Information Sciences 233 (2013) 15–24

Contents lists available at SciVerse ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Mining numerical association rules via multi-objective genetic algorithms

B. Minaei-Bidgoli *, R. Barmaki, M. Nasiri

Department of Computer Engineering, Iran University of Science and Technology, Narmak, Tehran, Iran

ARTICLE INFO

Article history:
Received 5 August 2009
Received in revised form 11 October 2011
Accepted 18 January 2013
Available online 8 February 2013

Keywords:
Numerical association rule
Multi-objective genetic algorithms
Confidence
Comprehensibility
Interestingness
Rough value

ABSTRACT

Association rule discovery is an ever increasing area of interest in data mining. Finding rules for attributes with numerical values is still a challenging point in the process of association rule discovery. Most of popular methods for association rule mining cannot be applied to the numerical data without data discretization. There have been efforts to resolve the problem of dealing with numeric data. These approaches suffer from problems which are discussed in this paper. This work proposes a multi-objective genetic algorithm approach for mining association rules for numerical data. Several measures are defined in order to determine more efficient rules. Three measures, confidence, interestingness, and comprehensibility have been used as different objectives for our multi objective optimization which is amplified with genetic algorithms approach. Finally, the best rules are obtained through Pareto optimality. This method is based on the notion of rough patterns that use rough values defined with upper and lower intervals to represent a range or set of values. Mutation and crossover operators give a powerful exploration ability to the method and allow it to find out the best intervals of existing numerical values. The experimental results show that the generated rules by this method are more appropriate – based on several different characteristics – than the similar approaches' results, and our method outperforms these methods.

1. Introduction

Data mining is the most instrumental tool in discovering knowledge from market basket transactions [12,1]. Nowadays, improvement in technology allows stores to collect various types of data about customer's market baskets. A basket shows items purchased by a customer at a specific time. Customers' purchases can be analyzed by vendors for future schematizations. Data mining technologies have thus been widely adopted to explore previously untapped knowledge to support decision making [32]. One of the most important applications of data mining is discovering association rules. This is one of the most significant methods for pattern recognition in unsupervised systems. This data mining method is very similar to that of people searching for gold in a large desert. Here, gold is an interesting rule which has not been discovered yet, and desert is a very huge dataset. Prevalent methods find all possible rules in the dataset. However, this can be considered a disadvantage because the large number of discovered rules makes them difficult to analyze. Some measures like support and confidence are used to indicate high quality rules. Most of the association rule algorithms are based on methods proposed by Agrawal et al. [3,2], Apriori [3], SETM [17], AIS [2] and Pincer search [22].

* Corresponding author.
E-mail addresses: b_minaei@iust.ac.ir (B. Minaei-Bidgoli), barmaki@comp.iust.ac.ir (R. Barmaki).

0020-0255/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2013.01.028

1.1. Background

In the early years, some optimization methods for association rule mining (ARM) have been proposed. This process was too resource consuming, especially when there is not enough physical memory available for the whole dataset. A solution to this problem is to use a genetic algorithm, which reduces both the cost and the time of rule discovery. Genetic algorithms, colony algorithms, evolutionary algorithms and particle swarm algorithms are instances of single objective association rule mining algorithms. A few of these algorithms have been used for multi objectives. Yan et al. proposed a method based on a genetic algorithm without considering minimum support [34]. The method uses an extension of elaborate encoding while relative confidence is the fitness function. A public search is performed based on genetic algorithms. Since the method does not use minimum support, a system automation procedure is used instead. It can be extended for quantitative-valued ARM. In [35] Qodmanan et al. has also proposed a multi objective association rule mining with genetic algorithm without specifying minimum support and minimum confidence which is working faster than the algorithm in [34]. In order to improve the algorithm's efficiency, it uses a generalized FP-tree. Here, only interesting rules with constant length are discovered [21].

Kaya and Alhajj proposed a genetic clustering method [18]. Chien et al. proposed a cluster based method for mining generalized fuzzy association rules [11]. Chen et al. proposed a cluster-based fuzzy-genetic mining method for association rules and membership functions [10]. Ghosh and Nath also used a genetic algorithm for ARM and proposed their approach based on multi objective genetic algorithms [15]. Their approach could only be used on market basket datasets and was unable to be used in numerical ARM. A multi-objective genetic fuzzy mining algorithm for extracting both membership functions and association rules from quantitative transactions is proposed in [9]. Here, like other similar approaches in fuzzy data mining, the association rules are shown using fuzzy membership functions.

Alatas et al. proposed a multi-objective differential evolution algorithm for mining numeric association rules [7]. Later, they proposed another numeric ARM method using rough particle swarm optimization (PSO) which had some improvements in performance and precision in comparison to the previous one [5]. They also proposed another numeric ARM method named chaos particle swarm algorithm [6]. Even though approaches that are based on classical PSO commonly have relatively high convergence speed, they have the risk of sticking in local optimum. Additionally, their method has the problem of data dependency.

