IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Special Issue, 4, Feb 2019 E-ISSN: 2347-2693

Parameter-Free Algorithm for Mining Rare Association Rules

S. Selvarani[1*], M. Jeyakarthic[2]

[1]PG Department of Computer Science,Government Arts College,C.Mutlur, Chidambaram.Tamil Nadu, India
[2]Department of Computer and Information Science,Annamalai University, AnnamalaiNagar,Tamil Nadu, India

Corresponding Author: samyselvaa@gmail.com

Available online at: www.ijcseonline.org

***Abstract*—**This paper exhibits a Parameter-Free grammar guided genetic programming algorithm for mining rare association rules. This algorithm utilizes a context-free grammar to represent individuals, encoding the solutions in a tree-shape conformant to the grammar, so they are more expressive and flexible. The algorithm here introduced has the advantages of utilizing evolutionary algorithms for mining rare association rules, and it also additionally takes care of the issue of tuning the tremendous number of parameters required by these algorithms. The principle highlight of this algorithm is the small number of parameters required, providing the possibility of discovering rare association rules in an easy way for non-expert users. We compare our approach to existing evolutionary and exhaustive search algorithms, obtaining important results and overcoming the drawbacks of both exhaustive search and evolutionary algorithms. The experimental stage reveals that this approach discovers infrequent and reliable rules without a parameter tuning.

Keywords—Genetic Programming, Association Rules, Free Parameters, Data Mining.

I. INTRODUCTION

Association rule mining (ARM), which is considered as an imperative territory of data mining, has received more and more consideration since it was defined by Agrawal et al [1]. ARM was considered as an unsupervised learning task, having a descriptive nature and searching for strong relationships among items that are clearly covered up in large datasets.

An association rule could be defined as a ramifications of the form IF antecedent THEN consequent, both the antecedent and consequent being disjoint sets, i.e., they have no items in common. The meaning of an association rule is that if all the items in the antecedent exist in a transaction, then it is very plausible that all the items in the consequent are also in the transaction [2].

First approaches for mining association rules are based on an exhaustive search methodology, trying to obtain rare and reliable association rules, i.e., those accurate rules that show up in a high percentage in the dataset. In the exhaustive search algorithms, the process of mining association rules is divided in two steps, firstly mining rare patterns, and afterward extracting as many reliable association rules as possible. In these first proposals, the mining process is frustrated because of these two steps, which require a lot of computational time and lot of memory.

With the developing enthusiasm for the storage of information, more and more amounts of memory are required, so the mining process is hampered. Moreover, this growing interest in the storage of information is offering ascend to the utilization of real-world datasets containing numerical values. In ARM, the utilization of numerical datasets is not a trivial issue, expanding the search space since numerical domains typically contain many distinct values. In such circumstances, exhaustive search ARM algorithms cannot specifically manage numerical domains as they turn out to be not really viable.

Evolutionary algorithms (EA) were proposed to conquer the high computational time and the memory requirements in the ARM field. The utilization of evolutionary methodologies also permits of mining numeric association rules, defining interval rules and taking care of the issue of increasing the search space. First evolutionary algorithms for mining associaiton rules were depended on genetic algorithms [3], looking up to the exhaustive search approaches computational and memory requirements.

As of late, this unsupervised learning task was studied by utilizing a grammar-guided genetic programming (G3P) approach [4]. This algorithm, called G3PARM [5], was introduced as the first G3P proposal for mining association rules, giving great results and defeating those drawbacks of current ARM algorithms in terms of execution time, the

mining of numerical domains, and solution complexity. The fundamental feature of utilizing G3P is that the solutions are represented in a tree form having variable-length hierarchical structures, where the size, shape and structural complexity are not constrained a priori. G3PARM was initially depicted as a completely configurable algorithm, where a number of input parameters were required, e.g., support and confidence thresholds, crossover and mutation probabilities, the number of generations, the population size, the number of rules to be mined, etc. The same occurs when utilizing any of the existing evolutionary algorithms in this field, which are profoundly recommended in many situations, particularly when they are utilized by qualified data miners. By and by, the possibility of mining association rules utilizing a high-performance algorithm without the need of specifying a large number of parameters is a requirement for non-expert users. This paper proposes a G3P approach for mining rare association rules without requiring as many parameters as

