IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 1 Issue 9, November 2014.
www.ijiset.com
ISSN 2348 – 7968

Survey of Classification Techniques in Data Mining

N.Satyanarayana[1], CH.Ramalingaswamy[2] and Dr.Y.Ramadevi[3]

[1] CSE,CVR College of Engineering,
Hyderabad, Telangana 501510, India

[2] CSE,CVR College of Engineering,
Hyderabad, Telangana 501510, India

[3] CSE CBIT Engineering College
Hyderabad, Telangana 500075, India

Abstract

Classification in data mining is a technique based on machine learning algorithms which uses mathematics, statistics, probability distributions and artificial intelligence... To predict group membership for data items or to represent descriptive analysis of data items for effective decision making .Now a day's data mining is touching every aspect of individual life includes Data Mining for Financial Data Analysis, Data Mining for the Telecommunications Industry Data Analysis, Data Mining for the Retail Industry Data Analysis, Data Mining in Healthcare and Biomedical Research Data Analysis, and Data Mining in Science and Engineering Data Analysis, etc. The goal of this survey is to provide a comprehensive review of different classification techniques in data mining based on decision tree, rule based Algorithms, neural networks, support vector machines, Bayesian networks, and Genetic Algorithms and Fuzzy logic.

Keywords: classifiers, data mining techniques, Decision Support System, Fuzzy sets, Genetic Algorithm.

1. Introduction

Classification is a data mining function that assigns items in a collection to target Categories or classes. The objective of classification is to accurately predict the target class for each record in the data. For example, a classification model used to identify loan applicants as low, medium, or high credit risks. A classification model that predicts credit risk could be developed based on observed data for many loan applicants over a period of time. In addition to the historical credit rating, the data might track employment history, home ownership years of residence, number and type of investments, And so on. Credit rating would be the target, the other attributes are the predictors, and the data for each Customer constitute a case. Classifications are discrete and do not imply order. A predictive classifier with a numerical target uses a regression algorithm, not a classification algorithm. The Simplest type of classification problem is binary classification. Where, the target attribute has only two values: for example, yes or no. Multiclass targets have more than two values: for example, low, medium, high, or unknown credit rating. Classification algorithm finds relationships between the values of the predictors and the values of the target. Different Classification algorithms use different techniques for finding relationships. These relationships are summarized in a model, which can then be applied to a different dataset in which the class assignments are unknown. Classification models are tested by comparing the predicted values to known target values in a set of test data. The historical data for a classification project is typically divided into two data sets: one for learning the model; the other for testing the model..

1.1 Building Classification Model:

Classifier learning can be done in two ways i.e., supervised learning and unsupervised learning. where unsupervised is finding the class label for a record in training data set for which class label is not predefined in fact clustering is most used unsupervised learning for example in marketing grouping customers based on similarities. Supervised is finding the class label for a record or object in training dataset for which class label is predefined. For example grade obtained by student in the exam predefined classes are first class, second class, third class and fail.

First describe a set of predetermined classes. Now each tuple is assumed to belong to a predefined class, as determined by the class label attribute (**supervised learning**).The set of tuples used for model construction: training set. The model is represented as classification rules, decision trees, or mathematical formulae.

1.2 Test metrics for classification Model:

For classifying previously unseen objects Estimate accuracy of the model using a test set of attributes The known label of test sample is Compared with the classified result from the model. Accuracy of classifier is measured by the percentage of test set samples that are correctly classified by the model.

ISSN 2348 – 7968

Test set is different from training set, otherwise over-fitting will occur. Or not for only few records where under fitting will occur.

A classification model is tested using test data with known target values and comparing the predicted values with the known values. The test records must have the attributes which are used to build the model and must be prepared in the same way the model is prepared .Typically the build data and test data come from the same historical data set. A percentage of the records are used to build the model; the remaining records are used to test the model. Test metrics are used to assess how accurately the model predicts the known values. If the model performs well and meets the business requirements, it can then be applied to new data to predict the future.

Three techniques are used to calculate a classifier's accuracy. One is to split the training set by using two-thirds for training and the other third for estimating performance. Other is, known as cross-validation; the training set is divided into mutually exclusive and equal-sized subsets and for each subset the classifier is trained on the union of all the other subsets. The average of the error rate of each subset is therefore an estimate of the error rate of the classifier. The third in is Leave-one-out validation which is a special case of cross Validation. All test subsets consist of a single instance. This type of validation is, of course, more expensive computationally, but useful when the most accurate Estimate of a classifier's error rate is required.

