Unsupervised classification using evolutionary strategies approach and the Xie and Beni criterion

M. Merzougui, A. EL Allaoui, M. Nasri, M. EL Hitmy and H. Ouariachi
LABO MATSI, EST, University of Mohammed I, B.P 473, OUJDA, MOROCCO.
nasri@est.ump.ma ; ahmadallaoui@yahoo.fr; ; medmerzougui@yahoo.fr

Abstract

The kmeans algorithm is an unsupervised classification algorithm. It has some drawbacks, the number of classes has to be known a priori, the initialization phase and the local optimums. We present in this paper an improvement based on evolutionary strategies and on the Xie and Beni criterion in order to get around these three difficulties. We design a new evolutionist kmeans algorithm. We suggest a new mutation operator that allows the algorithm to avoid local solutions and to converge to the global solution in a small computation time. We have optimized the Xie and Beni criterion by evolutionary strategies for the optimal choice of the number of classes. The proposed method is validated on several simulation examples. The experimental results obtained show the rapid convergence and the good performances of this new approach.

Keywords. *Classification, evolutionary strategies, kmeans algorithm, evolutionist kmeans algorithm, new mutation operator, Xie and Beni criterion.*

1. Introduction

The *kmeans* algorithm (*KM*) is an unsupervised classification method [1,2,3,4]. However the *KM* algorithm requires the *a priori* determination of the number of classes [3,5] and suffers from the initialization phase and the local optimums [6,7,8,9,16,17]:

- This algorithm requires the optimal choice of the classes number. This optimal choice guides the algorithm to provide a partition with the smallest error value possible.
- This algorithm converges in a finite number of iterations but the solution depends on the initial values. Indeed, if we reinitialize the algorithm with other values, it may converge to an other local solution, which may be different from the first one.

We present in this work some improvements to this algorithm based on the evolutionary strategies and the *Xie* and *Beni* criterion in order to get around these three difficulties. We designed a new evolutionist *kmeans* algorithm (*EKM*) which has many advantages over the conventional *KM* algorithm. These are viewed in its generality, its parallelism and the genetic operations. The *KM* algorithm deals with one solution at each iteration, while the proposed *EKM* algorithm deals with a population of solutions in the same time. These solutions are subjected, during the iterations steps, to a Gaussian perturbation, which makes it then possible to avoid the local solutions. We propose a new mutation operator in order to control the Gaussian disturbance level and to reduce the computation time required to converge towards the global solution. We optimized *Xie* and *Beni* criterion by evolutionary strategies for the optimal choice of the number of classes.

Section 2 introduces the evolutionary strategies. In section 3, we give some definitions, and recall the *KM* algorithm. Section 4 describes our evolutionist *KM* algorithm. In section 5, we present the *Xie* and *Beni* criterion, and we optimize this criterion by evolutionary strategies. This will make it possible to avoid the drawbacks of this criterion. While in section 6, the performances of this new method are evaluated by some experimental results.

2. Evolutionary strategies

Evolutionary strategies (*ES*) 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 [9].

An ES 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, these are reals.

At each iteration, called generation, is created a new population from its predecessor by applying the genetic operators: selection and mutation. The mutation operator perturbs with a Gaussian disturbance the chromosomes of the population in order to generate a new population permitting to further optimize the fitness function.
This procedure allows the algorithm to avoid the local optimums. The selection operator consists of constructing the population of the next generation. This generation is constituted by the pertinent individuals [3,9].
Figure 1 illustrates the different operations to be performed in a standard *ES* algorithm [9,10]:

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

Figure 1: Standard SE algorithm.

3. kmeans classification

3.1. 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 $R_i = (a_{ij})_{1\le j\le N}$ be a line vector of $\boldsymbol{R}^N$ 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):

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

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, $\boldsymbol{R}^N$ is the *observations space* [1] and *mat_obs* is the *observation matrix associated* with V. The i^th^ line of *mat_obs* is the observation R_i. R_i belongs to a class CL_s, s=1, …, C.

3.2. kmeans algorithm

The *kmeans* algorithm is one of the most common algorithms used for the classification. We are given *maxobs* observations $(R_i)_{1\le i\le M}$ which must be associated with C classes $(CL_s)_{1\le s\le C}$ of centers $(g_s)_{1\le s\le C}$. The centers $(g_s)_{1\le s\le C}$ are line vectors of N dimension.
The *kmeans* is based on the minimization of the optimization criterion given by [2,3,4]:

$$J = \frac{1}{2}\sum_{i=1}^{M}\sum_{s=1}^{C} \left\| R_i - g_s \right\|^2 \qquad (2)$$

where $\| \, . \, \|$ is a distance which is generally supposed to be Euclidean.

The *KM* algorithm supposes that the number of classes *C* is known a priori.
Figure 2 gives the *KM* algorithm flowchart [3,4]:

1. **Fix** the number of classes *C*.
2. **Initialize** the centers at random values in the observation space
3. **Assign** the observations to classes having the closest centers.
4. **Update** the class centers
5. **Stop** the algorithm when the centers do not change, **if not go to 3**.