$G = (\Sigma_N, \Sigma_T, P, S)$ with:
S = Rule
Σ_N = {Rule, Antecedent, Consequent,Comparison }
Σ_T = {'AND', '=', 'IN', 'Attribute categorical value', 'Attribute numerical value' }
P = {Rule = Antecedent, Consequent;
Antecedent = Comparison ;
Consequent = Comparison ;
Comparison = 'AND', Comparison, Comparison ;
Comparison = '=', 'Attribute categorical value' ;
Comparison = 'IN', 'Attribute numerical value', 'Attribute numerical value';

Figure. 1 Context-free grammar expressed in Extended BNF notation

G3PRARM and other evolutionary algorithms do. This free parameter algorithm makes the mining process easier for non-expert users, not requiring an optimal configuration of the parameters to carry out the mining. Besides, since this approach is depends on G3P, it gives every one of the benefits of G3PRARM [6]. This paper is structured as follows: the most pertinent related work is presented in Section II; Section III portrays the model proposed as well as its main characteristics; Section IV depicts the experiments, including the datasets utilized, and discusses the results obtained; finally, in Section V, some concluding remarks are outlined.

II. RELATED WORK

In ARM, the vast majority of the existing exhaustive search proposals are based on the Apriorialgorithm[1]. Apriori accomplishes great performance by reducing the number of candidate patterns by using the anti-monotone property, which builds up that if a length-k itemset is not frequent in a dataset, none of its length-(k+1) super-itemsets can be frequent. By and by, this algorithm is not appropriated in datasets with a large number of frequent patterns caused by quite low minimum frequency thresholds. So as to overcome some of the drawbacks of Apriori, Han et al. [7] proposed the FP-Growth algorithm, which stores information about patterns in an all-encompassing prefix-tree structure. The frequent-pattern mining just need to work on the tree rather than the whole dataset. Be that as it may, this algorithm still suffer with the growth of the number of transactions and the utilization of a very low minimum support threshold, causing a huge number of association rules.

As of late, a novel exhaustive search algorithm, called Predictive-Apriori, was proposed by Scheffer[8]. In this algorithm the author proposes to maximize the expected accuracy that the association rule will have for future data. This fast algorithm finds the n best rules that maximize the accuracy, not requiring any threshold to determine whether a pattern is frequent or not. Notwithstanding the described problems of the exhaustive search algorithms, these algorithms only works on datasets with categorical values. Be that as it may, more and more data in real-world applications usually consist of numerical values, so exhaustive search algorithms cannot be utilized specifically in the extraction of association rules. A well-known solution to deal with numerical values is the discretization of the dataset, i.e., the division of their domains into intervals pursued by applying categorical values to the intervals. Nevertheless, the use of a previous discretization step is not excluded from problems, requiring to choose the correct number of intervals. For the sake of overcoming the exhaustive search problems, i.e., the large amount of memory required, the huge computational time, and the fact of dealing with numerical attributes, evolutionary algorithms were proposed for mining association rules such as QuantMiner [9].

A novel algorithm for discovering PRAR,calledApriori-Inverse,was proposed by KohandRountree [10]. Amid the search process, Apriori-Inverse keeps those items with a support value greater than a minimum support threshold but less than a maximum value. At that point, similarly to the Apriori algorithm, a set of association rules is obtained by utilizing a confidence threshold over all the possible combinations of items previously obtained. Consequently, this algorithm mines very infrequent association rules since

the support of each one is less than or equal to that of the item with minimum support.

