Performance Analysis of Various Data Mining Algorithms: A Review

Dharminder Kumar
Professor & Dean FET
Guru Jambheshwar University of Science and Technology, Hisar-Haryana, India.

Suman
Research Scholar
Guru Jambheshwar University of Science and Technology, Hisar-Haryana, India.

ABSTRACT
Data warehouse is the essential point of data combination for business intelligence. Now days, there has been emerging trends in database to discover useful patterns and/or correlations among attributes, called data mining. This paper presents the data mining techniques like Classification, Clustering and Associations Analysis which include algorithms of Decision Tree (like C4.5), Rule set Classifier ,kNN and Naïve Bayes ,Clustering algorithms (like *k*-Means and EM)Machine Learning (Like SVM),Association Analysis(like Apriori). These algorithms are applied on data warehouse for extracting useful information. All algorithms contain their description, impact and review of algorithm. We also show the comparison between the classifiers by accuracy which shows ruleset classifier have higher accuracy when implement in weka.These algorithms useful in increasing sales and performance of industries like banking, insurance, medical etc and also detect fraud and intrusion for assistance of society.

Keywords: Decision Tree, Rule set Classifier, kNN, Naïve Bayes , *k*-Means, EM, SVM, Apriori.

1. INTRODUCTION
A data warehouse is a subject-oriented, integrated, non - volatile and time- variant collection of data in support of management's decisions [1]Processes like data mining [2] fetch the hidden predictive information from data warehouse. Data mining predicts future trends and behavior which makes businesses upbeat, knowledge-driven decisions. The most frequently used techniques in data mining are: **Clustering**: "Process of managing objects into groups whose members have similar property in some way". A *cluster* in [3] is therefore a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters. **Decision trees**: Tree-shaped structures in [2] that represent sets of decisions. These decisions are helpful in generating rules for the classification. **Naïve Bayes**: The Naive Bayes in [4] Classifier technique is based on the so-called Bayesian theorem and is particularly suited when the dimensionality of the inputs is high. Naive Bayes can often outperform more sophisticated classification methods. **Nearest neighbor method**: A technique in [2] that classifies each record in a dataset based on a combination of the classes of the k record(s) most similar to it in a historical dataset k-nearest neighbor technique. **Association Analysis**: Association analysis in [5] is useful in finding hidden relationship in large data warehouse. This hidden relationship can be representing in the form of frequent item sets or association rules. For example the Apriori Algorithm is an influential algorithm for mining frequent item sets for boolean association rules. The layout of the paper is as follows: Section (II) describes various classification Techniques. In section III clustering techniques are discussed. Section IV contains machine learning (SVM) technique then in Section V association analysis and Performance Analysis is discussed in section .VI. Section (VII) is the concluding section.

2. CLASSIFICATION
In data mining classifier is the important tool. Systems that construct classifiers as in [6] get input as a set of classes, each class consist of small number of classes and having fixed set of attribute values and give output which predict the class that belongs with a new class .

2.1 Decision tree (C4.5 and beyond)
Set of M of tuples, using divide-and-conquer method C4.5 first make an initial tree as follows in Figure1. :

- Create a node N.
- If tuples in M belong to the same class say C, then return N as leaf node labeled with the class C.
- Otherwise, apply attribute selection method on a single attribute. This attribute is the root of the tree with one child for each splitting attribute of the method, partition M into the corresponding tuples and grow subtrees for each partition

Fig1: Decision tree algorithm

In last step C4.5 use attribute measure: information gain, which calculates the total entropy and second approach, is gain ratio which divides the information gain by the information given by the attribute.

Decision trees prefer a splitting method and not return to splitting node. Lookahead methods are discussed in in Murthy & Salzberg (1995)[7][8] Decision tree update incrementally if more data is given described by Utgo(1997) [9][10] Methods for scaling to larger datasets are described by Shafer, Agrawal & Mehta (1996) [11], in SPRINT algorithm and by Freitas & Lavington (1998)[12][13] Friedman, Kohavi & Yun 1996[14][15], described Lazy decision trees. For given test instance lazy decision tree choose the best tree conceptually.To evade over fitting original tree is pruned. Pruned trees tend to be smaller and less complex and easier to comprehend. Two approaches to tree pruning: prepruning and postpruning. The cost complexity pruning algorithm used in CART is an example of the postpruning .C4.5 uses a method called pessimistic pruning. Espos-ito, Malerba & Semeraro (1995)[10][14] provide the difference of pruning and grafting methods. Kearns & Mansour (1998)[13]suggest a pruning algorithm. Quinlan & Rivest (1989)[15], Mehta, Rissanen & Agrawal (1995)[16], and Wallace & Patrick (1993) [17]

discussed two pruning methods MDL- (minimum description length) and MML- (minimum message length) .

