Feature Selection for Clustering using Genetic Algorithms

Amal BOURAOUI, Sahar REGAIEG, Salma JAMOUSSI, Yassine BEN AYED
Multimedia, InfoRmation Systems and Advanced Computing Laboratory
Higher Institute of Informatics and Multimedia
University of Sfax, Tunisia
contact@miracl.rnu.tn, {jamoussi, yassine.benayed, Amal.bouraoui, Sahar.regaieg}@gmail.com

ABSTRACT: *The present article introduces a genetic algorithm based method to select interesting features in a clustering framework. Indeed, we used the evolutionary paradigm to explore many subsets of attributes and evaluate them according to inertia criteria when the K-Means clustering method is used in different way to data clusters. The proposed method is applied to many benchmark datasets. The experimental obtained results show the efficiency of the proposed method where features were reduced by more than 50% and the efficiency of the clustering has been improved.*

Keywords: Feature Selection, Genetic Algorithm, Multi-criteria Optimization, Clustering, K-Means

Received: 2 March 2013, Revised 11 April 2013, Accepted 19 April 2013

1. Introduction

With the evolution of computer science and the storage technologies, data volume is in a constant increase as well as the attributes describing that data. One possible solution to face this problem is to reduce the dimension of data by selecting a subset of attributes that are the most interesting. The main idea of feature selection is to choose a subset of input features by eliminating features with little or no predictive information. In particular, feature selection removes irrelevant features, increases efficiency of learning tasks, improves learning performance and enhances comprehensibility of learned results [1-2]. However, it is possible to have the same accuracy with different subsets, so the result of the selection is not necessarily one only optimal solution. In general, feature selection refers to the study of algorithms that select an optimal subset from the input feature set. Optimality is normally dependent on the evaluation criteria or the application's needs.

Genetic algorithms (GA) have received much attention because of their ability to solve difficult problems in the optimization. In this study we decided to perform feature selection based on genetic algorithms using different evaluation criteria. To examine the clustering performances we used different benchmark datasets.

The remainder of this paper is organized as follows. We begin in Section 2 with an overview of related work of supervised and unsupervised features selection. Section 3 describes our proposed approach. We begin with a brief introduction to K-means and GA. And we present a genetic algorithm for feature selection using a reference clustering and genetic algorithm for feature selection using multiple clustering. The experimental results on UCI benchmark datasets are discussed in Section 4. We conclude this work in Section 5.

2. Related Work

Various approaches have been proposed for finding irrelevant features and remove them from the feature set.

In supervised classification task, Estevez and Caballero [3] propose a GA based method for selecting features for neural network classifiers. Their algorithm aims to find and maintain multiple optima. They also introduce a new mutation operator to speed up the convergence of the GA. Matsui and al. [4] use GA to select the optimal combination of features to improve the performance of tissue classification neural networks and apply their method to problems of brain MRI segmentation to classify gray matter/ white matter regions. Li Zhuo and Jing Zheng [5] propose a Genetic Algorithm (GA) based wrapper method for classification of hyper spectral data using Support Vector Machine (SVM). The genetic algorithm (GA), which seeks to solve optimization problems using the methods of evolution, specifically survival of the fittest, was used to optimize both the feature subset, of hyper spectral data and SVM kernel parameters simultaneously. A special strategy was adopted to reduce computation cost caused by the high-dimensional feature vectors of hyper spectral data when the feature subset part of chromosome was designed.

Rhee and Lee [6] present an unsupervised feature selection method using a fuzzy-genetic approach. The method minimizes a feature evaluation index which incorporates a weighted distance between a pair of patterns used to rank the importance of the individual features. A pattern is represented by a set of features and the task of GA is to determine the weighting coefficients of features in the calculation of weighted distance.

3. Proposed Approach

We present a new wrapper feature selection method, based on Genetic Algorithms and the K-means clustering method. The purpose is to optimize the used feature subset and achieve higher clustering accuracy. We briefly discuss these tools on the following subsections:

3.1 Brief introduction to K-Means

Clustering involves dividing a population of objects into subsets of objects called clusters or groups. All objects in the same group have to be similar and objects in separate groups have to be dissimilar.