Another RARM algorithm, called ARIMA, was first proposed by Szathmary et al.[11] as a naïve approach. . Not at all like the Apriori-Inverse algorithm, ARIMA is not restricted to calculating PRAR. This algorithm firstly mines the minimal rare itemsets (mRIs), that is, those rare itemsets whose proper subsets are frequent. In this way, at whatever point a candidate k-itemset endures the frequent k−1 subset test, but proves to be rare, it is kept as amRI. At that point,, the algorithm finds their appropriate supersets, avoiding zero itemsets, that is, those having a support of zero. Similarly to Apriori, rare association rules are generated utilizing the set of rare itemsets mined. For this final process, it is necessary to satisfy a minimum confidence threshold. Authors of the ARIMA algorithm proposed two different ways of mining rare itemsets. A first version, called Apriori-Rare, is a slightly modified version of Apriori, finding all frequent itemsets and storing the mRIs discovered. Szathmary et al.[12] also proposed a much more efficient proposal, the MRG-Exp algorithm, which reduces the search space by avoiding exploring all frequent itemsets. Rather, it is sufficient to search for frequent generators (FGs) only. An itemset X is a generator if it has no proper subset with the same support, that is, $\forall Y \subset X$, support(X)<support(Y). The Apriori-Rare requires a higher computational time since it lists all frequent itemsets before reaching the mRIs whereas MRG-Exp explores only the FGs. Both versions get association rules from the set of mRIs, the number of rules discovered being smaller than the naïve approach.

The first algorithm for mining association rules by utilizing grammar-guided genetic programming was introduced by Luna et al. [5]. This algorithm, called G3PARM, makes utilization of G3P to define expressive and justifiable individuals. These individuals are defined by using a context-free grammar that establishes the syntax constraints for each one, allowing both categorical and quantitative attributes to be defined and obtaining feasible solutions without requiring large amounts of memory. Rare-G3PARM[6], is helping rare association rules identification and separating them from noise, as well as a new genetic operator that guides the search process. In this proposal, the resulting set just involves the best rules found along the execution, and its size tends to the size specified by the data miner.In addition, this incorporates the lift measure together with support and confidence to defeat the issues in most algorithms and also G3PARM, when only the support–confidence framework is pursued.

III. G3P PARAMETER-FREE ALGORITHM

In this section, the proposed algorithm is described in depth. The fundamental idiosyncrasy of this algorithm is that it consolidates the strength of optimizing by means of evolutionary algorithms, the ability of representing rules in an expressive and flexible way thanks to G3P, and finally, it does not require a parameter tuning as evolutionary algorithms do.

A. Encoding Criterion

The algorithm presented in this paper represents the individuals by a genotype and a phenotype. The former is defined by means of a tree structure, having different shapes and sizes, conformant to a context-free grammar G (see Figure 1). The latter permits of representing the meaning of the tree structure, i.e., the phenotype represent a rule having an antecedent and a consequent. A context-free grammar could be formally defined as a four-tuple (ΣN, ΣT, P, S), ΣN being the non-terminal symbol alphabet, ΣT denoting the terminal symbol alphabet, P representing for the set of production rules, S for the start symbol, and ΣN and ΣT indicating disjoint sets, i.e., $\Sigma N \cap \Sigma T = \emptyset$. Any production rule follows the format $\alpha \rightarrow \beta$ where $\alpha \rightarrow \Sigma N$, and $\beta \varepsilon \{\Sigma T \cup \Sigma N\}^*$. Starting from the start symbol S, each individual is represented in a derivation syntax-tree as a sentence conformant to the grammar. To acquire individuals, a series of production rules is applied from the set P. This process begins from the start symbol Rule, which dependably has a child node representing the antecedent and the consequent of the rule. Once a grammar is defined either to describe valid expressions or to impose restrictions to the search space, it is important to validate this grammar. Along these lines, considering the grammar defined in this problem and depicted in Figure 1, the following language is obtained: L(grammar)={ (AND Condition)n Condition → (ANDCondition)mCondition : n ≥0m ≥0}. In this manner, the grammar is well-defined and structured since any rule having at least one condition in the antecedent and consequent is obtained. Notice that the antecedent and consequent are disjoint sets, i.e., they have no items in common. Utilizing this grammar it is possible to mine any association rule containing either numerical or nominal features. Numerical attributes are utilized by applying the operator IN, and arbitrarily choosing two feasible values.

B. The G3P Algorithm

The EA proposed pursues a generational schema, as depicted in Figure 2. It begins by generating an initial set of individuals. The size of this population relies upon the number of rules to be mined. These initial individuals are

originated conformant to the grammar and they have a prefixed length, depending on the type of rules desired by the data miner. The algorithm brings the possibility of searching for rules having a different number of conditions, showing the minimum and maximum number of conditions available in each association rule. The goal of this algorithm is to return the best n Rare Association Rules discovered along the evolutionary process. To do as such, a pool of individuals with a predefined size of n is utilized, this pool working as an elitist population to maintain the best n rules throughout the generations. In each generation, this elitist population is updated with the best individuals, i.e., the individuals are ranked according to their fitness function esteems, and the best n individuals are kept for new generations. For generating new individuals in each generation of the evolutionary process two genetic operators are applied, which are appropriately described in subsequent sections. These two genetic operators (crossover and mutation) are used based on certain probabilities.