Other research has been done in numerical association rules mining. In [19], an information-theoretic approach that uses a discretization approach to find numeric ARs (association rules) has been proposed. The proposed approach in [20] supplements the GA with an entropy based probabilistic initialization such that the initial population has more relevant and informative attributes. Some researchers partitioned the numeric data by means of fuzzy sets and the mined rules are named as fuzzy association rules [29]. Asadollahpoor et al. have suggested a fuzzy rule method that extracts numeric rules with dynamic memberships [36]. This algorithm has sufficient membership for each rule which is calculated with a genetic algorithm. Aumann and Lindell [8] used a numerical value as the criteria for inclusion in the ARs. The Idea was that an AR can be thought of as a population subset (the rule consequent) exhibiting some interesting behavior (the rule antecedent). Some researchers used geometric means to find numeric intervals for numeric values [14]. Another research is QuantMiner that uses a GA to mine numeric ARs [31]. However, QuantMiner uses predetermined rule templates that require the user to specify left-hand side and right-hand side attributes.

Fig. 1 which is inspired from [5] with a little changes, shows a taxonomy and an overall schema of proposed methods for numeric ARM. The first category refers to the methods which employ "Discretization". These techniques usually divide the domain of an attribute into several smaller intervals. For example, an attribute a_1, whose value is between 0 and 100, divides into 20 intervals $(0-5, 5-10, \ldots, 95-100)$ and in this manner several Boolean attributes are made. As an example, if the a_1's value was 49, the attribute which refers to the (45–50) interval becomes 1, and other attributes remain 0. The main problem of these methods is that their efficiency depends on the defined intervals, and defining appropriate intervals is difficult. In addition, discretization always results in some loss of information.

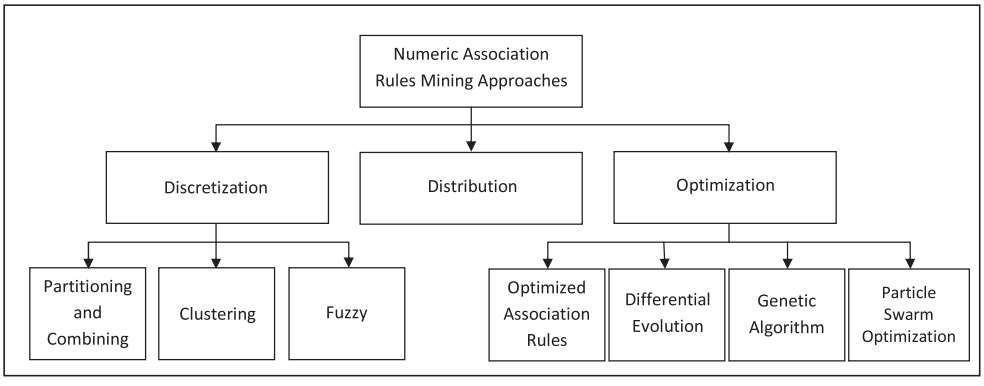

Fig. 1. Numeric ARs mining approaches [5].

The second category points to methods which use the distribution of a numerical value as the criteria for mining association rules. These approaches also suffer from the problem of a limited number of values in two sides of the association rules. In the third group, methods which are based on optimization methods are placed. These methods were briefly introduced earlier in this section.

1.2. Contributions of the article

Apriori based approaches suffer from fundamental defects. They usually include two distinct phases. The first phase finds frequent item sets, and the second applies minimum confidence in order to extract high confidence rules. So, it is not possible to use these methods in just one phase. Another problem is that they depend on data density or distribution, where the minimum support and confidence must be identified before running. In addition, Apriori based ARM methods are generally slow. When the number of attributes increases, the running time of these algorithms decreases exponentially. Eq. (1) shows the computation complexity of these methods. d shows the number of attributes, and N refers to the number of records in the dataset or transactions. Neither the rules with numeric attribute nor the rules in the form of $I_8I_{10}I_{12} \rightarrow I_4I_5I_9$ can be discovered by these methods.

$$\begin{aligned} \textit{time complexity} &= O(\textit{finding frequent item sets}) + O(\textit{rule generation}) \\ &= O(N * d * 2^d) + O\left(\sum_{k=1}^{d-1}\left[\binom{d}{k} \times \sum_{j=1}^{d-k}\binom{d-k}{j}\right]\right) \\ &= O(N * d * 2^d) + O(3^d - 2^{d+1} + 1) \\ &= O(N * d * 2^d) + O(3^d) \\ &= O(2^{d+1}) \end{aligned} \tag{1}$$

The complexity of genetic algorithms for problems with exponential complexity is in order of $O(n^2)$ [24]. In our method, the number of GA iterations is fixed. In this way, the complexity of the algorithm is equal to $O(N \times d)$ or $O(n^2)$.[1] This shows that the proposed method can be effectively used for datasets with too many records.

Two of the latest works which achieved considerable improvements in comparison with previous researches are the PSO based methods [5,6]. These are based on the optimization methods (third category of Fig. 1). Because of the mutation operator in genetic algorithms, generated rules in the proposed algorithm are more diverse than those created by PSO. PSO is more likely to stick in local optimum, but the probability of occurrence in a GA is much less than that in PSO, especially classic PSO. In the PSO based approaches, results are much more dependent on the initialization. Moreover, in the experimental section, we will compare our method with existing evolutionary based ARM approaches, and will show that this method outperforms similar methods.