Figure 2: Flowchart of the KM algorithm.

4. Evolutionary kmeans classification

4.1. Proposed coding

The *KM* algorithm consists of selecting among all of the possible partitions the optimal partition by minimizing a criterion. This yields the optimal centers $(g_s)_{1\leq s\leq C}$. Thus we suggest the real coding as:

$$chr = (g_{sj})_{1\leq s\leq C, 1\leq j\leq N} = \left(g_{11} .. g_{1N} g_{21} .. g_{2N} .. g_{s1} .. g_{sN} .. g_{C1} .. g_{CN}\right) \qquad (3)$$

The *chr* chromosome is a real line vector of dimension $C \times N$. The genes $(g_{sj})_{1\leq j\leq N}$ are the components of the g_s center:

$$g_s = (g_{sj})_{1\leq j\leq N} = (g_{s1} g_{s2} .. g_{sj} .. g_{sN}) \qquad (4)$$

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

$$g_{sj} \in [\min a_{ij 1\leq i\leq M}, \max a_{ij 1\leq i\leq M}] \qquad (5)$$

In the *EKM* algorithm, we discard any chromosome with a gene that does not satisfy this constraint. This gene, if any, is replaced by an other one which complies with the constraint.

4.2. The proposed fitness function

Let *chr* be a chromosome of the population formed by the centers $(g_s)_{1\leq s\leq C}$, for computing the fitness function value associated with chr, we define the fitness function F which expresses the behavior to be optimized (J criterion):

$$F(chr) = \sum_{i=1}^{M}\sum_{s=1}^{C} \left\| R_i - g_s \right\|^2 \qquad (6)$$

The chromosome chr is optimal if F is minimal.

4.3. The proposed mutation operator

The performances of an algorithm based on evolutionary strategies are evaluated according to the mutation operator used[11]. One of the mutation operator form proposed in the literature [7,12,13]is given by:

$$chr^* = chr + \sigma \times N(0,1) \qquad (7)$$

where chr^* is the new chromosome obtained by a Gaussian perturbation of the old chromosome chr. *N(0,1)* is a Gaussian disturbance of mean value 0 and standard deviation value 1, σ is the strategic parameter. σ is high when the fitness value of chr is high. When the fitness value of chr is low, σ must take very low values in order to be not far away from the global optimum.

We have been inspired from this approach to propose a new form of the mutation operator. The fact that we have proposed a new mutation operator is motivated by our interest to reach the global solution in a small computational time.

Let chr be a chromosome of the population formed by the centers $(g_s)_{1 \le s \le C}$.
Let $R_i \in CL_s si \| R_i - g_s \| = \min_{s'=1,C} \| R_i - g_s \|$, i.e. the class consisting of the Ri observations that are closest to the center gs. Let $g°_s$ center of gravity CLs (figure 3).

$$g_S^{\circ} = \frac{\sum_{R_i \in CL_s} R_i}{l_S} \quad \text{where} \quad l_S = card(CL_S) \qquad (8)$$

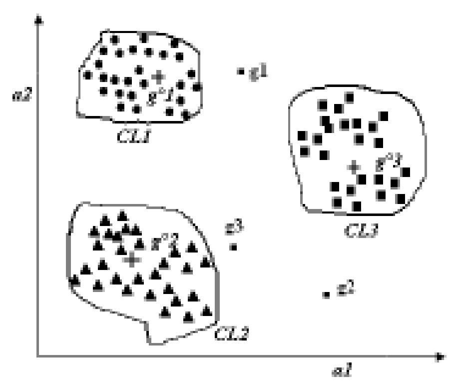

Figure 3: Illustration example in a two dimensional space.

The mutation operator which we propose in this work consists in generating, from the chr, the new chromosome chr^* formed by the centers $(g^*_s)_{1 \le s \le C}$, as:

$$g^*_s = g_s + f_m \times (g°_s - g_s) \times N(0,1) \qquad (9)$$

where f_m is a constant multiplicative factor taken to be between 0.5 to 1. The new strategic parameter proposed

$$\sigma' = f_m \times (g°_s - g_s)$$

is low when gs gets closer to g°s and is high when gs is far from g°s. The σ' proposed parameter has two advantages:

- When *chr* is far from the global solution, chr is subjected to a strong Gaussian perturbation allowing chr to move more quickly in the research space and in the same time to avoid local solutions.
- σ' controls the Gaussian perturbation level. Indeed, as the chromosome *chr* gets closer to the global solution, the Gaussian perturbation level is reduced until becoming null at convergence.

From generating children chromosomes from parent chromosomes we have adopted the technique of choice by ordering. We have also used the elitist technique [14].

4.4. The proposed *EKM* algorithm

Figure 4 shows the different steps of the proposed *EKM* algorithm.

Stage 1:

1.1. Fix:
- The size of the population *maxpop*.
- The maximum number of generations *maxgen*.
- The number of classes *C*.

