On the discovery of association rules by means of evolutionary algorithms

María J. del Jesus,[1] José A. Gámez,[2] Pedro González[1] and José M. Puerta[2]

Association rule learning is a data mining task that tries to discover interesting relations between variables in large databases. A review of association rule learning is presented that focuses on the use of evolutionary algorithms not only applied to Boolean variables but also to categorical and quantitative ones. The use of fuzzy rules in the evolutionary algorithms for association rule learning is also described. Finally, the main applications of association rule evolutionary learning covered by the specialized bibliography are reviewed. © 2011 John Wiley & Sons, Inc. *WIREs Data Mining Knowl Discov* 2011 00 1–19 DOI: 10.1002/widm.18

INTRODUCTION

Association rules (ARs) are a widely used formalism in data mining.[1,2] The idea is to use *if-then* rules to discover interesting relations between variables in large databases. In their origin, ARs were strongly associated with *market basket analysis*, because they were learnt from transactional data (e.g., point-of-sale data), and the information codified by the rules, e.g., if buy(bread) and buy(milk) then buy(butter) (in short, bread $\wedge$ milk $\Rightarrow$ butter), can later be used by the marketing department as the basis for decisions involving promotions, product placement, etc.[3] Nowadays, their use has been extended to many different fields, including electronic commerce,[4] web usage mining,[5] intrusion detection,[6] bioinformatics,[7] etc.

Given a set of items, objects, or binary variables $\mathbf{I} = \{I_1, I_2, \ldots, I_n\}$, an AR is formally[a] defined by Agrawal et al.[1] as an implication $\mathbf{X} \Rightarrow \mathbf{Y}$ where $\mathbf{X}, \mathbf{Y} \subseteq I$ and $\mathbf{X} \cap \mathbf{Y} = \emptyset$. Both the antecedent ($\mathbf{X}$) and the consequent ($\mathbf{Y}$) are interpreted as a conjunction of the variables they contain, e.g., $\mathbf{X} = I_1 \wedge I_2 \wedge I_k$. *AR learning* is usually stated as the problem of learning such types of rules from a dataset $\mathbf{D} = \{t_1, t_2, \ldots, t_d\}$ of transactions [Table 1, part (a)].

The problem is that, from the previous definition, the number of possible ARs, given a number of products or items d, is too large, $\#rules(d) = 3^d - 2^{d+1} + 1$, to be precise, for example:

$\#items$	$\#rules$
5	180
10	57002
100	5.1537752×10^{47}

It is, therefore, compulsory to filter them in some way before trying to analyze their usefulness. Thus, Piatetsky-Shapiro[2] introduced the concept of *strong* rules based on the use of measures of interestingness and Agrawal et al.[1] introduced the AIS algorithm to discover *significant* ARs from data. To do this, the following two measures are used:

- *Support*. The support of an itemset $\mathbf{X}$, $supp(\mathbf{X})$, is defined as the number of transactions (instances) containing it, that is, the prior probability $P(\mathbf{X})$ of $\mathbf{X}$ estimated from $\mathbf{D}$. The support of a rule $\mathbf{X} \Rightarrow \mathbf{Y}$ is computed as $supp(\mathbf{X} \Rightarrow \mathbf{Y}) = supp(\mathbf{X} \wedge \mathbf{Y})$.
- *Confidence*. The confidence of a rule $\mathbf{X} \Rightarrow \mathbf{Y}$ is computed as $\frac{supp(\mathbf{X} \wedge \mathbf{Y})}{supp(\mathbf{X})}$, which can be interpreted as the conditional probability $P(\mathbf{Y}|\mathbf{X})$.

Then, significant rules are those surpassing a minimum threshold for both support and confidence (min_s and min_c, respectively). Other measures such as lift, conviction, or coverage have also been used to identify significant ARs (e.g., see Brin et al.[8]).

*Correspondence to:

[1]Computer Science Department. University of Jaén. Campus Las Lagunillas s/n. Jaén, 23071, Spain.

[2]Computing Systems Deparment. University of Castilla-La Mancha. Campus Universitario s/n. Albacete, 02071. Spain.

DOI: 10.1002/widm.18

TABLE 1 | Different Representations of the Data Set: (a) transactional, (b) transactional (inverted by item), and (c) tabular

Tid	items	A	B	C	D	E	Tid	A	B	C	D	E	...	F	G
1	A, D	1	2	3	1	2	1	1	0	0	1	0		f_1	0.72
2	A, B, E	2	3	5	3	5	2	1	1	0	0	1		f_2	−2.7
3	B, C, D	4	4				3	0	1	1	1	0		f_1	3.54
4	A, B						4	1	1	0	0	0		f_3	4.92
5	C, E						5	0	0	1	0	1		f_2	1.12
	(a)			(b)							(c)				

Ais only learns ARs in the form $\mathbf{X} \Rightarrow I_k$, but later, Agrawal and Srikant[9] introduced the Apriori algorithm, which searches for general ARs ($\mathbf{X} \Rightarrow \mathbf{Y}$). The main and most computationally expensive step in AR rule learning is the generation of *frequent* itemsets, that is, itemsets having support greater than min_s. The Apriori algorithm is the most cited (and probably the most used) in the field of AR learning, as it introduced the *apriori principle*, also known as the *anti-monotone property of support*: 'if a pattern of length k is not frequent in the dataset, then none of its super-patterns of length $k+1$ can be frequent'. The use of this principle allows the Apriori algorithm to prune the space of itemsets in such a way that candidate frequent patterns of length $k+1$ are obtained 'only' by using previously found frequent patterns of length k.

Even with the reduction in the space of candidate itemsets achieved by Apriori, this step is still the bottleneck in the process of discovering ARs. Notice that Apriori needs to scan the dataset k times, k being the length of the largest frequent itemset found. Thus, some other approaches have been proposed, such as Eclat or Fp-Growth. Eclat,[10] proposed by Zaki, uses a vertical data (inverted) layout [Table 1, part (b)] and a depth-first strategy (in contrast with Apriori that uses a breadth-first one). Eclat is very efficient for large itemsets but less efficient for small ones. Following a different idea, Han et al.[11] presented Fp-Growth, which tries to overcome the k scans needed by Apriori by mining the frequent patterns without candidate generation. Fp-Growth uses only two passes over the dataset in order to build a frequent-pattern tree, which is later used to discover the frequent itemsets. Many other improvements (memory, parallelism, etc.) have been made for the problem of AR learning, but they are beyond the scope of this paper (see, e.g., Zhang and He[12] or Tan et al.[13]).

In this study, we review recent literature about AR learning that emphasizes the use of non-classical algorithms. By setting the problem as of a combinatorial optimization we focus on the use of evolutionary algorithms and also on the case of non-Boolean variables, that is, categorical and quantitative ones. Thus, in most cases, we consider the dataset in a tabular form instead of a transactional one [see Table 1, part (c)].

The rest of the paper is organized as follows: first, we give a brief introduction to the problem of learning rules-based systems by using evolutionary algorithms (Section *Learning Rule-Based Systems by Evolutionary Algorithms*). Then, Sections *Learning Boolean/Categorical Association Rules* and *Learning Quantitative/Numerical Association Rules* are devoted to reviewing the genetic algorithm (GA)-based approaches available in the literature for learning Boolean/categorical ARs and quantitative ARs, respectively. Section *Learning Fuzzy Association Rules* describes a different way of dealing with numerical attributes that is based on fuzzy set theory. In Section *Learning by Following a Multiobjective Approach*, we describe the importance of using a multi-objective approach when learning ARs. Finally, the last three sections cover some approaches based on metaheuristics that are different from GAs (Section *Swarm-Based Approaches for Learning ARs*), real applications (Section *Applications*) and our conclusions (Section *Conclusions*).

LEARNING RULE-BASED SYSTEMS BY EVOLUTIONARY ALGORITHMS

Metaheuristics[14] can be seen as general algorithms that can be applied to solve different combinatorial (or numerical) optimization problems by carrying out few modifications in order to adapt them to a specific problem. Depending on the way the search is carried out, we can distinguish between *local* and *global* metaheuristics. Evolutionary algorithms (EAs) are perhaps the most-used family in the case of global optimization.[15] EAs are population-based algorithms, in many cases bio-inspired, that solve the problem by simulating an evolutive process that tries to improve a population (of solutions) by evolving them through generations. The general scheme in EAs is as follows:

(1) Initialize population (random generation of potential solutions)
(2) Evaluate individuals in population
(3) Repeat until a stopping criterion is met
    (a) Select individuals from current population
    (b) Recombine them in order to obtain new individuals (offsprings)
    (c) Evaluate new individuals
    (d) Replace some or all the individuals of the current population by offsprings
(4) Return the best individual seen so far

Depending on the way Step 3(b) is designed, we get different EAs. Without any doubt, GAs[16] are the most famous EAs. In GAs, Step 3(b) is designed to use genetic operators such as crossover and mutation, which simulate the biological process of DNA recombination. A different philosophy is followed in the case of estimation of distribution algorithms (EDAs),[17] a recent family of EAs in which Step 3(b) is designed to learn (and sample) a probability distribution that codifies the selected individuals (Step 3(a)).

