Performance Analysis of Genetic Algorithm for Mining Association Rules

Indira K[1] and Kanmani S[2]

**[1] Research Scholar, Department of CSE, Pondicherry Engineering College
Puducherry, 605014, India**

**[2] Professor, Department of IT, Pondicherry Engineering College
Puducherry, 605014, India**

Abstract

Association rule (AR) mining is a data mining task that attempts to discover interesting patterns or relationships between data in large databases. Genetic algorithm (GA) based on evolution principles has found its strong base in mining ARs. This paper analyzes the performance of GA in Mining ARs effectively based on the variations and modification in GA parameters. The recent works in the past seven years for mining association rules using genetic algorithm is considered for the analysis. Genetic algorithm has proved to generate more accurate results when compared to other formal methods available. The fitness function, crossover rate, and mutation rate parameters are proven to be the primary parameters involved in implementation of genetic algorithm. Variations and modifications introduced in primary GA parameters are found to have greater impact in increasing the accuracy of the system moderately. The speedup of the system is found to increase when the selection and fitness function are altered.

Keywords: *Association rule, Genetic Algorithm, GA parameters, Accuracy, Speedup.*

1. Introduction

Data mining also referred as knowledge discovery in databases, is a process of nontrivial extraction of implicit, previously unknown and potential information from data in large databases [23]. The knowledge or information mined from databases is always expressed as association rules. Association rule mining is one of the important research areas in data mining [20]. Association rule mining describes the relationship among itemsets present in databases.

Mining of association rule were implemented using algorithms like Apriori, Eclat, FP growth tree etc. These algorithms traverse the databases repeatedly. The Input output overhead and computational complexity of these systems is more and cannot meet the requirements of large-scale database mining.

Genetic algorithm is a promising and upcoming research area for mining association rules. Genetic algorithm [25] is a method which simulates search of evolutional process. Genetic algorithm can dispose large-scale data gathered in a lot. It is widely applied in mining association rules. Genetic algorithms are typically implemented using computer simulations where optimization is the main criteria for solving the problem. For this problem, members of a space of candidate solutions, called individuals, are represented using abstract representations called chromosomes. The GA consists of an iterative process that evolves a working set of individuals called a population toward an objective function, or fitness function. Traditionally, solutions are represented using fixed length strings, especially binary strings, but alternative encodings have also been developed.

As many works have been carried out on mining association rules with genetic algorithms this paper surveys the existing work on application of Genetic algorithm in mining association rules and analyzes the performance of the methodology adopted . This paper is organized as follows. Section 2 discusses the preliminaries of association rule and Genetic algorithm for mining association rules. Section 3 analyzes the existing work for mining association rules based on genetic algorithms, section 4 lists the observations from the performance analysis on GA for mining AR followed by conclusion in section 5.

2. Preliminaries

The preliminaries of concept are explained in this section. The concept of association rule is explained first followed by the Genetic algorithm for association rule mining and then the Genetic operators.

2.1. Association Rules and Association Rule Mining

Association rules are if and then statements that help uncover relationships between seemingly unrelated data in a relational database or other information repository. An association rule has two parts, an antecedent (if) and a consequent (then). Association rule [2] is expressed as X=>Y, where X is the antecedent and Y is the consequent.

Each association rule has two quality measurements, support and confidence. Support implies frequency of occurring patterns, and confidence means the strength of implication and is defined as follows:

An itemset, X, in a transaction database, D, has a support, denoted
in D containing X. Or

$$\boldsymbol{sup}(\boldsymbol{x}) = \frac{\text{No,of transactions containing X}}{\text{total No.of transactions}} \qquad (1)$$

The confidence of a rule X => Y, written as conf(X=>Y), is defined as

$$conf(x) = \frac{\text{sup}(\text{X}\cup\text{Y})}{\text{sup(X)}} \qquad (2)$$

2.2 Genetic Algorithm for Association Rule Mining

Genetic Algorithm (GA) is an adaptive heuristic search algorithm based on the evolutionary ideas of natural selection and genetics. The evolutionary process of a GA [3] is a highly simplified and stylized simulation of the biological version. It starts from a population of individuals randomly generated according to some probability distribution, usually uniform and updates this population in steps called generations. In each generation, multiple individuals are randomly selected from the current population based upon some application of fitness, bred using crossover, and modified through mutation to form a new population.

