International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 5 (2018) pp. 2163-2167

Unsupervised classification by Isodata using genetic algorithm and Xie - Beni criterion

Mohammed Merzougui * and Ahmad EL Allaoui **

**Labo Matsi, Est, Ump, B.P 473, Oujda, Morocco.*
E-mail: merzouguimohammed61@gmail.com

***Department MI, Ensah, Ump Al Hoceima, Morocco.*
E-mail: hmad666@gmail.com

Abstract

The unsupervised classification by the Isodata algorithm is closely dependent on the two parameters: the threshold to divide one class and the other threshold to merge two classes. Poor optimization of these two parameters leads the algorithm to escape any control retaining only one class in the end. The objective of this work is to make improvements to this algorithm; we used genetic algorithms (GA). We have designed a GA that will estimate the two optimal thresholds. These two parameters thus found will then be used by the Isodata algorithm. This approach is tested on simulation examples. The experimental results obtained confirm favorably the good performances of the proposed algorithm.

Keyword (s): Classification, Genetic Algorithms (GA), Isodata, Xie-Beni criterion.

INTRODUCTION

The classification is usually to partition a set of objects into groups or classes so that objects belonging to the same class are more similar to each other than those belonging to different classes. Classification with more parameters requires the choice of one or more criteria for determining the appropriate parameters to the algorithm. The problem of classification with ISODATA algorithm becomes a problem of optimization of parameters.[1]

The Isodata algorithm suffers of adjustment of its parameters. This is a step that is always difficult as you want towards the expected solution. This algorithm converges in a finite number of iterations, but the resulting solution depends on the values of the parameters chosen. Indeed if the algorithm resets a second time with other parameters, it will converge to a solution completely different from the first or escape out of control leaving at the end one alone class.[2]

In this work we present improvements to this algorithm, based on genetic algorithm to overcome these drawbacks. With benefits such as generality, parallelism and genetic operations, we have designed genetic algorithm is made to delve into an area of research to finally give the optimal parameters which are then used by ISODATA. The ISODATA algorithm manipulates a single standard solution at each iteration, for against the proposed algorithm manipulates a population of solutions at the same time. These solutions suffer from the mutation operator, and as the iterations, a Gaussian disturbance which prevents local solutions and reduce the time of convergence to the global solution.

In Section 2, we give some definitions, and then we recall the ISODATA algorithm. In Section 3, we present an introduction to genetic algorithm. Section 4 describes the proposed genetic algorithm. In Section 5, we evaluate the performance of our method by experimental results. Finally, we give a conclusion.

ISODATA CLASSIFICATION

The Isodata method is the method developed by Ball, Hall and others in the 1960s. The Isodata method is a method which added division of a cluster, and processing of fusion to the K-means method. The individual density of a cluster is controllable by performing division and fusion to the cluster generated from the K-means method. The individual in a cluster divides past [a detached building] and its cluster, and the distance between clusters unites them with past close. The parameter which set up division and fusion beforehand determines. The procedure of the Isodata method is shown as follows:

1) Parameters, such as the number of the last clusters, a convergence condition of rearrangement, judgment conditions of a minute cluster, branch condition of division and fusion, and end conditions, are determined.

2) The initial cluster center of gravity is selected.

3) Based on the convergence condition of rearrangement, an individual is rearranged in the way of the k-means method.

4) It considers with a minute cluster that it is below threshold with the number of individuals of a cluster, and excepts from future clustering.

5) When it is more than the threshold that exists within fixed limits which the number of clusters centers on the number of the last clusters, and has the minimum of the distance between the cluster center of gravity and is below threshold with the maximum of distribution in a cluster, clustering regards it as convergence and ends processing. When not converging, it progresses to the following step.

6) If the number of clusters exceeds the fixed range, when large, a cluster is divided, and when small, it will unite. It divides, if the number of times of a repetition is odd when there

is the number of clusters within fixed limits, and if the number is even, it unites. If division and fusion finish, it will return to 3 and processing will be repeated.

- ✓ Division of a cluster: If it is more than threshold with distribution of a cluster, carry out the cluster along with the 1st principal component for 2 minutes, and search for the new cluster center of gravity. Distribution of a cluster is re-calculated, and division is continued until it becomes below threshold.
- ✓ Fusion of a cluster: If it is below threshold with the minimum of the distance between the cluster centers of gravity, unite the cluster pair and search for the new cluster center of gravity. The distance between the cluster center of gravity is re-calculated, and fusion is continued until the minimum becomes more than threshold.

