define for the case of a binary decision problem and then define for the general case.
The general entropy measure, which is defined as follows:
Given an arbitrary categorisation, C into categories c1, ..., cn, and a set of examples, S, for which the proportion of examples in ci is pi, then the entropy of S is:

$$Entropy(S) = \sum_{i=1}^{n} -p_i \log_2(p_i)$$

- **Gini index**

Gini index is an impurity-based criterion that measures the divergences between the probability distributions of the target attribute's values. The Gini index has been used in various works such as (Breiman et al., 1984) and (Gelfand et al., 1991) and it is defined as:

$$Gini(y,S) = 1 - \sum_{cj \in dom(y)} \left(\frac{|\sigma_{y=cj} S|}{|S|} \right)^2$$

Consequently the evaluation criterion for selecting the attribute a is defined as:

$$GiniGain(a_i,S) = Gini(y,S) - \sum_{v_{i,j} \in dom(a_i)} \frac{|\sigma_{a_i=v_{i,j}} S|}{|S|} \cdot Gini(y, \sigma_{a_i=v_{i,j}} S)$$

2.3 Stopping Criteria

The growing phase continues until a stopping criterion is triggered. The following conditions are common stopping rules:
1. All instances in the training set belong to a single value of y.
2.The maximum tree depth has been reached.
3. The number of cases in the terminal node is less than the minimum number of cases for parent nodes.
4. If the node were split, the number of cases in one or more child nodes would be less than the minimum number of cases for child nodes.
5.The best splitting criteria is not greater than a certain threshold.

2.4 Decision Trees algorithms

- **ID3**

The ID3 algorithm is considered as a very simple decision tree algorithm. ID3 uses information gain as splitting criteria. The growing stops when all instances belong to a single value of target feature or when best information gain is not greater than zero. ID3 does not apply any pruning procedures nor does it handle numeric attributes or missing values.

- **C4.5**

C4.5 is an evolution of ID3..It uses gain ratio as splitting criteria. The splitting ceases when the number of instances to be split is below a certain threshold. Error–based pruning is performed after the growing phase. C4.5 can handle numeric attributes. It can induce from a training set that incorporates missing values by using corrected gain ratio criteria as presented above.

- **CART**

CART stands for Classification and Regression Trees .It is characterized by the fact that it constructs binary trees, namely each internal node has exactly two outgoing edges. The splits are selected using the twoing criteria and the obtained tree is pruned by cost–complexity Pruning.
When provided, CART can consider misclassification costs in the tree induction. It also enables users to provide prior probability distribution.
An important feature of CART is its ability to generate regression trees.
Regression trees are trees where their leaves predict a real number and not a class. In case of regression, CART looks for splits that minimize the prediction-squared error (the least–squared deviation). The prediction in each leaf is based on the weighted mean for node.

3. EXCAMPLES

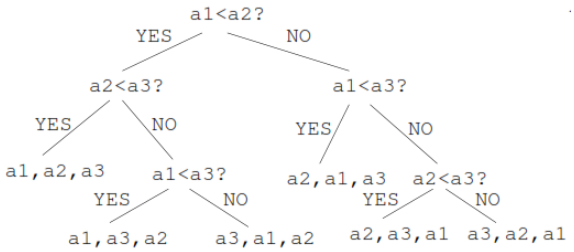
3.1 USING BTD IN SORTING

All algorithms sorts (Mergesort, Quicksort,Heapsort) Make fewer thsn 2^n comparisons.
The only operation that may be used to gain order information about a sequence is comparison of *pairs* of elements.

if A[i] < A[i + 1] , A[i] ↔ A[i + 1]

All sorts seen so far are comparison sorts: insertion sort, selection sort, merge sort, quick sort, heap sort, tree sort.

The next example show how can we use BTD to sort three elements : a1,a2 and a3

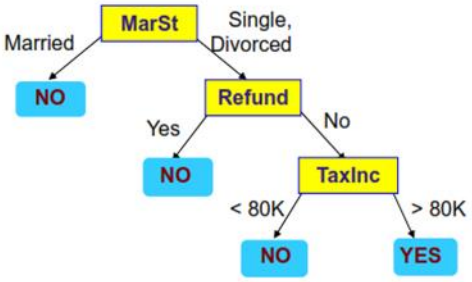


3.2 USING BTD IN CLASSIFICATION

Let see the next example shown by this table:

	categorical	categorical	continuous	class
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

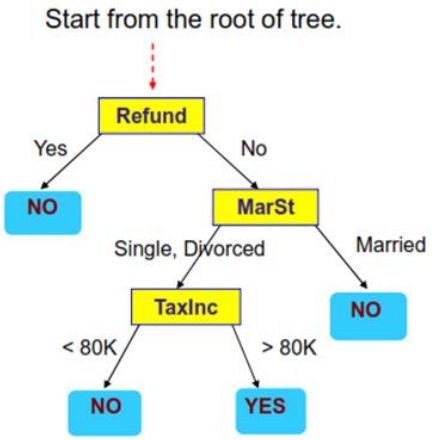
And the tree is :



Now to Apply Model to Test Data here :

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



After following the next path :
Refund N0 **MarSt** Married **No**
We get the result **NO**

5. CONCLUSION

Using Binary Decision Tree in Classifications or sorting date, reduce the time of executed the functions of operation, and get higher. Discrete
Binary Decision Tree is easy to built, and easy to understand,and it can applied to real problems, and able to process both numerical and categorical data.
Binary Decision Tree make no prior assumptions about the data.

REFERENCES

-.Richard Johnsonbaugh , Mathematics , second Edition.
Wikipedia, the free encyclopedia
-.Almuallim H., An Efficient Algorithm for Optimal Pruning of Decision Trees. Artificial Intelligence 83(2): 347-362, 1996.
-Bratko I., and Bohanec M., Trading accuracy for simplicity in decision trees, Machine Learning 15: 223-250, 1994.
-Breiman L., Friedman J., Olshen R., and Stone C.. Classification and Regression Trees. Wadsworth Int. Group, 1984.
-.L ecture Notes onBinary Decision Diagrams 15-122: Principles of Imperative Computation Frank Pfenning 1 Introduction Lecture 19 October 28, 2010.
- Notebook for EdmundM.Clarke,Jr. ComputerScienceDepartment CarnegieMellonUniversity Pittsburgh.
- Nils J.Nilson, Introduction to Machine Learning,second Edition,1996.
-Jonathan Trumbull Foote, Decision-Tree Probability Modeling for HMM Speech Recognition, partial fulllment of the requirements for the Degree of Doctor of Philosophy in the Division of Engineering at Brown University May 1994.
- Tony R. Martinez and Douglas M. Campbell, A Self-organizing Binary Decision Tree for Incrementally Defined Rule-Based Systems,IEEE VOL. 21, NO. 5, SEPTEMBERIOCTOBER 1991