One of the best ways for discretizing numeric attributes is using MOGA [30]. The methods that use constant coefficients for the different measures typically cannot find diverse rules. There are a considerable number of similar rules in the generated rules in this kind of method. We tried to solve this problem by applying MOGA. Additionally, because we use a Pareto based method in our approach, the user does not need to specify the measures coefficients. The user should be aware of the importance of each measure, in order to specify the coefficients. On the other hand, specifying the coefficients usually results in emphasizing some measures. In this manner, it is possible to produce rules having high confidence but very low support. Our Pareto based mechanism does not allow such rules to be produced.

Additionally, the implementation details of the embedded multi-objective genetic algorithm in the proposed method includes some innovative techniques aiming to improve the method's efficiency. Variations between the employed MOGA and the standard one are made in order to fit the method to the problem domain, and produce better results.

Our method employs three measures which enable it to generate more appropriate rules at the end. The end user (like a manger) is looking for a reasonable number (not too many, not too few) of rules with high confidence. These rules should contain some useful information and be clear to the user (comprehensible). Three objectives, confidence, interestingness and comprehensibility help our method to achieve these goals.

1.3. Outline

The rest of this paper is organized as follows. Section 2 presents a brief overview of multi-objective rule mining problems. Section 3 describes the proposed method. Section 4 briefly describes the used datasets and discusses the experimental results. Finally, Section 5 includes concluding remarks.

[1] The complexity of computing three measures which are used in our method (confidence, interestingness, and comprehensibility) is $O(N \times d)$. These measures are computed after production of the first generation of chromosomes. In each iteration, the chromosomes are compared with each other to determine the non-dominated ones. This process has the complexity of the number of GA population to the power of $2(O(p^2))$. The complexity of mutation and crossover operations (considering the types which are used in our method) are equal to the population number. These steps usually are taken within a bounded loop (c times), and the result of $N \times d$ is usually greater than cp^2. Therefore, the complexity of algorithm is $O(N \times d)$ or simply $O(n^2)$.

2. Multi-objective rule mining problems

Association rule mining is not a single objective problem and naturally is a multi objective one. Most of the traditional methods for ARM use 'Support' and 'Confidence' measures for rule mining. They usually contain two phases, the first one includes extracting 'Frequent-Item-sets' (FISs). In this phase the rules that do not satisfy a minimum support are eliminated. In the second phase, rules that do not satisfy minimum confidence are removed [26]. Support values show the portion of dataset's records where an association rule is true. Confidence measure is computed as shown in Eq. (2):

$$Confidence = SUP(A \cup C)/SUP(A) \tag{2}$$

Association rules are usually shown with $A \rightarrow C$, where 'A' stands for antecedent part of the rule and 'C' stands for the consequent part. This simply means that if A is present, then C will be present as well. But this measure cannot guarantee obtaining suitable association rules individually. In addition to having appropriate coverage and reliability, generated rules should also be interesting and comprehensible. This manner, the problem of ARM becomes a multi-objective problem instead of a single-objective one.

In addition to confidence measure, two other measures are used to mine more efficient association rules. In an association rule, if the number of conditions involved in the antecedent part is less than the number of conditions in the consequent part, the rule is more comprehensible. Therefore, we require a measure which is affected by the number of attributes in both parts of the rule [33]. Comprehensibility is the measure which is used for this purpose. It is computed through Eq. (3).

$$Comprehensibility = log(1 + |C|)/log(1 + |A \cup C|) \tag{3}$$

It is important that we extract only those rules that occur less frequently in the entire dataset. Such a surprising rule may be more interesting to the users; which is difficult to quantify like previous two measures. Interestingness has long been identified as an important issue in ARM [23]. It refers to finding rules that are interesting or useful to the user, not just all possible rules.

In some approaches, to find interestingness the entire dataset is divided based on each attribute presented in the consequent part. Since, different numbers of attributes can appear in the consequent part and because they are not predefined, this approach may not be feasible for association rule mining [33]. So, a new expression is defined which uses the support count of the antecedent and the consequent parts of the rules, and this expression is shown in Eq. (4).

$$Interestingness = [SUP(A \cup C)/SUP(A)] \times [SUP(A \cup C)/SUP(C)] \times [1 - SUP(A \cup C)/SUP(D)] \tag{4}$$

In this expression $|D|$ is total number of records in the dataset. The equation contains three parts. The first expression describes probability of generating the rule based on the antecedent part. The second expression shows the probability based on the consequent part, and the last one $(1 - SUP(A \cup C)/SUP(D))$ describes the probability of not generating the rule based on the whole dataset.[2] Therefore, a rule having a very high support count is measured as less interesting.