Although the Isodata method can adjust the number of certain within the limits clusters, and the homogeneity of a cluster by division and fusion, global optimal nature cannot be guaranteed. Since the Isodata method has more parameters than the K-means method, adjustment of the parameter is still more difficult.[2][3][4]

GENETIC ALGORITHMS

Genetic Algorithms (GA) are particular methods for optimizing functions. These techniques are based on the evolution of a population of solutions which under the action of some precise rules optimize a given behavior, which initially has been formulated by a given specified function called fitness function. [5][6][7]

A GA algorithm manipulates a population of constant size. This population is formed by candidate points called chromosomes. Each of the chromosomes represents the coding of a potential solution to the problem to be solved, it is formed by a set of elements called genes, and these are real.

At each iteration, called generation, a new population is created from its predecessor by applying the genetic operators: the crossing, the mutation and the selection. The first two operators change the chromosomes (previous generation) of the population in order to produce new chromosomes (next generation) to optimize selective function further and avoid local optima. The selection operator consists in constructing the next generation. This new population is constituted by the pertinent individuals [8] [2].

Fig1 illustrates the different operations to be performed in a standard GA algorithm [8][9][5][6] :

Random generation of the initial population Fitness
evaluation of each chromosome Repeat
  Select the parents
  Modifying genes by crossing and mutation
  Select the next generation
  Fitness evaluation of each chromosome Until
Satisfying the stop criterion

Figure 1. Standard GA algorithm.

GENETIC ALGORITHM PROPOSED

Descriptive elements

Consider a set of M objects $\{O_1, O_2,..., O_M\}$ characterized by N attributes, grouped in a line vector form $V = (a_1\ a_2\ ...\ a_N)$. Let $R_i = (a_{ij})$ $1<j<N$ be a line vector of R_N where a_{ij} is the value of the attribute a_j for the object O_i. Let mat_obs be a matrix of M lines (representing the objects O_i) and N columns (representing the attributes a_j):

$$\text{mat\_obs} = (a_{ij})_{\substack{1\le i\le M \\ 1\le j\le N}}$$

V is the attribute vector, R_i is the observation associated with O_i or the realization of the attribute vector V for this object, R_N is the observations space [1][8][9][10][11] and mat_obs is the observation matrix associated with V. The ith line of mat_obs is the observation R_i. R_i belongs to a class CL_s, s=1,…, C.

Proposed coding

The proposed algorithm consists of selecting among all of the possible partitions the optimal partition by minimizing a criterion. This yields the optimal parameters $(p_s)_{1<s<np}$. Thus, the real coding following is suggested:

$$chr = (p_s)_{1\le s\le np}$$

$$= (p_1, p_2, p_3 \ldots p_{np})$$

The chr chromosome is a real line vector of dimension np. The genes $(p_s)_{1\le s\le np}$ are the components of the chromosome chr.

To avoid that the initial solutions be far away from the optimal solution, each chromosome chr of the initial population should satisfy the condition:

$ps \in [minp, maxp]$ (chosen by the user) (3)

In our case and from the experiences already used the ISODATA algorithm we chose: minp = 0 and maxp = 1.

In the proposed algorithm, any chromosome with a gene that does not satisfy this constraint is eliminated. This gene, if any, is replaced by another one which complies with the constraint [8][9][12][13][14].

Xie-Beni criterion

Xie and Beni have defined validity criteria that measure the degree of compactness and separation of a given fuzzy partition. And the authors define their function of validity of compactness and separation as being the ratio of the compactness obtained to the separation relative to the fuzzy c-partition, which is none other than the square of the minimum of the distances between centers. [15][16]:

- the criterion of compactness, defined by:

$$comp = \frac{1}{M}\sum_{i=1}^{M}\sum_{s=1}^{C}\|R_i - g_s\|$$

- And the separability criterion, defined by:

$$sep = min_{s\neq s'}\|g_s - g_{s'}\|$$

- Thus, we obtain the Xie-Beni criterion V_{XB}:

$$V_{XB} = \frac{1}{M}\sum_{i=1}^{M}\sum_{s=1}^{C}\|R_i - g_s\| \Big/ min_{s\neq s'}\|g_s - g_{s'}\|$$