2. Issues in Classification algorithms

Inductive machine learning is the process of learning a set of rules from specific instances (examples in a training set), generally creating a classifier that can be used to generalize from new instances.

The first step is collecting data set. If a requisite expert is available, then she/he suggests which fields (attributes, features) are the most informative. If not, then the simplest method is that of "brute- force," which means measuring everything available in the hope that the right (informative, relevant) features can be isolated after performing rigorous exercise. However, a dataset collected by the "brute-force" method is may not directly suitable for induction. It contains in most cases noise and missing feature values, and therefore requires significant pre-processing [1].

The second step is the data preparation and pre-processing. Based on the circumstances, researchers have a different methods to select from to handle missing data [7] have recently introduced a survey of contemporary techniques for outlier (noise) detection. These researchers have identified the techniques 'and their pros and cons.

Instance selection used to handle noise and to cope with the infeasibility of learning from very large datasets. Instance selection is an optimization problem that attempts to maintain the mining quality while minimizing the sample size .It reduces data and enables a data mining algorithm to function and work effectively with very large datasets. There are different procedures for sampling instances from a large dataset. Feature subset selection or attribute selection for classifying records of data set is the process of identifying and removing as many irrelevant and redundant attributes as possible this reduces the dimensionality of the data and enables data mining algorithms to operate faster and more effectively. Objective is to consider only attribute so important and play critical role in decision making when classifier is build and used to classify an unseen record.

As a matter of fact that many features depend on one another often unduly influences the accuracy of supervised ML classification models. This kind of problem can be addressed by constructing new features from the basic feature set of individual records of data set. This technique is called feature construction/transformation. Now newly Obtained features may lead to the creation of more concise, simple and accurate classifiers. In addition, the discovery of meaningful features contributes to better comprehensibility of the produced classifier, and a better understanding and representation of the learned concept.

The choice of which specific learning algorithm we should use is a critical step. Once preliminary testing is judged to be satisfactory, the classifier (mapping from unlabeled instances to classes) is available for routine use. The classifier's evaluation is most often based on prediction accuracy (the percentage of correct prediction divided by the total number of predictions). If the error rate evaluation is unsatisfactory, we must return to a previous stage of the supervised ML process. A variety of factors must be observed: perhaps relevant features for the problem are not being considered, if a larger training Set is needed, the dimensionality of the problem is too high, the selected algorithm may not suitable or Parameter tuning is needed. One more problem could be that the dataset gets imbalanced. A common Method for comparison supervised ML algorithms is to perform some statistical comparisons for the

ISSN 2348 – 7968

accuracy of trained classifiers on specific datasets. If we have enough data,

We can sample a number of training sets of size N, run the two learning algorithms on each of them, and then estimate the difference in accuracy for each pair of classifiers on a large test set. The average of these differences is an estimate of the expected difference in generalization error across all possible training sets of size N, and also we calculate their variance. Our next step is to perform paired t-test. This test can produce two types of errors. Type I error is the probability that the test rejects the null hypothesis incorrectly .Type II error is the probability that the null hypothesis is not rejected, when there actually is a difference. The test's Type I error will be close to the chosen significance level.

In general, however, we often have only one dataset of size N and all estimates must be obtained from this dataset. Different training sets are obtained by sub-sampling, which are not used for sub sampling are used for testing. Unfortunately this violates the independence assumption necessary for proper significance testing. The consequence of this is that Type I errors exceed the significance level. This is problematic because it is important for the researcher to be able to control Type I errors and know the probability of incorrectly rejecting the null hypothesis. Several heuristic versions of the t-test have been developed to address this problem. here we try explore some classification algorithms for supervised learning techniques.

3. DECISION TREE INDUCTION BASED ALGORITHMS

Decision trees are trees that classify instances by sorting them based on attribute values. Each node in a decision tree represents a attribute 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 Attribute values. An example of a decision tree for the training set is given in the following table. 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 divides the training set. There are different methods for finding the feature that best divides the training data such as information gain. Relief algorithm estimates them in the context of other attributes.

.

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

However, most of studies have concluded that there is no single best method. 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.

Decision tree algorithm is based on greedy algorithm that constructs decision based on divide and conquers strategy. The algorithm, summarized as follows.

