Overview of Use of Decision Tree algorithms in Machine Learning

Arundhati Navada, Aamir Nizam Ansari, Siddharth Patil, Balwant A.Sonkamble
Department Of Computer Engineering
Pune Institute of Computer Technology
Pune, India.

Abstract— **A decision tree is a tree whose internal nodes can be taken as tests (on input data patterns) and whose leaf nodes can be taken as categories (of these patterns). These tests are filtered down through the tree to get the right output to the input pattern. Decision Tree algorithms can be applied and used in various different fields. It can be used as a replacement for statistical procedures to find data, to extract text, to find missing data in a class, to improve search engines and it also finds various applications in medical fields. Many Decision tree algorithms have been formulated. They have different accuracy and cost effectiveness. It is also very important for us to know which algorithm is best to use. The ID3 is one of the oldest Decision tree algorithms. It is very useful while making simple decision trees but as the complications increases its accuracy to make good Decision trees decreases. Hence IDA (intelligent decision tree algorithm) and C4.5 algorithms have been formulated.**

Keywords- Decision Tree; Machine Learning; ID3; IDA; C4.5; keyword spices; automatic learning; domain specific web search.

I. INTRODUCTION

Life is full of choices at every moment of time. There are multitudes of possibilities in front of you to choose from. You need to choose one course of action from all available options based on your judgment. Therefore we can say that the thought process of selecting a logical choice from among the available options is called decision making. In order to make a good decision a person must weigh the pros and cons of each option and also consider all the alternatives available. decision making is a great problem in front of mankind. Therefore it is easier for us to develop an algorithm that does this work for us more accurately taking all important attributes into consideration without missing out on a single point. For this we use decision trees.

A decision tree is the most widely used tool for decision making. To accomplish this one should draw a decision tree with different branches and leaves. These branches and leafs should point to all the various factors concerning a particular situation. A decision tree is almost like a decision support tool. It uses a tree-like graph of decisions and their possible outcomes which include resource costs, event outcomes, and utility. It is one way to display an algorithm. Depending on the situation and desired outcome there are various types of decision trees that you can use.

1. Classification Tree

One can get the most predictable outcome from the different pieces of information that one has calculated using the classification tree. This kind of tree would find its application in probability and statistics.

2. Regression Tree

In order to determine one single predetermined outcome from different pieces of information Regression Tree. This tree is used in calculations for real estate.

3. Decision Tree Forests

Several varied decision trees are created and then grouped together in order to accurately determine as to what will happen with a particular outcome.

4. Classification and Regression Tree

To make the most logical assumption outcome is predicted using dependent factors.

5. K Means Clustering

It is the least accurate of the decision trees. One combines all of the different factors that one has identified previously while using this decision tree.

Decision tree algorithms (a major part of machine learning [17]) find applications in many fields. It can be used to statistically compare data. It can be used in text classification or extraction. This algorithm can be implemented in libraries, where the titles of the books can be classified into categories of its genre. It can be used in schools, colleges, companies etc. to maintain student and staff records. The algorithm can also be used in hospitals for better diagnosis of a patient with any disease be it brain, heart etc. In stock market, instead of using statistics we could develop such a decision tree that would fulfill its purpose.

II. THEORY OF STUDY OF DECISION TREE

Sometimes we are given a set of records, which we have to analyze and form a conclusion accordingly. In order to do so we often use statistics [4]. But instead, now we propose the use of decision trees instead of statistics as they are more accurate and also they highlight some important attributes which we may often overlook. Let us take an example: measurements taken before and after a certain surgery of 50 records, each described by 20 attributes. We get the most important attributes after applying various different statistic procedures to the set of records. Using the same data set different decision trees was built. It was then observed that there was a very high similarity quotient between the results produced by some statistical methods [13] and some decision trees. The advantage of using decision trees here was that it did not require any statistical knowledge. Therefore we can

very safely say that decision trees can be used very widely in medical fields, to extract or recognize data, or to interpret incomplete data and so on.

