Soft Comput
DOI 10.1007/s00500-011-0705-4

ORIGINAL PAPER

An evolutionary algorithm to discover quantitative association rules in multidimensional time series

M. Martínez-Ballesteros · F. Martínez-Álvarez ·
A. Troncoso · J. C. Riquelme

Abstract An evolutionary approach for finding existing relationships among several variables of a multidimensional time series is presented in this work. The proposed model to discover these relationships is based on quantitative association rules. This algorithm, called QARGA (Quantitative Association Rules by Genetic Algorithm), uses a particular codification of the individuals that allows solving two basic problems. First, it does not perform a previous attribute discretization and, second, it is not necessary to set which variables belong to the antecedent or consequent. Therefore, it may discover all underlying dependencies among different variables. To evaluate the proposed algorithm three experiments have been carried out. As initial step, several public datasets have been analyzed with the purpose of comparing with other existing evolutionary approaches. Also, the algorithm has been applied to synthetic time series (where the relationships are known) to analyze its potential for discovering rules in time series. Finally, a real-world multidimensional time series composed by several climatological variables has been considered. All the results show a remarkable performance of QARGA.

M. Martínez-Ballesteros · J. C. Riquelme (✉)
Department of Computer Science,
University of Seville, Seville, Spain
e-mail: riquelme@us.es

M. Martínez-Ballesteros
e-mail: mariamartinez@us.es

F. Martínez-Álvarez · A. Troncoso
Department of Computer Science,
Pablo de Olavide University, Seville, Spain
e-mail: fmaralv@upo.es

A. Troncoso
e-mail: ali@upo.es

Keywords Time series · Quantitative association rules · Evolutionary algorithms · Data mining

1 Introduction

It is usual to find natural phenomena correlated to some other variables. Thus, real-world processes can be modeled by inferring knowledge from other associated variables that definitively have an effect on the original process. For instance, the existence of acid rain cannot be understood without the existence of other pollutant agents, such as monoxide carbon or sulfur dioxide. In other words, the knowledge of how some variables could affect other ones may be useful to obtain accurate behavior models.

Quantitative association rule (QAR) extraction in time series can be of the utmost usefulness for predictive purposes (Shidara et al. 2008; Wang et al. 2008). Thus, it could be interesting to find relationships among several time series to determine the range of values for a particular time series in a given time interval depending on the values of others for the same interval. For instance, rules such as $hour \in [10, 12] \wedge demand \in [12,000, 15,000] \Rightarrow price \in [3.2, 4.5]$ can provide useful knowledge for forecasting the electric energy price at peak hours (from 10 am to 12 pm) depending on the values of the energy demand during these hours. This information could help to obtain different models adjusted to different intervals or to develop a family of models for every rule. Hence, QAR are introduced in a new time series framework with the means of obtaining relationships among correlated time series that help to model their behavior.

Evolutionary algorithms (EA) have been extensively used for optimization and model adjustment in data mining tasks. In fact, the use metaheuristics in general, and of EA

Published online: 22 March 2011

in particular, to deal with data mining-based problems is a hot topic of research nowadays (Alcalá-Fdez et al. 2009a, 2010; Chen et al. 2010; del Jesús et al. 2009; Yan et al. 2009). Also, EA have been used to build rule-based systems (Aguilar-Ruiz et al. 2007; Berlanga et al. 2010; Orriols-Puig and Bernadó-Mansilla 2009).

Real-coded genetic algorithms (RCGA) are very important within EA due to the increasing interest in solving real-world optimization problems. The main problem of RCGA, in which many researchers have focused their works, is the definition of adequate genetic operators (Herrera et al. 2004; Kalyanmoy et al. 2002). In particular, a new RCGA, henceforth called QARGA (Quantitative Association Rules by Genetic Algorithm) is proposed in this work. It is worth noting that QARGA does not perform previous variable discretization, that is, it handles numeric data during the whole rule extraction process, in contrast with many other approaches that perform data discretization to discover rules (Agrawal et al. 1993; Aumann and Lindell 2003; Vannucci and Colla 2004). Furthermore, the approach allows several degrees of freedom in specifying the user's preference regarding both of the number of attributes and structure of the rules. On the other hand, besides the well-known support and confidence measures, the accuracy of the rules is also obtained with a measure called lift due to its usefulness in the specific area of time series analysis (Ramaswamy et al. 1998).

First, QARGA has been applied to datasets from the Bilkent University Function Approximation (BUFA) repository (Guvenir and Uysal 2000). These datasets have been chosen because the literature offers multiple EA applied to them (Alatas and Akin 2006; Alatas et al. 2008; Mata et al. 2002). Later, time series have been synthetically generated to determine the suitability of applying QARGA to temporal data. Finally, multidimensional real-world time series have been used to extract QAR. In particular, climatological time series have been analyzed to discover the factors that cause high ozone concentration levels in atmosphere.

The remainder of the paper is divided as follows: Sect. 2 provides a formal description of QAR, as well as introduces the quality indices applied to QARGA. Section 3 presents the most relevant related works found in literature. Section 4 describes the main features of QARGA used in this work. The results of applying the proposed algorithm to different datasets are reported and discussed in Sect. 5. Finally, Sect. 6 summarizes the conclusions.

2 Preliminaries

This section is devoted to formally describe QAR and to introduce the quality measures used in this paper.

2.1 Quantitative association rules

Association rules (AR) were first defined by Agrawal et al. (1993) as follows. Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of n items, and $D = \{tr_1, tr_2, \ldots, tr_N\}$ a set of N transactions, where each tr_j contains a subset of items. Thus, a rule can be defined as $X \Rightarrow Y$, where $X, Y \subseteq I$ and $X \cap Y = \emptyset$. Finally, X and Y are called antecedent (or left side of the rule) and consequent (or right side of the rule), respectively.

When the domain is continuous, the association rules are known as QAR. In this context, let $F = \{F_1, \ldots, F_n\}$ be a set of features, with values in $\mathbb{R}$. Let A and C be two disjunct subsets of F, that is, $A \subset F$, $C \subset F$, and $A \cap C = \emptyset$. A QAR is a rule $X \Rightarrow Y$, in which features in A belong to the antecedent X, and features in C belong to the consequent Y, such that

$$X = \bigwedge_{F_i \in A} F_i \in [l_i, u_i] \tag{1}$$

$$Y = \bigwedge_{F_j \in C} F_j \in [l_j, u_j] \tag{2}$$

where l_i and l_j represent the lower limits of the intervals for F_i and F_j, respectively, and the couple u_i and u_j the upper ones. For instance, a QAR could be numerically expressed as

$$F_1 \in [12, 25] \wedge F_3 \in [5, 9] \Rightarrow F_2 \in [3, 7] \wedge F_5 \in [2, 8] \tag{3}$$

where F_1 and F_3 constitute the features appearing in the antecedent and F_2 and F_5 the ones in the consequent.

2.2 Quality parameters

This section provides a description of the support, confidence and lift indices (Brin et al. 1997) used to measure the interestingness of rules and of a new index, called *recovered*, to ensure that the full search space is explored.

The support of an itemset X is defined as the ratio of transactions in the dataset that contain X. Formally:

$$sup(X) = \frac{\#X}{N} = P(X) \tag{4}$$

where $\#X$ is the number of times that X appear in the dataset, and N the number of transactions forming such dataset. Other authors prefer naming the support of X simply as the probability of X, $P(X)$.

Let X and Y be the itemsets that identify the antecedent and consequent of a rule, respectively. The confidence of a rule is expressed as follows:

$$conf(X \Longrightarrow Y) = \frac{sup(X \Longrightarrow Y)}{sup(X)} \tag{5}$$

and it can be interpreted as the probability that transactions containing X, also contain Y. In other words, how certain is the rule subjected to analysis.

Finally, the interest or lift of a rule is defined as

$$lift(X \Longrightarrow Y) = \frac{sup(X \Longrightarrow Y)}{sup(X)sup(Y)} \quad (6)$$

Lift means how many times more often X and Y are together in the dataset than expected, assuming that the presence of X and Y in transactions are occurrences statically independent. Lifts greater than one are desired because this fact would involve statistical dependence in simultaneous occurrence of X and Y and, therefore, the rule would provide valuable information about X and Y.

For a better understanding of such indices, a dataset comprising ten transactions and three features is shown in Table 1. Also consider an example rule

$$F_1 \in [180, 189] \wedge F_2 \in [85, 95] \Rightarrow F_3 \in [33, 36] \quad (7)$$

In this case, the support of the antecedent is 20%, since two transactions, t_2 and t_9, simultaneously satisfy that F_1 and F_2 belong to the intervals [180, 189] and [85, 95], respectively (two transactions out of ten, $sup(X) = 0.2$). As for the support of the consequent, $sup(Y) = 0.2$ because only transactions t_6 and t_9 satisfy that $F_3 \in [33, 36]$. Regarding the confidence, only one transaction t_9 satisfies all the three features (F_1 and F_2 in the antecedent, and F_3 in the consequent) appearing in the rule; in other words, $sup(X \Rightarrow Y) = 0.1$. Therefore, $conf(X \Rightarrow Y) = 0.1/0.2 = 0.5$, that is, the rule has a confidence of 50%. Finally, the lift is $lift(X \Longrightarrow Y) = 0.1/(0.2 * 0.2) = 2.5$, since $sup(X \Rightarrow Y) = 0.1, sup(X) = 0.2$ and $sup(Y) = 0.2$, as discussed before.

Finally, the measure *recovered* is defined for finding rules covering different regions of the search space. An example e is covered by the rule r if the values of attributes of e belong to the intervals defined by the rule r. That is,

$$cov(e, r) = \begin{cases} 1 & \text{if } e \text{ is covered by r} \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

Given a set of rules $r_1, \ldots, r_n$ the measure *recovered* for the rule r_{n+1} is defined by

Table 1 Illustrative dataset

Transaction	F_1	F_2	F_3
t_1	178	75	24
t_2	186	93	37
t_3	167	60	22
t_4	199	112	30
t_5	154	47	42
t_6	173	83	33
t_7	177	91	63
t_8	159	53	48
t_9	183	88	35
t_{10}	178	93	58

$$recov(r_{n+1}) = \frac{1}{N} \sum_{e \in D} \sum_{i=1}^{n} cov(e, r_i) \quad (9)$$

Thus, this index provides a measure of the number of instances which have already been covered by a set of previous rules.

3 Related work

A thorough review of recently published works reveals that the extraction of AR with numeric attributes is an emerging topic.

Mata et al. (2001) proposed a novel technique based on evolutionary techniques to find QAR that was improved in Mata et al. (2002). First, the approach found the sets of attributes which were frequently present in database and called *frequent itemsets*, and later AR were extracted from these sets.

Following with this topic, the optimization of the confidence—avoiding the initial threshold for the minimum support—was the main contribution of the work introduced in Yan et al. (2009). The authors used fitness function and non-threshold requirements for the minimum support. The fitness function is a key parameter in EA and the authors just used the relative confidence as fitness function.