K-Means [7] is an iterative clustering algorithm. It starts with a set of *K* reference individuals randomly selected. The data are partitioned in *K* groups; an individual belongs to a group if the center of this cluster is the most close to him (in terms of distance). Updating of centroids and assigning individuals to clusters of data are performed during the successive iterations. *K*-Means works to optimize the inertia criteria.

The intra-class inertia is used to evaluate the heterogeneity within classes. The clustering of a population is even better than its intra-class inertia I_A is small.

$$I_A = \sum_{K=1}^{g} I_T\,(C_K) \tag{1}$$

When P is a population consisting of N individuals, $P = \{I_1, I_2, \ldots I_N\}$ and B_p their center then I_T is given by:

$$I_T\,(C_k) = \frac{1}{N} \sum_{K=1}^{N} d\,(B_p, I_K)^2 \tag{2}$$

Where N_K is the number of objects in the class k and g is the number of the generated clusters.

The inter-class inertia (I_E) evaluates the similarity between classes. More the inertia I_E is great, the better clusters are separated, so heterogeneous and therefore the clustering is better.

$$I_E = \sum_{k=1}^{g} d\,(B_p, B_{c_k})^2 \tag{3}$$

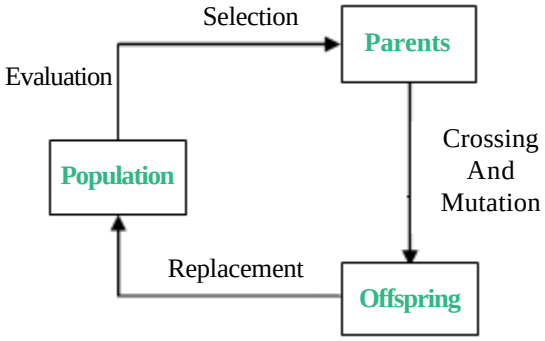

Figure 1. The Genetic Algorithm (GA) Process

Where B_{C_k} is the center of the group *k*.

3.2 Brief introduction to GA

Genetic algorithm is an optimization algorithm which is based on techniques derived from human genetics and mechanisms of natural selection. GA works with a set of candidate solutions called a population. Based on the Darwinian principle of '*survival of the fittest*', GA obtains the optimal solution after a series of iterative computations. GA generates successive populations of alternative solutions that are represented by individuals (called chromosomes), until acceptable results are obtained. In general, the genetic information is represented by a bit string (such as binary strings of 0s and 1s), and sets of bits encode solutions. Associated with the characteristics of exploitation and exploration search, GA can deal with large search spaces efficiently, and hence has less chance to get local optimal solution than other algorithms. A fitness function is used to evaluate the quality of a solution. The crossover and mutation functions are the main operators that randomly impact the fitness value. Crossover creates two offspring strings from two parents strings by copying selected bits from each parent, whereas mutation randomly changes the value of a single bit (with small probability). Chromosomes or solutions are selected for reproduction by evaluating their fitness value. The fitter chromosomes have higher probability to be selected for GA operations [8-10]. This cycle is repeated until a termination criterion is met.

