*Impact Factor (SJIF): 3.632*

**International Journal of Advance Research in Engineering, Science & Technology**

*e-ISSN: 2393-9877, p-ISSN: 2394-2444*

***(Special Issue for ITECE 2016)***

# Comparative Survey Of Different Classification Techniques In Data Mining

**Asst. Prof. Nishant Sanghani[1], Asst. Prof. Pooja Vasani[2], Asst. Prof. Ravi Khimani[3]**

[1]*Computer Science & Engineering, SLTIET*

[2]*Computer Science & Engineering, AITS*

[3]*Computer Science & Engineering, SLTIET*

***Abstract — Classification is a data mining (machine learning) technique used to predict group membership for data instances. Classification is used to find out in which group each data instance is related within a given dataset. It is used for classifying data into different classes according to some constrains. Several major kinds of classification algorithms / techniques including decision tree induction, Bayesian networks, k-nearest neighbor classifier, case-based reasoning, genetic algorithm, C4.5, ID3 and fuzzy logic techniques. The goal of this survey is to provide a comprehensive review of different classification techniques in data mining.***

***Keywords — Bayesian, classification technique, fuzzy logic, ID3, k-nearest neighbour, Decision Tree induction.***

## I. INTRODUCTION

We are overwhelmed with data. People have been seeking patterns in data since human life began. Hunters seek patterns in animal migration behavior, farmers seek patterns in crop growth, and politicians seek patterns in voter opinion. A scientist's job is to make sense out of data, to discover the underlying model that governs the functioning of the physical world and encapsulate the same in theories that can be used for predicting the future. As the background of all scientific discoveries especially theories has been same, what is new about Data Mining (DM)? The simple answer is that, in DM the volume of the stored data is in the digital form and the search is automated or augmented by a computer.

In DM, it is important to understand the difference between a model and a pattern. Model is a global summary of the dataset and makes statements about any point in the full measurement space while pattern describes a structure, relationship to a relatively small part of the data or the space in which the data would occur [3]. In 1960's, computers were increasingly applied to data analysis problems and it was noted that if one searches long enough, one can always find some model to fit in the dataset but complexity and size of the model were important considerations. Also the aim is to generalize beyond the available data. Figure 1 shows the history of databases systems and DM [4]. And Figure 2 presents the scope of data mining in KDD. Fayyad [5] defined DM as a process of finding models, interesting trends or patterns in large datasets in order to guide decisions about future activities. It requires tools that can help in explaining the data and which are also capable to make predictions out of that. The data takes the form of a set of examples and the output takes the form of predictions on the new examples.

Data mining involves the use of sophisticated data analysis tools to discover previously unknown, valid patterns and relationships in large data set. These tools can include statistical models, mathematical algorithm and machine learning methods. Consequently, data mining consists of more than collection and managing data, it also includes analysis and prediction. Classification technique is capable of processing a wider variety of data than regression and is growing in popularity [1].

There are several applications for Machine Learning (ML), the most significant of which is data mining. People are often prone to making mistakes during analyses or, possibly, when trying to establish relationships between multiple features. This makes it difficult for them to find solutions to certain problems. Machine learning can often be successfully applied to these problems, improving the efficiency of systems and the designs of machines.

Numerous ML applications involve tasks that can be set up as supervised. In the present paper, we have concentrated on the techniques necessary to do this. In particular, this work is concerned with classification problems in which the output of instances admits only discrete, unordered values [1].

Classification techniques in data mining are capable of processing a large amount of data. It can be used to predict categorical class labels and classifies data based on training set and class labels and it can be used for classifying

newly available data. The term could cover any context in which some decision or forecast is made on the basis of presently available information. Classification procedure is recognized method for repeatedly making such decisions in new situations. Here if we assume that problem is a concern with the construction of a procedure that will be applied to a continuing sequence of cases in which each new case must be assigned to one of a set of pre defined classes on the basis of observed features of data. Creation of a classification procedure from a set of data for which the exact classes are known in advance is termed as pattern recognition or supervised learning. Contexts in which a classification task is fundamental include, for example, assigning individuals to credit status on the basis of financial and other personal information, and the initial diagnosis of a patient's disease in order to select immediate treatment while awaiting perfect test results [2].

Some of the most critical problems arising in science, industry and commerce can be called as classification or decision problems. Three main historical strands of research can be identified: statistical, machine learning and neural network. All groups have some objectives in common. They have all attempted to develop procedures that would be able to handle a wide variety of problems and to be extremely general used in practical settings with proven success [2].