To predict the outcome of a severe head injury is indeed a difficult task. What are the clinical parameters that should be taken into consideration to evaluate the severity of the head injury [3] also needs to be predicted in due course of time. This is made easier with the help of induced decision trees. The reason why this prediction is important is because; it gives us better knowledge of the pathophysiological events after head injury and hence makes therapeutic decisions easier. Traditional approaches to this problem is to collect relevant factors pertaining to the head injury, values and then search for the most optimum combinations of these factors that influence the actual outcome after a severe head injury. These approaches are based on gathering in large numbers the data of patients with severe head injury and on the application of statistical methods.

While diagnosing a patient, a doctor often asks for some tests to be performed. Quite often some of these tests turn out to be futile in the final diagnosis of the patient. Hence, the patient and sometimes the insurance company have to bear the wasteful cost of these tests. In order to minimize the total cost [7], [24] of medical tests and also misdiagnosis a cost sensitive machine learning algorithm is designed. In the medical field one can say that medical tests are like misclassification which also adds to the cost and are called as misclassification costs. From the previous outcomes and results for a particular medical disease we can build a classification model for medical diagnosis that takes both attribute cost [21]and misclassification costs [21], [22],[24] in to consideration. But when a doctor sees a new patient he must order for few basic tests in order to reduce the chances for a misclassification cost. This is test strategy. This can be achieved simply by first doing lazy decision tree [23] learning that improves on the previous decision tree algorithm that minimizes the sum of misclassification and attributes costs. After many experiments, test strategies we can also determine the order of the tests, for diagnosis to reduce the total cost. Test strategies so performed were sequential test, single batch test, and multiple batch tests. These tests determine the attributes needed to be taken into consideration and performed, and also its order. After using single decision trees and naïve Bayes [21] on these tests we noticed that Lazy-tree Optimal.

The problem of text classification [2] and keyword extraction is quite tedious. Therefore a new approach to this problem is proposed with the help of decision trees. Here a class of representations is used for classifying text data based on decision trees and also an algorithm for learning it inductively is formulated. This is robust for noisy data and also does not require any language processing technique. This algorithm is hence used for automatic extraction of keywords for text retrieval and also automatic text categorization.

Classification of matter is very important in science. But often the existence of incomplete data makes it more difficult to classify or rather make classification models. Missing data [8] can be understood with the help of the following example. Suppose we have a set X1=(1,2,3,4), now if this X1 was represented as (?,2,3,4) then we would say that X1 has 25% incomplete data and (1,?,3,?) has 50% incomplete data. There are two parts to this classification one being the learning phase and two the classification phase. Based on the data and the relation between the data the classification model is built in the learning phase and the classification phase is to classify the unknown cases to one of the known classes. Sometimes incomplete values appear in classification phase. Based on the given attribute values for the data a decision tree classifier can classify them in their respective classes. C4.5 algorithm [8], [19] is one such that can work for the incomplete data in both classification and learning phases. But C4.5 is not very efficient, so a new algorithm is proposed. The proposed decision tree can solve the incomplete data classification problem very well and can also resolve two other problems being rule refinement problem and importance preference problem. In order to generate a more refined tree the rule refinement problem helps add new coming data into a decision tree. Important preference problem is to combine the information from the different sensors, in different environment with different solutions. Therefore this means that their decision tree can be used to combine the sensors in different environment.

It is often very difficult to recognize hand printed characters automatically. Therefore, a set of measurements on the characters are taken, (which can be manipulated reasonably) so as to categorize these patterns into certain sub-groups. These particular measurements that are taken are the most important ones that reflect the characteristics of these patterns. A binary matrix[9] representation is used. Highleyman and Chow use n weighting matrixes for n input classes to obtain n recognizable patterns. These weightings are computed according to a priori data obtained from the pattern set. Methods taking advantage of the interdependence between points of the pattern may, in general, be expected to have increased recognition ability. The features used are lines and enclosures.

