

Analysis of the Effectiveness of the Genetic Algorithms based on Extraction of Association Rules

Jesus Alcalá-Fdez^{*†}

*Department of Computer Science and A.I.
CITIC-UGR
University of Granada
18071 Granada, Spain
jalcala@decsai.ugr.es*

Andrea Bonarini

*Department of Electronics and Information
Politecnico di Milano
20133 Milano, Italy
bonarini@elet.polimi.it*

Nicolo Flugy-Pape

*Department of Electronics and Information
Politecnico di Milano
20133 Milano, Italy
nicolo.flugy@mail.polimi.it*

Francisco Herrera^{*}

*Department of Computer Science and A.I.
CITIC-UGR
University of Granada
18071 Granada, Spain
herrera@decsai.ugr.es*

Abstract. Data Mining is most commonly used in attempts to induce association rules from transaction data which can help decision-makers easily analyze the data and make good decisions regarding the domains concerned. Most conventional studies are focused on binary or discrete-valued transaction data, however the data in real-world applications usually consists of quantitative values. In the last years, many researches have proposed Genetic Algorithms for mining interesting association rules from quantitative data. In this paper, we present a study of three genetic association rules extraction methods to show their effectiveness for mining quantitative association rules. Experimental results over two real-world databases are showed.

Keywords: Association Rules, Data Mining, Evolutionary Algorithms, Genetic Algorithms

^{*}This paper has been supported by the Spanish Ministry of Education and Science under Project TIN2008-06681-C06-01.

[†]Address for correspondence: Department of Computer Science and Artificial Intelligence, CITIC-UGR (Research Center on Information and Communications Technology), University of Granada, 18071 Granada, Spain

1. Introduction

Data Mining (DM) is the process used for the automatic discovery of high level knowledge through obtaining information from real-world, large and complex data sets. In the last decade, the digital revolution has provided relatively inexpensive and accessible means to collect and store data. The increase in data volume has caused greater difficulty in the extraction of useful information for decision support. The use of DM to facilitate decision support can lead to the improved performance of decision making and can enable the tackling of new types of problems that have not been addressed before [30, 36].

Discovering association rules is one of several data mining techniques described in the literature [15]. Association rules are used to represent and identify dependencies between items in a database [42]. These are an expression of the type $X \rightarrow Y$, where X and Y are sets of items and $X \cap Y = \emptyset$. This means that if all the items in X exist in a transaction then all the items in Y with a high probability are also in the transaction, and X and Y should not have any common items [1, 2]. Knowledge of this type of relationship can enable proactive decision making to proceed from the inferred data. Many problem domains have a need for this type of analysis, including risk management, medical diagnostic, fire management in national forests and others. For example, fire management decision makers can make better decisions regarding prescribed burns and other suppression techniques when looking at current climatic information, if they know how these climatic factors influence fire risk.

Many previous studies for mining association rules focused on databases with binary or discrete values, however the data in real-world applications usually consists of quantitative values. While the classical algorithms are effective and efficient, they can difficultly be used directly in the discovery of association rules from quantitative data because the numerical attributes typically contain many distinct values. A commonly used method is to partition the domains introducing new attributes with intervals. The support for any particular value is likely to be low, while the support for intervals is much higher. So, quantitative association rules such as $Age \in [5, 10] \rightarrow height \in [0.5, 1.5]$ can be discovered using classical algorithms.

Lately, many researchers have proposed Evolutionary Algorithms (EAs) [10] for mining quantitative association rules [27, 28, 29, 40]. EAs, particularly Genetic Algorithms (GAs) [12, 19], are considered to be one of the most successful search techniques for complex problems and it has proved to be an important technique for learning and knowledge extraction. The main motivation for applying GAs to knowledge extraction tasks is that they are robust and adaptive search methods that perform a global search in place of candidate solutions (for instance, rules or other forms of knowledge representation). Moreover, GAs let us obtain feasible solutions in a limited amount of time. Hence, there has been growing interest in EAs in the field of DM [9, 11, 14, 21, 31, 38].

The aim of this paper is to show the effectiveness of GAs for mining quantitative association rules. We present an experimental study of two real-world databases to show the behaviour of three GAs for mining association rules:

- EARMGA: Evolutionary Association Rules Mining with Genetic Algorithm [40]
- GAR: Genetic Association Rules [29],
- GENAR: GENetic Association Rules [28],

As a point of reference, we must highlight the two following keypoints in the study:

- This study includes the comparative analysis of two classical methods, a Tree-based implementation of Apriori [7] and Eclat [41], using a uniform partition in each quantitative attribute.
- An important aspect in the analysis of the genetic extraction is the scalability of methods. Experiments have also been carried out paying attention to the scalability of the genetic extraction methods.