Pseudo Code: Decision Tree algorithm

1. Root node is to be created labeled as N
2. If samples all are of 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. test-attribute is selected, the attribute among different attributes of the sample with the highest information gain;
7. Label node N with test-attribute;
8. For each known value a_i of test-attribute.
9. Grow a branch from node N for the condition test-attribute= a_i;
10. Let s_i be the set of samples for which test-attribute= a_i;
11. If s_i is empty then
12. A leaf labeled with the most common class in samples;
13. Else attach the node returned by Generate_decision_tree

There are many specific decision-tree algorithms. Few of them are given below:

- ☐ ID3 (Iterative Dichotomies 3)
- ☐ C4.5 (successor of ID3)
- ☐ CART (Classification And Regression Tree)

ISSN 2348 – 7968

- CHAID (CHi-squared Automatic Interaction Detector). Performs multi-level splits when computing classification trees
- MARS extends decision trees to handle numerical data better.

3.1. ITERATIVE DICHOTOMISER 3(ID3) ALGORITHM

ID3 algorithm begins with the original set as the root node. On each iteration of the algorithm, it iterates through every unused attribute of the set and based on entropy or information gain calculated Then selects the attribute which has the smallest entropy (or largest information gain) value. The set is **S** then split by the selected attribute (e.g. age < 50, 50 <= age < 100, age >= 100) to produce subsets of the data. The algorithm continues to recurse on each subset, considering only attributes never selected before. Recursion on a subset may stop in one of these cases:
Every element in the subset belongs to the same class (+ or -), then the node is turned into a leaf and labeled with the class of the examples

There are 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 the subset

There are no examples in the subset, this happens when no example in the parent set was found to be matching a specific value of the selected attribute, for example if there was no example with age >= 100. Then a Leaf is created, and labeled with the most common class of the examples in the parent set.

Throughout the algorithm, the decision tree is constructed with each non-terminal node representing the selected attribute on which the data was split, and terminal nodes representing the class label of the final subset of this branch.

Pseudo code: ID3 (Examples, Target Attribute, Attributes)

1. Create first node designated as root node

If all examples are positive, Return the single-node tree Root, with label = +. If all examples are negative, Return the single-node tree Root, with label = -.

2. If number of predicting attributes is empty, then Return the single node tree Root, with label = most common value of the target attribute in the examples. Otherwise Begin A ← The Attribute that best classifies examples.

Decision Tree attribute for Root = A. For each possible value, V_I, of A,
3. Add a new tree branch below Root, corresponding to the test A =V_I.
Let Examples (V_I) be the subset of examples that have the value V_I for A
If Examples (V_I) is empty

4. Then below this new branch add a leaf node with label = most common target value in the examples Else below this new branch add the sub tree ID3 (Examples (V_i), Target Attribute, Attributes – {A}) end

3.2 C4.5 ALGORITHM

C4.5 is an algorithm used to generate a decision tree developed by Ross Quinlan. C4.5 is an extension to ID3 algorithm. The decision trees generated by C4.5 can be used for classification, and for this reason, C4.5 is often referred to as a statistical classifier One limitation of ID3 is that it is overly sensitive to features with large numbers of values. This must be overcome if you are going to use ID3 as an Internet search agent. This problem is addressed by the C4.5 algorithm, an ID3 extension.C4.5 is highly sensitive to samples with large numbers of values is illustrated by Social Security numbers. Since Social Security numbers are unique for everyone, testing on its value will always give low conditional entropy values. However, this is not that much useful test. To overcome this problem, C4.5 uses a metric called "information gain," which is defined by subtracting conditional entropy from the base entropy; that is, Gain (P|X) =E (P)-E (P|X). This computation does not, in itself, produce useful information. However, it allows you to measure a gain ratio. Gain ratio, defined as Gain Ratio (P|X) =Gain (P|X)/E(X), where*(X)* is the entropy of the examples relative only to the attribute. Which has an enhanced method of tree pruning that reduces misclassification errors due noise or too-much details in the training data set. Like ID3 the data is sorted at every node of the tree in order to determine the best splitting attribute. it uses gain ratio impurity measure to calculate best split attribute of given sample. Decision trees are built in C4.5 by using a set of training data or data sets as in ID3. At each node of the tree, C4.5 chooses one attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. Its criterion is the

ISSN 2348 – 7968

normalized information gain (difference in entropy) that results from choosing an attribute for splitting the data. The attribute with the highest normalized information gain is selected as decision attribute.

