ISSN 1870-4069

A Hybrid Approach Based on Cores-Clouds, STAR Methodology and Genetic Algorithms for Rules Extraction

Irene López Rodríguez[1], Blanca Tovar Corona[2], Blanca Alicia Rico Jiménez[3],
Laura Ivonne Garay Jiménez[4]

[1] Instituto Politécnico Nacional, UPIITA, Ciudad de Mexico,
Mexico

[2] Instituto Politécnico Nacional, UPIITA, Sistemas, Ciudad de Mexico,
Mexico

[3] Instituto Politécnico Nacional, UPIITA, Informática, Ciudad de Mexico,
Mexico

[4] Instituto Politécnico Nacional, UPIITA, SEPI, Ciudad de Mexico,
Mexico

ireneloro@hotmail.com,bltovar@ipn.mx,bricoj@ipn.mx, lgaray@ipn.mx

Abstract. The automatic extraction of rules during the search of information in a database (DB) is an important task in the discovery of knowledge, specially, when working with unstructured DB. This hybrid algorithm of unsupervised classification was based on combinatorial logic approach, conceptual clustering and genetic algorithms in order to identify relevant features and find out the semantic of the resulting classes. The algorithm was tested using a benchmark dataset which, with the immersed genetic algorithm gave β_0 values from 0.78 to 0.92 for a covering of 100% within 9 rules. The use of CC-STAR-GA algorithm allowed inferring the minimum number of rules with maximum coverage and minimum intersection between classes into the DB. Finally, the algorithm performance was compared with the results of GAJA2 and ROUGH SET and results shows that this hybrid system could be an option for rule extraction.

Keywords: Automatic rules extractor, genetic algorithms, fuzzy classification, conceptual clustering, combinatorial logical approach.

1 Introduction

Commonly, in the expert systems, the human expert is responsible for deciding which variables are relevant to establish the classification rules. In this situation, the designer of the system requires quantitative information that the expert does not usually have or is not prepared to provide in an understandable way because there are many factors that are not always consciously considered when deciding to which class it belongs. Under these conditions, the extraction of knowledge is hard to accomplish and time consuming [1].

The alternative to overcome these limitations is to free the expert of this task, through techniques that automatically search the most relevant variables from the evidence stored in databases, formulate rules that express persistent relationships in the data, analyze the information in order to discover semantic structures and, finally, extract knowledge from real and complex domains. Once the information is grouped and its structure is stablished, new patterns can be classified efficiently [2].

Some automatic classifiers, such as the neural networks or support vector machines, had showed a relevant performance with n dimensional sets of patterns in generating groups or classes based on the values associated to the variables in the databases, but they are unable to provide an explanation or comprehensible justification for the solutions they reach [3]. Therefore, decision trees are preferred in mining data because, besides the efficient classification, it is also important to discover the structure of the information regardless the type of the study domain. The main goal of these algorithms is to increase the accuracy in the classification of new patterns using previously generated rules. Nowadays, genetic algorithms are combined with basic rule extraction algorithms, such as C4.5 and SPRINT, and they show better accuracy than other decision tree algorithms, but their performance is determined by the database [4].

The aim of this paper is to present a hybrid methodology that combines a Cores-Clouds algorithm with a method based on the STAR methodology as well as the use of a genetic algorithm for optimizing the value of β_0 threshold, which is an input parameter for the Cores-Clouds algorithm [5]. These techniques had proved their efficiency separately and with actual speed and memory improvement in hardware could be a viable option.

This paper is organized as follows: In section 2 a summarized description of the algorithms which were selected for this approach of rule generation and its evaluation technique is presented, the Cores-Clouds algorithm, basic STAR methodology, the genetic algorithm application and the validation index. The proposed hybrid algorithm, CC-STAR-GA, is detailed in section 3. The experimental results on a benchmark dataset to investigate the feasibility and validity of our proposed algorithm are shown in section 4 as well as the results of the hybrid algorithm performance and comparison with other algorithms were discussed. Conclusions and future work are presented in Section 5.