Recently, Alcalá-Fdez et al. presented a study about three algorithms to analyze their effectiveness for mining QAR. In particular, EARMGA (Yan et al. 2009), GAR (Mata et al. 2002) and GENAR (Mata et al. 2001) were applied to two real-world datasets, showing their efficiency in terms of coverage and confidence.

On the other hand, data mining techniques for discovering AR in time series can be found in Bellazzi et al. (2005). The authors successfully mined temporal data retrieved from multiple hemodialysis sessions by applying preprocessing, data reduction and filtering as a previous step of the AR extraction process. Finally, AR were obtained by following the well-known Apriori itemset generation strategy (Venturini 1994).

An algorithm to discover frequent temporal patterns and temporal AR was introduced in Winarko and Roddick (2007). The algorithm extends the MEMISP algorithm (Lin and Lee 2002) which discovers sequential patterns by using a recursive find-then-index technique. Especially remarkable was the maximum gap time constraint included to remove insignificant patterns and consequently to reduce the number of temporal association rules.

Usually, the sliding window concept has been successfully applied to forecast time series (Martínez–Álvarez et al. 2011; Nikolaidou and Mitkas 2009). However, this concept has been recently used in (Khan et al. 2010) with the purpose of obtaining a low use of memory and low computational cost of the Apriori-based algorithm presented for discovering itemsets whose support increase over time.

In the non-supervised classification domain, the authors in Wan et al. (2007) made use of clustering processes to discretize the attributes of hydrological time series, as a first step of the rules extraction, which were eventually obtained by means of the Apriori algorithm. Following with clustering techniques, fuzzy clustering was used in Chen et al. (2010) to speed up the calculation for requirement satisfaction with multiple minimum supports, enhancing thus the results published in its initial work (Chen et al. 2009).

The work introduced in Huang et al. (2008) mined ocean data time series in order to discover relationship between salinity and temperature variations. Concretely, the authors discovered spatio-temporal patterns from the aforementioned variables and reported QAR using PrefixSpan and FITI algorithms (Pei et al. 2001; Tung et al. 2003).

Different models to forecast the ozone concentration levels have been recently proposed. Hence, the authors in Agirre-Basurko et al. (2006) developed two multilayer preceptron and a linear regression model for this purpose and prognosticated eight hours ahead for the Spanish city of Bilbao. They concluded that the insertion of extra seasonal variables may improve the general forecasting process. On the other hand, an artificial neural network model was presented in Elkamel et al. (2001). The authors also predicted the ozone concentrations by considering the analysis of additional climatological time series. Finally, temporal variations of the tropospheric ozone levels were analyzed in four sites of the Iberian Peninsula (Adame-Carnero et al. 2010) by means of statistical approaches.

Alternatively, the application of QAR can also be found in the data streams domain. In fact, the authors in Orriols-Puig et al. (2008) developed a model capable to classify on-line generated data for both continuous and discrete data streams.

MODENAR is a multi-objective pareto-based genetic algorithm that was presented in Alatas et al. (2008). In this approach, the fitness function aimed at optimizing four different variables: Support, confidence, comprehensibility of the rule and the amplitude of the intervals that constitutes the rule. A similar issue was addressed in Tong et al. (2005), in which the authors conducted research on the determination of existing conflicts when minimum support and minimum confidence are simultaneously required.

In Alatas and Akin (2008), the use of rough particle swarm techniques as an optimization metaheuristic was presented. In this work, the authors obtained the values for the intervals instead of frequent itemsets. Moreover, they proposed the use of some new operators such as rounding, repairing or filtrating.

Finally, QAR have also been used in the bioinformatics field. Thus, microarray data analysis by means of QAR was addressed in Georgii et al. (2005). The main novelty proposed by the authors was the definition of an AR as a linear combination of weighted variables, against a constant. Also in this context, the authors in Gupta et al. (2006) introduced a multi-step algorithm devoted to mine QAR for protein sequences. Once again, an Apriori-based methodology was used in Nam et al. (2009) to discover temporal associations from gene expression data.

4 Description of the search of rules

In a continuous domain, it is necessary to group certain sets of values that share same features and therefore it is required to express the membership of the values to each group. Adaptive intervals instead of fixed ranges have been chosen to represent the membership of such values in this work.

The search for the most appropriate intervals has been carried out by means of QARGA. Thus, the intervals are adjusted to find QAR with high values for support and confidence, together with other measures used in order to quantify the quality of the rule.

In the population, each individual constitutes a rule. These rules are then subjected to an evolutionary process, in which the mutation and crossover operators are applied and, at the end of the process, the individual that presents the best fitness is designated as the best rule. Moreover, the fitness function has been provided with a set of parameters so that the user can drive the process of search depending on the desired rules. The punishment of the covered instances allows the subsequent rules, found by QARGA, trying to cover those instances that were still uncovered, by means of an iterative rule learning (IRL) (Venturini 1993).

The following subsections detail the general scheme of the algorithm as well as the fitness function, the representation of the individuals, and the genetic operators.

4.1 Codification of the individuals

Each gene of an individual represents the upper and lower limit of the intervals of each attribute. The individuals are represented by an array of fixed length n, where n is the number of attributes belonging to the database. Furthermore, the elements are real-valued since the values of the attributes are continuous. Two structures are available for the representation of an individual:

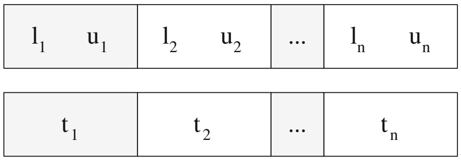
- Upper structure. All the attributes included in the database are depicted in this structure. The limits of the intervals of each attribute are stored, where l_i is the inferior limit of the interval and u_i the superior one.
- Lower structure. Nevertheless, not all the attributes will be present in the rules that describe an individual. This structure indicates the type of each attribute, t_i, which can have three different values:
  - 0 when the attribute does not belong to the individual,
  - 1 when the attribute belongs to the antecedent, and
  - 2 when it belongs to the consequent.

Figure 1 shows the codification of a general individual of the population.

With the proposed codification, if an attribute is wanted to be retrieved for a specific rule, it can be done by modifying the value equal to 0 of the type by a value equal to 1 or 2. Analogously, an attribute that appears in a rule may stop belonging to such rule by changing the type of the attribute from values 1 or 2 to 0. An illustrative example is depicted in Fig. 2. In particular, the rule $X_1 \in [20, 34] \wedge X_3 \in [7, 18] \Rightarrow X_4 \in [12, 27]$ is represented. Note that attributes X_1 and X_3 appear in the antecedent, X_4 in the consequent, and X_2 is not involved in the rule. Therefore, $t_1 = t_3 = 1, t_2 = 0$ and $t_4 = 2$.

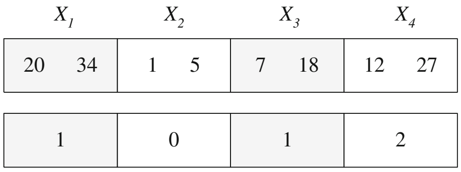
4.2 Generation of the initial population

The individuals of the initial population are randomly generated. In other words, the number of attributes appearing in the rule and the type and interval for each attribute are randomly generated. To assure that the individuals represent sound rules when the genes are generated, the following constraints are considered:



l_1 u_1	l_2 u_2	...	l_n u_n
t_1	t_2	...	t_n

Fig. 1 Representation of an individual of the population



X_1	X_2	X_3	X_4
20 34	1 5	7 18	12 27
1	0	1	2

Fig. 2 Example of an individual

- Limits of the interval:
  - The lower limit of the interval has to be less than the upper limit of the interval. If the randomly generated values do not fulfill this requirement, the limits are swapped.
  - The lower and upper limits of the interval have to be greater and less than the lower and upper limits of the domain of the attribute, respectively. Otherwise, the corresponding limits of the domain of the attribute are assigned.
- Type of the attribute:
  - The number of attributes of the rule has to be greater than a minimum number of attributes defined by the user depending on the desired rule.
  - The number of attributes belonging to the antecedent of the rule has to be greater than 1.
  - The number of attributes belonging to the consequent of the rule has to be greater than 1, and less than a maximum number of allowed consequents, which is a parameter defined by the user depending for the desired rule.

4.3 Genetic operators

This section describes the genetic operators used in the proposed algorithm, that is, selection, crossover and mutation operators.

1. *Selection*. An elitist strategy is used to replicate the individual with the best fitness. By contrast, a roulette selection method is used for the remaining individuals rewarding the best individuals according to the fitness. Note that the tournament selection was also used in preliminary studies, showing similar performance to that of roulette selection.
2. *Crossover*. Two parent individuals x and y, chosen by means of the roulette selection, are combined to generate a new individual z. Formally, let $[l_i^x, u_i^x], [l_i^y, u_i^y]$ and $[l_i^z, u_i^z]$ be the intervals in which the attribute a_i vary for the individuals x, y and z, respectively, and let t_i^x, t_i^y and t_i^z be the type of the attribute a_i for the individuals x, y and z, respectively. Then, for each attribute a_i two cases can occur:
   - $t_i^x = t_i^y$: The same type is assigned to the descendent and the interval is obtained by generating two random numbers among the limits of the intervals of both parents, as shown in Eqs. 10 and 11.

$$t_i^z = t_i^x \tag{10}$$

$$[l_i^z, u_i^z] = [random(l_i^x, l_i^y), random(u_i^x, u_i^y)] \tag{11}$$

- $t_i^x \neq t_i^y$: One of the two types is randomly chosen between both of the parents, without modifying the intervals of such attribute, as shown in Eqs. 12 and 13.

$$[l_i^z, u_i^z] = [l_i^x, u_i^x], \quad \text{if } t_i^z = t_i^x \tag{12}$$

$$[l_i^z, u_i^z] = [l_i^y, u_i^y], \quad \text{if } t_i^z = t_i^y \tag{13}$$

The limits and types of the attributes of the offspring are checked, as described in Sect. 4.2, to assure that it represents sound rules. If any attribute does not fulfill the required constraints regarding the type of attributes the individual is discarded and a new individual is obtained from the same parents. The crossover process is depicted in Fig. 3.

3. *Mutation*. The mutation process consists in modifying according to a probability the genes of randomly selected individuals. The mutation of a gene can be focused on

   - *Type of the attribute*. Two equally probable cases can be distinguished:
     - Null Mutation. The type t_i of the selected attribute is different to null, and eventually changed to null.
     - Not Null Mutation. The type t_i of the selected attribute is null and changed to antecedent or consequent.
   - *Intervals of the attribute*. Three equally probable cases are possible:
     - *Lower Limit*. A random value is added or subtracted to the lower limit of the interval.
     - *Upper Limit*. A random value is added or subtracted to the upper limit of the interval.
     - *Both Limits*. A random value is added or subtracted to both limits of the interval.

       For all the three cases, the random value is generated between 0 and a percentage (usually 10%) of the amplitude of the interval and it will be added or subtracted according to a certain probability.

The choice between the mutation of the type or the mutation of the interval depends on a given probability.

The limits and types of the attributes of the offspring are checked, as described in Sect. 4.2, to assure that it represents sound rules. If any attribute does not fulfill the required constraints regarding of the type of attributes, the individual is discarded and a new mutation is obtained from the same original individual.