The web is very important for all age groups of today's world. Domain specific [5] web search engines are those that only return web pages relevant to certain domains. Example, Cora[16] (it is a domain specific search engine for computer science research papers). Unlike domain specific search engine there are general purpose search engines (Google, AltaVista) which are not restricted to a specific domain. In these search engines, the user can search through all indexed pages. But this is often complicated to use, as the user may search for a query, which may find a match in pages that are not relevant. Therefore searching in such search engines need a lot of expertise in entering the right keywords [15] which often many users lack. This problem can be greatly reduced by using special search engines designed for the topic of interest. Example: search engines dedicated to recipes will only return cooking recipes of 'beef' as an ingredient, when we search for beef. A domain specific search engine can be built by collecting and indexing only relevant pages available

on the net. Now if these indices are manually constructed, it will be too wearisome and it would be difficult for us to keep up to the pace of the increasing web pages on that particular topic. Another approach is to use crawlers. Even Cora uses crawlers. Cora's crawlers start from the home pages of computer science departments and laboratories and find all research papers effectively using machine learning algorithms [17]. These systems establish their own local databases and can apply various knowledge representation techniques to the data. But the time and network bandwidth consumed by crawlers are excessive in domain that are dispersed across many websites (like home pages). Therefore crawlers are not efficient.

Another approach can be to use keyword spices [20]. The keyword spice model forwards the users input query with a domain specific keyword spice (Boolean expression).This then passes this extended query to the general purpose search engines better classified. Therefore we can assume that all the pages received from the search engine are relevant and no further processing would be required. The importance of this method is its simplicity. Therefore all that is required is a short program that adds keyword spices[20] to the users input and forwards it to a general purpose search engine. The web page is embedded with this program.

The figure (Figure 1) is a typical example of a decision tree, where the preference of a customer to want a particular feature is shown. The customers requesting a particular feature is the first attribute of this tree. If yes the number of customers are further divided into the range of how many requested. If no the number is divided into why the customers wanted this particular feature.

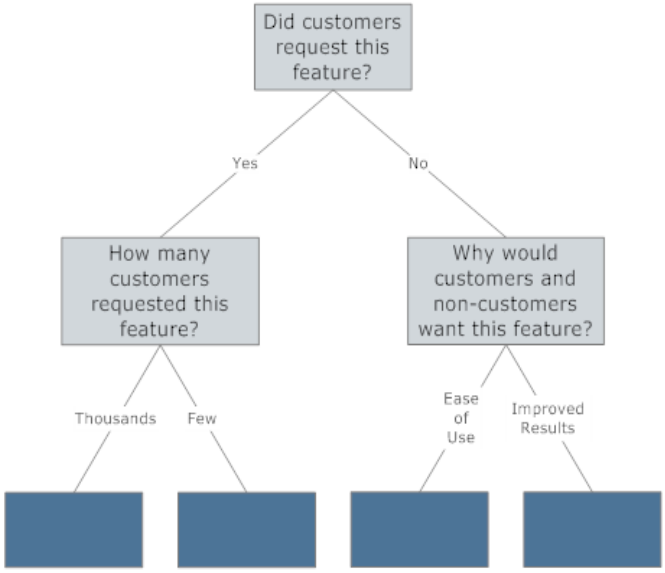

Figure 1.Example of a decision tree

In order to develop an effective decision tree the size of a decision tree is very important and hence needs to be controlled. These measures depend on information entropy theories such as information gain [13], gain ratio, distance based measure. The size can be controlled by avoiding over-fit in the learning process. Pruning [6], [10] is the most effective method to deal with over- fit in decision tree based machine learning process. These are of two types of post and pre pruning. Post-pruning [19] has two commonly used methods which are reduced-error pruning and rule post-pruning. Reduced-error [18] pruning is to consider each of the decision nodes in the tree as a candidate for pruning. The pre-pruning method is to stop the growing of a decision tree earlier, before it perfectly classifies the training data.

