ISSN: 2277-9655
Impact Factor: 1.852

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Analysis of Classification Techniques in Data Mining

K.S.Thirunavukkarasu*1, Dr. S.Sugumaran2
*1 Ph.D Research Scholar, Manonmaniam Sundaranar University,Assistant Professor, Department of computer science, Nehru Memorial College, Trichy, TamilNadu, India
²Associate Professor, Department of computer science,Erode Arts and Science College, Erode, TamilNadu, India
thirukst@gmail.com

Abstract

Data mining is a part of knowledge discovery process and information industry due to the vast availability of large amounts of data. The data mining is one comprehensive application of technology item relying on the statistical analysis, artificial intelligence and it has shown great commercial value and gradually to other profession penetration in the retail, insurance telecommunication, power industries use. Data mining technique usually fall into two categories Predictive and Descriptive. Predictive mining predict the trends and properties of unknown data based on the known data. Descriptive mining describes concepts or task relevant data sets in concise, summarative, informative and discriminative forms The objective of this paper is to analyze various Classification algorithms in Data mining. The Classification algorithm includes KNN, Decision tree, Naïve Bayes and Neural Network. The algorithms performances are analyzed with various dimensions.

Keywords: KNN, Naïve Bayes, Decision tree, Neural Network, data mining, review, classification

Introduction

Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases. The complete aim of data mining procedure is to gain information from large data set and transform it into an acceptable structure for future use. The most important Techniques in data mining include Classification, Clustering, and Association rule mining. Classification arises frequently from bioinformatics such as disease classifications using high throughput data like micorarrays. Classification rule mining classifies data in constructing a model based on the training set and the values or class labels in a classifying attribute and uses it in classifying new data. Currently, a various modeling techniques are detailed for data mining. The details of data mining and machine knowledge in related and network domains are dependent and comparatively distributed. The technique particularly achieves the statistical belief among occurrences in order to enhance classification accuracy. An attention on dependencies is made where the ability to draw classification accuracy is affected in improving performance of the model.

Classification is a data mining task that maps the data into predefined groups and classes. It is also called as supervised learning. It consists of two steps:
1. **Model construction**: It consists of set of predetermined classes. Each tuple/sample is assumed to belong to a predefined class. The set of tuple used for model construction is training set. The model is represented as a classification rules, decision trees or mathematical formulae.
2. **Model usage**: This model is used for classifying future or unknown objects. The known label of test sample is compared with the classified result from the model. Accuracy rate is the percentage of test set samples that are correctly classified by the model. Test set is independent of training set. If the accuracy is acceptable then use the model to classify data tuples whose class labels are unknown [4].

Organization of the Paper

The framework of this paper includes the following sections: .Our next section deals with a study on K-Nearest Neighbor (KNN), section III describes Decision tree, section IV with Naïve Bayes. Neural network is described in section V. Finally the last section concludes the work.

KNN

The kNN (IBk) locates the *k* nearest instances to the query instance and determines its class by identifying the single most frequent class label. A pseudo-code example for the instance base learning methods is illustrated.

```
procedure InstanceBaseLearner(Testing
Instances)
for each testing instance
{
find the k most nearest instances of
the training set according to a
distance metric
Resulting Class= most frequent class
label of the k nearest instances
} [2]
```

The closeness is defined in terms of distance metric, among which Eculidiean distance is calculated with two tuples say

$X1 = (x_{11,} x_{12,} \dots x_{1n})$ and
$X2 = (x_{21,} x_{22,} \dots x_{2n})$

$$Dist(X1, X2) = \sqrt{\sum_{i=1}^{k} (\boldsymbol{x_{1i} - x_{2i}})^{\,2}}$$

We normalize the values of the attributes before using it in the Euclidean equation.

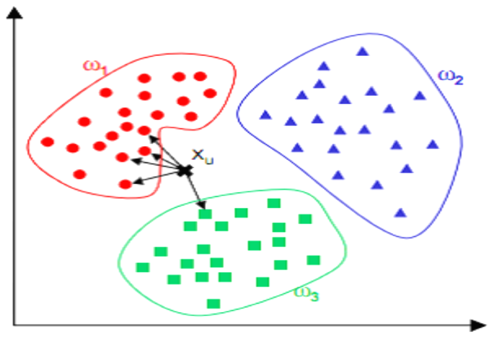

Fig. 2.1 Example of K-NN classification

The KNN has some limitations [18].