2.2 Ruleset classifiers

In decision tree information about one class is generally dispersed throughout the tree, so complex decision tree can be thorny to understand. C4.5 gives a different formalism represented by as a set of IF-THEN rules. It consists of a set of rules of the form "if X and Y and Z and ... then class A". IF-THEN rule is an expression of the form:

IF condition THEN conclusion

If condition in a rule antecedent holds true for a given tuple, we say that the rule is satisfied. When there is no rule is satisfied by the tuple then a fallback or default rule can be set to indicate a default class. C4.5 rulesets are created from the decision tree which is not pruned.

Biao Qin et al. [18] proposed an algorithm uRule for classifying uncertain data which shows directly mining uncertain datasets. This algorithm has good performance when data is uncertain. Jiuyong Li et al [19] discussed a criterion which compares the robustness for different ruleset from a database. In [20] distinguish the relationships between *k*-optimal rule sets and a traditional classification rule set. Through optimal association rule approach they proposed a method to find the k-optimal rule set. They showed experimentally that a *k*-optimal rule set created from the proposed algorithm performs better than a *k*-optimal rule set generated by an extension of C4.5Rules.

2.3 *k*nn: *k*-nearest neighbor classification

In Eager learners,training tuples, are given and construct classifier model and then test the tuples to classify. A lazy learner memorize the training tuple and performs classificatiion only if the attributes of test tuples match with the any one training tuple. These approaches do less work. One of the lazy learners is k-nearest neighbor classifier (kNN). Nearest neighbor was originally proposed by Fix and Hodges [21] in 1952. kNN [22,23]first find out k objects in the training data that are neighbor to the test object. These k training tuples are the k-nearest neighbors" of the unknown test tuple. Then assign this test tuple with the class of training tuple. Patrick and Fischer [24] take a broad view of the nearest neighbor rules consist of weighting of different types of error and problems "in which the training datasets available are not in the same proportions as in the priori class probabilities" [22] proposed an algorithm, PEBLS which is based on *k*-NN classification that includes similarity measure for class information. In text classification *k*-nearest neighbor (k-NN) classification shown to be very effective because it is object based learning algorithm. [20, 25]The distance from the unlabelled object to the labeled object is computed, k-nearest neighbors are recognized and choose the class label of the object with the class labels of these neighbors Algorithm for kNN as in [5,6, 22]is given below.

Given a training data *M* and a test instance *z* = (**a**' b'), the algorithm calculates the distance (or similarity) between *test instance z* and training instances (**a**, b) ∈ *M* to find out its nearest-neighbor list, M_z. (**a** is the tuple value of a training instance, while *b* is its class. Likewise, **a**' is the tuple value of the test instance and *b*' is its class. Algorithm for *k*nn shown in Figure 2.

Input: M **is** the set of training instances and test instance **z=** (**a**' b')
Process:
Calculate d(**a'** , **a)** , the distance between z and every object, **(a,b)** ∈**M**

Select, $M_z \subseteq$ M the set of nearby training instances to z.

Output: $Z' = \underset{b}{argmax} \sum_{(a_i,b) \in M} A(a = b_i)$

Fig 2: The *k*-nearest neighbor classification algorithm

Eui-Hong (Sam) Han et al [23] suggested an algorithm Weight Adjusted *k*-Nearest Neighbor (WAKNN) classification algorithm which is based on the *k*-NN classification. In WAKNN weight vector maintained the importance of each word in training document set which is classify. These weight vectors utilize in similarity measure. Dennis l. Wilson [26], presented results for a large class of problems the nearest neighbor rules form a set of very powerful decision rules. Their results were also shown that the modified three-nearest neighbor rule improves the performance of the single-nearest neighbor rule and the modified single-nearest neighbor rule.

2.4 Naive Bayesian Classifier

Simple classifier called naïve bayes classifier which is based on bayes theorem. One assumption called class conditional independence in this classifier i.e. the attribute value on a given class is independent of the value of the other attributes.