Some examples of all the kind of mutations are illustrated in Figs. 4, 5, 6, 7 and 8.

4.4 The fitness function

The fitness of each individual allows deciding which are the best candidates to remain in subsequent generations. In order to make this decision, it is desirable that the support would be high, since this fact implies that more samples from the database are covered. Nevertheless, to take into consideration only the support is not enough to calculate the fitness because the algorithm would try to enlarge the amplitude of the intervals until the whole domain of each attribute would be completed. For this reason, it is

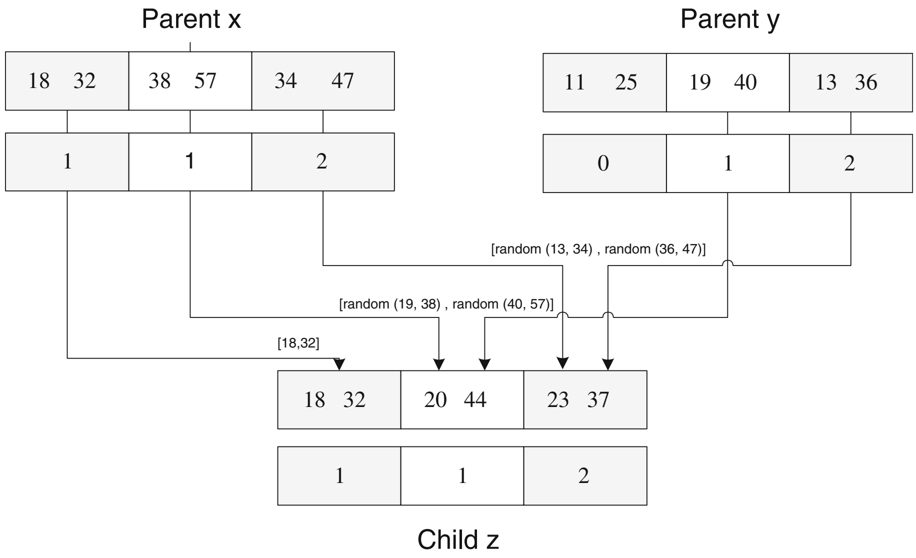


Fig. 3 Crossover for the individuals *x* and *y*

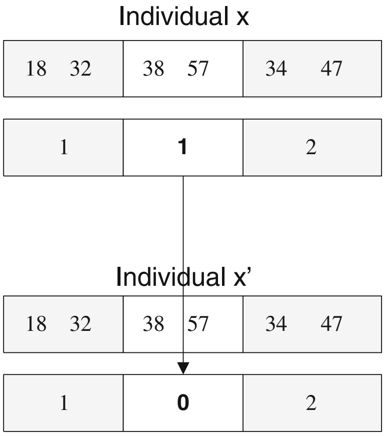


Fig. 4 Scheme of Null Mutation

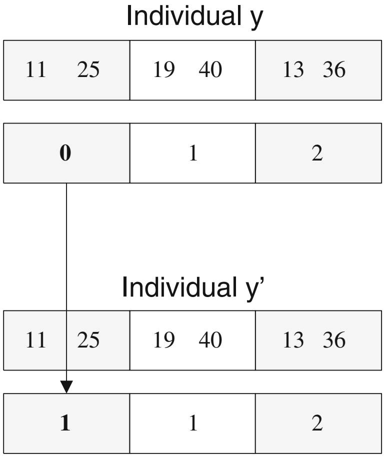


Fig. 5 Scheme of Not Null Mutation

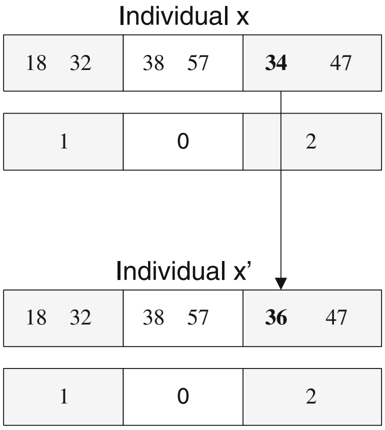


Fig. 6 Scheme of Lower Limit Mutation

necessary to include a measure to limit the growth of the intervals during the evolutionary process. The chosen fitness function to be maximized is

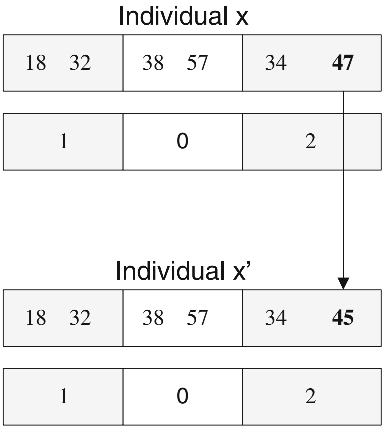


Fig. 7 Scheme of Upper Limit Mutation

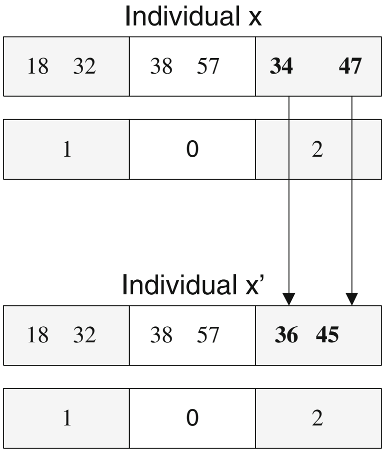


Fig. 8 Scheme of Lower and Upper Limits Mutation

$$f = w_s \cdot sup + w_c \cdot conf - w_r \cdot recov + w_n \cdot nAttrib - w_a \cdot ampl \tag{14}$$

where *sup*, *conf* and *recov* are defined in Sect. 2.2, *nAttrib* is the number of attributes appearing in the rule, *ampl* is the average size of intervals of the attributes that compose the rule, and w_s, w_c, w_r, w_n and w_a are the weights to drive the search, and will vary depending on the required rules.

The support rewards the rules fulfilled by many instances and the weight w_s can increase or decrease its effect.

The confidence together with the support are the most widely measures used to evaluate the quality of the QAR. The confidence is the grade of reliability of the rule. High values of w_c may be used when rules without error are desired, and viceversa.

The number of recovered instances is used to indicate that a sample has already been covered by a previous rule. Thus, rules covering different regions of the search space are preferred. The process of punishing the covered instances is now described. Every time the evolutionary process ends and the best individual is chosen as the best rule, the database is processed in order to find those instances already covered by the rule. Hence, each instance has a counter that increases its value by one every time a rule covers it.

The number of attributes of a rule can be adjusted by means of the weight w_n. Thus, when w_n is set to a value close to 0, few attributes are obtained and, on the other hand, when w_n is set to a value close to 1, many attributes appear in rules.

The amplitude controls the size of the intervals of the attributes that compose the rules and those individuals with large intervals are penalized by means of the factor w_a, which allows the rules be more or less permissive regarding the amplitude of the intervals.

Hence, the user can model the behavior of the rules that can be obtained by varying the weights in the fitness function. Therefore, the user can obtain rules according to their needs without a previous data discretization.

4.5 The IRL approach

The proposed algorithm is based on the iterative rule learning (IRL) process, whose general scheme is shown in Fig. 9.

The EA is applied in each iteration obtaining one rule per iteration, which is precisely the best individual discovered. While the number of desired rules is not reached, IRL allows penalization in already covered instances, with the aim of finding rules that cover those instances that have not been covered yet in subsequent iterations. The main advantage of the approach is that attempts at covering every region in the solutions domain, that is, the set of rules will cover all the consequent domain. The iterative process ends when it finds the desired number of rules.

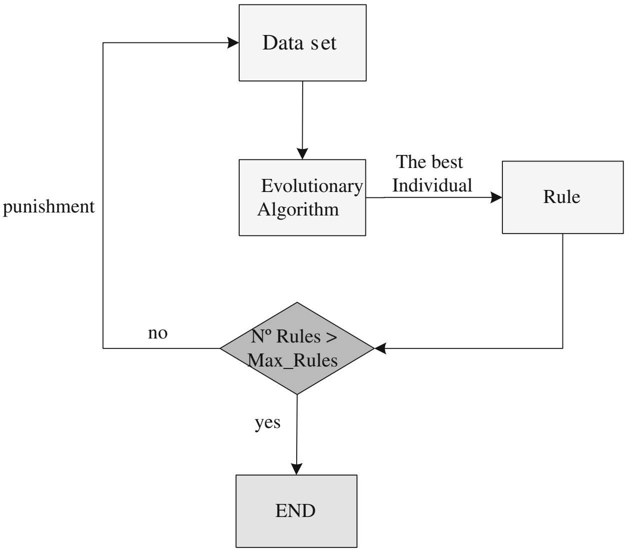


Fig. 9 Scheme of IRL

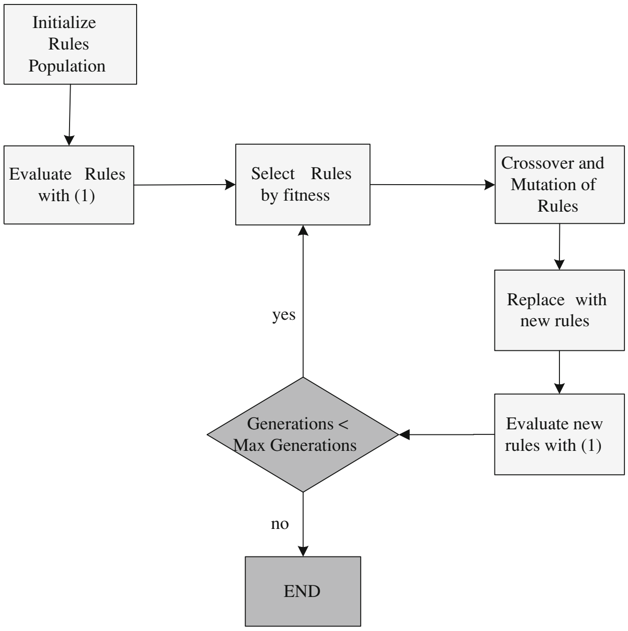

Initialize Rules Population
Evaluate Rules with (1)
Select Rules by fitness
Crossover and Mutation of Rules
Replace with new rules
Evaluate new rules with (1)
Generations < Max Generations
yes
no
END

Fig. 10 EA scheme

Figure 10 provides the scheme of the EA, which is the main step of the IRL process depicted in Fig. 9.

First, the rules population is initialized and evaluated. All rules are evaluated according to (14). Thus, in each iteration the selection operator is applied to select the best rules on the basis of the fitness function. Then, the crossover operator is applied to the selected rules while the population size is not completed. Individuals are randomly selected in order to apply the mutation operator. Finally, the new population is again evaluated by the fitness function and the evolutionary process restarts. Note that the process will be repeated as many times as the maximum number of preset generations indicates.

5 Results

In this section the results obtained from the application of the proposed approach to different datasets are presented. First, Sect. 5.1 provides a detailed description of all used datasets. A summary of the key parameters configuration used for all the algorithms can be found in Sect. 5.2. Finally, the results are gathered and discussed in Sect. 5.3.

