S.Archana *et al*, International Journal of Computer Science and Mobile Applications,
Vol.2 Issue. 2, February- 2014, pg. 65-71 **ISSN: 2321-8363**

Survey of Classification Techniques in Data Mining

S.Archana[1], Dr. K.Elangovan[2]
[1] Research scholar, School of Computer Science and Engineering, Bharathidasan University
Tiruchirappalli-620023, India
asarchana788@gmail.com

[2]Assistant professor, School of Computer Science and Engineering, Bharathidasan University
Tiruchirappalli-620023, India
murthy.elango@gmail.com

Abstract
Classification is a data mining technique based on machine learning which is used to classify each item in a set of data into a set of predefined classes or groups. Classification methods make use of mathematical and statistical techniques such as decision trees, linear programming, neural network and support vector machines. Classification is process of generalizing the data according to different instances. Several major kinds of classification algorithms including C4.5, k-nearest neighbour classifier, Naive Bayes, SVM, and IB3.This paper provide an inclusive survey of different classification algorithms and their advantages and disadvantages.

Keywords— C4.5, ID3, Naive Bayes, SVM, and k-nearest neighbour

I. INTRODUCTION

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. 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. Machine learning can often be successfully applied to these problems, improving the efficiency of systems and the designs of machines .Classification is the organization of data in given classes. Also known as supervised classification, the classification uses given class labels to order the objects in the data collection. Classification approaches normally use a training set where all objects are already associated with known class labels [1].

II. CLASSIFICATION ALGORITHM'S IN DATA MINING

1. C4.5 ALGORITHM

C4.5 is an algorithm used to generate a decision tree developed by Ross Quinlan. C4.5 is an extension of Quinlan's earlier 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. I address this difficulty by borrowing from the C4.5 algorithm, an ID3 extension.ID3's sensitivity to features with large numbers of values is illustrated by Social Security numbers. Since 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. 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 anything new. 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. It 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 method to evaluate the splitting attribute. 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 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 chosen to make the decision. The C4.5 algorithm then recourses on the smaller sub lists.

Pseudo Code:

1. Check for base cases.
2. For each attribute a calculate:
   i. Normalized information gain from splitting on attribute
3. Select the best a, 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 of a and add those nodes as children node

2. 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 calculates the entropy (or information gain **IG(A)**) of that attribute. 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 labelled 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 labelled 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 labelled 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)
  Create a root node for the tree
  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 = -.
  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,
      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
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 subtree ID3 (Examples (V_i), Target Attribute, Attributes – {A})
End

3. 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 modelling the conditional probability distribution ***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 ***argmaxc P(C = c|D = d)***. A Bayesian approach splits this posterior distribution into a prior distribution ***P(C)*** and a likelihood ***P(D|C):P(D = d|C = c)P(C = c)***

argmaxc P(C = c|D = d) = argmaxc

$$p\left(D = \frac{d}{c} = c\right) p(C = c) \quad \longrightarrow \textbf{(1)}$$

The denominator ***P(D = d)*** is a normalising 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 ***P(D = d|C = c),*** the likelihood of the given description given the 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 ***a1. . . an*** for a fixed set of attributes ***A1, . . . ,An.*** Determining ***P(D = d|C = c)*** here requires an estimate of the joint probability ***P(A1 = a1, . . . ,An = an|C = c)***, abbreviated to ***P(a1, . . . ,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:

$$P\,(A1 = a1, \ldots, An = an|C = c) = \Pi^{n}_{i=1} P(Ai = ai|C = c) \quad \longrightarrow \quad \textbf{(2)}$$

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. [4], [5].

4. SUPPORT VECTOR MACHINES (SVM)

SVMs introduced in COLT-92 by Boser, Guyon & Vapnik. Became rather popular since. Theoretically well motivated algorithm developed from Statistical Learning Theory (Vapnik & Chervonenkis) 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 Φ (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

5. 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 neighbour (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 makes use of the more than one nearest neighbour 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 neighbour 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 ← SelectClass (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 neighbour. K-NN requires an integer k, a training data set and a metric to measure closeness.
.

III. ADVANTAGES AND DISADVANTAGES OF CLASSICATION ALGORITHM

S.NO	ALGORITHMS	ADVANTAGES	DISADVANTAGES
1	C4.5 Algorithm	1. It produces the accurate result. 2. It takes the less memory to large program execution. 3. It takes less model build time. 4. It has short searching time.	1. Empty branches. 2. Insignificant branches. 3. Over fitting.
2	ID3	1. It produces the more accuracy result than the C4.5 algorithm. 2. ID3 algorithm generally uses nominal attributes for classification with no missing values. 3.It produces false alarm rate and omission rate decreased, increasing the detection rate and reducing the space Consumption	1 It has long searching time. 2. It takes the more memory than the C4.5 to large program execution.
3	Naive bayes Algorithm	1.To improves the classification performance by removing the irrelevant features. 2. Good performance. 3.it is short computational time	1. The naive Bayes classifier requires a very large number of records to obtain good results. 2. it is instance-based or lazy in that they store all of the training samples
4	Support vector machine Algorithm	1. Produce very accurate classifiers. 2. Less over fitting, robust to noise. 3. Especially popular in text classification problems where very high-dimensional spaces are the norm. 4. Memory-intensive.	1. SVM is a binary classifier. To do a multi-class classification, pair-wise classifications can be used (one class against all others, for all classes). 2. Computationally expensive, thus runs slow.
5	k-nearest neighbour Algorithm	1. It is an easy to understand and easy to implement classification technique. 2. Training is very fast. 3. Robust to noisy training data. 4.It is particularly well suited for multi-modal classes	1. It is sensitive to the local structure of the data. 2. Memory limitation. 3. Being a supervised learning lazy Algorithm i.e., runs slowly.

IV. 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. Classification methods are typically strong in modelling interactions 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 data of patients, financial data according to performances. 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 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 the natural disasters like cloud bursting, earth quake, etc

REFERENCES

[1] Delveen Luqman Abd Al.Nabi, Shereen Shukri Ahmed, "Survey on Classification Algorithms for Data Mining: (Comparison and Evaluation)" (ISSN 2222-2863)
Vol.4, No.8, 2013
[2] Vidhya.K. G.Aghila. "A Survey of Naïve Bayes Machine Learning approach in Text Document Classification",(IJIST) Vol. 7, No. 2, 2010.
[3] Nitin Bhatia, Vandana," Survey of Nearest Neighbor Techniques" (IJCSIS) Vol. 8, No. 2, 2010, ISSN 1947-5500.
[4] Riaan Smit" An Overview of Support Vector Machines, 30 March 2011.
[5] B. Kotsiantis · I. D. Zaharakis · P. E. Pintelas, "Machine learning: a review of classification and combining techniques", Springer Science10 November 2007
[6] Ashis Pradhan., "Support Vector Machines a survey, ISSN 2250-2459, Volume 2, Issue 8, August 2012
[7] Thair Nu Phyu," Survey of Classification Techniques in Data Mining", IMECS 2009, March 18 - 20, 2009.
[8] H. Bhavsar, A. Ganatra,"Variations of Support Vector Machine Classification: A survey", International Journal of Advanced Computer Research, Volume 2, Number 4, Issue 6 (2012) 230–236.
[9] 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
[10] Raj Kumar, Dr. Rajesh Verma," Classification Algorithms for Data Mining P: A Survey" IJIET Vol. 1 Issue August 2012, ISSN: 2319 – 1058.