Although metaheuristics in general, and EAs in particular, were not originally designed for learning, their success when solving other (NP-hard) combinatorial/numerical optimization problems has led to their extensive use in solving data mining problems in the last few years. In fact, the use of these types of algorithms in data-mining-based problems is a hot research topic nowadays.[18, 19]

Learning rule-based system, mainly for supervised problems such as classification and regression, has perhaps been the data mining task for which EAs have most often been applied. This is also the topic of interest in this study, and so, we provide a concise introduction here. Detailed studies can be found in Fernández et al.[20] and Freitas.[19]

Approaching the problem of learning rule-based systems by using EAs (or metaheuristics in general) requires the specification or design of different components. Here, we pay special attention to individual representation, fitness evaluation, and operators.

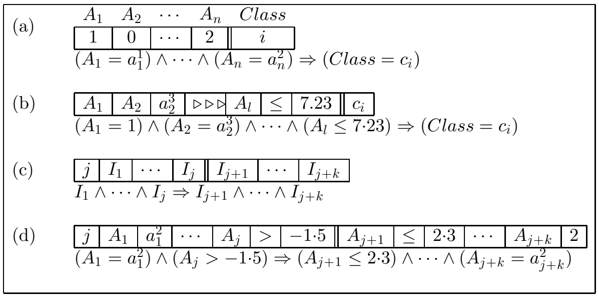
(a)
A_1 A_2 $\cdots$ A_n $Class$
1 | 0 | $\cdots$ | 2 | i
$(A_1 = a_1^1) \wedge \cdots \wedge (A_n = a_n^2) \Rightarrow (Class = c_i)$
(b)
A_1 | A_2 | a_2^3 | $\triangleright\triangleright\triangleright$ | A_l | $\leq$ | 7.23 | c_i
$(A_1 = 1) \wedge (A_2 = a_2^3) \wedge \cdots \wedge (A_l \leq 7{\cdot}23) \Rightarrow (Class = c_i)$
(c)
j | I_1 | $\cdots$ | I_j | I_{j+1} | $\cdots$ | I_{j+k}
$I_1 \wedge \cdots \wedge I_j \Rightarrow I_{j+1} \wedge \cdots \wedge I_{j+k}$
(d)
j | A_1 | a_1^2 | $\cdots$ | A_j | $>$ | $-1{\cdot}5$ | A_{j+1} | $\leq$ | $2{\cdot}3$ | $\cdots$ | A_{j+k} | 2
$(A_1 = a_1^2) \wedge (A_j > -1{\cdot}5) \Rightarrow (A_{j+1} \leq 2{\cdot}3) \wedge \cdots \wedge (A_{j+k} = a_{j+k}^2)$

FIGURE 1 | Some examples of individual representation (chromosome = rule) and their corresponding decodified rules.

Individual Representation

As stated above, an individual is a *potential* solution to the problem we are solving. From this point of view, in our problem, a solution is a set of unordered ARs, and consequently, an individual can represent a set of rules. This approach is known in GA learning systems as the *chromosome = Base of Rules (BoR)* or *Pittsburgh* approach[21] and represents a direct extension of GAs to supervised learning problems. There is, however, a different way to deal with this problem that considers that all the population is in fact the solution, and so, each individual or chromosome represents a single rule. This approach to the problem, known as (*chromosome = rule*) or learning classifier systems, has given rise to different approaches: *Michigan*,[22] *iterative rule-learning*,[23] and *genetic cooperative-competitive learning*.[24]

Following the typical constraints of defining the antecedent (and the consequent) as a conjunction of single clauses, represented here by attribute-value pairs, Figure 1(a) shows a standard individual representation for a classification rule. In this case, each individual is represented by a vector of length $n + 1$ where the first n positions contain the value taken by the $i - th$ attribute in the antecedent, and the last one contains the value for the class variable (consequent). In the chromosome = BoR approach, each individual would be a concatenation of this type of vector. However, if n is large, the previous representation is not useful; in this case, the representation in Figure 1(b) can be used. Now, the vector can have variable length, and if we read the chromosome from left to right, we can find a number codifying the referred variable followed by:

- Nothing more if such a variable is binary because its mere presence denotes the *true* state (or *1*).
- A number representing the state taken by the variable in the case that the referred variable is discrete/nominal.

- A number codifying the condition ($\leq$, $<$, ...) and another real number with the value to be used as threshold.

Finally, the last variable will contain the number codifying the class label for this rule. In the figure, for reasons of clarity, instead of showing only numbers, we also show the attribute, condition, and state.

Parts (c) and (d) in Figure 1 show the adaptation of this representation for supervised learning to unsupervised learning, as is the case of ARs. Thus, part (c) shows the case of Boolean ARs, where the the first value in the vector represents the cut-point between the antecedent and the consequent. Part (d) shows a general case with different types of variables (nominal, quantitative, etc.).

Fitness Evaluation

In the case of supervised learning, the fitness is measured by computing some precision value over the training (or a different validation) dataset, for example, accuracy for classification and mean square error for regression. When moving to ARs, instead of these measures, we can use, for example, the averaged support and confidence of the discovered rules with respect to the training dataset. However, other measures should also be taken into account when evaluating our solutions, such as:

- the compactness of the discovered rule set: number of rules and their complexity (length), and
- how general our rule set is, that is, the percentage of instances in our training set covered by the discovered rules.

Thus, the fitness function could have the following appearance:

$$\begin{aligned} fitness(sol) = w_1 \times accuracy(sol) + w_2 \\ \times simplicity(sol) + w_3 \times coverage(sol), \end{aligned}$$

with w_1, w_2, and w_3 being real numbers that represent the weight given to each component, and where sol is the rule set (solution) being evaluated.

However, different evaluation mechanisms are possible, as we will detail in Section *Learning by Following a Multiobjective Approach*.

Operators and the Genetic Model

Once the representation and fitness function are defined, conventional/typical evolutionary schemes and operators can be used. However, because of the complexity of the problem under study, specific genetic operators are usually considered. In the next sections, we will comment on some of these operators and schemes when reviewing main/recent approaches for association rule learning with EAs. For details, the interested reader is referred to the literature cited.

LEARNING BOOLEAN/CATEGORICAL ASSOCIATION RULES

This section and the next one are devoted to reviewing the main contributions of evolutionary algorithms to the problem of mining ARs from data. Here, we focus on *classical* ARs, that is, considering only Boolean attributes, and also their extension to categorical or nominal attributes. We leave the case of quantitative or numerical attributes for the next section.

In evolutionary AR learning algorithms, we can distinguish two clearly differentiated approaches: (1) the evolutionary algorithm is used to evolve itemsets and then rules are extracted at a post-processing stage, or (2) the evolutionary algorithm directly tries to evolve ARs, and therefore, they usually identify highly qualified ARs. In the second case, in all the literature reviewed in this paper relating to this section, a chromosome = rule-like approach is followed, that is, each individual in the population codifies a single rule. GAs are used to guide the search process.

The main advantages of using GAs are as follows:

- In general, the use of a minimum support (and/or confidence) threshold for itemsets and rules can be avoided if a multi-objective approach is used, because in this case, a more complex way of assessing the fitness of an individual can be considered, as can be seen in Section *Learning by Following a Multiobjective Approach*.
- A global optimization/search is carried out that allows the exploration of the cooperation between the rules included in a population.
- GAs are very flexible algorithms that can deal with many different AR learning problems, for example, dynamic databases, data streams, distributed learning approaches, multi-objective approaches, negative ARs, etc.

However, the use of GAs also has some disadvantages that are listed below:

- Mechanisms for diversity preservation should be incorporated because otherwise the algorithm will converge to a few high quality ARs.

- A combination of different quality measures should be used to evaluate the fitness, because if only support and/or confidence are considered, then the algorithms will likely fall into local optima.

Once we have described the general setting, let us review the different approaches found in the literature. We focus the study on the structure of the GA used, mainly representation and genetic operators. We also pay attention both to the GA model used and to the way the initial population is created when the standard approach is not followed.

Representation

Approaches based on evolving itemsets such as DMARG[27] and DDMARG[28] use binary representation, in which each allele in the chromosome accounts for a given item. It is apparent how to extend this representation to more than two values per attribute, just representing a concrete value instead of only 0 or 1 in each allele.

If we move on to the case of evolving rules, two different representations can be found in the literature. First, in the ARMGA algorithm,[29] a length k is set for the rules, then, a chromosome of length $k + 1$ is used. Positions 1 to k take as value the index of items, while position/gene 0 contains the *cut point* between the antecedent and the consequent of the rule. See Figure 1(b) for an example. Notice that this *Boolean* representation can be extended to the categorical (nonbinary) case by using the scheme commented in Section *Learning Rule-Based Systems by Evolutionary Algorithms* and illustrated in Figure 1(d).

A different choice is considered by Dehuri et al.[25] and Wakabi-Waiswa and Baryamureeba,[26] in which all the items/attributes are considered in the chromosome. Thus, if we have n attributes, the chromosome will have $2n$ alleles, the first two for the first attribute, third and fourth for the second attribute, and so on. Then, each pair of alleles must be decoded as: (00), the referred attribute is included in the rule in the antecedent; (11) the referred attribute is included in the consequent of the rule; (10) and (01) if the referred attribute is not included in the rule. See Figure 2 for an example. One of the advantages of this representation is that we can have rules of different lengths, while the main disadvantage is the length of the chromosome. In order to deal with categorical (non-binary) attributes, the same idea can be used, but using, for example, the first allele to indicate the value taken by the attribute (0 means not included) and the second allele to indicate antecedent (0) or consequent (1).