The approach has been initially tested on several widely studied datasets from the public BUFA repository, and the accuracy of QARGA has been compared with that of the algorithms introduced in Yan et al. (2009) and Mata et al. (2001) in Sect. 5.3.1. On the other hand, two different kind of time series are analyzed: synthetically generated and real-world multidimensional temporal data. Sect. 5.3.2 is devoted to evaluate the accuracy of QARGA when it is applied to synthetically generated multidimensional time series. Likewise, the real-world case is reported in Sect. 5.3.3, where QAR are obtained to discover relationships between the tropospheric ozone and other climatological time series.

5.1 Dataset description

This section presents the number of records and number of attributes of the BUFA repository datasets as well as how synthetic time series were generated and what real-world time series consist of.

5.1.1 Public datasets

QARGA has been applied to 15 public datasets: Basketball, Bodyfat, Bolts, Kinematics, Longley, Normal Body Temperature, Plastic, Pollution, Pw Linear, Pyramidines, Quake, Schools, Sleep, Stock Price and Vineyard, which can be found at BUFA repository (Guvenir and Uysal 2000). Relevant information about these datasets is summarized in Table 2.

5.1.2 Synthetic multidimensional time series

In this section two different synthetically generated multidimensional time series are described: time series without and with disjunctions, respectively. In particular, multidimensional time series are generated, that is, time series characterized by more than one variable in each time stamp. Or, in other words, two or more time series simultaneously observed that characterize the same phenomenon. Formally, a multidimensional time series *MTS* can be expressed as $MTS = [X_1(t), \ldots, X_n(t)]^T$, where each $X_i(t)$ is a variable measured along with the time, t, and n is the number of inter-related time series that identifies the whole *MTS*. Thus, the goal of applying QAR to *MTS* is to discover existing relationships among those X_i forming the *MTS*, along with the time.

Regarding the time series with no disjunctions, Table 3 defines a three-dimensional time series, $n = 3$, in which three variables X_1, X_2 and X_3 share static relationships in fixed intervals of time.

Thus, 100 values for each variable X_1, X_2 and X_3 were generated and uniformly distributed in four intervals. To obtain these series, values for variables X_1, X_2 and X_3 were randomly selected for every t_i according to constraints listed in Table 3, where t_i varies from 1 to 100. Finally, the resulting time series are depicted in Fig. 11.

With reference to time series with disjunctions, a bi-dimensional time series represented by two variables, X_1 and X_2, has been generated with, again, 100 values for each time series uniformly distributed in four intervals. However, the main difference regarding the previous situation lies in the fact that now the variables X_1 and X_2 can be defined by more than one possible set of values.

Table 4 shows the constraints considered to generate the time series with disjunctions. The series is generated then as follows: For every t_i and X_j one interval is randomly chosen and, then, a value is randomly chosen from the interval previously selected. For instance, X_1 can indistinctively belong to intervals $[10, 20]$ or $[15, 35]$ when $t \in [26, 50]$ in set #1.

Table 2 Public datasets.

Dataset	Records	Attributes
Basketball	96	5
Bodyfat	252	18
Bolts	40	8
Kinematics	8192	9
Longley	16	7
Normal Body Temperature	130	3
Plastic	1650	3
Pw Linear	200	11
Pollution	60	16
Pyramidines	74	28
Quake	2178	4
School	62	20
Sleep	57	8
Stock price	950	10
Vineyard	52	4

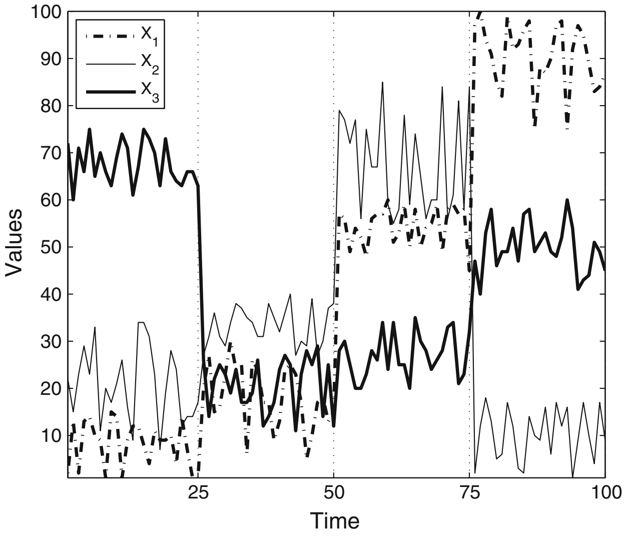


Fig. 11 Time series with no disjunctions

Table 3 Time series with no disjunctions

ID	Sets	Sup. (%)
#0	$t \in [1, 25] \Longrightarrow X_1 \in [1,\ 15] \wedge X_2 \in [7,\ 35] \wedge X_3 \in [60,\ 75]$	25.0
#1	$t \in [26, 50] \Longrightarrow X_1 \in [5,\ 30] \wedge X_2 \in [25,\ 40] \wedge X_3 \in [10,\ 30]$	25.0
#2	$t \in [51, 75] \Longrightarrow X_1 \in [45,\ 60] \wedge X_2 \in [55,\ 85] \wedge X_3 \in [20,\ 35]$	25.0
#3	$t \in [76, 100] \Longrightarrow X_1 \in [75,\ 100] \wedge X_2 \in [0,\ 20] \wedge X_3 \in [40,\ 60]$	25.0

Table 4 Time series with disjunctions

ID	Sets
#0	$t \in [1,\ 25] \Longrightarrow X_1 \in [20,\ 30] \wedge X_2 \in [50,\ 80]$
#1	$t \in [26,\ 50] \Longrightarrow (X_1 \in [10,\ 20] \vee X_1 \in [15,\ 35]) \wedge X_2 \in [40,\ 60]$
#2	$t \in [51,\ 75] \Longrightarrow X_1 \in [1,\ 15] \wedge (X_2 \in [40,\ 50] \vee X_2 \in [60,\ 70])$
#3	$t \in [76,\ 100] \Longrightarrow (X_1 \in [15,\ 25] \vee X_1 \in [30,\ 40]) \wedge (X_2 \in [30,\ 45] \vee X_2 \in [40,50])$

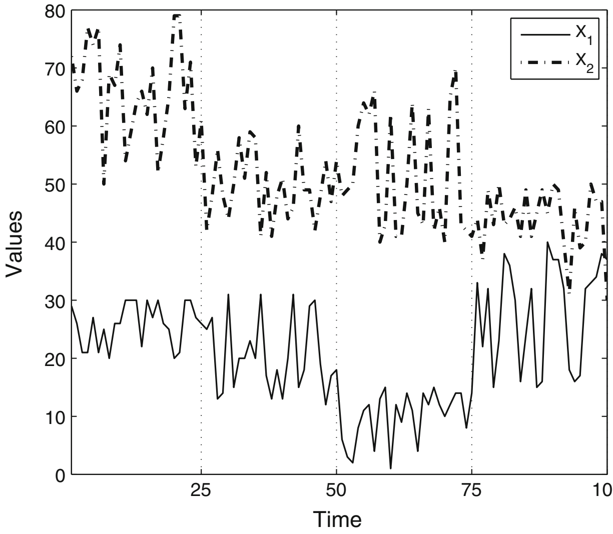


Fig. 12 Time series with disjunctions

Table 5 Expected rules from time series with disjunctions

ID	Sets	Sup. (%)
$\#0$	$t \in [1,\ 25] \wedge X_1 \in [20,\ 30] \wedge X_2 \in [50,\ 80]$	25.0
$\#1_1$	$t \in [26,\ 50] \wedge X_1 \in [10,\ 20] \wedge X_2 \in [40,\ 60]$	12.5
$\#1_2$	$t \in [26,\ 50] \wedge X_1 \in [15,\ 35] \wedge X_2 \in [40,\ 60]$	12.5
$\#2_1$	$t \in [51,\ 75] \wedge X_1 \in [1,\ 15] \wedge X_2 \in [40,\ 50]$	12.5
$\#2_2$	$t \in [51,\ 75] \wedge X_1 \in [1,\ 15] \wedge X_2 \in [60,\ 70]$	12.5
$\#3_1$	$t \in [76,\ 100] \wedge X_1 \in [15,\ 25] \wedge X_2 \in [30,\ 45]$	6.25
$\#3_2$	$t \in [76,\ 100] \wedge X_1 \in [15,\ 25] \wedge X_2 \in [40,\ 50]$	6.25
$\#3_3$	$t \in [76,\ 100] \wedge X_1 \in [30,\ 40] \wedge X_2 \in [30,\ 45]$	6.25
$\#3_4$	$t \in [76,\ 100] \wedge X_1 \in [30,\ 40] \wedge X_2 \in [40,\ 50]$	6.25

The resulting *MTS* according with Table 4 is illustrated in Fig. 12. As it can be observed, relationships between time and variables are considerably more difficult to be mined. In fact, the expected rules are listed in Table 5, where every disjunction in the interval for each variable involve two possible conjunctions. For instance, the set #0 in Table 4 would solely generate rule #0 from Table 5. Nevertheless, set #1 would diverge into two different expect rules, $\#1_1$ and $\#1_2$ from Table 5, and so on. Note that the support in Table 5 for all possible rules is, actually, the expected support assuming that the portion of values of a variable for every disjunction were equal.

5.1.3 *Real-world time series application: ozone concentration*

The proposed algorithm has also been applied in order to discover QAR in real-world multidimensional time series. Specifically, QAR are intended to be found among climatological time series such as temperature, humidity, direction and speed of the wind, several temporal variables such as the hour of the day and the day of the week and, finally, the tropospheric ozone. These variables have influence on the ozone concentration in the atmosphere which is the target agent.

All variables have been retrieved from the meteorological station of the city of Seville in Spain for the months from July to August during years 2003 and 2004, generating a dataset with 1488 instances. The reason for selecting such periods is because during these periods the highest concentration of ozone was reported.

For predictive purposes, the climatological time series have been forced to belong to the antecedent and the ozone to the consequent. As a result, a prediction of the ozone is achieved on the basis of the rules extracted from these variables.

5.2 Parameters configuration

In this section, the values for the parameters of each method analyzed in Sect. 5.3 are described. This section is divided in subsections for all used datasets. It is noteworthy that the parameters of every method with which QARGA is compared were obtained from the original papers.

5.2.1 *Configuration for public datasets*

1. *EARMGA*. This algorithm (Yan et al. 2009) was executed five times and the average values of such executions were presented. The main parameters of EARMGA algorithm are 100 for the number of the rules, 100 for the size of the population and 100 for the number of generations. EARMGA use 0.0 for the minimum support and minimum confidence; 0.75 for the probability of selection; 0.7 for the probability of crossover, 0.1 for the probability of mutation; 0.01 for difference boundary and 4 for the number of partitions for numeric attributes.
2. *GENAR*. This algorithm was executed five times and the average values of such executions were presented. The main parameters of GENAR algorithm (Mata et al. 2001) are 100 for the number of the rules, 100 for the size of population and 100 for the number of generations. GENAR use 0.0 for the minimum support and minimum confidence; 0.25 for the probability of selection; 0.7 for the probability of crossover and 0.1 for the probability of mutation; 0.7 for the penalization factor and 2 for the amplitude factor.
3. *QARGA*. It has been executed five times, and the average results are also shown for this case. The main parameters of QARGA are 100 for the number of the rules, 100 for the size of the population, 100 for the number of generations, 0.0 for the minimum support and minimum confidence and 0.8 for the probability of mutation.