This paper is arranged as follows. The next section provides a brief preliminary on genetic rules extraction. Section 3 describes the five methods included in the study. Section 4 shows the results of experimental studies that have been performed on two real-world databases. Finally, Section 5 provides some conclusions.

2. Preliminary: Genetic Rules Extraction

Although GAs were not specifically designed for learning, but rather as global search algorithms, they offer a set of advantages for machine learning. Many methodologies for machine learning are based on the search for a good model inside the space of possible models. In this sense, they are very flexible because the same GA can be used with different representations. Genetic learning processes cover different levels of complexity according to the structural changes produced by the algorithm, from the simplest case of parameter optimization to the highest level of complexity for learning the rule set of a rule-based system, via the coding approach and the cooperation or competition between chromosomes.

When considering a rule based system and focusing on learning rules, the different genetic learning methods follow two approaches in order to encode rules within a population of individuals [17]:

- The "Chromosome = Set of rules", also called the Pittsburgh approach, in which each individual represents a rule set [32]. In this case, a chromosome evolves a complete Rule Base (RB) and they compete among themselves along the evolutionary process. GASSITS and GABIL are proposals that follows this approach [4, 5, 23].
- The "Chromosome = Rule" approach, in which each individual codifies a single rule, and the whole rule set is provided by combining several individuals in a population (rule cooperation) or via different evolutionary runs (rule competition). Within the "Chromosome = Rule" approach, there are three generic proposals:
 - Michigan approach, in which each individual codifies an association rule. These kinds of systems are usually called learning classifier systems [18]. They are rule-based, messagepassing systems that employ reinforcement learning and a GA to learn rules that guide their performance in a given environment. The GA is used for detecting new rules that replace the bad ones via the competition between the chromosomes in the evolutionary process. An interesting study on the topic can be found in [34].
 - The IRL (Iterative Rule Learning) approach, in which each chromosome represents a rule. Chromosomes compete in every GA run, choosing the best rule per run. The global solution is formed by the best rules obtained when the algorithm is run multiple times. SIA [37] is a proposal that follows this approach.

- The GCCL (genetic cooperative-competitive learning) approach, in which the complete population or a subset of it encodes the RB. In this model the chromosomes compete and cooperate simultaneously. COGIN [13] is an example of this representation.

In the literature we can find interesting works based on the Michigan approach [39, 40] and the IRL approach [28, 29] for mining association rules from quantitative data.

3. Association Rules Mining: Algorithms for the Analysis

In this paper, we have analyzed five methods for mining association rules:

- Classical algorithms: Apriori [7, 33] and Eclat [41].
- GAs for mining association rules: EARMGA [40], GAR [29] and GENAR [28].

In the next subsections we describe these methods in detail. First, we introduce the classical methods and then the methods based on GAs for mining association rules.

3.1. Association rules mining through classical algorithms: Apriori and Eclat

In this study we use two classical algorithms as a point of reference, Apriori and Eclat. Among classical algorithms, it is worthwhile to mention Apriori because is the first successful algorithm used for mining association rules. Several implementations based on this method can be found in the literature, with the basic aim of speeding up the support counting [6, 7, 16]. In this paper, we have used a fast implementation of Apriori which uses Tree [7]. In addition we have chosen to analyze Eclat [41] because it exploits a different strategy to search for frequent itemsets.

The main aim of Apriori is to exploit the search space by means of the *downward closure property*. The latter states that any subset of a frequent itemset must also be frequent. As a consequence, it generates candidates for the current iteration by means of itemsets considered frequent at the previous iteration. Then it enumerates all the subsets for each transaction and increments the support of candidates matching them. Finally, those having the user-specified minimum support (*minSup*) are marked as frequent for the next iteration. This process is repeated until all frequent itemsets have been found. Thus, Apriori follows a *breadth-first* strategy to generate candidates.

By contrast, Eclat employs a *depth-first strategy*. It generates candidates by extending prefixes of an itemset until an infrequent one is found. In that case, it simply backtracks to the previous prefix and then recursively applies the above procedure. Unlike Apriori, the support counting is achieved by adopting a vertical layout. That is, for all the items in a database, it first constructs a list of all the transaction identifiers (tid-list) containing that item. Then it counts the support by merely intersecting two or more tid-lists to check whether they have items in common. In that case, the support is equal to the size of the resulting set.

These algorithms usually identify relationships among transactions in databases with binary values, however the data in real-world applications usually consist of quantitative values. The classical algorithms are effective and efficient but they can difficultly be used directly in the discovery of association rules from quantitative data. A commonly used method is to partition the domains introducing new attributes with intervals. So quantitative association rules can be discovered using classical algorithms.

