Implementation of Improved ID3 Algorithm Based on Association Function

Prof. Mr. A.M Bhadgale, Ms. Sharvari Natu, Ms. Sharvari G. Deshpande, Mr. Anirudha J. Nilegaonkar.

Department of Computer Engineering, PVG’s COET, Pune, India,

amb_comp@pvgcoet.ac.in, sharvari.natu@gmail.com, sharvarigdeshpande@gmail.com, aninile9495@gmail.com.

Abstract

Decision tree learning is a discipline to create a predictive model to map the different items in the set and respective target values and associate them in a way that is true to every element. This concept is used in data mining, statistics and machine learning due to its simple and effectiveness. There are a variety of algorithms which employ decision tree learning. ID3 is one of the simplest and widely used algorithms. ID3 gives importance to attributes having multiple values while selecting a particular node. These shortcomings affect the accuracy of the tree which is generated. Thus in this paper focus is on improvement over ID3 algorithm using Association Function.

Key Words and Phrases: ID3 algorithm, Improved ID3 algorithm, Entropy, Information gain.

1 Introduction

With the development of technology the amount of data being produced daily is increasing. As a result the ability of people to collect and produce data is enhanced. Thus data mining came into existence, which helped in discovering useful knowledge. In the data present there is lots of implicit information. One of the solution to this information gaining is decision tree induction. Decision tree is an
abstraction to visually and explicitly represent the dependencies between attributes of elements in a set and their sorting on the tree determines a set of rules which explain and summarize the relations of all items throughout the set.

ID3 constructs decision tree using top-down Greedy strategy. This algorithm provides the possibility to create a decision tree based on a fixed set of examples, in order to classify future samples. The tree outputted by this algorithm represents a simple abstraction to explain all the elements of the set and offers in a clever and intuitive way the overall dependencies among them to better understand the system and prepare a decision.

The accuracy of ID3 algorithm can be improved using proposed improved version of ID3 which uses association function along with information gain to decide splitting attribute. This approach also overcomes the shortcoming of choosing multi-valued attributes of ID3 algorithm.

2 ID3 Algorithm

The ID3 algorithm is a recursive procedure, where in at each step there is an evaluation of a subset and creation of decision node, based on a metric called Information Gain, until the subset in evaluation is specified by the same combination of attributes and its values. ID3 algorithm creates a tree by using top-down approach by using the given set of values by checking each attribute at every node. Information gain is used as metric to generate tree to select the best attribute at each step.

The ID3 decision makes use of two concepts when creating a tree from top down:

1. Entropy
2. Information Gain

Using these two metrics, the nodes to be created and the attributes to split on can be determined. Entropy is a measure in information theory to measure the impurity of an arbitrarily collection of item.

For a given set $S$, $p_i$ is the probability of $S$ belonging to class $i$, we have

$$\text{Entropy } H(P) = \sum_{i=1}^{n} p_i \log_2 p_i$$

Information Gain is the expected reduction in entropy by splitting the collection $S$ by a given outcome for attribute $A$, with an associated subset $S_v$.

Information Gain uses the entropy in order to determine what attribute is best used to create a split with. By
by using that attribute. So, the column with the higher Gain will be used as the node of the decision tree.

\[
\text{Information Gain} = I(S_1, S_2, S_3, \ldots, S_m) - E(A)
\]

Algorithm for generating a decision tree according to a given data sets.\[6\]

Input: training samples, each attribute taking discrete value, a candidate attribute set available for induction is attribute_list.

**A. Output: a decision tree.**

- Create a node M.
- If the node of all samples are falling are in the same category X, then return M as a leaf node and mark with category X. The root node corresponds to all the training samples.
- If attribute_list is empty, then return M as a leaf node and mark the node as a type whose samples contain the largest number of categories.
- select a test_attribute with the largest information gain from attribute_list, and mark node M with test_attribute;
- For each given value \(a_i\) of test_attribute, the sample set contained in node N is portioned.
- According to the condition of test_attribute = \(a_i\); a corresponding branch is generated from the node N to indicate the test conditions.
- Set \(i_s\) is the obtained sample set under the condition of test_attribute = \(a_i\). If \(i_s\) is empty, then mark the corresponding leaf node with category of including the most number of sample types. Otherwise, It will be marked with a return value:

\[(\text{Generate decision tree } s_i, \text{attribute list} - \text{test attribute})\]

**B. Properties**

ID3 does not guarantee an optimal solution. It selects the best attribute from the given set. It then splits the dataset in each iteration. Thus it uses a greedy approach.