2 Preliminaries

2.1 Clustering with Cores-Clouds Algorithm

Clustering is an automatic process that assembles related information together. The grouping could be restricted or free unsupervised. The main difference is that the first considers an initial sample or a proposed number of classes, and the second does not. These types of groupings are called semantic clustering and they can be used to find structures.

The Cores-Clouds algorithm is a combination of clustering by density and hierarchical methods derived from the graphical representation of a covering or grouping patterns. Its main feature is that it considers crisp groupings (cores) and then a fuzzy analysis in order to generate clusters, called clouds, which establish a membership degree of the different classes. This method is a significant support for generating classes and, consequently, for their interpretation.

The β_0 value determines the accuracy for finding classes and rules from a specific database. Once a fixed β_0 value is set, the structures (grouping) found for several grouping criteria are related and a hierarchy among the structures or groupings is generated. Then, the structures are organized from general to specific in a descendent order.

There are different types of restricted unsupervised groupings for the Cores-Clouds algorithm and, according to the criteria considered, they are classified as: β_0-connected grouping, β_0-compacted grouping, strictly β_0-compacted grouping and maximally β_0-completed grouping. Each type mentioned is calculated differently. The last two require more computational resources because of the number of combinations that are calculated in the process. The criteria of β_0-connected Cores and Clouds was used in this wok because it requires minimum computer resources.

2.1.1 $\boldsymbol{\beta_0}$-Connected Core Section

Let Ω be a universe of known objects, $\Omega = \{O_1, \ldots, O_n\}$, and f a function of the differences between the patterns associated to each object that belongs to Ω. β_0 is a difference threshold and it is a real number in the codomain of f. Therefore, two objects O_i and O_j of Ω are β_0-similar if and only if $f(o_i, o_j) \geq \beta_0$. NU, a subset of Ω, is a β_0-connected core if and only if every pair of objects in the core are β_0-similar through their patterns.

The algorithm for computing cores with β_0-connected items is as follows: (1) the matrix of difference of each pair of patterns is calculated using any known technique, in this case a Euclidian distance [6]; then, it is normalized in a range from 0 to 1. (2) For each pattern, a new matrix is calculated with all new β_0-similar patterns (NB_j); and finally, (3) patterns with intersection are joined.

2.1.2 Clouds $\boldsymbol{\beta_0}$-Connected Section

For each β_0-connected core (NU_j) obtained with the β_0-connected core algorithm, a β_0-connected cloud (NB_j) is computed using function $\pi(o_i, NB_j)$ as the fuzzy membership function of all objects to each cloud where $j = 1, \ldots, r$ clouds or subsets, and $i = 1, \ldots, n$ patterns considered in the universe (as shown in equation 1).

$$\pi(o_i, NB_j) = \begin{cases} 1, & if\ NU_j = \{o_i\}, \\ \max\limits_{\substack{o_p \in NU_j \\ o_p \neq o_i}} \{f(o_i, o_p)\}, & otherwise, \end{cases} \tag{1}$$

where o_p are patterns of the elements of the set Ω and o_i is the pattern being evaluated, f is the difference function applied. In this case, all patterns belong to the cloud but with different degree of membership unless they belong to the core. If any group has only one pattern, then this object has a degree of membership of 1; otherwise, the maximum is considered with the maximal membership defined in the previous equation.

This part of the algorithm has four steps: (1) Compute the cores, (2) Define one cloud for each core; and (3) Generate the membership function and (4) compute the membership degree for each pattern of every defined cloud.

2.2 Conceptual Clustering: Star Methodology

The idea of grouping objects into categories described by concept was introduced by Michalski in the late 70s and early 80s [7]. The conceptual clustering proposes a structured space for forming the groups, and gives meaningful information about the relevance of belonging to the same group. This means that the methodology provides the features or concepts, which are defined by the attributes of objects that make up the groups, for these clusters. The aim is to create relations from the database attributes that have a meaning to the expert in the application area [8].