The proposed fitness function

Unlike the K-means algorithm that requires you to make the number of classes a priori, the Isodata algorithm itself determines the number of classes does not exceed a maximum previously selected and is among the parameters that must provide user. We are inspired by the behavior of the algorithm Isodata to choose selective function. Indeed, the repetition of this algorithm several times with parameters automatically generated by evolution strategies lead to the end of the optimal number of classes, this is the ultimate goal. To this end the test, well known, Xie-Beni criterion is selected.

chr is a chromosome of the population formed by the parameters $(p_s)_{1<s<np}$, to calculate the selective value of chr we define the function F which reflects the selective behavior to optimize [8][13]:

$$F(chr) = \frac{comp}{sep}$$

$$= \frac{1}{M}\sum_{i=1}^{M}\sum_{s=1}^{C}\|R_i - g_s\| \Big/ min_{s\neq s'}\|g_s - g_{s'}\|$$

chr is optimal if F is minimal.

Suggested crossing and mutation operators

The performances of a genetic algorithm are judged according to the crossing and mutation operators used [12]. For the crossover operator, we chose the arithmetic mean of the 2 parent individuals on each of their parameters. Indeed, if we have two chromosomes:

$$chr = (p_s)_{1\leq s\leq np} \quad \text{and} \quad chr' = (p_s')_{1\leq s\leq np}$$

The result chromosome chr^c is:

$$chr^c = (\frac{p_s + p_s'}{2})_{1\leq s\leq np}$$

The crossing is made between the first and each of the chromosomes of the population, which will preserve this first chromosome in the next generation.

As for the mutation operator, the one proposed in the literature [13] [5] [9] is given by the following expression:

chr * = chr + σ x N (0,1)

where chr * is the new chromosome produced by Gaussian perturbation of the chromosome chr. N (0,1) is a Gaussian of mean 0 and variance 1, σ is called strategic parameter.

In GAs, the crossover operator is most dominant in affecting almost all chromosomes. It is for this reason that the mutation takes place on a parameter only with a probability fixed at Pmut. We set Pmut to 0.1, that is to say that a parameter has a 10% chance of muting

The chromosomes (parents) that will be mutated to generate the chromosomes (wires), are chosen by the technique of choice by storage and the elitist technique [6] [15].

Genetic algorithm proposed

Figure 2 shows the steps of the proposed algorithm:

1.1. To start:

- The size of the population maxpop.

- The maximum number of generation maxgen.

- Maximum number of C classes.

- The probabilities of crossing and mutation

1.2. Random generation of population P

$P = \{chr_1, .., chr_k, ..., chr_{maxpop}\}$

1.3. Check for each *chr* of P the constraint: $p_s \in [minp, maxp]$

repeat

Check for each chr of P the constraint

2.1. Isodata for each chr of P

2.2. Calculate the selective value for each chr of P

2.3. Classification of chr in ascending order of their selective values

2.4 Cross the first chr with all the others

$$chr^{c} = (\frac{p_s + p_s^{'}}{2})_{1 \le s \le np}$$

2.5. Mutation of all the chr of P except the first (elitist technique): $p^*_s = p_s + \sigma \times N(0,1)$

2.6. Check for each chr of P the constraint: $p_s \in [minp, maxp]$

(*The population P obtained is the population of the next generation*)

Until *Nb_gen (Number of generation)> maxgen*

3.1. To keep the optimal *chr*: the first of the last P

3.2. *Isodata* for the optimal *chr*.

Figure 2. Genetic algorithm proposed

EXPERIMENTAL RESULTS

Four simulation tests are retained in an observation space of dimension N = 2. These tests differ from one another according to the distribution of the classes in the observation space. In each test, the classes are generated randomly by Gaussian distributions and each class contains 100 observations. Confusion matrices are given and the formula below to calculate the error of the classification [14].

$$\tau = (n\text{-}c) / n * 100\%$$

With n: total number of observations

and c: the number of correctly classified observations.