1.2. Generate randomly the population *P*:

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

1.3. Verify for each *chr* of *P* the constraint:

$$g_{sj} \in [min\ a_{ij}, max\ a_{ij}],\ 1 \le i \le M$$

1.4. Attribute for each *chr* of *P*, the observations R_i to the corresponding classes:

$$R_i \in CL_s if \|R_i - g_s\| = \min_{\substack{s=1,C \\ s \neq s'}} \|R_i - g_s\|$$

1.5. Update the population *P,* for each *chr* of *P* do:

$$g'_s = \frac{g_s + \sum_{R_i \in CL_s} R_i}{1 + l_s} \quad \text{where } l_s = card(CL_s)$$

1.6. Compute for each *chr* of *P* its fitness value *F*(*chr*).

Stage 2:

Repeat

2.1. Order the chromosomes *chr* in *P* from the best to the poor (*in an increasing order of F*).

2.2. Choose the best chromosomes *chr*.

2.3. Attribute for each *chr* of *P*, the observations R_i to the corresponding classes:

$$R_i \in CL_s if \|R_i - g_s\| = \min_{\substack{s=1,C \\ s \neq s'}} \|R_i - g_s\|$$

2.4. Generate randomly the constant f_m ($f_m \in [0.5, 1]$).

2.5. Mutation of all the chromosomes *chr* of *P* except the first one (e*litist technique*):

$$g^*_s = g_s + f_m \times (g^\circ_s - g_s) \times N(0,1)$$

2.6. Attribute for each *chr* of *P* except the first one, the observations R_i to the corresponding classes:

$$R_i \in CL_s if \|R_i - g_s\| = \min_{\substack{s=1,C \\ s \neq s'}} \|R_i - g_s\|$$

2.7. Update the population *P,* for each *chr* of *P* except for the first one, do:

$$g'_s = \frac{g_s + \sum_{R_i \in CL_s} R_i}{1 + l_s} \quad \text{where } l_s = card(CL_s)$$

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

2.8. Compute for each *chr* of *P* its fitness value *F*(*chr*).

Until *Nb_gen* (*generation number*) ! *maxgen*

Figure 4: The proposed EKM algorithm.

5. Determination of the optimal number of classes

5.1. *Xie* and *Beni* criterion

Choosing the right number of classes C, In many partition problems, is a difficult task. Several criteria for choosing the optimal number of classes, based on different approaches, have been proposed in the literature[5,12,13,14,15]. We have retained in this paper the Xie and Beni criterion which is based on a measure of separability and compacity of the classes. The separability and compacity measures have been used to define criteria which have permitted to evaluate the classification performances [3]:

- A compacity criterion is:

$$Comp(C) = \frac{1}{M}\sum_{i=1}^{M}\sum_{s=1}^{C}\left\|R_i - g_s\right\|^2 \tag{10}$$

- And a separability criterion is:

$$Sep(C)=\min_{s\neq s'}\left\|g_s-g_{s'}\right\|^2 \tag{11}$$

Xie and Beni [5] proposed to choose C_{opt} number as that which minimizes the ratio:

$$C_{opt}=\arg\min_C \frac{Comp(C)}{Sep(C)} \tag{12}$$

5.2 Optimization of Xie and Beni criterion by evolutionary strategies

Let F_{XB} be a function which expresses the Xie and Beni criterion:

$$F_{XB}(C, chr) = \frac{\frac{1}{M}\sum_{i=1}^{M}\sum_{s=1}^{C}\left\|R_i - g_s\right\|^2}{\min_{s\neq s'}\left\|g_s - g_{s'}\right\|^2} \tag{13}$$

where chr is a chromosome formed by the centers $(g_s)_{1\leq s\leq C}$.

For a given value of C , $F_{XB}(C,chr)$ only depends on chr. The algorithm for computing $F_{XB}(C,chr)$ value with a fixed value of C, must obtain at convergence the global optimal partition (chr is a global optimum) for which $F_{XB}(C,chr)$ is minimal. Let $f_{XB}(C) = \min_{chr} F_{XB}(C,chr)$ be this value. Xie and Beni criterion consists in choosing C_{opt} such as
$f_{XB}(C_{opt}) = \min_C f_{XB}(C)$. Figure 4 gives an illustration example which shows that if the computation of $f_{XB}(C)$ (with a fixed C) presents local solutions, the Xie and Beni criterion does not obtain the correct optimal number of classes C_{opt}.

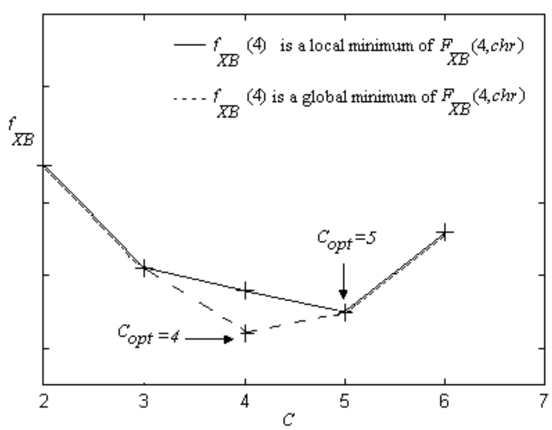

Figure 4: Illustration example.