Most evolutionary algorithms require fixed values, so the optimal probability values are determined by the data miner based on the dataset utilized. A noteworthy feature of the G3P free-parameter algorithm introduced in this paper is its ability to update the genetic operator probabilities, not requiring any previous study of the parameters to acquire the optimal results. In the initial generation, both operators have initial values, and in subsequent generations, these probabilities are updated according to the average fitness esteem acquired in the elite population previously mentioned. The algorithm proposed continues its iterative process without requiring a maximum number of generations as many evolutionary algorithms do. In the algorithm depicted in this paper, the evolutionary process is completed if the elite population improves with the pass of the generations, estimating this enhancement by utilizing the average fitness function values of the n best rules. At long last, if the rules mined do not improve in 20 generations, then the algorithm finishes and the best rules discovered in the evolutionary process are given to the data miner. So as to optimize those numerical intervals wanted by the data miner, the algorithm brings the likelihood of carryingout a post-processing step. In this final step, a subset of rules mined by the algorithm could be chosen, and the intervals could be optimized.

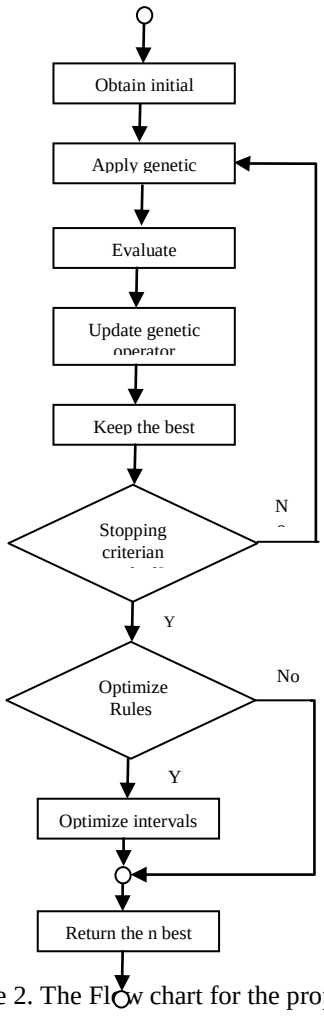

Figure 2. The Flow chart for the proposed algorithm

C. Genetic Operators

To acquire new individuals in every generation of the evolutionary process, the proposal portrayed in this paper uses two genetic operators: crossover and mutation. Crossover: This genetic operator swaps a randomly selected condition in one parent for another randomly selected condition in the other parent. Consequently, the objective of this genetic operator is to obtain new individuals having the genotype of the parents. For a superior comprehension, the pseudocodeof this genetic operatoris shown in Algorithm1.

Require: parents
Ensure: offsprings
1: offsprings ← Ø
2: for all individuals in parents do
3: ind1,ind2 ← getIndividuals(parents)
4: if random() <crossoverProbability then
5: cond1 ← getRandomCondition(ind1)
6: cond2 ← getRandomCondition(ind2)
7: newInd1 ← exchange(ind1,cond1,cond2)
8: offsprings ← offsprings ∪ newInd1


```
9: newInd2 ← exchange(ind2,cond1,cond2)
10: offsprings ← offsprings ∪ newInd2
11: end if
12: end for
13: return offsprings
```

Algorithm 1 Crossover operator

Mutation: The main objective of this genetic operator is to keep up the diversity in the population. In such a way, a randomly chosen condition in an individual is mutated to obtain a new one. Algorithm 2 demonstrates the pseudocode of this genetic operator.

```
Require: parents
Ensure: offsprings
1: offsprings ←∅
2: for all individuals in parents do
3: ind ← getIndividual(parents)
4: if random() <mutationProbability then
5: cond ← getRandomCondition(ind)
6: newCond ← newCondition()
7: newInd ← exchange(ind,cond, newCond)
8: offsprings ← offsprings∪newInd
 9: end if
10: end for
11: return offsprings
```