As we know the most widely used classification algorithms is the ID3 algorithm[25]. It uses the greedy search procedure and the strategy of learning from examples [12]. Therefore to undo these follies in the decision tree classification algorithm a new classification is proposed called the [1] intelligent decision tree algorithm (IDA). IDA is the improved version of ID3 with better computational efficiency and a better performance at getting a more desired accurate output (decision tree).

III. MACHINE ALGORITHMS FOR DECISION TREES

A. ID3 algorithm

ID3 is a supervised learning algorithm. It is explicitly taught from a series of training examples from several classes. It predicts the class of an item based on the theory that it formulates. ID3 attempts to identify properties (or features) that differentiate one class of examples from others. ID3 requires that all features be known in advance and that each feature is well behaved (that is, all possible values are known in advance). This means a given property must be a continuous number or drawn from a set of options. Age, height, temperature, and country of citizenship are all well-behaved features. The information theory [13] and pattern recognition have been used to develop and formulate the ID3 algorithm. A key feature of information theory is the term information that can often be given a mathematical meaning as a numerically measurable quantity, on the basis of a probabilistic model, in such a way that the solutions of many important problems of information storage and the transmission can be formulated in terms of this measure of the amount of information. The information measure function entropy is used as the criterion function. [1]Given a two group classification problem with each object a n- dimensional vector A= ($a_1, a_2, a_3 \ldots a_n$), to measure the uncertainty of the two classification outcomes, group x_1 or x_2 , the entropy function, can be defined as

$$H(a)= \sum_{A} [-P(x_1 |A)\log_2 P(x_1 |A) - P(x_2 |A)\log_2 P(x_2 |A)],$$

Where P (x_i |A) is the posteriori probability (In the posterior probability of a random event or an uncertain proposition is the conditional probability that is assigned after the relevant evidence is taken into account) of A in population x_i .
Entropy is basically a way of measuring the lack of order that exists in a system.

To determine the most effective partitions ID3 uses a calculation called conditional entropy. Using a feature to partition the data and compute the conditional entropy one can determine which features are most important. The most important feature is the one that gives the lowest entropy. Consider entropy function *H (*a_i*) (i* = 1,. . . ,*n)*.

ID3 Algorithm:
Step 1: The attribute with the smallest entropy is selected after measuring the entropy H (a_i) for each attribute a_i.
Step 2: The corresponding sub-nodes are generated after dividing the entire object set according to their values in attribute a_i. The sub-node is a terminal sub-node if all the objects in a sub-node belong to the same class
Step 3: Otherwise it is a non-terminal sub node. For each such node choose the next attribute a_j with the smallest entropy H(a_i, a_j).
Repeat Step (2) for attribute a_j.

B. C4.5

ID3 has a few limitations. One such being, that it is overly sensitive to features with large number of values. This limitation should be overcome if you are going to use ID3 as an Internet search agent, or any other classification which involves a very large number of attributes or values. This is done by using the C4.5 algorithm [19] which is an extension to the ID3 algorithm. ID3 is sensitive to features with large numbers of values and this is shown by the example of Social Security numbers. Social Security numbers are unique for every individual. Testing on its value will always yield low conditional entropy values. However, this is not a useful test as Social Security numbers do not help predict whether a future medical patient needs surgical intervention or not.

To overcome this problem, C4.5 algorithm uses a metric called information gain[13]. The information gain I(Y |X) of a given attribute X with respect to the class attribute Y is the reduction in uncertainty about the value of Y when we know the value of X. The uncertainty about the value of Y is measured by its entropy, H(Y). The uncertainty about the value of Y when we know the value of X is given by the conditional entropy of Y given X, H(Y |X).

$I(Y|X) = H(Y) - H(Y|X)$.