5.2.2 *Configuration for synthetic time series with no disjunctions*

1. *QARGA*. The proposed algorithm has been executed five times. The main parameters are as follows: 100 for the size of the population, 100 for the number of generations, 20 for the number of rules to be obtained, and 0.8 for the mutation probability. After an experimental study to assess the influence of the weights on the rules to be obtained, the weights chosen for the fitness function were 3 for w_s, 1 for w_c, 2 for w_r, 0.2 for w_n and 0.5 for w_a.

5.2.3 *Configuration for synthetic time series with disjunctions*

1. *QARGA*. The main parameters for these time series are exactly the same that those used to generate rules for synthetic time series with no disjunctions. That is, 100 for the size of the population, 100 for the number of generations, 20 for the number of rules to be obtained, and 0.8 for the mutation probability. In this case, the weights chosen for the fitness function were 1.5 for w_s, 0.5 for w_c, 0.2 for w_r, 0.2 for w_n and 0.3 for w_a.

5.2.4 *Configuration for ozone time series*

For this time series, QARGA has been compared with Apriori and due to its previous required discretization, two different kind of experimentation are distinguished. First, all the continuous variables have been discretized with three intervals. Thus, the obtained rules by Apriori present high amplitudes and therefore high supports. Second, all the real-valued attributes have been discretized in ten intervals, which involves rules with small amplitudes and low supports. For both of the experimentations, the selected rules are the ones that presented greater confidence.

For the first kind of experimentation, the main parameters of QARGA have been set as follows: 100 for the size of the population, 100 for the number of generations, 20 for the number of rules to be obtained and 0.8 for the mutation probability. After an empirical study to test the influence of the weights on the rules to be obtained, the weights of the fitness function, 3 for w_s, 0.2 for w_c, 0.2 for w_r, 0.3 for w_n and 0.2 for w_a have been chosen. This study consisted in determining the values of the weights for which the confidence of the rules was maximized. Note that w_s is high compared to the other ones because rules with high support are desired, making thus possible the comparison with the rules obtained by Apriori.

For the second kind of experimentation, the main parameters of QARGA have been set as follows: 100 for the size of the population, 100 for the number of generations, 20 for the number of rules to be obtained and 0.8 for the mutation probability. After an empirical study to test the influence of the weights on the rules to be obtained, the weights of the fitness function, 1 for w_s, 0.2 for w_c, 0.2 for w_r, 1 for w_n and 0.2 for w_a have been chosen. Analogous to the first experimentation, the weight associated with support is low to make possible the comparison with the Apriori algorithm.

5.3 Analysis of results

This section discusses all the results obtained from the application of QARGA to the selected datasets introduced in previous sections.

5.3.1 *Results in public datasets*

To carry out the experimentation and make a comparison with QARGA, the evolutionary algorithms EARMGA

(Yan et al. 2009) and GENAR (Mata et al. 2001), available in the KEEL tool (Alcalá-Fdez et al. 2009b), have been chosen.

Table 6 shows the results obtained by EARMGA, GENAR and QARGA for every dataset. The column *Number of rules* indicates the average of the number of the rules found by each algorithm after executions specified in Sect. 5.2. The percentage of the records covered by the rules for these datasets is shown in column *Records*.

It can be noticed that the average of the number of rules and the average of the percentage of the records covered by the rules founds by QARGA are greater than the rest of the algorithms.

Table 6 Number of rules and percentages of records covered by the mined rules obtained by QARGA and all other algorithms

Dataset	Number of rules			Records (%)		
	EARMGA	GENAR	QARGA	EARMGA	GENAR	QARGA
Basketball	100	100	100	74.16	91.04	99.48
Bodyfat	100	100	100	46.67	69.44	100
Bolts	100	100	100	57.50	51.00	100
Kinematics	100	100	100	42.33	38.84	89.17
Longley	100	11.70	100	50.00	100	100
Normal Body Temperature	100	100	100	100	97.08	98.00
Plastic	96.40	100	100	100	99.44	99.67
Pw Linear	100	100	100	53.00	21.40	98.00
Pollution	100	100	100	42.67	52.33	96.67
Pyramidines	100	82.15	100	43.78	100	100
Quake	100	100	100	97.41	82.12	91.85
School	100	100	100	55.08	85.57	100
Sleep	100	100	100	67.84	88.24	100
Stock price	100	100	100	59.37	87.98	100
Vineyar	100	100	100	93.85	94.62	100
	99.76	92.92	100	65.58	77.27	98.19

Table 7 Quality measurements of rules obtained by QARGA and all other algorithms.

Dataset	Support (%)			Confidence (%)		
	EARMGA	GENAR	QARGA	EARMGA	GENAR	QARGA
Basketball	2.70	30.82	33.52	100	96.52	97.43
Bodyfat	4.97	41.52	74.06	100	96.53	98.17
Bolts	11.43	14.43	11.17	100	100	99.76
Kinematics	2.09	0.53	22.02	100	96.50	83.89
Longley	12.69	24.37	36.68	100	100	100
Normal Body Temperature	22.29	64.27	7.70	100	72.89	97.55
Plastic	10.06	24.43	8.51	97.76	55.17	98.56
Pw Linear	3.68	1.09	16.47	100	100	98.50
Pollution	5.36	22.64	49.83	100	99.72	99.90
Pyramidines	7.84	2.19	11.34	38.49	100	99.82
Quake	3.40	35.17	7.73	100	64.40	94.66
School	7.21	8.13	41.73	100	100	99.23
Sleep	9.05	36.69	49.71	100	71.17	99.68
Stock price	4.05	30.71	32.71	100	91.49	98.93
Vineyar	7.80	43.75	39.35	100	98.60	99.31
	7.64	25.38	29.50	95.75	89.53	97.69

Table 7 shows some quality measurements of rules obtained by every algorithm. The first column, *support* (%), reports the average support obtained, that is, the percentage of covered instances. The next column, *confidence* (%), shows the average confidence obtained by every algorithm.

Concentrating on the results themselves, it can be appreciated that the average support found by QARGA is greater than that found by the other algorithms in almost all the datasets. The average confidence obtained by QARGA is greater than that of EARMGA and GENAR, that is, QARGA provides more reliable rules with smaller errors.

Table 8 and Table 9 show the average number of attributes and the average amplitude for both the antecedent and the consequent for the rules extracted by EARMGA, GENAR and QARGA. From its observation it can be

Table 8 Amplitudes for the antecedents, consequents and rules obtained by QARGA and all other algorithms

Dataset	Antecedent amplitude (%)			Consequent amplitude (%)			Rule amplitude (%)		
	EARMGA	GENAR	QARGA	EARMGA	GENAR	QARGA	EARMGA	GENAR	QARGA
Basketball	40.37	49.55	26.84	69.71	49.85	32.35	46.55	49.61	29.22
Bodyfat	39.19	47.20	28.80	81.57	49.99	26.86	48.65	47.35	27.80
Bolts	41.75	38.49	11.69	67.67	31.85	9.49	48.67	37.45	10.57
Kinematics	39.23	49.80	29.54	100	49.61	30.65	51.80	49.78	30.03
Longley	44.85	42.31	26.85	57.20	43.90	22.09	48.70	42.53	24.44
Normal Body Temperature	45.23	74.90	12.60	89.90	50	16.21	60.13	66.60	13.92
Plastic	32.06	49.17	15.89	81.96	46.90	25.78	48.70	48.41	19.43
Pw Linear	32.63	47.84	18.81	89.40	41	13.57	44.36	47.22	15.51
Pollution	40	45.73	14.30	53.38	49.82	13.90	43.18	45.98	14.15
Pyramidines	35.32	32.16	8.09	46.29	45.60	8	100	32.56	8.05
Quake	35.59	49.90	10.75	94.55	48.63	11.28	50.59	49.65	10.88
School	41.36	40.92	20.77	80.90	46.13	19.86	50.28	41.18	20.35
Sleep	43.84	41.92	10.51	79.13	49.89	10.33	51.74	42.91	10.41
Stock price	38.45	48.07	29.89	96.18	48.56	30.21	50.39	48.12	30.06
Vineyar	39.57	48.97	29.86	72.30	45.43	29.87	47.88	48.08	29.82
	39.30	47.13	19.68	77.34	46.48	20.03	52.77	46.50	19.64

Table 9 Size of the antecedents, consequents and rules obtained by QARGA and all other algorithms

Dataset	Antecedent size			Consequent size			Rule size		
	EARMGA	GENAR	QARGA	EARMGA	GENAR	QARGA	EARMGA	GENAR	QARGA
Basketball	3.96	4	1.37	1.04	1	1.01	5	5	2.38
Bodyfat	3.83	17	1.26	1.17	1	1.05	5	18	2.31
Bolts	3.60	7	4.31	1.40	1	2.15	5	8	6.46
Kinematics	3.97	8	1.96	1.03	1	1.03	5	9	2.99
Longley	3.58	6	1.08	1.42	1	1.08	5	7	2.15
Normal Body Temperature	2.00	2	1.87	1.00	1	1.04	3	3	2.92
Plastic	1.99	2	1.86	1.01	1	1.00	3	3	2.86
Pw Linear	3.97	10	1.65	1.03	1	1.01	5	11	2.65
Pollution	3.81	15	6.13	1.19	1	1.42	5	16	7.54
Pyramidines	3.65	27	12.51	1.35	1	2.62	5	28	15.13
Quake	2.97	3	2.79	1.03	1	1.02	4	4	3.81
School	3.90	19	1.37	1.10	1	1.13	5	20	2.50
Sleep	3.72	7	1.50	1.28	1	1.38	5	8	2.88
Stock price	3.96	9	1.48	1.04	1	1.01	5	10	2.49
Vineyar	2.98	3	1.09	1.02	1	1.03	4	4	2.11
	3.46	9.27	2.82	1.14	1.00	1.26	4.60	10.27	4.08

concluded that QARGA mined rules with short antecedent and short consequent, which helps to the comprehensiveness of the rules. The number of attributes per rule obtained by QARGA is similar to that of EARMGA and does not present relevant differences. For the case of the amplitude, QARGA obtained amplitudes smaller than EARMGA and GENAR in all datasets.

In short, QARGA presents greater average support, less number of attributes and smaller amplitudes than the other ones, which leads to the conclusion that QARGA obtained better rules in general terms.

From the reported results, it can be seen that rules with high support and confidence as well as moderate amplitude of intervals with small number of attributes have been found. In terms of support, confidence and amplitude, QARGA outperforms EARMGA and GENAR which leads to the obtention of more precise as well as comprehensible rules, since the number of attributes that appear in both antecedent and consequent is small, helping the user to easily understand them.