There are four basic concepts in this methodology: Selector, Complex, Coverage, and Seed. Selector allows obtaining information about an object. Its syntax is (Attribute, Operator, Values), where the attribute is a characteristic belonging to the pattern, the operator could be {=, <,>, ≥, ≤} and values are continuous or discrete, depending on the domain of each attribute, for example (Color = Blue). Complex is a combination of selectors. If the object is represented by a pattern, then each attribute could set a relationship, for example: $l = \{(Color = Blue) \cup (Size = Big)\}$. If all the elements of a set satisfy the definition of complex, then it is a complex set s_1, and if s_2 is a set of a more generalized complex, then $l_1 \subset l_2 \rightarrow s_1 \subset s_2$. Similarly, a union is defined by $l_1 \cup l_2 \rightarrow s_1 \cup s_2$ and a junction is $l_1 \cap l_2 \rightarrow s_1 \cap s_2$. In contrast, a disjunction is when $s_1 \cap s_2 = \phi$. The third concept is coverage of s_1 with respect to s_2, so coverage is a disjunction of the complex set s_1 and the complex set s_2. Finally, seed is an initial value of the attribute.

The STAR methodology is an inductive method used to find expressions that distinguish a specific group of data given a set with both, elements that satisfy the complex (positive examples) and elements that do not satisfy the complex (negative examples). *E* is a set of expressions that is generated for describing all positive examples but dismisses negative examples [9].

The STAR methodology can be summarized as follows:

1. Let ***E*** be the set of complex to deliver, initially ***E*** = *{}*.
2. Let ***L*** be a list of complex to be selected, initially ***L*** = *{}*.
3. Let ***S*** be the set of the selectors associated to the seed.
4. Do while ***L*** is not empty:
   a. Create a set ***E'*** of complex created by the junction between the elements of ***L*** and ***S***.
   b. Remove from ***E'*** those elements that have been already included in ***E***.
   c. If a complex ***E'*** does not cover any negative example, add it to the complex ***E***.
   d. Update the ***L*** list with the remaining elements of set ***E'***.
5. At the end of the algorithm, a set ***E*** is presented using the defined function of the lexicographic evaluation.

The lexicographic evaluation function (LEF) is defined by a sequence of the pairs (evaluation criteria, tolerance threshold). In order to select the best rules, each cluster is evaluated with defined criteria and only those rules that meet the threshold of tolerance are retained [10]. In this case, the criteria was the maximum coverage with the minimum number of premises, so the number of positive examples covered was evaluated with a set of candidate rules and then the best rule was selected.

2.3 Genetic Algorithms

Genetic algorithms are adaptive algorithms intended to find the global optimal solution to a problem based on the emulation of the natural evolution process. The evolutionary techniques are characterized by using three basic operations (1) selection, which is in charge of selecting the individuals that will have an opportunity to reproduce and which ones will not; (2) crossover, which provides a mechanism to inherit characteristics to their offspring; and (3) mutation, which is a random deformation of the gene strings [11]. In genetic algorithms, a right balance between exploration and exploitation has to be obtained. Therefore, mutation operators are used primarily to provide exploration, and crossover operators are used to direct the population to a good solution (exploitation).

So, while the crossing attempts to converge to a specific point, the mutation does everything possible to avoid convergence in order to be able to explore more areas. Then, if the mutation rate is too high, the likelihood of seeking in more areas into the search space increases; however, this prevents population to converge to an optimal solution. On the other hand, if a mutation rate is too small, the resulted value could converge to a local optimum rather than to a global optimum [12]. In this work, the genetic algorithm was used for obtaining the threshold β_0 of the difference function in order to have a better clustering. The performance of the selection was evaluated with the fitness function named index I.

2.3.1 Fitness Function

Index I is a cluster separation measurement with internal criteria which considered the compactness of the cloud and the maximum distance between clouds. It is used in the genetic algorithm as a fitness function that quantifies how optimal a solution which considered the β_0 is. In this case, the result was the selection of an optimal chromosome whose bases will be combined; thus moving towards a new better generation of β_0.