(01)	(11)	(00)	(00)	(00)	(11)	(01)
I_1	I_2	I_3	I_4	I_5	I_6	I_7

$I_3 \wedge I_4 \wedge I_5 \Rightarrow I_2 \wedge I_6$

FIGURE 2 | Individual representation (chromosome = rule) used in.[25, 26]

Genetic Operators

Because of the use of standard binary and n-ary codifications, standard crossover and the mutation operators are used, for example, two-points crossover and swapping for mutation.

Initial Population

In general, it is completely generated at random, but in some cases, as in the papers by Shenoy et al.,[27, 28] a special initial population is used. In these algorithms, all the itemsets of size 1 passing some condition (e.g., min_s) are included in the initial population.

Fitness

A traditional approach based on measures of confidence and support is normally used. However, combinations of other measures such as comprehensibility, diversity, J-measure, perplexity, and coverage are also commonly used. For example, in Dehuri et al.,[25] confidence plus comprehensibility plus interestingness is defined as the function to be optimized (maximized in this case). In Wakabi-Waiswa and Baryamureeba,[26] the three previous measures are also considered together with J-measure and perplexity, but instead of a simple addition, a linear combination is used. Finally, in ARMGA,[29] the function $f(\mathbf{A} \Rightarrow \mathbf{B}) = supp(\mathbf{AB}) - supp(\mathbf{A}) \times supp(\mathbf{B})/supp(\mathbf{A}) \times (1 - supp(\mathbf{B})$ is defined as the fitness function for individuals.

Models

As most approaches use a compound fitness function, for example, a linear combination or aggregation of measures, traditional GAs can be used to perform the search. However, the multiobjective approach (e.g., MOGA,[30] see, Section *Learning by Following a Multiobjective Approach* together with a specific scheme to distribute the algorithm) has also been used to evolve the ARs Pareto-front in a distributed cluster of workstations. More information about AR learning following a multi-objective approach can be found in Section *Learning by Following a Multiobjective Approach*. Another important aspect is that, to a greater or lesser extent, all these approaches use some mechanism to preserve diversity, a common (and inherent) way of working in multi-objective models. Finally,

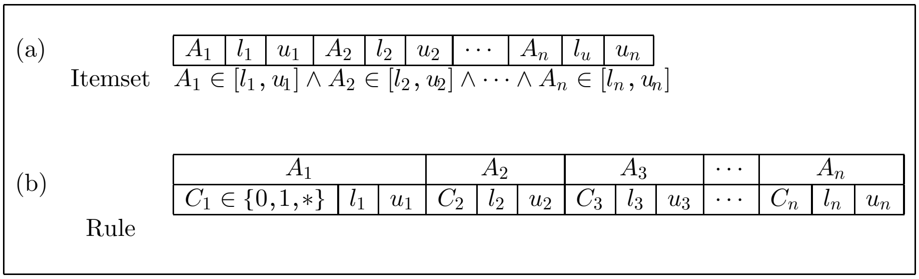
(a)
Itemset
A_1 | l_1 | u_1 | A_2 | l_2 | u_2 | $\cdots$ | A_n | l_u | u_n
$A_1 \in [l_1, u_1] \wedge A_2 \in [l_2, u_2] \wedge \cdots \wedge A_n \in [l_n, u_n]$
(b)
Rule
A_1 | A_2 | A_3 | $\cdots$ | A_n
$C_1 \in \{0,1,*\}$ | l_1 | u_1 | C_2 | l_2 | u_2 | C_3 | l_3 | u_3 | $\cdots$ | C_n | l_n | u_n

FIGURE 3 | Individual representation (chromosome = rule or itemset).

elitism and tournament are widely used for replacement and selection.

LEARNING QUANTITATIVE/ NUMERICAL ASSOCIATION RULES

Early attempts to deal with numerical variables in ARs were based on discretization.[31] Thus, the domain of the numerical variable is partitioned into k intervals (e.g., using equal width or equal frequency), and as a consequence, a categorical variable with k values is obtained. Of course, the main advantage of this approach is that once we obtain nominal variables, then algorithms described in the previous section can be applied. Nevertheless, discretization for numerical variables in ARs has two important drawbacks:

- The size of the intervals: Larger intervals will receive greater support, so if support is our guide, we should consider a single interval because it will receive all the support, but of course the resulting item (and rules) is meaningless. Some solutions to this problem are to limit the intervals' width and to use some penalty in order to negatively rate wider intervals.
- *Sharp boundary*. Item values close to intervals' boundaries are overemphasized.

An alternative approach to handle numerical variables is to model the variable by using a statistical distribution, as Aumann and Lindell did in,[32] and then to use its parameters (mean, variance, etc.) in the rule definition. However, this approach does not solve the discretization problem because, in general, intervals are used in the antecedent and parameters in the consequent. Another example of this approach is the work by Webb,[33] in which interestingness and impact measures are used in the rule consequent.

In this section, we focus on evolutionary approaches to deal with numerical variables in ARs, known as *quantitative ARs*. We highlight that one of the advantages of using GAs is their flexible representation that allows the mixing of different kinds of attributes: numerical and nominal. As we will see, perhaps the more important advantage is that now the variable is not preprocessed, but, on the contrary, the interval for each numerical variable used in a rule is defined by taking that rule into account.

This section shares many aspects with the previous one, such as the fact of looking for frequent itemsets or directly from ARs. We also try to maintain a similar structure, describing the main components of an evolutionary algorithm for the different proposals found in the literature.

Representation

There are several alternatives for representing continuous attributes inside a population. One of the first attempts is the algorithm GAR[34] (see Figure 3(a)), which uses a representation of an itemset. To do this, each attribute is codified through a set of three consecutive genes, the first one being the index of the variable, and then an interval is represented by storing the lower and upper bound of the interval. Furthermore, a variable-length representation for each itemset is used in this case. Kwasnicka and Switalski have proposed an extension named EGAR[35] that uses continuous and nominal variables simultaneously.

Another of the most frequently mentioned algorithms in the literature is QUANTMINER.[36] In this algorithm, the individuals represent rules directly. Quantitative variables are also defined by the limits (upper and lower bounds) of an interval for each variable, and the algorithm basically evolves to search for the best set of intervals for the numerical attributes.

The same authors of GAR presented the GENAR algorithm,[37] which directly represents ARs rather than itemsets. Each individual represents an association rule, as QuantMiner does, but using a pre-fixed size of rules. Algorithm CSAR, which uses a direct representation of rules, is presented in.[38] The particularity of this algorithm, based on LCS, XCS, and UCS systems,[39] is that it is focused on the use of data streams rather than static transactional data. It uses

ARs with only one variable in the consequent. CSAR can also deal with both nominal and real-valued attributes apart from a pre-specified parameter with the maximal range of intervals for continuous variables.

MODENAR,[40] proposed by Alatas et al., uses a representation of rules with three components for each attribute, indicating whether the first attribute is selected or not and if so, whether it is in the consequent or antecedent. The other two components are the lower and upper limits of ranges, respectively (see Figure 3 (b)). The same representation is used in Ref 41, but a new component is added to indicate whether the corresponding interval represents the values inside the range (e.g., "A in [a1,a2]" is added to the rule) or values outside of the range (e.g., "A in [a1,a2]" is added to the rule).

Genetic Operators

In general, classical genetic operators are the most used. However, there are special cases depending on the representation used by the algorithm. For example, in GAR,[34] variable-length individuals are used, and so, specific genetic operators are defined. In crossover, once two parents are selected, two offsprings of the same length, and each of their parents, are generated, and the values for each gene are randomly selected from a parent, which is a special case of uniform crossover. The mutation is simply to swap the value of some of the intervals present in the individual by randomly shifting the interval (right or left), or to increase or decrease their size. Finally, a process to adjust the chosen individual is carried out. This consists in decreasing the size of its intervals until the number of covered records is smaller than the records covered by the original itemset. On the other hand, QUANTMINER[36] uses a uniform crossover choosing the intervals defined by the parent selected or alternatively mixes via addition or subtraction of limit bounds of intervals. A similar crossover is defined in CSAR[38] together with three types of mutation operators: introduction/removal of antecedent variables, swapping values of variables, and changing the consequent variable.

In Alatas and Akin,[41] an informed mutation based on the algorithm UNIFORM POPULATION[42] is used and the probability of mutation is increased as the chromosomes become similar to each other. The crossover operation is also based on the UNIFORM POPULATION algorithm and the arithmetic mean of the upper and lower limits from the chromosomes selected. GENAR uses one-point crossover as the general strategy, in addition to the adaptation of the operators also used in GAR.

In the case of MODENAR,[40] mutation plays the main role in the evolutionary process, being in fact a DIFFERENTIAL EVOLUTION algorithm. The basis for the operators used is combinations (arithmetic, linear, etc.) of individuals randomly selected from the population, and it is also possible to include random noise in these combinations. Finally, a repairing operator is used in cases in which individuals fall outside the range of the attributes they represent.

Initial Population

Normally, the initialization of the population is done randomly, but there are several algorithms with particular initialization. One of them is QUANTMINER,[36] an algorithm that has a more specific initialization of the population because all numeric attributes are assigned the maximum range, specified by minimum and maximum values present in the transactional data, then individuals are created (1) decreasing the width of the ranges, and (2) once these ranges are set, then the bounds of the intervals are randomly generated. The values for nominal attributes are predetermined in the rules in the initial population, and they are the same during the evolution. Theses values are calculated via the computation of frequent itemsets by using the APRIORI algorithm.