Last, a statistical analysis has been conducted to evaluate the significance of QARGA, following the non-parametric procedures discussed in García et al. (2009). For this purpose, the lift obtained from the application of QARGA, EARMGA and GENAR to the 15 datasets has been calculated, and it is shown in Table 10. From this table, it can be noticed that the algorithm QARGA reaches the highest rank in 15 datasets, EARMGA reaches the second and third positions in 10 and 5 datasets, respectively, and finally, GENAR obtains the second and third positions in 5 and 10 datasets. The average ranking for each algorithm is summarized in Table 11. It can be observed that the lowest value of average ranking is obtained by QARGA which is, therefore, the control algorithm.

Table 10 Lift of the mined rules by QARGA and all other algorithms

Dataset	Lift		
	EARMGA	GENAR	QARGA
Basketball	1.34	1.09	2.01
Bodyfat	1.77	1.11	4.49
Bolts	1.73	1.60	9.10
Kinematics	1.00	1.38	5.77
Longley	2.79	2.46	2.82
Normal Body Temperature	1.02	0.99	2.78
Plastic	1.26	1.10	3.44
Pw Linear	1.00	1.57	2.04
Pollution	3.00	1.23	6.86
Pyramidines	2.23	2.24	6.78
Quake	1.01	1.00	2.03
School	1.24	1.62	3.50
Sleep	1.93	1.15	9.48
Stock price	1.08	1.64	2.49
Vineyar	1.29	1.27	2.52
	1.58	1.43	4.41

Table 11 Average rankings of the algorithms.

Algorithm	Ranking
GENAR	2.66
EMARGA	2.33
QARGA	1.00

Friedman and Iman-Davenport (ID) tests have been applied to assess if there are global differences in the lifts obtained for three algorithms. The results obtained by both tests for the level of significance $\alpha = 0.05$ are summarized in Table 12. Note that the values in columns *Value in* χ^2 and *Value in* F_F have been retrieved from Tables A4 and A10 in Sheskin (2006), respectively. As the p values obtained from both of the tests are lower than the level of significance considered, it can be stated that there exist significant differences among the results obtained by three algorithms and a post-hoc statistical analysis is required.

The Holm and Hochberg tests have been applied to compare separately QARGA to GENAR and EMARGA. Table 13 shows the sorted p values obtained by GENAR and EMARGA for two levels of significance ($\alpha = 0.05$ and $\alpha = 0.10$). Both of the tests allow concluding that QARGA is better than EMARGA and GENAR for both levels of significance, as the two tests reject all hypotheses.

In addition, it is interesting to discover the precise p value for which each hypothesis can be rejected. These exact values are called adjusted p values and how to obtain them is thoroughly described in Wright (1992). Table 14

Table 12 Results of the Friedman and Iman-Davenport tests with $\alpha = 0.05$

Lift					
Friedman	Value in χ^2	p	ID	Value in F_F	p
23.33	5.99	8.57×10^{-6}	48.99	3.34	7.16×10^{-10}

Table 13 Holm and Hochberg tests results with QARGA as control algorithm

i	Algorithm	z	p	α/i ($\alpha = 0.05$)	α/i ($\alpha = 0.10$)
2	GENAR	4.56	5.01×10^{-6}	0.025	0.05
1	EMARGA	3.65	2.61×10^{-4}	0.05	0.10

Table 14 Adjusted p values when QARGA is compared to the remaining algorithms

Algorithm	Unadjusted p	p_{BD}	p_{Holm}	p_{Hoch}
GENAR	5.01×10^{-6}	1.00×10^{-5}	1.00×10^{-5}	1.00×10^{-5}
EMARGA	2.61×10^{-4}	5.21×10^{-4}	2.61×10^{-4}	2.61×10^{-4}

shows the adjusted p values for Bonferroni-Dunn (BD), Holm and Hochberg tests. It can be appreciated that the Holm and Hochberg tests show that QARGA is significantly better than the others with the lowest confidence level compared to the remaining tests ($\alpha = 2.61 \times 10^{-4}$). Again, the three tests coincide in rejecting all hypotheses for levels of significance $\alpha = 0.05$ and $\alpha = 0.10$, determining that QARGA is the best algorithm.

5.3.2 Results in synthetic time series

Once compared QARGA with other EA in public datasets—that were static and non-temporal-dependent—the algorithm is assessed when applied to time series. For this reason, two different types of synthetic time series were generated, as described in Sect. 5.1.2.

Table 15 shows the rules obtained by an execution of QARGA when multidimensional synthetic time series with no disjunctions (see Table 3 for detailed data description) were analyzed. Similar rules have been obtained by other executions of QARGA.

From the ten discovered rules, the four first ones (rules #0 to #3) highlight and are considered especially meaningful insofar as they represent, exactly, the intervals used in Table 3 to generate the time series itself. That is, QARGA was able to precisely discover the rules that model the synthetic time series generation.

It can also be observed that the support (*Sup*. column) in these rules is 25%, which coincides with the preset support when the time series were generated. Equally remarkable is that the confidence (*Conf*. column) is 100% for all the four rules. It is also noteworthy the precision of the intervals found since most of the limits discovered by QARGA coincide with those of the Table 3, which means a great level of reliability in the rules. Finally, note that the lift is much greater than one, in other words, such antecedents and consequents are likely to appear together.

As for the six remaining rules (rules #4 to #9), they correspond to rules with smaller support, confidence and lift. This fact can be justified by taking into consideration that when an IRL algorithm is applied, the search space is constantly being decreased and, therefore, the obtained rules cover less samples with less precision.

On the other hand, Table 16 shows the 12 most relevant rules obtained by QARGA when synthetic time series with disjunctions (see Table 4 for detailed data description) were analyzed. To facilitate the analysis, they are listed according to the interval to which the time variable –t belongs to, as listed in Table 5.

In general terms, each rule in Table 16 represents one of the expected rules listed in Table 5, except for some cases, which are discussed now. Thus, rule #0 in Table 16 represents the expected rule #0 in Table 5. The support is 19%, a value very close to the expected one. In addition, this rule has a 100% confidence.

When the time is in the interval $[26, 50]$, there were two possible expected conjunctions, $\#1_1$ and $\#1_2$. In this case, rule #1 approximately represents $\#1_1$ and rule #2 does $\#1_2$. Regarding the support, it is not significantly different from the expected one. Finally, the confidence is nearly 100% for both of the rules.

Rules #4 and #5 identify rules with $t \in [54, 75]$ and correspond to conjunctions $\#2_1$ and $\#2_2$, respectively. Again, the support is not very different from the expected one and the confidence is 100% for all of them.

Four different conjunctions were expected—$\#3_1$, $\#3_2$, $\#3_3$ and $\#3_4$—when $t \in [76, 100]$. Rule #8 identifies

Table 15 Rules found by QARGA for time series with no disjunctions

ID	Rules	Conf. (%)	Lift	Sup. (%)	Amp. (%)
#0	$X_1 \in [1, 15] \wedge X_3 \in [60, 75] \Longrightarrow t \in [1, 25] \wedge X_2 \in [7, 34]$	100	4.0	25.0	20.0
#1	$t \in [26, 50] \wedge X_1 \in [5, 30] \Longrightarrow X_2 \in [27, 40] \wedge X_3 \in [11, 29]$	100	4.0	25.0	20.0
#2	$X_1 \in [45, 60] \wedge X_2 \in [55, 85] \wedge X_3 \in [20, 35] \Longrightarrow t \in [51, 75]$	100	4.0	25.0	21.0
#3	$t \in [76, 100] \wedge X_1 \in [75, 100] \wedge X_3 \in [40, 60] \Longrightarrow X_2 \in [1, 18]$	100	2.9	25.0	21.5
#4	$t \in [1, 11] \wedge X_1 \in [1, 14] \wedge X_2 \in [20, 29] \Longrightarrow X_3 \in [66, 75]$	100	5.9	7.0	10.3
#5	$t \in [82, 92] \wedge X_2 \in [8, 17] \Longrightarrow X_1 \in [82, 92] \wedge X_3 \in [49, 55]$	50.0	7.1	3.0	8.7
#6	$t \in [48, 58] \wedge X_1 \in [47, 57] \Longrightarrow X_2 \in [72, 81] \wedge X_3 \in [20, 30]$	50.0	10.0	4.0	9.6
#7	$X_1 \in [1, 12] \wedge X_2 \in [24, 34] \wedge X_3 \in [64, 75] \Longrightarrow t \in [11, 21]$	85.7	7.8	6.0	10.5
#8	$X_2 \in [3, 17] \wedge X_3 \in [41, 49] \Longrightarrow t \in [82, 100] \wedge X_1 \in [84, 99]$	75.0	4.7	9.0	13.8
#9	$X_1 \in [53, 62] \Longrightarrow t \in [68, 78] \wedge X_2 \in [53, 61] \wedge X_3 \in [20, 26]$	17.6	5.9	3.0	8.7

Table 16 Rules found by QARGA for time series with disjunctions

ID	Rules	Conf. (%)	Lift	Sup. (%)	Amp. (%)
#0	$X_1 \in [20, 30] \wedge X_2 \in [61, 79] \Longrightarrow t \in [1, 25]$	100	4.00	19	17.33
#1	$t \in [28, 50] \wedge X_2 \in [47, 58] \Longrightarrow X_1 \in [12, 20]$	93.3	2.39	14	13.67
#2	$t \in [26, 46] \wedge X_1 \in [25, 31] \Longrightarrow X_2 \in [41, 49]$	100	2.38	7	11.33
#3	$t \in [32, 47] \wedge X_2 \in [46, 53] \Longrightarrow X_1 \in [16, 22]$	62.5	2.98	5	9.35
#4	$t \in [58, 75] \wedge X_2 \in [40, 50] \Longrightarrow X_1 \in [4, 15]$	100	3.23	13	12.67
#5	$X_1 \in [4, 14] \wedge X_2 \in [60, 70] \Longrightarrow t \in [54, 72]$	100	5.26	8	12.67
#6	$t \in [79, 95] \wedge X_1 \in [12, 18] \Longrightarrow X_2 \in [39, 49]$	85.7	1.82	6	10.62
#7	$t \in [76, 91] \wedge X_1 \in [15, 24] \Longrightarrow X_2 \in [41, 50]$	85.7	1.75	6	10.95
#8	$t \in [77, 95] \wedge X_1 \in [16, 22] \Longrightarrow X_2 \in [31, 39]$	100	33.33	3	10.62
#9	$t \in [76, 99] \wedge X_1 \in [18, 40] \Longrightarrow X_2 \in [31, 50]$	100	1.79	18	21.33
#10	$X_1 \in [28, 34] \wedge X_2 \in [36, 43] \Longrightarrow t \in [83, 98]$	60.0	4.29	3	9.35
#11	$t \in [76, 99] \wedge X_1 \in [30, 38] \Longrightarrow X_2 \in [40, 50]$	100	1.89	13	13.67

conjunction $\#3_1$ and rule #7 is approximately $\#3_2$. In the same fashion, rule #10 is related to $\#3_3$ and rule #11 to $\#3_4$.