Task partition, however, is a critical problem in the extraction of association rules because the information is not classified, as opposed to classification models. In this paper we use a uniform partition in each quantitative attribute [26], the usual discretization method used when we don't have additional information for using methods based on the information theory [35, 25] or other concepts [22].

3.2. Association rules mining through genetic algorithms: EARMGA, GAR and GENAR

We have used three GAs in the literature to achieve the association rules mining task:

- EARMGA [40]: It is based on the discovery of quantitative association rules.
- GAR [29]: It searches for frequent itemsets by dealing with numerical domains.
- GENAR [28]: It directly mines association rules by handling numerical domains.

A chromosome in EARMGA encodes a generalized k -rule, where k indicates the desired length. Since we may handle association rules with more than one item in the consequent, the first gene stores an index representing the end of the antecedent part. In order to uniquely encode a rule into a chromosome, both antecedent attributes and consequent attributes are sorted two-segmentally in an ascending order. Then the remaining k genes encode items. Each item is represented by a pair of values, where the first value is an attribute's index ranged from 1 to the maximum number of attributes in the database, whereas the second stands for a *gapped interval*. The authors have defined a gapped interval as the union of a finite number of *base intervals* obtained once a uniform discretization process has been accomplished over all attributes in the database. Notice that we do not need to partition the domains of categorical attributes because here the lower and the upper bounds basically coincide. Nevertheless, a base interval is always represented by an integer number apart from the kind of attributes we deal with. As a consequence, a gapped interval is a set of these integers. Now we will give some details of the genetic operators applied to each chromosome:

- Selection: it is achieved by computing the fitness value along with a random number, so that the chromosome will be selected only if this product is less than a given probability of selection (p_s).
- Crossover: all the selected chromosomes have the chance to reproduce offspring at a probability of crossover (p_c). This operation simply consists of exchanging a segment of genes between the first chromosome and the second one and vice-versa, depending on two crossover-points randomly generated.
- Mutation: by considering both a probability of mutation (p_m) and the fitness value, a chromosome is altered in the way that the boundary between antecedent attributes and consequent attributes could be changed within the same rule. In addition, the operator randomly chooses a gene and modifies the attribute's index along with the gapped interval associated with it. Notice that the new gapped interval is always a union of base intervals which now form a sub-domain of the new attribute.

Finally, the association rules mining problem has been restated by the authors of this work because the algorithm searches for k -association rules only by considering fitness values given by a measure of interest known as positive confidence [40].

By contrast, GAR follows different strategies. First, a chromosome is composed of a variable number of genes, between 2 and n , where n is the maximum number of attributes. However, as we find frequent itemsets with this method, it is afterwards necessary to run another procedure in order to generate association rules. Moreover, it is unnecessary to discretize a priori the domain of the attributes since each gene is represented by an upper and a lower bound along with an identifier for that attribute. To briefly recall the genetic operators for this method:

- **Selection:** it simply selects a percentage (p_s) of the chromosomes in the current population which have the best fitness. These ones will be the first individuals of the new-made population.
- **Crossover:** the new-made population is completed by reproducing offspring until reaching a desired size. To do that, the parents are randomly chosen at a probability of p_c . Then, we only obtain two different offspring when their parents have genes containing the same attribute. In that case, their intervals could simply be exchanged considering all the possible combinations between them, but, in the end, two chromosomes should always be generated. Finally, only the best one will be added to the population.
- **Mutation:** as usual, at the probability of p_m , it alters one gene in such a way that each limit could randomly decrease or increase its value.

Its fitness function tends to reward frequent itemsets that have a high support as well as a high number of attributes. In addition, it punishes frequent itemsets which have already covered a record in the database and whose intervals are too large.

GENAR was the first attempt by the same authors of GAR to handle continuous domains. Here a chromosome is encoded as an association rule which contains intervals as in the case of GAR. Nevertheless, the length of the rules is always fixed to the number of attributes and only the last attribute forms the consequent. Similar considerations can be taken into account regarding the definition of genetic operators, except for cross-over which employs a one-point strategy to reproduce offspring chromosomes [28]. By contrast, its fitness function only considers the support count for the rules and punishes those which have already covered the same records in the database.

4. Experimental Study

Several experiments have been carried out in this paper to evaluate the efficiency and effectiveness of the methods considered for comparison. In the following subsections, first we describe the two real-world databases used in these experiments, then we introduce the experimental set-up (determining all the parameters used) and analyze the number of intervals per attribute, later on we show the results obtained in the different experiments, and finally we present an analysis of the scalability of the algorithms.