Naïve Byes classification

The naïve Bayesian classifier, or simple Bayesian classifier, works as follows [8, 27, 28]:

1. Let *D* be a training set of tuples and their class labels.

2. *m* classes, C_1, C_2, … , C_m. the naïve Bayesian classifier predicts that tuple ***X*** belongs to the class C_i if and only if

$P(C_i/X) > P(C_j/X)$ for $1 \le j \le m, j \ne i$

The class C_i for which $P(C_i/X)$is maximized is called the *maximum posteriori hypothesis*. By Bayes' theorem

$$P(C_i/X) = \frac{P(X/C_i)P(C_i)}{P(X)}$$

3. For class conditional independence

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

In other words, the predicted class label is the class C_i for which $P(X/C_i)P(C_i)$ is the maximum. Clark, Niblett (1989) [29], Cestnik (1990) [30] and Langley, Iba, and Thompson (1992)[27]contrast simple bayesian classifier with two rule learners and a decision-tree learner. Pazzani, Muramatsu, and Billsus (1996) [31] compared various learners on an information filtering tasks. John and Langley (1995) [32] showed for numeric attributes, if Gaussian distributions,is swapped by kernel density estimation then the Bayesian classifier's performance can be improved . Langley and Sage (1994)[28] discussed that, when two attributes are interrelated, it would be better to remove one attribute than to assume the two are conditionally independent. They found that an algorithm for feature subset selection (forward sequential selection) improved accuracy on some data sets. In a related approach, Kubat, Flotzinger, and Pfurtscheller (1993) [33] found that in the domain of EEGsignal classification decision-tree learner to select features for use in the Bayesian classifier gave good results. Langley (1993)[34] proposed the use of "recursive Bayesian classifiers" ,in which the tuple space is

recursively divided into sub regions by a hierarchical clustering process and a Bayesian classifier is applied on each region .

3. CLUSTERING

Clustering is important technique in exploratory data analysis. It finds out the useful pattern and correlation between attributes [3] Here we discussed k-means and EM algorithm for clustering.

3.1 K-means clustering Technique

D is a data set of n objects and k is number of clusters. Partitioning algorithm distributes the objects into k clusters such that objects within the cluster are similar and object with other cluster are dissimilar. First, it arbitrarily selects *k* of the objects, each of which initially represents a cluster mean or center. For each of the remaining objects, an object is assigned to the cluster to which it is the most similar, based on the distance between the object and the cluster mean. It then computes the new mean for each cluster [8] The *k*-means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster [8] shown in Figure 3.

Input: *k*: the number of clusters,
A: a data set containing *n* objects.
Output: A set of *k* clusters.
Method:

- randomly choose *k* substance from *A* as the initial cluster centers; repeat until no change
- (re)allocate each substance to that cluster with which the object is the most similar, based on the mean value of the substance in the cluster;
- Modify the cluster means, i.e., calculate the mean value of the substance for each cluster;

Fig 3: *k-means* partitioning algorithm.

O.M. San *et al.*[35] proposed an algorithm k-representative algorithm for clustering categorical data. Concept of "clusters centers" is apply on the categorical objects. Proposed algorithm shows better and stable results than k-modes algorithm. In (MacQueen, 1967)[36] Partitioning for the large datasets with numerical objects k-means clustering technique is used. A hybrid numeric-symbolic method has been proposed by Ralambondrainy (1995) [37] which combine extended version of k-means algorithm and theoretical characterization algorithm for cluster description.By Haung[38] k-modes algorithm combined with the k-means algorithm called k-prototypes algorithm which clustered mixed numerical and categorical objects. Figure 4 shows the clustering process based on given algorithm.

(a) (b) (c)

Fig 4: Clustering of a set of objects based on the *k*-means method. (The mean of each cluster is marked by a "+".)

In [39, 40], distance between objects are calculated by probability distribution function for k-means algorithm. Aggarwal and Yu [41, 42] presented an expansion of micro-clustering technique for uncertain data. In [43] discussed a conceptual clustering algorithm for uncertain categorical data.

3.2 The EM algorithm

Dempster et al. (1977) [44] given the name EM algorithm. Historical perspective of the EM algorithm can be found in McLachlan and Krishnan (1997) [45] Model-based clustering methods try to optimize the fit between the mathematical model and given data. Each cluster can be represented mathematically by a parametric probability distribution. So we cluster the data using a finite mixture [45] density model of k probability distribution Now finding the parameter estimates of probability distributions which can be best fit in the data The EM(Expectation-Maximization) algorithm is a recursive refinement algorithm that can be used for finding the parameter estimates. It can be shown as an extension of the *k*-means approach , in which object is assigned to that cluster whose cluster mean is similar to that object In EM algorithm new means are calculated by weighted measures. Then object is assigned to cluster according to weighted measures that represent the membership probability

The algorithm is described as follows [8]:

1. Initially guess the parameter vector. Randomly selecting *k* objects to represent the cluster means or centers
2. Recursively process the parameters (or clusters) based on the following two steps:
(a) Expectation Step: Assign each object ***xi*** to cluster *Ck* with the probability

$$P(x_i \epsilon c_k) = p(c_k/x_i) = \frac{p(c_k)p(x_i/c_k)}{p(x_i)}$$

where $p(x_i/c_k) = N(m_k, E_k(x_i))$ follows the normal (i.e., Gaussian) distribution around mean, m_k, with expectation, E_k.