## II. TECHNIQUES

### 2.1. ID3 ALGORITHM

Id3 calculation starts with the original set as the root hub. On every cycle of the algorithm it emphasizes through every unused attribute of the set and figures the entropy (or data pick up IG(A)) of that attribute. At that point chooses the attribute which has the smallest entropy (or biggest data gain) value. The set is S then split by the selected attribute (e.g. marks < 50, marks < 100, marks >= 100) to produce subsets of the information. The algorithm proceeds to recurse on each and every item in subset and considering only items never selected before [2]. Recursion on a subset may bring to a halt in one of these cases:

•Every element inthe subset belongs to the same class (+ or -), then the node is turned into a leaf and labeled with the class of the examples
• If thereare no more attributes to be selected but the examples still do not belong to the same class (some are + and some are -) then the node is turned into a leaf and labeled with the most common class of the examples in that subset.
• If thereare no examples in the subset, then this happens when parent set found to be matching a specific value of the selected attribute. For example if there was no example matching with marks >=100 then a leaf is created and is labeled with the most common class of the examples in the parent set.

Working steps of algorithm is as follows,
• Calculate the entropy for each attribute using the data set S.
•Split the set S into subsets using the attribute for which entropy is minimum (or, equivalently, information gain is maximum)
• Construct a decision tree node containing that attribute in a dataset.
•Recurse on each member of subsets using remaining attributes.

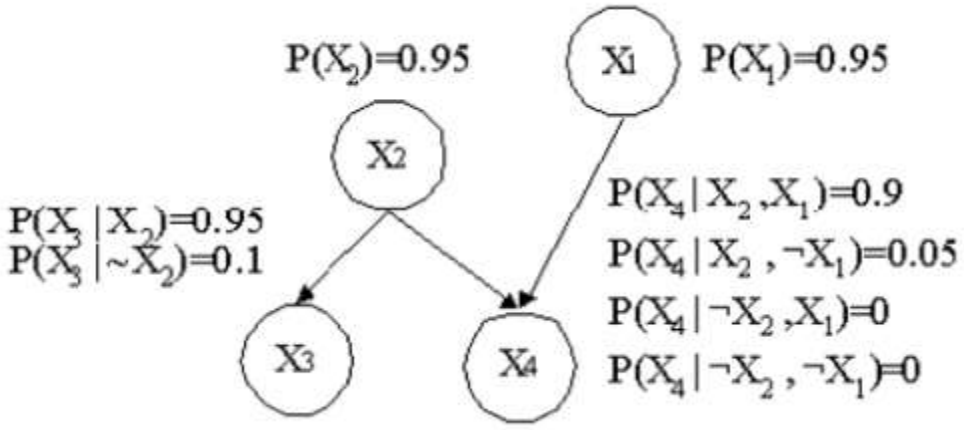
### K NEAREST NEIGHBORS ALGORITHM

The closest neighbour ( NN ) rule distinguishes the classification of unknown data point on the basis of its closest neighbor whose class is already known. M.Cover and P.E.Hart purpose k nearest neighbour (KNN) in which nearest neighbor is computed on the basis of estimation of k that indicates how many nearest neighbors are to be considered to characterize class of a sample data point. It makes utilization of the more than one closest neighbor to determine the class in which the given data point belongs to and consequently it is called as KNN. These data samples are needed to be in the memory at the run time and hence they are referred to as memory-based technique. T. Bailey and A. K. Jain enhance KNN which is focused on weights. The training points are assigned weights according to their distances from sample data point. But at the same time the computational complexity and memory requirements remain the primary concern dependably. To overcome memory limitation size of data set is reduced. For this the repeated patterns which don't include additional data are also eliminated from training data set. To further enhance the information focuses which don't influence the result are additionally eliminated from training data set. The NN training data set can be organized utilizing different systems to enhance over memory limit of KNN. The KNN implementation can be done using ball tree, k-d tree, nearest feature line (NFL), principal axis search tree and orthogonal search tree. The tree structured training data is further divided into nodes and techniques like NFL and tunable metric divide the training data set according to planes. Using these algorithms we can expand the speed of basic KNN algorithm. Consider that an object is sampled with a set of different attributes. Assuming its group can be determined from its attributes; different algorithms can be used to automate the classification process. In pseudo code k-nearest neighbor classification algorithm can be expressed,

K →number of nearest neighbors
For each object Xin the test set do
calculate the distance D(X,Y) between X and every
object Y in the training set
neighborhood! the k neighbors in the training set
closest to X
X.class → SelectClass (neighborhood)
End for

