COMPARISON OF CLASSIFICATION TECHNIQUES

Savika Bishnoi*

1. 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. **Data mining** is the process of extracting patterns from data. Data mining is seen as an increasingly important tool by modern business to transform data into an informational advantage. It is currently used in a wide range of profiling practices, such as marketing, surveillance, fraud detection, and scientific discovery.

Classification technique is capable of processing a wider variety of data than regression and is growing in popularity. Classification analysis 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. The classification algorithm learns from the training set and builds a model. The model is used to classify new objects. Classification is a supervised machine learning procedure in which individual items are placed into groups based on quantitative information on one or more characteristics inherent in the items (referred to as traits, variables, characters, etc) and based on a training set of previously labeled items.

Supervised learning (classification)

- Supervision: The training data (observations, measurements, etc.) are accompanied by labels indicating the class of the observations
- New data is classified based on the training set

*Lecturer

Classification is a data mining technique used to map a data item into one of several predefined classes. In this task the goal is to predict the value of a user-specified goal attribute based on the values of other attributes, called the predicting attributes. For example, you may wish to use classification to predict whether the weather on a particular day will be “sunny”, “rainy” or “cloudy”. Popular Classification predicts categorical class labels (discrete or nominal). Classifies data (constructs a model) based on the training set and the values (class labels) in a classifying attribute and uses it in classifying new data.

We test these three algorithms on six datasets. Experimental results show that the k-nearest neighbor algorithm gets better performance than the other two algorithms in classification. The rest of this paper is organized as follows. Section 2 Introduces the Decision tree algorithm. Section 3 describes the kNN algorithm. Section 4 describes the Bayesian networks algorithm Section 5 reports experimental results of comparison of algorithm. Finally, Section 5 concludes this paper and gives future research directions In the present paper, we have concentrated on the techniques necessary to do this. The objective of this research work is to introduce such classification techniques which reveal the importance or significance of items. Plenty of work done by many researchers in this field, but our work is different from them in the sense that, we have assigned different techniques using weka software. Mainly, we use different classification techniques for comparison. The scope of this work is modest: to provide an introduction to classification analysis in the field of data mining, where we define data mining to be the discovery of useful, but non-obvious, information or patterns in large collections of data. Much of this work is necessarily consumed with providing a general background for classification analysis, but we also discuss a number of classification techniques that have recently been developed specifically for data mining. .

2. DECISION TREE INDUCTION

The Decision Tree algorithm is based on conditional probabilities. Decision trees generate **rules**. A rule is a conditional statement that can easily be understood by humans and easily used within a database to identify a set of records.

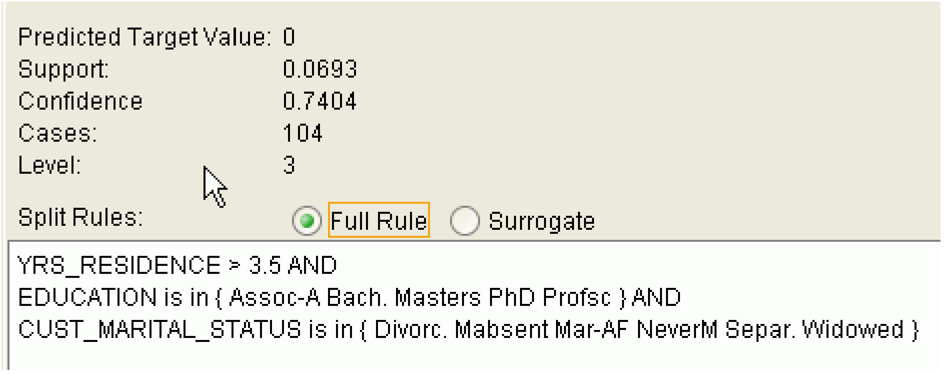
In some applications of data mining, the accuracy of a prediction is the only thing that really matters. It may not be important to know how the model works. In others, the ability to explain the reason for a decision can be crucial. For example, a Marketing professional would need complete descriptions of customer segments in order to launch a successful marketing campaign. The Decision Tree algorithm is ideal for this type of application.

