ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 1, July 2012

A Modified Approach to Construct Decision Tree in Data Mining Classification

Rahul A. Patil, Prashant G. Ahire, Pramod. D. Patil, Avinash L. Golande

Abstract - ***During the late 1970s and early 1980s, J. Ross Quinlan, a researcher in machine learning, developed a decision tree algorithm known as ID3 (Iterative Dichotomiser). This work expanded on earlier work on concept learning systems, described by E. B. Hunt, J. Marin, and P. T. Stone. ID3 adopt a greedy (i.e., no backtracking) approach in which decision trees are constructed in a top-down recursive divide-and-conquer manner. ID3 uses information gain as its attribute selection measure. This measure is based on pioneering work by Claude Shannon on information theory, which studied the value or "information content" of messages. The principle of selecting attribute A as test attribute for ID3 is to make E (A) of attribute A, the smallest. Study suggest that there exists a problem with this method, this means that it often biased to select attributes with more taken values however, which are not necessarily the best attributes. To overcome the shortcoming stated above, attribute related method is firstly applied to computer the importance of each attribute. Then, information gain is combined with attribute importance, and it is used as a new standard of attribute selection to construct decision tree. The experiment results show that the proposed algorithm can overcome ID3's shortcoming effectively and get more reasonable and effective rules.***

Index Terms: **Association, Decision Tree, ID3, Function (AF).**

I. INTRODUCTION

ID3 is a simple decision tree learning algorithm developed by Ross Quinlan (1983). The basic idea of ID3 algorithm is to construct the decision tree by employing a top-down, greedy search through the given sets to test each attribute at every tree node. In order to select the attribute that is most useful for classifying a given sets, we introduce a metric---information gain.

A. Decision Tree

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.

B. Decision Tree Learning Algorithms

Decision tree learning is a method for approximating discrete-valued target functions, in which the learned function is represented by a decision tree. Decision tree learning is one of the most widely used and practical methods for inductive inference'. (Tom M. Mitchell, 1997, p52) Decision tree learning algorithm has been successfully used in expert 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. Decision trees classify instances by traverse from root node to leaf node. We start from root node of decision tree, testing the attribute specified by this node, then moving down the tree branch according to the attribute value in the given set. This process is the repeated at the sub-tree level.

C. Decision Tree Learning Algorithm Suited For:

1. Instance is represented as attribute-value pairs. For example, attribute 'Temperature' and its value 'hot', 'mild', 'cool'. We are also concerning to extend attribute-value to continuous-valued data (numeric attribute value) in our project.

2. The target function has discrete output values. It can easily deal with instance which is assigned to a boolean decision, such as 'true' and 'false', 'p(positive)' and 'n(negative)'. Although it is possible to extend target to real-valued outputs, we will cover the issue in the later part of this report.

3. The training data may contain errors. This can be dealt with pruning techniques that we will not cover here. The 3 widely used decision tree learning algorithms are: ID3, ASSISTANT and C4.5. We will cover ID3 in this paper.

D. Entropy --- Measuring Homogeneity Of Learning Set

In order to define information gain precisely, we need to discuss entropy first. First, let's assume, without loss of generality, that the resulting decision tree classifies instances into two categories, we'll call them P (positive) and N (negative). Given a set S, containing these positive and negative targets, the entropy of S related to this Boolean classification is:

Entropy(S) = - P(positive)log2P(positive) - P(negative)log2P(negative)

P(positive): proportion of positive examples in S P(negative): proportion of negative examples in S For example, if S is 0.5+, 0.5-) then Entropy(S) is 1, if S is (0.67+, 0.33-) then entropy(S) is 0.92, if P is (1+, 0-) then Entropy(S) is 0. Note

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 1, July 2012

that the more uniform is the probability distribution, the later is its information.

E .Information Gain: Measuring the Expected Reduction in Entropy

As we mentioned before, to minimize the decision tree depth, when we traverse the tree path, we need to select the optimal attribute for splitting the tree node, which we can easily imply that the attribute with the most entropy reduction is the best choice. We define information gain as the expected reduction of entropy related to specified attribute when splitting a decision tree node.