Thus, to avoid local solutions when computing $f_{XB}(C)$ for a given value of C, we have designed an evolutionist algorithm called *XB_ES*. This algorithm performs the same steps as the *EKM* algorithm, except that the fitness function F is replaced by the function F_{XB}.

The *XB_ES* algorithm runs for several values of C, $C\in[C_{min}, C_{max}]$ ($2\leq C_{min}$ and $C_{max} \ll M$). For each value of C, the algorithm obtains at convergence $f_{XB}(C)$. The optimal number of classes C_{opt} corresponds to the value of C for which $f_{XB}(C)$ is minimal.

6. Experimental results and evaluations

6.1. Introduction

We have considered four simulation tests in the observations space of dimension 2 (N=2). These tests are different from each other by the repartition type of the classes in the observations space. In each test, the classes are generated randomly by Gaussian distributions and each class contains 100 observations.

6.2. Test 1

In this test, the number of classes chosen is C=3 and the overlapping degree between the classes is null. The classes are well separated between them. Table 1 gives the real centers of the classes and figure 5 shows the repartition of the observations in the observations space.

Table 1: Real centers of the classes.

Class	CL_1	CL_2	CL_3
Center Vector	6 3	8 5	4 5

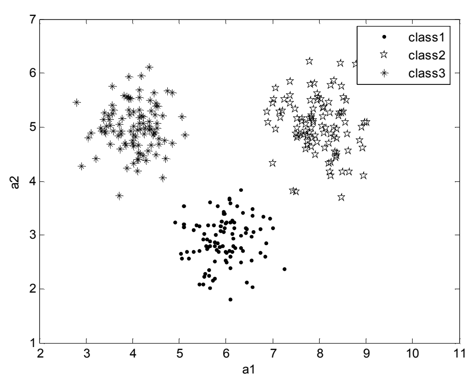

Figure 5: Repartition of the observations in the space.

The proposed evolutionist algorithm runs quickly. Figure 6 shows the evolution of the fitness value of the best chromosome of the current population as long as the generations progress. The optimal chromosome chr_{opt} obtained is:

chr_{opt}= **(5.9641 2.8913 7.9981 5.0404 4.0456 4.9975)** (14)

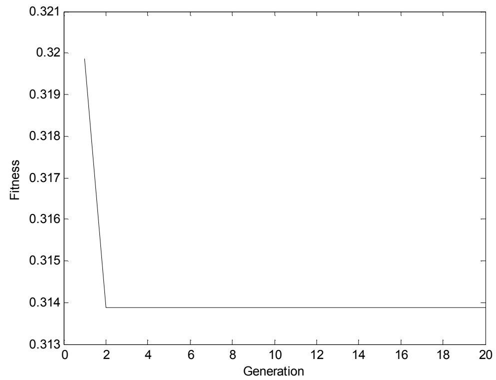

Figure 6: Fitness evolution.

We noticed that in very few generations, the *EKM* algorithm converges to the global optimum and determines the class centers. This is due to the parallel nature of the evolutionist algorithm and also to the nature of the proposed mutation operator which has rapidly guided the algorithm, by means of an adapted Gaussian perturbation, to the global solution. The local solutions have well been avoided. The centers obtained are slightly shifted from the real centers.
The classification results obtained by the proposed evolutionist algorithm are summarized in figure 7 and table 2.

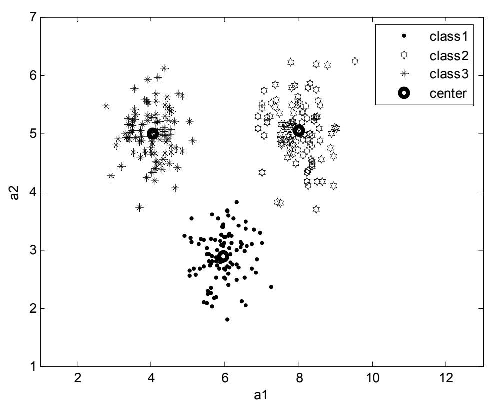

Figure 7: Optimal classes and centers obtained by the EKM algorithm.

Table 2: Confusion matrix.

	Estimated CL_1	Estimated CL_2	Estimated CL_3
CL_1	100	0	0
CL_2	0	100	0
CL_3	0	0	100

These results show that all the observations are correctly attributed to their corresponding classes, the error rate obtained is null.

Thus, we notice that the proposed *EKM* algorithm has improved the performances of the *KM* algorithm. The initialization problem is removed, the result obtained is the same for many different initializations. The proposed mutation operator has permitted to the algorithm to avoid local optimums and to converge rapidly to the global solution.

6.3. Test 2

In this test, we have considered three other classes, but the overlapping degree in this case is high. The classes are very close to each other and have the same centers as the classes of test 1. Figure 8 shows the repartition of the observations in the observations space. We notice that it is difficult to find the optimal partition in this case.