(b) Maximization Step: Use the probability estimates from above to re-estimate (or refine) the model parameters. For example, $m_k = \frac{1}{n}\sum_{i=1}^{n} \frac{x_i P(x_i \epsilon c_k)}{\sum_j P(x_i \epsilon c_j)}$

For improving the convergence rate of the EM algorithm Neal and Hinton (1998)[46] presented the incremental EM (IEM) algorithm .Neal and Hinton (1998)[46] suggested another algorithm sparse EM (SPEM) algorithm. Speeding up the EM algorithm have been considered in Bradley et al. (1998)[47] and Moore (1999) [48]For learning hidden variables

4. MACHINE LEARNING

4.1 Support vector Machine

Recently machine learning applications, support vector machines (SVM) [6] are ought to be use because it offers algorithms for classification which are robust and accurate. The Support Vector Machine (SVM) is introduced by Boser, Guyon, and Vapnik in 1992[49]If the data is not normalized [50] the accuracy of an SVM will be degrade. Recently SVM also extend in the domain of regression problems (Vapnik et al., 1997) [51] In SVM a linear classification function $R(a_n)$ is determined that separate hyperplane which passes through the middle of two classes. How training data are linearly separable shown in Figure 5

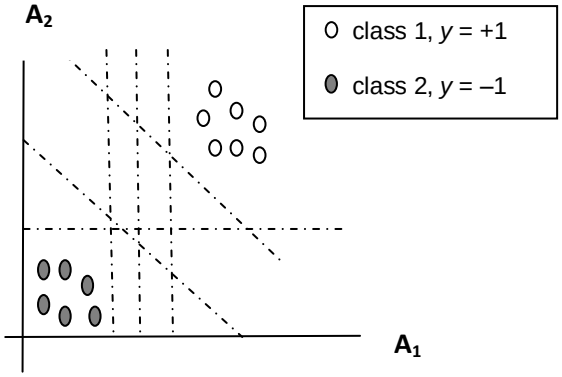

Fig 5: The 2-D training data are linearly separable.

There are many hyperpalne from which SVM select the best one for separating the instances by maximal the margin between the two classes. Margin is defined as the shortest distance from a hyperplane to one side of its margin is equal to the shortest distance from the hyperplane to the other side of its margin, where the "sides" of the margin are parallel to the hyperplane.
Fig 6 and 7 shows two possible separating hyperplanes with small and large margin.

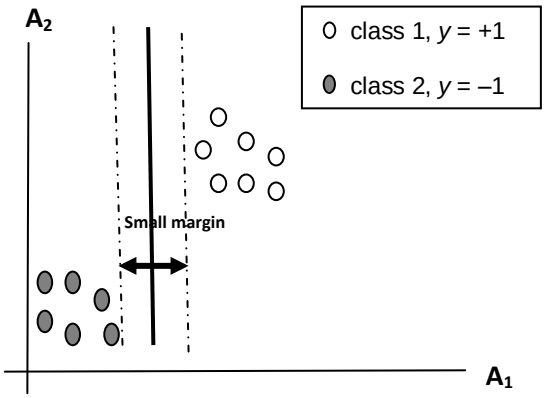

Fig 6: Hyperplane with small margin

A separating hyperplane can be written as

$$\boldsymbol{W}\_\boldsymbol{X}+b = 0;$$

For maximum margin hyperplanes, following function is to be maximized with respect to **w** and *b*.

$$L_P = \frac{1}{2}\|w\| - \sum_{i=1}^{n} a_i\, y_i(w.x_i + b) + \sum_{i=1}^{t} a_i$$

where n is the number of training tuples, and a_i , i = 1, . . . , t, are positive numbers . L_p is called the Lagrangian function and a_i are the Lagrangian multipliers. Variable **w** and constant *b* define the hyperplane. There are an infinite number of (possible) separating hyperplanes or "decision boundaries."Which one is best?

