Assessing Different Rule Quality Measures in a Genetic Algorithm for Discovering Association Rules

Wilson Soto[1,2] and Amparo Olaya–Benavides[1]

[1] Intelligent Systems and Spatial information Research Group (SIGA)
Central University
Bogotá, Colombia
{wsotof,aolayab}@ucentral.edu.co
[2] Research Group on Algorithms and Combinatorics (ALGOS-UN)
National University of Colombia
Bogotá, Colombia
wesotof@unal.edu.co

Abstract. The genetic algorithms have seen applied in knowledge discovery and specially for discovering association rules. In this paper, we explore the use of different rule quality measures in the fitness function in a genetic algorithm for discovering association rules. Also, we present an improvement for this algorithm: (*i*) the mutation stage is calculated with a probability independent for each individual and (*ii*) the selection stage is calculated with Boltzmann selection.
The proposed version was tested with 10 different rule quality evaluation functions on 6 benchmark datasets.

Keywords. Data Mining, Knowledge Discovery, Association Rules, Genetic Algorithm.

1 Introduction

In Knowledge Discovery and Data Mining (KDD) most of the problems are represented as combinatorial optimization problems. One of the most important problems is the extraction of association rules.

The association rule technique was introduced in 1993 by Agrawal *et al.* [1] and it is particularly useful for discovery of hidden relations which might be interesting when it comes to large databases.

The genetic algorithms are important when discovering association rules because they work with global search to discover the set of items frequency and they are less complex than other algorithms often used in data mining [2, 3].

The genetic algorithms for discovery of association rules have been put into practice in real problems such as commercial databases, biology and fraud detection event sequential analysis [4].

One of the most important developed algorithms for discovery of association rules is ASGARD (Adaptive Steady state Genetic Algorithm for association Rule Discovery)[3], specially designed for bioinformatics.

In [4, 5] there are a comparison of the main algorithm techniques for the extraction of association rules as well as a comparison of some rules quality measures, such as: *Confidence*, *Piatetsky-Shapiro*, *Conviction* and *Jmeasure*.

The article is divided into seven sections. Section II basic concepts of association rules. Section III presents a description of genetic algorithms. Section IV presents the description of one of the related work to genetic algorithm for association rule discovery, followed by an extension of the algorithm. In section VI there are the experimental results and in section VII there are the conclusions of work.

2 Association rules

Association Rules Mining (ARM) consists in finding the set of all item subsets or attributes which often happen in database, it is also about extracting relations, associations, patterns or casual structures which can help to determine how an item subset influences the presence of other subsets [6].

The idea of association rules is originated since the market-basket where you want to find dependence between two items X and Y. A good example is the way "A client who wishes to buy products X_1 and X_2 will also buy product Y".

An association rule is an implication $X \rightarrow Y$, where X is the antecedent (a conjunction of conditions) and Y is the consequent (predict class). Besides, X and Y are disjoint sets of items, i.e., $X \cap Y = \emptyset$.

There are three general characteristics that discovery of rules must satisfy; to have specifically a high precision prediction, to be understandable and to be interesting [7].

A measure to predict the association rule precision $X \rightarrow Y$ is the *confidence*. This measures the reliability of inference made by the rule which is defined like:

$$\mathcal{C} = \frac{|X \cup Y|}{|X|} \tag{1}$$

Where $|X|$ is the number of examples that satisfies every condition in the antecedent X and $|X \cup Y|$ is the number of examples both of which satisfy the antecedent X and it has the class predicted by the consequent Y. But the *confidence* favors the rules overfitting the data [7]. Due to this it is necessary to determine the way a rule is applicable in dataset, such as, *support*. It is defined as:

$$\mathcal{S} = \frac{|X \cup Y|}{N} \tag{2}$$

Where, N is the total number of examples. *Support* is often used to eliminate non interesting rules.

[3] http://www.lifl.fr/ jourdan/download/asgard.html

A measure to determine a rule intestingness is to find surprisingness of an attribute based on each attribute information gain [8].