The information gain, Gain(S,A) of an attribute A,

$$\text{Gain}(S,A) = \text{Entropy}(S) - \sum_{i=1}^{n} (|Sv|/|S|) * \text{Entropy}(Sv)$$

F. ID3 ALGORITHM:

1. Create a node N
2. If (All samples are in same class)
   Return node as leaf with classname;
3. If (attribute list is empty)
   Return node as leaf node labeled with most common class;
4. Select list attribute i.e. attributes having highest information gain
5. Label node N with test attribute
6. for each known value of a; of test attribute, grow branches from node N for the condition test attribute =a;
7. Let Si be set of samples for which test attribute = ai;
8. If (Si is empty) then attach the leaf labeled with most common class in sample
9. Else attach the node returned by generate_decision_tree(Si,attribute_list_test_attribute).

Example:

Table 1: Training sample

Outlook	Temperature	Humidity	Wind	Decision
Sunny	Hot	High	False	N
Sunny	Hot	High	True	N
Overcast	Hot	High	False	P
Rain	Mild	High	False	P
Rain	Cool	Normal	False	P
Rain	Cool	Normal	True	N
Overcast	Cool	Normal	True	P
Sunny	Mild	High	False	N
Sunny	Cool	Normal	False	P
Rain	Mild	Normal	False	P
Sunny	Mild	Normal	True	P
Overcast	Mild	High	True	P
Overcast	Hot	Normal	False	P
Rain	Mild	high	True	N

HOT => >40 Mild => 21-40 Cool => <=20

Expected Information for Database Classification

$$I(S_1, S_2,....S_m) = -\sum_{i=1}^{m} pi \log_2(p_i)$$

For example,

$I(S_1, S_2) =$ I(5,9)

= (5/14) $\log_2$ (5/14) - (9/14) $\log_2$ (9/14)

= 0.9403 bits/msg

Gain On Attribute OUTLOOK

a) for outlook = "SUNNY"

$S_{11} = 3$ $S_{21} = 2$

= - 3/5 $\log_2$(3/5) – 2/5 $\log_2$(2/5)

= 0.971 bits/msg

b) for outlook =" Overcast"

$S_{12} = 0$ $S_{22} = 4$

= - 0/4 $\log_2$(0/4) – 4/4 $\log_2$(4/4)

=0 bits/msg

c) for outlook= "Rain"

$S_{13} = 2$ $S_{23} = 3$

= - 2/5 $\log_2$(2/5) – 3/5 $\log_2$ (3/5)

= 0.9710 bits/msg

Entropy:

$$E(A) = \sum_{j}^{v} S_{ij} + ... + S_{mj} / S * I(S_{ij} + ... + S_{mj})$$

$$\text{E (outlook)} = \frac{5}{14} I(S_{11}, S_{21}) + \frac{4}{14} I(S_{12}, S_{22}) + \frac{5}{14} I(S_{13}, S_{23})$$

$$= \frac{5}{14} \times (0.971) + \frac{4}{14} \times (0) + \frac{5}{14} \times (0.971)$$

= 0.694 bits/msg

Information Gain :

Gain (A) = I ($S_1, S_2,....S_m$) - E (A)

Information Gain of outlook attribute

Gain (outlook) = I (S_1, S_2) – E (Outlook)

= 0.9403 – 0.694

= 0.2463 bits/msg

Gain on attribute "TEMP"

a) for temp> 40

$S_{11} = 2$ $S_{22} = 2$

$$= -\frac{2}{4}\log_2(\frac{2}{4}) - \frac{2}{4}\log_2(\frac{2}{4})$$

=1 bits/msg

b) For temp=> 21-40

$S_{12} = 2$ $S_{22} = 4$

$$= -\frac{2}{6}\log_2(\frac{2}{6}) - \frac{4}{6}\log_2(\frac{4}{6})$$

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 1, July 2012

= 0.9183 bits/msg

c) For temp <= 20

$S_{13} = 1 \quad S_{23} = 3$

$= -\frac{1}{4}\log_2(\frac{1}{4}) - \frac{3}{4}\log_2(\frac{3}{4})$

= 0.8112 bits/msg

Entropy

E (Temperature)= $\frac{4}{14}\times(I) + \frac{6}{14}\times(0.9183) + \frac{4}{14}\times(0.8112)$