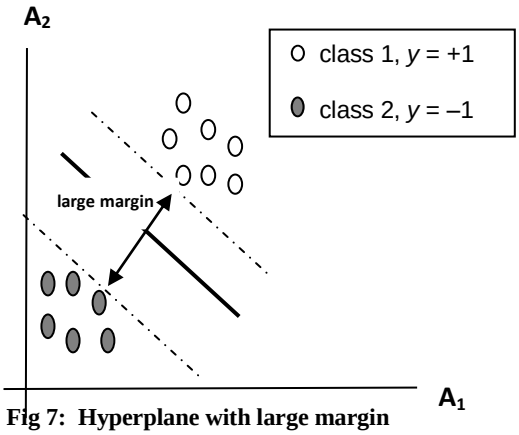

Fig 7: Hyperplane with large margin

5. ASSOCIATION

5.1 Apriori algorithm

This algorithm is one of the most important data mining approaches. This is to find frequent item sets as in [52]from a dataset and derive association rules. When frequent item sets are generated, it is easy to generate association rules with confidence greater than or equal to a give minimum confidence. For finding frequent itemsets, Apriori is a influential algorithm using candidate generation [53]
Let A_k is set of frequent itemsets of size k and B_k are their candidates. Apriori initially search the database for frequent itemsets of size 1 by collecting the count for each item that having minimum support count. To improve the efficiency of the level-wise generation of frequent itemsets, an important property called Apriori property: "*All nonempty subsets of a frequent itemset must also be frequent",* is used. Apriori algorithm shown in Figure 8.

A_i=(Frequent itemsets of Cardinality 1)
for(k=1;$A_k \neq \phi$;k++) do begin
  B_{k+1}=Aprior-gen(A_k) //New candidates
  for all transactions t $\in$ Database do begin
    B_t=subset(B_{t+1},t);//Candidates contained in t
    for all candidates b$\in$ B_t do
      b.count++
    end
  A_{k+1}={B=B_{k+1}b.count>=minimum support}
  end
end
Answer $\cup$k Ak

Fig 8: Apriori algorithm

Finding frequent itemset and association mining from uncertain datasets discussed in [54, 55] Mining association rules over basket data was initiated in [8] Rakesh and Ramakrishnan [53] presented two algorithms, Apriori and AprioriTid, for finding all significant association rules between items in a large database of transactions. They compared AIS [52] and SETM [56] algorithms with these algorithms. Results show that proposed algorithms perform better than AIS and SETM.

6. PERFORMANCE ANALYSIS

The evaluation takes in to account the cost of making wrong decision, wrong classification. When we take the two classes yes and no , single prediction take the four possible outcomes shown in Figure 9 i.e. true positive (TP), true negative, (TN) flse positive (FP), false negative(FN). TP and TN are the correct classification and FP and FN are incorrect classification.

Table1. Different outcomes of two class prediction

Predicted Class			
Actual Class		YES	NO
	YES	True positive	False negative
	NO	False positive	True negative

A *false positive* (FP) occurs when the outcome is incorrectly predicted as *yes* (or positive) when it is actually *no* (negative). A *false negative* (FN) occurs when the outcome is incorrectly predicted as negative when it is actually positive. The *true positive rate* is TP divided by the total number of positives, which is TP + FN; the *false positive rate* is FP divided by the total number of negatives, FP + TN. The overall success rate is the number of correct classifications divided by the total number of classifications: $\frac{TP+TN}{TP+TN+FP+FN}$

Finally, the error rate is one minus this.

ROC curves depict the performance of a classifier without regard to class distribution or error costs. They plot the number of positives included in the sample on the vertical axis, expressed as a percentage of the total number of positives, against the number of negatives included in the sample, expressed as a percentage of the total number of negatives, on the horizontal axis. Information retrieval researches define parameters called recall and precision.

$$recall = \frac{\text{number of documents retrieved that are relevant}}{\text{total number of documents that are relevant}}$$

$$Precision = \frac{\text{number of documents retrieved that are relevant}}{\text{total number of documents that are retrieved}}$$

F-measure is another information retrieval measure that is calculated from TP, FP, FN or recall or precision values

$$F - measure = \frac{2*recall*Precsion}{recall+Precsion}$$

$$F - measure = \frac{2 * TP}{2 * TP + FP + FN}$$

Table 2. Name and function of algorithm in Weka

Name	Function
J48	C4.5 decision tree learner (implements C4.5 revision 8),
Prism	Simple covering algorithm for rules
IBk	*k*-nearest-neighbor classifier
NaiveBayes	Standard probabilistic Naïve Bayes classifier
SMO	Sequential minimal optimization algorithm for support vector classification

We implement all the classifier in weka on inbuilt data weather nominal. Name with their functions of classifier for weka given in Figure 10. Below shows the accuracy result of all the classifiers in all measures discussed above. Chart depicts that the PRISM have higher accuracy than all the classifier because all the measure have higher value in measure than other classifiers. After PRISM SVM give higher accuracy and then IBK which rule based learner. One thing is to be noticed that Naïve bayes have higher FP rate which is not required. So naïve bayes is not good classifier for this data. Overall we check the F-measure value which is higher in PRISM. After SMO then IBK and naïve bayes and then J48. So J48 have less accuracy than all other classifier.