Algorithm 2 Mutation operator

A noteworthy feature of this G3P algorithm is its ability to refresh both genetic operator probabilities. In the first generation both probabilities have the initial value of 0.5. In subsequent generations, these probabilities are refreshed depending on if a higher diversity or a higher convergence is required. The algorithm calculates the average fitness value acquired from the pool, i.e., the average fitness of the elite population. . In view of this average esteem, it increases the population diversity by investigating the search space or it reduces the diversity by exploiting the knowledge represented within the population. If the average fitness esteem acquired from the algorithm is improving along the generations, then the exploration should gradually change into exploitation by enhancing the crossover probability and decreasing the mutation probability.

Despite what might be expected, if the fitness esteem acquired is not enhancing along the generations, then a higher exploration is required by decreasing the crossover probability and increasing the mutation probability. It is fascinating to take note of that in the earliest generations, both probabilities increase and decrease while the optimal average fitness esteem is not found. At that point, the investigation begins to be more important than the exploitation since it is more difficult to enhance the solutions, and in this manner, the optimal values are gotten. After 20 generations where the average fitness function esteem is not enhanced, the algorithm finishes returning the elitist population having the best n rules found.

D. Evaluate Individuals

The fundamental issue in any evolutionary model is the process of evaluating each individual or solution, permitting to assign a fitness function to each individual so as to decide how encouraging certain individual is, i.e., how close a given solution is to achieving the aim. Different researchers have portrayed some target measures for evaluating association rules[13]. Two of the most important and widely utilized measures in this field are support and confidence. The former is defined as the proportion of the number of transactions including the antecedent and consequent in a dataset. The latter is defined as the proportion of the number of transactions that incorporate the antecedent and consequent among all the transactions that comprise the antecedent.

In this paper, we additionally propose the utilization of a third measure to bring more effective information to the data mining task. This third measure is lift, which serves to calculate how many times more often the antecedent and consequent are related in a dataset than would be expected if they were statistically independent. The lift measure is calculated as the confidence of the rule divided by the consequent support. Contrary to Apriori-based algorithms, the EA portrayed in this paper does not require two phases to mine rules. In this algorithm, each rule is assessed by a fitness function, which could be established by the data miner among the three measures previously mentioned. In such a way, it is conceivable to pick the way the algorithm searches for rules, establishing an order of priority. In this paper, we determine that the first measure is support, the second one is the confidence measure and the last one is the lift measure. If the rules have the same support value, then the confidence is utilized to determine which one is better, and so on.

E. Optimizing Intervals

Once the algorithm has finished, it is conceivable to run a post-processing algorithm whose point is the optimization of those intervals desired by the data miner. The minimum and maximum bound of the interims are set up by the data miner, so the point of this final step is to run a local search model to optimize the rule within the interval defined by the expert. This post-processing algorithm pursues the notable hill climbing optimization technique, which is an iterative algorithm that begins with an arbitrary solution to a problem and it endeavors to find a better solution by incrementally changing a single element of the solution. If the change produces a better solution, an incremental change is made to

the new solution, repeating until no further enhancements can be found.

IV. EXPERIMENTAL STUDY

In this section, a complete analysis of the effectiveness of this proposal compared to other existing proposals for mining rare association rules is done. JCLEC[14], a Java library for the evolutionary computation, was utilized in the proposal presented in this paper. In the experimentation stage and in order to analyse the performance of our proposal, a series of executions were performed using different algorithms and different datasets. The results acquired by each evolutionary algorithm are the average results acquired running each one five times using different seeds each time. G3PRARM was originally compared with other exhaustive search (Apriori-Inverse and ARIMA), providing excellent results. At a significance level of p = 0.01, G3PRARM was significantly not the same as exhaustive search algorithms in support and confidence measures. Concentrating on evolutionary algorithms and the support measure, at a significance level of p = 0.05, G3PRARM was significantly different from the other algorithms. Along these lines, for reasons of curtness, only the G3PRARM evolutionary algorithm and the Predictive Apriori algorithms are utilized in this experimental stage.