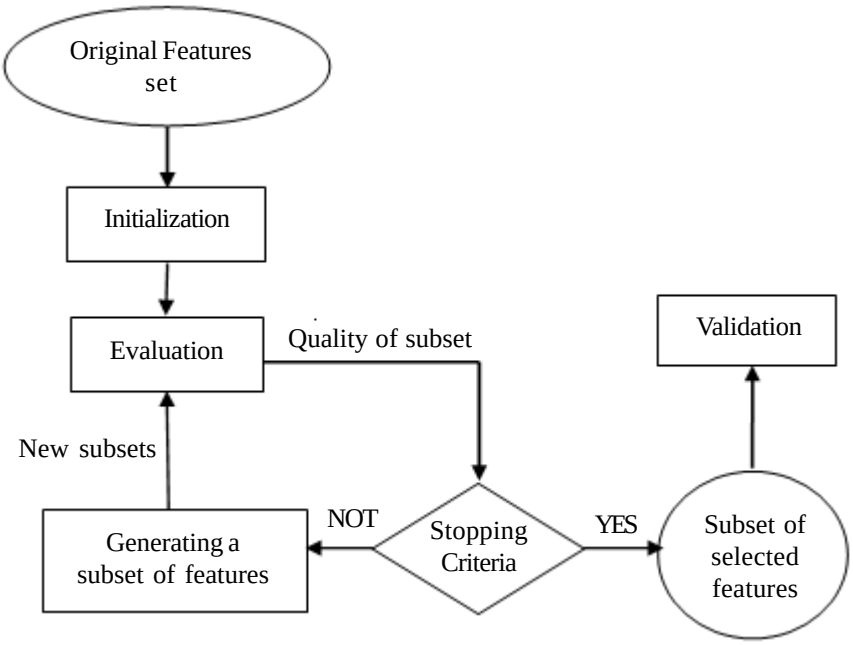

Figure 2. The Feature Selection Process

3.3 Genetic algorithm for feature selection using a reference clustering

We propose a general structure for our attribute selection algorithm. First, we begin with an initialization step which consists on generating some initial arbitrary subsets of attributes. Then we proceed iteratively. Each iteration begins with an evaluation of the quality of candidate subsets in order to choose the most optimal. After that, we check the stopping criteria. If they are not met, we regenerate other subsets to fuel the next iteration.

The application of our genetic algorithm requires a data clustering to measure the fitness of our solutions. At the beginning, we choose to start with an initial grouping using the *K*-Means algorithm by considering all the attributes. In this version, *K*-Means is applied once (at the beginning) (see Figure 3).

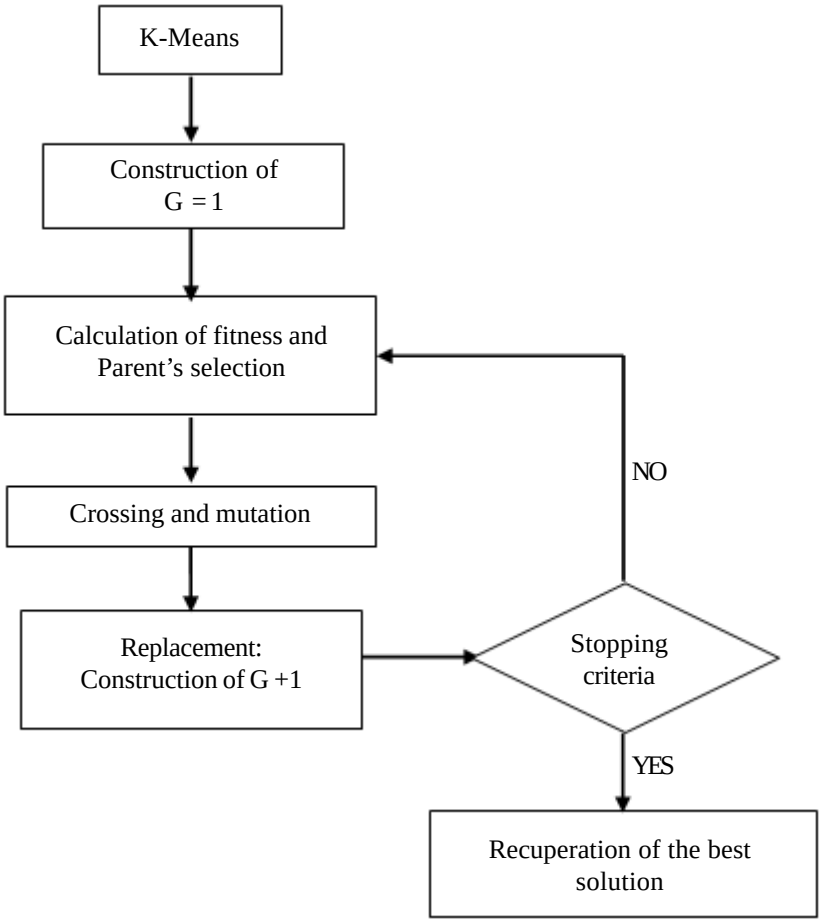

Figure 3. The Architecture of our Method

3.3.1 Chromosome design and initialization

Individuals represent subsets of features by means of binary strings. Each binary digit (gene) stands for the presence (1) or the absence (0) of a given feature.