4.1. Databases

In order to analyze the performance of the methods considered we have considered two real-world databases:

- *Stulong*: It is a database concerning a study of the risk factors of atherosclerosis in a population of 1419 middle-aged men in the years 1976 - 1999¹. The Stulong database consists of four data matrices: Entry, Control, Letter and Death. Here, we use the data of Entry and extract five quantitative attributes out of a total of 64 attributes. The selected attributes are height, weight, systolic blood pressure, diastolic blood pressure and cholesterol level.
- *House_16H*: It concerns a study to predict the median price of the houses in a region by considering both the demographic composition and the state of housing market. This data was collected as part of the 1990 US census. These are mostly counts cumulated at different survey levels. For the purpose of this database, only a level State-Place was used and data from all states was obtained. This database contains 22,784 transactions and 17 quantitative attributes².

4.2. Experimental set-up

The values considered for the input parameters of each analyzed method are:

- Apriori and Eclat: minSup = 0.1 and minConf = 0.8.
- EARMGA: maxloop = 100, popsize = 30 (100 with HH), k = 2 (4 with HH), ps = 0.75, pc = 0.7, pm = 0.1 and $\alpha = 0.01$.
- GAR: nItemset = 30 (100 with HH), nGen = 100, popsize = 100, ps = 0.25, pc = 0.7, pm = 0.1, af = 2.0, $\omega = 0.4$, $\psi = 0.7$ and $\mu = 0.5$.
- GENAR: nRules = 30 (100 with HH), nGen = 100, popsize = 100, ps = 0.25, pc = 0.7, pm = 0.1, af = 2.0 and pf = 0.7.

Notice that the number of Itemsets obtained with GAR, the number of rules obtained with GENAR and the length of the rules obtained with EARMGA are higher in the problem HH, since the number of attributes and transactions are higher in this problem. Furthermore, for all the experiments conducted in this study, the results shown in the tables always refer to association rules having a minimum confidence (minConf) greater than or equal to 0.8.

4.3. Analysis of the interval discretization for quantitative attributes

As we have commented in Subsection 3.1, a commonly used method to discover association rules from quantitative data is to partition the domains introducing new attributes with intervals. In this paper we use a uniform partition in each quantitative attribute [26], the usual discretization method used when we don't have additional information for using methods based on the information theory [35, 25] or other concepts [22].

¹The study (STULONG) was performed at the 2nd Department of Medicine, 1st Faculty of Medicine of Charles University and Charles University Hospital, under the supervision of Prof. F. Boudk with collaboration of M. Tomeckov and Ass. Prof. J. Bultas. The data were transferred to electronic form by the European Centre of Medical Informatics, Statistics and Epidemiology of Charles University and Academy of Sciences. The data resource is on the web page <http://euromise.vse.cz/challenge2004>. At present, the data analysis is supported by the grant of the Ministry of Education CR Nr LN 00B 107

²This database was designed on the basis of data provided by US Census Bureau [<http://www.census.gov>] (under Lookup Access [<http://www.census.gov/cdrom/lookup>]: Summary Tape File 1).

The problem is to find appropriate number of intervals for quantitative value more. This problem was firstly introduced in [33] where the authors pointed out that if too many intervals are defined for a attributes, rules based on this attributes might not hit minimum support thresholds. Furthermore, too large an interval results in confidence thresholds not being met.

In order to select the number of intervals, several experiments have been carried out with tree, four and five intervals per attribute. Table 1 shows the results obtained by Apriori and Eclat with these intervals, where $\#I$ stands for the number of the discovered frequent itemsets, $\#R$ for the number of the generated association rules, Av_{Sup} and Av_{Conf} for, respectively, the average support and the average confidence of the rules found, Av_{Amp} for the average length of the rules in terms of attributes involved in the antecedent, and $\%Tran$ for the percentage of transactions covered by the rules on the total transactions in the database.

Table 1. Results for Apriori and Eclat with tree, four and five intervals per attribute