1. High calculation complexity: To find out the k nearest neighbor samples, all the similarities between the training samples must be calculated. When the number of training samples is less, the KNN classifier is no longer optimal, but if the training set contains a huge number of samples, classifier needs more time to calculate the similarities.

2. No weight difference between samples: All the training samples are treated equally; there is no difference between the samples with small number of data and huge number of data.

3. Dependency on the training set: The classifier is generated only with the training samples and it does not use any additional data. This makes the algorithm to depend on the training set excessively; it needs recalculation even if there is a small change on the training set.

Neural Network

The neural network model can be broadly divided into the following three types:

(a) Feed-forward networks: It regards the perception back-propagation model and the function network as representatives, and mainly used in the areas such as prediction and pattern recognition;

(b) Feedback network: It regards Hopfield discrete model and continuous model as representatives, and mainly used for associative memory and optimization calculation;

(c) Self-organization networks*:* it regards adaptive resonance theory (ART) model and Kohonen model as representatives, and mainly used for cluster analysis.

In the most common family of feed-forward networks called multilayer preceptor, neurons are organized into layers that have undirected connection between them. Different connectivity yield different networks behaviors.

Feed-forward networks are static that is, they produce only one set of output values rather than a sequence of values from a given input. Feed forward networks are memory- less in the sense that their response to an input is independent of the previous network state. Recurrent or feedback networks on the other hand, are dynamic systems. When a new input pattern is presented, the neuron outputs are computed. Because of the feedback paths, the input to each neurons are them modified, which leads the network to enter a new state.

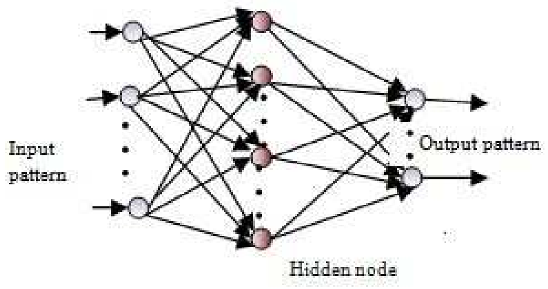

Fig 3.1 Artificial Neural Network

The back propagation algorithm performs learning on a multilayer feed-forward neural network. A multilayer feed forward neural network consists of an input layer, one or more hidden layers and an output layer. Back propagation learns by iteratively processing a data set of training tuples, comparing the networks prediction for each tuple with the actual target value. For each training tuple, the weights are modified so as to minimize the mean squared error between the network's prediction and the actual target value .These modifications are done in backward direction.

```
1. Initialize the weights in the network (often randomly)
2. Repeat
* For each example e in the training set do
1. O=neural –net-output (network   e);
Forward pass
2. T = teacher output for e
3. Calculate error (T   - O) at the output units
4. Compute delta _wi for all weights from hidden layer to output layer;
Backward pass
5. Compute delta_wi for all weights from input layer to output layer;
Backward pass continued
6. Update the weights in the network
* End
3. until all examples classified correctly or stopping criterion satisfied
4. Return (network)
```

The idea of the back propagation algorithm is to reduce this error, until the ANN *learns* the training data. The use of neural networks data mining is a promising field of research especially given the ready availability of large mass of data sets and the reported ability of neural networks to detect and assimilate relationship between a large numbers of variables [17].

Decision Tree

Decision tree is tree data structures that represent sets of decisions at foliage nodes. This data structure gives a set of rules for the classification of a training dataset. Decision tree is one of the most frequently used data mining approach because of its transparency. In decision tree technique, the root of the decision tree is a simple question or condition that has multiple answers. Each answer then leadings to a set of questions or conditions that help us to determine the data so that we can make the final decision based on it. Final result is a decision tree in which each branch represents a possible scenario of decision and its outcome.

