International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org, editorijettcs@gmail.com
Volume 8, Issue 5, September - October 2019 ISSN 2278-6856

Designing an Improved ID3 Decision Tree Algorithm

[1]O Yamini , [2]Dr. S Ramakrishna

[1]Research Scholar, Department of Computer Science,
Sri Venkateswara University, Tirupati, India

[2]Professor, Department of Computer Science,
Sri Venkateswara University, Tirupati, India

Abstract: ***The decision tree is an significant classification process in data mining classification. Aiming at deficiency of ID3 algorism, a new improved categorization algorism is proposed in this paper. The new algorithm combines attitude of Taylor formula with information entropy solution of ID3 algorism, and simplifies the information entropy solution of ID3 algorithm, then assigns a weight value N to beginner's information entropy. It avoids deficiency of ID3 algorism which is apt to sample much value for testing.***

Keywords: ID3 (Iterative Dichotomizer 3,) Decision Tree Classification, Entropy, Information Gain, Taylors Series

1. Introduction to Decision Trees

In data mining and as well as in machine learning, decision tree is one of the predictive model that is mapping from observations about an item to conclusions about its target value. The machine learning technique for producing a decision tree from data is called decision tree learning [1]. Decision tree learning may be a technique for approximating the discrete–valued target functions, within which the learned perform is depicted by a choice tree. Decision tree learning is one in every of the foremost wide used and sensible strategies for inductive reasoning.

Decision tree learning algorithm has been successfully used in various applications and in expert systems in capturing knowledge. The main task performed in these systems is victimization inductive ways to the given values of attributes of associate unknown object to work out applicable classification in step with call tree rules.[2].

2. ID3 (Iterative Dichotomizer 3)

Set Iterative Dichotomizer 3 algorithm [2] is one of the most used algorithms in machine learning and data mining due to its easiness to use and effectiveness. J. Rose Quinlan developed it in 1986 based on the Concept Learning System (CLS) algorithm. It builds a decision tree from some fastened or historic symbolic knowledge so as to find out to classify them and predict the classification of recent data. The data should have many attributes with totally different values. Meanwhile, this data also has to belong to diverse predefined, discrete classes (i.e. Yes/No). Decision tree chooses the attributes for decision making by using information gain (IG) .[3]

3. Implementation of ID3 Algorithm

Step 1: we have choosen the following sample dataset as an example to show the result.

Attributes				Classes
Column 0	**Column 1**	**Column 2**	**Column 3**	**Column 4**
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medim	Bus
Female	1	Cheap	Medim	Train
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medim	Bus
Male	0	Standard	Medim	Train
Female	1	Standard	Medim	Train
Female	1	Expensive	High	Car
Male	2	Expensive	Medim	Car
Female	2	Expensive	High	Car

Note: Column 0: Gender, Column 1: car ownership, Column 2: Travel cost, Column 3: Income level, Column 5: Transportation

Step 2: Entropy and Information gain calculation The basic ID3 method selects each instance attribute classification by using statistical method beginning in the top of the tree. The core question of the method ID3 is how to select the attribute of each pitch point of the tree. A statistical property called Information Gain is defined to measure the worth of the attribute. The statistical quantity Entropy is applied to define the Information Gain, to choose the best attribute from the candidate attributes. The notations of Entropy and Information Gain is defined as:

$$Ent(D) = -\sum_{d=1}^{|k|} P_k \log_2 P_k \quad \text{--(1)}$$

$$Gain(D, a) = Ent(D) - \sum_{v=1}^{|V|} \frac{|D^v|}{|D|}, Ent(D^V) \quad \text{---(2)}$$

Where D = {(x_1, y_1), (x_2, y_2), _ _ _ , (x_m, y_m)} stands for the training sample set, and |D| represents the number of training samples. A = {a_1, a_2, --- , a_d} denotes the attribute set of |D|, and d = {1, 2, ---, |k|}. $p_k(k = 1, 2, --- , |D|)$ stands for the probability that a tuple in the training set S belongs to class C_i. A homogenous data set consists of only one class. In a homogenous data set, p_k is 1, and $\log_2(p_k)$ is zero. Hence, the entropy of a homogenous data set is zero. Assuming that there are V different values ($V = \{ a^1, a^2, --- , a^V\}$ in an attribute a_i, D^i represents sample subsets of every value, and $|D^i|$ represents the number of current samples.