This computation does not, in itself, produce anything new. However, it allows you to measure a gain ratio[13] . Gain ratio, defined as Gain Ratio (Y|X) = I(Y|X)/H(X),

Where H(X) is the entropy of the examples relative only to the attribute X, measures the information gain of feature X relative to the raw information of the X distribution.

By using the gain ratio instead of the plain conditional entropy, C4.5 reduces problems from artificially low entropy values such as Social Security numbers. So basically the advantage of gain ratio is that Information gain ratio biases the decision tree against considering attributes with a large number of distinct values. So it solves the drawback of information gain -- namely, information gain applied to attributes that can take on a large number of distinct values might learn the training set too well. One of the input attributes might be the customer's credit card [14] number. This attribute has a high information gain, because it uniquely identifies each customer, but we do not want to include it in the decision tree, deciding how to treat a customer based on their credit card number is unlikely to generalize to customers we haven't seen before.

C. IDA

The divergence measure [26] is used by the IDA algorithm and not the entropy measure. Divergence is defined as the act of diverging or the degree by which things diverge.

There are various ways to define divergence for different types of variables. A few of them are,

For probability distributions [1] R and S of a discrete random variable their divergence is defined to be

$$\text{Div}(R|S) = \Sigma\, R(i) \log(R(i)/S(i)) \quad (1)$$

In words, it is the average of the logarithmic difference between the probabilities R and S, where the average is taken using the probabilities R. The divergence is only defined if R and S both sum to 1 and if S(i) > 0 for any i such that R(i) > 0.

For distributions R and S of a continuous random variable, divergence is defined to be the integral:

$$\text{Div}(R|S) = \int_{-\infty}^{\infty} r(x)\log(r(x)/s(x))dx, \quad (2)$$

where r and s denote the densities of R and S.

More generally, if R and S are probability measures over a set X, and S is absolutely continuous with respect to R, then the divergence from R to S is defined as

$$\text{Div}(R|S) = -\int_x \log(dS/dR)\, dR \quad (3)$$

Where dS/dR is the derivative of S with respect to R, and provided the expression on the right-hand side exists.

Let us consider the previous example of a n-dimensional vector A= (a_1 , a_2 , ….. , a_n) , each of these being an attribute. The relation of dependency for attributes a_i and a_j and their performance in combined classification, given a value of attribute a_i , the divergence measure of these two attributes, can be defined as

$$\text{Div}(a_j|a_i) = P(a_j|a_i)(P_1(a_j|a_i) - P_2(a_j|a_i))\log(P_1(a_j|a_i)/P_2(a_j|a_i)) \quad (4)$$

where $P(a_j|a_i)$ is the average conditional probability of a_j given a_i.

When $P_1(a_j|a_i) = P_2(a_j|a_i)$,the conditional divergence function is equal to zero. When the distance between probabilities $P_1(a_j|a_i)$ and $P_2(a_j|a_i)$ increases then $\text{Div}(a_j|a_i)$ increases too. When either $P_1(a_j|a_i) = 0$ or $P_2(a_j|a_i) = 0$ the conditional divergence function becomes discontinuous. No further classification is possible and hence conditional divergence function can be defined as

$$\text{Div}(a_j|a_i) = P(a_j|a_i)(P_1(a_j|a_i) - P_2(a_j|a_i))\log(P_1(a_j|a_i)/P_2(a_j|a_i)) \quad \text{If } P_1(a_j|a_i) \text{ and } P_2(a_j|a_i) > 0; \quad (5)$$

$$\text{Div}(a_j|a_i) = k \times P(a_j|a_i) \quad (6)$$

If $P_1(a_j|a_i)$ or $P_2(a_j|a_i)$, where k is an arbitrary large number.

If two attributes are related closely then together they can worsen the overall classification performance even though when taken individually they may prove to be the best attributes. To take the dependency between attributes into consideration IDA utilizes the global dependency structure as a classification criterion. For the entire data set the global dependency structure is obtained and measured by the conditional divergence function. The locally best solutions calculated may be substituted by other neighboring alternatives as required to better the classification performance globally. The IDA uses the nearest neighbor dependency [11] structure.