= 0.9111 bits/msg

Information Gain for temperature

Gain (Temperature) = 0.9403 – 0.91111

= 0.0029 bits/msg

Information Gain On attributes "Humidity"

For humidity = 'High'

$S_{11} = 4 \quad S_{21} = 3$

$= -\frac{4}{7}\log_2(\frac{4}{7}) - \frac{3}{7}\log_2(\frac{3}{7})$

= 0.9852 bits/msg

For humidity = "Normal"

$S_{12} = 1 \quad S_{22} = 6$

$= -\frac{1}{7}\log_2(\frac{1}{7}) - \frac{6}{7}\log_2(\frac{6}{7})$

= 0.5917 bits/msg

Entropy

E(Humidity) = $\frac{7}{14}(0.9852) + \frac{7}{14}(0.5917)$

= 0.7884

Information Gain for humidity

Gain (humidity) = 0.9403 – 0.7884

= 0.1518 bits/msg

Gain on attribute "WIND"

For wind = "False"

$S_{11} = 2 \quad S_{21} = 6$

$= -\frac{2}{8}\log_2(\frac{2}{8}) - \frac{6}{8}\log_2(\frac{6}{8})$

= 0.8112 bits/msg

For wind= "True"

$S_{12} = 3 \quad S_{22} = 3$

$= -\frac{3}{6}\log_2(\frac{3}{6}) - \frac{3}{6}\log_2(\frac{3}{6})$

= 1 bits/msg

Entropy

E(Wind) = $\frac{8}{14}(0.8112) + \frac{6}{14}(1)$

= 0.8921 bits/msg

Information Gain For "Wind"

=0.9403 – 0.8921

= 0.0482 bits/msg

Table 2: Comparison of Information gain

Attributes	Gain
Outlook	0.2463
Temperature	0.029
Humidity	0.1518
wind	0.0482

For second iteration,

Temp 0.571

Humidity 0.971

Wind 0.020

For third,

Temp 0.020

Wind 0.971

Wind – false - P

Wind – true – N

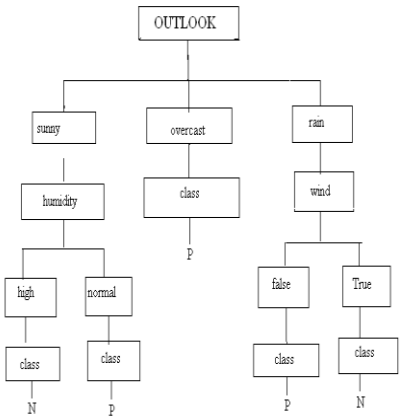

Fig 1: Resultant Decision Tree Using ID3 Algorithm

The shortcoming of ID3 algorithm

The principle of selecting attribute A as test attribute for ID3 is to make E (A) of attribute A, the smallest. Study suggest that there exists a problem with this method, this means that it often biased to select attributes with more taken values, however, which are not necessarily the best attributes. In other words, it is not so important in real situation for those attributes selected by ID3 algorithm to be judged firstly according to make value of entropy minimal. Besides, ID3 algorithm selects attributes in terms of information entropy which is computed based on probabilities, while probability method is only suitable for solving stochastic problems. Aiming at these shortcomings for ID3 algorithm, some improvements on ID3 algorithm are made and a improved decision tree algorithm is presented.