| Method | Stulong | | | | | | House_16H | | | | | |
|-----------|---------|----|------------|-------------|------------|----------|-----------|---------|------------|-------------|------------|----------|
| | #I | #R | Av_{Sup} | Av_{Conf} | Av_{Amp} | $\%Tran$ | #I | #R | Av_{Sup} | Av_{Conf} | Av_{Amp} | $\%Tran$ |
| Apriori-3 | 73 | 99 | 0.35 | 0.92 | 2.32 | 100 | 298573 | 2175300 | 0.35 | 0.98 | 6.71 | 100 |
| Apriori-4 | 77 | 89 | 0.30 | 0.92 | 1.93 | 99.85 | 305229 | 1982211 | 0.22 | 0.96 | 7.00 | 100 |
| Apriori-5 | 90 | 44 | 0.24 | 0.99 | 2.11 | 100 | 208078 | 1214084 | 0.19 | 0.95 | 7.30 | 100 |
| Eclat-3 | 73 | 99 | 0.35 | 0.92 | 2.32 | 100 | 298573 | 2175300 | 0.35 | 0.98 | 6.71 | 100 |
| Eclat-4 | 77 | 89 | 0.30 | 0.92 | 1.93 | 99.85 | 305229 | 1982211 | 0.22 | 0.96 | 7.00 | 100 |
| Eclat-5 | 90 | 44 | 0.24 | 0.99 | 2.11 | 100 | 208078 | 1214084 | 0.19 | 0.95 | 7.30 | 100 |

Analysing the results obtained in *House_16H* we can see that $\#I$, $\#R$, Av_{Sup} and Av_{Conf} decrease along with the increase of the number of intervals. On the other hand, the number of rules that can satisfy the confidence threshold decreases quickly with the increase of the number of intervals in *Stulong*. So, a good selection could be to use four intervals per attribute in both databases. In the next experiments, we will use a uniform partition with four intervals for each quantitative attribute present in the databases if needed by the method.

4.4. Stulong database analysis

The results obtained in this problem by the analyzed methods are shown in Table 2 (this kind of table was described in the previous subsection). Analysing the results presented in Table 2, we can present the following conclusions:

- Genetic association rule extraction methods let us obtain a reduced set of association rules involving few attributes in their antecedent, giving the advantage of easier understanding from a user's perspective. Notice that the GENAR algorithm generates rules of the maximal length due to the fact that it always involves all the attributes (without the attribute of the consequent) in the database.

Table 2. Results for the database Stulong

| <i>Algorithm</i> | <i>#I</i> | <i>#R</i> | <i>Av_{Sup}</i> | <i>Av_{Conf}</i> | <i>Av_{Amp}</i> | <i>%Tran</i> |
|------------------|-----------|-----------|-------------------------|--------------------------|-------------------------|--------------|
| Apriori | 77 | 89 | 0.30 | 0.92 | 3 | 99.85 |
| Eclat | 77 | 89 | 0.30 | 0.92 | 3 | 99.85 |
| EARMGA | - | 30 | 0.32 | 1.00 | 1 | 100 |
| GAR | 30 | 78 | 0.61 | 0.94 | 2 | 99.93 |
| GENAR | - | 30 | 0.88 | 0.98 | 4 | 96.19 |

- The rules returned by the genetic association rule extraction methods achieve high average confidences and a good coverage of the records (100% with the ERMGA algorithm with only 30 rules), providing the user with high quality rules.

An example of quantitative association rule mined out by the analyzed methods for this problem is:

If **Height** is [160.0, 196.0] and
Weight is [57.0, 110.0] and
Cholesterol Level is [185.0, 280.0]
then **Diastolic Blood Pressure** is [65.0, 110.0]
Factor of confidence: 0.93

4.5. House_16H database analysis

The results obtained in this problem by the analyzed methods are shown in Table 3 (this kind of table was described in the previous subsection). Analysing the results presented in Table 3, we can stress the following facts:

Table 3. Results for the database HH

| <i>Algorithm</i> | <i>#I</i> | <i>#R</i> | <i>Av_{Sup}</i> | <i>Av_{Conf}</i> | <i>Av_{Amp}</i> | <i>%Tran</i> |
|------------------|-----------|------------|-------------------------|--------------------------|-------------------------|--------------|
| Apriori | 305229 | 1982211 | 0.22 | 0.96 | 7 | 100 |
| Eclat | 305229 | 1982211 | 0.22 | 0.96 | 7 | 100 |
| EARMGA | - | 100 | 0.18 | 1.00 | 3 | 100 |
| GAR | 100 | 235 | 0.37 | 0.95 | 2 | 99.98 |
| GENAR | - | 100 | 0.46 | 0.99 | 16 | 88.60 |

- In this database with more attributes, the classical methods return a large set of association rules (approximately 2 million) with minimum support and confidence. By contrast, genetic association

rule extraction methods mine again a reduced set of association rules involving only a few attributes in their antecedent, giving the advantage of easier understanding from a user's perspective. Notice that the GENAR algorithm considers rules of the maximal length due to the fact that it always involves all the attributes in the database.

- The rules obtained by the genetic association rule extraction methods maintain a good coverage of the database and a high average confidence, denoting more interesting patterns. For instance, the EARMGA algorithm mines 100 rules with the best average confidence.