The CSAR algorithm[38] uses a very specific strategy to generate ARs randomly. In this case, data are assumed to be obtained from a continuous flow or data stream, and therefore, instances are processed one by one. Thus, the algorithm checks whether there are enough rules in the population covering the last example/data. If it is not the case, a random generation of rules covering this example is run until a threshold (maximal number of rules) is reached. A different approach is used in MODENAR to generate the initial population that is based on a systematic use of the mutation operator.

Fitness

In general, an aggregated function is used as fitness. This function includes confidence and support measures plus some other extra criteria, such as, intervals size, etc. GAR and GENAR algorithms also include penalty terms: $f(i) = covered - (marked * \omega) - (amplitude * \phi) + (nAtr * \mu)$, where *Covered* indicates the number of records that belong to the itemset/rule that represents the individual. It is very close to support measure. *Marked* indicates that a record has been covered previously by an itemset/rule. With this measure, the authors try to force the algorithm to discover different itemsets/rules in later searches. *Amplitude* penalizes the width of intervals that make up the itemset/rule.

QuantMiner uses a modified version of the Gain measure, $Gain(\mathbf{A} \Rightarrow \mathbf{B}) = supp(\mathbf{AB}) - minconf * supp(\mathbf{A})$, where $minconf$ is the user-specified minimum confidence. A particular form of fitness is used in CSar,[38] in which a combination $(support \times confidence)^{v}$ is applied, v being a user-set parameter that permits to control the pressure toward highly fit classifiers. Besides, another set of parameters is taken into account, such as: Experience—number of times an antecedent fits an example; Consequent Sum—number of times that the entire rule fits an example; Numerosity—number of copies of the rule in the population; or Age—time of the rule in the population.

Models

Gar, Genar, and Egar use a classical evolutionary model with elitism and an aggregate fitness function instead of a multi-objective approach in the evolution. The algorithm CSar uses a particular model because of the type of data being managed; only when certain conditions are met, an elitism-based steady state GA is run to obtain a high quality set of rules. In order to ensure the quality of the rule set, the GA also used a niching scheme. Modenar uses a different approach because it implements a differential evolution algorithm rather than a GA. Moreover, a multi-objective framework is used to find the Pareto-front. For more information about this last approach see Section *Learning by Following a Multiobjective Approach*.

LEARNING FUZZY ASSOCIATION RULES

Quantitative ARs present different problems caused by the sharp boundary between intervals that are not intuitive with respect to human perception. The problem can be handled smoothly by introducing fuzziness into the model with fuzzy ARs. As claimed by Dubois et al.,[43] the use of fuzzy sets to describe associations between data extends the types of relationships that may be represented, facilitates the interpretation of rules in linguistic terms, and avoids unnatural boundaries in the partitioning of the attribute domainsÃ®. It is especially useful in domains where the boundaries of a piece of information used may not be clearly defined.

The hybridization between fuzzy logic and evolutionary algorithms, known as evolutionary fuzzy systems (EFSs),[44,45] provides novel and useful tools for pattern analysis and for extracting fuzzy ARs. In this section, we focus on EFSs for the extraction of fuzzy ARs. It must be highlighted that in this problem, the membership functions used may have a critical influence on the final mining results. Thus, in general, EFSs have been employed for ARs in the literature in the following two ways:

- for the learning or tuning of the membership functions of the fuzzy variables used to mine the fuzzy ARs, or
- for the learning (in addition to the above) of the minimum single or multiple fuzzy support.

In the following, we describe the EFSs found in the literature for ARs according to their evolutionary components.

Representation

- *Learning or tuning the membership functions*. For this data mining process, two different ways to represent the information in a chromosome have been proposed:
  1. One chromosome codifies the information for all the variables or items.

     With this idea, Wang and Bridges[46] use a representation based on a matrix of real numbers. The matrix dimensionality is $3 \times 2n$ for linguistic partitions with three linguistic labels per variable (i.e., 6 real parameters per variable) using standard Z, π, and S membership functions for them.

     Kaya and Alhajj[47] propose real-coded chromosomes with three parameters for each variable defining five triangular fuzzy sets for the corresponding linguistic labels. These authors use a different coding scheme in Ref 55, with five real parameters for each three-label linguistic partition. In this proposal, the fuzzy partition obtained is more descriptive than the one extracted in Ref 54, and the fuzzy ARs generated are weighted fuzzy ARs based on support and confidence specified as linguistic terms.

     A real coding scheme based on Parodi and Bonelli's representation[49] is applied in Refs 19 and 45 by Hong et al. In these EFSs, in order to effectively encode the membership functions for all the variables, pairs of real parameters are used to represent isosceles-triangle fuzzy set definitions.

Alcalá-Fdez et al.[52] proposed a codification scheme based on the two-tuple linguistic representation model with $n \times m$ real parameters for a set of m linguistic labels for each one of the n linguistic variables.

2. One chromosome codifies the information related to the fuzzy partition for one item or variable.

   There is only one EFS with this divide-and-conquer strategy for learning fuzzy partitions, developed by Hong et al.[53] It uses three real parameters to represent a membership function, and so $3 * |I_j|$ real numbers in a chromosome (I_j being the number of linguistic labels for item I_j, which is previously fixed). The proposed EFS maintains conceptually multiple populations (in a parallel or sequential way), each for one item's membership functions. The final solution is composed of the best individual (membership function definition) in each population. With this representation scheme, the chromosome length is short when compared with other approaches in the previous item, and so, the convergence of the solutions can easily be obtained.

- *Learning the fuzzy partitions and the single or multiple fuzzy minimum support.* For this data mining problem, when it is solved by means of an EFS, it is usual to divide the representation into two parts: one for the support/or supports and the other for the membership functions, again in one of the following two ways:

  1. One chromosome codifies the information for all the variables or items.

     Hu [47] proposed a binary chromosome representation with two components: one substring for each quantitative attribute by the encoding method proposed by Ishibuchi and Murata,[55] which considers triangular and trapezoidal membership functions; and the other for the minimum support not easily specified by the users. This representation scheme makes it possible to determine not only the shapes and parameters for the fuzzy sets but also the number of linguistic labels for each variable and the single minimum fuzzy support.

     In real applications, different items may have different criteria to judge their importance and quantitative data may exist. It can be considered thus a fuzzy data mining approach for multiple minimum support fuzzy mining problems. Within this category, Chen et al. [18] proposed an EFS with a real coding scheme in which the substring corresponding to the membership functions is codified using the same scheme as in Ref 45, that is, two real numbers per linguistic label.

  2. One chromosome codifies the information related to the fuzzy partition for one item or variable.

     Chen et al. [18] proposed an EFS for multiple minimum support fuzzy mining problems with a real coding scheme in two parts: the first part encodes minimum support of a certain item by a real number and the second one handles the set of membership functions with three real parameters for each linguistic label (with the same coding scheme used in Ref 44). It must be highlighted that the information coded in a chromosome is related to a specific item and the length for each one is the same (so, the number of linguistic labels for each item is the same for all items and pre-fixed before the genetic learning).

Genetic Operators

Most of the proposals use standard genetic operators for real coding[48, 46] and binary coding.[54] For real coding, other operators such as arithmetical crossover, max-min arithmetical, and Parent Centric BLX (PCBLX) crossover are applied in Refs 54, 19, 20, 44, 45, and 7, respectively.

Initial Population

In all the EFSs developed for the extraction of fuzzy ARs, the population is randomly generated but in Ref 20, the initial sets of chromosomes are obtained according to the initialization information provided by a k-means clustering approach.

Fitness

The EFS proposed by Wang and Bridges[46] starts from a previously obtained set of fuzzy ARs and tunes the membership functions. To do so, the fitness function used is based on the maximization for a set of normal rules and the minimization for the similarity of normal and abnormal rule sets.

Kaya and Alhajj's EFSs[47, 48] are based on the maximization of the number of all the large itemsets extracted by the membership functions represented in the chromosome. In Ref 55, this fitness function is analyzed with respect to the maximization of the average of confidence intervals, and the results for the experiments have shown that the method that employs the first fitness function outperforms the one with the average confidence intervals in terms of the required runtime and even the number of interesting rules.

In Ref 45, the fitness of each chromosome is evaluated by the number of frequent 1-itemsets and by the suitability of the membership functions (defined as a combination of the overlap and coverage ratio for the items). The evaluation cost of this fitness function is reduced in Ref 19 by dividing the chromosomes in a population of *k* clusters (using the *k*-means algorithm). All the chromosomes in a cluster then use the number of large 1-itemsets derived from the representative chromosome in the cluster and their own suitability of membership functions to calculate the fitness values. This alternative fitness function speeds up the evaluation process—due to the time-saving in finding 1-itemsets—and achieves nearly the same quality solutions as that in Ref 45.

The fitness definition in Ref 45 is adapted in Ref 44 to the chromosome representation (which codifies the fuzzy partition for only one item): the fitness value for each set of membership functions is determined according to two factors: suitability of membership functions and fuzzy supports of large 1-itemsets.

Other fitness definitions based on the use of fuzzy ARs for classification task are proposed. In Ref 47, a weighted combination of classification rate and number of fuzzy rules is used to promote a balance between accuracy and simplicity.