The datasets utilized in this experimental stage are: computer hardware dataset (Cpu) having 209 instances and 9 numerical attributes, diabetes dataset (Diab) having 768 instances and 9 numerical attributes, and finally glass dataset (Glass) having 214 instances and 10 numerical attributes. Concentrating on the optimal parameters of the evolutionary algorithms, the best results for G3PRARM are those given by the authors, i.e., a population size of 50 individuals, 100 generations, 70% crossover probability, 14% mutation probability,a maximum derivation number of 24, an external population of size 20, a 90% external confidence threshold and a 70% external support threshold. With respect to the Predictive Apriori algorithm and the proposed algorithm, see that they do not need any parameter, only the number of rules to be mined is required. At long last, and in order to carry out a fair comparison, the notEqual operator is not bear in mind neither by the G3PRARM algorithm nor by the algorithm proposed in this paper, since the Predictive Apriori algorithm does not utilize this operator, which permits of mining rules having a higher support as depicted. One of the problem of the G3PRARM algorithm is that it requires support and confidence thresholds, so a previous knowledge of the data distribution is required. In some cases, the fact of utilizing high thresholds gives rise to a small set of rules discovered. Table I demonstrates the number of rules discovered by the algorithms using the configuration parameters recently portrayed; where Cpu−N, Diab−N and Glass−N represent the datasets discretized in N intervals.

Table I NUMBER OF RULES DISCOVERED BY THE ALGORITHMS

Dataset	Predictive-Apriori	G3PRARM	Proposal
Cpu-5	20	8.6	20
Cpu-10	20	1.0	20
Cpu-15	20	0.0	20
Diab-5	20	0.0	20
Diab-10	20	0.0	20
Diab-15	20	0.0	20
Glass-5	20	3.0	20
Glass-10	20	0.0	20
Glass-15	20	0.0	20

In this experimental stage, it is analysed which fitness function behaves better in terms of average confidence, average lift and average number of rules. Focusing on the confidence and lift measures, a fitness function is better than another if it obtains higher values for these measures. As for the average number of rules discovered, the best fitness function will be the one that obtains a number of rules closer to the maximum previously established by the data miner. In such a way, and in order to determine whether there exist significant differences among these four functions, a series of statistical tests were carried out.

If the Friedman test rejects the null-hypothesis indicating that there are signicant differences, then a Bonferroni-Dunn test is performed to reveal these differences. The results obtained in this study are shown in Table II. It should be noted that the original datasets, i.e., the datasets without any previous discretization step, cannot be used with the Predictive Apriori since it does not deal with numerical values, so the symbol "-" is included to show it. The Friedman average ranking statistics for the average lift measure distributed according to FF is 18.009, which does not belong to the critical interval [0,(FF)0.01,3,27= 4.510]. Thus, we reject the null-hypothesis that all algorithms perform equally well for this measure. In order to analyze whether there are significant differences among them, the Bonferroni-Dunn test is used, 1.171 being the critical difference value for p =0.1; 1.318 for p =0.05; and 1. for p =0.01, so there exist significant differences between the Predictive Apriori algorithm and the proposal presented in this paper at a significance level of p =0.01, the new proposal being statistically better.

Concluding the analysis, it is possible to state that despite the fact FF provides the best ranking for two out of three measures, it is interesting to discover interesting rules. Therefore, when the domains under application do not provide a clear difference between noisy, rare and frequent rules, FF is the best fitness function to be used, discovering

rules that are interesting and very reliable, and providing confidence values close to the maximum. As for the confidence and lift measure, this new approach behaves better than the G3PRARM algorithm.