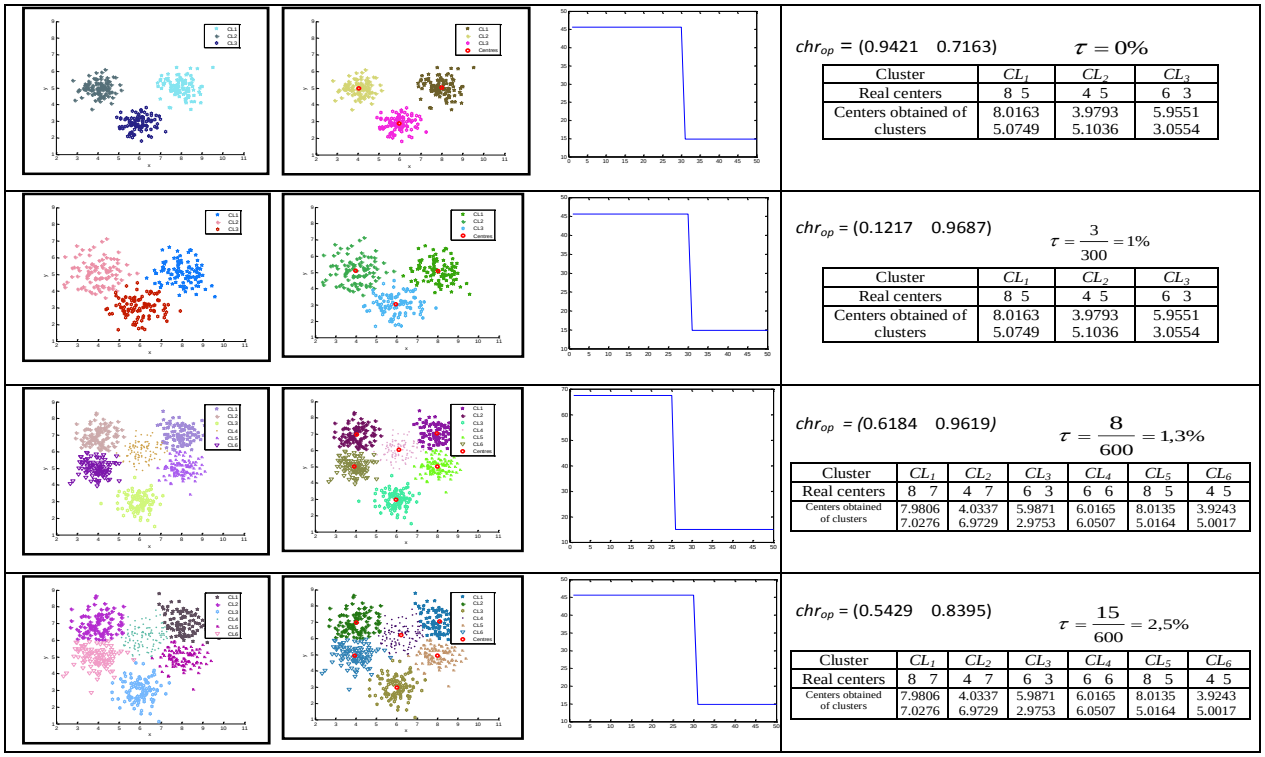
In the first two tests, the number of classes chosen is C = 3 and the degree of overlap between classes is zero in the first, and important in the second.

In the last two tests, the number of classes is C = 6. The degree of overlap between classes is low in one of the tests and it is important in the other.

The proposed genetic algorithm runs each time quickly (in the sense of the number of generations). The following summary shows:

- the distribution of observations in space.
- the actual centers of the generated classes.
- the evolution of the selective value of the best chromosome of the current population over generations.
- the optimal chromosome obtained each time.
- the coordinates of the real centers of the generated classes.

We notice clearly that the centers obtained are slightly offset from the real centers.



chr_{op} = (0.9421 0.7163) $\tau = 0\%$

Cluster	CL_1	CL_2	CL_3
Real centers	8 5	4 5	6 3
Centers obtained of clusters	8.0163 5.0749	3.9793 5.1036	5.9551 3.0554

chr_{op} = (0.1217 0.9687) $\tau = \frac{3}{300} = 1\%$

Cluster	CL_1	CL_2	CL_3
Real centers	8 5	4 5	6 3
Centers obtained of clusters	8.0163 5.0749	3.9793 5.1036	5.9551 3.0554

chr_{op} = (0.6184 0.9619) $\tau = \frac{8}{600} = 1{,}3\%$

Cluster	CL_1	CL_2	CL_3	CL_4	CL_5	CL_6
Real centers	8 7	4 7	6 3	6 6	8 5	4 5
Centers obtained of clusters	7.9806 7.0276	4.0337 6.9729	5.9871 2.9753	6.0165 6.0507	8.0135 5.0164	3.9243 5.0017

chr_{op} = (0.5429 0.8395) $\tau = \frac{15}{600} = 2{,}5\%$

Cluster	CL_1	CL_2	CL_3	CL_4	CL_5	CL_6
Real centers	8 7	4 7	6 3	6 6	8 5	4 5
Centers obtained of clusters	7.9806 7.0276	4.0337 6.9729	5.9871 2.9753	6.0165 6.0507	8.0135 5.0164	3.9243 5.0017

CONCLUSION

The unsupervised classification by the Isodata algorithm presents the difficulty of adjusting its parameters. The latter control its convergence. The wrong choice of these parameters can lead the algorithm to escape any control leaving only one class at the end.

