Implementation of Decision Trees Using Iterative Dichotomizer (ID3) Algorithm

Sudarvizhi Kulothungan
Accenture Services Private Ltd., India
{s.kulothungan@accenture.com}

ABSTRACT
Data mining is the process of finding the previously unknown and potentially interesting patterns and relations in large databases. Data Models developed by mining large databases are used to alleviate the exploration problems. Effective and accurate decision-making can be enabled by data mining. For example, when the data describes customers, the rows of the model set are often called Customer Signatures. For decision-making, decision tree methods are good choices. Decision trees reveal so much about the data to which they are applied and they are used to assign each record to one of few broad categories which can be easily understood, explained and translated into SQL or a natural language.

Decision trees can be applied in many different situations.
- To explore a large dataset to pick out useful variables.
- To predict future states of important variables in an industrial process.
- To form directed clusters of Customers for a recommendation system.

ID3 (Iterative Dichotomizer 3) algorithm is the best algorithm that uses an entropy based measure known as Information gain that will best separate the samples into individual classes so that in future, if any new values entered into the database, the classes will be assigned to the data which helps the customer to maintain the customer relationship management and it provides the accuracy of the classification of the data.

It leads us to draw a decision tree where Iterative Dichotomizer algorithm is used to find out which one of the attribute from the application database can made as Root node so that after classification, the number of records in the database should accurately match total number of records in the entire database.

KEYWORDS
Iterative Dichotomizer 3 (ID3)

INTRODUCTION
Data mining is the process of finding previously unknown and potentially interesting patterns and relations in large databases [1]. It is the process of exploration and analysis by automatic or semiautomatic means of large quantities of data in order to discover meaningful patterns and rules. The extracted information can:
1. Be organized by a data analyst into a prediction or classification model;
2. Provide a summary of the database(s) being mined; and
3. Be used to refine an existing model.

WHAT CAN DATA MINING DO?
The six activities are Classification, Estimation, Prediction, and Affinity grouping or association rules, clustering and Description and Visualization [1].

Mining databases to develop a model can be performed in either a top-down or a bottom-up fashion. They are directed data mining and undirected data mining [2].

MODEL DEVELOPMENT AND DATA MINING
Making efficient and accurate decisions directly from the data stored in database is difficult for two reasons.
1. These databases are frequently very large.
2. The database is usually described in terms of low-level concepts and relations that are cognitively distant from the concepts used by the decision-maker.

TREE CONSTRUCTION PRINCIPLE

SPLITTING ATTRIBUTE
With every node of the decision tree, there is an associated attribute whose values determine the partitioning of the data when the node is expanded [3].

SPLITTING CRITERION
The qualifying condition on the splitting attribute for data
Splitting at a node is called the splitting criterion at that Node. For a numeric attribute, the criterion can be an equation or an inequality. For a categorical attribute, it is a membership condition on a subset of values [3].

The construction of the decision trees involves following three main phases.
- Construction phase.
- Pruning phase.
- Processing the pruned tree.

BEST SPLIT
Best split is defined as one that does the best job of separating the records into groups where a single class predominates [1]. There are two different methods of determining the goodness of the split.
1. One index is based on the information theory (i.e. Information gain based on entropy)
2. Other one is derived from economics as measure of diversity. This is called Gini index.
**ENTROPY**

It provides an informatics theoretic approach to measure the goodness of a split.

Assume that there are ‘n’ equally probable possible messages [2].

Probability P of each is 1 / n.

Information conveyed by a message is \( \log_2 (p) = \log_2 (\frac{1}{n}) \)

**Definition**

If probability distribution \( P = (p_1, p_2, …, p_n) \) is given, then the information conveyed by this distribution is called entropy [2].

\[
\text{Entropy (p)} = - (p_1 \log (p_1) + p_2 \log (p_2) + \cdots + p_n \log (p_n)) \quad [1]
\]

In context of decision trees, if the outcome of a node is to classify the records into two classes, \( c_1, c_2 \), the outcome can be viewed as a message that is being generated and entropy gives the Measure of information for the message to be \( c_1 \) or \( c_2 \) [1].

\[
\text{Gain (x, t)} = \text{Info (t)} - \text{Info(x, t)}
\]

\[
\text{Info (t)} = \text{Entropy (p)}
\]

**GAIN RATIO**

\[
\text{Gain ratio (x, t)} = \frac{\text{gain(x, t)}}{\text{Info(x, t)}}
\]

**ID3 ALGORITHM (ITERATIVE DICHOTOMIZER 3)**

A well known greedy tree growing algorithm for generating decision tree is Quinlan’s ID3 algorithm [6].