To jointly qualify minimum supports and membership functions in Refs 18 and 20, a fitness function is proposed that combines the requirement satisfaction defined as the closeness of the derived strength of fuzzy regions of large 1-itemsets and the suitability of the membership functions.

Models

Most of the EFS proposals for fuzzy ARs[57, 56, 51, 54, 47, 48, 46] are generational GAs with elitism. In Refs 20 and 44, EFSs are proposed that use multiple populations in a conceptual way without cooperation between them in the evolutionary process (so they are not coevolutionary proposals). These EFSs can be implemented in an iterative—sequential—or parallel way.

LEARNING BY FOLLOWING A MULTIOBJECTIVE APPROACH

AR mining can be seen as a multi-objective optimization problem rather than as a single objective one in which the different measures used to evaluate the rules (such as support, confidence) can be considered as different objectives of the association rule mining problem. In a formal way, a multi-objective optimization problem can be defined in the following way:

$$\begin{aligned} \min/\max\ y &= f(\overrightarrow{x}) \\ &= f_1(\overrightarrow{x}),\ f_2(\overrightarrow{x}), \ldots,\ f_n(\overrightarrow{x}) \end{aligned} \qquad (1)$$

where $\overrightarrow{x} = (x_1, x_2, \ldots, x_n)$ is the decision vector and $\overrightarrow{y} = (y_1, y_2, \ldots, y_n)$ is the objective vector (a tuple with n objectives). The objective of any multi-objective optimization algorithm is to find decision vectors for which the corresponding objective vectors cannot be improved in one dimension without being degraded in the other one.

The main difference between a single-objective and a multi-objective optimization task is that the solution in a single-objective task is a single optimum solution, whereas for a multi-objective optimization problem, a number of optimal solutions are obtained due to the trade-offs between conflicting objectives.

In the last two decades, an increasing interest has been shown in the use of GAs for multi-objective optimization. There are multiple proposals for multi-objective GAs,[58, 59] such as MOGA,[30] NSGA II,[60] or SPEA2,[61] for instance. In this area, most of the proposals to solve the AR mining problem using multi-objective evolutionary algorithms use GAs.

In this section, the multi-objective proposals for the extraction of ARs are described. Therefore, the representation of the individuals, the way the fitness function is used and defined, the genetic operators, and the models proposed are presented.

Individual Representation

As mentioned above, there are two basic approaches to represent the rules in a genetic learning process, namely the chromosome = BoR and chromosome = rule ones. The latter is the most common approach for multi-objective association rule mining.

Ghosh et al.[62] propose a modified chromosome = rule approach in which each attribute is tagged with two bits. If these two bits are 00, then the attribute next to these two bits appears in the antecedent part; if their values are 11, the attribute appears in the consequent part; the other two combinations, 01 and 10, indicate the absence of the attributes in both the antecedent and the consequent part of the rule. An example can be seen in Figure 2. This individual representation is also used by Dehuri et al.[25]

As explained in Section *Learning Quantitative/Numerical Association Rules*, the representation in MODENAR[40] uses three components for each attribute (the first to indicate the selection or not of the attribute, and the rest to define the interval, see Figure 3 (b)). In the works of Kaya et al.,[63, 64] the authors propose the use of real coded chromosomes with three parameters for each five-label linguistic variable or with five parameters for each three-label linguistic variable.

Fitness Evaluation

Three different approaches can be found to tackle the objectives in fitness function in multi-objective problems[65]:

- Transforming the original multi-objective problem into a single-objective problem by using a weighted function. It involves the use of a GA whose fitness function is the weighted average of different objectives.
- The lexicographical approach, in which the objectives are ranked in order of priority.
- The Pareto approach, which consists of as many non-dominated solutions as possible and returning the set of Pareto front to the user.

Most of the studies using multi-objective GAs for AR mining have been performed using the Pareto approach. Among them, Wakabi-Waiswa and Baryamureeba[26] proposed the use of *support, confidence,* and *J-Measure* as objectives in a SPEA-2[61] based association rule extraction algorithm, but computing the fitness of the rules as a weighted average using user-defined weights for these objectives; thus, this approach is very similar to the single-objective approach. Ghosh and Nath[62] proposed an evolutionary multi-objective algorithm to search for Pareto-optimal ARs formulating the problem as a three-objective optimization problem with three objectives: *confidence, comprehensibility,* and *interestingness*. In Ref 56, a five-objective formulation of the problem was suggested and a multi-objective evolutionary algorithm was employed for the identification of an optimal set of ARs on a gene expression data set. The algorithm MODENAR[40] uses four objectives: *support, confidence, comprehensibility,* and *amplitude* of the intervals that make up the itemset and rule.

The works of Ishibuchi et al.[67, 68] propose the use of multi-objective GAs to extract fuzzy ARs using *confidence* and *support* as objectives functions. In the works of Kaya et al., several proposals for the use of this type of algorithms for the extraction of fuzzy ARs have been developed: in Ref 53, the authors propose the use of *number of rules* and *execution time* as objective functions; in Ref 52, they propose *support, confidence,* and *comprehensibility*. Also, working with fuzzy ARs, Santhi-Thilagam and Ananthanarayana[69] propose a solution approach for mining optimized fuzzy association considering *fuzzy support, fuzzy confidence,* and *rule length* as objectives of the multi-objective algorithm proposed. In the study by Dehuri et al. [25] in which the authors exploit the parallelism of both data and control using a homogeneous dedicated network of workstations, *confidence, comprehensibility,* and *interestingness* are used as objective functions.

In the coevolutionary proposal of Hu and Yang-Li,[70] two new measures are used as objective functions, namely *correlation* and *comprehensibility*, to enhance the correlation degree and comprehensibility of ARs.

Genetic Operators

The use of standard multi-objective evolutionary algorithms for the AR mining also supposes the use of classical and standard crossover and mutation operators.

Models

Most of the proposals for multi-objective algorithms, aimed at solving the association rule mining problem, are multi-objective GAs, such as the works of Ghosh and Nath,[62] Khabzaoui et al.,[66] or Wakabi et al.[26]

The use of a Pareto-based multi-objective differential evolution algorithm, MODENAR, has been proposed by Alatas et al. [6] as a search strategy for mining accurate and comprehensible numeric ARs. It mines ARs directly without generating frequent itemsets.

One of the areas where multi-objective evolutionary algorithms have been applied to solve the AR mining problem is the extraction of fuzzy ARs.[67, 68, 63, 64] In Ref 6, an automated method is proposed to decide on the number of fuzzy sets and for the autonomous mining of both fuzzy sets and fuzzy ARs. Also, working with fuzzy ARs, Santhi-Thilagam and Ananthanarayana[69] propose a solution approach for mining optimized fuzzy ARs defining membership functions for all the continuous attributes in a database by using clustering techniques.

A promising area is explored in the study by Dehuri et al. [25] in which the authors propose to exploit the parallelism of both data and control by distributing the data being mined and the population of individuals across the processors of a homogeneous dedicated network of workstations using the inherent parallel processing nature of GAs.

New approaches have recently been developed using coevolutionary algorithms to solve the multiobjective optimization problem of AR mining. Hu and Yang-Li[70] propose a new coevolutionary algorithm to enhance the correlation degree and comprehensibility of ARs.

SWARM-BASED APPROACHES FOR LEARNING ARs

Besides GAs, other population-based approaches for learning ARs can be found. This is the case of ant colony optimization (ACO)[71] and particle swarm optimization (PSO).[72] Both methods fall into the category of the *Swarm Intelligence*[73] approach, in which a population of simple agents interact locally among themselves and with the environment, in order to obtain an *intelligent* global behavior, which cannot be obtained by only individual agents.

The application of ACO to AR learning[75] is based on allowing each ant (agent) to identify frequent itemsets. Thus, ants walk through the complete graph defined over all the possible items and use the frequency of buying items *i* and *j* together as heuristic information for that edge in the graph. The algorithm proposed by Kuo et al.[75] combines these ideas with the typical rules in ACO algorithms in order to identify frequent itemsets, then it obtains the ARs from them. However, another important point in Ref 58 is that the proposed algorithm is not run over the whole data set but over subsets of it that have previously been identified by (ACO-based) clustering, and in this way, the CPU time is drastically reduced.

An alternative swarm approach is proposed by Kuo et al.,[74] which consists of the application of PSO to the AR learning problem. The idea is based on using the *swarm* to determine the threshold values of support and confidence. In this approach, each *particle* codifies an association rule by following a similar representation to the one shown in Figure 1(c). At each iteration, every particle (rule) receives a fitness based on its support, confidence, and complexity (length). After each iteration, each particle updates its velocity and position by using the two best particles. Once the population converges to the same position (or a pre-defined number of iterations is exceeded), the best particle (rule) is identified, and its support and confidence are used as a threshold to mine the set of ARs.

Alatas and Akin[76] also propose the use of PSO to mine ARs. This approach is different from the previous one because the goal is to mine directly numeric ARs. Treatment of numerical values is done by using rough patterns. Thus, each particle has length $3 \cdot n$, n being the number of available variables or items. For each item, two positions of the particle are used to codify the lower and upper limit of the rough pattern (interval), while the third one is used to codify its position in the corresponding AR (antecedent, consequent, or none). The evolutionary process is guided by a fitness function that *decides* the amplitude of the intervals in order to obtain interesting rules. In Alatas and Akin,[77] the authors extend their study by replacing rough patterns with chaos numbers, which instead of using lower and upper limits, consists of midpoint and radius of the values as opposed to precise values.