In fact, mining numeric ARs is a hard optimization problem rather than being a simple discretization one. That is why some researchers have characterized this as an optimization problem and tried to mine ARs using global optimization algorithms [5]. Genetic algorithms are one of the best global optimization algorithms and because of our problem's nature – having multiple objectives in ARM, a multi-objective genetic algorithm can be a useful approach.

3. The proposed method

In this section, we illustrate our approach for numerical association rule mining. This approach is based on a multi-objective genetic algorithm. The objectives which were introduced in previous section will be used in our multi-objective method in order to rank the chromosomes. Like the methods which rely on genetic algorithms, in the first step, the encoding or representation of applied chromosomes must be defined. Then, we explain how to compute the fitness value of chromosomes and mutation and crossover operations will be presented. Additionally, a separate population is used to improve the algorithm's performance.

3.1. Chromosome representation

There are two ways of representing rules in chromosomes. The first one is the 'Pittsburgh' approach. In this method, every chromosome represents a set of rules. The other one is 'Michigan' approach. Based on this method, each chromosome contains only one rule. The choice between these two approaches strongly depends on the kind of rules that are to be discovered. This is related to which kind of data mining task to be addressed. For example, in the case of classification, the goal is to evaluate the quality of the rule set as a whole, rather than the quality of a single rule. In other words, the interaction between the rules is important. In this case, the Pittsburgh approach seems more natural [13]. On the other hand, the Michigan approach might be more natural in other kinds of data mining tasks. In the Michigan approach the individuals are simpler and

[2] $(SUP(A \cup C)/SUP(D)$ shows the probability of generating rule based on the whole dataset and $(1 - SUP(A \cup C)/SUP(D))$ which is its complement shows the probability of not generating the rule.

syntactically shorter. This tends to reduce the time taken to compute the fitness function and simplifies the design of genetic operators [28].

The proposed method applies the Michigan approach for representing rules in chromosomes and a notion of rough patterns that use rough values defined with upper and lower intervals representing a range or set of values. With each attribute, two extra tag bits and two numbers are associated. The first number represents lower bound and the second number represents upper bound of the attribute. If the two tag bits are 00 then the attribute next to these two bits will appear in the antecedent part and if they are 11 then the attribute will appear in the consequent part. The other two combinations – 01 and 10 – will indicate the absence of the attribute in both of these parts. For example, $A00(2.1)(39.5)$ means that A is in the antecedent part and its range is from 2.1 to 39.5. As another example, note the expression $A00(2.1)(39.5)B11(1.4)(86.4)C00(12.4)(98.9)D01(-)(-)$ that simply is represented by $AC \rightarrow B$. This expression means that 'if (A is between 2.1 and 39.5) and (C is between 12.4 and 98.9) then (B will be between 1.4 and 86.4)'. We assumed that only four attributes A, B, C *and* D exist in the dataset. Fig. 2 shows this more clearly.

3.2. Fitness computation

To compute fitness value of generated chromosomes/rules in each iteration of the GA, Pareto theory is used. 'Pareto Optimality' is widely used to solve multi-objective optimization problems. In traditional single-objective (scalar) GA approaches, to fully search the state space, we can vary the weights or treat most of the objectives as varying constraints and optimize just the main objective, employing multiple runs to generate Pareto-optimal solutions sequentially. For MOEAs (Multi-objective evolutionary algorithms), we wish instead to generate all the Pareto-optimal solutions in a single run, for efficiency [25]. In this idea, we find a set of non-dominated solutions to solve the problem. A solution – rule in our method –, say a, is said to be dominated by another solution, say b, if and only if, with respect to all the corresponding objectives, the solution b is better than or equal to solution a, and b is strictly better than a in at least one objective. Here, solution b is called a non-dominated solution. If a solution remains non-dominated, compared to all other solutions, it will be a candidate for an optimal solution.

Fitness values are calculated using the solutions' ranks, which are calculated from the non-dominance property of the chromosomes. Three measures introduced earlier (confidence, interestingness and comprehensibility), are used to determine this property. Fig. 3 shows this idea more distinctly. The ranking step tries to find the non-dominated solutions, and those solutions are ranked as one. Among the rest of the chromosomes, if p_i individuals dominate a chromosome then its rank is assigned as $1 + p_i$. This process continues until all the chromosomes are ranked. Then, the fitness values are assigned to the chromosomes such that the chromosomes having the smallest rank get the highest fitness and the chromosomes having equivalent rank get the same fitness [15].