A. **[Start]** Generate random population of *n* chromosomes

B. **[Fitness]** Evaluate the fitness *f(x)* of each chromosome *x* in the population

C. **[New population]** Create a new population by repeating the following steps until the new population is complete

i. **[Selection]** Select two parent chromosomes from a population according to their fitness

ii. **[Crossover]** With a crossover probability cross over the parents to form a new offspring (children)

iii. **[Mutation]** With a mutation probability mutate new offspring at each locus (position in chromosome)

iv. **[Accepting]** Place new offspring in a new population

D. **[Replace]** Use new generated population for a further run of algorithm

E. **[Test]** If the end condition is satisfied, **stop**, and return the best solution in current population

F. **[Loop]** Go to step **B**

2.3. Genetic Operators

The GA maintains a population of n chromosomes (solutions) with associated fitness values. Parents are selected to mate, on the basis of their fitness, producing offspring via a reproductive plan (mutation and crossover). Consequently highly fit solutions are given more opportunities to reproduce(selected for next generation), so that offspring inherit characteristics from each parent. As parents mate and produce offspring, room must be made for the new arrivals since the population is kept at a static size (population size). In this way it is hoped that over successive generations better solutions will thrive while the least fit solutions die out. The representation scheme, Population Size, Crossover rate, Mutation rate, and fitness function and selection operator are the GA operators and are discussed below.

2.3.1. Encoding chromosomes

The process of representing the individual chromosomes is called encoding. The representation can be in the form of bits, numbers, trees, arrays, lists or any other objects. The encoding method adopted mainly depends on the problem being solved. The decision on best coding systems is a part of design of evaluation function. Some coding schemes are shown in table 1.

Table 1. Encoding schemes

Coding scheme	Chromosome
Binary Encoding	1 0 1 1 1 0 1 0 1 1 1 0 1 1 1 1 0 0 1 1 0 0 0 1
Octal Encoding	23147632 15754231
Hexadecimal encoding	9DA4 F34E
Value Encoding	3.1234 5.3214 7.9812 2.1567 AHGYNBRYGUJJHUYIUYIU (back) (right) (forward) (left)

2.3.2. Fitness function

The fitness of an individual in a genetic algorithm is the value of an objective function for its phenotype. For calculating fitness, the chromosome has to be first decoded and the objective function has to be evaluated. The fitness not only indicates how good the solution is, but also corresponds to how close the chromosome is to the optimal one.

Each solution or chromosome needs to be awarded a figure of merit, to indicate how close it came to meeting the overall specification, and this is generated by applying the fitness function to the test, or simulation, results obtained from that solution.

2.3.3. Selection operator

The process of choosing the two parents from mating pool for reproduction is characterized by the selection operator. The selection is based on the fitness of the individual. Higher the fitness, more the chance of the individual being selected. The convergence of the algorithm largely depends upon the chromosomes being selected for reproduction.

Some popular selection methods are roulette wheel selection, random selection, tournament selection, universal sampling etc. Elitism is introduced to eliminate the chance of losing information during mutation.

2.3.4. Crossover operator

Crossover is the process of taking two parent solutions and producing from them a two new offspring. Crossover is a recombination operator that proceeds in three steps:

- The reproduction operator selects at random a pair of two individual strings for the mating
- A cross site is selected at random along the string length
- Finally, the position values are swapped between the two strings following the cross site

Single point crossover, two point crossover, multipoint crossover, uniform crossover etc are the different crossover techniques adopted.

2.3.5. Mutation operator

Mutation is a genetic operator that alters one or more gene values in a chromosome from its initial state. Mutation of a bit involves flipping a bit, changing 0 to 1 and vice-versa. Mutation plays the role of recovering the lost genetic materials as well as randomly disturbing genetic information. It is an insurance policy against the irreversible loss of genetic material. Mutation helps to escape from local minima's trap and maintains diversity in the population.