Index I was composed by three factors that looked for a minimum number of classes K with the maximum coverage and it is defined as follows:

$$I(K) = \left(\frac{1}{K} * \frac{E_1}{E_K} * D_K\right)^p \; for\, p \in N. \quad (2)$$

In the first factor $1/K$, K was defined by the Core –Clouds clustering algorithm so if K increased, $I(K)$ decreased. The factor Ratio E_1/E_k was the effect of the function of differences f for each cluster over the distribution of the entire dataset. So, E_1 was a constant for the entire set and E_k was the measurement of the distance given by the group to a pattern o_i. If E_k decreased, then index I increased.

So E_k is defined by equation 3:

$$E_k = \sum_{i=1}^{n} \sum_{j=1}^{k} \left[f\left(o_i, h_j\right) \pi\left(o_i, C_j\right) \right], \tag{3}$$

where $f\left(o_i, h_j\right)$ is the distance obtained with the difference function from the object o_i to the centroide h_j of each cluster and $\pi\left(o_i, C_j\right)$ is the membership value of each object o_i to the cluster C_j.

The last factor, D_k was the maximum distance obtained between two different clusters, over all possible pairs obtained by their objects:

$$D_k = \max_{\substack{p,q=1\dots k \\ p\neq q}} f(h_p, h_q). \tag{4}$$

The value of p was used to define the contrast between different clustering configurations. In this application, the p value was set to 2 because we were considering a Euclidean distance [13].

3 Development of the Hybrid Algorithm CC-STAR-GA

This section describes the methodology of the algorithm's design of the hybrid algorithm CC-STAR-GA. It is a combination of core cloud algorithm and STAR methodology used in order to reduce the computational time and work for the resolution of a task. It avoids the generation of the complex list to study on each iteration, and generates all the possible features and its combinations at the beginning. The input variables for this algorithm were the dataset to study, attribute selectors set, maximum number of seeds and type of distance to be computed. The genetic algorithm also requires predefining the number of individuals, chromosome size, maximal number of iterations, and crossover and mutation thresholds. The maximal number of iterations and the variations of the fitness function through each generation worked as stop conditions for the procedure that is detailed below.

1. Creation of an initial population of individuals, β_0 value was computed with a random process.
2. Difference matrix was computed and normalized using the data set given.
3. Repeated until the stop condition was obtained:

a. Evaluation of each individual of the group using index I.

   i. For each pattern vector, β_0 - similar patterns were generated and concatenated when the patterns were intersected; thus creating cores.

   ii. A cloud was generated for each computed core and it was considered a class C_j.

   iii. Index I was computed as fitness function for the individual.

b. Selection of the best individual by means of the genetic operators. This individual was used in the next computation.

c. Generation of new individuals using crossover and mutation.

d. Replacement of the worst individuals of the population with new individuals.

Intermediate results: Cores and Clouds with best value of β_0 given by the genetic algorithm that is described in steps 1 – 3.

4. Seed number per class randomly set and restricted by the predefined maximum number.
5. For the selection of the seeds, the objects into the generated cores with higher membership degree were chosen.
6. Complex list E to be delivered was initialized.
7. Each complex was evaluated in the Cloud and Core considered. Only complexes that accomplished the characterization of the positive examples of each class were maintained in the list.

Final results: Rules were listed considering the maximum coverage and the minimum of premises.

The rule evaluation was performed considering the following concepts about association analysis of the rules of each algorithm: Let $P(X)$ be the probability of appearance of item X in a set of transactions D and let $P(Y|X)$ be the conditional probability of appearance of item Y given item X appeared in D. If $X, Y \subseteq I$, which was a set of items, then $support(X)$ was defined as the fraction of transactions $Ti \in D$ such that $X \subseteq Ti$. If $P(X) = support(X)$, the support of a rule $X \rightarrow Y$ was defined as $support(X \rightarrow Y) = P(X \cup Y)$. Then, an association rule $X \rightarrow Y$ had a measure of reliability called $confidence(X \rightarrow Y)$ which was defined as $P(Y|X) = P(X \cup Y)/P(X) = support(X \cup Y)/support(X)$.