Decision tree learning is a common method used in data mining. The goal is to create a model that predicts the value of a target variable based on several input variables. Each interior node corresponds to one of the input variables; there are edges to children for each of the possible values of that input variable. Each leaf represents a value of the target variable given the values of the input variables represented by the path from the root to the leaf.

Decision Tree Rules provide model transparency, a window on the inner workings of the model. Rules show the basis for the model's predictions. Data Mining supports a high level of model transparency. While some algorithms provide rules, *all* algorithms provide model details**.**

Figure 1-1 shows a rule generated by a Decision Tree model. This rule comes from a decision tree that predicts the probability that customers will increase spending if given a loyalty card. A target value of 0 means not likely to increase spending; 1 means likely to increase spending.

Figure 1-1 Sample Decision Tree Rule



```
Predicted Target Value: 0
Support:                0.0693
Confidence              0.7404
Cases:                  104
Level:                  3
Split Rules:            ● Full Rule  ○ Surrogate
YRS_RESIDENCE > 3.5 AND
EDUCATION is in { Assoc-A Bach. Masters PhD Profsc } AND
CUST_MARITAL_STATUS is in { Divorc. Mabsent Mar-AF NeverM Separ. Widowed }
```

Description of "Figure 1-1 Sample Decision Tree Rule"

The rule shown in Figure 1-1 represents the conditional statement:

IF

(Current residence > 3.5 and has college degree and is single)

THEN

Predicted target value = 0

This rule is a full rule. A surrogate rule is a related attribute that can be used at apply time if the attribute needed for the split is missing.

Confidence and Support

Confidence and support are properties of rules. These statistical measures can be used to rank the rules and hence the predictions.

Support: The number of records in the training data set that satisfies the rule.

Confidence: The likelihood of the predicted outcome, given that the rule has been satisfied.

For example, consider a list of 1000 customers (1000 cases). Out of all the customers, 100 satisfy a given rule. Of these 100, 75 are likely to increase spending, and 25 are not likely to increase spending. The **support of the rule** is 100/1000 (10%). The **confidence of the prediction** (likely to increase spending) for the cases that satisfy the rule is 75/100 (75%).

3. K-NEAREST NEIGHBOR CLASSIFER

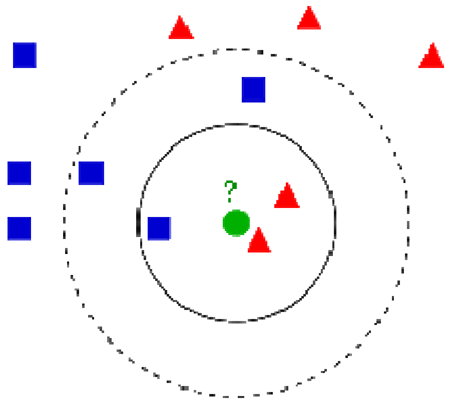
Another category under the header of statistical methods is Instance-based learning. Instance-based learning algorithms are lazy-learning algorithms (Mitchell, 1997), as they delay the induction or generalization process until classification is performed. Lazy-learning algorithms require less computation time during the training phase than Eager-learning algorithms (such as decision trees, neural and Bayes nets) one of the most straightforward Instance-based learning algorithms is the nearest neighbor algorithm. Aha (1997) and De Mantaras and Armengol (1998) presented a review of instance-based learning classifiers. In pattern recognition, the ***k*-nearest neighbour's algorithm** (*k*-NN) is a method for classifying objects based on closest training examples in thefeature space. *K-NN* is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. The *k*-nearest neighbour algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbours, with the object being assigned to the class most common amongst its *k* nearest neighbours (*k* is a positive integer, typically small). If *k* = 1, then the object is simply assigned to the class of its nearest neighbour.

The neighbours are taken from a set of objects for which the correct classification (or, in the case of regression, the value of the property) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required. The *k*-nearest neighbour algorithm is sensitive to the local structure of the data.