Table II RESULTS OBTAINED BY THE ALGORITHMS

Support				Confidence			Lift		
Dataset	Predictive-Apriori	G3PRARM	Proposal	Predictive-Apriori	G3PRARM	Proposal	Predictive-Apriori	G3PRARM	Proposal
Cpu-5	0.321	0.234	**0.213**	0.953	**1.078**	0.913	1.504	1.004	**1.631**
Cpu-10	0.11	**0.199**	0.200	0.971	**1.000**	0.999	1.298	1.192	**1.341**
Cpu-15	0.221	0.210	**0.199**	0.975	**1.089**	0.981	**1.699**	1.091	1.432
Diab-5	0.313	0.311	**0.303**	0.956	**1.000**	0.999	1.393	1.292	**1.425**
Diab-10	0.217	**0.201**	0.211	0.996	0.989	**1.078**	1.788	1.086	**1.823**
Diab-15	0.192	0.163	**0.150**	0.992	0.879	**1.178**	1.077	1.172	**1.254**
Glass-5	0.376	**0.125**	0.243	0.897	0.934	**1.20**	**1.589**	1.085	1.189
Glass-10	0.294	0.211	**0.200**	0.986	1.001	**1.089**	1.289	1.026	**1.299**
Glass-15	0.134	**0.127**	0.132	0.976	0.999	**1.233**	**1.456**	1.252	1.455

V. CONCLUSION

In this paper, a G3P proposal for mining Rare Association Rules without requiring any parameter tuning was introduced. In this proposal, each solution is represented as a derivation syntax-tree conformant to a context-free grammar. This portrayal of the individuals gives expressiveness, flexibility, and the ability to confine the search space over any domain.

The experimental study uncovers that the new proposal behaves statistically better than the other algorithms for the support measure. Concentrating on the confidence and lift measure, it is absurd to expect to assert that there are significant differences among the three algorithms, but the new proposal acquires the best ranking for the lift measure and furthermore very reliable association rules, obtaining rules having a confidence value higher than 0.9 much of the time.

REFERENCES

[1] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in Proceedings of the 20th International Conference on Very Large Data Bases, ser. VLDB '94. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994, pp. 487–499.

[2] C. Romero, J. M. Luna, J. R. Romero, and S. Ventura, "Rmtool: A framework for discovering and evaluating association rules," Advances in Engineering Software, vol. 42, no. 8, pp. 566–576,2 011.

[3] X. Yan, C. Zhang, and S. Zhang, "Armga: Identifying interesting association rules with genetic algorithms," Applied Artificial Intelligence, vol. 1 9, no. 7, pp. 677–689,2005.

[4] R. McKay, N. Hoai, P. Whigham, Y. Shan, and M. ONeill, "Grammar-based genetic programming: a survey," Genetic Programming and Evolvable Machines, vol. 11, pp. 365–396, 2010.

[5] J. M. Luna, J. R. Romero, and S. Ventura, "Design and behavior study of a grammar-guided genetic programming algorithm for mining association rules", Knowledge and Information Systems, vol. 32, no. 1, pp. 53–76, 2012.

[6] J. M. Luna · J. R. Romero · S. Ventura " On the adaptability of G3PARM to the extraction of rare association rules", Knowledge and Information Systems, vol,38, pp 391-418, 2014.

[7] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns without candidate generation: A frequent-pattern tree approach," Data Mining and Knowledge Discovery, vol.8, pp. 53–87,2004.

[8] T.Scheffer, "Finding association rules that trade support optimally against confidence," in 5th European Conference on Principles of Data Mining and Knowledge Discovery. Springer, 2001, pp. 424–435.

[9] A. Salleb-Aouissi, C. Vrain, and C. Nortet, "Quantminer: A genetic algorithm for mining quantitative association rules," in Proceedings of the 20th International Joint Conference on Artificial Intelligence, ser. IJCAI '97, Hyberadad, India, 2007, pp. 1035–1040.

[10] Koh YS, Rountree N (2005) Finding sporadic rules using apriori-inverse. Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics), 3518:97–106

[11] [11] Szathmary L, Napoli A,Valtchev P(2007)Towards rare itemset mining.In:Proceedings of the 19th IEEE international conference on tools with artificial intelligence, ICTAI '07, Patras, Greece, pp 305–312.

[12] Szathmary L, Valtchev P, Napoli A (2010) Generating rare association rules using the minimal rare itemsets family. Int J Softw Inf 4(3):219–238.

[13] P. Tan and V. Kumar, "Interestingness measures for association patterns: A perspective," in Workshop on Postprocessing in Machine Learning and Data Mining, ser. KDD '00, New York, USA, 2000, pp. 293–313.

[14] S. Ventura, C. Romero, A. Zafra, J. A. Delgado, and C. Herv´as, "Jclec: A java framework for evolutionary computation," Soft Computing, vol. 12, no. 4, pp. 381–392, 2008.