A k-itemset with support above a minimum threshold was called frequent. We used a third significance metric for association rules called $lift(X \rightarrow Y) = P(Y|X)/P(Y) = confidence(X \rightarrow Y)/support(Y)$. Lift quantified the predictive power of $X \rightarrow Y$ [14].

4 Results and Evaluation

The main purpose of the experiments was to gain an insight into the behaviour of CC-STAR-GA. In order to set the condition of this algorithm as rule generator, it was tested with a dataset of acute inflammation information provided by UCI, which is a machine learning repository [15]. This dataset contained 120 records with 6 attributes associated to the following symptoms: body temperature, continuous need to urinate, micturition pain, urethra swelling, lumbar pain, and occurrence of nausea; as well as 2 class attributes: inflammation of urinary bladder and nephritis of renal pelvis origin. This dataset was created for the diagnosis of acute inflammations of the urinary bladder.

First, some parameters were selected by an exploratory analysis going through the possible values in a linearly and bidirectional way in order to find the appropriate rates for this specific dataset, considering the exploration and exploitation observed along the process. Second, the hybrid algorithm proposed here was applied using the entire symptom dataset in order to extract the linguistic rules and to define the range of optimal values of β_0, which were obtained by the genetic algorithm. The analysis of β_0 for this specific dataset was performed through index I and its value changes.

Third, the symptom dataset was randomly split into 10 pairs of training and testing sets. Each training set consisted of 63.2% of the dataset and each testing set consisted of the remaining 36.8%. The multiple confusion matrix was used to assess the quality of this structure considering the classes obtained through sensitivity, specificity, positive prediction value (PPV), negative prediction value (NPV), detection rate, detection prevalence, and balance accuracy, using CARET, a package of RStudio [16].

Finally, rules obtained with other algorithms and the same dataset were compared with the rules generated by the CC-STAR-GA algorithm. The concept of association analysis was used to evaluate the results of the comparison.

4.1 Definition of the Optimal Range of β_0 and Pre-Setting

After the exploratory analysis of the performance of the genetic algorithm for this dataset, proper adjustments of the rate of mutation, crossover, and number of individuals in the initial population were performed. The parameters considered were (1) number of individuals = 6, (2) size of chromosome= 16, (3) maximum number of generations = 20 iterations, (4) mutation threshold = 0.001, and (5) crossing threshold 0.7. Figures 1 and 2 show the performance of the algorithm using the values obtained as explained above. It can be seen that β_0 value converged in a range of 0.78 to 0.92 after 20 generations. The procedure was repeated 10 times to confirm the results. The mean β_0 value for generation 20th was 0.856 with a SEM of 0.013.

4.2 CC-STAR-GA Application

The evaluation of the fitness function, where the minimum value of index I is associated with experiment 'P2' and 'P4'; then the best grouping for these data had an index I of 0.065. Two tendencies were observed. The first group was around the mean value 0.197, which was associated to a local minimum; and the second, with a mean value 0.066, which was considered the global minimum.

4.3 Generated Rules

Core-Cloud algorithm generated 9 classes and STAR methodology found 16 rules. They are the full set of decisive rules for each class found with the hybrid algorithm without any diagnosis available. After considering the presumptive diagnosis of two diseases of urinary systems: D1, inflammations of urinary bladder, and D2, nephritis of renal pelvis origin, this number of rules was reduced. The hybrid algorithm shows the full set of generated rules with maximum coverage and minimum premises, as an output of the program. But it is possible to further reduce the rules, selecting only one rule per class. So, Table 1 presents the reduced table of rules.