The *k*-Nearest Neighbor Algorithm

- All instances correspond to points in the n-D space
- The nearest neighbor are defined in terms of Euclidean distance, dist(**X1**, **X2**)
- Target function could be discrete- or real- valued
- For discrete-valued, *k*-NN returns the most common value among the *k* training examples nearest to *xq*

Algorithm example *Figure 1-2*



Example from figure 1-2 of k-NN classification. The test sample (green circle) should be classified either to the first class of blue squares or to the second class of red triangles. If k = 3 it is classified to the second class because there are 2 triangles and only 1 square inside the inner circle. If k = 5 it is classified to first class (3 squares vs. 2 triangles inside the outer circle).

The training examples are vectors in a multidimensional feature space, each with a class label. The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples.

4. BAYESIAN NETWORKS

Naive Bayesian (NB) Algorithm has been widely used for document classification, and shown to produce very good performance. The basic idea is to use the joint probabilities of words and categories to estimate the probabilities of categories given a document. NB algorithm computes the posterior probability that the document belongs to different classes and assigns it to the class with the highest posterior probability. The posterior probability of class is computed using Bayes rule and the testing sample is assigned to the class with the highest posterior probability. The naive part of NB algorithm is the assumption of word independence that the conditional probability of a word given a category is assumed to be independent from the conditional probabilities of other words given that category. There are two versions of NB algorithm. One is the multi-variate Bernoulli event model that only takes into account the presence or absence of a particular term, so it doesn't capture the number of occurrence of each word. The other model is the multinomial model that captures the word frequency information in documents

- Foundation: Based on Bayes' Theorem.
- Performance: A simple Bayesian classifier, *naïve Bayesian classifier*, has comparable performance with decision tree and selected neural network classifiers
- Incremental: Each training example can incrementally increase/decrease the probability that a hypothesis is correct — prior knowledge can be combined with observed data
- Standard: Even when Bayesian methods are computationally intractable, they can provide a standard of optimal decision making against which other methods can be measured

Bayesian Theorem: Basics

- Let **X** be a data sample ("*evidence*"): class label is unknown
- Let H be a *hypothesis* that X belongs to class C
- Classification is to determine P(H|**X**), (*posteriori probability),* the probability that the hypothesis holds given the observed data sample **X**
- P(H) (*prior probability*), the initial probability
- E.g., **X** will buy computer, regardless of age, income, …
- P(**X**): probability that sample data is observed
- P(**X**|H) (likelyhood), the probability of observing the sample **X**, given that the hypothesis holds
- E.g., Given that **X** will buy computer, the prob. that X is 31..40, medium income

Bayesian Theorem

- Given training data **X**, *posteriori probability of a hypothesis* H, P(H|**X**), follows the Bayes theorem

- Informally, this can be written as

  Posteriori = likelihood x prior/evidence

- Predicts **X** belongs to C2 if the probability P(Ci|**X**) is the highest among all the P(Ck|X) for all the *k* classes
- Practical difficulty: require initial knowledge of many probabilities, significant computational cost

Naïve Bayesian Classifier

- Let D be a training set of tuples and their associated class labels, and each tuple is represented by an n-D attribute vector **X** = (x1, x2, …, xn)
- Suppose there are *m* classes C1, C2, …, Cm.
- Classification is to derive the maximum posteriori, i.e., the maximal P(Ci|**X**)
- This can be derived from Bayes' theorem
- Since P(X) is constant for all classes, only needs to be maximized

5. EXPERIMENTS

Datasets used from UCI

We have tested 6 datasets for each algorithm. In our experiments, 6 data sets are used, available in the UCI repository website (*Http: //archive.ics.uci.edu/ml/*). For each data set, 90% of all examples were randomly selected as training examples and the rest 10% as testing ones. The detailed information of the 6 data sets is shown in TABLE I, where the data set name listed in the table is the first word of its full name.