**ALGORITHM**

Generate decision tree. Generate a decision tree from the given training data.

**INPUT:** The training samples, samples, represented by discrete – valued attributes, the set of candidate attributes, attribute_list [5].

**OUTPUT:** A decision tree.

**METHOD:**

Step 1: Create a node N;

Step 2: if samples are all of the same class, C then

Step 3: return N as a leaf node labeled with the class C;

Step 4: if attribute_list is empty then

Step 5: return N as a leaf node labeled with the most common class in samples;

Step 6: select test_attribute, the attribute among attribute_list with the highest information Gain;

Step 7: label node N with test_attribute;

Step 8: for each known value of a I of test_attribute

Step 9: Grow a branch from node N for the condition test_attribute=a_i

Step 10: Let s_i is the set of samples in samples for which

Step 11: if s_i is empty then

Step 12: attach a leaf labeled with the most common class in samples; [2].

Step 13: else attach the node returned by generate_decision_tree (s_i, attribute list- test_attribute);

**IMPLEMENTATION OF ID3 ALGORITHM**

ID3 starts with all the training examples at the root node of the tree [3].

An attribute is selected to partition these examples. For each values of the attribute, a branch is created and the corresponding subsets of examples that have the attribute value specified by the branch are moved to the newly created child node.

The algorithm is applied recursively to each child until either all examples at a node are of one class or all the examples at that node have the same values for all the attributes [4].

Every leaf in the decision tree represents a Classification rule. An attribute is selected to partition these examples. For selecting the attribute to fit at the root node, the information gain measure is used to select the test_attribute at each node in the tree. Such a measure is referred to as an attribute selection measure or a measure of the goodness of split.

The attribute with the highest information gain is chosen with test attribute for the current node [3].

A decision tree is constructed by looking for regularities in data.

**EXAMPLE 1:**

**ATTRIBUTE POSSIBLE VALUES**

<table>
<thead>
<tr>
<th>Age</th>
<th>&lt;25, &gt;=25 &amp;&lt;=50, &gt; 50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile</td>
<td>Salaried, Retired, Self Employed, Student</td>
</tr>
<tr>
<td>Type of Plan</td>
<td>Local, Local + STD, Local + STD + ISD</td>
</tr>
<tr>
<td>Internet</td>
<td>Yes, No</td>
</tr>
</tbody>
</table>

In this case, Name of the table is CUSTOMER, Goal for the final decision is LOYAL, Decision factor columns are AGE, PROFILE, TYPEOFPLAN and INTERNET.
Implementation of Decision Trees Using ID3 Algorithm

Table 1: Training data set

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Profile</th>
<th>Type of Plan</th>
<th>Internet</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Angeline</td>
<td>25</td>
<td>salaried</td>
<td>LOCAL</td>
<td>yes</td>
<td>loyal</td>
</tr>
<tr>
<td>Christy</td>
<td>30</td>
<td>salaried</td>
<td>LOCAL+STD</td>
<td>yes</td>
<td>loyal</td>
</tr>
<tr>
<td>Deepak</td>
<td>45</td>
<td>student</td>
<td>LOCAL+STD+ISD</td>
<td>no</td>
<td>none</td>
</tr>
<tr>
<td>Eswar</td>
<td>60</td>
<td>self-employed</td>
<td>LOCAL</td>
<td>no</td>
<td>loyal</td>
</tr>
<tr>
<td>Sam</td>
<td>70</td>
<td>retired</td>
<td>LOCAL+STD</td>
<td>yes</td>
<td>none</td>
</tr>
<tr>
<td>John</td>
<td>55</td>
<td>salaried</td>
<td>LOCAL+STD+ISD</td>
<td>yes</td>
<td>loyal</td>
</tr>
<tr>
<td>Fintie</td>
<td>18</td>
<td>self-employed</td>
<td>LOCAL</td>
<td>no</td>
<td>loyal</td>
</tr>
<tr>
<td>Nonika</td>
<td>15</td>
<td>student</td>
<td>LOCAL</td>
<td>no</td>
<td>loyal</td>
</tr>
<tr>
<td>Shalini</td>
<td>22</td>
<td>student</td>
<td>LOCAL+STD</td>
<td>yes</td>
<td>none</td>
</tr>
<tr>
<td>Paul</td>
<td>50</td>
<td>retired</td>
<td>LOCAL+STD</td>
<td>yes</td>
<td>loyal</td>
</tr>
</tbody>
</table>