3. Mining Association rules using Genetic algorithm

The existing work on mining association rules based on genetic algorithms is taken up for performance analysis in this section. Mining of both the prediction rules and classification rules using genetic algorithm is taken up for analysis.

The analysis is carried out based on the genetic parameters and methodology adopted. The datasets used for both rules vary from medical, business, education, finance, administration, problem solving etc. Both benchmark datasets and synthetic datasets are bound to give better accuracy when GA is adopted for rule mining.

From the works carried out it is significant that in addition to modifying the parameters the changes when made in algorithm based on the problem under study without altering the concept enhances the efficiency of the system. If the accuracy of the rules generated is not up to the expected value then data modification process [10] is introduced. In such cases the unselected attributes are taken and classified till expected accuracy is attained. This attribute is added to the training set for further generations. The accuracy of the system is enhanced through data modification process. The insert or remove operator when introduced [2] controls the size of the rules evolved and hence influences the comprehensibility of the rules. Rule pruning [4] removes the irrelevant attributes included in the rule over evolution thereby reducing the number of attributes involved in processing.

3.1. Representation

The concept of fixed length of chromosomes is adopted in [6]. Binary string of representation is followed in [7, 10, 18, 21]. Array encoding [4, 12, 13, 20] allows to store variable number of attributes. It enables mutation and crossover to be achieved at file level thereby speeding up the system. The significance of this system is that the number of attributes need not be fixed earlier. Attributes addition and removal is easier. Array implementation facilitates the easy implementation of Genetic operators.

Multilevel phenotype structure [9] is simple and helps to expand indicators for further studies. The chromosome's length in first level is the number of indicators encoded in this chromosome. The value of each gene indicates whether a specific kind of relation to this indicator, exists or not. The changes introduced in representation of rules in rule space [11] changes the fitness function values and this could be used to estimate the distance to the global maximum. By individual representation [17] each individual is directly defined as an expression formed by a conjunction of predicates over some attributes. Individual representation as set of rules (ruleset)[19] facilitates the reduction of search space size.

3.2. Fitness Function

The fitness criteria for classification rule is found to be carried out in almost all studies depends upon two major factors namely comprehensibility metric and confidence factor. Comprehensibility metric is the count of number of rules and number of conditions in these rules.

If a rule can have at most 'A_c' conditions, the comprehensibility metric Comp(R) of the rule 'R' can be defined as

$$\mathrm{Comp(R\,)} = 1 - \ (\mathrm{NC(R)} \,/\, \mathrm{AC}) \qquad (3)$$

Confidence factor Con(R) is measured by

$$\mathrm{Con(R)} = \frac{\mathrm{SUP(AUC)}}{\mathrm{SUP\ (A)}} \qquad (4)$$

where A is the number of rules satisfying the condition and |AUC| is the number of rules satisfying both the antecedent and consequent. Weight measures are added to the above two factors and fitness is a measure combining these factors in effective way.

The fitness function of prediction rules depend upon the support value 'sup' of the itemset under analysis and user defined minimum support factor 'minsup'[18, 20].

$$\mathrm{f(x)} = \frac{\mathrm{sup(x)}}{\mathrm{minsup}} \qquad (5)$$

Support and confidence values of the itemset under study is applied for calculating the fitness function [13] as indicated in equation below.

$$f(x) = a * \sup\ (x) + \ b\ * con\ (x) \qquad (6)$$

Fitness function [6] is defined as follows.

$$Let\ x\ be\ the\ rule\ t \to s,$$

$$f(x) = \begin{cases} -x, & x < 0 \\ x, & x \geq 0 \end{cases}$$

$$\mathrm{f(x)} = \begin{cases} \mathrm{k}\left(1 - \frac{\mathrm{f(x)}}{\mathrm{n}}\right) + \ \mathrm{abs}(\eta(\mathrm{x})), & \delta(\mathrm{t})\delta(\mathrm{s}) \neq 0 \\ 0\,, & (\mathrm{t})\delta(\mathrm{s}) = 0 \end{cases} \qquad (7)$$

Where *f(x)* the number of genet that does not include'0' in the chromosome of rule *x*, *n* denotes the total number of attributes in the system $abs(\eta(x))$ denotes the absolute value of the strength of implication of rule *x* and *k* is the adjustment parameter between 0 and 1. $\delta(t)$ and $\delta(s)$ are strength of implication for *t* and *s*.
The absolute value of the strength of implication of rule *x* can control a chromosome evolution along the direction that the strength of implication becomes strong, and can control the rule chromosome direction towards the simplest rule evolution.

