Data Quality Measurement With Threshold Using Genetic Algorithm

J.Malar Vizhi[1] and Dr. T.Bhuvaneswari[2]

[1]Research scholar,Bharathiar University,Coimbatore,Tamil Nadu,India.
[2]Assistant Professor in the Department of Computer Science, Government College for women, Burgoor,Tamil Nadu,India.

ABSTRACT

Our basic idea is to employ association rule for the purpose of data quality measurement. Strong rule generation is an important area of data mining. We purpose a Genetic Algorithm to generate high quality Association Rules with four metrics they are confidence, completeness, interestingness and comprehensibility. These metrics are combined as an objective fitness function. Fitness function evaluates the quality of each rule. The advantage of using genetic algorithm is to discover high level prediction rules is that they perform a global search and cope better with attribute interaction than the greedy rule induction algorithm often used in data mining. Association Rule Mining is one of the most applicable techniques in data mining. Association rules that satisfy threshold specified by the user are referred to as strong association rules and a considered interesting.

Keywords:-Association Rule, Threshold, Michigan approach, Genetic Algorithm, Data quality Mining.

1. INTRODUCTION

Data Mining is the most instrumental tool in discovering knowledge from transactions [1, 2].The most important application of data mining is discovering association rules. This is one of the most important methods for pattern recognition in unsupervised systems. These methods find all possible rules in the database. Some measures like support and confidence are used to indicate high quality rules. Data mining techniques can be employed in order to improve Data quality. Poor data quality is always a problem in practical application. Data Quality Mining can be defined as the deliberate applications of Data Mining techniques for the purpose of Data quality, measurement and improvement [3].

Data Quality Mining is one of the most important tasks in data mining community and is an active research area because the data being generated and stored in databases of organizations are already enormous and continues to grow very fast. This large amount of stored data normally contains valuable hidden knowledge, which if harnessed could be used to improve the decision making process of an organization. However the volume of the archival data often exceeds several gigabytes and sometimes even terabytes. Such an enormous volume of data is beyond the manual analysis capability of human beings. Thus, there is a clear need for developing automatic methods for extracting knowledge from data that not only has a high accuracy but also comprehensibly and interestingness by user.

We describe a first approach to employ Association Rules for the purpose of data quality mining. Data mining algorithms like Association Rule Mining perform an exhaustive search to find all rules satisfying some constraints [4]. Genetic Algorithm is a new approach for Association rule mining (ARM). Finding the frequent rules is the most resource consuming phase in association rule mining and always does some extra comparison against the whole database [5].Although Genetic Algorithm is good at searching for undetermined solution, it is still rare to see that Genetic Algorithm, is used to mine Association Rules. By introducing Genetic Algorithm on the rules generated by Association Rule, it is based on four Objectives Confidence, Completeness, Comprehensibility and Interestingness.

Common approach in rule generation process is steps.

1. Applying the minimum support to find all frequent itemsets in a data base
2. Applying the minimum confidence constraints on the frequent itemset to form the rules.

The process of discovering an interesting and an unexpected rule from large datasets is known as ARM. The major aim of ARM is to find the set of all subsets of items or attributes that frequently occur in many database records or transactions and additionally, to extract rules on how a subset of items influences the presence of another subset. An AR is a way of describing a relationship that can be observed between database attributes, of the form. If the set of attributes values A is found together in a database record, then it is likely that the set B will be present also. A rule of this form ,A→B is of interest only if it meets atleast two threshold requirements defines the number of database records within which the association can be observed. The confidence in the rule is the ratio of its support to that of its antecedent, ARM aims to uncover all such relationship that are present in a database for specified thresholds of support and confidence.

A rule is excavated out if and only if both of its confidence and support are greater than two thresholds, the minimum confidence and the minimum support, respectively. Therefore, users need to appropriately specify these two thresholds before they start this mining job.

In general, each measure is associated with a threshold that can be controlled by the user. Rules that do not meet the threshold are considered uninteresting, and hence are not presents to the user as knowledge. Association rule that specify both a user specified minimum confidence and user specified minimum support are referred to as strong association rules, and are considered interesting.