To build decision tree, row to column and column to row relationships are established. Then all possible outcome instances are tested to check whether they are falling under the same class or not. If all the cases are falling under the same class, the node is delineated with single class name, differently choose the splitting attribute to classify the instances. Decision Tree is a classifier which has the form similar to that of a tree and has the following structure elements:

- Root node: Left-most node in a decision tree
- Decision node: Specifies a test on a single attribute
- Leaf node: Indicates the value of target attribute
- Edge: Split of an attribute
- End-point: Right most node representing final outcome

DT is constructed using divide and conquer (D&C) approach [5]. Each path in DT determines a decision rule. Usually it follows a greedy approach from top to bottom ie; from root node to the ending node recursively for determining the final outcome and hence can deal with uncertainties. D&C strategy approaches a problem in the following manner:

- Breaking the problem into different sub-problems which are the instances of the given problem
- Recursively solving each of these problems
- Finally combining each answers of these sub-problems into a single one

Decision Tree can be considered as more interpretable compared to that of neural networks and support vector machines since they combines more data in an easily understandable format. Even small changes in the input data may lead to great variations in constructing the DT. In some cases it has to deal with uncertainties. This can be solved using sequential decision making of DT. The process of determining the expected values from the end node back to the root node is known as decision tree roll-back

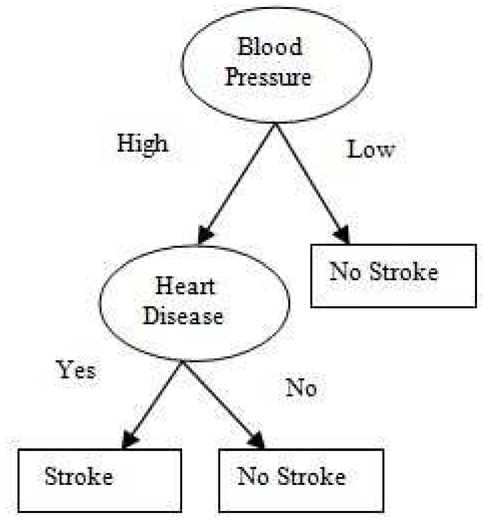

Fig 4.1 Example for Decision Tree

The basic algorithm for decision tree induction is a greedy(i.e,non-backtracking) approach that constructs decision trees in a top-down recursive divide and conquer manner[8].Induction is an entity that accepts the training set and forms a classifier that represents the generalize relationship between the input and the target attribute[8].

```
1. Create a node N;

2. If tuples in D are all of the same class then

3. Return N as a leaf node labeled with the class C;

4. If attribute_list is empty then

5. Return N as leaf node labeled with majority class in D; //
   majority voting

6. Apply attribute_selection_method (D, attribute_list) to
   find the "best splitting_criterion:

7. Label node N with splitting_criterion;

8. If splitting_attribute is discrete-valued and Multiway
  splits allowed then //not restricted to binary trees

9.Attribute_list<--attribute_list-splitting_attribute//remove
  splitting attribute

10. For each outcome j of splitting-criterion//partition the
    tuples and grow sub trees for each partition

11. let Dj be the set of data tuples in D satisfying
        outcome J; //a partition

12. If Dj is empty then

13. Attach a leaf labeled with the majority class in D to
     node N;

14. Else attach the node returned by generate
        _decision_tree(Dj,attribute _list) to node N;

End for

15. Return N;
```

Fig 2.1 Basic algorithm for including a decision tree from the training tuple

The majority step of algorithm is attribute_selection_methode to select the most appropriate attribute for splitting the node. Greatest amount of uncertainty must be reduced with the chosen splitter.The three broadly used decision tree learning algorithms are: ID3, and C4.5, CART.

CART- CART stands for Classification and Regression trees, introduced by Breiman. It is based on Hunt's algorithm. CART addresses both continuous and categorical attributes to build a decision tree. CART handles missing values. Gini Indexing is used in CART as an attribute selection measure to build a decision tree. Dissimilar to ID3 and C4.5 algorithms, CART produces binary splits. Therefore, it produces binary trees. Gini Index measurement does not use probabilistic assumptions like ID3, C4.5. CART uses cost complexness pruning to remove the unreliable branches from the decision tree to improve the accuracy.