The minimum support and minimum confidence factor [8] are not specified for fitness function. The support factor alone decides on the fitness value. The fitness function is defined as

$$0.5 \leq \frac{(1+\sup\,(XUY))^2}{1+\sup\,(X)} \leq 2 \qquad (8)$$

The Interestingness factor (INF) and completeness factor (CF) are used for evaluating the rules [12]. The evaluation is based on the confusion matrix created with classification labels namely true positive(TP), true negative(TN), false positive(FP) and false negative(FN).

$$\text{Interestingess Factor (INF)} = \frac{\mathrm{TP}}{\mathrm{TP-FP}} \qquad (9)$$

$$\text{Completeness Factor (CF)} = \frac{\mathrm{TP}}{\mathrm{TP-FN}} \qquad (10)$$

Pareto dominance based fitness factor [2] is introduced to find the single global solution or multi objective problem depending on the non dominance criteria in each generation. Pareto based methods measure individual's fitness according to their dominance property. The non dominant individuals in the population are regarded as fittest regardless of their single objective values. In [22] evaluation based on all confidence and collective strength factor enables creating quality rules and avoids infeasible rule generation.

Predictive accuracy and sensitivity test [9] are introduced for measuring rulesets. Sensitivity test measures the performance of the system by varying the GA parameters and compare the results for the sensitivity of these parameters. The sustainable index, creditable index and inclusive index finds its place [4] for generating the evaluation function. These indexes are the measures based on the number of rules satisfying antecedent, number of rues satisfying the consequent, and number of cases satisfying the particular rule. The recall and precision parameters [19] are used in calculating the fitness evaluation.

$$\mathrm{recall} = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \qquad (11)$$

$$\text{precision} = \frac{\text{TP}}{\text{TP+FN}} \quad (12)$$

3.3. Selection Operator

The strength of implication [6] extracts proper rules for reproduction and hence increases the efficiency of the system. It controls the rule chromosome direction towards simple rule evolution. The selection based on fitness criteria [12, 13, 20] tends to increase the efficiency and speeds up the system. In this method boundaries are set by the user to values closer to 1 as to maintain selection of high quality rules.

When Immune concept [16] is adopted for selection it maintains the diversity of individuals in population. This avoids premature convergence of the system. Selection strategy based on self adaptive suppression and promotion [18] ensures the individuals which have greater fitness values to be retained for further generations. It also ensures the diversity of the population. The concentration plays suppressive role and avoid premature.

Niched Pareto based selection [1, 3] uses standard deviation function when the difference in absolute count is less. This promotes the accurate selection of candidates for reproduction and saves system time too. Roulette wheel based selection method is adopted in [2]. Parents are selected according to their fitness. The better the chromosomes are, more are the chances for them to be selected. Tournament based selection [5] enhances the random selection of candidates for the process of symbiogenesis. Elitist recombination selection [17] retains the appropriate rule for next generation bypassing the fitness criteria.

3.4. Crossover operator

To avoid invalid chromosome production order-1 crossover [8] is adopted. One segment is selected from both parents equally and replaced into each other offspring. Then the offspring copies information from corresponding parent that does not exist starting from right of the segment. Generation of rules with high number of attributes [12] is made possible with single crossover operator. Attributes are selected randomly from antecedent and consequent of the rules. Then exchange occurs to generate new offspring.

R_1 : AB=>CD Reproduces R_1' : F =>CHJ
R_2 : EFG=> HIJ R_2' : EGAB => ID

Random and heuristic crossover [14] helps in achieving diversity of the group and obtains more frequent itemsets quickly. When the crossover operator is made dynamic process of evolution [15], it helps in evolving the new population based on last generation population. This is found to enhance the diversity of colony. Multipoint crossover [18] classifies the domains of each attribute into a group and sets crossover point based on continuous attributes.