II. THE PROPOSED IMPROVED ID3 ALGORITHM

To overcome the shortcoming stated above, attribute related method is firstly applied to computer the importance of each attribute. Then, information gain is combined with

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 1, July 2012

attribute importance, and it is used as a new standard of attribute selection to construct decision tree. The conventional methods for computing attribute importance are sensitivity analysis (SA), information entropy based joint information entropy method (MI), Separation Method (SCM), Correlation Function Method(AF), etc. SA needs not only to compute derivatives of output respect to input or weights of neural network, but also to train the neural network. This will increase computational complexity. MI needs to compute density function and it is not suitable for continuous numerical values. SCM computes separation property of input-output and the correlation property of input and output attributes and is suitable for both continuous and discrete numerical values, but computation is complex. AF not only can well overcome the ID3's deficiency of tending to take value with more attributes, but also can represent the relations between all elements and their attributes. Therefore, the obtained relation degree value of attribute can reflect its importance. AF algorithm: Suppose A is an attribute of data set D, and C is the category attribute of D. the relation degree function between A and C can be expressed as follows:

$$AF(A) = \frac{\sum_{i=1}^{n} |x_{i1} - x_{i2}|}{n}$$

Where x_{ij} (j = 1, 2 represents two kinds of cases) indicates that attribute A of D takes the i-th value and category attribute C takes the sample number of the j-th value, n is the number of values attribute A takes. Then, the normalization of relation degree function value is followed. Suppose that there are m attributes and each attribute relation degree function value are

AF(1), AF(2),…. AF(m), respectively. Thus, there is

$$V(k) = \frac{AF(k)}{AF(1) + AF(2) + \cdots + AF(m)}$$

Which $0 < k \leq m$, Then, equation (3) can be modified as

$$Gain'(A) = (I(s_1, s_2, \cdots, s_m) - E(A)) \times V(A)$$

Gain'(A) can be used as a new criterion for attribute selection to construct decision tree according to the procedures of ID3 algorithm. Namely, decision tree can be constructed by selecting the attribute with the largest Gain'(A) value as test attribute. By this way, the shortcomings of using ID3 can be overcome. It construct the decision tree, this tree structure will be able to effectively overcome the inherent drawbacks of ID3 algorithm.

III. EXPERIMENTAL RESULTS

After testing the original ID3 algorithm and proposed improved ID3 algorithm on following dataset we have got the following results. In Proposed improved ID3 algorithm we are getting the more no. Of nodes and more no. of rules means the decision tree we are getting from proposed improved ID3 algorithm is more efficient than original ID3 algorithm.

Table 3: Results: Node and Rule count

Dataset	Record Num n	Node(count)		Rules(count)	
		ID3	Improved ID3	ID3	Improved ID3
Db1	170	4	8	8	13
MyDataSet	768	48	73	95	120
Sales	10	3	4	10	11
MyVote	233	15	19	30	38

Table 4: Results: Time Complexity

Dataset	Record Num n	Time(MS)	
		ID3	Improved ID3
Db1	170	15	81
MyDataSet	768	73	1195
Sales	10	0	20
MyVote	233	43	639

IV. CONCLUSION

Proposed Improved ID3 Algorithm differs from original ID3 in following way:

1. Use of extra Association Function to overcome the short comings of Id3
2. In Improved Id3 more reasonable and effective rules are generated
3. Missing values can be considered(will not have impact on accuracy of decision)
4. Accurate rules
5. Time complexity is more in improved ID3, but it can be neglected because now faster and faster computers are present
6. More no of rules to increase the accuracy of decision
7. Root node is decided not only on value of information gain of attribute but also it considers one more additional function called association function(AF)
8. better information of rules can be excavated

Disadvantages

1. Time complexity is more in improved ID3, but it can be neglected because now days faster and faster computers are available

REFERENCES

[1] Y. T. Zhang, L. Gong, Principles of Data Mining and Technology, Publishing House of Electronics Industry.

[2] D. Jiang, Information Theory and Coding [M]: Science and Technology of China University Press, 2001.

[3] S. F. Chen, Z. Q. Chen, Artificial intelligence in knowledge engineering [M]. Nanjing: Nanjing University Press, 1997.

[4] Z. Z. Shi, Senior Artificial Intelligence [M]. Beijing: Science Press, 1998.

[5] M. Zhu, Data Mining [M]. Hefei: China University of Science and Technology Press, 2002.67-72.