Step 3: Calculating highest information gain.
Now consider the above dataset shown in above table. As this algorithm is based upon information gain of each attribute with entropies. First we need to calculate the Entropy for all the columns (classes and attributes), based on the Entropies of each attributes calculate the information gain for each record set according to attributes given in dataset

Calculating Entropy for the classes:
Record set: [Bus, Bus, Train, Bus, Bus, Train, Train, Car, Car, Car]
Entropy: 1.571
Calculating Entropy for the Attributes Column 0:
Record Set: [Male, Male, Female, Female, Male, Male, Female, Female, Male, Female].
For Male: Record set: [Bus, Bus, Bus, Train, Car]
Entropy: 1.522
For Female: Record set: [Train, Bus, Train, car, Car]
Entropy: 1.371
Information gain for Column 0: 0.12

As per the same way we calculated the information gain for each record set of each and every column.
Below we give the calculation for each record set
Column 1 :
Record set: [0, 1, 1, 0, 1, 0, 1, 1, 2, 2]
Entropy calculated for distinct data of particular column
Information gain: 0.534

Column 2:
Calculating gain for record set: [Cheap, Cheap, Cheap, Cheap, Cheap, Standard, Standard, Expensive, Expensive, Expensive]
Information gain: 1.21

Column 3:
Calculating gain for record set:[Low, Medium, Medium, Low, Medium, Medium, Medium, High, Medium, High]
Information gain: 0.695

Step 4: Generate Information Gain Table

Attributes	**Information Gain**
Column 0	0.125
Column 1	0.534
Column 2	1.21
Column 3	0.695

Highest gain value is identified for (Travelling) column 2 : 1.21
So As per the result we get highest information gain for column no. 2. We design a decision tree based upon the results as we get the highest information gain on column no. 2.

The decision tree is given below

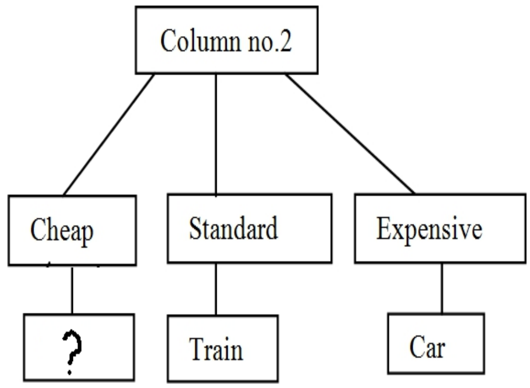


Figure.1 Decision tree based on ID3

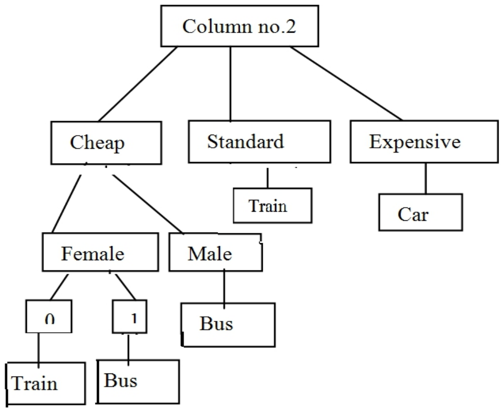
Here we Identified that the root node column 2 has the node Cheap doesn't have the result. By continuing the same procedure we calculate the gain for remaining class data avoiding the results such as Train and Car. Basing on Result calculating entropy of column 4: Record set: [Bus, Bus, Train, Bus, Bus] Entropy: 0.722
Same procedure is followed for each and every column and Information is represented as:

Attributes	**Information Gain**
Column 0	0.322
Column 1	0.171
Column 3	0.171

Highest gain value is identified for (Gender) column 0 : 0.322
So As per the result we get highest information gain for column 0. We design a decision tree based upon the results as we get the highest information gain on column no. 0 behind column 2.

The decision tree is given below

It takes a lots of time to draw a ID3 based Decision It



It takes a lots of time to draw a ID3 based Decision Tree. Data may be Over fitted or Over classified, if a small sample is tested and at a time only single attribute is tested for making a Decision. Classifying continuous data may be computationally expensive, as many trees must be generated to see where to break the continuum.

4.IMPLEMENTATION OF IMPROVED

ID3 ALGORITHM

A new method that puts the solutions along is intended, and it is accustomed build a additional pithy call tree in a shorter running time than ID3. Basing on the Formulae of Entropy we need to use and calculate the Algorithms which increase the complexity of the calculation. If we can find a simpler computing formula, the speed of building a decision tree would be faster. The process of simplification is organized as follows:

As is thought to any or all, in two equivalent expressions, the running speed of the exponent and the logarithmic expression is slower than that of the four arithmetic operations. If the logarithmic expression in Entropy of Id3 is replaced by the four arithmetic operation, the running speed of the whole process to build the decision tree will improve rapidly.