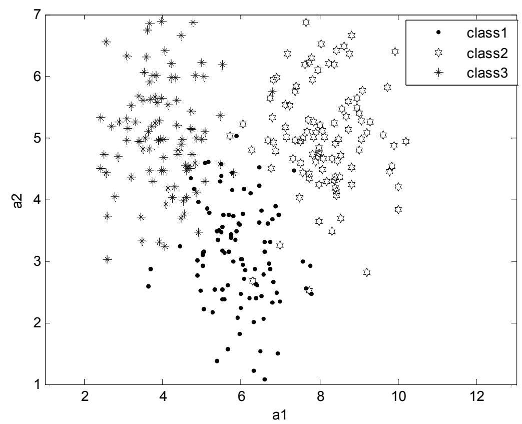

Figure 8: Repartition of the observations in the space.

Figure 9 shows the evolution of the fitness value of the best chromosome of the current population with respect to the progressing generations. It shows that the proposed algorithm converges rapidly to the global solution. The rapidity of the algorithm is not sensitive to the overlapping degree. The optimal chromosome chr_{opt} is obtained:

chr_{opt}= (**6.0230 3.0166 8.1836 5.0796 4.0740 5.0656**) (15)

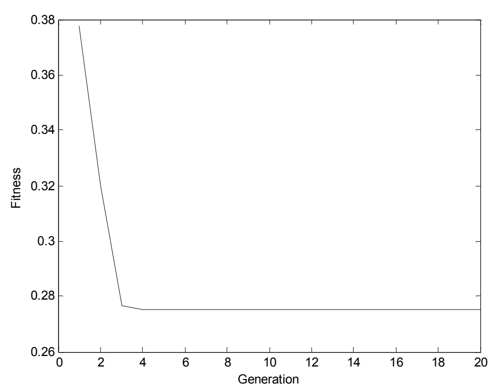

Figure 9: Fitness evolution.

Figure 10 and table 3 summarize the classification results obtained by the proposed algorithm.

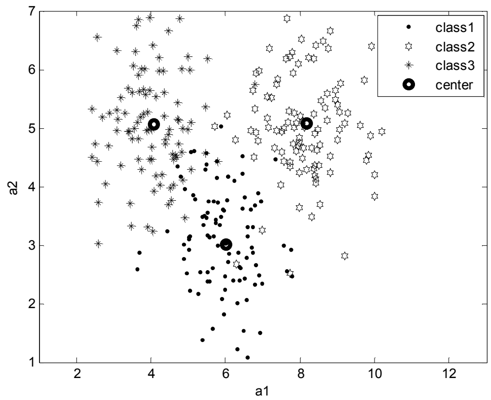

Figure 10: Optimal classes and centers obtained by the EKM algorithm.

Table 3: Confusion matrix.

	Estimated CL_1	Estimated CL_2	Estimated CL_3
CL_1	***91***	***1***	***8***
CL_2	***4***	***94***	***2***
CL_3	***2***	***1***	***97***

The number of misclassified observations in this case is 18. The corresponding error rate is:

$$\tau = \frac{18}{300} = 6\% \qquad (16)$$

The error rate has increased with the overlapping degree. By analyzing the repartition of the classes, we noticed that the misclassified observations are situated:

- Either far away from the space of their corresponding classes, for instance the class CL3 contains 8 observations of class CL1 (figure 8).
- Either in the boundaries of separation between the classes, for instance the boundary which separates the two classes CL2 and CL3 (figure 8).

It is then normal that these observations are misclassified, this explains the high error rate value obtained.

6.4. Test 3

In this test, we evaluate the performance of the algorithm *EKM* for a high number of classes, we chose $C = 6$. The degree of overlap between classes is low. The real centers of 6 classes generated are shown in Table 4, and Figure 11 shows the distribution of observations in the observations space.

Table 4: Real centers of the classes.

Class	CL_1	CL_2	CL_3	CL_4	CL_5	CL_6
Center Vector	6 3	8 5	8 7	4 7	4 5	6 6

The proposed evolutionary algorithm runs quickly. Figure 12 shows the evolution of the fitness value of best chromosome of the current population with respect to the progressing generation generations. The optimal chromosome chr_{opt} is obtained:

$$chr_{opt} = (\mathbf{5.9771 \quad 2.9653 \quad 8.0050 \quad 4.9850 \quad 7.9742 \quad 7.0039 \quad 4.0437 \quad 6.9720 \quad 3.9240 \quad 5.0007 \quad 6.0951 \quad 6.0457}) \qquad (17)$$

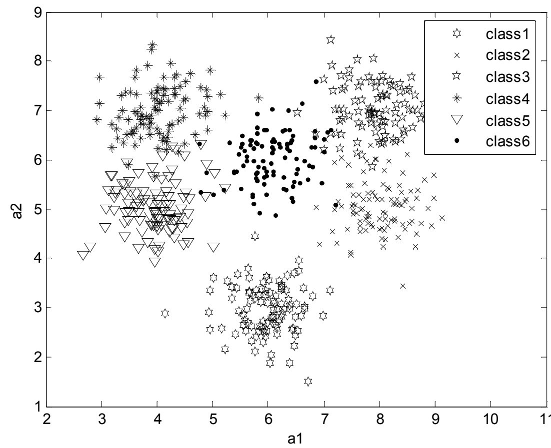

Figure 11: Repartition of the observations in the space.