3.3. Mutation and crossover operators

Based on the described chromosome representation, now, crossover and mutation operators which are used in the proposed approach, can be defined. The place and value of chromosomes' attribute symbols $(A, B, \ldots)$ are fixed during both crossover and mutation operations. Two tag bits associated with each attribute are mutated through bit-flip mutation. This means that the existing bit pairs $(00, 01, 10, 11)$ are randomly transformed into each other. Two numbers showing upper and lower bounds of the attribute intervals, are randomly generated within the attribute range, such that the value of lower bound is smaller than the upper bound value. These values can be rounded to the nearest desired value (like the nearest integer). Considering the representation of chromosomes, for the crossover operation some arbitrary version of 'k-point crossover' can be used. In the experimental results section, we discuss that which kind of k-point crossover leads to a better result in our

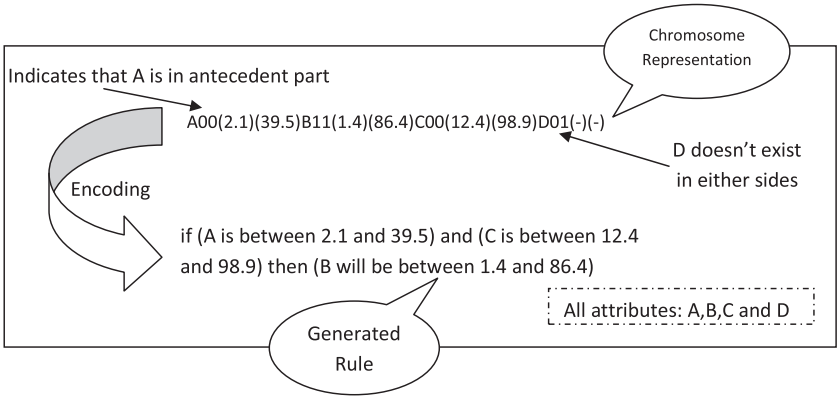

Fig. 2. Rule representation in a chromosome.

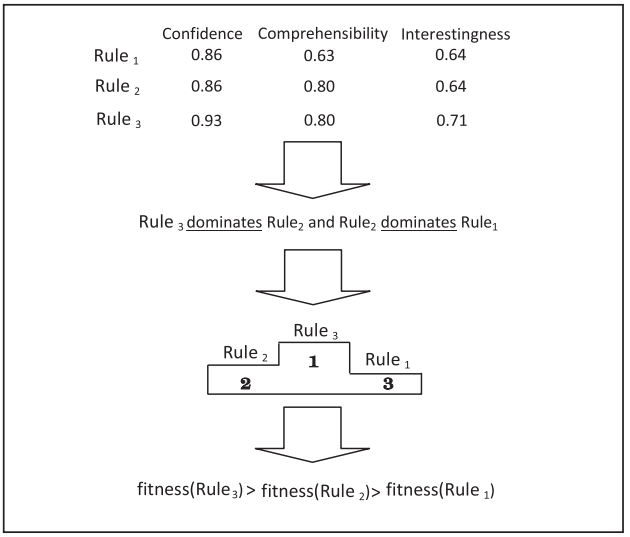

Fig. 3. Fitness computation process of three generated rules/chromosomes.

experiments. The termination criterion of the algorithm can be a predefined number of iterations or can be when the number of Pareto archive members almost remains fixed.

3.4. Some notes

As in a multi-objective optimization algorithm, we are looking for all solutions of the best compromise. In order to reach this goal, best solutions encountered over generations are filed into a secondary population called the 'Pareto archive'. In the selection process, solutions can also be selected from this Pareto archive. The main purpose of holding such a new population is to preserve efficient rules that are generated in each generation, and we are not going to miss them during genetic operations. This idea is similar to the "elitism" idea in the GA, which keeps the best chromosome/s of each population. In this population no genetic operation is performed. It simply contains only the non-dominated chromosomes of the previous generations. At the end of first generation, it will contain the non-dominated chromosomes of the first generation. After the next generation, it will contain those chromosomes, which are non-dominated among the current population as well as among the non-dominated solutions until the previous generation [15]. The algorithm continues until the termination criteria are met. The chromosomes which remain in the Pareto archive are the rules we are looking for. These chromosomes should be encoded, and the final rules will be generated.

To illustrate this method, its pseudo-code is presented in Fig. 4.

```
1: Initialize the population        //Based on the described chromosomes representation, bit tags are generated randomly, and the intervals are produced within the corresponding attribute boundaries.
2: Pareto archive t = Null
3: While (Num. of Iterations is less than X) or (Pareto archive still changes) DO
        4: Compute(SUP)          //the support value of chromosomes is computed
        5: Confidence = SUP(A ∪ C)/SUP(A)
        6: Comprehensibility = log(1+ | C |)/log(1+ | A ∪ C |)
        7: Interestingness = [SUP(A ∪ C)/SUP(A)] × [SUP(A ∪ C)/SUP(C)] ×[1 − SUP(A ∪ C)/SUP(D)]
        8: Compute(fitness)    //Based on chromosomes' ranks and using the non-dominance property
        9: AddtoParArch(non-dominated solutions)
        10: RemovefromParArch(dominated sloutions)
        11: New population = Crossover(current population)
        12: Mutate(new population)
13: EndWhile
```

Fig. 4. Pseudo-code for proposed method.

4. Experimental results and discussion

We evaluate our proposed method in three public domain datasets: Basketball, Bodyfat and Quake. These datasets are available from the Bilkent University Function Approximation Repository [16]. Table 1 shows the parameter settings of our experiments. Table 2 shows the properties of the datasets that have been used for our experimental studies. Because of stochastic characteristics of evolutionary methods including MOGA, several trials of these algorithms should be run to get more efficient and reliable results. Accordingly, we examined our method 10 times over each dataset and then the average values of such executions are presented. The termination criterion was 400 rounds of running the algorithm. The population size is set to 800, the crossover probability to 0.8 and the mutation probability to 0.01. The mutation operator was illustrated earlier, and 2-point crossover is used as the crossover operator. Based on trying several experiments for choosing the crossover type, we have chosen 2-point crossover which works better than other types of k-point crossovers.