We proposed a new approach to overcome this difficulty of the Isodata algorithm. This approach is based on genetic algorithms. We have presented a real coding of the parameters and we have defined a suitable selective function for the behavior to optimize. We proposed the crossing and mutation operators to allow the algorithm on the one hand to avoid local solutions and on the other hand to converge towards the global solution in a small number of generations.

The proposed genetic algorithm is tested on simulation examples. The experimental results obtained show the convergence and the good performances of the classification method presented. The problem of choosing the two parameters is eliminated. We note, however, that the other parameters are fixed empirically. Genetic estimation of all parameters will be the subject of a next search.

REFERENCES

[1]. Hamad, D., El Saad, S. et Postaire, J.G. "Algorithmes d'apprentissage compétitif pour la classification automatique". TISVA'98, Oujda, Maroc, pp. 159-164, 27-28 April 1998.

[2]. Presberger, T., Koch, M. “Comparison of evolutionary strategies and genetic algorithms for optimization of a fuzzy controller ". Proc. of EUFIT’95, Aachen, Germany, august 1995.

[3]. Glorennec, P. Y. "Constrained optimization of FIS using an evolutionary method ". In Genetic Algorithms and Soft Computing, F. Herrera and J. L. Verdegay Eds., Physica-Verlag, ISBN 3-7908-0956-X, 1996.

[4]. Nargess Memarsadeghi and al. ‘A fast implementation of the isodata clustering algorithm’, IJCGA, Vol. 17, No. 1, pp. 71-103, 2007.

[5]. Fogel, D. B. "An introduction to simulated evolutionary optimization". IEEE Trans. on Neural Networks, Vol. 5, No. 1, 1994.

[6]. Renders, J.M., Algorithmes génétiques et Réseaux de Neurones. Editions HERMES, 1995.

[7]. H.Ouariachi, D. Hamad, M.Barboucha, Optimisation de la classification non–Supervisée par la méthode des Nuées Dynamique Evolutionnioste. JTEA2000, pp. 110-115, Nabeul Hammamet, Tunisie, Mars 2000.

[8]. Glorennec, P. Y. "Algorithmes d’apprentissage pour systèmes d’inférence floue". Colloque Neuro-mimétisme, Lyon, juin 1994.

[9]. Sarkar, M. et al. "A clustering algorithm using an evolutionary-based approach". Pattern Recognition Letters, 1997.

[10]. EL Allaoui, M. Merzougui, M. Nasri, M. EL Hitmy and H. Ouariachi. Optimization of Unsupervised Classification by Evolutionary Strategies. IJCSNS International Journal of Computer Science and Network Security, ISSN: 1738-7906, Vol. 10, No. 6, pp. 325-332, June, 2010.

[11]. M. Merzougui, A. EL Allaoui, M. Nasri, M. EL Hitmy and H. Ouariachi. Unsupervised classification using evolutionary strategies approach and the Xie and Beni criterion. IJAST International Journal of Advanced Science and Technology, ISSN: 2005-4238, Vol. 19, pp. 43-58 June, 2010.

[12]. Michalewicz, Z. "Genetic algorithms + data structures = evolution programs". 2nd Eds., Springer-Verlag, 1994.

[13]. Schwefel, H. P. "Numerical optimization of computer models". Wiley, 1981.

[14]. Kohei Arai and XianQiang Bu, ‘ISODATA clustering with parameter (threshold for merge and split) estimation based on GA: Genetic Algorithm’ Reports of the Faculty of Science and Engineering, Saga University, Vol. 36, No.1, pp. 17-23, 2007.

[15]. M. Nasri, M. EL Hitmy, H. Ouariachi and M.Barboucha. Optimization of a fuzzy classification by evolutionary strategies. In proceedings of SPIE Conf., 6 th international conference on quality control by artificial vision. Repulished as an SME Technical Paper by The society of manufacturing Engineers (SME). Paper number MV03-233, IDTP03PUB135, Vol. 5132, pp. 220230, USA, 2003.

X. Xie and G. Beni. A validity measure for fuzzy clustering. IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI), Vol, 13, N. 8, pp, 841-847, 1991.