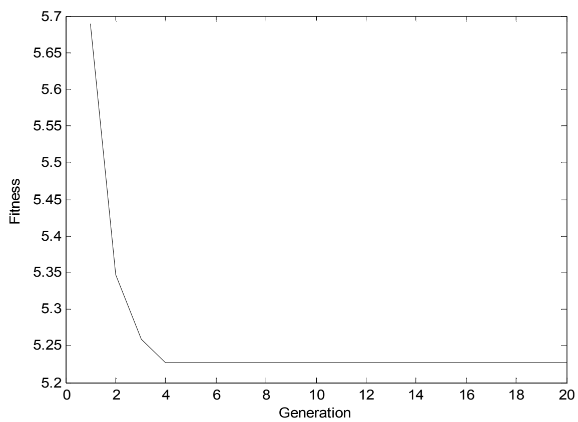

Figure 12: Fitness evolution.

The classification results obtained by the *EKM* algorithm are summarized in figure 13 and table 5.

Table 5: Confusion matrix.

	Estimated CL_1	Estimated CL_2	Estimated CL_3	Estimated CL_4	Estimated CL_5	Estimated CL_6
CL_1	100	0	0	0	0	0
CL_2	0	93	5	0	0	2
CL_3	0	1	96	0	0	3
CL_4	0	0	0	97	2	1
CL_5	0	0	0	5	92	3
CL_6	0	1	3	1	3	92

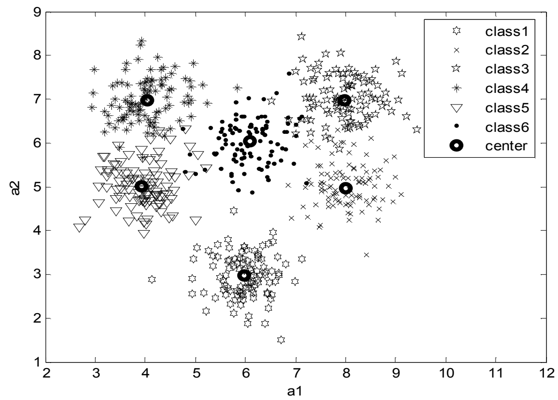

Figure 13: Optimal classes and centers obtained by the EKM algorithm.

The table 5 shows the number of misclassified observations (30 observations, the corresponding error rate is:

$$\tau = \frac{30}{600} = 5\% \qquad (18)$$

The error rate obtained by the algorithm *EKM* remains low, which confirms the good performance.

6.5. Test 4

For this test, the same class centers are taken as for test 3 hawever, the overlapping degree between the classes is high.
Figure 14 shows the repartition of the classes in the observations space, it shows that it is difficult to find the best partition for such a case. The observations of each class are indeed not concentrated around their class center. It is then possible to find observations of a class CLs which are more close to the center of an other class CLs' than they are to their own center (figure 14). These observations are generally misclassified.

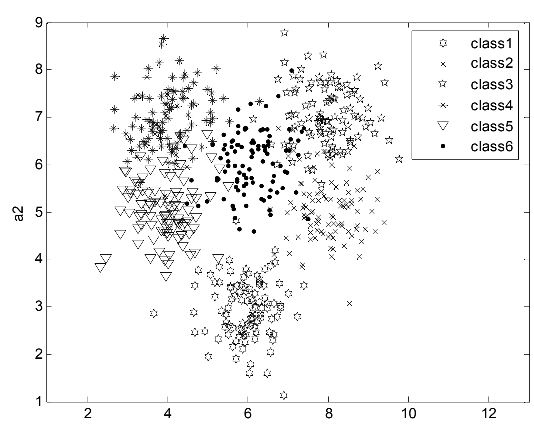

Figure 14: Repartition of the observations in the space.

The proposed *EKM* algorithm converges in a small number of generations (not more than 6) towards the global optimum (figure 15). The optimal chromosome chr_{opt} is obtained:

chr_{opt} = **(6.0089 2.9531 8.0318 4.9672 7.9680 7.0387 4.0193 6.9713 3.8823 4.9214 6.0691 6.0332)** (19)

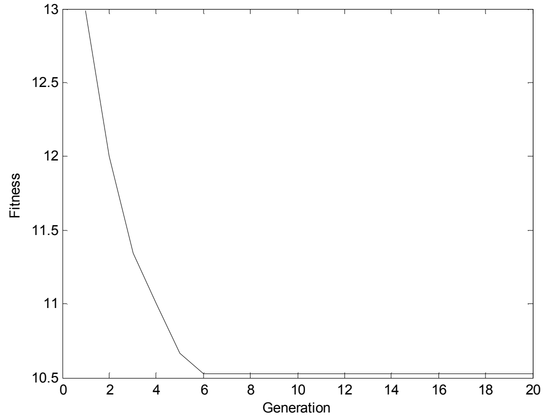

Figure 15: Fitness evolution.