For more illustration, here, we have presented one of the interesting rules that are obtained through our method on the bodyfat dataset:

$(71.2 < a12 < 115.4) \rightarrow (0.7 < a3 < 1.0)(54.7 < a8 < 220.5)(26.4 < a11 < 118.8)(25.9 < a16 < 39.0)$

Comprehensibility: 0.8983, Confidence: 0.988, Interestingness: 0.0273, Support: 215.

The comprehensibility, confidence and support count of this rule are high, although its interestingness measure value is fairly low. In the tables and results which are shown in this section, confidence, comprehensibility and interestingness refer to three measures which were introduced in the Section 2. 'Number of rules' shows the number of total association rules that the method found and finally generated. The value of the "Size" column shows the mean number of attributes contained in the rules.

In Tables 3 and 4, the results obtained from our method (MOGAR) are compared with results from Alatas and Akin [4] and Alatas and Akin [5]. In the first method, a genetic algorithm (GA) is proposed as a search strategy for not only positive, but also negative quantitative association rule (AR) mining within datasets. The second method is based on the rough particle swarm optimization (RPSO) and uses this method in order to mine numeric rules [5]. In Table 4, we add the results from the genetic association rule mining (GAR) algorithm [27] to the other results. This method uses an evolutionary algorithm to find the intervals of each attribute that conform a frequent itemset. The evaluation function itself will be the one that decides the amplitude of these intervals. Finally, they evaluate the tool with synthetic and real databases to check the efficiency of this algorithm [27]. Three datasets which are used here to evaluate the proposed method are used in the above methods too, and the results are directly reported from the original works. So, the parameter settings for these algorithms are the same as the ones described in the corresponding papers.

Our observation of the generated rules shows that we have obtained much better results in the case of number of rules in the Bodyfat and Basketball datasets. Average improvement in the first dataset is about 50% and in the second dataset is almost 32%. But, our number of rules in the Quake dataset is a little less than the number of rules in [5]. Our results of the confidence measure are much better than previous works and in average they are 20% higher than the others. In the support and size measures, we had almost the same results. The number of generated rules and their confidence values are more efficient than those of the previous works, and it can be concluded that our generated rules are useful for the users.

Table 1
The parameter settings.

Parameter	Population size	Crossover probability	Mutation probability	Termination criteria
Value	800	0.8	0.01	Algorithm iterates 400 rounds

Table 2
Properties of test datasets.

Dataset	No. of records	No. of features	Target feature
Basketball	96	5	Points per minute
Bodyfat	225	18	Body height
Quake	2178	4	Richter

Table 3
Comparison of results.

Dataset	No. of rules			Confidence		
	Alatas and Akin [4]	RPSO	MOGAR	Alatas and Akin [4]	RPSO	MOGAR
Basketball	33.8	34.2	50	0.60	0.60	0.83
Bodyfat	44.2	46.4	84	0.59	0.61	0.85
Quake	43.8	46.4	44.87	0.62	0.63	0.82

Table 4
Comparison of results (cont.).

Dataset	Support (%)				Size			
	Alatas and Akin [4]	RPSO	GAR	MOGAR	Alatas and Akin [4]	RPSO	GAR	MOGAR
Basketball	32.21	36.44	36.69	50.82	3.21	3.21	3.38	3.24
Bodyfat	63.29	65.22	65.26	57.22	6.94	7.06	7.45	6.96
Quake	38.74	38.74	36.96	30.12	2.10	2.22	2.33	2.38

Table 5
Percentages of records covered by the mined rules.

Dataset	Records %			
	Alatas and Akin [4]	RPSO	GAR	MOGAR
Basketball	100.00	100.00	100.00	100.00
Bodyfat	84.12	86.11	86.00	93.52
Quake	87.6	87.92	87.5	91.07

Table 6
Other results of suggested approach.

Dataset	Comprehensibility	Interestingness
Basketball	0.72	0.53
Bodyfat	0.80	0.56
Quake	0.68	0.46

Table 7
Standard deviations of the number of generated rules from different algorithm runs.

	No. of rules	Standard deviation
Basketball	50	3.76
Bodyfat	84	7.62
Quake	44.87	6.43

In Table 5, we compare 'Percentages' of records covered by the mined rules with other works. These values show what portion size of the records, these rules apply to. The results are competitive with the other three works and in the case of Bodyfat and Quake datasets, we obtain significant improvements.

Moreover, in our approach as was described, we use comprehensibility and interestingness measures to mine association rules. Values of these measures are shown in Table 6. These results are also obtained with the settings mentioned in Table 1. Using genetic algorithms can bring randomness and diversity of solutions for obtained results. To demonstrate the extent of this effect on the suggested method, standard deviations of the number of generated rules from different algorithm runs are shown in Table 7). Since standard deviation values are low, it can be concluded that the number of generated rules after a different algorithm runs is not too diverse.