Pseudo Code:

1. Test for base cases.
2. For each attribute a calculate:
   Normalized information gain from splitting on Attribute
3. Select the best attribute that has highest information gain.
4. Create a decision node that splits on best of a, as rot node.
5. Recurs on the sub lists obtained by splitting on best attribute and add those nodes as children node

3.3 CART ALGORITHM

CART (Classification and Regression Trees) is very similar to C4.5, but it differs in that it supports numerical target variables (regression) and does not compute rule sets. CART constructs binary trees using the feature and threshold that yield the largest information gain at each node. It is a Binary decision tree algorithm Recursively partitions data into 2 Subsets so that cases within each subset are more homogeneous Allows consideration of misclassification costs, prior distributions, cost-complexity pruning.

Pseudo Code:

1. The basic idea is to choose a split at each node so that the data in each

Subset (child node) is “purer” than the data in the parent node. CART

Measures the impurity of the data in the nodes of a split with an impurity measures i (t).

2. If a split s at node t sends a proportion pL of data to its left child node tL and a corresponding proportion pR of data to its right child node tR , the decrease in impurity of split s at node t is defined as

$$\Delta i(s, t) = i(t) - pLi(Tl) - pRi(TR)$$

= impurity in node t – weighted average of impurities in nodes tL and tR

3. A CART tree is grown, starting from its root node (i.e., the entire training data set) t=1, by searching for a split s* among the set of all possible candidates S which give the largest decrease in impurity.

4. The above split searching process is repeated for each child node.

5. The tree growing process is stopped when all the stopping criteria are met.

3.4 CHAID

CHAID is a type of decision tree technique, based upon adjusted significance testing (Bonferroni testing). The technique was developed in South Africa and was published in 1980 by Gordon V. Kass, who had completed a PhD thesis on this topic. CHAID can be used for prediction (in a similar fashion to regression analysis, this version of CHAID being originally known as XAID) as well as classification, and for detection of interaction

Between variables. CHAID stands for **Ch**i-squared **A**utomatic **I**nteraction **D**etection, based upon a formal extension of the US AID (Automatic Interaction Detection) and THAID (Theta Automatic Interaction Detection) procedures of the 1960s and 70s, which in turn were extensions of earlier research, including that performed in the UK in the 1950s.

In practice, CHAID is often used in the context of direct marketing to select groups of consumers and predict how their responses to some variables affect other variables, although other early applications were in the field of medical and psychiatric research.

Like other decision trees, CHAID's advantages are that its output is highly visual and easy to interpret. Because it uses multiway splits by default, it needs rather large sample sizes to work effectively, since with small sample sizes the respondent groups can quickly become too small for reliable analysis. One important advantage of CHAID over alternatives such as multiple regressions is that it is non-parametric.

3.5MARS (Multivariate adaptive regression splines)

Multivariate adaptive regression splines (MARS) is a form of regression analysis introduced by Jerome H. Friedman in 1991.It is a non-parametric regression technique and can be seen as an extension of linear cubic spline models that automatically models non-linearity’s and interactions between variable[10].

3.6 Decision tree Advantages:

- ☐ **Simple to understand and interpret.** People are able to understand decision tree models after a brief explanation.
- ☐ **Requires little data preparation.** Other techniques often require data normalization, dummy variables need to be Created and blank values to be removed.
- ☐ **Able to handle both numerical And categorical data.** Other techniques are usually specialized in analyzing datasets that

ISSN 2348 – 7968

have only one type of variable. (For example, relation rules can be used only with nominal variables while neural networks can be used only with numerical variables.)

- **Uses a white box model.** If a given situation is observable in a model the explanation for The condition is easily explained by Boolean logic. (An example of a black box model is an artificial neural network since the explanation for the results is difficult to understand.)
- **Possible to validate a model using statistical tests.** That makes it possible to account for the reliability of the model.
- **Robust.** Performs well even if its assumptions are somewhat violated by the true model from which the data were generated.
- **Performs well with large datasets.** Large amounts of data can be analyzed using standard computing resources in reasonable time.