According to the differentiation theory in advanced arithmetic, that which means of Taylor formula will modify advanced functions. The Taylor series is named after the British mathematician Brook Taylor (1685-1731).The Taylor formula is an expanded form at any point, and the Maclaurin formula is a function that can be expanded into Taylor's series at point zero. The difficulty in computation of the knowledge entropy regarding the

ID3 rule maybe reduced supported an approximation formula of Maclaurin formula, that is useful to create a decision tree in a short period of time. The Taylor formula is given by:

$$f(x) = f(x_0) + f^{(1)}(x_0)(x - x_0) + o(x - x_0) \quad \text{----(3)}$$

When x=0, then the above equation will be changed in the following form

$$f(x) = f(0) + f^{(1)}(0) + \frac{f^2(0)}{2!}x^2 + \cdots + \frac{f^n(0)}{n!}x^n + R(x) \quad \text{----(4)}$$

The above Equation is also called as the Maclaurin formula. Where

$$R(x) = \frac{f^{n+1}(x)}{(n+1)!}(x - x_0)^{(n+1)} \quad \text{----(5)}$$

For easy calculation, the final equation applied here is written as:

$$f(x) = f(0) + f^{(1)}(0) + \frac{f^2(0)}{2!}x^2 + \cdots + \frac{f^n(0)}{n!}x^n \quad \text{----(6)}$$

Let us assume that there are n counter examples and p positive examples in sample set D, the information entropy of D can be written as:

$$Ent(D) = \frac{-p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n} \quad \text{---(7)}$$

Assuming that there are V different values included in the attribute a_i of D, and every value contains n_i counter examples and p_i positive examples, the information gain of the attribute a_i can be written as:

$$Gain(D, a) = Ent(D) - \sum_{i=1}^{v} P_{\frac{i+n_i}{p+n}}\ Ent(D^v) \text{---(8)}$$

Where

$$Ent(D^v) = \frac{-p_i}{p_i+n_i}\log_2\frac{p_i}{p_i+n_i} - \frac{n_i}{p_i+n_i}\log_2\frac{n_i}{p_i+n_i} \text{---(9)}$$

For the simplification purpose , if the formula ln(1 + x) ≈x is true in the situation of very small variable x and the constant included in every step can be ignored based on Equation (6), Equation (7) can be rewritten as

$$Ent(D) = \frac{2pn}{p+n} \quad \text{---(10)}$$

Simillarly the expression $\sum_{i=1}^{v}\frac{p_i+n_i}{p+n}Ent(D^V)$ can be rewrite as

$$\sum_{i=1}^{v}\frac{2P_i n_i}{p_i + n_i}$$

Hence Equation (8) can be written as:

$$Gain(D, a_i) == \frac{2pn}{p+n} - \sum_{i=1}^{v}\frac{2P_i n_i}{p_i+n_i} \quad \text{---(11)}$$

Hence, Equation (11) can be used to calculate the information gain of every attribute and the attribute that has the maximal information gain will be selected as the node of a decision tree. The new information gain expression in Equation (11) that only includes addition, $Ent(D^v) = \frac{-p_i}{p_i+n_i}\log_2\frac{p_i}{p_i+n_i} - \frac{n_i}{p_i+n_i}\log_2\frac{n_i}{p_i+n_i}$---(9)

For the simplification purpose , if the formula ln(1 + x) ≈x is true in the situation of very small variable x and the constant included in every step can be ignored based on Equation (6), Equation (7) can be rewritten as

$$Ent(D) = \frac{2pn}{p+n} \quad \text{---(10)}$$

Simillarly the expression $\sum_{i=1}^{v}\frac{p_i+n_i}{p+n}Ent(D^V)$ can be rewrite as

$$\sum_{i=1}^{v}\frac{2P_i n_i}{p_i + n_i}$$

Hence Equation (8) can be written as:

$$Gain(D, a_i) == \frac{2pn}{p+n} - \sum_{i=1}^{v} \frac{2P_i n_i}{p_i+n_i} \quad \text{---(11)}$$

Hence, Equation (11) can be used to calculate the information gain of every attribute and the attribute that has the maximal information gain will be selected as the node of a decision tree. The new information gain expression in Equation (11) that only includes addition, Subtraction multiplication, and division greatly reduces the difficulty in computation and increases the data-handling capacity.

Calculating Information Gain for same dataset reduces the time and complexity. Let us calculate Entropy and Information Gain for the above dataset using Improved ID3Algorithm.

Entropy for data set: 7.2
Information Gain for colomn0: 4.40
Information Gain for colomn1:5.60
Information Gain for colomn2:7.20
Information Gain for colomn3:5.20

From the above the Highest Information Gain of column 2(Travel cost) is Considered as the Root and the same process is repeated as in ID3 algorithm for the splitted data set.

5. RESULTS AND ANALYSIS:

As Compared with the ID3 Algorithm, the calculated time is reduced for Improved ID3 Algorithm. The Practical implementation of these algorithms is done using the MATLAB. The comparison analysis is given below:

Algorithm	Time
ID3	0.1693
Improved ID3	0.0364

Graph

The Resultant Graph is drawn based on the time taken to execute the ID3 Algorithm as well as Improved ID3 Algorithm using MATlab.