3 Genetic Algorithm

Genetic algorithms are methods based on biological mechanisms, such as, Mendel's laws and Darwin's fundamental principle of natural selection. The most important biological terminology used in a genetic algorithm is [4]:

- The chromosomes are elements on which the solutions are built (individuals).
- Population is made of chromosomes.
- Reproduction is the chromosome combination stage. Mutation and crossover are reproduction methods.
- Quality factor (fitness) is also known as performance index, it is an abstract measure to classify chromosomes.
- The evaluation function is the theoretical formula to calculate a chromosome's quality factor.

Genetic algorithms simulate the evolution process of populations. A problem is represented by individuals (also called chromosome or genotype of the genome), which create a population of solutions. The genetic changes, which are simulated on the chromosome, are performed using operators such as crossover and/or mutation. These changes are applied in order to achieve a population of solutions increasingly adapted to the problem. This adaptation is then evaluated by a quality factor called fitness.

4 Genetic Algorithm for Discovering Association Rules

Recently algorithms development for discovery of decision rules has been about increasing efficiency. That is reducing computational cost.

Association rules mining computational cost can be reduce to four ways [6]. One of this ways is reducing the number of passes over the database. The work is oriented towards this direction, using *lambdaj*[4]. *Lambdaj* is a library which makes easier the access collections manipulation without explicit loops in a pseudo-functional and static way.

Now, we present here a brief overview of each stage of a genetic algorithm.

4.1 Initial population

The individual representation is based on the *Pittsburgh* [9], approach which is a number of rules $X \rightarrow Y$ (IF X THEN Y) they are codified as a string and handled as an individual (chromosome). Every individual has a fitness value.

[4] http://code.google.com/p/lambdaj/

4.2 Fitness function

Fitness function is the need of measuring the rule quality. Thus, *confidence* and *support* are used for high precision prediction; to measure comprehensibility, a value which represents a relation between the number of rules and the number of conditions and to measure the interestingness, the attribute's information gain. A rule's prediction can be represented by a matrix, called confusion matrix (Table 1). Association rule *confidence* (Eq. 1) can be defined, based on the confusion matrix as:

$$\mathcal{C} = \frac{TP}{TP + FP} \tag{3}$$

Where TP is the number of instances which match with the rule's antecedent and consequent; FP is the instances which match only with the rule's antecedent. This measure is known as *positive predictive value* or *precision*. The *support* (Eq. 2) can also be represented in confusion matrix terms like:

$$\mathcal{S} = \frac{TP}{TP + FN + TN + FP} \tag{4}$$

Where, FN is the instances which match only with the rule's consequence.

Table 1. Confusion matrix

	Predicted class	
True class	Yes	No
Yes	*TP: True–Positive*	*FN: False–Negative*
No	*FP: False–Positive*	*TN: True–Negative*

A value ($\mathcal{K}$) which represents a number of rules and the number of these rules conditions, can be taken for the characteristic of comprehensibility [10]. It is a proportional inversely value relative to the number of conditions $\mathcal{N}(X)$ in the rule's antecedent X. Whether a rule can have at least $\mathcal{M}$ conditions the comprehensibility can be defined as:

$$\mathcal{K} = 1 - (\mathcal{N}(X)/\mathcal{M}) \tag{5}$$

It is necessary to calculate the entropy $\mathcal{H}$ for the database for the interestingness characteristic:

$$\mathcal{H}(X) = -\sum_{i=1}^{n} p(x_i) \log p(x_i) \tag{6}$$

Where, n is the number of different values in the dataset X; p_i is the values frequency i in the dataset X. Now the expected entropy is calculated by the attribute A as:

$$\mathcal{H}(X, A) = \mathcal{H}(X) - \sum_{j=1}^{m} p(a_j)\mathcal{H}(X_{a_j}) \tag{7}$$

Where, m is the attribute different values number A in X; a_j is j^{th} the possible value of A and $\mathcal{H}(X_{a_j})$ is a subset of X which contains all the items where the value of A is a_j.