You can analyze this data to construct a decision tree (or algorithm) for predicting whether loyal customer or not [7].

To run an analysis, the name of the database table should be specified to analyze, the name of the Result column (the table column that represents the final decision desired), and the names of one or more columns that are factors in making the decision [4].

**WHAT WE NEED TO DO WITH ID3**

1. Create root Node, containing the whole learning set as its subset, then compute

   Entropy (rootNode.subset) = -(6/10) log2 (6/10) - (4/10)log2(4/10)

   Here in this case,
   6/10 = Number of records with value “loyal” in the Result column
   4/10 = Number of records with value “None” in the Result column

2. Compute information gain for each attribute:
   Gain(S, Age) = 0.229
   Gain(S, Profile) = 0.281
   Gain(S, Type of Plan) = 0.038
   Gain(S, Internet) = 0.042

3. Select attribute with the maximum information gain, which is ‘profile’ for splitting [7].

**Given this input, id3 algorithm will generate a decision tree showing how to find a loyal customer**

![Decision Tree Diagram]

**RULE GENERATION**

Once a decision tree has been constructed, next step is to convert it into an equivalent set of rules.

A decision rule is nothing but a set of if-then statements. It is searched sequentially for an appropriate if - then statement to be used as a rule [6] [8].

**DECISION RULES**

Here in this problem, the following are set of decision rules

<table>
<thead>
<tr>
<th>Rule</th>
<th>IF</th>
<th>THEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Customer’s Profile is salaried</td>
<td>Considered to be Loyal</td>
</tr>
<tr>
<td>2</td>
<td>Customer’s Profile is Retired</td>
<td>Considered to be Loyal</td>
</tr>
<tr>
<td>3</td>
<td>Customer’s Profile is Student</td>
<td>Considered to be Loyal</td>
</tr>
<tr>
<td>4</td>
<td>Customer’s Profile is Self Employed</td>
<td>Considered to be Loyal</td>
</tr>
</tbody>
</table>

A decision tree is important not because it summarizes what we know, i.e. the training set, but because we hope it will classify correctly new cases. When building classification models one should have both training data to build the model and test data to verify how well it actually well it actually works [6].

**CONCLUSION**

Caring about your customers and making it easy for them to do business with you is the most important factor in driving loyalty in this industry. Quality of products and services, as well as brand equity, influence customer relations greatly. Using this Decision trees, we can able to find solution for this following question, “Are telecom providers actually in tune with that which makes customers happy and what it takes to make them stay?” [8].

**FUTURE SCOPE**

ID3 algorithm can be generalized by overcoming the disadvantages so that it does not necessarily branch on each value of the chosen attribute. The improved algorithm can branch on arbitrary individual values of an attribute and the rest of the values in a single default branch. Algorithm can be designed to overcome the problems with the information entropy attribute selection measure itself. C4.5 is a software extension of the basic ID3 algorithm designed by Quinlan to address the above mentioned problem stated by ID3

**DISCLAIMER**

The author(s) of this report gratefully acknowledge Accenture for their encouragement in the development of this research. The information contained in this document represents the views of the author(s) and the company is not liable to any party for any direct/indirect consequential damages.
REFERENCES


