In this paper, a genetic algorithm (GA) is proposed as a search strategy for not only positive but also negative quantitative association rule (AR) mining within databases. Contrary to the methods used as usual, ARs are directly mined without generating frequent itemsets. The proposed GA performs a database-independent approach that does not rely upon the minimum support and the minimum confidence thresholds that are hard to determine for each database. Instead of randomly generated initial population, uniform population that forces the initial population to be not far away from the solutions and distributes it in the feasible region uniformly is used. An adaptive mutation probability, a new operator called uniform operator that ensures the genetic diversity, and an efficient adjusted fitness function are used for mining all interesting ARs from the last population in only single run of GA. The efficiency of the proposed GA is validated upon synthetic and real databases.

Keywords Data mining · Quantitative association rules · Negative association rules · Genetic algorithm

1 Introduction

Data mining is the extraction of implicit, valid, and potentially useful knowledge from large volumes of raw data [1]. The extracted knowledge must be not only accurate but also readable, comprehensible and ease of understanding.

Association rule (AR) mining is one of the important research problems in data mining field where the goal is to derive multi-feature (attribute) correlations from databases. In this study, the goal is to find positive and negative ARs in quantitative databases without the necessity of previously preparing the data. Itemsets are considered within the antecedent or the consequent being negated. Then the rules to be considered are of the form:

\[
\begin{align*}
B & \in \neg [b_1, b_2] \Rightarrow C \in [c_1, c_2] \land D \in [d_1, d_2] \\
A & \in [a_1, a_2] \land B \in [b_1, b_2] \Rightarrow D \in \neg [d_1, d_2] \\
A & \in [a_1, a_2] \land C \in [c_1, c_2] \Rightarrow B \in \neg [a_1, a_2] \\
\land D & \in [d_1, d_2]
\end{align*}
\]

Here \(A, B, C, \) and \(D\) show the quantitative attributes, \(\neg [l_1, l_2]\) shows the intervals not in \([l_1, l_2]\), and, \(l_1\) and \(l_2\) show the lower and upper bound of the interval.

In order to get this objective, an efficient genetic algorithm (GA) has been designed to simultaneously search for intervals of quantitative attributes and the discovery of ARs that these intervals conform in only single run. Furthermore, a database-independent approach that does not rely upon the minimum support and the minimum confidence thresholds that are hard to determine for each database has been performed. Contrary to the methods used as usual, ARs have directly been mined without generating frequent itemsets. Uniform population (UP) for initial population [2]; adaptive mutation, a new operator called uniform operator (UO), and an adjusted fitness function that will be described in Sect. 3 have been used for effectively mining all interesting positive and negative quantitative association rules in the last population of GA.

This paper is organized as follows. Section 2 describes the quantitative and negative ARs with related works. Section 3 explains the details of the algorithm. Section 4 briefly describes the used databases and discusses the experimental results. Finally Sect. 5 concludes the paper.

2 Quantitative and negative ARs

The Boolean AR mining problem over basket data was introduced in [3]. In this and many algorithms proposed afterwards for mining the ARs are divided into two stages: the first is to find the frequent itemsets; the second is to use the frequent itemsets to generate ARs. The mined rules have certain support and confidence values. Though Boolean ARs are
meaningful, there are many other situations where data items concerned are usually categorical or quantitative. That is why, quantitative AR mining algorithms have been proposed in [4] by first partitioning the attributes domains into small intervals and combining adjacent intervals into larger one such that the combined intervals will have enough supports. In fact, quantitative problem has been transformed to a Boolean one. Some researchers used geometric means to find numeric intervals for quantitative values [5]. The found association rules had no more than two numeric variables in the antecedent and a single Boolean variable in the consequent. Aumann and Lindell used the distribution of a numerical value as the criteria for inclusion in the association rule [6]. Their contention was that an association rule could be thought of as a population subset (the rule consequent) exhibiting some interesting behavior (the rule antecedent). They investigated two types of quantitative rules: categorical⇒quantitative rules and quantitative⇒quantitative rules.

Limitations on these quantitative association rule algorithms, in general, were the numbers of variables allowed in either the consequent or antecedent of the rules. In addition, it is not allowed for both Boolean and multiple quantitative values to be in both the consequent and/or in the antecedent of the rule.

Diverse researchers afterwards have used clustering techniques, partitioning by means of fuzzy sets, however all of them have in common the fact that they need information a priori from the user.

The main problem of all these approaches is preparation of the data before applying the algorithm. This preparation, either by means of the user or by means of an automatic process, conveys a loss of information because the rules will only be generated departing from the partitions previously created. Furthermore, except fuzzy sets these approaches have some drawbacks. The first problem is caused by sharp boundary between intervals that is not intuitive with respect to human perception. The algorithms either ignore or over-emphasize the elements near the boundary of the intervals. Furthermore, distinguishing the degree of membership for the interval method is not easy. The idea of using GA for mining only positive frequent sets was first applied in [7]. However, the encoding used in this work is not much effective for genetic operators to be performed because of variable size. Furthermore, only positive frequent itemsets were mined by running the GA as many times as frequent itemsets that have been wanted to obtain and this has a big computational cost.

Negative ARs are ARs between the antecedent and consequent of the rule. Either the antecedent or consequent or both have to be negated in order for the rule to be a negative rule. There has been no work for mining negative quantitative ARs, however these rules further complete associated relationships among attributes as a system in science and technology and, they offer more information that might be of use in supporting decisions for applications. Furthermore, sometimes intervals for attributes that conform rules may be in \([L, x_1] \cup [x_2, U]\) where \(x_2 > x_1\), \(L\) is the lower bound and \(U\) is upper bound of the attribute as shown in gray colored area of Fig. 1.

![Fig. 1 Negative interval](image)

### 3 The proposed GA

In this work, a high-quality population, UP, in the beginning has been generated and then the search of positive and negative ARs has been started. The chromosomes of initial population are far away from each others and dispersed the feasible region uniformly. For mining all rules in a single GA run, efficient techniques have been used. The flowchart of the proposed GA is shown in Fig. 2. In each generation of GA, two fittest chromosomes are selected for new operator called UO and two genetic operators, crossover and mutation, are performed to stochastically selected chromosomes for evolution. In each iteration, the number of new chromosomes is equal to the population size. Thus, the number of chromosomes in the population is doubled. At this stage, the rules are evaluated and adjusted fitness value is computed. One half of the high-quality chromosomes are retained and passed to the next iteration. The whole process is performed iteratively until the maximum number of generations is reached. The main characteristics of this algorithm have been described in the following subsections.

#### 3.1 Encoding

In this work, the chromosomes that are being produced and modified along the genetic process represent rules. Each chromosome consists of genes that represent the items and intervals. A positional encoding, where the \(i\)-th item is encoded in the \(i\)-th gene has been used. Each gene has four parts. The first part of each gene represents the antecedent or consequent of the rule and can take three values: ‘0’, ‘1’ or ‘*’. If the first part of the gene is ‘0’, it means that this item will be in the antecedent of the rule and if it is ‘1’, this item will be in the consequent of the rule. If it is ‘*’, it means that this item will not be involved in the rule. All genes that have ‘0’ on their first part will form the antecedent of the rule while genes that have ‘1’ on their first part will form the consequent of the rule. Second part of the gene represents the positive or negative ARs. This part can take two values: ‘0’ or ‘1’. If the second part of the gene is ‘0’, this means that the interval of this item will be negated in the rule, that is, it forms a negative rule. If it is ‘1’ it will be used for mining positive ARs. While the third part represents the lower bound, the