Single point uniform crossover [2] otherwise named hybrid crossover combines the best attributes of single point and uniform crossover. In single point crossover the swapping is done in adjacent genes and in uniform crossover the swapping is performed on genes in distributed location. The advantages of both systems are combined, thereby creating diverse population. N point crossover is adopted in [4,20,21] enables crossover on attributes at n different points as per the crossover points set. The crossover when altered to symbiotic combination [5] results in creation of new ruleset combining the attributes of both the parents rather resulting in invalid rules. Symbiotic combination operator takes two partially specified chromosomes and makes an offspring with the sum of their characteristics. In [9] crossover is carried out on the first level genes of chromosomes rather at all level over the crossover point indicated. This prevents generation of infeasible rules. If same attributes are present [17] in both the parents selected then crossover is attained at the same attributes randomly selected. In case of absence of common attributes the crossover is performed on randomly selected attributes.

Best class crossover (BCX) [20] based on crossover matrix created from individuals' fitness and random number uses specific domain knowledge and links to individuals more effectively.

3.5. Mutation operator

In most cases the mutation operator remains fixed to the probability P_m. The mutation rate [8] prevents generation of invalid chromosomes. The mutation in such cases brings about changes in confidence of the rules alone thereby maintains the support intact and hence the fitness function. Either the antecedent or consequent [12] alone is selected or mutation is carried out on that attribute alone. This avoids evolution of invalid chromosomes.

Adaptive mutation rate [13] helps in attaining local optimal solution. Adaptive mutation is based on the fitness of individual in present and previous generation and the highest fitness in individual stocks. It avoids excessive variation in fitness at earlier generation thereby avoiding non convergence. This enhances the efficiency of the genetic algorithm. Heuristic mutation [14] generates more new frequent itemsets. By execution of heuristic operators over generation some current maximal frequent itemsets are created.

When number of attributes in the dataset is large then the chromosome length becomes larger. In such cases multipoint mutation [21] is adopted for efficient reproduction of offspring's. Mutation matrix is used in adaptive GA [20] so as to avoid usage of external parameters. In this study a random number p is generated by the system and based on this number and the matrix the mutation is performed.

In [21] multilevel mutation is performed on all levels of chromosomes while crossover is on top level of phenotype structure alone. Mutation is carried out at attribute level [17] where the attributes are either deleted or mutated based on the probability P_m and the fitness of the individual chromosome under analysis. Directed mutation [19] based on mutation matrix generated from fitness values of individuals and random number helps in formulating the mutation rate. This evolution process increases the mutation rate if the selected rule is of low quality. This enables the production of quality rules.

4. Performance Analysis

Traditional rule mining methods, are usually accurate, but have brittle operations. Genetic algorithms on the other hand provide a robust and efficient approach to explore large search space. In recent years numerous works have been carried out using genetic algorithm for mining ARs. Selected works in the past seven years for mining ARs using genetic algorithm have been studied and performance analysis of these methods are presented in this section.

From analysis it is observed that the Genetic parameters namely selection, crossover, mutation and fitness function when fixed to optimum or changes introduced enhances the efficiency of the system. These parameters are considered to be the primary parameters.

The other GA parameters such as population size, selection methodology, encoding scheme and termination condition have least significance on accuracy of the rules mined. The GA is found to produce optimum results for both Association rule mining and Classification rule mining. The GA extracts association rules from incremental database with single pass of the whole dataset whereas other methods go through the dataset many times to produce the result.

Effects of Genetic Operators on accuracy are

- Using selection alone will tend to fill the population with copies of the best individual from the population
- Using selection and crossover operators will tend to cause the algorithms to converge on a good but sub-optimal solution
- Using mutation alone induces a random walk through the search space.
- Using selection and mutation creates a parallel, noise-tolerant, hill climbing algorithm

When the primary GA parameters are modified in accordance with the dataset used for mining ARs, then the accuracy of rules generated is increased. The accuracy of the mined rules is mainly based on fitness function, crossover operator and mutation operator. Self adaptive mechanism or evolution process when introduce in GA parameters increases the accuracy marginally. The fitness function is the key for selecting accurate rules into the next generation.