ID3 (iterative dichotomiser) - This is a decision tree algorithm introduced in 1986 by Quinlan Ross. ID3 is based on Hunts algorithm. The tree can be built in two stages. The two stages are tree building and pruning. Basic idea of ID3 Algorithm is to construct the decision tree by applying a top-down, greedy search through the given sets to test each attribute at every tree node.ID3 uses information gain measure to select the splitting attribute. It merely accepts categorical attributes in building a tree model. It does not give exact outcome when there is noise. To dispatch the noise pre-processing technique has to be used. Continuous attributes can be handled using the ID3 algorithm by discrediting or directly, by considering the values to find the best split point by taking a threshold on the attribute values. ID3 does not support pruning by default it can be applied after building data model.

C4.5 and C5.0 – C4.5 and C5.0 both algorithms are successor of ID3, developed by Quinlan Ross. It is based on Hunt's algorithm. C4.5 handles both continuous and categorical attributes to construct a decision tree. In order to address continuous attributes, C4.5 separates the attribute values into two partitions based on the selected threshold such that all the values above the threshold as one child and the remaining as another child. Both C4.5 and C5.0 also handles missing attribute values. C4.5 usages Gain Ratio as an attribute selection measure to develop a decision tree. It withdraws the biasness of information gain when there are many outcome values of an attribute.

At first, calculate the gain ratio of each attribute. Gain ratio of the root node will be maximal. C4.5 uses

pessimistic pruning to remove unnecessary branches in the decision tree to improve the accuracy of classification. The main attribute of data mining is that it subsumes Knowledge Discovery (KD) is a nontrivial process of identifying legal, novel, possibly useful and ultimately understandable patterns in data processes, thereby contributing to predicting trends of outcomes by profiling performance attributes that supports effective decisions making.
C4.5 selects the test that maximizes gain ratio value.

The difference between ID3 and C4.5 algorithm is that C4.5 algorithm uses multi-way splits, whereas ID3 uses binary splits. In order to reduce the size of the decision tree, C4.5 uses post-pruning technique; whereas an optimizer combines the generated rules to eliminate redundancies. The improved version of C4.5 is C5.0, which includes cross-validation and boosting capabilities.

Both C4.5 and C5.0 can produce classifiers expressed either as decision trees or rule sets. In many applications, rule sets are preferred because they are simpler and easier to understand than decision trees, but C4.5's rule set methods are slow and memory- hungrier. C5.0 embodies new algorithms for generating rule sets, and the improvement is substantial. In C4.5, all errors are treated as adequate, but in practical applications some classification errors are more serious than others. C5.0 allows a separate cost to be determined for each predicted/actual class pair; if this option is used, C5.0 then builds classifiers to minimize expected miss classification costs rather than error rates.

Naïve Bayes

A Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem with strong independence assumption. It works as follows

1. Let D be a training set of tuples and their associated class labels. As usual, each tuple is represented by an n-dimensional attribute vector, X=(x1, x2,…, xn), depicting n measurements made on the tuple from n attributes, respectively, A1, A2,.., An.

2. Suppose that there are m classes, C1, C2… Cm. Given a tuple, X, the classifier will predict that X belongs to the class having the highest posterior probability, conditioned on X. That is, the naïve Bayesian classifier predicts that tuple x belongs to the class Ci if and only if

$$P\left({}^{C_i}/_X\right) > P\left({}^{C_j}/_X\right)\ j \le m, j \ne i$$

Thus we maximize P(Ci|X). The class Ci for which P(Ci|X) is maximized is called the maximum posteriori hypothesis. By Bayes' theorem

$$P\left({}^{C_i}/_X\right) = \left(\frac{P(X|C_i)P(C_i)}{P(X)}\right)$$

3. As P(X) is constant for all classes, only P (X|Ci) P (Ci) need be maximized. If the class prior probabilities are not known, then it is commonly assumed that the classes are equally likely, that is, P(C1)=P(C2)=…=P(Cm), and we would therefore maximize P(X|Ci). Otherwise, we maximize $P(X|C_i)P(C_i)$. Note that the class prior probabilities may be estimated by P(Ci)=|Ci,D|/|D|**,** where |Ci,D| is the number of training tuples of class Ci in D.