The classification results obtained by the *EKM* algorithm are summarized in figure 16 and table 6.

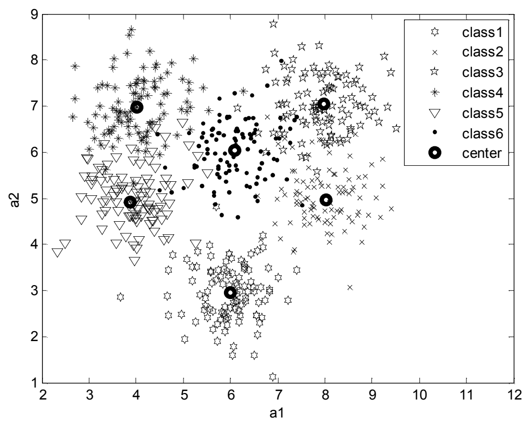

Figure 16: Optimal classes and centers obtained by the EKM algorithm.

Table 6: Confusion matrix.

	Estimated CL_1	Estimated CL_2	Estimated CL_3	Estimated CL_4	Estimated CL_5	Estimated CL_6
CL_1	97	0	0	0	2	1
CL_2	1	88	8	0	0	3
CL_3	0	3	91	0	0	6
CL_4	0	0	0	95	2	3
CL_5	1	0	0	8	87	4
CL_6	0	2	8	1	4	85

The number of misclassified observations is 57, the corresponding error rate is:

$$\tau = \frac{57}{600} = 9.5\% \qquad (20)$$

Whilst the number of classes increases with a high overlapping degree between the classes, the error rate value
obtained remains low. This confirms the good performances of the *EKM* algorithm presented even when the number of classes is high.

6.6. Estimation of the optimal number of classes

We here evaluate the performances of *Xie* and *Beni* criterion optimized by evolutionary strategies. For this, we have retained the four experimental tests presented above. In each test, the *XB_ES* algorithm was run for several values of *C* in [2,6] for tests 1 and 2 , C in [2,10] for tests 3 and 4. Figures 17 to 20 show the evolution of the f_{XB} function with respect to the number of classes *C*.

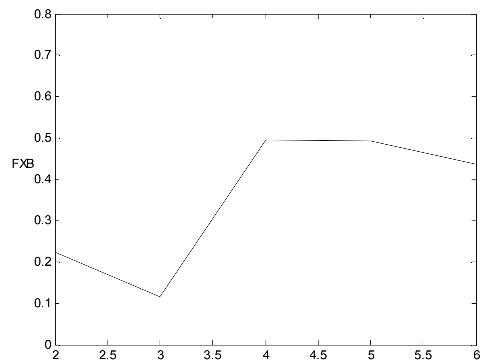

Figure 17: Evolution of fXB with respect to C for test 1, Copt =3.

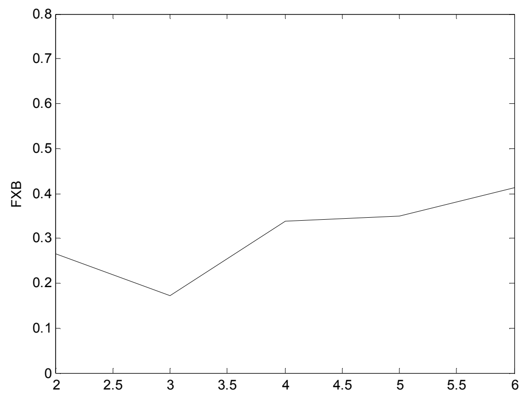

Figure 18: Evolution of fXB with respect to C for test 2, Copt =3.

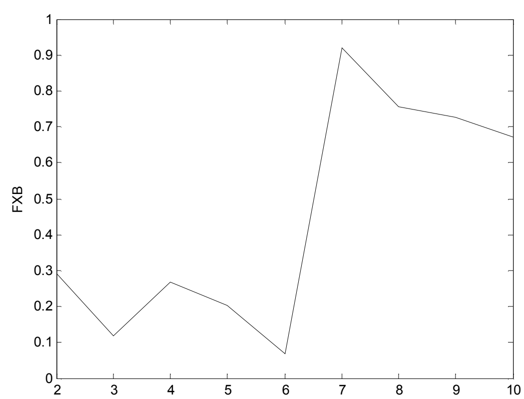

Figure 19 : Evolution of fXB with respect to C for test 3, Copt =6.

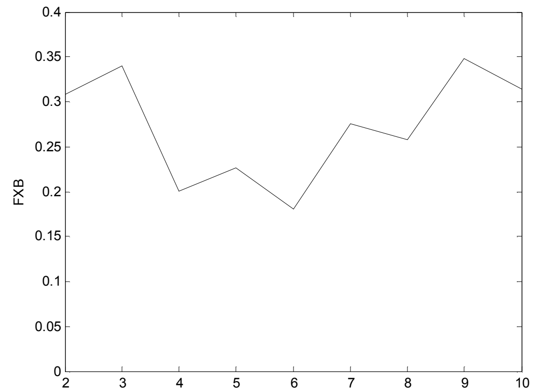

Figure 20 : Evolution of fXB with respect to C for test 4, Copt =6.