It is worth mentioning that, in the first glimpse, it may seem that using Pareto optimization method for association rule mining is problematic. The first problem is initiated from the concept of 'dominance' property which may not be enough to guarantee that the information encoded in one rule is subsumed by the dominant rule. This problem occurs when a potentially proper rule becomes dominated by another independent and dissimilar rule. But, our experiments show that this is rare and infrequent. On the Bodyfat dataset, we changed our algorithm such that if a chromosome is dominated more than 10 times (1/40 total number of iterations), then it is kept in the Pareto archive as one of the final results. If a good rule becomes dominated and does not enter into the Pareto archive, there is a good chance that it will be reproduced during the next iterations. This probability is high because the rule has a good rank and consequently good chance for reproduction.[3] Obtained rules by this modification were compared with rules generated by our Pareto-based approach and there were only a few – between 6 and 10 – rules which were not generated by our approach – versus 84 average generated rules by the approach. This is outcome of 10 experiments on the Bodyfat dataset. The experiment was done on other two datasets and results were almost the same. Another point here is that end users usually are not looking for all possible rules. They are willing to find the rules containing useful information. So, if we have a method that can find such rules without requiring to specify minimum

[3] This is one of explanations for the observed results.

values – data dependency –, they will profit much more. After all, multi-objective association rule mining approaches have several significant advantages over Apriori-based methods – as mentioned in the introduction section, although they generally suffer from the risk of excluding some proper rules. In Table 3, we have shown that the proposed method generates more rules in comparison with other multi-objective methods.

Another problem about using Pareto-optimal rules is that it is possible for some weak rules – rules with low support, or other described measures – to stay non-dominated and be generated as one of the outputs. To resolve this problem, we use a minimum support value that eliminates non-dominated rules with high values in confidence, interestingness and comprehensibility, but with low support. Furthermore, to avoid generation of rules having very small intervals, a 'minimum range length' is used. This minimum length is proportional to the original range of existing attributes.

5. Conclusions

In this article, we proposed a new approach for numerical association rule mining using multi-objective genetic algorithms which is based on the notion of rough patterns. The method uses rough values which are defined with upper and lower intervals to represent a range or set of values.

Association rule mining is not a single objective problem and naturally is a multi-objective one. We used three measures to mine more efficient association rules: Confidence, Comprehensibility and Interestingness. A new method based on the Michigan approach is used for representing rules in chromosomes. Moreover, the idea of Pareto optimality was employed to solve multi-objective optimization problems and to improve the performance of algorithm.

Multi-objective genetic algorithm association rule mining (the proposed method) is used in data mining within databases that include numeric attributes. Our experimental results on datasets with numeric values show that our proposed technique is useful and helpful for discovering numerical association rules.

Acknowledgments

The authors would like to thank Rahmatollah Beheshti, Amy K. Hoover and Meisam Fathi Salmi for their thoughtful comments and assisting with the manuscript preparation