An example of a quantitative association rule mined out by the analyzed methods for this problem is:

If **Price** is [14292, 84718]
 then **Percentage of males** is [0.43, 0.54]
 Factor of confidence: 0.94

4.6. Analysis of scalability

Several experiments have also been carried out to analyse the scalability of the methods in the problem House_16H. The methods were implemented in Java and all of the experiments were performed using a Pentium Corel 2 Quad, 2.5GHz CPU with 4Gb of memory and running Linux. Figures 1 and 2 show the relationship between the runtime and the number of transactions and attributes, respectively. It can easily be seen from Figure 1 how the runtime of the classical methods increases compared to the runtime of the genetic association rule extraction methods as we increase the size of the problem.

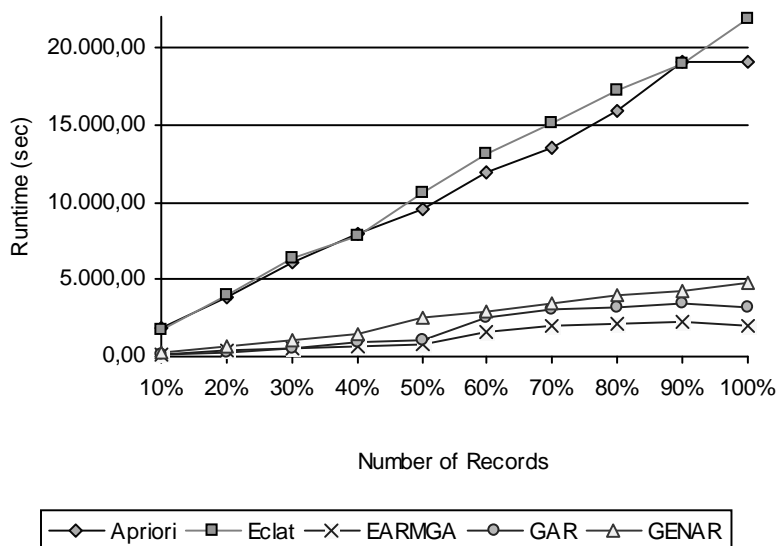


Figure 1. Relationship between the runtime and the number of transactions with all the attributes

In Figure 2, we can see that the classical algorithms expend a large amount of time mining the association rules when the number of its attributes is high. By contrast, the results plotted in these figures show that the genetic association rule extraction methods scale quite linearly for the database used in the experiment. Notice that the EARMGA algorithm obtains the best runtimes, increasing a little its runtime as we increase the size of the problem.

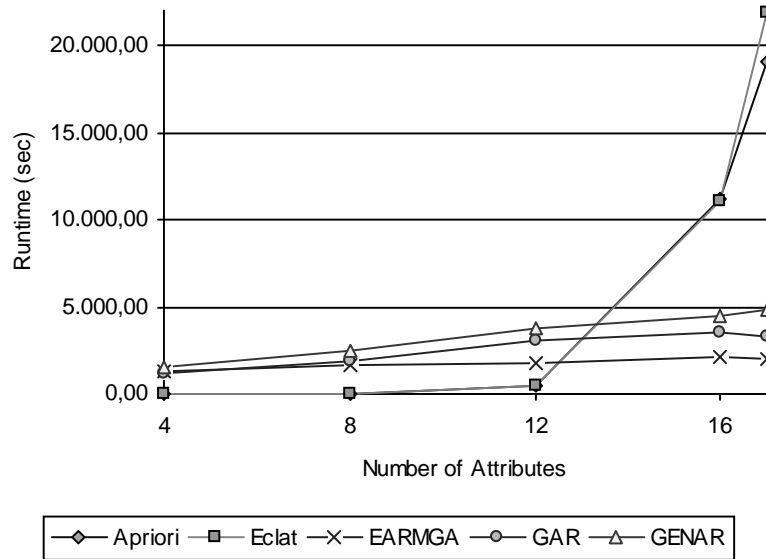


Figure 2. Relationship between the runtime and the number of attributes with the 100% of transactions

Finally, it is worthwhile to remark that the genetic association rule extraction methods expended a reasonable amount of time mining a reduced set of high quality association rules. Nevertheless, the runtimes of the genetic association rule extraction methods increase when the population size increases and, these runtimes could eventually be higher than those of Apriori and Eclat.