**BAYESIAN NETWORKS**

A Bayesian Network (BN) is a graphical model for probability relationships among a set of variables features. The Bayesian network structure S is a directed acyclic graph (DAG) and the nodes in S are in one-to-one correspondence with the features X. The arcs represent casual influences among the features while the lack of possible arcs in S encodes conditional independencies. Moreover, a feature (node) is conditionally independent from its non-descendants given its parents (X1 is conditionally independent from X2 given X3 if P(X1|X2,X3)=P(X1|X3) for all possible values of X1, X2, X3)




Typically, the task of learning a Bayesian network can be divided into two subtasks: initially, the learning of the DAG structure of the network, and then the determination of its parameters. Probabilistic parameters are encoded into a set of tables, one for each variable, in the form of local conditional distributions of a variable given its parents. Given the independences encoded into the network, the joint distribution can be reconstructed by simply multiplying these tables. Within the general framework of inducing Bayesian networks, there are two scenarios: known structure and unknown structure. In the first scenario, the structure of the network is given (e.g. by an expert) and assumed to be correct. Once the network structure is fixed, learning the parameters in the Conditional Probability Tables (CPT) is usually solved by estimating a locally exponential number of parameters from the data provided (Jensen, 1996). Each node in the network has an associated CPT that describes the conditional probability distribution of that node given the different values of its parents. In spite of the remarkable power of Bayesian Networks, they have an inherent limitation. This is the computational difficulty of exploring a previously unknown network. Given a problem described by n features, the number of possible structure hypotheses is more than exponential in n. If the structure is unknown, one approach is to introduce a scoring function (or a score) that evaluates the "fitness" of networks with respect to the training data, and then to search for the best network according to this score. Several researchers have shown experimentally that the selection of a single good hypothesis using greedy search often yields accurate predictions (Heckerman et al. 1999), (Chickering, 2002). The most interesting feature of BNs, compared to decision trees or neural networks, is most certainly the possibility of taking into account prior information about a given problem, in terms of structural relationships among its features [1]. This prior expertise, or domain knowledge, about the structure of a Bayesian network can take the following forms:

1. Declaring that a node is a root node, i.e., it has no parents.
2. Declaring that a node is a leaf node, i.e., it has no children.
3. Declaring that a node is a direct cause or direct effect of another node.
4. Declaring that a node is not directly connected to another node.
5. Declaring that two nodes are independent, given a condition-set.
6. Providing partial nodes ordering, that is, declare that a node appears earlier than another node in the ordering.
7. Providing a complete node ordering.

A problem of BN classifiers is that they are not suitable for datasets with many features (Cheng et al., 2002). The reason for this is that trying to construct a very large network is simply not feasible in terms of time and space. A final problem is that before the induction, the numerical features need to be discredited in most cases.

**DECISION TREE INDUCTION**

Decision trees are trees that classify instances by sorting them based on feature values. Each node in a decision tree represents a feature in an instance to be classified, and each branch represents a value that the node can assume. Instances are classified starting at the root node and sorted based on their feature values. An example of a decision tree for the training set of Table I.

**Table I. Training set**

| At1 | At2 | At3 | At4 | class |
|---|---|---|---|---|
| a1 | a2 | a3 | a4 | yes |
| a1 | a2 | a3 | b4 | yes |
| a1 | b2 | a3 | a4 | yes |
| a1 | b2 | b3 | b4 | no |
| a1 | c2 | a3 | a4 | yes |
| a1 | c2 | a3 | b4 | no |
| b1 | b2 | b3 | b4 | no |
| c1 | b2 | b3 | b4 | no |

Using the decision tree as an example, the instance At1 = a1, At2 = b2, At3 = a3, At4 = b4∗ would sort to the nodes: At1, At2, and finally At3, which would classify the instance as being positive (represented by the values "Yes"). The problem of constructing optimal binary decision trees is an NP complete problem and thus theoreticians have searched for efficient heuristics for constructing near-optimal decision trees. The feature that best divides the training data would be the root node of the tree. There are numerous methods for finding the feature that best divides the training data such as information gain (Hunt et al., 1966) and gini index (Breiman et al., 1984). While myopic measures estimate each attribute independently, ReliefF algorithm (Kononenko, 1994) estimates them in the context of other attributes. However, a majority of studies have concluded that there is no single best method (Murthy,1998). Comparison of individual methods may still be important when deciding which metric should be used in a particular dataset. The same procedure is then repeated on each partition of the divided data, creating sub-trees until the training data is divided into subsets of the same class. The basic algorithm for decision tree induction is a greedy algorithm that constructs decision trees in a top-down recursive divide-and-conquer manner [1]. The algorithm, summarized as follows.