Limitations:

- The problem of learning an optimal decision tree is known to be NP-complete under several aspects of optimality and even for simple concepts. Consequently, practical decision-tree learning algorithms are based on heuristics such as the greedy algorithm where locally-optimal decisions are made at each node. Such algorithms cannot guarantee to return the globally-optimal decision tree. To reduce the greedy effect of local-optimality some methods such as the dual information distance (DID) tree were proposed.
- Decision-tree learners can create over-complex trees that do not generalize well from the Training data. (This is known as over fitting Mechanisms such as pruning are necessary to avoid this Problem (with the exception of some algorithms such as the Conditional Inference approach that does not require pruning
- There are concepts that are hard to learn because decision trees do not express them Easily, such as XOR, parity or multiplexer problems. In such cases, the decision tree becomes prohibitively large. Approaches to solve the problem involve either changing the representation of the problem domain (known as propositionalisation)[] or using learning algorithms based on more expressive representations (such as statistical relational learning or inductive Logic programming).
- For data including categorical variables with different numbers of levels, information gain in decision trees is biased in favor of those attributes with more levels. However, the issue of biased predictor selection is avoided by the Conditional Inference approach.

4.NAIVE BAYES ALGORITHM

The Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be "independent feature model". Bayesian classification provides practical learning algorithms and prior knowledge and observed data can be combined. Bayesian Classification provides a useful perspective for Understanding and evaluating many learning algorithms. It calculates explicit probabilities for hypothesis and it is robust to noise in input data. A naive Bayes classifier assumes that the presence or absence of a particular feature is unrelated to the presence or absence of any other feature, given the class variable. For some types of probability models, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. It also called idiot's Bayes, simple Bayes, and independence Bayes. This method is important for several reasons. It is very easy to construct, not needing any complicated iterative parameter estimation schemes. This means it may be readily applied to huge data sets. It is easy to interpret, so users unskilled in classifier technology can understand why it is making the classification it makes. And finally, it often does surprisingly well: it may not Probabilistic approaches to classification typically involve modeling the conditional probability distribution $\boldsymbol{P(C|D)}$, where C ranges over classes and D over descriptions, in some language, of objects to be classified. Given a description d of a particular object, we assign the class $\boldsymbol{argmax_c P(C = c|D = d)}$. A Bayesian approach splits this posterior distribution into a prior distribution $\boldsymbol{P(C)}$ and a likelihood $\boldsymbol{P(D|C)}$: $\boldsymbol{P(D = d|C = c)\ P(C = c)\ argmax_c P(C = c|D = d) = argmax_c p(\ )\ (\ )}$

The denominator $\boldsymbol{P(D = d)}$ is a normalizing factor that can be ignored when determining the maximum *a* posteriori class, as it does not depend on the class. The key term in Equation (1) is $\boldsymbol{P(D = d|C = c)}$, the likelihood of the given description given the

ISSN 2348 – 7968

class (often abbreviated to ***P* (*d*|*c*)**). A Bayesian classifier estimates these likelihoods from training data, but this typically requires some additional simplifying assumptions. For instance, in an attribute-value representation (also called propositional or single-table representation), the individual is described by a vector of Values ***a*1. . . *an*** for a fixed set of attributes ***A*1, . . *An*.** Determining ***P* (*D* = *d*|*C* = *c*)** here requires an estimate of the joint probability ***P*(*A*1 = *a*1, . . . ,*An* = *an*|*C* = *c*)**, abbreviated to ***P*(*a*1, . . . ,*an*|*c*)**. This joint probability Distribution is problematic for two reasons:

(1) Its size is exponential in the number of attributes *n*, and

(2) It requires a complete training set, with several examples for each possible description. These problems vanish if we can assume that all attributes are independent Given the class:

$$P(A1 = a1, \ldots, An = an|C = c) = \Pi^{n}_{i=1} P(Ai = ai|C = c)$$

This assumption is usually called the naive Bayes assumption, and a Bayesian classifier using this assumption is called the naive Bayesian classifier, often abbreviated to „naive Bayes". Effectively, it means that we are ignoring interactions between attributes within individuals of the same class.