The mutation operator and crossover operator when designed effectively will avoid premature convergence thereby increasing the efficiency of rules generated. The chromosomes created after crossover should ensure that they should not violate the existing chromosomes. Hence the crossover operator is designed accordingly. The comprehensibility factor tends to have major role in fitness function. The support and confidence factor based fitness function is noted to have significance in survey. The Pareto based ranking dominance is also adopted for fitness functions to avoid premature convergence.

The crossover operator when fixed to optimum value converges the results early thereby speeding up the results. Hence different crossover operators are implemented in the papers. The mutation factor alters the chromosomes. So in order to have valid rules the mutation factor is set up with significant analysis to maintain the validity of rules mined. The measures for the validity of the rules mined are found to be similar in almost all the works.

The predictive accuracy based on comprehensibility metric and confidence factor is applied in more than two third of the work taken for this analysis. The interestingness measure, strength of implication and number of rules generated for the given threshold of confidence and support were noted as a measure of rule set quality.

The observation based on predictive accuracy is listed in table 2.

Table 2. Predictive Accuracy

Dataset	Predictive Accuracy	Method Applied
Nursery	89	Elitist multiobjective GA [2]
Adult	86	Elitist multiobjective GA [2]
Wisconsin breast cancer	98.15	GA with information entropy [4]
Wisconsin breast cancer	96.14 to 96.99	GA with data modification process [10]
Wisconsin breast cancer.	72.62 to 91.6	Incremental Association Mining [16]
Tic tac toe	97.86	GA with information entropy [4]
Dermatology	95.61	GA with information entropy [4]
Cleveland heart disease	63.58	GA with information entropy [4]
Stock trading data	90-100	GA-ACR [9]
Iris	99.1	CAREX [19]
Diabetics	76.43	CAREX [19]
Glass	83.54	CAREX [19]
Wine	100	CAREX [19]
Wisconsin Diagnostic breast cancer.	78.57-95.16	Incremental Association Mining [16]
Wisconsin Prognostic breast cancer.	73.33-76.19	Incremental Association Mining [16]

Genetic algorithm gives promising results for mining association rules when compared to rules mined by other methods indicated by the works taken up for analysis.

Accuracy achieved ranges from as low as 63.58 to maximum of 100 percent. The dataset containing attributes with wide range of values for e.g. age results to less accurate rues mined compared to attributes having narrow range values range This could be noted from medical dataset where age is part of the attribute generates rules with less accuracy.

Analysis based on number of rules generated by the methods is listed in table 3. The execution time mining ARs could be decreased considerably by altering the factors involved while bringing in minimum changes in accuracy. The number of rules generated usually depends on the support factor set. Based on the perspective or objective for the system under study the number of rules generated is varied. For the adult dataset when method [17] generates five rules, whereas method [22] generates around three hundred and fifty rules.

Table 3. Number of rules generated

Dataset	Number of rules Mined	Method Applied
Balance scale	34	ARMMGA [8]
Nursery	4	ARMMGA [8]
Nursery	5	GA based Classification [17]
Monk's Problems	4	ARMMGA [8]
Solar flare	23	ARMMGA [8]
SPECT heart	23	ARMMGA [8]
Car Evaluation	11	M-GARM with fitness threshold 0.85 [12]
Group data of finance service	3-10	GA based on evolution strategy [15]
Company's daily record of API	3	IOGA [16]
Adult	5	A based Classification of R. CF above 0.6 [17]
Adult	Around 350	GA for prioritization of rules [22]
Abalone	13	MAR-IGA [18]
IBM QUEST synthetic database	40-50	AGA [20]
Chess	Around 470	GA for prioritization of rules [22]
Wine	Around 230	GA for prioritization of rules [22]
Zoo	Around 325	GA for prioritization of rules [22]

Table 4lists the comparison of the methods based on execution time. Execution time for mining of AR based on genetic algorithm is less over methods done using conventional methods.