2. RELATED WORKS

Existing algorithms for mining association rules are mainly based on the approach suggested by Agrawal et al. [6, 7]. Apriori [7], SETM [8], AIS [7], Pincer search [9], DIC [10] etc. are some of the popular algorithms based on this approach. These algorithms work on a binary database, termed as market basket database. On preparing the market basket database, every record of the original database is represented as a binary record where the fields are defined by a unique value of each attribute in the original database. The fields of this binary database are often termed as an item. For a database having a huge number of attributes and each attribute containing a lot of distinct values, the total number of items will be huge. Existing algorithms, try to measure the quality of generated rule by considering only one evaluation criterion, i.e., confidence factor or predictive accuracy. This criterion evaluates the rule depending on the number of occurrence of the rule in the entire database. More the number of occurrences better is the rule. The generated rule may have a large number of attributes involved in the rule thereby making it difficult to understand [11]. If the generated rules are not understandable to the user, the user will never use them. Again, since more importance is given to those rules, satisfying number of records, these algorithms may extract some rules from the data that can be easily predicted by the user. It would have been better for the user, if the algorithms can generate some of those rules that are actually hidden inside the data. These algorithms do not give any importance towards the rare events, i.e., interesting rules [12, 13].of mining association rules over market basket data was first introduced by Agrawal. The task in the present work we used the comprehensibility and the interestingness measure of the rules in addition to predictive accuracy. According to Zaki [1999] the mining task involves generating all association rules in the database that have a support greater than the minimum support (the rules are frequent) and have a confidence greater than minimum confidence (rules are strong).

Traditionally, ARM was predominantly used in market-basket analysis but it is now widely used in other application domains including customer segmentation, catalogue design, store layout, and telecommunication alarm prediction.[14]

3 Association Rule

The process of discovering an interesting and an unexpected rule from large data sets is known as Association rule mining. Association rule mining is one of the most challenging areas of data mining which was introduced in Agrawal to discover the associations or co-occurrence among the different attributes of the dataset. Several algorithms like Apriori(Agrawal), SETM(Houtsma and Swami), AprioriTID(Agrawal and Srikant), DIC(Brin), Partition algorithm(Savasere), Pincer search(Lin and Kedem), FP-tree(Han) etc have been developed to meet the requirements of this problem. These algorithms work basically in two phases: the frequent itemset generation and the rule generation. Since the first phase is the most time consuming, all the above mentioned algorithms maintain focus on the first phase. A set of attributes is termed as frequent set if the occurrence of the set within the dataset is more than a user specified threshold called minimum support. After discovering the frequent itemsets, in the second phase rules are generated with the help of another user parameter called minimum confidence.

The typical approach is to make strong simplifying assumption about the form of the rules and limit the measure of rule quality to simple properties such as support and confidence [7]. Support and confidence limit the level of interestingness of the generated rules. Completeness and accuracy are metrics that can be used together to find interesting association rules .The major aim of Association rule mining is to find the sets of all subsets of items or attributes that frequently occur in many databases records or transaction. Association Rule Mining algorithm discover high level prediction rules in the form

IF THE CONDITION of the values of the predicating attributes are TRUE
THEN
Predict values for some goal attributes.

An Association Rule is a conditional implication among itemsets X→Y where X→Y are itemsets and X∩Y=Φ. An itemset is said to be frequent or large if it support is more than a user specified minimum support value. The confidence of an Association Rule is given as Support(XUY)/Support(X) is the conditional probability that a transaction contains Y given that is also contains X.

Example1: (adapted from Brin et.al[1997]).Suppose we have a market basket database from a grocery store, consisting of n baskets. Let us focus on the purchase of tea (denoted by t) and coffee (denoted by c). When supp(t)=0.25 and supp(tUc)=0.2,we can apply the support-confidence framework for a potential association rule t→c. The support for this rule is 0.2,which is fairly high. The confidence is the conditional probability that a customer who buys tea buys coffee, i.e., conf(t→c)=supp(tUc)/supp(t) =0.2/.25=0.8, which is very high. In this case we would conclude that the rule t→c is a valid one.