5. SUPPORT VECTOR MACHINES (SVM)

SVMs introduced in COLT-92 by Boser, Guyon & Vapnik. Theoretically well motivated algorithm developed from Statistical Learning Theory since the 60s.The support vector machine usually deals with pattern classification that means this algorithm is used mostly for classifying the different types of patterns. Now, there is different type of patterns i.e. Linear and non-linear. Linear patterns are patterns that are easily distinguishable or can be easily separated in low dimension whereas non-linear patterns are patterns that are not easily distinguishable or cannot be easily separated and hence these type of patterns need to be further manipulated so that they can be easily separated. Basically, the main idea behind SVM is the construction of an optimal hyper plane, which can be used for classification, for linearly separable patterns. The optimal hyper plane is a hyper plane selected from the set of hyper planes for classifying patterns that maximizes the margin of the hyper plane i.e. the distance from the hyper plane to the nearest point of each patterns. The main objective of SVM is to maximize the margin so that it can correctly classify the given patterns i.e. larger the margin size more correctly it classifies the patterns. The equation shown below is the hyper plane:

Hyper plane, $aX + bY = C$

The given pattern can be mapped into higher dimension space using kernel function, $\Phi(x)$.
i.e. $x \longrightarrow \Phi(x)$ electing different kernel function is an important aspect in the SVM-based classification, commonly used kernel functions include LINEAR,POLY, RBF, and SIGMOID. For e.g.: the equation for Poly Kernel function is given as:

K(x, y) =<x, y>^p

The main principle of support vector machine is that given a set of independent and identically distributed training sample $\{(x_i, y_i)\}^N$ i=1, where x ϵ Rd and yi ϵ {−1, 1}, denote the input and output of the classification.

The goal is to find a hyper plane $w^T.x + b = 0$, which separate the two different samples accurately. Therefore, the problem of solving optimal classification now translates into solving quadratic programming problems. It is to seek a partition hyper plane to make the bilateral blank area (2/||w||) maximum, which means we have to maximize the weight of the margin. It is expressed as:

Min $\Phi(w) = ½ \| w \| 2 = ½ (w, w)$,

Such that: $\mathbf{y_i (w.x_i + b) >= 1}$

SVM can be easily extended to perform numerical calculations. Here we discuss two such extensions. To extend SVM to perform regression analysis, where the goal is to produce a linear function that can approximate that target function.

6. K-NEARESTNEIGHBOUR (kNN) ALGORITHM

The nearest neighbor (NN) rule identifies the category of unknown data point on the basis of its nearest neighbor whose class is already known. This rule is widely used in pattern recognition text categorization ranking models object recognition and event recognition applications. M. Cover and P. E. Hart purpose k-nearest neighbor (kNN) in which nearest neighbor is calculated on the basis of value of k that specifies how many nearest neighbors are to be considered to define class of a sample data point. It

ISSN 2348 – 7968

makes use of the more than one nearest neighbor to determine the class in which the given data point belongs to and hence it is called as K-NN. 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 improve kNN which is based on weights the training points are assigned weights according to their distances from sample data point. But still, the computational complexity and memory requirements remain the main concern always. To overcome memory limitation, size of data set is reduced. For this, the repeated patterns, which do not add extra information, are eliminated from training samples .To further improve, the data points which do not affect the result are also eliminated from training data set . The NN training data set can be structured using various techniques to improve over memory limitation of kNN. The kNN implementation can be done using ball tree, k-d tree, nearest feature line (NFL), tunable metric, principal axis search tree and orthogonal search tree. The tree structured training data is divided into nodes, whereas techniques like NFL and tunable metric divide the training data set according to planes. These algorithms increase the speed of basic kNN algorithm. Suppose that an object is sampled with a set of different attributes, but the group to which the object belongs is unknown. Assuming its group can be determined from its attributes.

Different algorithms can be used to automate the classification process. With the k-nearest neighbor technique, this is done by evaluating the k number of closest neighbors In pseudo code, k-nearest neighbor classification algorithm can be expressed

K ← number of nearest neighbors **For each** object *X* in 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 ← Select Class (neighborhood)

End for

The k-nearest neighbors‟ algorithm is the simplest of all machine learning algorithms. It has got a wide variety of applications in various fields such as Pattern recognition, Image databases, Internet marketing, Cluster analysis etc In binary (two class) classification problems, it is helpful to choose *k* to be an odd number as this avoids tied votes. Here a single number k‟ is given which is used to determine the total number of neighbors that determines the classification. If the value of k=1, then it is simply called as nearest neighbor. K-NN requires an integer k, a training data set and a metric to measure closeness.