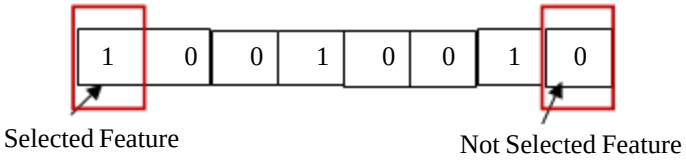

Figure 4. The Chromosome Design

Each initial solution is generated randomly. The value of each binary digit is defined according to a fixed probability.

At the beginning, this probability value was set at 0.5, but we founded that in this case generated individuals had several bits equals to 0 which leads to premature convergence. So, we tried to increase this value step by step. Finally, we conclude that the best value is 0.8. Table 1 shows the results of our various tests applied on benchmarks downloaded from the UCI ML-Repository [11].

Probability	Benchmark	IA	I E	F
0.5	B01	2.7	9.85	0.13
	B02	3701.31	4999.08	0.05
	B03	7380.54	3399.25	0.03
	B04	2.9	0.25	0.07
0.6	B01	2.5	10.21	0.15
	B02	3542.09	5213.58	0.13
	B03	7269.69	3752.68	0.07
	B04	2.3	0.40	0.11
0.8	B01	1.60	13.30	0.26
	B02	2360.14	8728.14	0.36
	B03	5423.60	6871.56	0.14
	B04	0.09	0.52	0.27

Table 1. Validation Tests Of The Choice Of Probability

Where ***IA*** is the intra-class inertia, ***IE*** is the inter-class inertia, ***F*** refers to the F-measure well known measure which is calculated by this formula.

$$F = \frac{2 * R * P}{R + P} \tag{4}$$

When R refers to the Recall and P refers to Precision well known measure which.

$$R(i, j) = \frac{n_{i,j}}{N_i} \tag{5}$$

$$P(i, j) = \frac{n_{i,j}}{N_i} \tag{6}$$

Where n_{ij} is the number of data presented in the reference class and in class result. N_i and N_j represent respectively the total number of data in this classes.

B01, B02, B03 and B04 refer respectively to Iris, Image Segmentation, Pima Indians Diabetes and Mouvements Libras databases. If the size of the population (denoted T) is too large, then the complexity of the algorithm is too high, and the computation time is too large. If the size of the population is too small, then the performance of the algorithm is reduced, and the algorithm can be plunged into local optima solutions. After many experiments we concluded that a size range from 20 to 100 individuals is suitable.

3.3.2 Fitness function design

The intra-class inertia is used to evaluate the fitness of the individuals. this measure depends on the number of features N (see equation 1). In our case, we use only selected features to calculate the intra inertia. As the number of selected features (denoted N_S) differs from an individual to another so this fitness will favor individuals having a minimum number of selected features. In order to give individuals having greatest NS the same chance of being reproduced than those with a small N_S we have divide the intra-class inertia by the N_S value. So, the Fitness function is designed as follows:

$$fitness = \frac{\sum_{j=1}^{K} \sum_{i=1}^{NI} d(B_j, I_i)^2}{K \times N_S} \quad (7)$$

0	1	0	1	0	1

Figure 5. Design of the used Mask

Where K is the number of groups, N_I is the total number of instances in the group I, N_S is the number of the selected features and $d(B_j, I_i)$ is the distance between the center of class j and the instance I_i.

3.3.3 Selection

During the selection step we select best individuals in current population as parents to generate offspring. Fitness, Inter inertia and the number of selected attributes is used as criteria to judge whether individuals are fittest. We use the binary tournament method for the selection step. The idea consists in randomly select six solutions from which we keep only three; each of them is optimal for one criterion.