The rules thus generated will influenced by the choice of ARM parameters employed by the algorithm (typically support and confidence threshold values).The effect of this choice will affect predictive accuracy. The accuracy can almost always improve by a suitable choice of parameter. Rules whose support and confidence values are below user-specified thresholds are considered uninteresting. In general, each measure is associated with a threshold that can be controlled by the user. Rules that do not meet the threshold are considered uninteresting, and hence are not presented to the user as knowledge.

Association rules that satisfy both a user-specified minimum confidence threshold and user-specified minimum support threshold are referred to as strong association rules, and are considered interesting. Rules that satisfy both a minimum support threshold (min sup) and a minimum confidence threshold (min conf) are called strong.

One of the main challenges in ARM is to identify frequent itemsets in very large transactions databases that comprise million of transactions and items, another main challenge is that the performance of these frequent itemset mining algorithms is heavily dependent on the user-specified threshold of minimum support . Very often a minimum support value is too big and nothing is found in a database, whereas a slightly small minimum support leads to low-performance. Users have to give a suitable minimum support for a mining task.

4. GENETIC ALGORITHM FOR RULE MINING

Usually, genetic algorithms for rules mining are partitioned into two categories according to their encoding of rules in the population of chromosomes (Freitas, 2003). One encoding method is called Michigan approach, where each rule is encoded into an individual. Another is referred to as Pittsburgh approach, with which a set of rules are encoded into a chromosome. For example, Fidelis, Lopes, and Freitas (2000) gave a Michigan type of genetic algorithm to discover comprehensible classification rules, having an interesting chromosome encoding and introducing a specific mutation operator. But the method is impractical when the number of attribute is large. Weiss and Hirsh (1998) also followed the Michigan method to predict rare events. Pei, Goodman, and Punch (1997), on the other hand, used the Pittsburgh approach for discovery of classes and feature patterns.

Although it is known that genetic algorithm is good at searching for nondeterministic solution, it is still rare to see that genetic algorithm is used to mine association rules. We are going to further investigate the possibility of applying genetic algorithm to the association rules mining in the following sections In the context of this research the use of the term Michigan approach will denote any approach where each GA individual encodes a single prediction rule. The choice between these two approaches strongly depends on which kind of rule is to be discovered. This is related to which kind of data mining task being addressed. Suppose the task is classification. Then evaluate the quality of the rule set as a whole, rather than the quality of a single rule. In other words, the interaction among the rules is important. In this case, the Pittsburgh approach seems more natural [Frietas 2002]. However, this approach leads to syntactically-longer individuals, which tends to make fitness computation more computationally expensive. In addition, it may require some modifications to standard genetic operators to cope with relatively complex individuals. [18]

By contrast, in the Michigan approach the individuals are simpler and syntactically shorter. This tends to reduce the time taken to compute the fitness function and to simplify the design of genetic operators. However, this advantage comes with a cost. First of all, since the fitness function evaluates the quality of each rule separately, now it is not easy to compute the quality of the rule set as a whole - i.e. taking rule interactions into account. Another problem is that, since we want to discover a set of rules, rather than a single rule, we cannot allow the GA population to converge to a single individual which is what usually happens in standard GAs. Michigan approach uses the niching method as mentioned earlier to avoid the running of GA several times, by discovering a different rule each time. This introduces the need for some kind of niching method.[16]

5. GENETIC BASED LEARNING

Association rule can be represented as an *IF A THEN C* statement. The only restriction here is that the two parts should not have a common attribute. To solve this kind of mining problem by multiobjective genetic algorithm, the first task is to represent the possible rules as individuals known as individual representation. Second task is to define the fitness function and then genetic materials. We propose to solve the ARM problem with a pareto based MOGA. There have been many applications of genetic algorithms in the field of DM and knowledge discovery.[17]

Genetic Algorithm (GA) was developed by Holland in 1970. This incorporates Darwinian evolutionary theory with sexual reproduction. Genetic Algorithm is stochastic search algorithm modeled on the process of natural selection, which underlines biological evolution. Genetic Algorithm has been successfully applied in many search, optimization, and machine learning problems.[19] Genetic Algorithm process works in an iteration manner by generating new populations of strings from old ones. Every string is the encoded binary, real etc., version of a candidate solution. Standard Genetic Algorithm apply genetic operators such selection, crossover and mutation on an initially random population in order to compute a whole generation of new strings.