The remaining rules discovered relationships varying among several conjunctions. For rule #6, note that the X_2 time series has values ranging in an interval formed from the union of rules $\#3_1$ and $\#3_2$. Last, rule #9 is a rule resulting from the four possible conjunctions in $t \in [76, 100]$, that is, it combines the intervals for both X_1 and X_2. Therefore, it would be a rule shared by $\#3_1, \#3_2, \#3_3$ and $\#3_4$. In general, rules in this interval of time share a support close to the expected one as well as a confidence verging on 100% for most cases.

Finally, it can be concluded that all the rules discovered by QARGA can be considered interesting since the lift is high for all of them. Moreover, the amplitude of intervals is moderate and intervals limits are very similar to those initially set in Table 5.

5.3.3 Results in ozone time series

Now QARGA is applied to ozone time series and other inter-dependant temporal variables. Table 17 shows the support, confidence, number of records, average amplitude and lift of the obtained rules by QARGA when the ozone is imposed to be in the consequent. The climatological variables that most frequently appear are temperature, humidity and hour of the day. Consequently, it can be concluded that the other variables are not as correlated with ozone as the aforementioned ones.

Some other interesting conclusions can be extracted from these rules. Hence, when the temperature reaches high values, the ozone concentration in the atmosphere presents high values, even reaching 203 μg/m^3. Nevertheless, when the temperature is relatively low, the concentration of ozone falls to values around 116 μg/m^3. That is, there exists a perfect correlation between the ranges of the temperature and the ozone. With reference to the humidity, there exists an inversely proportional relationship to the ozone. Thus, when examining the first rule, in contrast to the temperature, when the humidity falls, the ozone raises, and viceversa, as occurred in the fourth rule (rule #3).

From the remaining rules, it can also be observed that the time slot is present in two rules. This fact is due to the close association existing between the temperature and the hour of the day and, possibly, to the traffic, whose density varies along the day and typically generates high concentrations of ozone. Note that during the night and first hours

Table 17 Association rules found by QARGA with high confidence

ID	Rule	Sup. (%)	Conf. (%)	#r	Ampl.	Lift
#0	temp. $\in [32, 42]$ and hum. $\in [19, 41]$ and hour $\in [13, 19] \Longrightarrow O_3 \in [104, 203]$	14	84	213	34	2.27
#1	hour $\in [2, 11] \Longrightarrow O_3 \in [16, 97]$	37	90	557	45	1.64
#2	hour $\in [13, 22] \Longrightarrow O_3 \in [88, 189]$	35	84	522	55	1.67
#3	temp. $\in [16, 22]$ and hum. $\in [75, 90] \Longrightarrow O_3 \in [22, 110]$	16	100	234	36	1.43
#4	temp. $\in [18, 29]$ and speed $\in [0, 10] \Longrightarrow O_3 \in [23, 116]$	41	93	613	38	1.28
		28.6 (±12.6)	90.2 (±6.7)	427.80 (±189.5)	41.6 (±8.6)	1.7 (±0.4)

of the day, the ozone is relatively low, reaching values similar to that of low temperatures. However, from midday to the nightfall—the rushing hours—the amount of ozone increases considerably, reaching values near to 200 µg/m^3 as it happened with high temperatures.

Also note that in one rule the speed of the wind appears indicating that when it is low the ozone also is. However, this rule is not conclusive and the authors do not dare to state that the speed of the wind is directly proportional to the ozone.

With the aim of comparing the results and evaluating the quality, the Apriori algorithm has been applied to these time series. The most remarkable feature of this algorithm is that is based on a previous or *a priori* knowledge of the frequent itemsets in order to reduce the space of search and, consequently, increase the efficiency. Besides, the user has to establish the constraints for minimum support and confidence. It is also worth mentioning that Apriori does not work with real values directly and it performs a previous discretization of all continuous variables.

Hence, Table 18 collects the results provided by Apriori when discretizing the continuous variables with three intervals. In this case, the temperature and the humidity appear again but, by contrast, the hour of the day does not seem to be an important variable. The speed and the direction of the wind also appear in the antecedent.

It can also be observed that low temperatures also involve low ozone concentrations, and viceversa, as it happened with the rules shown in Table 17. With regard to the humidity the same situation is reported: it is inversely proportional to the ozone. However, when analyzing the direction of the wind in some rules, the results are not conclusive. Actually, for equal values of the direction, different ranges of ozone are mined, which means that this variable presents no proportional (neither direct nor inverse) relationship with the ozone and, therefore, it does not contribute with meaningful information. Finally, the speed of the wind presents the same behavior shown in Table 17, that is, low values involve low ozone concentrations.

The comparison between Tables 17 and 18 reveals that the support reached by QARGA is much greater in three rules whereas two rules present slightly lower supports. The confidence for the majority of the rules found by QARGA overcomes 90%, even reaching 100% in the fourth rule. This fact highlights the small errors committed by QARGA, providing exact rules in the majority of cases. Furthermore, the number of covered instances is higher than the ones by Apriori due to the direct relation existing with the support. The average amplitude for the rules provided by QARGA is much smaller, ranging from 38 to 55, while the intervals found by Apriori varies from 32 to 98. The lift is very similar in QARGA and Aprori, and for both algorithms it is greater than 1.

Last, Apriori has just found rules in which the values of the ozone varied only in two of the three possible intervals associated with the labels previously generated during the discretization process. Furthermore, it is unable to find rules with ozone concentrations higher than 183 µg/m^3. On the contrary, QARGA obtained rules for concentrations of ozone higher than 200 µg/m^3.

To sum up, from this kind of experimentation it can be concluded that QARGA obtains better results compared with Apriori, since support and confidence are higher and amplitude is smaller, which involves less errors in rules.

Table 19 shows the rules obtained by QARGA when the target is to find rules with the highest number of attributes, the highest confidence and the smallest amplitude possible, even if this fact may lead to lower supports. It can be observed that the majority of rules have a large number of attributes. The variables that most frequently appear, therefore the most meaningful ones, are the temperature, the humidity and the hour of day.

From the extracted rules, several conclusions can be drawn. First, note that the selected rules are those that present high concentrations of ozone in the consequent, since this is the situation that really involves environmental concerns. As with the first experimentation, a directly proportional relation between the temperature and the ozone has been discovered. In other words, when the

Table 18 Association rules found by Apriori (three intervals used for discretization)

ID	Rule	Sup. (%)	Conf. (%)	#r	Ampl.	Lift
#0	temp. ∈ [16, 25] and hum. ∈ [65, 91] and speed ∈ [0, 9] $\Longrightarrow O_3$ ∈ [14, 99]	20	90	297	32	1.59
#1	temp. ∈ [16, 25] and dir. ∈ [120, 240] and speed ∈ [0, 9] $\Longrightarrow O_3$ ∈ [14, 99]	16	85	239	56	1.49
#2	dir. ∈ [240, 360] and speed ∈ [0, 9] $\Longrightarrow O_3$ ∈ [14, 99]	16	79	241	71	1.38
#3	hum. ∈ [14, 40] and dir. ∈ [120, 240] $\Longrightarrow O_3$ ∈ [99, 183]	18	73	261	77	1.80
#4	temp. ∈ [25, 35] and dir. ∈ [120, 240] $\Longrightarrow O_3$ ∈ [99, 183]	20	70	296	98	1.70
		18.0 (±2.0)	79.4 (±8.3)	266.8 (±28.5)	66.8 (±24.6)	1.6 (±0.2)

Table 19 Association rules found by QARGA with high confidence

ID	Rule	Sup. (%)	Conf. (%)	#r	Ampl.	Lift
#0	temp. $\in [38, 42]$ and hum. $\in [25, 33]$ and hour $\in [15, 18] \Longrightarrow O_3 \in [140, 206]$	3	90	47	20	6.61
#1	temp. $\in [26, 33]$ and hum. $\in [29, 46]$ and dir. $\in [149, 231]$ and speed $\in [12, 20]$ and hour $\in [15, 19] \Longrightarrow O_3 \in [103, 160]$	3	93	40	29	2.90
#2	temp. $\in [29, 37]$ and hum. $\in [21, 36]$ and dir. $\in [161, 187]$ and hour $\in [14, 19]$ $\Longrightarrow O_3 \in [109, 180]$	5	83	70	25	2.83
#3	temp. $\in [34, 42]$ and hum. $\in [22, 32]$ and dir. $\in [152, 189]$ and speed $\in [9, 16]$ and hour $\in [15, 17] \Longrightarrow O_3 \in [134, 206]$	2	87	33	23	4.99
#4	hum. $\in [22, 32]$ and hour $\in [13, 18] \Longrightarrow O_3 \in [144, 198]$	7	48	97	23	4.27
		4.0 (±2.0)	80.2 (±18.4)	57.4 (±26.1)	24.0 (±3.3)	4.32 (±1.58)

temperature reaches values of almost 40 °C the ozone level raises up to values greater than 200 μg/m^3.

Moreover, the humidity also presents an inversely proportional relationship with the ozone. Thus, when it reaches high values, the concentration of ozone is low, as it can be determined from the observation of the second rule. Alternatively, when the humidity increases, the ozone level decreases, as listed in the remaining rules.

The hour of the day is also present in the majority of the rules. The time slot is similar for all the rules since, as discussed previously, ozone and the hour share a directly proportional relationship. The peaks of ozone are reached during the rushing hours (from midday to nightfall), that is, during the hours in which the temperature is high and the traffic is usually heavy.

In some rules, the speed of the wind appears as a crucial factor. However, there does not exist such a higher correlation with the ozone because greater speeds should have be found in the fourth rule (in which the ozone presents the highest concentration). By contrast, in the second rule, in which the ozone is lower, the speed of the wind is slightly superior.

The analysis of the direction of the wind reveals that it is not a variable that determines the amount of ozone in the atmosphere. However, when the direction is comprised in an interval from 150° and 200°, the concentration of ozone increases.

Table 20 gathers the results obtained by Apriori when data were discretized in ten intervals. The temperature is, again, the main variable. However, it is worth pointing out that no relevant rules were discovered in which the humidity or the hour of the day appear.

One of the most remarkable feature of the extracted rules by Apriori when discretizing with ten intervals is that they all have only one attribute in the antecedent. This situation highlights, once again, that rules provided by QARGA enhance that of Apriori, since they are more expressive and provide more information due to a greater number of attributes in antecedents.

With respect to the temperature, two rules with different antecedent but same consequent have been discovered. Note that they could have been fused into one rule as the consequent is the same. Besides, the obtained confidence is quite low which leads to rules with considerably high errors.

The case of the direction of the wind is similar to that of the temperature. The third and fifth rules share the same antecedent for the same direction and, however, the consequent for both rules is different even when they could have been fused into just one rule. The confidence hardly reaches 30%, which leads to an almost null reliability.

The speed of the wind appears in one rule in which its value is low and the ozone presents medium values. However, the confidence is quite low.