5. Concluding Remarks

In this paper, we have presented a study of the genetic extraction of quantitative association rules by means of GAs. By evaluating the results over two real-world databases, we can make the following conclusions about the effectiveness of these methods:

- The association rules obtained by genetic association rule extraction methods maintain a high confidence and a good coverage of the database, providing the user with high quality rules.
- Genetic association rule extraction methods let us obtain a reduced set of association rules, although the number of rules is restricted by the population size. Moreover, these rules consider few attributes in the antecedent, giving the advantage of easier understanding from a user’s perspective.
- The runtime of the genetic association rule extraction methods scales quite linearly when we increase the size of the problem.

In the future, we will attempt to extend this study to include fuzzy DM algorithms for extracting fuzzy association rules [3, 8, 20, 24].

References

- [1] Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases, *SIGMOD*, Washington D.C. (USA), May 1993.
- [2] Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules, *International Conference on Very Large Data Bases*, Santiago de Chile (Chile), September 1994.
- [3] Alcalá-Fdez, J., Alcalá, R., Gacto, M. J., Herrera, F.: Learning the membership function contexts for mining fuzzy association rules by using genetic algorithms, *Fuzzy Sets and Systems*, **160**(7), 2009, 905–921.
- [4] Bacardit, J., Garrell, J.-M.: Evolving Multiple Discretizations with Adaptive Intervals for a Pittsburgh Rule-Based Learning Classifier System, in: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)* (E. Cantu-Paz, J. Foster, Eds.), vol. 2724 of *Lecture Notes in Computer Science*, Springer, Heidelberg, 2003, 1818–1831.
- [5] Bacardit, J., Goldberg, D.-E., Butz, M.-V.: Improving the Performance of a Pittsburgh Learning Classifier System Using a Default Rule, in: *Learning Classifier Systems* (T. Kovacs, X. Llor, K. Takadama, P. L. Lanzi, W. Stolzmann, S. W. Wilson, Eds.), vol. 4399 of *Lecture Notes in Computer Science*, Springer, Heidelberg, 2007, 291–307.
- [6] Bodon, F.: A trie-based APRIORI implementation for mining frequent item sequences, *1st International Workshop on Open Source Data Mining: Frequent Pattern Mining Implementations*, Chicago, Illinois, USA, August 2005.
- [7] Borgelt, C.: Efficient Implementations of Apriori and Eclat, in: *Workshop on Frequent Itemset Mining Implementations*, vol. 90, CEUR Workshop Proc., Florida, USA, 2003, 280–296.
- [8] Chen, C.-H., Hong, T.-P., Tseng, V. S., Lee, Y.-C.: A genetic-fuzzy mining approach for items with multiple minimum supports, *Soft Computing*, **13**(5), 2008, 521–533.
- [9] Cordon, O., Herrera, F., Hoffmann, F., Magdalena, L.: *GENETIC FUZZY SYSTEMS. Evolutionary tuning and learning of fuzzy knowledge bases. Advances in Fuzzy Systems - Applications and Theory*, World Scientific, 2001.
- [10] Eiben, A. E., Smith, J. E.: *Introduction to Evolutionary Computing*, SpringerVerlag, 2003.
- [11] Freitas, A. A.: *Data Mining and Knowledge Discovery with Evolutionary Algorithms*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.
- [12] Goldberg, D. E.: *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
- [13] Greene, D., Smith, S.: Competition-based induction of decision models from examples, *Machine Learning*, **13**(2–3), 1993, 229–257.
- [14] Grefenstette, J. J.: *Genetic Algorithms for Machine Learning*, Kluwer Academic Publishers, Norwell, MA, USA, 1994.
- [15] Han, J., Kamber, M.: *Data Mining: Concepts and Techniques. Second Edition*, Morgan Kaufmann, San Francisco, 2006.
- [16] Han, J., Pei, J., Yin, Y., Mao, R.: Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach, *Data Mining and Knowledge Discovery*, **8**(1), 2004, 53–87.