APPLICATIONS

Although the applications of association rule learning are extensive, here, we restrict ourselves to those cases in which evolutionary algorithms have been used. Thus, the following applications can be found:

- Business:
  - Investors' stock purchase behavior. Kuo et al.[74] propose to apply their AR learning method based on PSO (see Section *Swarm-Based Approaches for Learning ARs*) to carry out a study of investors' stock purchase behavior for a security firm in Taiwan.
  - Marketing: Consumer behavior modeling. Casillas et al.[78] propose the use of ACO to discover a set of fuzzy ARs to model consumer behavior with the goal of explaining customer trust in Internet shopping. Because the number of variables is limited (6) and the target ones are known (2), the authors propose to use an advanced methodology (COR,[79]) to learn the fuzzy rules. A specific ACO algorithm is used as a search engine inside COR. The result is compared with the one obtained by using structural equation modeling. Orriols-Puig et al.[80] also

propose to apply fuzzy ARs to this problem. However, they use a specific algorithm (CSAR) based on GAs to directly evolve a population of ARs. That is, a purely descriptive approach is followed instead of adapting a predictive one as in Ref 17.

- Medicine:
  - Management. Discovery of hidden relationships between diseases from (a small sample of) the data set provided by the National Health Insurance Research Database (Taiwan). Kuo et al.[75] carried out this study by using their proposal for ARs learning based on ACO.
  - Analysis of cancer data sets. Kwasnicka and Switalski[35] analyzed two medical data sets (Sutek–breast cancer and Szyjka–cancer of the cervix/uterus) by using ARs. The algorithm used (EGAR) is based on the use of a GA combined with Michigan representation. EGAR is able to deal with both qualitative and quantitative attributes. According to the authors, interesting results were obtained, specially when forcing some variables (e.g., period of survival and time of cancer recurrence) to be included in the rules.
  - Study of risk factors of atherosclerosis. Salleb-Aouissi et al.[81] carried out a study over the Stulong data set, which contains information of a 20 years' lasting study of the risk factors of the atherosclerosis in a population of 1419 middle- aged men. The study is based on applying QUANTMINER algorithm[36] for the discovering of quantitative ARs, and the main goal is to obtain descriptions for some concrete variables, for example, limiting the left hand side of the AR to be Death? = true (false).
- Bioinformatics:
  - Analysis of gene expression data. Khabzaoui et al.[66] apply knowledge discovery in the form of ARs to the analysis of gene expression data in order to identify patterns of genes and regulatory network. A multi-criteria GA is used to discover qualitative ARs from microarray data, where gene expression levels have been previously discretized (e.g., overexpressed, underexpresed). Because of the multi-criteria approach used, which combines many scores in the fitness function, a large number of rules are returned. The authors have developed a visualizing tool to help the selection of rules in a multi-criteria context.
  - Generation of ARs from spatial gene expression data. Anandhavalli et al.[82] propose a GA to perform global searching for generating interesting ARs from spatial gene expression data. The novelty in this approach is the fact that it is not necessary for the users to specify thresholds. Instead of generating an unknown number of ARs, only the most interesting rules are generated according to interestingness measure as defined by the fitness function.
  - Mine gene network from large-scale gene expression data. Du et al.[83] propose to combine qualitative ARs with GAs to mine gene network from large-scale gene expression data. Because not all the rules mined by classical AR discovering algorithm (e.g., APRIORI) are biologically meaningful, the authors propose to optimize the obtained rules, thus decreasing the number of redundant ones by using a tailored GA. The results show that the proposed method is able to discover some important interactions between genes from global gene expression data sets.
  - Gene expression associative classification. He and Hui[84] investigate ant-based algorithms to discover *class-based* ARs for gene expression associative classification. In this process, the consequent of the rule is set to be a label of the class variable, and the ant colony aims to identify the left hand side of the rule. The algorithm follows a classical ant colony procedure, and the graph contains all the genes as nodes (1-itemsets) whose support is greater than *mins*. During the tour, the ant tries to add new

TABLE 2 | Summary of Described Applications

	Qualitative ARs		Quantitative ARs		Fuzzy ARs	
Business	Kuo et al.[74]	PSO			Orriols-Puig et al.[80]	GA
					Casillas et al.[78]	ACO
Medicine	Kuo et al.[75]	PSO	Kwasnicka et al.[35]	GA		
			Salleb et al.[81]	GA		
BioInformatics	Khabzaoui et al.[66]	GA				
	Anandhavalli et al.[82]	GA				
	Du et al.[83]	GA				
	He et al.[84]	ACO				
Education	Romero et al.[85]	GA				
	Romero et al.[86]	GA				
Manufacture	Li et al.[87]	GA			Wang et al.[88]	GA
Others	Guillet et al.[89]	GA			Dhanalakshmi et al.[90]	GA
	Venugopal et al.[28]	GA	Orriols-Puig et al.[38]	GA		

genes to the current itemset, one at each time; however, if the confidence of the enlarged rule falls under *minc*, the node (gene) under study is discarded. The proposed algorithm Ant-ARM has been tested on the acute lymphoblastic leukemia (ALL)/acute myeloid leukemia (AML) data set, generating about 30 high accuracy classification rules.

- Education:
  - E-learning. Romero et al.[85] apply GAs and ARs to the problem of mining student information from a Web-based Educational Adaptive Hypermedia System. The objective is to obtain interesting ARs so that the teacher can improve the performance of the system. The GA fitness function is defined in order to discover *strong* qualitative ARs. The same approach is used by Romero et al.,[86] with the aim to perform data mining in three-levels of difficulty course implemented in the AHA! system, in order to find good candidates for meaningful relations between reading times, difficulty levels, and test results.
- Manufacture:
  - Extraction of qualitative ARs from a manufacturing information system. Li and Yang[87] use a GA to mine a set of qualitative ARs from a manufacturing information system (MIS) data set. Continuous variables are discretized previously, and a comparison with Apriori algorithm shows that the proposed model is more efficient when dealing with large itemsets.
  - Efficient management of networked manufacturing resources. Wang et al.[88] propose the use of fuzzy ARs learned by using a GA to deal with the problem of choosing and managing networked manufacturing resources efficiently. Considering the characteristics of networked manufacturing resources, double-level encoding and label-bit switching operator were designed to make the improved GA available to networked manufacturing resources. As a result, some useful fuzzy ARs are discovered that provide decision support for manufacturing resources management.
- Others:
  - Visualization. GAs have been used not only to discover the set of interesting ARs from the data but also to obtain a friendly representation of them. This is the case of the proposal by Guillet et al.,[89] where a GA for drawing AR rule graphs is presented that incorporates the restriction of dealing with a dynamic layout. Thus, new solutions close to the previous ones are included very quickly when slight modifications are inserted.

- Computer security. Dhanalakshmi and Ramesh-Babu[90] propose the use of a GA to mine fuzzy ARs in order to find associations among different sets of security features. The reason for using fuzzy ARs lies in the fact that in this problem, there are many quantitative features in which there is no clear separation between normal operations and anomalies.
- Data streams. For those cases in which the database is dynamic, for example, a data stream of transactions, algorithms must be able to adapt the current AR set without having to re-explore the full data set. Evolutionary algorithms have recently been proposed to tackle this problem.[38, 28] For example, Orriols-Puig et al.[38] presented the algorithm CSAR, a Michigan-like GA based on online learning that incrementally evolves the learnt knowledge, and is able to adapt quickly the discovered AR set to the appearance or disappearing of associations among the variables with the arrival of new data.

A summary of the aforementioned applications is shown in Table 2, classified by type of AR used and application field. For each application, the metaheuristic technique used is shown. From this table, it is easy to conclude that boolean/qualitative ARs and GAs have been the most applied options. However, this may change in the future, specially because real applications usually deal with numerical variables, and there are currently many approaches to cope directly with them, avoiding the need of an *a priori* discretization.

CONCLUSIONS

A survey of research into AR learning using evolutionary algorithms has been given in this paper, and we have tried to cover both classical and recent literature related to the topic.

The main properties and elements of this task have been presented, specially those related with approaches using evolutionary algorithms. Approaches using different types of variables (Boolean, categorical, or numerical) have been studied. The use of fuzzy rules and multi-objective approaches for AR mining has also been covered, and some approaches based on metaheuristics that are different from GAs have been reviewed. In addition, applications of AR approaches to real-world problems have been presented.

NOTE

[a] Notation: we use uppercase letters to denote variables and boldface uppercase letters to denote sets of variables. We use lowercase letters to denote values of variables ($\{i_1, i_2, i_3\}$, $\{t, f\}$, etc.) and boldface lowercase letters to denote configurations of values for a set of variables. For binary variables (e.g. $A = \{t, f\}$) we will sometimes use A for $A = t$, while the absence of A will mean $A = f$.

ACKNOWLEDGEMENTS

This paper was supported by the Spanish Ministry of Education, Social Policy and Sports under projects TIN2008-06681-C06-02 and TIN2007-67418-C03-01, by the Andalusian Research Plan under project TIC-3928, and by the JCCM Research Plan under project PCI08-0048-8577.