An attribute is selected on the basis of individual classification effect and also its combined classification effect with other attribute using the look ahead method. If their combined classification effects with other attributes are poor, then even the individually best attributes will not be selected through this process. Given a collection of n objects A= (a_1 , a_2 , ….. , a_n, C) where C is the class to which an object belongs, the

Intelligent Decision-tree Algorithm:

Step 1: The divergence measures Div($a_j \mid a_d$) is computed for each value a_d with the remaining attributes a_j. The largestest divergence measure Div ($a_j^L|a_d$) is selected.

Step 2: Average divergence measure E (a_d)[Div($a_j^L|a_d$)] = Σ P(a_d) Div($a_j^L|a_d$). is computed for each value of a_i (7)
a_d

Step 3: The first node in the tree is the attribute with the largest divergence measure. E[Div($a_j^L|a_i$)]

Step 4:Sub-nodes are created for each value a_d of a_i.

Step 5: The attribute a_j with the largest divergence measure Div($a_j^L \mid a_d$),will be next attribute for each nonterminal sub-node ,which in turn creates sub-nodes a_{js} . Repeated checks to be performed

Step 6: For each nonterminal sub-node a_{js} generate the next attribute a_k. which is checked against a_d its grandparent node. If attributes a_h and a_i are the same or attribute a_k and a_i are too closely related, go back to Step (6) to find other neighboring alternatives, else proceed to Step (7). Attribute a_k is closely related to a_d if $Div(a_k \mid a_d) <$ $_i$
where $_i$, is the median of {Div(a_k |a_d) k = 1,. . . m} .

Step 7: All attributes with conditional divergence measures greater than or equal to $_i$, in the neighborhood are to be considered. Right from the nearest neighbor with the closest divergence measure till the end. If this attribute is not dependent on its grandparent node too closely and has also not been used, continue, otherwise find the next nearest neighbor until an applicable attribute is found. The last attempted attribute is chosen if no neighboring attribute is found to be applicable.

Step 8: Sub-nodes are created and terminal sub-nodes are marked.

Step 9: For each of the remaining nonterminal nodes go to Step (5).

IV. COMPARATIVE STUDY OF MACHINE ALGORITHMS

A. Comparison between IDA and ID3

The ID3 algorithm uses the greedy search [1]procedure. In the creation of inferior decision trees the ID3 algorithm produces incorrect trees on a lot of occasions. Better classification process is used by the IDA algorithm and also is more computationally [1] more accurate than ID3. A time analysis[1] shows that IDA is more computationally efficient than ID3. The dependency relations among variables are not considered by the ID3 algorithm which worsens the overall classification process outcome of the decision tree. This is successfully overcome by IDA algorithm. As opposed to ID3, global dependency among variables is used by the IDA algorithm. ID3 is compared to the IDA algorithm on the grounds of accuracy, efficiency and effectiveness in a time analysis and a simulation study. The IDA algorithm is more efficient and effective than the ID3 algorithm.

Their computational complexity is analyzed using the elementary steps of both algorithms that is firstly a criterion function to compute, secondly a comparing the values in the search procedure and thirdly an initial solution substitution. Taking an assumption of n discrete attributes with e values each. Then after experimentation the following result was acquired. The total computer time needed by ID3 [1] is approximated by

$F(n,e) = \Sigma\, e^{i-1}\,[(n-i-1) + (n-i)]$ where i extends from 1 to n (8)

The total computer time needed by IDA is approximated by

$G(n,e) = A + \Sigma\, e^{i-1}\,(n-i-1)/2$ (9)

where i extends from 3 to n, $A = ne(n-1)[1+\log_2(m-1)]+(2m-1)$

$F(n,e) = e^{n-1} + O(e^{n-2})$ and $G(n,e) = e^{n-1}/2 + O(e^{n-2})$ (10)(11)

As we can notice the second term of each function is of the same order of magnitude in the 0- Notation, the first term can be used to analyze their differences. The computational complexity of the ID3 algorithm is two times that of IDA algorithm when the input data is in large numbers.