4. Given data sets with many attributes, it would be extremely computationally expensive to compute P(X|Ci).In order to reduce computation in evaluating P(X|Ci), the naïve assumption of class conditional independence is made. This presumes that the values of the attributes are conditionally independent of one another, given the class label of the tuple (i.e., that there are no dependence relationships among the attributes). Thus,

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i) = P(x_1|C_1) * P(x_2|C_2) \ldots P(x_n|C_n)$$

We can easily estimate the probabilities P(x1|Ci), P(x2|Ci),… ,P(xm|Ci) from the training tuples. Recall that here xk refers to the value of attribute Ak for tuple X. For each attribute, we look at whether the attribute is categorical or continuous-valued. For instance, to compute P(X|Ci), we consider the following:

(a) If Ak is categorical, then P(Xk|Ci) is the number of tuples of class Ci in D having the value xk for Ak,divided by |Ci,D|**,** the number of tuples of class Ci in D.
(b) If Ak is continuous valued, then we need to do a bit more work, but the calculation is pretty straightforward. A continuous-valued attribute is typically assumed to have a Gaussian distribution with a mean μ and standard deviation σ, defined by

$$g(x,\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$
$$p(x_k/c_i) = g(x_{k,},\mu_{c_i},\sigma_{c_i})$$

We need to compute µci and σci, which are the mean and standard deviation, of the values of attribute Ak for training tuples of class Ci. We then plug these two quantities into the above equation.

5. In order to predict the class label of X, P(X|Ci)P(Ci) is evaluated for each class Ci. The classifier predicts that the class label of tuple X is the class Ci if and only if

$$P(X/C_i)P(C_i) > P(X/C_j)P(C_j) \ for\ 1 \leq j \leq m, j \neq i.$$

In other words, the predicted class label is the class Ci for which $P(X/C_i)P(C_i)$ is the maximum.[1]

TABLE 1
COMPARISON OF CLASSIFICATION TECHNIQUES
(**** represents the best and * represents the worst performance)

Dimension	**KNN**	**Neural Network**	**Decision tree**	**Naïve Bayes**
Accuracy in general	**	***	**	*
Speed of learning with respect to number of attributes and the number of instances	****	*	***	****
Speed of classification	*	****	****	****
Tolerance to missing values	*	*	***	****
Tolerance to irrelevant attributes	**	*	***	**
Tolerance to redundant attributes	**	**	**	*
Tolerance to highly interdependent attributes	*	***	***	*
Dealing with discrete/binary/ continuous attributes	***(not directly discrete)	***(not discrete)	****	***(not continuous)
Tolerance to noise	*	**	**	***
Dealing with danger of overfitting	***	*	**	***

The performance chart depicts the suitability of algorithms in various parameters.

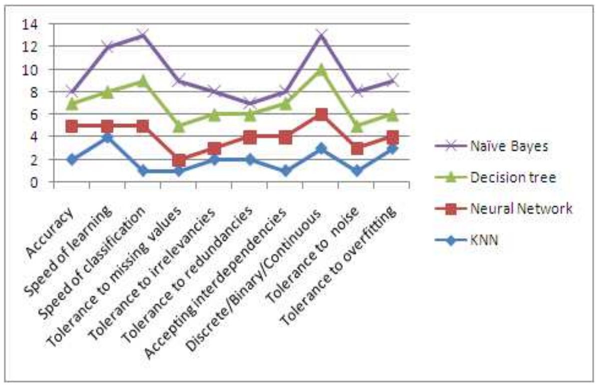

Fig 5.1 Performance chart

Conclusion

This paper deals with various classification techniques used in data mining and a study on each of them. 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. Decision tree and Naïve Bayes generally have different operational profiles, when one is very accurate the other is not and vice versa. Classification methods are typically strong in modeling interactions.

References

[1] Jiawei Han, Micheline Kamber,‖ DataMining:Concepts and Techniques, Second Edition, ISBN 13: 978-1-55860-901-3, Elsevier, 2006.