In the study case, the rules selected for this study covered completely all of the items of each class in this database even though only the rules that showed maximum coverage and fewer premises using the lexical function evaluation (LEF) were chosen. And it is possible to interpret the rules or give them a clinical meaning using the results shown in Table 1, for example, rule R5 can be transformed or interpreted as:

if LUMBAR PAIN IS NOT PRESENT and BURNING OF URETHRA is PRESENT then IT IS INFLAMMATION OF URINARY BLADDER and IT IS NOT NEPHRITIS OF RENAL PELVIS ORIGIN

The rules selected for this study covered completely all of the items of each class in this database even though only the rules that showed maximum coverage and fewer premises using the lexical function evaluation (LEF) were chosen (Table 2).

4.4 Evaluation of CC-STAR-GA Algorithm

Overall accuracy rate with a 95% confidence interval was calculated. The total accuracy was 97.53% with a confidence interval from 91.36% to 99.7%. Table 8 lists the parameters that were commonly computed from the confusion matrix. It can be seen that sensitivity remained close to 1.0 for all classes, so almost all positive examples were correctly grouped. Specificity was 1.0, except for class 2, so almost all negative cases were discarded.

If the prevalence of the classes is considered [0.07 to 0.17], then the positive and negative predictive values went from 0.87 to 1.0. On the other hand, the false positive or negative assignation considered in the detection rate had a range of 0.07-0.18. Accuracy or error rate, determined by the effectiveness of the model to cover the data, went from 0.86 to 1 along the classes proposed.

Table 1. Rules after pruning per class using CC-STAR-GA algorithm for the Acute Inflammations dataset used.

CLASSES		RULES
1	RU1	IF ((OCCURRENCE = 0) (LUMBAR = 1) (URINE = 0)) THEN D1=0, D2=0
2	RU2	IF ((LUMBAR = 0) (BURNING = 1)) THEN D1=1, D2=0
3	RU3	IF ((OCCURRENCE = 0) (MICTURITION = 1) (BURNING = 0)) THEN D1=1, D2=0
4	RU4	IF ((LUMBAR = 0) (URINE = 1) (MICTURITION = 0)) THEN D1=1, D2=0
5	RU5	IF ((MICTURITION = 0) (BURNING = 1)) THEN D1=0, D2=1
6	RU6	IF ((OCCURRENCE = 1) (BURNING = 1)) THEN D1=1, D2=1
7	RU7	IF ((OCCURRENCE = 1) (URINE = 1) (BURNING = 0)) THEN D1=1, D2=1
8	RU8	IF ((LUMBAR = 0) (URINE = 0)) THEN D1=0, D2=0
9	RU9	IF ((OCCURRENCE = 1) (URINE = 0)) THEN D1=0, D2=1

Table 2. Results of CC-STAR-GA for the acute inflammations dataset.

Classes	Number of items	Coverage	Premises	Number of rules*
1	20	20	3	4
2	20	20	2	1
3	10	10	3	2
4	10	10	3	2
5	21	21	2	1
6	9	9	2	1
7	10	10	3	2
8	10	10	2	1
9	10	10	2	2

Table 3. Statistics of the 9 classes obtained with CC-STAR-GA algorithm.

Metric / Class	1	2	3	4	5	6	7	8	9
Sensitivity	1.0	1.0	0. 71	1.0	1.0	1.0	1.0	1.0	1.0
Specificity	1.0	0. 97	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Positive Prediction Value	1.0	0. 87	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Negative prediction Value	1.0	1.0	0. 97	1.0	1.0	1.0	1.0	1.0	1.0
Prevalence	0.16	0. 16	0. 09	0. 09	0.17	0.07	0. 09	0. 09	0. 09
Detection Rate	0.16	0. 16	0. 06	0. 09	0.17	0.07	0. 09	0. 09	0. 09
Detection Prevalence	0.16	0. 18	0. 06	0. 09	0.17	0.07	0. 09	0. 09	0. 09
Balanced Accuracy	1.00	0. 98	0. 86	1.0	1.0	1.0	1.0	1.0	1.0

4.5 Evaluation and Comparison of Generated Rules

The support, confidence, and lift of each rule that belonged to GAJA2, ROUGH SET and CC-STAR-GA were compared and results are presented in Table 4. We were interested in rules such as lift(X→Y)>1 and c(X→Y)=1. Table 4 shows the support, confidence, and lift of each rule that belonged to GAJA2, ROUGH SET and CC-STAR-GA. We were interested in rules such as lift(X→Y)>1 and c(X→Y)=1.