REFERENCES

1. Agrawal R, Imieliński T, Swami A. Mining association rules between sets of items in large databases. In: Buneman P, Jajodia S, eds. *SIGMOD '93: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*. New York: ACM Press; 1993, 207–216.
2. Piatetsky-Shapiro G. Discovery, Analysis, and Presentation of Strong Rules. In: Piatetsky-Shapiro G,

Frawley WJ, eds. *Knowledge Discovery in Databases*. Cambridge, MA: AAAI/MIT Press; 1991.

3. Blattberg RC, Kim B, Neslin SA. Market basket analysis. In: *Database Marketing (chapter 13, Market Basket Analysis)*. New York: Springer; 2008, 39–35.
4. Natarajan R, Shekar B. Interestingness of association rules in data mining: Issues relevant to e-commerce. *Sadhana*, 30:291–309, 2005.
5. Fong J, Hughes J, Zhu J. Online web mining transactions association rules using frame metadata model. In: *Proceedings of the First International Conference on Web Information Systems Engineering (WISE'00)*. IEEE Computer Society; 2000, 2121–2130.
6. Mao P, Zhu Q. Association rules applied to intrusion detection. *Wuhan University Journal of Natural Sciences*, 7:426–430, 2002.
7. Martinez R, Pasquier N, Pasquier C. GenMiner: mining non-redundant association rules from integrated gene expression data and annotations. *Bioinformatics*, 24(22):2643–2644, 2008.
8. Brin S, Motwani R, Ullman J, Tsur S. Dynamic itemset counting and implication rules for market basket data. In: Peckham J, ed. *SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data*. New York: ACM Press; 1997, 255–264.
9. Agrawal R, Srikant R. Fast algorithms for mining association rules in large databases. In: Bocca JB, Jarke M, Zaniolo C, eds. *VLDB '94: Proceedings of the 20th International Conference on Very Large Data Bases*. San Francisco, CA: Morgan Kaufmann Publishers Inc.; 1994, 487–499.
10. Zaki MJ. Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering*, 12(3):372–390, 2000.
11. Han J, Pei J, Yin Y, Mao R. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Mining and Knowledge Discovery*, 8(1):53–87, 2004.
12. Zhang M, He C. Survey on association rules mining algorithms. In: Luo Q, ed. *Lecture Notes in Electrical Engineering*. Berlin Heidelberg: Springer; 2010, 56:111–118.
13. Tan P, Steinbach M, Kumar V. *Introduction to Data Mining*. Boston, MA: Addison-Wesley Longman Publishing Co., Inc.; 2005.
14. Blum C, Roli A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3):268–308, 2003.
15. Eiben AE, Smith J. *Introduction to Evolutionary Computing*. Berlin Heidelberg: Springer-Verlag; 2003.
16. Goldberg DE. *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*. Boston, MA: Kluwer Academic Publishers; 2002.
17. Larrañaga P, Lozano JA, eds. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Boston, MA: Kluwer Academic Publishers; 2001.
18. del Jesus MJ, Gámez JA, Puerta JM. Evolutionary and metaheuristics based data mining. *Soft Computing*, 13(3):209–212, 2009.
19. Freitas AA. *Data Mining and Knowledge Discovery with Evolutionary Algorithms*. New York: Springer-Verlag; 2002.
20. Fernández A, García S, Luengo J, Bernadó-Mansilla E, Herrera F. Genetics-based machine learning for rule induction: State of the art and taxonomy and comparative study. *IEEE Trans Evol Comput*. In press.
21. Smith SF. *A learning system based on genetic adaptive algorithms*. PhD thesis, Pittsburgh, PA, USA, 1980.
22. Booker LB, Goldberg DE, Holland JH. Classifier systems and genetic algorithms. *Artificial Intelligence*, 40(1-3):235–282, 1989.
23. Venturini G. Sia: A supervised inductive algorithm with genetic search for learning attributes based concepts. In: Lavrac N, Gamberger D, Todorovski L, Blockeel H, eds. *ECML '93: Proceedings of the European Conference on Machine Learning*. Berlin, Heildelberg: Springer-Verlag; 1993, 280–296.
24. Greene DP, Smith SF. Competition-based induction of decision models from examples. *Machine Learning*, 13(2-3):229–257, 1993.
25. Dehuri S, Jagadev AK, Ghosh A, Mall R. Multi-objective genetic algorithm for association rule mining using a homogeneous dedicated cluster of workstations. *American Journal of Applied Sciences*, 3:2086–2095, 2006.
26. Wakabi-Waiswa PP, Baryamureeba V. Extraction of interesting association rules using genetic algorithms. *International Journal of Computing and ICT Research*, 2(1):1818–1139, 2008.
27. Shenoy PD, Srinivasa KG, Venugopal KR, Patnaik LM. Evolutionary approach for mining association rules on dynamic databases. In: Whang KY, Jeon J, Shim K, Srivastava J, eds. *PAKDD '03: Proceedings of the 7th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, volume Lecture Notes in Artificial Intelligence LNAI 2637. Berlin Heildelberg: Springer-Verlag; 2003, 325–336.
28. Shenoy PD, Srinivasa KG, Venugopal KR, Patnaik LM. Dynamic association rule mining using genetic algorithms. *Intelligent Data Analysis*, 9(5):439–453, 2005.
29. Yan X, Zhang C, Zhang S. Genetic algorithm-based strategy for identifying association rules without specifying actual minimum support. *Expert Systems with Applications*, 36(2):3066–3076, 2009.
30. Fonseca C, Fleming P. Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. In: *ICGA'93: Fifth International Conference on Genetic Algorithms*. 1993, 207–216.
31. Yang Y, Webb G, Wu X. Discretization methods. In: Maimon O, Rokach L, eds. *Data Mining and*

Knowledge Discovery Handbook. 2nd ed. Springer, NY: Springer; 2010, 113–130.

32. Aumann Y, Lindell Y. A statistical theory for quantitative association rules. *J Intell Inf Syst*, 20(3):255–283, 2003.
33. Webb GI. Discovering associations with numeric variables. In: *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. New York: ACM; 2001, 383–388.
34. Mata J, Álvarez J, Riquelme J. Discovering numeric association rules via evolutionary algorithm. In: Cheng MS, Yu PS, Liu B, eds. *PAKDD '02: Proceedings of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*. Berlin Heildelberg: Springer-Verlag; 2002, 40–51.
35. Kwasnicka H, Switalski K. Discovery of association rules from medical data - classical and evolutionary approaches. In: Ganzha M, Paprzycki M, Wachowicz J, Węcel K, eds. *Proceedings of the XI Autumn Meeting of Polish Information Processing Society*. Wisla, Poland: Polish Information Processing Society; 2005, 163–177.
36. Salleb-Aouissi A, Vrain C, Nortet C. Quantminer: a genetic algorithm for mining quantitative association rules. In: Veloso M, ed. *IJCAI'07: Proceedings of the 20th international joint conference on Artifical intelligence*. San Francisco, CA: Morgan Kaufmann Publishers Inc.; 2007, 1035–1040.
37. Mata J, Álvarez J, Riquelme J. Mining numeric association rules with genetic algorithms. In Science SC, editor, *5th Internacional Conference on Artificial Neural Networks and Genetic Algorithms*, volume Artificial Neural Nets and Genetics Algorithms, pages 264–267, 2001.
38. Orriols-Puig A, Casillas J, Bernadó-Mansilla E. First approach toward on-line evolution of association rules with learning classifier systems. In: Keijzer M, ed. *GECCO '08: Proceedings of the 2008 GECCO conference companion on Genetic and evolutionary computation*. New York: ACM Press; 2008, 2031–2038.
39. Bernadó E, Garrell J. Accuracy-based learning classifier systems: Models, analysis and applications to classification tasks. *Evolutionary Computation*, 11(3):209–238, 2003.
40. Alatas B, Akin E, Karci A. Modenar: Multi-objective differential evolution algorithm for mining numeric association rules. *Applied Soft Computing*, 8(1):646–656, 2008.
41. Alatas B, Akin E. An efficient genetic algorithm for automated mining of both positive and negative quantitative association rules. *Soft Comput 2006*, 10(3):230–237.
42. Karci A. Novelty in the generation of initial population for genetic algorithms. In: *KES*. 2004, 268–275.
43. Dubois D, Prade H, Sudkamp T. On the representation, measurement and discovery of fuzzy associations. *IEEE Transactions on Fuzzy Systems*, 13:250–262, 2005.
44. Cordón O, Herrera F, Hoffmann F, Magdalena L. *Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*. Singapore: World Scientific; 2001.
45. Herrera F. Genetic fuzzy systems: taxonomy, current research trends and prospects. *Evolutionary Intelligence*, 1:27–46, 2008.
46. Wang W, Bridges SM. Genetic algorithm optimization of membership functions for mining fuzzy association rules. In: Wang PP, ed. *International Joint Conference on Information Systems, Fuzzy Theory and Tecnology*. Atlantic City, NY: Association for Intelligent Machinery; 2000, 1–4.
47. Kaya M, Alhajj R. Genetic algorithm based framework for mining fuzzy association rules. *Fuzzy Sets and Systems*, 152:587–601, 2005.
48. Kaya M, Alhajj R. Utilizing genetic algoriths to optimize membership functions forfuzzy weigted association rules mining. *Applied Intelligence*, 24:7–15, 2006.
49. Parodi A, Bonelli P. A new approach of fuzzy classifier systems. In: Forrest S, ed. *5th International Conference on Genetic Algorithms*. San Francisco, CA: Morgan Kaufmann Publishers Inc.; 1993, 223–230.
50. Chen C, Hong T, Tseng VS. Speeding up genetic-fuzzy mining by fuzzy clustering. In: *2009 IEEE International Conference on Fuzzy Systems*. Washington, DC: IEEE Computer Society; 2009, 1695–1699.
51. Hong TP, Chen CH, Wu YL, Lee YC. A ga-based fuzzy mining approach to achieve a trade off between number of rules and suitability of membership functions. *Soft Computing*, 10:1091–1102, 2006.
52. Alcalá-Fdez J, Alcalá R, Gacto MJ, Herrera F. Learning the membership function contexts for mining fuzzy association rules by using genetic algorithms. *Fuzzy Sets and Systems*, 160:905–921, 2009.
53. Hong TP, Chen CH, Lee YC, Wu YL. Genetic-fuzzy data mining with divide and-conquer strategy. *IEEE Transactions on Evolutionary Computation*, 12(2):252–265, 2008.
54. Hu YC. Determining membership functions and minimum fuzzy support in finding fuzzy association rules for classification problems. *Knowledge-Based Systems*, 19:57–66, 2006.
55. Ishibuchi H, Murata T. A genetic-algorithm-based fuzzy partition method for pattern clasification problems. In: Herrera F, Verdegay JL, eds. *Genetic Algorithms and Soft Computing*. Berlin Heidelberg: Physica-Verlag; 1996, 555–578.
56. Chen CH, Tseng V, Hong TP. An improved approach to find membership functions and multiple minimum