Table 4. Execution Time

Dataset	Execution time (ms)	Method
Balance Scale	8	GEA- DM [11]
Chess	768	GEA- DM [11]
Car Evaluation	9	GEA- DM [11]
Nursery	132	GEA- DM [11]
Nursery	286.38	INPGA [1]
Tic Tac toe	42	GEA- DM [11]
Adult	1844	GEA- DM [11]
Iris	6.75	INPGA [1]
Iris	40	SEA [5]
Vote	89	SEA [5]

Wine	98	SEA [5]
KDD99	7012	SEA [5]
Student achievement database	20-10	Improved GA for varied support [13]
Abalone	48	MAR-IGA [18]

From the performance analysis carried out the further exploration for mining ARs using GA could be done by analysis on other domains to be taken up. Methods to deal with noisy, imprecise, and uncertain information could be further explored. Careful selection of attributes in preprocessing step might result in better predictive accuracy. Further enhancement of self adaptive mechanism might lead to better performance. Other interesting measures could be incorporated.

5. Conclusion

Performance analysis on mining association rules using GA was performed in recent researches on mining ARs using GA. The use of GA has resulted in both predictive and classification ARs with higher predictive accuracy. Fitness function, Crossover rate and mutation rate influences the accuracy more than other GA parameters. The right indicators when used in fitness function generated high quality rules. This avoids generation of infeasible rule in ruleset discovered. The fitness function is found to be the key for selecting accurate rules into the next generation. The cross over rate and mutation rate when made optimum avoids premature convergence of the algorithm. This leads to the generation of feasible rules.

GA algorithm is found have produced better results in all type of datasets ranging from medicine to problem solving. The selection method plays major role in reducing the execution time by selecting right parents for reproduction. The right representation scheme adopted tends to speed up the system. The capability of GA to scan the dataset quickly when designed effectively reduces the execution time. Self adaptive mechanism or evolution process when introduced in GA parameters increases the accuracy marginally.

References

[1]. Junlin Lu, Fan Yang, Momo Li, Lizhen Wang, “Multi-objective rule discovery using Niched Pareto genetic algorithm”, in Third IEEE international conference on measuring technology and mechatronics automation, 2011, pp. 657-661.

[2]. S. Dehuri, S. Patnaik, A. Ghosh, R. Mall, “Application of elitist multi-objective genetic algorithm for classification rule generation”, Applied Soft Computing, 2008, pp. 477–487.

[3]. Dehuri.S, Mall. R, “Predictive and comprehensible rule discovery using a multiobjective genetic algorithm”, Knowledge based systems, Elsevier, vol 19, 2006, pp. 413-421.

[4]. Hua Tang, Jun u, “A hybrid algorithm combines with Genetic Algorithm with information entropy for data mining”, in second IEEE international conference on Industrial electronics and applications, 2007, pp.753-757.

[5]. Ramin Halavathi, Saee Bagheri Shouraki, Pooya Esfandiar, Sima Lotfi, “Rule based classifier using symbiotic Evolutionary algorithm”, in 19th IEEE international conference on tools and artificial intelligence, 2007, pp.458-461.

[6]. Zhou Jun ,Li Shu-you Mei, Hong-yan Liu, Hai-xia, “A Method for Finding Implicating Rules Based on the Genetic Algorithm”, in Third International Conference on Natural Computation, ICNC 2007, pp. 400-405.

[7]. Anandhavalli M., Suraj Kumar Sudhanshu, Ayush Kumar and Ghose M.K., “Optimized association rule mining using genetic algorithm”, Advances in Information Mining, ISSN: 0975–3265, Volume 1, Issue 2, 2009, pp. 01-04.

[8]. Hamid Reza Qodmanan , Mahdi Nasiri, Behrouz Minaei-Bidgoli, “Multi objective association rule mining with genetic algorithm without specifying minimum support and minimum confidence", Expert Systems with Applications, 2011, pp. 288–298.

[9]. Ya Wen Chang Chien , Yen-Liang Chen, “Mining associative classification rules with stock trading data – A GA-based method”, Knowledge-Based Systems , 2010: 605–614.

[10]. Ta-Cheng Chen, Tung-Chou Hsu, “GA based approach for mining breast cancer pattern”, Expert Systems with Applications, 2006, pp. 674–681.

[11]. Zhan-min Wang, Hong-liang Wang, Du-wa Cui, “A growing evolutionary algorithm for data mining”, in 2nd IEEE international conference on information engineering and computer science, 2010, pp.01-04.