References

[1] A. Abraham, L. Jain, Evolutionary multiobjective optimization, in: Evolutionary Multiobjective Optimization, Advanced Information and Knowledge Processing, Springer, Berlin Heidelberg, 2005, pp. 1–6.
[2] R. Agrawal, T. Imieliński, A. Swami, Mining association rules between sets of items in large databases, in: ACM SIGMOD Record, vol. 22, ACM, pp. 207–216.
[3] R. Agrawal, R. Srikant, et al., Fast algorithms for mining association rules, in: Proc. 20th Int. Conf. Very Large Data Bases, VLDB, vol. 1215, Citeseer, pp. 487–499.
[4] B. Alataş, E. Akin, An efficient genetic algorithm for automated mining of both positive and negative quantitative association rules, Soft Computing-A Fusion of Foundations, Methodologies and Applications 10 (2006) 230–237.
[5] B. Alatas, E. Akin, Rough particle swarm optimization and its applications in data mining, Soft Computing-A Fusion of Foundations, Methodologies and Applications 12 (2008) 1205–1218.
[6] B. Alatas, E. Akin, Chaotically encoded particle swarm optimization algorithm and its applications, Chaos, Solitons & Fractals 41 (2009) 939–950.
[7] B. Alatas, E. Akin, A. Karci, Modenar: multi-objective differential evolution algorithm for mining numeric association rules, Applied Soft Computing 8 (2008) 646–656.
[8] Y. Aumann, Y. Lindell, A statistical theory for quantitative association rules, Journal of Intelligent Information Systems 20 (2003) 255–283.
[9] C. Chen, T. Hong, V. Tseng, L. Chen, A multi-objective genetic-fuzzy mining algorithm, in: IEEE International Conference on Granular Computing, 2008, GrC 2008, IEEE, pp. 115–120.
[10] C. Chen, V. Tseng, T. Hong, Cluster-based evaluation in fuzzy-genetic data mining, IEEE Transactions on Fuzzy Systems 16 (2008) 249–262.
[11] B. Chien, Z. Lin, T. Hong, An efficient clustering algorithm for mining fuzzy quantitative association rules, in: IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th, vol. 3, IEEE, pp. 1306–1311.
[12] K. Cios, W. Pedrycz, R. Świniarski, R. Swiniarski, Data Mining Methods for Knowledge Discovery, Kluwer Academic Publishers., 1998.
[13] A.A. Freitas, A Survey of Evolutionary Algorithms for Data Mining and Knowledge Discovery, Springer Verlag New York, Inc., New York, NY, USA, 2003. pp. 819–845.
[14] T. Fukuda, Y. Morimoto, S. Morishita, T. Tokuyama, Mining optimized association rules for numeric attributes, in: Proceedings of the Fifteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of database systems, PODS '96, ACM, New York, NY, USA, 1996, pp. 182–191.
[15] A. Ghosh, B. Nath, Multi-objective rule mining using genetic algorithms, Information Sciences 163 (2004) 123–133.
[16] H. Guvenir, I. Uysal, Bilkent University Function Approximation Repository, 2000. <http://funapp.cs.bilkent.edu.tr/>.
[17] M. Houtsma, A. Swami, Set-oriented mining for association rules in relational databases, International Conference on Data Engineering, 1995, pp. 25.
[18] M. Kaya, R. Alhajj, Genetic algorithm based framework for mining fuzzy association rules, Fuzzy Sets and Systems 152 (2005) 587–601.
[19] Y. Ke, J. Cheng, W. Ng, An information-theoretic approach to quantitative association rule mining, Knowledge and Information Systems 16 (2008) 213–244.
[20] D. Kumar, A genetic algorithm with entropy based probabilistic initialization and memory for automated rule mining, Advances in Computer Science and Information Technology (2011) 604–613.
[21] T. Li, X. Li, Novel alarm correlation analysis system based on association rules mining in telecommunication networks, Information Sciences 180 (2010) 2960–2978.
[22] D. Lin, Z. Kedem, Pincer-search: an efficient algorithm for discovering the maximum frequent set, IEEE Transactions on Knowledge and Data Engineering (2002) 553–566.
[23] B. Liu, W. Hsu, S. Chen, Y. Ma, Analyzing the subjective interestingness of association rules, IEEE Intelligent Systems (2000) 47–55.
[24] F.G. Lobo, D.E. Goldberg, M. Pelikan, Time complexity of genetic algorithms on exponentially scaled problems, in: Proceedings of the Genetic and Evolutionary Computation Conference, Morgan-Kaufmann, 2000, pp. 151–158.
[25] C. Lucas, Practical Multiobjective Optimisation, 2006. <http://www.calresco.org/lucas/pmo.htm>.

[26] G. Mansingh, K.M. Osei-Bryson, H. Reichgelt, Using ontologies to facilitate post-processing of association rules by domain experts, Information Sciences 181 (2011) 419–434.
[27] J. Mata, J. Alvarez, J. Riquelme, Discovering numeric association rules via evolutionary algorithm, Advances in Knowledge Discovery and Data Mining (2002) 40–51.
[28] M. Nasiri, L. Taghavi, B. Minaee, Multi-objective rule mining using simulated annealing algorithm, Journal of Convergence Information Technology 5 (2010) 60–68.
[29] D. Olson, Y. Li, Mining fuzzy weighted association rules, in: 40th Annual Hawaii International Conference on System Sciences, 2007. HICSS 2007, IEEE, pp. 53–53.
[30] V. Pachón, J. Mata, J. Domínguez, M. Maña, Multi-objective evolutionary approach for subgroup discovery, Hybrid Artificial Intelligent Systems 6679 (2011) 271–278.
[31] A. Salleb-Aouissi, C. Vrain, C. Nortet, Quantminer: a genetic algorithm for mining quantitative association rules, in: Proceedings of the 2007 International Join Conference on Artificial Intelligence, pp. 1035–1040.
[32] F.S. Tseng, Y.H. Kuo, Y.M. Huang, Toward boosting distributed association rule mining by data de-clustering, Information Sciences 180 (2010) 4263–4289.
[33] P. Wakabi-Waiswa, V. Baryamureeba, Extraction of interesting association rules using genetic algorithms, International Journal of Computing and ICT Research 2 (2008) 1139–1818.
[34] X. Yan, C. Zhang, S. Zhang, Genetic algorithm-based strategy for identifying association rules without specifying actual minimum support, Expert Systems with Applications 36 (2009) 3066–3076.
[35] H.R. Qodmanan, M. Nasiri, B. Minaei-Bidgoli, Multi objective association rule mining with genetic algorithm without specifying minimum support and minimum confidence, Expert Systems with applications 38 (1) (2011) 288–298.
[36] M. Asadollahpoor-Chamazi, B. Minaei-Bidgoli, M. Nasiri, Deriving support threshold values and membership functions using the multiple-level cluster-based master-slave IFG approach, Soft Computing (2013), http://dx.doi.org/doi/10.1007/s00500-012-0973-7.