TABLE 1: Data sets

S. No.	Data set Name	Instances	Attributes
1.	Vehicle.arff	846	19
2.	Car. ARFF	1733	7
3.	Autos.ARFF	205	26
4.	Glasss.ARFF	215	10

5.	Flags.ARFF	194	30
6.	Labor.ARFF	57	17

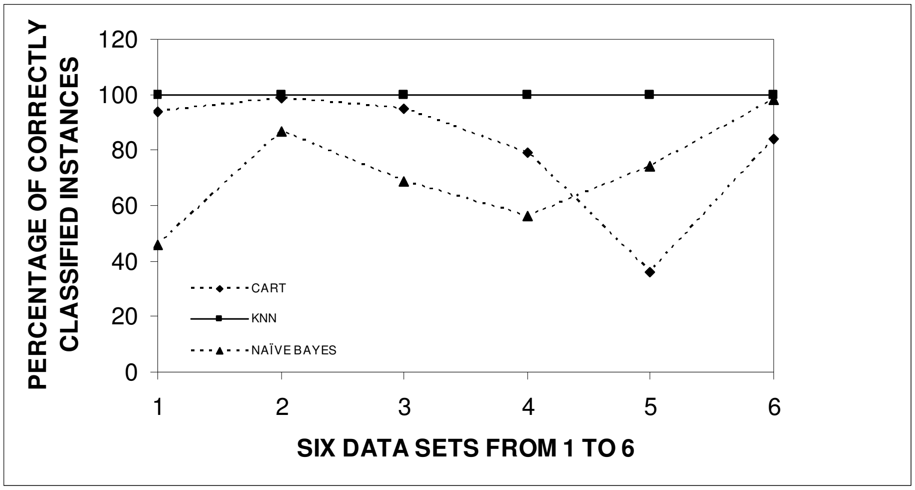
First of at all, we take six different data sets from UCI Machine Learning Repository. After that we change the format of these data into ARFF format with the help of WEKA We take three algorithms of Classification techniques- Decision tree (CART), K-nearest neighbor, Naive Bayes classifier Algorithms.

Six experiments discussed above shown that three algorithms applied on a particular dataset and range of correct and incorrect instance and time to build model. According to time and accuracy now the following table and Graph show the comparative result of all the three algorithm that applied on six data set

TABLE 2: Percentage of Correctly classified instances by algorithms on six Data sets

		Classification Algorithms		
S.No	Datasets	Decision Tree(CART)	KNN	Naive Bayes
1.	Vehicle	94	100	46
2.	Car	99	100	87
3.	Autos	95	100	69
4.	Glass	79	100	56
5.	Flags	36	100	74
6	Labour	84	100	98

Figure 1-3 Graphical representation of Percentage of Correctly classified instances



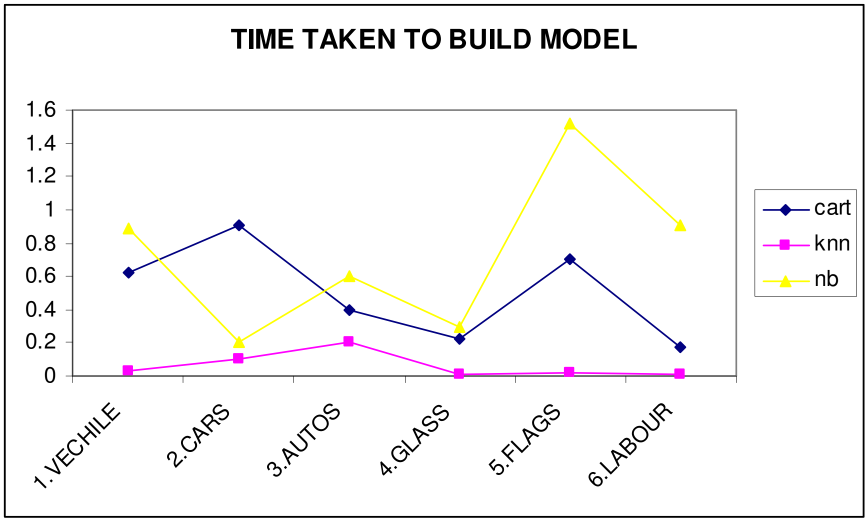
The following table and Graph show the comparative result of time taken to build model by the three algorithms