1. Create a node N;
2. if samples are all of the same class, C then
3. return N as a leaf node labeled with the class C;
4. if attribute-list is empty then
5. return N as a leaf node labeled with the most common class in samples;
6. select test-attribute, the attribute among attribute-list with the highest information gain;
7. label node N with test-attribute;
8. for each known value ai of test-attribute
9. grow a branch from node N for the condition test-attribute= ai;
10. let si be the set of samples for which test-attribute= ai;
11. if si is empty then
12. attach a leaf labeled with the most common class in samples;
13. else attach the node returned by Generate_decision_tree(si,attribute-list_test-attribute)

A decision tree, or any learned hypothesis h, is said to over fit training data if another hypothesis h2 exists that has a larger error than h when tested on the training data, but a smaller error than h when tested on the entire dataset. There are two common approaches that decision tree induction algorithms can use to avoid over fitting training data: i) Stop the training algorithm before it reaches a point at which it perfectly fits the training data, ii) Prune the induced decision tree. If the two trees employ the same kind of tests and have the same prediction accuracy, the one with fewer leaves is usually preferred. Breslow & Aha (1997) survey methods of tree simplification to improve their comprehensibility.

To sum up, one of the most useful characteristics of decision trees is their comprehensibility. People can easily understand why a decision tree classifies an instance as belonging to a specific class. Since a decision tree constitutes a hierarchy of tests, an unknown feature value during classification is usually dealt with by passing the example down all branches of the node where the unknown feature value was detected, and each branch outputs a class distribution. The output is a combination of the different class distributions that sum to 1. The assumption made in the decision trees is that instances belonging to different classes have different values in at least one of their features. Decision trees tend to perform better when dealing with discrete/categorical features [1][3].

**FUZZY LOGIC**

Fuzzy logic, which may be viewed as an extension of classical logical systems, provides an effective conceptual framework for dealing with the problem of knowledge representation in an environment of uncertainty and imprecision [Zad89]. Some of the essential characteristics of fuzzy logic relate to the following:

1. In fuzzy logic, exact reasoning is viewed as a limiting case of approximate reasoning.
2. In fuzzy logic everything is a matter of degree.
3. Any logical system can be fuzzified.
4. In fuzzy logic, knowledge is interpreted as a collection of elastic or equivalently, fuzzy constraint on a collection of variables.

Summary of basic concepts and techniques underlying the application of fuzzy logic to knowledge representation and description of number of examples relating to its use as a computational system is provided in [6]. Fuzzy logic in its pure form is not a technique for classification but it has been a very useful concept in many hybrid techniques for classification.

## III. CONCLUSION

Classification methods are typically strong in modeling interactions. Several of the classification methods produce a set of interacting loci that best predict the phenotype. However, a straightforward application of classification methods to large numbers of markers has a potential risk picking up randomly associated markers.This paper focuses on various classification techniques used in data mining and a study on each of them. Data mining can be used in a wide area that integrates techniques from various fields including machine learning, Each of these methods can be used in various situations as needed where one tends to be useful while the other may not and vice-versa. These classification algorithms can be implemented on different types of data sets like share market data, data of patients, financial data,etc. Hence these classification techniques show how a data can be determined and grouped when a new set of data is available. Each technique has got its own feature and limitations as given in the paper. Based on the Conditions, corresponding performance and feature each one as needed can be selected.

Decision trees and Bayesian Network (BN) generally have different operational profiles, when one is very accurate the other is not and vice versa. On the contrary, decision trees and rule classifiers have a similar operational profile. The goal of classification result integration algorithms is to generate more certain, precise and accurate system results. Numerous methods have been suggested for the creation of ensemble of classifiers. Although or perhaps because many methods of ensemble creation have been proposed, there is as yet no clear picture of which method is best.

## REFERENCES

[1] Thair Nu Phyu," Survey of Classification Techniques in Data Mining", International MultiConference of Engineers and Computer Scientists 2009, Vol. No. 1, March 18 - 20, 2009.
[2] Sagar S. Nikam, "A Comparative Study of Classification Techniques in Data Mining Algorithms", Oriental Journal Of Computer Science & Technology, Vol. No. 8, Issue No.1, Pp. 13-19, April 2015.
[3] Hand, D., Mannila, H., Smyth, P., Principles of Data Mining, Prentice Hall of India, 2001
[4] Han, J., Kamber, M. Data Mining Concepts and Techniques, Morgan Kaufmann Publisher, 2001
[5] Fayyad, U., Data Mining and Knowledge Discovery: Making Sense out of Data, IEEE Expert, Oct. 20-25, 1996.
[6] Zadeh, L. A., Knowledge Representation in Fuzzy Logic, IEEE TKDE, Vol. No.1, Issue No. 1, pp 89-99, 1989.