The information gain $\mathcal{G}$ from the attribute A is used to calculate the rule's interestingness like:

$$\mathcal{G}(\mathcal{H}(X, A)) = \mathcal{H}(X) - \mathcal{H}(X, A) \tag{8}$$

Therefore, the interestingness rule evaluation objective is:

$$\mathcal{I} = \frac{1}{\sum_{j=1}^{k} \mathcal{G}(X, A_j)} \tag{9}$$

Finally the fitness function is:

$$\mathcal{F} = (w_1 \times \mathcal{C} \times \mathcal{S}) + (w_2 \times \mathcal{K}) + (w_3 \times \mathcal{I}) \tag{10}$$

Where, w_1, w_2 and w_3 are user-defined weights.

4.3 Selection

The proposed genetic algorithm uses tournament selection (τ). This strategy consists in choosing individuals randomly uniform from the current population to execute many tournaments, where each tournament winner is the best fitness individual.

4.4 Crossover

The proposed algorithm uses a specific type of crossover for the chosen attributes: crossover *Subset Size-Oriented Common feature Crossover Operator* (SSOCF), the advantages of this type of crossover are [4]:

- Conservation of the useful information sets
- Non-dominated set solutions better exploration
- To produce children with the same parents distribution

The common attributes are preserved by the children and the non-common attributes are inherited by i^{th} father with probability $\frac{n_i - n_c}{n_u}$, where, n_i is the number of chosen attributes from the parents, n_c is the number of common attributes among them and n_u is the number of non-shared chosen attributes.

4.5 Mutation

The mutation stage selects a number n of genes for changing (ngm), this bits are chosen randomly and changed by a non-symmetric probability. To change from 1 to 0 the probability is equal to ϕ and to change from 0 to 1 the probability is equal to 1.

5 Extension Genetic Algorithm for Discovering Association Rules

In our experiments we use the algorithm presented in the previous Section with the following updates:

5.1 Fitness function

The fitness function is for measuring the rule quality. For that reason, the fitness function can be one of the rule quality evaluation functions that are show in the Table 2.

Table 2. Description Rule Quality

Rules Quality Measures	Formula
Sensitivity × *Specificity*	$\frac{TP}{TP+FN} \times \frac{TN}{TN+FP}$
Support	$\frac{TP}{(TP+FP+TN+FN)}$
Confidence	$\frac{TP}{(TP+FP)}$
Interest	$\frac{TP}{(TP+FP)\times(TP+FN)}$
Odds Ratio	$\frac{TP\times TN}{(FP\times FN)}$
Kappa	$\frac{TP+TN-((TP+FP)\times(TP+FN)+(FP+TN)\times(TP+FP))}{1-((TP+FP)\times(TP+FN)+(FP+TN)\times(TP+FP))}$
Conviction	$\frac{(TP+FP)\times(FP+TN)}{FP}$
Piatetsky-Shapiro	$\frac{TP}{TP+FP+TN+FN} - ((TP+FP)\times(FP+TN))$
Certainty Factor	$\frac{TP+((TP+FP)-(FP+TN))}{1-(FP+TN)}$
Jaccard	$\frac{TP}{(TP+FP)+(FP+TN)-TP}$

5.2 Selection

The tournament selection has the disadvantage that if the tournament size, is not too large, tournament selection prevents the best individual from dominating,

thus having a lower selection probability. On the contrary, if is too small, the probability that bad individuals are selected increase. For that reason we use in this stage the strategy of the Boltzmann selection. Boltzmann selection is based on the thermodynamical principles of simulated annealing [11].

Otherwise, The Boltzmann selection can be used to select two individuals, *i.e.*, if

$$\mathcal{P}_{(0,1)} > \frac{1}{1 + e^{(\mathcal{F}_x - \mathcal{F}_y)/T}} \tag{11}$$

then individual y is selected; otherwise, individual x is selected. Where, $\mathcal{F}_x$ is the fitness function value for individual x, $\mathcal{F}_y$ is the fitness function value for individual y and T is the "temperature" parameter at (outer loop) iteration k, such that:

$T > 0$ for all k and $\lim_{k \to +\infty} T = 0$. The T value can be calculated as $T_{init}/\log(k)$.

5.3 Mutation

The number of genes for changing (ngm) are chosen randomly for each individual, the objective is growth the diversity within the population and avoid premature convergence.