B. Comparison between ID3 and C4.5

C4.5 is an extension of ID3. It can account for unavailable values, continuous attribute value ranges, pruning of decision trees, rule derivation, and so on. Training sets that have records with unknown attribute values can be dealt effectively by evaluating the gain, or the gain ratio, for an attribute by considering only the records where that attribute is defined. By estimating the probability of the various possible results we can classify records that have unknown attribute values.

The ID3 algorithm has certain drawbacks like it can only deal with nominal data. Also it is not able to deal with noisy data sets and it may not be robust in the presence of noise.

The advantages of C4.5 algorithm are that it is robust in the presence of noise. Avoids over fitting and deals with continuous attributes. It also deals with missing attributes and helps convert trees to rules.

C. Comparison between C4.5 and IDA

C4.5 is better than IDA as it deals with continuous attributes and IDA only deals with discrete attributes. Hence C4.5 can be used in more real life situations. The advantages of C4.5 algorithm over IDA is that in the C4.5 algorithm can handle both discrete attributes as well as continuousattributes. In order to handle continuous attributes, C4.5 creates a threshold and then splits the list into those whose attribute value is above the threshold and those that are less than or equal to it. C4.5 marks an attribute as ? while handling missing attributes. The C4.5 algorithm handles attributes with differing costs.

V. CONCLUSION

The ID3 algorithm [25] is a very widely used algorithm, however it has many drawbacks. The variables depend on each other, this is not taken into consideration by the ID3 algorithm; this can worsen the overall classification performance of the decision tree. Also, the variables used can

only be discrete. This algorithm cannot deal with noisy data sets. But all in all, the ID3 algorithm works decently well to make simple decision trees.

An improvement in the ID3 algorithm, where a global dependency between variables is taken into consideration and a ‘ looks ahead’ procedure is adopted to select good attributes in order to get a good classification decision tree, this algorithm is called the intelligent decision tree algorithm (IDA). The computational complexity of the IDA algorithm is half of the ID3 algorithm. But this algorithm has certain drawbacks as well. It can only handle discrete data sets and cannot use training data with missing attribute values.

Therefore a better algorithm has been developed, that overcomes all these shortcomings that is the C4.5 algorithm. This algorithm is more efficient. It deals with continuous and missing attribute values. It is also robust in the presence of noise and it avoids over fitting. Also it handles attributes with varying different costs. So this algorithm can be applied to more real life applications and hence it is more useful.

REFERENCES

[1] Pea-Lei Tu, Jen- Yao Chung “A New Decision-Tree Classification Algorithm for Machine Learning,” Proc. of the 1992 IEEE Int. Conf. on Tools with AI Arlington, VA, Nov. 1992.

[2] Yasubumi Sakakibara, Kazuo Misue, Takeshi Koshiba,” Text classification and keyword extraction by learning decision trees”, 1993 IEEE.

[3] Iztok A. Pilih, Dunja MladeniC, Nada LavraE, Tine S. Prevec3B. Smith, “Using Machine Learning for Outcome Prediction of Patients with Severe Head Injury,” Tenth IEEE Symposium on Computer-Based Medical Systems.

[4] Milan Zorman, Peter Kokol, Milojka Molan stiglic, Alojz GregoriE,” ARE DECISION TREES WAY ARROUND SOME STATISTIC METHODS?”, Proceedings of the 20th Annual International Conference of the ZEEE Engineering in Medicine and Biology Society, Vol. 20, No 3.1998.

[5] Satoshi Oyama, Takashi Kokubo, and Toru Ishida, Fellow, IEEE,” Domain-Specific Web Search with Keyword Spices”, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 16, NO. 1, JANUARY 2004.