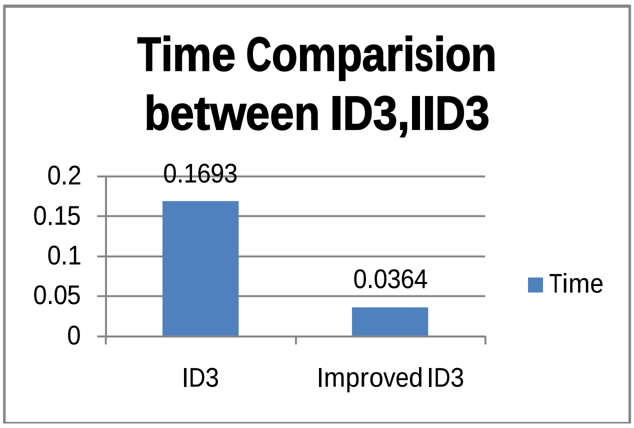


Fig: Comparison between ID3 and IID3 Algorithms

6. CONCLUSION

This paper proposed an improved ID3 algorithm, in which the Entropy and the Information Gain equation that includes the Arithmetic Operations (Addition, Subtraction, Multiplication, and Division) firstly replaced the original information entropy expression that includes complicated logarithm operation in the ID3 along with the above Arithmetic Operations, method for economizing large amounts of running time. It's practically implemented through the application MAT Lab.

Refrence

[1] J. Han and M. Kamber. Data Mining : Concepts and Techniques. Morgan Kaufmann, 2000
[2] http://www.roselladb .com
[3] Xindong Wu, V Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, Dan Steinberg "Top 10 algorithms in data mining," Journal of International Business Studies, XXIII (4), pp. 605-635, 1992. (journal style)
[4] Ahmed S, Coenen F, Leng PH (2006) " Tree-based partitioning of date for association rule mining". KnowlInf Syst 10(3):315 -33 1
[5] Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) , "Classification and regression trees". adsworth, Belmont
[6] Breiman L(1968) "Probability theory". Addison-Wesley,Reading.Republished(1991) in Classics of mathematics.SIAM, Philad elphia.
[7] O. Cord´o n, F. Gomide, F. Herrera, F. Hoffmann, a nd L. Magdalena. "Ten years of genetic fuzzy systems: Current framework and new trends. Fuzzy Sets and Systems"141: 5-31. 2001.
[8] W. W. Cohen. "Fast effective rule induction. In Machine Learning", Proceedings of the 12th International Systems (FUZZ-IEEE'0 4). Piscataway, NJ: IEEE Press, pp. 1 337-1342. 2004Conference, Morga n Kaufmann, pp. 115-1 23. 199 5.
[9] R. B. Bhatt and M. Gopal. "FRID: Fuzzy rough interactive ichotomizers", IEEE International Conference on Fuzzy
[10] Atika Mustafa, Ali Akbar, and Ahmer Sulta n Na tiona,"Knowledge Discovery using Text Mining: A Programmable Implementation on Information Extraction and Categorization", University of Computer and Emerging Sciences-FAST, International Journal of Multimedia and Ub iquito us Engineering Vol. 4, No. 2
[11] C. Apte, F. n Damerau, and S.M. Weiss,"Automated Learning of Decisio n Rules for Text Categorization", ACM Transactions on Information Systems, 1994.
[12] P.P.T.M. van Mun ,"Text Classification in Information Retrieval using Winnow", Department of Computing Science, Catholic

University of Nijmegen Toernooiveld 1, NL-6 525 ED, Nijmegen, The Netherla nds Von Altrock, Constantin (1995).

[13] Zimmermann, H ,"Fuzzy logic and Neuro Fuzzy applications explained. Upper Sad dle River", NJ: Prentice Hall PTR. ISBN 0-13-368465-2 . (2001).

[14] Boston," Fuzzy set theory and its applications", Kluwer Academic Publishers. ISBN 0-7923-743 5-5 .

[15] Andrew Colin, "Building Decision Trees with the ID3 lgorithm", Dr. Dobbs Journal, June 19 96.

[16] George B. Arfken and Hans J. Weber , "Mathematical Methods for Physicists", 2001, pp. 334-340, Academic Press.

AUTHOR

O. Yamini received her MSc. degree from Sri Venkateswara University in 2008. She is currently a research scholar, Computer science department in Sri Venkateswara University, Tirupati. Her research interests includes Data Mining, Data Structures, Artificial Intelligence and Machine learning.

Dr. S Ramakrishna received Ph.D. in mathematics in 1988. He is a professor in Computer science department, Sri Venkateswara University, Tirupati. His research interests include Fluid Dynamics, Network Security and Data Mining. He has guided about 20 Ph.D., students so far. To his credit he has published around 95 research articles in reputed journals.