3.3.4 Crossing

At the beginning, we used the one point crossover. We noticed that the genetic inheritance of offspring and their parents are very similar, so it leads to a premature convergence. For this reason, we decided to use another type of crossing: two cross points. However, this choice does not solve the problem, especially when the number of features (N) is important. So we tested the uniform crossover and we had the best results. Thus, we adopt it. The used mask is then designed as the following:

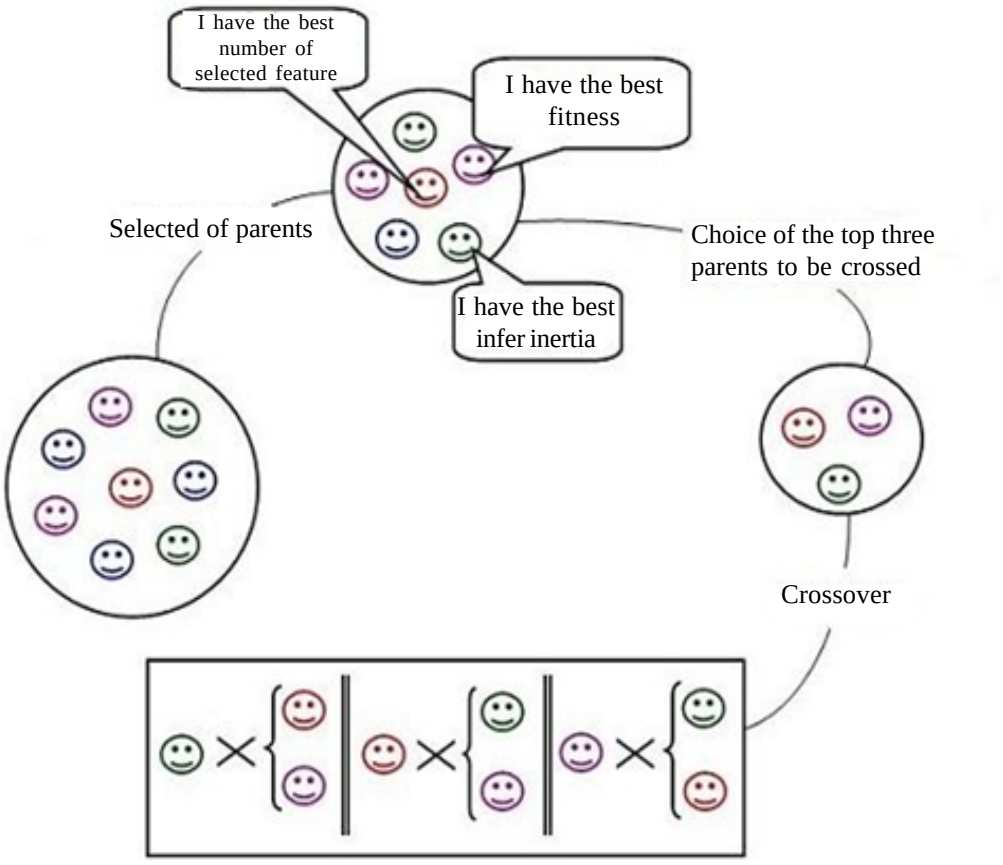

Figure 6. Selection and crossing using multiple criteria

The three retained individuals at the selection step are crossed each other. For example, if we take the parent having the best number of selected features so he was crossed with the best parent for the I_A based fitness and with the best parent for *IE* (see Figure 6).

This process is repeated a number of iterations equal to *T*/5 (where *T* is the population size). This choice is not arbitrary since we want to have a number of parents at least equal to *T*/2 to ensure diversity.

3.3.5 Mutation

Our mutation is simply an inversion of a bit being in a locus position randomly determined. The mutation probability is calculated as follows:

$$P_{mut} = 1 - \frac{K}{N_S} \quad (8)$$

where *K* is fixed experimentally. In our application, it is fixed to 5 and N_S is the number of the selected features.

3.3.6 Replacement and choice of the final solution

This operator consists in reintroducing the offspring in the population's parents. Our strategy is based on the construction of a population with children, parents and remaining individuals in the generation (*G*). Then, we ordered these solutions in three vectors according to the criterion to be optimized. Finally, we select the best for each objective to reach the size threshold for a generation (see Figure7).