- Selection deals with the probabilistic survival of the fittest, in that fit chromosome are chosen to survive, where fitness is a comparable measure of how well a chromosome solves the problem at hand.
- Crossover takes individual chromosomes from Parents, combines them to form new ones.
- Mutation alters the new solutions so as to add stochastic in the search for better solutions.

In general the main motivation for using Genetic Algorithms in the discovery of high-level prediction rules is that they perform a global search and cope better with attribute interaction than the greedy rule induction algorithms often used in data mining. This section discusses the aspects

of Genetic Algorithms for rule discovery.

5.1 Selection:

We used tournament selection somewhat tailored for our data mining-oriented fitness function, as follows. In a conventional version of tournament selection, first *k* individuals are randomly picked, with replacement, from the population. Then the individual with the best fitness value, out of the *k* individuals, is selected as the winner of the tournament. This process is repeated *P* times, where *P* is the population size.

we have modified our tournament selection procedure so that the winner of the tournament is the individual with the largest fitness among the set of individuals whose Information Gain is greater than or equal to the average Information Gain of all

the *k* individuals playing the tournament. (We have used a tournament size (*k*) of 5.) Once the tournament selection process has selected *P* individuals, these individuals undergo crossover and mutation

5.2 Mutation

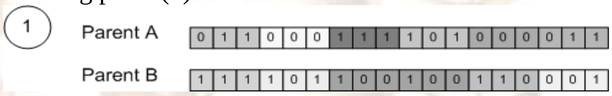
This part of the genetic algorithms, require great care, here there are two probabilities, one usually called as pm, this probability will be used to judge whether mutation has to be done or not, when the candidate fulfills this criterion it will be fed to another probability and that is, locus probability that is on which point of the candidate the mutation has to be done. In the case of database provided, binary encoding is used thus simple toggling operator is required for mutation,

5.3 Crossover

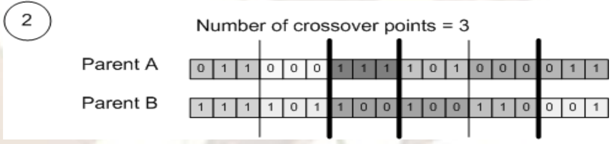
Same as the case with Mutation here two probabilities are there, one for, whether crossover has to be performed or not, i.e. pc, and other for finding the location, the point where, crossover must be done.

Due to the fact that there may be a large number of attributes in the database, we propose to use multipoint crossover operator.

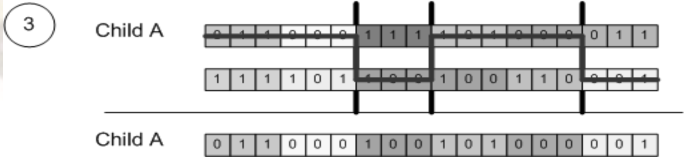
Multi point crossover method is widely used to shuffle the different characteristics of an individual and create children out of parents inheriting the characteristics directly. A chromosome of the Parent A and the same chromosomes of the Parent B are aligned. Each border of the genes can be used as a crossing point (1).



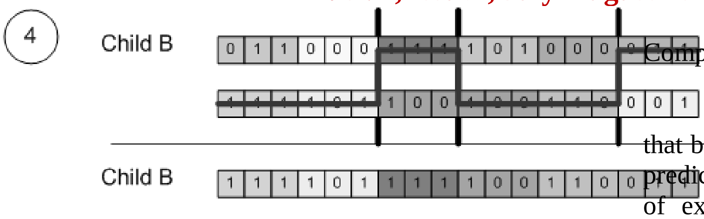
Depending on the number of multi crossing points the crossing method chooses the crossing points randomly (2).