7. Rule based classifier

Rule-based classifier makes use of set of IF-THEN rules for classification. We can express the rule in the following from: **IF condition THEN conclusion**

Let us consider a rule R1 R1: IF age=youth AND student=yes THEN buy computer=yes

The IF part of the rule is called rule antecedent or precondition. The THEN part of the rule is called rule consequent. In the antecedent part the condition consists of one or more attribute tests and these tests are logically ANDed.The consequent part consist class prediction.

Note: We can also write rule R1 as follows: R1: (age = youth) ^ (student = yes))(buys computer = yes)

If the condition holds the true for a given tuple, then the antecedent is satisfied. Here we will learn how to build a rule based classifier by extracting IF-THEN rules from decision tree. Points to remember to extract rule from a decision tree, one rule is created for each path from the root to the leaf node. To from the rule antecedent each splitting criterion is logically ANDed.The leaf node holds the class prediction, forming the rule consequent.

7.1 Rule Induction Using Sequential covering Algorithm:

Sequential Covering Algorithm can be used to extract IF-THEN rules form the training data. We do not require to generate a decision tree first. In this algorithm each rule for a given class covers many of the tuples of that class. Some of the sequential Covering Algorithms are AQ, CN2, and RIPPER. As per the general strategy the rules are learned one at a time. For each time rules are learned, a tuple covered by the rule is removed and the process continues for rest of the tuples. This is because the path to each leaf in a decision tree corresponds to a rule. The Decision tree induction can be considered as learning a set of rules simultaneously. The Following is the sequential learning Algorithm where rules are learned for one class at a time. When learning a rule from a class Ci, we want the rule to cover all the tuples from class C only and no tuple form any other class.

7.2 Pseudo code: Sequential Covering

Input: a data set class-labeled tuples (D),

Att_vals, the set of all attributes and their possible values.

Output: A Set of IF-THEN

ISSN 2348 – 7968

rules. Method:

Rule_set={ }; *// initial set of rules learned is empty*
for each class c do
Repeat
Rule = Learn_One_Rule (D, Att_valls);
remove tuples covered by Rule form D;
until termination condition;
Rule_set=Rule_set+Rule; *// add a new rule to rule-set*
End for
Return Rule_Set;

7.3 Rule Pruning

The rule is pruned is due to the following reason: The Assessment of quality is made on the original set of training data. The rule may perform well on training data but less well on subsequent data. That's why the rule pruning is required. The rule is pruned by removing conjunct. The rule R is pruned, if pruned version of R has greater quality than what was assessed on an independent set of tuples. FOIL is one of the simple and effective methods for rule pruning. For a given rule R,FOIL_Prune = pos-neg/ pos+neg Where pos and neg is the number of positive tuples covered by R, respectively. This value will increase with the accuracy of R on pruning set. Hence, if the FOIL Prune value is
Higher for the pruned version of R, then we prune R.

8. Genetic Algorithms

The idea of Genetic Algorithm is derived from natural evolution. In Genetic Algorithm first of all initial population is created. This initial population consists of randomly generated rules. We can represent each rule by a string of bits.

For example, suppose that in a given training set the samples are described by two Boolean attributes such as A1 and A2. And this given training set contains two classes such as C1 and C2.

We can encode the rule IF **A1 AND NOT A2 THEN C2** into bit string **100**. In this bit representation the two leftmost bit represent the attribute A1 and A2, respectively. Likewise the rule **IF NOT A1 AND NOT A2 THEN C1** can be encoded as **001**.If the attribute has K values where K>2, then we can use the K bits to encode the attribute values. The classes are also encoded in the same manner.

Based on the notion of survival of the fittest, a new population is formed to consist of the fittest rules in the current population and offspring values of these rules a well .The fitness of the rule is assessed by its classification accuracy on a set of training samples. The genetic operators such as crossover and mutation are applied to create offspring .In crossover the substring from pair of rules are swapped to form new pair of rules. In mutation, randomly selected bits in a rule's string are inverted.

8.1 Rough Set Approach

To discover structural relationship within imprecise and noisy data we can use the rough set. This approach can only be applied on discrete-valued attributes. Therefore, continuous-valued attributes must be discredited before its use. The Rough Set Theory is base on establishment of equivalence classes within the given training data. The tuples that forms the equivalence class are indiscernible. It the data. There are some classes in given real world data, which cannot be distinguished in terms of available attributes. We can use the rough sets to **roughly** define such classes. For a given class, C, the rough set definition is approximated by two sets as follows:
Table 2: Comparison of Different Algorithms

ISSN 2348 – 7968

Lower Approximation of C –
The approximation of C consists of all the data tuples that Base on knowledge of attribute. These attribute are certain to belong to class C.