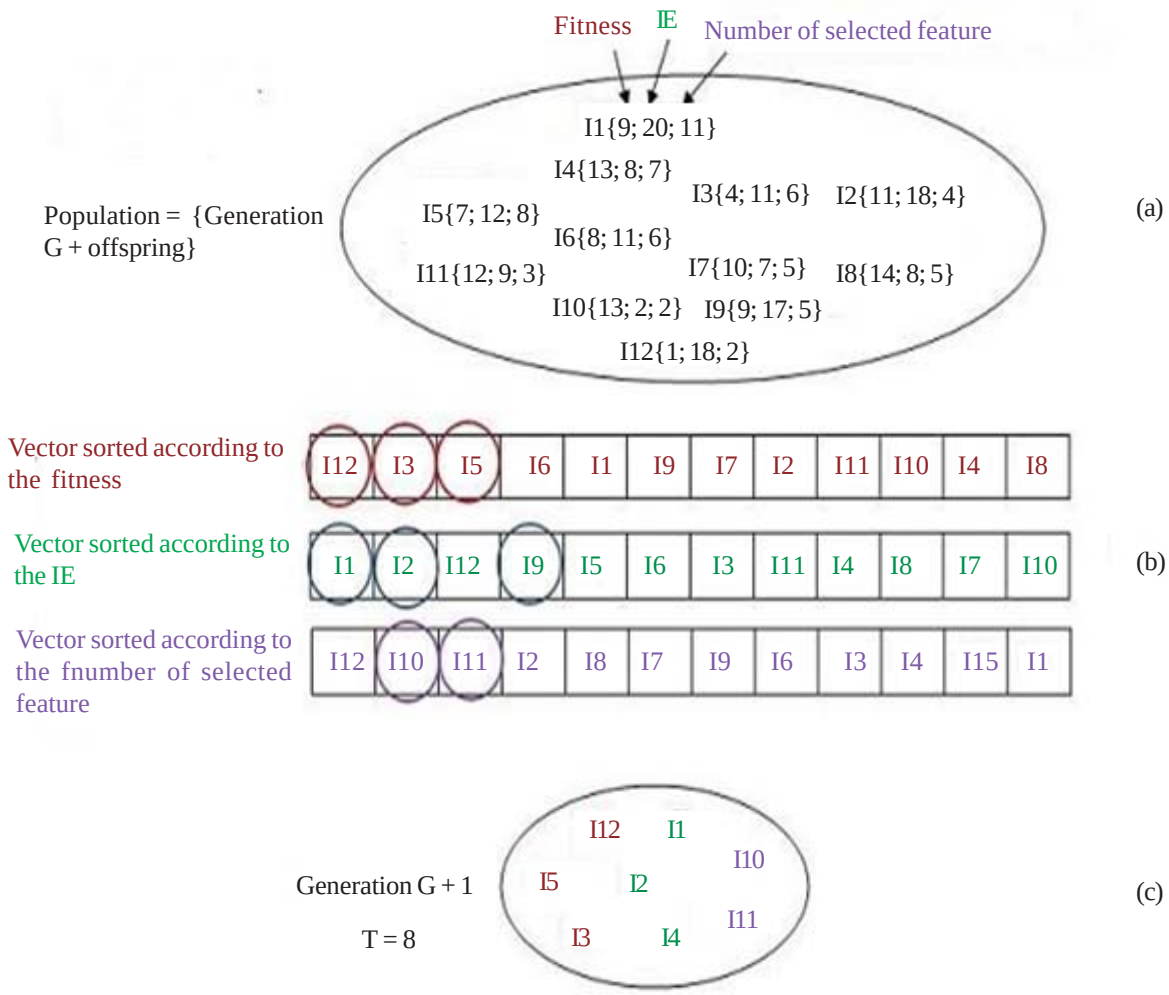

Figure 7. Replacement using multiple criteria

The optimal solution is the best solution in the last generation that satisfies the better our three optimization criteria.