Once chosen this points the children can be created. Lets start with child A. Child A gets the first genes of parent A until the first crossing point is reached. Then all genes of parent B are copied to the chromosome of child A until the next crossing point is reached. This procedure is continued until the whole chromosome of child A is generated (3).



The chromosome of child B is created in the similar way but starting copying the genes of parent B first (4).



6. EVALUATING THE QUALITY OF A RULE:

The fitness function used in this paper consists of four metrics including. We combine these metrics into an objective fitness function. The complementary set of measures include confidence defined in equation (1), completeness in equation (2) , Interestingness defined in equation (3) and comprehensibility defined in equation (4) .

Let a rule be of the form: IF A THEN C, where A is the antecedent (a conjunction of conditions) and C is the consequent (predicted class), as discussed earlier. A very simple way to measure the predictive accuracy of a rule is to compute the so-called confidence factor (CF) of the rule, defined as:

$$CF = |A \& C| / |A| \qquad (1)$$

Where |A| is the number of examples satisfying all the conditions in the antecedent A and |A & C| is the number of examples that both satisfy the antecedent A and have the class predicted by the consequent C.

We can now measure the predictive accuracy of a rule by taking into account not only its CF but also a measure of how “complete” the rule is, i.e. the proportion of examples is, having the predicted class C that is actually covered by the rule antecedent. The rule completeness measure is computed by the formula:

$$\text{Completeness} = |A \& C| / |C| \qquad (2)$$

Where |C| is the number of examples satisfying all the conditions in the consequent C and |A & C| is the number of examples that both satisfy the antecedent A and have the class predicted by the consequent C.

The rule interestingness measure is computed by the formula:

$$\text{Interestingness}= |A\&C| - (|A|*|C|)/N \qquad (3)$$

Where |A&C| is the number of examples that both satisfy the antecedent A and have the class predicted by the consequent C minus the product of |A| is the number of examples satisfying all the conditions in the antecedent A and |C| is the number of examples satisfying all the conditions in the consequent. The Following expression can be used to quantify the comprehensibility of an association rule

$$\text{Comprehensibility}=\log(1+|C|)/\log(1+|A\&C|) \quad (4)$$

Where |A&C| is the number of examples that both satisfy the antecedent A and have the class predicted by the consequent C and |C| is the number of examples satisfying all the conditions in the consequent

The fitness function is calculated as the arithmetic weighted average confidence, completeness, interestingness and comprehensibility. The fitness function (*f*(*x*)) is given by:

$$f(x)=\frac{W1*\text{comprehensibility}+W2*\text{interestingness}+W3*\text{completeness}+W4*\text{confidence}}{W1+W2+W3+W4}$$

(5)

Where w1,w2,w3.w4 are used defined weights

7. THE PROPOSED ALGORITHM

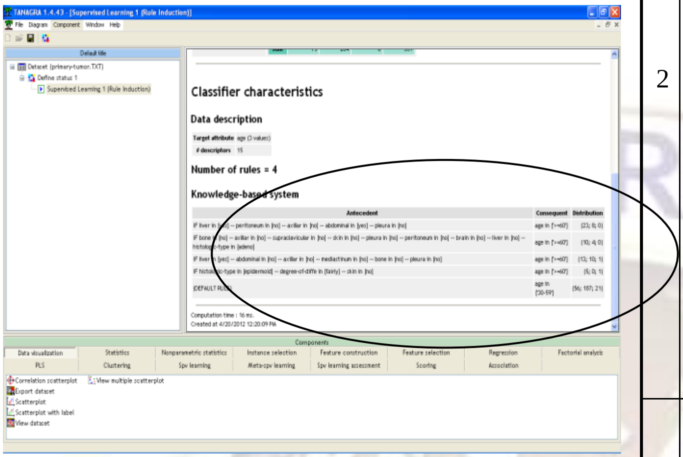
Our approach works as follows:

1. Start
2. Load a sample of records from the database that fits into the memory.
4. Apply Apriori algorithm to find the frequent itemsets with the minimum support threshold. Suppose S is set of the frequent item set generated by Apriori algorithm and th is the threshold.
5. Apply Genetic Algorithm for generation of all rules.
6. Set Q=Ø where Q is the output set, which contains the entire association rule.
7. Set the Input termination condition of genetic algorithm.
8. Represent each frequent itemset of S as binary encoding, string using the combination of representation specified in method above.
9. Select the two members (string) from the frequent item set using tournament method.
10. Apply Genetic Algorithm operators using multipoint crossover on the selected members (string) to generate the association rules.
11.Find confidence factor, completeness, comprehensibility and interestingness for x=>y each rule.
12. If generated rule is better than previous rule then
13. Set Q = Q U {x =>y}
14. If the desired number of generations is not completed, then go to Step 3
15. Decode the chromosomes in the final stored generations and get the generated rules.
16. Select rules based on comprehensibility and interestingness
17. Stop.

8. RESULTS AND DISCUSSION

Experiments were conducted using real world Primary-tumor dataset. The Primary-tumor database contains 18 attributes and 339 instances. Default Values of the parameters are Population size: 339, mutation rate=0.5,crossover rate=0.8,threshold=5.

Rule induction using Tanagra data mining software



Classifier characteristics

Data description

Target attribute	age (3 values)
# descriptors	15

Number of rules = 4

Knowledge-based system

Antecedent	Consequent	Distribution
IF liver in [yes] -- peritoneum in [no] -- axillar in [no] -- abdominal in [yes] -- pleura in [no]	age in ['>=60']	(23; 8; 0)
IF bone in [no] -- axillar in [no] -- supraclavicular in [no] -- skin in [no] -- pleura in [no] -- peritoneum in [no] -- brain in [no] -- liver in [no] -- histologic-type in [adeno]	age in ['>=60']	(10; 4; 0)
IF liver in [yes] -- abdominal in [no] -- axillar in [no] -- mediastinum in [no] -- bone in [no] -- pleura in [no]	age in ['>=60']	(13; 10; 1)
IF histologic-type in [epidermoid] -- degree-of-diffe in [fairly] -- skin in [no]	age in ['>=60']	(5; 0; 1)
(DEFAULT RULE)	age in ['30-59']	(56; 187; 21)

Computation time : 16 ms.
Created at 4/20/2012 12:20:09 PM

Fig 1 Tanagra produces association rules

Following table is the results of the experiment conducted. In these tests, different predictions were made by combining different attributes to determine a result

Table 1. Rules generated from Primary-tumor dataset.

Rule no.	Discovered rule (antecedent->consequent)		Acc/ CF	Comp	Inter	Compre	F
	antecedent	consequent					
1	IF liver in [yes] --peritoneum in [no]-- axillar in [no] --abdominal in [yes]--pleura in [no]	age in ['>=60']	11	3	13	0.8	1.4
2	IF bone in [no] --axillar in [no] --supraclavicular in [no] --skin in [no] --pleura in [no] --peritoneum in [no] -- brain in [no] --liver in [no] --histologic-type in [adeno]	age in ['>=60']	8	3	11	0.8	1.4
3	IF liver in [yes] --abdominal in [no] --axillar in [no] --mediastinum in [no] -- bone in [no] --pleura in [no]	age in ['>=60']	14	3	14	0.8	1.4
4	IF histologic-type in [epidermoid] --degree-of-diffe in [fairly] --skin in [no]	age in ['>=60']	0	1	127	1	1.9

(note:acc/cf-Predictive Accuracy/Confidence factor, comp-completeness,inter-interestingness,compre-comprehensibility, and F-fitness function)

9. CONCLUSION

We have used multi objective evolutionary framework for ARM, offers tremendous flexibility. Each rule is influenced by threshold, controlled by the user. Association rules that satisfy both a user-specified minimum confidence threshold and user-specified minimum support threshold are referred to as strong association rules, and are considered interesting. We have used tournament selection and multipoint crossover methods; it is flexible for large number of attributes in the database. We have adopted Michigan approach to represent the rules as chromosomes, where each chromosome represents a separate rule. We can use other technique to minimize the complexity of the genetic algorithm. Moreover, we tested the approach only with the numerical and categorical valued attributes. We also intend to extend the algorithm proposed in this paper to cope with continuous data.