Table 3. Accuracy Table

	J48	**PRISM**	**IBK**	**Naïve Bayes**	**SMO**
TP Rate	0.5	0.75	0.571	0.571	0.643
FP Rate	0.544	0.35	0.505	0.594	0.465
Precision	0.521	0.825	0.571	0.528	0.629
Recall	0.5	0.75	0.571	0.571	0.643
ROC	0.633	0.635	0.484	0.578	0.589
F – measure	0.508	0.718	0.571	0.539	0.632

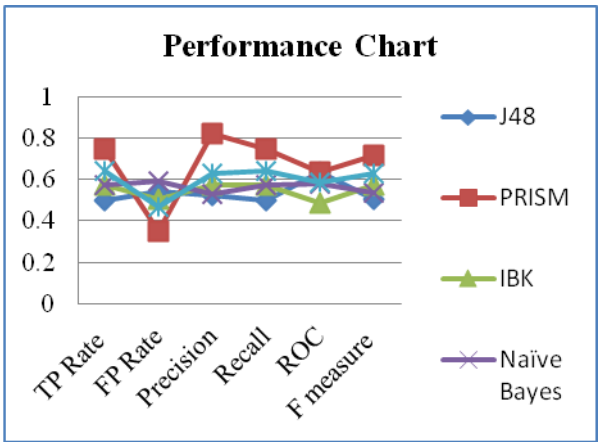

Fig 9: Performance Chart

In Clustering we discussed above two algorithm EM algorithm and k-means. We implement these two in weka and find the result that makes the clusters. In weka clustering algorithm shows only the number of instances is correctly clustered and incorrectly clustered. From this we are able to know that which algorithm is best. Table show the results that the EM algorithm correctly clustered more instances than k-means algorithm.

Table 4. Result of clustering algorithm

	EM	k-means
Correctly clustered	9	7
Incorrectly Clustered	5	7

7. CONCLUSION

Information plays a major role in every field. Data mining is a tool that exploits to discover patterns from raw data, extraction of useful information stored. Data mining is wide area that assimilates techniques from various fields including pattern recognition, artificial intelligence, database systems and machine learning. We discussed here few data mining algorithms which are used to perform data analysis tasks in different fields. Simple covering algorithm (PRISM in weka) has higher accuracy than other classifiers. These algorithms employed in fraud detection, intrusion detection, Health care and finance for extraction of useful information.

8. REFERENCES

[1] P.Ponniah, "Data Warehousing Fundamentals- "A comprehensive guide for IT professionals", Ist ed.,second reprint , ISBN-81-265-0919-8, Glorious Printers: New Delhi India, 2007.

[2] An Introduction to Data Mining,Review: http://www.thearling.com/text/dmwhite/dmwhite.htm

[3] A Tutorial on Clustering Algorithms, Review http://home.dei.polimi.it/matteucc/Clustering/tutorial_html

[4] Naive Bayes Classifier Review: http://www.statsoft.com/textbook/naive-bayes-classifier/

[5] Pang-Ning Tan,Michael Steinbach,Vipin Kumar, "An Introduction to Data Mining", ISBN : 0321321367. Addison-Wesley, 2005 .

[6] XindongWu · Vipin Kumar et all, "Top 10 algorithms in data mining" Knowl Inf Syst (2008) 14:1–37 DOI 10.1007/s10115-007-0114-2

[7] Murthy, S. & Salzberg, S. (1995), Lookahead and pathology in decision tree induction,in C. S. Mellish, ed., `Proceedings of the 14th International Joint Conference on Articial Intelligence', Morgan Kaufmann, pp. 1025-1031.

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

[9] Utgo, P. E. (1997), `Decision tree induction based on efficient tree restructuring', Machine Learning 29, 5.