3.3.7 Genetic algorithm for feature selection using multiple clustering

Our goal is to minimize the number of attributes while trying to improve the efficiency of the clustering, that's why we made a second version of our method where we apply *K*-Means for each individual in a generation. In this case *K*-Means is run multiple times and we keep the individual which has the best fitness calculated on the new clustering to use it in the next generation and so on (see Figure 8).

We use the same operators of the genetic algorithm applied in the previous section (selection, crossover, mutation and replacement) and we compare results in the next section.

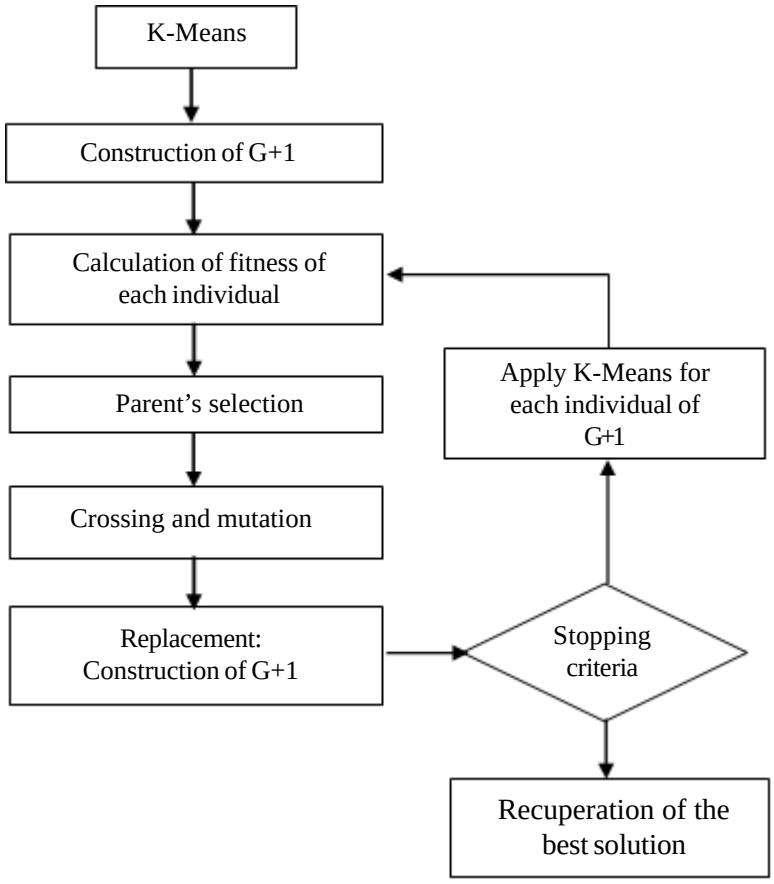

Figure 8. Feature selection using the *K*-Means algorithm as wrapper method

4. Experimental Results And Discussion

To validate and evaluate our algorithms we conducted a series of experiments and tests on benchmarks downloaded from the UCI ML-Repository [11]. The reason behind choosing these datasets is the diversity of numbers of features and data sizes.

The information regarding these datasets is listed in table 1.

The following tables compare the obtained results in both cases when using *K*-Means just once in the beginning and when it is used as a wrapper method for feature selection:

Where NBAtt is the number of selected features, *IA* is the intra-class inertia, *IE* is the inter-class inertia, *F* refers to the *F*-measure well known measure, B01, B02, B03 and B04 refers respectively to Iris, Image Segmentation, Pima Indians Diabetes and Mouvements Libras databases.

Designation	Databasename	Number of features	Number of classes	Number of instances
B01	Iris	4	3	150
B02	Image-Segmentation	19	7	210
B03	PimaIndians-Diabetes	8	2	180
B04	MovementLibras	90	15	360

Table 2. Informations About The Used Datasets

Benchmark	F	K-Means applied only once			
		NBAt	***IA***	***IE***	***F***
B01	0.40	2	0.98	8.82	0.30
B02	0.42	10	1836.85	4863.96	0.56
B03	0.28	4	3274.42	9063.76	0.29
B04	0.29	52	0.85	0.28	0.30

Table 3. A Comparison Between Results Obtain When Using *K*-Means Just Once And *K*-Means With All Features

A comparison between the results obtained by applying *K*-Means once and those obtained by applying *K*-Means as wrapper method shows that the latter is more effective especially when the initial number of attributes is very high. Consider the case of the database B04 (MovementLibras), the number of disregarded attributes during the application of *K*-Means as wrapper method is 50. This result is more satisfactory than that obtained when applying *K*-Means just once (which is equal to 38). Thus, it is noted that the approach of *K*-Means as wrapper method does not only minimizes the number of attributes but also improves the classification rate significantly (from 0.30 to 0.62).