[6] DE-SHENG YIN, GUO-YIN WANG, W WU,” A SELF-LEARNING ALGORITHM FOR DECISION TREE PRE-PRUNING”, Proceedings of the Third Intemationd Conference on Machine Leaming and Cybemetics, Shanghai, 26-29 August 2004.

[7] Charles X. Ling, Victor S. Sheng, and Qiang Yang, Senior Member, IEEE,” Test Strategies for Cost-Sensitive Decision Trees”, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 18, NO. 8, AUGUST 2006.

[8] Jun Wu, Yo Seung Kim, Chi-Hwa Song and Won Don Lee,” A New Classifier to Deal with Incomplete Data”, Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing.

[9] D. M. Stern, Ph.B., M.S.E.E., and D. W. C. Shen, B.Sc, Ph.D., Associate Member,“ Character recognition by context-dependent transformations”,PROC. IEEE, Vol. Ill, No. II, NOVEMBER 1964.

[10] Quinlan, J. R., “Discovery and use of decision trees,” in *Illinois Interdisciplinary Workshop on Decision Making,* Champaign-Urbana, Illinois, June,1988.

[11] Chow, C. K., “A recognition method using neighbor dependence,” *IRE Transaction on Electronic Computers,* vol. 11, 1962, pp. 683-690.

[12] Michalski, R. S., Carbonell, J. G., and Mitchell, T. M. (Eds.), *Machine Learning: An Artificial Intelligence Approach,* Palo Alto: Tioga Publishing Company, 1983.

[13] Quinlan JR, Induction of decision trees, *Machine learning,* no.1, pp. 81-106, 1986.

[14] Carter, C. and Catlett, J., “Assessing credit card applications using machine learning,” *IEEE Ezpert,* vol. 2, no. 3, 1987, pp. 71-79.

[15] D. Butler, “Souped-Up Search Engines,” Nature, vol. 405, pp. 112- 115, 2000.

[16] A. McCallum, K. Nigam, J. Rennie, and K. Seymore, “A Machine Learning Approach to Building Domain-Specific Search Engines,” Proc. 16th Int’l Joint Conf. Artificial Intelligence (IJCAI-99), pp. 662- 667, 1999.

[17] T.M. Mitchell, Machine Learning. McGraw-Hill, 1997.

[18] Qinlan, **J.** R., Simpliying decision trees. International Journal of Man-Machine studies, 1987,27(3):221-234

[19] Qinlan, J. R., C4.5: Programs for Machine Learning. SanMateo, CA Morgan Kaufmann, 1993

[20] S. Oyama, T. Kokubo, T. Ishida, T. Yamada, and Y. Kitamura, “Keyword Spices: A New Method for Building Domain-Specific Web Search Engines,” Proc. 17th Int’l Joint Conf. Artificial Intelligence (IJCAI-01), pp. 1457-1463, 2001.

[21] X. Chai, L. Deng, Q. Yang, and C.X. Ling, “Test-Cost Sensitive Naı¨ve Bayesian Classification,” Proc. Fourth IEEE Int’l Conf. Data Mining, 2004.

[22] C. Elkan, “The Foundations of Cost-Sensitive Learning,” Proc. 17th Int’l Joint Conf. Artificial Intelligence, pp. 973-978, 2001.

[23] J. Friedman, Y. Yun, and R. Kohavi, “Lazy Decision Trees,” Proc. 13th Nat’l Conf. Artificial Intelligence, 1996.

[24] C.X. Ling, Q. Yang, J. Wang, and S. Zhang, “Decision Trees with Minimal Costs,” Proc. 21st Int’l Conf. Machine Learning, 2004.

[25] Jackson, A. H., “Machine learning,” *Ezpert Systems,* vol. 5, 1988, pp. 132-150.

[26] Kullback, S., *Information Theory and Statistics,* John Wiley and Sons, Inc., 1959.