ID3 can overfit to the training data. As a solution to this, instead of larger trees smaller trees should be preferred. Though this algorithm specifies a solution, it does not always guarantee an optimal solution.
continuous, it is harder to split the dataset into one specific point. Thus, searching for the best value to split becomes a time consuming job.

C. Experimental Results

A simple Mutual Fund application dataset is shown in below Table. The category attribute of the sample set is "Risk", which will decide in which mutual fund the customer can invest.

TABLE I. TRAINING SAMPLE

<table>
<thead>
<tr>
<th>No.</th>
<th>Age</th>
<th>Tenure</th>
<th>Status</th>
<th>Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Young</td>
<td>Long Term</td>
<td>Unmarried</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>Middle</td>
<td>Short Term</td>
<td>Married</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>Old</td>
<td>Long Term</td>
<td>Married</td>
<td>Yes</td>
</tr>
<tr>
<td>4</td>
<td>Old</td>
<td>Short Term</td>
<td>Married</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>Middle</td>
<td>Long Term</td>
<td>Unmarried</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>Young</td>
<td>Short Term</td>
<td>Married</td>
<td>Yes</td>
</tr>
<tr>
<td>7</td>
<td>Young</td>
<td>Short Term</td>
<td>Unmarried</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Here the “yes“ and “No’s” will give the names of respective Mutual Funds as applicable to the customer. Both algorithms that is, improved ID3 algorithm and ID3 algorithm are applied on this dataset to construct decision trees and comparison is made. Figures below show the generated decision trees using the ID3 algorithm and the improved ID3 algorithm, respectively.

Fig 1. This is the Decision Tree produced when ID3 Algorithm is applied.
D. Shortcomings of ID3 Algorithm

There exists one problem with this approach: ID3 selects the attribute having more number of values, which are not necessarily the best attribute. Data may be over-fitted or over-classified, if a small sample is tested. At a time only one attribute is used for the testing purpose. As specified above, continuous data is difficult to analyze as many trees need to be generated to find the perfect place to split the data. This makes the algorithm computationally expensive.

3 Improved ID3

To overcome the drawbacks seen so far improved version of ID3 comes into picture. The uneven nature of basic ID3 of selecting multi-valued attributes as a decision node is eliminated by improved ID3.

In Improved version we use the same gain initially calculated in basic ID3 which get changed every time when the dataset get modified as tree grows. In this improved version we first calculate Association function (AF) for each attribute and these obtained values are further used to calculate the normalized gain for each attributes. Now this normalized gain is combined with initial gain of attributes to get new gain for each attribute which is used as standard for making decisions. Normalized gain function, Association function etc are methods used for computation of attribute importance. Association function not only very well handles the inadequacy of basic ID3 but also clearly represent relation among elements and attributes.

Figure below shows decision tree generated by using association function:
Suppose attribute belongs to A dataset, and let B be any category attribute of dataset A. Then the relation degree function between Z and B can be given as:

\[
AF(Z) = \frac{\sum_{i=1}^{n} |x_3 - x_2|}{N}
\]

By closely observing both the trees we found that basic ID3 select Age as root node for making decisions which seems to be worthless and has lower importance, because by selecting age as root for customer loan dataset does not give that much information, but on the other hand decision tree generated by Improved ID3 leads us to efficient and fruitful decision making some sense for given dataset. Where, \(x_{i1}\) is the number of negative attributes \(x_{i2}\) is the number of positive attributes, \(n\) is number of types of attributes that Z contains. Now, after calculating \(AF\) we calculate Normalized gain for each attribute as follows:

\[
V(k) = \frac{AF(k)}{AF(1)+AF(2)+...+AF(m)}
\]

Where \(0 < k <= m\).

Further we calculate a new gain for each attribute that allows us to select a decision node.

\[
Gain'(A) = (I(s1,s2,...,sm) - E(A)) \times V(A)
\]

Thus the drawbacks of ID3 can be easily overcome by using the above mention equations.

That means if we consider “status” as root then one can get clear cut understanding about whether there is risk associated or not. So improved version of ID3 is better than the basic version.

4 Conclusion

Accuracy of ID3 algorithm can be improved using association function and more optimal decision trees can be generated using proposed improved ID3 algorithm. In Improved Id3 more reasonable and effective rules are generated.

Time complexity is more in improved ID3, but it can be neglected because now faster and faster computers are present.
5 References