[10] Esposito, F., Malerba, D. & Semeraro, G. (1995), Simplifying decision trees by prun-ing and grafting: New results, in N. Lavrac & S. Wrobel, eds, `Machine Learning:ECML-95 (Proc. European Conf. on Machine Learning, 1995)', Lecture Notes in Articial Intelligence 914, Springer Verlag, Berlin, Heidelberg, New York,pp. 287-290.

[11] Shafer, J., Agrawal, R. & Mehta, M. (1996), Sprint: a scalable prallel classier for data mining, in `Proceedings of the 22nd International Conference on Very Large Databases (VLDB)'.

[12] Freitas, A. A. & Lavington, S. H. (1998), Mining Very Large Databases with Parallel Processing, Kluwer Academic Publishers.

[13] Kearns, M. & Mansour, Y. (1998), A fast, bottom-up decision tree pruning algorithm with near-optimal generalization, in J. Shavlik, ed., `Machine Learning: Proceedings of the Fifteenth International Conference', Morgan Kaufmann Publishers,Inc., pp. 269-277.

[14] Friedman, J., Kohavi, R. & Yun, Y. (1996), Lazy decision trees, in `Proceedings of the Thirteenth National Conference on Articial Intelligence', AAAI Press and the MIT Press, pp. 717-724.

[15] Quinlan, J. R. & Rivest, R. L. (1989), `Inferring decision trees using the minimum description length principle', Information and Computation 80, 227-248.

[16] Mehta, M., Rissanen, J. & Agrawal, R. (1995), MDL-based decision tree pruning,in U. M. Fayyad & R. Uthurusamy, eds, `Proceedings of the first international conference on knowledge discovery and data mining', AAAI Press, pp. 216-221.

[17] Wallace, C. & Patrick, J. (1993), `Coding decision trees', Machine Learning 11, 7-22.

[18] Biao Qin, Yuni Xia et al. "A Rule-Based Classification Algorithm for Uncertain Data" IEEE International Conference on Data Engineering 2009, pp 1633-1640.

[19] Jiuyong Li et al. Construct robust rule sets for classification, SIGKDD '02 Edmonton, Alberta, Canada.

[20] Y. Yang. Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. In SIGIR-94, 1994.

[21] E. Fix and J. L. Hodges, Jr., "Discriminatory analysis, nonparametric discrimination: consistency properties," U.S. Air Force Sch. Aviation Medicine, Randolf Field, Tex., Project 21-49-004, Contract AF 41(128)-31, Rep. 4, Feb. 1951.

[22] S. Cost and S. Salzberg. A weighted nearest neighbor algorithm for learning with symbolic features. Machine Learning,10(1):57–78, 1993.

[23] Eui-Hong (Sam) Han et al.," Text Categorization Using Weight Adjusted k-Nearest Neighbor Classification.

[24] E. A. Patrick and F. P. Fischer, III, "A generalized k-nearest neighbor rule," Inform. Contr., vol. 16, pp. 128-152, Apr. 1970.

[25] W.W. Cohen and H. Hirsh. Joins that generalize: Text classification using WHIRL. In Proc. of the Fourth Int'l Conference on Knowledge Discovery and Data Mining, 1998.

[26] Dennis l. Wilson, Asymptotic properties of nearest neighbor rules using edited data" IEEE transactions on systems, man, and cybernetics, vol. Smc-2, no. 3, july 1972.

[27] Langley, P., Iba, W., & Thompson, K. (1992). An analysis of Bayesian classifiers. Proceedings of the Tenth National Conference on Artificial Intelligence (pp. 223–228). San Jose, CA: AAAI Press.

[28] Langley, P., &Sage, S. (1994). Induction of selective Bayesian classifiers. In Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence (pp. 399–406). Seattle, WA: Morgan Kaufmann.

[29] Clark, P., & Niblett, T. (1989). The CN2 induction algorithm. Machine Learning, 3, 261–283.

[30] Cestnik, B. (1990). Estimating probabilities: A crucial task in machine learning. Proceedings of the Ninth European Conference on Artificial Intelligence. Stockholm, Sweden: Pitman.

[31] Pazzani, M., Muramatsu, J.,&Billsus, D. (1996). Syskill&Webert: Identifying interesting web sites. Proceedings of the Thirteenth National Conference on Artificial Intelligence (pp. 54–61). Portland, OR: AAAI Press.

[32] John, G., & Langley, P. (1995). Estimating continuous distributions in Bayesian classifiers. Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence (pp. 338–345) . Montr´eal, Canada: Morgan Kaufmann.