6 Experimental results

We evaluate a number of rule quality evaluation functions in six datasets. The Table 2 describes the rules quality measures used in our experiments.

6.1 Datasets

The datasets used in the experiments are shown Table 3.

Table 3. Datasets characteristics

Name	Abbrev.	Attributes	Instances
car evaluation	*car*	6	1728
post operative patient	*pos*	8	90
teaching assistant evaluation	*tae*	5	151
tic-tac-toe	*tic*	9	958
nursery	*nur*	8	12960
zoo	*zoo*	17	101

The datasets are extracted from UCI dataset repository [12].

Only *tic-tac-toe* dataset has a binary-valued class attribute. The rest have more than two values in the domain of the class attribute.

6.2 Parameters setting

The Table 4 show the parameters fitting to make the experiments.

The parameters for each experiment are the population, number of iterations, the number of individuals involved in crossover for a given generation (crossover rate) and the mutation probability ϕ (in the change from 1 to 0).

Table 4. Parameters setting

Population	Iterations	Crossover rate	ϕ
100	100	50	0.6

Each experiment is run 10 times and the average of predictive accuracy of the generated rules is calculated to evaluate quality of the experiment.

6.3 Result analysis

The experimental results are summarized in Table 5, where, the predictive *accuracy* is presented for each a rule quality evaluation functions. The predictive *accuracy* based on the confusion matrix (Table 1) is equals:

$$\mathcal{A} = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

For the corresponding dataset the table shows the mean ($\overline{x}$) and standard deviation (s) as ($\overline{x} \pm s$). Additionally, the values in bold represent evaluation function with the highest in accuracy in rule for the corresponding data set.

Besides, in the Table 5 are included the results with the algorithm C4.5 [13] (algorithm used for generate a decision tree based rule inducer) in order to show the behaviour of the proposed algorithm compared with non-GA based systems.

The results are based in predictive accuracy, *Piatetsky-Shapiro* came at the first rank achieving 76% average accuracy, followed by *Sensitivity*×*Specificity* with 71%. The *Conviction* and *Interest* have the same average accuracy achieving 70%. The lowest average accuracy was 32% achieved by *Kappa*.

The possible relation between predictive accuracy and dataset attributes is interesting because as shown in Table 5 can be compared the results for each rule quality measure of the dataset *teaching assistant evaluation* (5 attributes) and *zoo* (17 attributes) with low average accuracy and high average accuracy, respectively.

7 Conclusions

In this paper, we have explored the use of ten rule quality measures, in the ambience of the genetic algorithms. Some of these rules have not previously been used in genetic algorithms for discovering association rules.

Table 5. Experimental results

Rule Quality Measure	*car*	*pos*	*tae*	*tic*	*nur*	*zoo*	Avg. Accuracy
Sensitivity × *Specificity*	0.62 ± 0.04	0.62 ± 0.05	0.58 ± 0.03	0.66 ± 0.03	0.86 ± 0.17	$\mathbf{0.93 \pm 0.11}$	0.71 ± 1.36
Support	0.69 ± 0.18	0.64 ± 0.21	0.52 ± 0.27	0.67 ± 0.11	0.73 ± 0.14	0.65 ± 0.15	0.65 ± 1.24
Confidence	0.50 ± 0.15	0.53 ± 0.25	0.60 ± 0.10	0.66 ± 0.03	0.75 ± 0.19	0.91 ± 0.12	0.66 ± 1.25
Interest	$\mathbf{0.72 \pm 0.06}$	0.53 ± 0.23	0.61 ± 0.08	0.69 ± 0.10	0.72 ± 0.05	0.92 ± 0.09	0.70 ± 1.33
Odds Ratio	0.63 ± 0.01	$\mathbf{0.68 \pm 0.11}$	0.41 ± 0.02	0.69 ± 0.05	0.83 ± 0.17	0.88 ± 0.18	0.69 ± 1.31
Kappa	0.43 ± 0.10	0.39 ± 0.32	0.33 ± 0.00	0.48 ± 0.05	0.41 ± 0.04	0.33 ± 0.16	0.32 ± 0.64
Conviction	0.61 ± 0.03	0.53 ± 0.09	0.59 ± 0.09	0.70 ± 0.40	$\mathbf{0.90 \pm 0.16}$	0.86 ± 0.17	0.70 ± 1.33
Piatetsky-Shapiro	0.66 ± 0.04	0.62 ± 0.18	$\mathbf{0.69 \pm 0.00}$	$\mathbf{0.77 \pm 0.07}$	0.89 ± 0.15	$\mathbf{0.93 \pm 0.11}$	0.76 ± 1.45
Certainty Factor	0.38 ± 0.09	0.41 ± 0.13	0.37 ± 0.09	0.50 ± 0.06	0.33 ± 0.44	0.21 ± 0.15	0.37 ± 0.70
Jaccard	0.65 ± 0.01	0.71 ± 0.00	0.44 ± 0.10	0.67 ± 0.11	0.76 ± 0.15	0.85 ± 0.20	0.68 ± 1.30
C4.5	0.69 ± 0.17	0.71 ± 0.05	0.62 ± 0.09	0.95 ± 0.06	0.87 ± 0.13	0.80 ± 0.12	0.77 ± 0.10