Table 4. Results of the association evaluation of the rules of each algorithm tested.

ALGORITHMS	RULES	$\sigma(X)$	$\sigma(Y)$	$\sigma(X \cup Y)$	$s(X)$	$s(Y)$	$s(X \rightarrow Y)$	$c(X \rightarrow Y)$	LIFT
GAJA2	R1	29	50	29	0.24	0.42	0.24	1.00	2.40
	R2	50	70	50	0.42	0.58	0.42	1.00	1.71
	R3	80	59	59	0.67	0.49	0.49	0.74*	1.50
	R4	40	61	40	0.33	0.51	0.33	1.00	1.97
ROUGH SET	RN1	30	30	30	0.25	0.25	0.25	1.00	4.00
	RN2	11	31	11	0.09	0.26	0.09	1.00	3.87
	RN3	34	31	18	0.28	0.26	0.15	0.53*	2.05
	RN4	30	40	30	0.25	0.33	0.25	1.00	3.00
	RN5	9	40	9	0.08	0.33	0.08	1.00	3.00
	RN6	16	19	16	0.13	0.16	0.13	1.00	6.32
CC-STAR-GA	RU1	20	30	20	0.17	0.25	0.17	1.00	4.00
	RU2	20	40	20	0.17	0.33	0.17	1.00	3.00
	RU3	10	40	10	0.08	0.33	0.08	1.00	3.00
	RU4	10	40	10	0.08	0.33	0.08	1.00	3.00
	RU5	21	31	21	0.18	0.26	0.18	1.00	3.87
	RU6	9	19	9	0.08	0.16	0.08	1.00	6.32
	RU7	10	19	10	0.08	0.16	0.08	1.00	6.32
	RU8	10	30	10	0.08	0.25	0.08	1.00	4.00
	RU9	10	31	10	0.08	0.26	0.08	1.00	3.87

*Values with confidence less than 1.

The relevance of knowing the optimal value of β_0 in order to generate structures for grouping and getting rules with total coverage was shown because otherwise there would have been a risk of finding rules that were unable to classify new patterns within a defined group. The genetic algorithm improved the efficiency for the extraction of the class as well as for the reduction of the number of premises required for a total coverage of the dataset. The results showed that it was possible to obtain a better coverage (up to 97.7%) with the hybrid algorithm.

The results of the CC-STAR-GA algorithm were compared with the results of GAJA2 [17] and ROUGH SET [18] because both algorithms use the same data set and present the rules generated. These results showed that the number of rules obtained with CC-STAR-GA (9 rules) were more than with GAJA2, which found 4 rules, and with ROUGH SET algorithm, which found 6 rules. The rules proposed by the algorithm CC-STAR-GA had an accuracy and coverage from 97% to 100%, while GAJA2 had a coverage from 65.83% to 82.5%, and ROUGH SET had a coverage above 95%. The confidence value was equal to 1 for all the rules generated for CC-STAR-GA, which means that the reliability of the inference made by each rule was higher than the results obtained with the other algorithms. The lift value was more than 3 for each rule of CC-STAR-GA, even though we were just looking for a lift superior to 1. To sum up, all the rules in the algorithm CC-STAR-GA were potentially useful for predicting the consequent in future sets.

5 Conclusions

This work presents the performance of the algorithm CC-STAR-GA under an unsuper vised search where the main objective is to look for classes that allow inferring new k nowledge about the relationship among the attributes in a dataset. Automatic generatio n of rules as well as optimal definition of the number of premises has been an achieve d goal because it was possible to define relations into the structure and type of informa tion of the database. Besides, in a supervised case, the hybrid algorithm managed to fi nd a grouping structure that provided a minimum distance of groups and a maximum d istance between groups with the help of the value of the optimized β_0 given by the ge netic algorithm. Once a range of optimal $\beta_0$was defined, the rules obtained showed no intersections on their premises and covered completely each group´s pattern.