The results obtained, for each test, show that the estimated optimal number of classes C_{opt} coincide with the real number C_{real} (i.e., $C_{opt} = C_{real} = 3$ for tests 1 and 2, and $C_{opt} = C_{real} = 6$ for tests 3 and 4). Thus, the optimization of Xie and Beni criterion by evolutionary strategies has permitted to determine successfully the optimal number of classes. This confirms the good performances of the proposed approach.

7. Conclusion

The unsupervised classification by the *KM* algorithm requires the *a priori* determination of the number of classes and suffers from the initialization phase and the local optimums.

We have proposed a new approach to get around these difficulties. Our approach is based on evolutionary strategies and on the *Xie* and *Beni* criterion. We have proposed a new evolutionist *KM* algorithm. We presented a real coding and defined an adequate fitness function suitable for the behavior to be optimized. We proposed a new mutation operator that have permitted to the algorithm to avoid local solutions and to converge rapidly to the global solution. We also optimized the *Xie* and *Beni* criterion by evolutionary strategies in order to estimate correctly the optimal number of classes.

The proposed approach was tested on several simulation examples. The experimental results obtained show the rapidity of convergence and the good performances of this classification method. The optimal number of the classes estimated by the *Xie* and *Beni* criterion coincide with the real one. The two problems of initialization and local optimums are discarded in the *EKM* algorithm.

8. References

[1] Hamad, D., El Saad, S. and Postaire, J.G. "Algorithmes d'apprentissage compétitif pour la classification automatique". TISVA'98, Oujda, Maroc, 27-28 April 1998, pp. 159-164.
[2] Bezdek, J., C. "Cluster validity with fuzzy sets". J. Cybernetics, Vol.3, N°3, pp. 58-73, 1974.
[3] Glorennec, P., Y. "Algorithmes d'apprentissage pour systèmes d'inférence floue". Colloque Neuro-mimétisme, Lyon, juin 1994.
[4] Karayiannis, N., B., Bezdek, J., C. and Hathaway, R., J. "Repairs to GLVQ: a new family of competitive learning schemes". IEEE Trans. on Neural Networks, Vol.7, N°5, pp. 1062-1071, 1996.
[5] Xie, X. and Beni, G. "A validity measure for fuzzy clustering". IEEE Trans. on Pattern Analysis and Machine Intell., Vol. 13, N° 8, pp. 841-847, 1991.
[6] Bezdek, J., C. and Hathaway, R., J. "Optimization of fuzzy clustering criteria using genetic algorithms". In Proc. First IEEE Conf. Evolutionary Computation, Piscataway, NJ: IEEE Press, Vol.2, pp. 589-594, 1994.
[7] Kuncheva, L., I. and Bezdek, J., C. "Selection of cluster prototypes from data by a genetic algorithm". EUFIT, pp.1683-1688, Aachen, Germany, september 1997.
[8] Sarkar, M. et *al.* " A clustering algorithm using an evolutionary-based approach". Pattern Recognition Letters, 1997.
[9] Hall, L., Q., Özyurt, I., B. and Bezdek, J. C. "Clustering with a genetically optimized approach ". IEEE

Trans. on Evolutionary Computation, Vol.3, N°2, pp. 103-110, july 1999.
[10] Presberger, T. and Koch, M. "Comparison of evolutionary strategies and genetic algorithms for optimization of a fuzzy controller". Proc. of EUFIT'95, Aachen, Germany, august 1995.
[11] Renders, J., M., *Algorithmes génétiques et Réseaux de Neurones*. Editions HERMES, 1995.
[12] Hoffman, R. and Jain A., K. "Segmentation and classification of range images". IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 9, N°5, pp. 608-620, 1987.
[13] Nguyen, H., H. and Cohen, P. "Gibbs random fields, fuzzy clustering, and the unsupervised segmentation of textured images". CVGIP, Vol. 55, N°1, pp. 1-19, janvier 1993.
[14] Rissanen, J. "The minimal description length principle". Research Report RJ-8236, IBM US research centers, july 1991.
[15] Olivier, C., Jouzel, F., Matouat, A., El. and Courtellemont, P. "Un nouveau critère pour la sélection de l'ordre d'un modèle". In GRETSI, pp. 451-454, Grenoble, 1997.
[16] 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,vol.5132,pp. 220230,USA, 2003.Repulished as an SME Technical Paper by The society of manufacturingn Engineers(SME). Paper number MV03-233, IDTP03PUB135,
[17] H.Ouariachi, D. Hamad, M.Barboucha, Optimisation de la classification non –Supervisée par la méthode des Nuées Dynamique Evolutionnioste.JTEA2000, pp. 110-115, Mars 2000.Nabeul Hammamet, Tunisie