TABLE 3: Time taken to built model by algorithms on six data sets

In Seconds

Datasets	Decision Tree(CART)	KNN	Naïve Bayes
1.VECHILE	0.62	0.03	0.89
2.CARS	0.91	0.1	0.2
3.AUTOS	0.4	0.2	0.6
4.GLASS	0.22	0.01	0.3
5.FLAGS	0.7	0.02	1.52
6.LABOUR	0.17	0.01	0.91

Figure 1-4 Graphical representation of Time taken to built model by algorithms on six data sets



5. CONCLUSION

5.1 Summary of Contributions

Classification has a number of applications in every field of life. We are applying this technique whether knowingly or unknowingly in day-to-day life.

Classification is a data mining function that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data. For example, a classification model could be used to identify loan applicants as low, medium, or high credit risks.

In this research, we compare the basic classification techniques algorithms. The goal of this study is to provide a comprehensive review of different three classification techniques decision tree induction, Bayesian networks, k-nearest neighbor in data mining. In order to compare these three algorithms based on the accuracy and time parameters we come to know which one algorithm is more efficient to use The performance of the each algorithm is tested on six different data sets from UCI Machine Learning Repository based on following parameters.

- Accuracy
- Time

After run each algorithm of Classification we will get:

- The numbers of "Correctly Classified Instances" and the "Incorrectly Classified Instances”. This demonstrates the accuracy of the model.
- Other important factor time taken to build model.

Our proposed method has been implemented in Machine standard Java programming language, which is available in WEKA-3.6.4. (Machine learning data mining software) [HFH+09].Experimental results indicate that k- nearest neighbor algorithm performs better than the other two algorithms. By comparing three algorithms on six data sets in Weka. From the Weka result charts and table have been formed.

According to our research we came to know that K nearest neighbor (Star) is the best model among three algorithms based on accuracy rate and time.

REFERENCES

[1] Jiawei Han, Michelin Kamber, “Data Mining Concepts and Techniques” [M], Morgan Kaufmann publishers, USA, 2001,pp. 70-181.

[2] N. Japkowicz, “Learning from imbalanced data sets: a comparison of various strategies,” AAAI workshop on learning from imbalanced data sets, 2000, pp. 10-15.

[3] S. Tan, “Neighbor-weighted *k*-nearest neighbor for unbalanced text corpus,” Expert Systems with Applications, vol. 28, 2005, pp. 667-671.

[4] S.A. Dudani, “The distance-weighted *k*-nearest neighbor rule,” IEEE Transactions on Systems, Man and Cybernetics, vol. 6, 1976, pp. 325-327.

[5] T. Hastie and R. Tibshirani, “Discriminant adaptive nearest neighbor classification,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, 1996, pp. 607-616.

[6] K.Q. Weinberger and L.K. Saul, “Distance metric learning for large margin nearest neighbor classification,” The Journal of Machine Learning Research, vol. 10, 2009, pp. 207-244.

[7] K. Mouratidis, D. Papadias and M. Hadjieleftheriou, “Conceptual partitioning: an efficient method for continuous nearest neighbor monitoring,” Proceedings of the 2005 ACM SIGMOD international conference on Management of data, 2005, pp. 634-645.

[8] H. Zhang and J. Su, “Naive bayes for optimal ranking,” Journal of Experimental and Theoretical Artificial Intelligence,2008 vol. 20, pp. 79-93;

[9] M. Yousef, M. Nebozhyn, H. Shatkay, S. Kanterakis, L. C. Showe, and M. K. Showe, "Combining multi-species genomic data for microRNA identification using a Naive Bayes classifier," *Bioinformatics,2006,* vol. 22, no. 11, pp. 1325-1334.