The evaluation showed a good performance of the algorithm CC-STAR-GA with th is case of study. Previously, used techniques had proved their efficiency separately and considering actual speed and memory improvement in hardware, this hybrid system co uld be a viable option for rule extraction in unstructured databases.CC-STAR-GA algo rithm could be a part of a predicting system based on knowledge such as a learning ma chines or decision support systems contributing with structure information.

Acknowledgments. This work was supported by Instituto Politécnico Nacional (IPN, México) and Secretaría de Ciencia y Tecnología del Distrito Federal (SECITI, México) with grant PIC-SA/12/201.

References

1. Coiera, E.: Guide to Health Informatics. Sydney, Australia, CRC Press (2003)
2. Vázquez, F.: Caracterización e interpretación de descripciones conceptuales en dominios pocos estructurados. Centro de Investigación en Computación, Distrito Federal (2008)
3. Shahbazkia, H., Al-Maqaleh, B. M.: A genetic algorithm for discovering classification rules in data mining. Int J Comput Appl, Vol. 41, No. 18, pp. 40–44 (2012)
4. Thankachan, T. A., Raimond, K.: A Survey on Classification and Rule Extraction Techniques for Datamining. IOSR Journal of Computer Engineering, Vol. 8, No. 5, pp. 75–78 (2013)
5. Martínez, J. F., Guzmán, A.: The logical combinatorial approach to pattern recognition, an overview through selected works. Pattern Recognition, Vol. 34, pp. 741–751 (2001)
6. Basu, M., Prasad, B.: Average distance and minimum average distance of binary constant weight code and its properties. Discrete Mathematics, Algorithms and Applications, Vol. 2, No. 3, pp. 379 (2010)
7. Michalski, R. S., Stepp, R. E.: Learning from observation: Conceptual Clustering. Machine Learning: An artificial intelligence approach (1983)
8. Reyes, Y., Martínez, N.: Algoritmos Conceptuales: Una perspectiva para el modelado del estudiante en los Sistemas Tutoriales Inteligentes (2011)
9. Guerra, A., Vega, S., Ruiz, J.: Algoritmos de agrupamiento conceptuales: un estado del arte. Reporte técnico reconocimiento de patrones (2012)
10. Meyrowitz, A. L., Chipman, S.: Foundations of Knowledge Acquisition: Machine Learning. USA: Kluwer Academic Publishers (1993)
11. Ponce, P.: Inteligencia Artificial con Aplicaciones a la Ingeniería. México: Alfaomega (2013)
12. Greenwell, R. N., Angus, J. E., Finck, M.: Optimal Mutation Probability for Genetic Algorithms. Mathl. Comput. Modelling, Vol. 21, No. 8, pp. 1–11 (1995)
13. Maulik, U., Bandyopadhyay, S.: Performance evaluation of some clustering algorithms and validity indices. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 12, pp. 1650–1654 (2002)
14. Tan, P. N., Steinbach, M., Kumar, V.: Introduction to Data Mining. First ed., Boston, MA: Addison-Wesley Longman Publishing (2005)
15. UCI Machine Learning Repository: Acute Inflammations Data Set. Available: https://archive.ics.uci.edu/ml/datasets/Acute+Inflammations (2003)
16. Yu-Wei, C.: Machine Learning with R Cookbook. UK: Packt Publishing (2015)
17. Kooptiwoot, S.: Mining Acute Inflammations of urinary system using GAJA2: A new data mining algorithm. In: Computer Science and Information Technology (ICCSIT), 3rd IEEE International Conference, Vol. 3, pp. 278–281 (2010)
18. Czerniak, J., Zarzycki, H.: Application of rough sets in the presumptive diagnosis of urinary system diseases. Artificial Intelligence and Security in Computing Systems, The Springer International Series in Engineering and Computer Science, Vol. 752, pp. 41–51 (2003)