Upper Approximation of C - The upper approximation of C consist of all the tuples that based On knowledge of attributes, cannot be described as not belonging to C.

8.2 Fuzzy Set Approaches

Fuzzy Set Theory is also called Possibility Theory. This theory was proposed by Lotfi Zadeh in 1965. This approach is an alternative **Two-value logic**. This theory allows us to work at high level of abstraction. This theory also provides us means for dealing with imprecise measurement of data.

The fuzzy set theory also allows dealing with vague or inexact facts. For example being a member of a set of high incomes is inexact (eg. if $50,000 is high then what about $49,000 and $48,000). Unlike the traditional CRISP set where the element either belongs to S or its complement but in fuzzy set theory the element can belong to more than one fuzzy set.

Probability approach:

- We may assign the statement "Helen is old" the truth value of 0.95. The interpretation is that there is 95% chance of Helen is old

Fuzzy approach:

- The statement could be translated into fuzzy set terminology as follows:
- Helen is a member of the set of old people.
- It could be expressed in symbolic notation of fuzzy set as $\square_{OLD}$(Helen) = 0.95 i.e., Helen's degree of membership within the set of old people = 0.95

CONCLUSION

This paper deals with various classification techniques used in data mining and a study on each of them. Data mining is a wide area that integrates techniques from various fields including machine learning, artificial intelligence, statistics and pattern recognition, for the analysis of large volumes of data. These classification algorithms can be implemented on different types of data sets like data of patients, customers in banking, customers in telecom industry, customers in life, health insurance industry, customers in sales industry students in education, online social networks, web related files, audio files, voice files ,image files and video files and text files 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 pros and cons as given in the paper. Based on the

SNO	Algorithms	Advantages	Disadvantages
1	C4.5	1).It takes less time to build the model 2).produces accurate result 3).it has short search time	1).empty branches 2). Insignificant branches 3).over fitting
2	ID3	1). Produce more accurate result than C4.5 2) uses nominal attribute for classification with no missing values	1).it has long searching time 2). Takes more memory than C4.5
3	Naïve Bayes	1). Improves the performance by removing irrelevant features 2).it takes short computational time	1). Requires large number of records to obtain good results. 2)it stores all the training samples
4	Support Vector Machine	1). Produces very accurate results 2).less over fitting ,robust to noise 3). Good at text classification when high dimensional data to be classified	1). Only good for binary classification 2).computational expensive 3). Runs very slowly
5	K-nearest neighbor	1). Easy to implement 2) . Suitable for multi class classification 3). Training is very fast 4).robust to noisy training data	1).Sensitive to local set of data 2) runs slowly 3).Memory limitation

ISSN 2348 – 7968

needed Conditions each one as needed can be selected On the basis of the performance of these algorithms, these algorithms can also be used to detect and predict the natural disasters like cloud bursting, earth quake, tsunami etc.

REFERENCES

[1] B. Kotsiantis · I. D. Zaharakis · P. E. Pintelas, "Machine learning: a review of classification and combining techniques", Springer Science10 November 2007

[2] Raj Kumar, Dr. Rajesh Verma," Classification Algorithms for Data Mining P: A Survey" IJIET Vol. 1 Issue August 2012, ISSN: 2319 – 1058.

[3] Ms. Aparna Raj, Mrs. Bincy, Mrs. T.Mathu "Survey on Common Data Mining Classification Techniques", International Journal of Wisdom Based Computing, Vol. 2(1), April 2012

[4] http://www.tutorialspoint.com/data_mining/dm_rbc.htm

[5] S.Archana Survey of Classification Techniques in Data Mining International Journal of Computer Science and Mobile Applications Vol.2 Issue. 2, February- 2014 **ISSN: 2321-8363**

[6] Thair Nu Phyu Survey of Classification Techniques in Data Mining Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol I IMECS 2009, March 18 - 20, 2009, Hong Kong

[7] Jensen, F. (1996). An Introduction to Bayesian Networks.Springer

[8] Cover, T., Hart, P. (1967), nearest neighbor pattern classification

[9]http://docs.oracle.com/cd/B28359_01/datamine.111/b_28129.pdf

[10] Text Book: Introduction to data mining by Michael steinback, Pang-Ning Tan, Vipin Kumar.