supports in fuzzy data mining. *Expert Systems with Applications*, 36:10016–10024, 2009.

57. Chen CH, Tseng V, Hong TP. Cluster-based evaluation in fuzzy-genetic data mining. *IEEE Transactions on Fuzzy Systems*, 16(1):249–292, 2008.
58. Coello C, Veldhuizen DV, Lamont G. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Secaucus, NJ: Springer-Verlag; 2006.
59. Deb K. *Multi-Objective Optimization using Evolutionary Algorithms*. New York: Wiley & Sons; 2001.
60. Deb K, Pratap A, Agarwal A, Meyarivan T. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2002.
61. Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In: Giannakoglou KC, Tsahalis DT, Periaux J, Fogarty T, eds. *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001)*. Barcelona: CIMNE; 2002, 95–100.
62. Ghosh A, Nath B. Multi-objective rule mining using genetic algorithms. *Information Sci*, 173:123–133, 2004.
63. Kaya M. Multi-objective genetic algorithm based approaches for mining optimized fuzzy association rules. *Soft Computing*, 10:578–586, 2006.
64. Kaya M, Alhajj R. Integrating multi-objective genetic algorithms into clustering for fuzzy association rules mining. In: Rastogi R, Morik K, Bramer M, Wu X, eds. *4th IEEE International Conference Data Mining*. Washington, DC: IEEE Computer Society; 2004, 431–434.
65. Ghosh A, Dehuri S, Ghosh S, eds. *Multi-Objective Evolutionary Algorithms for Knowledge Discovery from Databases*. Studies on Computational Intelligence 98. Berlin Heidelberg: Springer; 2008.
66. Khabzaoui M, Dhaenens C, Talbi E. A multicriteria genetic algorithm to analyze microarray data. In: Greenwood G, ed. *IEEE Congress Evolutionary Computation vol. 2*. Washington, DC: IEEE Computer Society; 2004, 1874–1881.
67. Ishibuchi H, Kuwajima I, Nojima Y. Evolutionary multi-objective rule selection for classification rule mining. In: Ghosh A, Dehuri S, Ghosh S, eds. *Studies in Computational Intelligence 98*. Berlin Heidelberg: Springer; 2008, 47–79.
68. Ishibuchi H, Yamamoto T. Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining. *Fuzzy Sets and Systems*, 141:50–88, 2004.
69. Santhi-Thilagam P, Ananthanarayana VS. Extraction and optimization of fuzzy association rules using multi-objective genetic algorithm. *Pattern Analysis and Applications*, 11(2):159–168, 2008.
70. Hu J, Yang-Li X. Association rules mining using multi-objective coevolutionary algorithm. In: Wang Y, ed. *CIS Workshops 2007, 2007 International Conference on Computational Intelligence and Security Workshops*. Washington, DC: IEEE Computer Society; 2007, 405–408.
71. Dorigo M, Stützle T. *Ant Colony Optimization*. Cambridge, MA: Bradford Book; 2004.
72. Shi Y, Eberhart R. A modified particle swarm optimizer. In: Simpson PK, Haines K, Zurada J, Fogel D, eds. *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*. Washington, DC: IEEE Computer Society; 1998, 69–73.
73. Bonabeau E, Dorigo M, Theraulaz G. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford: Oxford University Press; 1999.
74. Kuo R, Chao C, Chiu Y. Application of particle swarm optimization to association rule mining. *Applied Soft Computing*, In Press, 2009.
75. Kuo RJ, Lin SY, Shih CW. Mining association rules through integration of clustering analysis and ant colony system for health insurance database in taiwan. *Expert Systems with Applications*, 33(3):794–808, 2007.
76. Alatas B, Akin E. Rough particle swarm optimization and its applications in data mining. *Soft Computing*, 12(12):1205–1218, 2008.
77. Alatas B, Akin E. Chaotically encoded particle swarm optimization algorithm and its applications. *Chaos, Solitons & Fractals*, 41(2):939–950, 2009.
78. Casillas J, Martínez-López F, Martínez F. Fuzzy association rules for estimating consumer behaviour models and their application to explaining trust in internet shopping. *Fuzzy Economic Review*, 9(2):3–26, 2004.
79. Casillas J, Cordón O, Herrera F. COR: a methodology to improve ad hoc data-driven linguistic rule learning methods by inducing cooperation among rules. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 32(4):526–537, 2002.
80. Orriols-Puig A, Casillas J, Martínez-López F. Unsupervised learning of fuzzy association rules for consumer behavior modeling. *Mathware and Soft Computing*, 16:29–43, 2009.
81. Salleb-Aouissi A, Turmeaux T, Vrain C, Nortet C. Mining quantitative association rules in a atherosclerosis dataset. In: Berka P, Cremilleux B, eds. *PKDD Discovery Challenge*. 2004, 98–103. Available at: http://lisp.vse.cz/challenge/ecmlpkdd2004/.
82. Anandhavalli M, Ghose MK, Gauthaman K, Boosha M. Global search analysis of spatial gene expression data using genetic algorithm. In: Meghanathan N, Boumerdassi S, Chaki N, Nagamalai D, eds. *Recent Trends in Network Security and Applications (Series on Communications in Computer and Information Science, Volume 89)*. Berlin Heidelberg: Springer Verlag; 2010, 593–602.

83. Du F, Rao N, Guo J, Yuan Z, Wang R. Mining gene network by combined association rules and genetic algorithm. In: Zhu C, Au OC, Liao W, Fang Y, eds. *Proceedings of the Int. Conf. on Communications, Circuits and Systems*. Washington, DC: IEEE Computer Society; 2009, 581–585.

84. He Y, Hui S. Exploring ant-based algorithms for gene expression data analysis. *Artificial Intelligence in Medicine*, 47(2):105–119, 2009.

85. Romero C, Ventura S, Castro CD, Hall W, Ng MH, Ng H. Using genetic algorithms for data mining in web-based educational hypermedia systems. In: Brusilovsky P, Henze N, Millán E, eds. *in Proceedings of the Workshop Adaptive Systems for Web-based Education*. Málaga, Spain: Universidad de Málaga; 2002, 137–142.

86. Romero C, Bra PD, Ventura S, Castro CD. Using knowledge levels with aha! for discovering interesting relationships. In: Driscoll M, Reeves TC, eds. *Proceedings of the World Conference on E-Learning in Corporate, Government, Healthcare & Higher Education*. Norfolk, VA: Association for the Advancement of Computing in Education; 2002, 2721–2722.

87. Li C, Yang M. Association rules data mining in manufacturing information system based on genetic algorithms. In: Benqing G, Xiaowen X, eds. *Proceedings of the 3rd Int. Conf. on Computational Electromagnetics and Its Applications*. Washington, DC: IEEE Computer Society; 2004, 153–156.

88. Wang B, Zhou N, Liu D, Zhou L, Wang P. Genetic algorithm-based rules discovery for networked manufacturing resources management. In: Zhou H, Cuthbert L, Wang ZT, eds. *Proceedings of the 4th Int. Conf. on Wireless Communications, Networking and Mobile Computing*. Washington, DC: IEEE Computer Society; 2008, 1–4.

89. Guillet F, Kuntz P, Lehn R. A genetic algorithm for visualizing networks of association rules. In: Imam I, Kodratoff Y, El-Dessouki A, Ali M, eds. *IEA/AIE '99: Proceedings of the 12th international conference on Industrial and engineering applications of artificial intelligence and expert systems*. Berlin Heidelberg: Springer-Verlag; 1999, 145–154.

90. Dhanalakshmi Y, Ramesh-Babu I. Intrusion detection using data mining along fuzzy logic and genetic algorithms. *International Journal of Computer Science and Network Security*, 8(2):27–32, 2008.