One also notices that the number of not selected features created by the two different approaches could be unchanged but the global efficiency will be better when we apply *K*-Means as wrapper method. Such as the database B01's (Iris) results show that the classifier applied once performs slightly more poorly than the wrapper method (0.3 vs. 0.68) whereas we have the same number of selected features. We conclude that the *K*-Means used as a wrapper method is more likely to eliminate uninformative features and to retain the most significant features than the first method.

Benchmark	F	K-Means applied only once			
		NBAt	***IA***	***IE***	***F***
B01	0.40	2	0.32	8.89	0.68
B02	0.42	9	625.6	2895.15	0.71
B03	0.28	3	882.96	5369.78	0.50
B04	0.29	40	0.14	0.37	0.62

Table 4. A Comparison Between Results Obtain When Using *K*-Means As Wrapper Method And K-Means With All Features

5. Conclusions

In this paper, we present a new method based on genetic algorithms for unsupervised feature selection. This task aims to reduce the full set of attributes to a subset which contains only the relevant attributes while improving clustering performance. The reported results indicate that an unsupervised feature selection strategy based on genetic algorithms using K-means as wrapper

method can yield a significant reduction in the number of features which is better than the approach using K-means just once in the beginning.

We plan to use other complementary techniques that could improve our results, such as hybrid methods. We can also envisage the application of the same selection strategies with other classification methods, such as SVM in a supervised framework.

References

[1] Chen, Z., Lü, K. (2006). A preprocess algorithm of filtering irrelevant information based on the minimum class difference. *Knowledge-Based Systems*, 19 (6) 422–429.

[2] Zhu, F., Guan, S. Feature selection for modular GA-based classification. Applied Soft Computing, 4 (4) 381–393.

[3] Estevez, P., Caballero, R. (1998). A niching genetic algorithm for selecting features for neural classifiers, *In*: Proceedings of the Eighth International Conference on Artificial Neural Networks, vol. 1, Springer, London, p. 311–316.

[4] Matsui, K., Suganami, Y., Kosugi, Y. (1999). Feature selection by genetic algorithm for MRI segmentation, *Systems and Computers in Japan*, 30 (7) 69–78. Scripta technical.

[5] Li Zhuo, Jing Zheng, Fang Wang, Xia Li, Bin Ai, Junping Qian. (2008). A genetic algorithm based wrapper feature selection method for classification of hyperspectral images using support vector machine. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. V. XXXVII. Part B7.

[6] Rhee, Y., Lee, Y. (1999). Unsupervised feature selection using a fuzzy-genetic algorithm, *In*: Proceedings of the IEEE International Fuzzy Systems Conference, Piscataway, *NJ*, V. 3, p. 1266–1269.

[7] MacQUEEN, J. (1967). Some methods for classification and analysis of multivariate observations. *In*: Proceedings of the Berkeley Symposium on Mathematical Statics and Probability, p. 281-297.

[8] Hamdani, T. M., Alimi, A. M., Karray, F. (2006). Distributed genetic algorithm with bi-coded chromosomes and a new evaluation function for features selection, *In*: Proceeding of IEEE Congress on Evolutionary Computation, Canada, p. 581–588.

[9] Huang, C. L., Wang, C. J. (2006). A GA-based feature selection and parameters optimization for support vector machines. *Expert Systems with Applications*, 31 (2) 231–240.

[10] Oh, I. S., Lee, J. S., Moon, B. R. (2004). Hybrid genetic algorithms for feature selection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26 (11) 1424–1437.

[11] ML-Repository. http ://archive.ics.uci.edu/ml/.