REFERENCES

[1] K.J. Cios, W.Pedryc, R.W Swiniarki, “Data Mining Methods for Knowledge Discovery”, kluwer Academic Publishers, Boston, MA (2000).

[2] L.Jain, A.Abraham, R.Goldberg, “Evolutionary Multiobjective Optimization” , Second Edition, Springer (2005).

[3] Jochen Hipp, Ulrich G¨untzer and Udo Grimmer, “Data Quality Mining - Making a Virtue of Necessity”, In Proceedings of the 6th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD) 2001.

[4] R. Agrawal, R. Srikant, “Fast algorithms for mining association rules”, in Proceeding of the 20th Int’l Conference on Very Large Databases, Chile, 1994

[5] Ali Hadian,Mahdi Nasiri,Behrouz Minaei-Bidgoli “Clustering Based Multi-objective Rule Mining using Genetic Algorithm” International Journal of Digital Content Technology and its Applications, Volume 4,Number 1(2010)

[6] R. Agrawal, R. Srikant, Fast algorithms for mining association rules, in: Proceedings of the 20th International Conference on Very Large Databases, Santiago, Chile, 1994.

[7] R. Agrawal, T. Imielinski, A. Swami, Mining association rules between sets of items in large databases. in: Proceedings of ACM SIGMOD Conference on Management of Data, 1993, pp. 207–216.

[8] Houtsma and M. Swami, Set-oriented mining of association rules, Research Report, 1993.

[9] D.I. Lin and Z.M. Kedem, Pincer-search: an efficient algorithm for discovering the maximal frequent set, Proceedings of 6th European Conference on Extending Database Technology,1998.

[10] S. Brin, et al., Dynamic itemset counting and implication rules for market basket data, in: Proceedings of ACM SIGMOD International Conference on Management of Data, Tucson,AZ, 1997.

[11] M.V. Fedelis, Discovering comprehensible classification rules with a genetic algorithm, in: Proceedings of Congress on Evolutionary Computation, 2000

[12] A.A. Freitas, Data Mining and Knowledge Discovery with Evolutionary Algorithms, Springer-Verlag, New York, 2002.

[13] E. Noda, Discovering interesting prediction rules with a genetic algorithm, Proceedings of the Congress on Evolutionary Computation 1999, pp. 1322–1329.

[14] Peter P. Wakabi-Waiswa, Venansius Baryamureeba, and Karunkaran sarukesi ,Optimized Association Rule Mining With Genetic Algorithm, Proceedings of Seventh International conference in natural computation IEEE 2011,1116-1120.

[15] x.yan,CH,zhang,and sh hang,”Genetic algorithm-based strategy for identifying association rules without specifying actual minimum support”,Eleviser Expert Systems with applications,vol,36,no,2,208 pp3066-3076

[16] Z. Michalewicz, “Genetic Algorithms + Data Structure = Evolution Programs”, Springer-Verlag, Berlin, 1994.

[17] GHOSH, A., NATH, B. 2004. Multi-objective rule mining using genetic algorithms.InformationSciences 163 pp 123-133.

[18] A.A. Freitas, A survey of evolutionary algorithms for data mining and knowledge discovery, in: A. Ghosh, S. Tsutsui (Eds.), Advances in Evolutionary Computing, Springer-Verlag, NewYork, 2003, pp. 819–845.

[19] DEHURI, S., JAGADEV, A. K., GHOSH A. AND MALL R. 2006. Multi-objective Genetic Algorithm for Association Rule Mining Using a Homogeneous Dedicated Cluster of Workstations. American Journal of Applied Sciences 3 (11):2006, 2086-2095, ISSN 1546-9239