Table 20 Association rules found by Apriori (ten intervals used for discretization)

ID	Rule	Sup. (%)	Conf. (%)	#r	Ampl.	Lift
#0	temp. $\in [27, 30] \Longrightarrow O_3 \in [90, 115]$	5	37	79	14	1.43
#1	temp. $\in [24, 27] \Longrightarrow O_3 \in [90, 115]$	6	33	85	14	1.43
#2	dir. $\in [144, 180] \Longrightarrow O_3 \in [90, 115]$	10	29	156	31	1.18
#3	speed $\in [6, 8] \Longrightarrow O_3 \in [90, 115]$	5	24	75	14	1.03
#4	dir. $\in [144, 180] \Longrightarrow O_3 \in [115, 141]$	6	16	88	31	1.27
		6.4 (±2.1)	27.8 (±8.2)	96.6 (±33.6)	20.8 (±9.3)	1.27 (±0.17)

The comparison of the Tables 19 and 20 leads to several conclusions. As regards the attributes, QARGA always obtains rules with greater number of them and, consequently, the information brought by these rules is higher than that of Apriori.

When taking into consideration the support, QARGA presents low values since there are few instances in the dataset with high ozone values. However, if the confidence of the rules for both algorithms is compared, it can be observed that QARGA has values even greater than 90% while Apriori never overcomes 40%.

Unlike the lift values from the first kind of experimentation, where the interest of the rules in QARGA and Apriori was quite similar, it can be observed that, in this case, the results of the lift are very different. Rules found by QARGA present lift values between 3 and 6, while Apriori never exceeds 1.50. This is an important result that indicates that QARGA find more interesting rules than Apriori does.

Another relevant remark is that Apriori discovers rules with different intervals for the same variable in the antecedent but equal consequents and viceversa. This fact never occurs in QARGA.

The ozone levels obtained by Apriori never exceeds 140 $\mu g/m^3$, while QARGA reached values greater than 200 $\mu g/m^3$. This appreciation is of the utmost relevance, since environment is really concerned by high levels of ozone and, consequently, discovering rules with these values of ozone is useless.

6 Conclusions

An evolutionary algorithm has been proposed in this work to obtain QAR from time series. In order to evaluate its performance, the approach has been applied to several datasets and compared with the most recently published results. Thus, a bank of public datasets retrieved from the BUFA repository has been used to test the accuracy of the algorithm. The algorithm has shown to be efficient when mining synthetically generated multidimensional time series. Also, the proposed methodology has successfully obtained meaningful QAR from multidimensional real-world time series. In particular, relevant dependencies between the ozone concentration in the atmosphere and other climatological-related time series have been found.

Acknowledgments The financial support from the Spanish Ministry of Science and Technology, project TIN2007-68084-C02, and from the Junta de Andalucía, project P07-TIC-02611, is acknowledged. The authors also want to acknowledge the support by the Regional Ministry for the Environment (*Consejería de Medio Ambiente*) of Andalucía (Spain), that has provided all the pollutant agents time series.

References

Adame-Carnero JA, Bolfvar JP, de la Morena BA (2010) Surface ozone measurements in the southwest of the Iberian Peninsula. Environ Sci Pollut Res 17(2):355–368

Agrawal R, Imielinski T, Swami A (1993) Mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM SIGMOD international conference on management of data, pp 207–216

Agirre-Basurko E, Ibarra-Berastegi G, Madariagac I (2006) Regression and multilayer perceptron-based models to forecast hourly o_3 and no_2 levels in the Bilbao area. Environ Model Softw 21:430–446

Aguilar-Ruiz JS, Giráldez R, Riquelme JC (2007) Natural encoding for evolutionary supervised learning. IEEE Trans Evol Comput 11(4):466–479

Alatas B, Akin E (2006) An efficient genetic algorithm for automated mining of both positive and negative quantitative association rules. Soft Comput 10(3):230–237

Alatas B, Akin E (2008) Rough particle swarm optimization and its applications in data mining. Soft Comput 12(12):1205–1218

Alatas B, Akin E, Karci A (2008) MODENAR: multi-objective differential evolution algorithm for mining numeric association rules. Appl Soft Comput 8(1):646–656

Alcalá-Fdez J, Alcalá R, Gacto MJ, Herrera F (2009a) Learning the membership function contexts forming fuzzy association rules by using genetic algorithms. Fuzzy Sets Syst 160(7):905–921

Alcalá-Fdez J, Sánchez L, García S, del Jesus MJ, Ventura S, Garrell JM, Otero J, Romero C, Bacardit J, Rivas VM, Fernández JC, Herrera F (2009b) Keel: a software tool to assess evolutionary algorithms for data mining problems. Soft Comput 13(3):307–318. http://sci2s.ugr.es/keel

Alcalá-Fdez J, Flugy-Pape N, Bonarini A, Herrera F (2010) Analysis of the effectiveness of the genetic algorithms based on extraction of association rules. Fundam Inform 98(1):1001–1014

Aumann Y, Lindell Y (2003) A statistical theory for quantitative association rules. J Intell Inf Syst 20(3):255–283

Bellazzi R, Larizza C, Magni P, Bellazzi R (2005) Temporal data mining for the quality assessment of hemodialysis services. Artif Intell Med 34:25–39

Berlanga FJ, Rivera AJ, del Jesus MJ, Herrera F (2010) GP-COACH: genetic programming-based learning of compact and accurate fuzzy rule-based classification systems for high-dimensional problems. Inf Sci 180(8):1183–1200

Brin S, Motwani R, Silverstein C (1997) Beyond market baskets: generalizing association rules to correlations. In: Proceedings of the 1997 ACM SIGMOD international conference on management of data, vol 26, pp 265–276

Chen CH, Hong TP, Tseng V (2009) Speeding up genetic-fuzzy mining by fuzzy clustering. In: Proceedings of the IEEE international conference on fuzzy systems, pp 1695–1699

Chen CH, Hong TP, Tseng V (2010) Genetic-fuzzy mining with multiple minimum supports based on fuzzy clustering. Soft Comput (in press)

del Jesús MJ, Gámez J, Puerta J (2009) Evolutionary and metaheuristics based data mining. Soft Comput Fusion Found Methodol Appl 13:209–212

Elkamel A, Abdul-Wahab S, Bouhamra W, Alper E (2001) Measurement and prediction of ozone levels around a heavily industrialized area: a neural network approach. Adv Environ Res 5:47–59

García S, Fernández A, Luengo J, Herrera F (2009) A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability. Soft Comput 13(10):959–977

Georgii E, Richter L, Ruckert U, Kramer S (2005) Analyzing microarray data using quantitative association rules. BMC Bioinformatics 21(2):123–129

Gupta N, Mangal N, Tiwari K, Pabitra Mitra (2006) Mining quantitative association rules in protein sequences. Lect Notes Artif Intell 3755:273–281

Guvenir HA, Uysal I (2000) Bilkent university function approximation repository. http://funapp.cs.bilkent.edu.tr

Herrera F, Lozano M, Sánchez AM (2004) Hybrid crossover operators for real-coded genetic algorithms: an experimental study. Soft Comput 9(4):280–298

Huang YP, Kao LJ, Sandnes FE (2008) Efficient mining of salinity and temperature association rules from ARGO data. Expert Syst Appl 35:59–68

Kalyanmoy D, Ashish A, Dhiraj J (2002) A computationally efficient evolutionary algorithm for real-parameter optimization. Evol Comput 10(4):371–395

Khan MS, Coenen F, Reid D, Patel R, Archer L (2010) A sliding windows based dual support framework for discovering emerging trends from temporal data. Res Dev Intell Syst Part 2:35–48

Lin MY, Lee SY (2002) Fast discovery of sequential patterns by memory indexing. In: Proceedings of the 4th international conference on data warehousing and knowledge discovery, pp 150–160

Martínez–Álvarez F, Troncoso A, Riquelme JC, Aguilar JS (2011) Energy time series forecasting based on pattern sequence similarity. IEEE Trans Knowl Data Eng (in press)

Mata J, Álvarez J, Riquelme JC (2001) Mining numeric association rules with genetic algorithms. In: Proceedings of the international conference on adaptive and natural computing algorithms, pp 264–267

Mata J, Álvarez JL, Riquelme JC (2002) Discovering numeric association rules via evolutionary algorithm. Lect Notes Artif Intell 2336:40–51

Nam H, Lee K, Lee D (2009) Identification of temporal association rules from time-series microarray data sets. BMC Bioinformatics 10(3):1–9

Nikolaidou V, Mitkas PA (2009) A sequence mining method to predict the bidding strategy of trading agents. Lect Notes Comput Sci 5680:139–151

Orriols-Puig A, Bernadó-Mansilla E (2009) Evolutionary rule-based systems for imbalanced data sets. Soft Comput Fusion Found Methodol Appl 13:213–225

Orriols-Puig A, Casillas J, Bernadó-Mansilla E (2008) First approach toward on-line evolution of association rules with learning classifier systems. In: Proceedings of the 2008 GECCO genetic and evolutionary computation conference, pp 2031–2038

Pei J, Han JW, Mortazavi-Asl B, Pinto H, Chen Q, Dayal U, Hsu MC (2001) Prefixspan: mining sequential patterns efficiently by prefix-projected pattern growth. In: Proceedings of IEEE conference on data engineering, pp 215–224

Ramaswamy S, Mahajan S, Silberschatz A (1998) On the discovery of interesting patterns in association rules. In: Proceedings of the 24th international on very large data bases, pp 368–379

Sheskin D (2006) Handbook of parametric and nonparametric statistical procedures. Chapman and Hall/CRC

Shidara Y, Kudo M, Nakamura A (2008) Classification based on consistent itemset rules. Trans Mach Learn Data Min 1(1):17–30

Tong Q, Yan B, Zhou Y (2005) Mining quantitative association rules on overlapped intervals. Lect Notes Artif Intell 3584:43–50

Tung AKH, Han J, Lu H, Feng L (2003) Efficient mining of intertransaction association rules. IEEE Trans Knowl Data Eng 15(1):43–56

Vannucci M, Colla V (2004) Meaningful discretization of continuous features for association rules mining by means of a som. In: Proceedings of the European symposium on artificial neural networks, pp 489–494

Venturini G (1993) SIA: a supervised inductive algorithm with genetic search for learning attribute based concepts. In: Proceedings of the European conference on machine learning, pp 280–296

Venturini G (1994) Fast algorithms for mining association rules in large databases. In: Proceedings of the international conference on very large databases, pp 478–499

Wan D, Zhang Y, Li S (2007) Discovery association rules in time series of hydrology. In: Proceedings of the IEEE international conference on integration technology, pp 653–657

Wang YJ, Xin Q, Coenen F (2008) Hybrid rule ordering in classification association rule mining. Trans Mach Learn Data Min 1(1):17–30

Winarko E, Roddick JF (2007) ARMADA—an algorithm for discovering richer relative temporal association rules from interval-based data. Data Knowl Eng 63:76–90

Wright SP (1992) Adjusted *p*-values for simultaneous inference. Biometrics 48:1005–1013

Yan X, Zhang C, Zhang S (2009) Genetic algorithm-based strategy for identifying association rules without specifying actual minimum support. Expert Syst Appl Int J 36(2):3066–3076