[12]. Avendano J. Christian, Gutierrez P Martin, “Optimization of association rules with genetic algorithms”, in 29th IEEE international conference of the Chilean computer science society, 2010, pp.193-197.

[13]. Hong Guo, Ya Zhou, “An algorithm for mining association rules based on improved genetic algorithm and its applications”, in Third IEEE international conference on Genetic and evolutionary computing, 2009, pp.117-120.

[14]. Wenxiang Dou, Jinglu Hu, Kotaro Hirasawa and Gengfeng Wu, “Quick Response Data Mining Model Using Genetic Algorithm” , in SICE Annual Conference 2008, pp.1214-1219.

[15]. Xiaoyuan Zhu, Yongquan Yu, Xueyan Guo, “Genetic Algorithm based on Evolution Strategy and the Application in Data Mining”, in First IEEE International Workshop on Education Technology and Computer Science, 2009, pp. 848-852.

[16]. Genxiang Zhang, Haishan Chen, “Immune Optimization based Genetic Algorithm for incremental association rules mining”, in International Conference on Artificial Intelligence and Computational Intelligence, 2009, pp. 341-345.

[17]. Xian Jun Shi, Hong Lei, “A genetic algorithm based approach for classification rule discovery”, in IEEE International conference on information management,

innovation management and industrial engineering, 2008, pp.175-178.

[18]. Guangjun Yang, “Mining association rules from data with hybrid attributes based on immune genetic algorithm”, in 7th international conference on fuzzy systems and knowledge discovery,2010, pp.1446-1449.

[19]. Powel B.Myszkowski, “Coevolutionary Algorithm for Rule Induction”, in Proceedings of the IEEE International Multiconference on computer science and information technology, 2010, pp.73-79.

[20]. Min Wang, Qin Zou, Caihui Liu , “Multi-dimension Association Rule Mining Based on Adaptive Genetic Algorithm”, in IEEE International Conference on Uncertainty Reasoning and Knowledge Engineering, 2011, pp.150-153.

[21]. B. Nath, D K Bhattacharyya & A Ghosh, “Discovering Association Rules from Incremental Datasets”, International Journal of Computer Science & Communication, Vol 1, No. 2, July-December 2010, pp. 433-441.

[22]. M. Ramesh Kumar, Dr. K. Iyakutti, “Application of Genetic algorithms for the prioritization of Association Rules”, IJCA Special Issue on Artificial Intelligence Techniques - Novel Approaches & Practical Applications.AIT, 2011, pp. 1-3.

[23]. Z. Michalewicz, Genetic Algorithms + Data Structure = Evolution Programs, Berlin : Springer-Verlag, 1994.

[24]. Agrawal, T. Imielinski, and A.Swami. “Mining association rules between sets of items in large databases”. in the Proc. of the ACM SIGMOD International conference on Management of Data, 1993.

[25]. J.H. Holland, Adaptation in Natural and Artificial Systems, Ann Arbor : Univ.Michigan Press, MI, 1975.

K.Indira received her M.E. degree in 2005 from Department of Computer Science and Engineering, FEAT, Annamalai University, Chidambaram. She had been working as the Head of the Department of Computer Science for the past 12 years in Theivanai Ammal College for Women, Tamil Nadu, India from 1998 to 2007 and E.S. College of Engineering and Technology, Affiliated to Anna University , Chennai, India. Currently she is working towards her Ph.D degree in Evolutionary Algorithms applied for data Mining. Her areas of interest are Data Mining, Artificial Intelligence and Evolutionary Computing.

Dr. S. Kanmani received her B.E and M.E in Computer Science and Engineering from Bharathiyar University and Ph.D in Anna University, Chennai. She had been the faculty of Department of Computer Science and Engineering, Pondicherry Engineering College from 1992 onwards. Presently she is Professor in the Department of Information Technology, Pondicherry Engineering College. Her research interests are Software Engineering, Software testing, Object oriented system, and Data Mining. She is Member of Computer Society of India, ISTE and Institute of Engineers, India. She has published about 50 papers in various international conferences and journals.