- [17] Herrera, F.: Genetic Fuzzy Systems: Taxonomy, Current Research Trends and Prospects, *Evolutionary Intelligence*, **1**(1), 2008, 27–46.
- [18] Holland, J., Reitman, J.: Cognitive systems based on adaptive algorithms, in: *Pattern-directed inference systems* (D. Waterman, F. Hayes-Roth, Eds.), Academic Press, London, 1978, 1148–1158.
- [19] Holland, J. H.: *Adaptation in natural and artificial systems*, MIT Press, Cambridge, MA, USA, 1975.
- [20] Hong, T.-P., Chen, C.-H., Lee, Y.-C., , Wu, Y.-L.: Genetic-Fuzzy Data Mining With Divide-and-Conquer Strategy, *IEEE Transactions on Evolutionary Computation*, **12**(2), 2008, 252–265.
- [21] Jain, L. C., Ghosh, A.: *Evolutionary Computation in Data Mining (Studies in Fuzziness and Soft Computing)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.
- [22] Janssens, D., Brijs, T., Vanhoof, K., Wets, G.: Evaluating the performance of cost-based discretization versus entropy- and error-based discretization, *Computers and Operations Research*, **33**(11), 2006, 3107–3123.
- [23] Jong, K. D., Spears, W., Gordon, D.: Using genetic algorithms for concept learning, *Machine Learning*, **13**(2–3), 1993, 161–188.
- [24] Kaya, M., Alhajj, R.: Utilizing Genetic Algorithms to Optimize Membership Functions for Fuzzy Weighted Association Rules Mining, *Applied Intelligence*, **24**(1), 2006, 7–15.
- [25] Lee, C.-H.: A Hellinger-based discretization method for numerical attributes in classification learning, *Knowledge-Based Systems*, **20**(4), 2007, 419–425.
- [26] Liu, H., Hussain, F., Tan, C., Dash, M.: Discretization: An Enabling Technique, *Data Mining and Knowledge Discovery*, **6**(4), 2002, 393–423.
- [27] Mata, J., Alvarez, J., Riquelme, J.: Discovering Numeric Association Rules via Evolutionary Algorithm, in: *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)* (G. Rawlin, Ed.), vol. 2336 of *Lecture Notes in Computer Science*, Springer, Heidelberg, 1991, 40–51.
- [28] Mata, J., Alvarez, J., Riquelme, J.: Mining Numeric Association Rules with Genetic Algorithms, *5th International Conference on Artificial Neural Networks and Genetic Algorithms*, Taipei, Taiwan, April 2001.
- [29] Mata, J., Alvarez, J., Riquelme, J.: An Evolutionary Algorithm to Discover Numeric Association Rules, *ACM Symposium on Applied Computing*, Madrid, Spain, March 2002.
- [30] Mladenic, D., Lavrac, N., Bohanec, M., Moyle, S., Eds.: *Data Mining and Decision Support: Integration and Collaboration*, Kluwer Academic Publishers, Norwell, MA, USA, 2002.
- [31] Pal, S. K., Wang, P. P., Eds.: *Genetic Algorithms for Pattern Recognition*, CRC Press, Inc., Boca Raton, FL, USA, 1996.
- [32] Smith, S.: *A learning system based on genetic algorithms*, Ph.D. Thesis, University of Pittsburgh, 1980.
- [33] Srikant, R., Agrawal, R.: Mining Quantitative Association Rules in Large Relational Tables, *ACM SIGMOD International Conference on Management of Data*, Montreal, Quebec, Canada, June 1996.
- [34] T-Kovacs: *Strength or accuracy: credit assignment in learning classifier systems*, Springer-Verlag, Berlin, 2004.
- [35] Tsai, C.-J., Lee, C.-I., Yang, W.-P.: A discretization algorithm based on Class-Attribute Contingency Coefficient, *Information Science*, **178**(3), 2008, 714–731.
- [36] Tsumoto, S., Matsuoka, K., Yokoyama, S.: Risk Mining for Infection Control, in: *Communications and Discoveries from Multidisciplinary Data* (S. Iwata, Y. Ohsawa, S. Tsumoto, N. Zhong, Y. Shi, L. Magnani, Eds.), vol. 123 of *Studies in Computational Intelligence*, Springer, Heidelberg, 2008, 283–297.

- [37] Venturini, G.: SIA: a supervised inductive algorithm with genetic search for learning attribute based concepts, in: *Proceedings of the European Conference on Machine Learning*, vol. 667 of *Lecture Notes in Computer Science*, Vienna, Austria, April 1993, 280–296.
- [38] Wong, M. L., Leung, K. S.: *Data Mining Using Grammar-Based Genetic Programming and Applications*, Kluwer Academic Publishers, Norwell, MA, USA, 2000.
- [39] Yan, X., Zhang, C., Zhang, S.: ARMGA: Identifying Interesting Association Rules with Genetic Algorithms, *Applied Artificial Intelligence*, **19**(7), 2005, 677–689.
- [40] Yan, X., Zhang, C., Zhang, S.: Genetic algorithm-based strategy for identifying association rules without specifying actual minimum support, *Expert Systems with Applications*, **36**(2), 2009, 3066–3076.
- [41] Zaki, M. J.: Scalable Algorithms for Association Mining, *IEEE Transactions on Knowledge and Data Engineering*, **12**(3), 2000, 372–390.
- [42] Zhang, C., Zhang, S.: *Association Rule Mining: Models and Algorithms Series*, Lecture Notes in Computer Science, LNAI 2307, Springer-Verlag, Berlin, 2002.