[33] Kubat, M., Flotzinger, D., &Pfurtscheller, G. (1993). Discovering patterns in EEG-Signals: Comparative study of a few methods. Proceedings of the Eighth European

Conference on Machine Learning (pp. 366–371). Vienna, Austria: Springer-Verlag.

[34] Langley, P. (1993). Induction of recursive Bayesian classifiers. Proceedings of the Eighth European Conference on Machine Learning (pp. 153–164). Vienna, Austria: Springer-Verlag.

[35] O.M. San et al.," An alternative extension of the k-means algorithm for clustering categorical data" Int. J. Appl. Math. Comput. Sci., 2004, Vol. 14, No. 2, 241–247.

[36] MacQueen J.B. (1967): Some methods for classification and analysis of multivariate observations.—Proc. 5-th Symp.Mathematical Statistics and Probability, Berkelely, CA,Vol. 1, pp. 281–297.

[37] Ralambondrainy H. (1995): A conceptual version of the kmeans algorithm. — Pattern Recogn. Lett., Vol. 15, No. 11, pp. 1147–1157.

[38] Huang Z. (1998): Extensions to the k-means algorithm for clustering large data sets with categorical values. — Data Mining Knowl. Discov., Vol. 2, No. 2, pp. 283–304.

[39] W. K. Ngai, B. Kao, C. K. Chui, R. Cheng, M. Chau, and K. Y. Yip, "Efficient clustering of uncertain data," in IEEE International Conference on Data Mining (ICDM) 2006, pp. 436–445.

[40] M. Chau, R. Cheng, B. Kao, and J. Ng, "Data with uncertainty mining: An example in clustering location data," in Proc. of the Methodologies for Knowledge Discovery and Data Mining, Pacific-Asia Conference (PAKDD 2006), 2006.

[41] A. C, "On density based transforms for uncertain data mining," in Proceedings of IEEE 23rd International Conference on Data Engineering,2007, pp. 866–875.

[42] A. C and Y. PS, "A framework for clustering uncertain data streams," in Proceedings of IEEE 24rd International Conference on Data Engineering, 2008, pp. 150–159.

[43] Y. Xia and B. Xi, "Conceptual clustering categorical data with uncertainty," in IEEE International Conference on Tools with Artificial Intelligence (ICTAI), 2007, pp. 329–336.

[44] Dempster, A.P., Laird, N.M., and Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39:1–38.

[45] McLachlan, G.J. and Krishnan, T. (1997). *The EM Algorithm and Extensions*. Wiley, New York.

[46] Neal, R.M. and Hinton, G.E. (1998). A view of the EM algorithm that justifies incremental,sparse, and other variants. In Jordan, M.I., editor, *Learning in Graphical Models*, pages 355–368. Kluwer, Dordrecht.

[47] Bradley, P.S., Fayyad, U.M., and Reina, C.A. (1998). Scaling EM (expectation maximization) clustering to large databases. Technical Report No. MSR-TR-98- 35 (revised February, 1999), Microsoft Research, Seattle.

[48] Moore, A.W. (1999). Very fast EM-based mixture model clustering using multiresolution *k*d-trees. In Kearns, M.S., Solla, S.A., and Cohn, D.A., editors, *Advances in Neural Information Processing Systems 11*, pages 543–549. MIT Press, MA.

[49] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, 5th Annual ACM Workshop on COLT, pages 144{152, Pittsburgh, PA,1992. ACM Press.

[50] C-C. Chang and C-J. Lin. LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[51] V. Vapnik, S. Golowich, and A. Smola. Support vector method for function approximation, regression estimation, and signal processing. In M. Mozer, M. Jordan, and T. Petsche, editors, Advances in Neural Information Processing Systems 9, pages 281–287, Cambridge, MA, 1997. MIT Press.

[52] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In Proc. of the ACM SIGMOD Conference on Management of Data, Washington, D.C., May 1993.

[53] Rakesh Agrawal Ramakrishnan Srikan," Fast Algorithms for Mining Association Rules," Proceedings of the 20th VLDB Conference Santiago, Chile, 1994

[54] Z. Yu and H. Wong, "Mining uncertain data in low-dimensional subspace," in *International Conference on Pattern Recognition (ICPR) 2006*,pp. 748–751.

[55] C. Chui, B. Kao, and E. Hung, "Mining frequent itemsets from uncertain data," in *Proc. of the Methodologies for Knowledge Discovery and Data Mining, Pacific-Asia Conference (PAKDD) 2007*, pp. 47–58.

[56] M. Houtsma and A. Swami. Set-oriented mining of association rules. Research Report RJ 9567, IBM Almaden Research Center, San Jose, California, October 1993.