[2] H. Witten, E. Frank, and M. A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, 3rd ed. Morgan Kaufmann, 2011

[3] Rohit Arora, Suman" Comparative Analysis of Classification Algorithms on Different Datasets using WEKA , *International Journal of Computer Applications (0975 – 8887)Volume 54– No.13, September 2012*

[4] Dharminder Kumar, Suman" Performance Analysis of Various Data MiningAlgorithms: A Review, *International Journal of Computer Applications (0975 – 8887) Volume 32– No.6, October 2011*

[5] N.Sugunaand Dr.Thanuskodi" An improved k-Nearest neighbor Classification Using Genetic

Algorithm "IJCSI International Journal of Computer Science Issues,Vol.7,Issue 4,No 2,July 2010 ISSN (online) : 1694-0784 ISSN(Print):1694-0814 18

[6] Ansul Goyal and Rajni Mehta, assistant Professor, CSE"Performance Comparison of Naïve Bayes and J48 Classification Algorithms "International Journal of Applied Engineering Research, ISSN 0973-4562 Vol.7 No.11 (2012)

[7] Nitu Mathuriya, Dr.Ashish Bansal "Comparison of K-Means and Backprobagation Data Mining Algorithms "International Journal of Computer Technology and Electronics Engineering (IJCTEE) Volume 2, Issue 2

[8] S.B.Kotsiantis Department of Computer Science and Technology University of Peloponnese, Greece "Supervised Machine Learning: A Review of Classification Techniques Informatics 31(2007) 249-268

[9] B. Mehala, P. Ranjit Jeba Thangaiah, and K. Vivekanandan "Selecting Scalable Algorithms to Deal ith Missing Values" International Journal of Recent Trends in Engineering, Vol. 1, No. 2, May 2009

[9] G.K.Gupta "Introduction to Data mining

[10] Aftab Ali Haider and Sohail Asghar" A survey of logic based Classifiers "International Journal of Future Computer and Communication, Vol 2, No.2, April 2013

[11]J.R .Quinlan,"Simplifying decision trees,"International Journal of Man-Machine Studies, Vol.27, No.3, 987, pp.221-234.

[12] Dipak V.Patil, R.S.Bichkar"Issues in Optimization Tree Learning: A survey "International Journal of Applied Information Systems (IJAIS)-ISSN: 2249-0868 Foundation of Computer Science FCS, NewnYork, USA Volume 3-No.5, July 2012 –www.ijais.org

[13] J.R.Quinlan,Decision Trees and multivalued attributes,J.Richards,ed.,Machine Intelligence,V.11,Oxford ,England,Oxford University .Press ,pp.305-318,1988.

[14] Floriana Esposito,Donato Malerba and Giovanni Semeraro (1997).A comparative analysis of methods for pruning decision trees.*IEEE Transactions On Patterns Analysis And Machine Intelligence* ,Vol.19,No.5,pp.476-491.

[15]T.Windeatt and G.Ardeshir (2001).An empirical comparison of pruning methods for ensamples classifiers .*Proc.of 4th International Conference on Advances in Intelligent Data Analysis*, Cascasis, Portugal, pp.208-217.

[16] Z.Haying,"A short Introduction to data mining and its applications", IEEE, 2011

[17] Rohit Arora, Suman, " Comparative Analysis of Classification Algorithms on Different Datasets using WEKA

[18] K.Gayathri , M.chitra " Performance Evaluation of Classification Techniques in Data Mining" International Journal of Computer Science Research & Technology (IJCSRT) ISSN: 2321-8827, Vol. 1 Issue 4, September – 2013

[19] Sunitha B Aher, Mr. LOBO L.M.R.J." Data mining in educational Systems using WEKA"

[20] Lior Rokach (*Ben-Gurian University*, *Israel*), Oded Maimon (*Tel-Aviv University*, *Israel*)"Data Mining with decision Trees Theory and applications"

[21]Thair Nu Phyu,proceedings of the Internationalulticonference of Engineers and Computer Scientists 2009 Vol I,IMECS 2009,March18-20,2009 Hong Kong"Survey of Classification Techniques in Data Mining"