In this investigation, the aim was to assess different rule quality measures using a benchmark suite of 6 widely-used datasets. These experimental results suggest that the rule *Piatetsky-Shapiro* provides a high accuracy in the genetic algorithm used. Further, a new variant for discovering association rules based in genetic algorithms is proposed.

The approach proposed has a number of parameters, in future work, experimental investigations are needed to estimate the effects of these parameters and develop methods to set the parameters appropriately.

References

1. R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules Between Sets of Items in Large Databases," *ACM SIGMOD*, vol. 22, no. 2, pp. 207–216, 1993.
2. G. Soumadip, B. Sushanta, S. Debasree, and S. Partha, "Mining Frequent Itemsets Using Genetic Algorithm," *International Journal of Artificial Intelligence & Applications* (IJAIA), vol. 1, no. 4, pp. 133–143, 2010.
3. S. Xian-Jun, and L. Hong, "A Genetic Algorithm-Based Approach for Classification Rule Discovery," in *Proc. International Conference on Information Management, Innovation Management and Industrial Engineering (ICIII 2008)*, Taipei, Taiwan, 2008, pp. 175–178.
4. J. Laetitia, "Métaheuristiques pour l'extraction de connaissances: application a la génomique," Ph.D. Thesis, Université des sciences et technologies de Lille, France. 2003.
5. K. Salama, and A. Abdelbar, "Exploring Different Rule Quality Evaluation Functions in ACO-based Classification Algorithms," in *Proc. IEEE Symposium on Swarm Intelligence (SIS 2011)*, Paris, France, 2011. pp. 1–8.
6. S. Kotsiantis, and D. Kanellopoulos, "Association Rules Mining: A Recent Overview," *GESTS International Transactions on Computer Science and Engineering*, vol. 32, no. 1, pp. 71–82, 2006.
7. A. Freitas, *In Advances in evolutionary computing*. Ed. Springer Verlag New York, Inc., 2003, pp. 819–845.
8. A. Freitas, "On rule interesting measures," *Knowledge Based System*, vol. 12, no. 5, pp. 309–315, 1999.
9. C. Pitangui, and Z. Gerson, "Genetic Based Machine Learning: Merging Pittsburgh and Michigan, an Implicit Feature Selection Mechanism and a New Crossover Operator," in *Proc. Sixth International Conference on Hybrid Intelligent Systems (HIS 2006)*, Auckland, New Zealand, 2006, p. 58.
10. S. Dehuri, A. Ghosh, and R. Mall, "Genetic algorithm for multi-criterion classification and clustering in data mining," *International Journal of Computing and Information Sciences*, vol. 4, pp. 143–154, 2006.
11. A. P. Engelbrecht, *Computational intelligence: an introduction*. Ed. John Wiley & Sons Ltd., 2007.
12. A. Frank, and A. Asuncion. (2010) UCI Machine Learning Repository. [Online]. Available: http://archive.ics.uci.edu/ml/
13. J.R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1995.