Optimization of Unsupervised Classification by Evolutionary Strategies

A. EL Allaoui, M Merzougui, M. Nasri, M. EL Hitmy and H. Ouariachi

LABO MATSI, EST, University of Mohammed I, B.P 473, OUJDA, MOROCCO.

ABSTRACT
The kmeans algorithm is an unsupervised classification algorithm. This algorithm however, suffers from two difficulties which are the initialization phase and the local optimums. We present in this paper some improvements to this algorithm based on the evolutionary strategies in order to get around these two difficulties. We have designed a new evolutionist kmeans algorithm. We have proposed a new mutation operator in order for the algorithm to avoid local solutions and to converge to the global solution for a low computational time. This approach is validated on some simulation examples. The experimental results obtained confirm the rapidity of convergence and the good performances of the proposed algorithm.

Keywords
Classification, evolutionary strategies, evolutionist kmeans algorithm, mutation operator

1. Introduction
Classification consists of partitioning a set of objects into groups or classes in such a way that all objects belonging to one same class are all resembling between them and different from objects of other classes. This approach requires both a technique for measuring the resemblance between objects and the choice of an adequate criterion which measures the quality of the obtained grouping of objects. The classification problem becomes then a problem of optimizing a criterion. The kmeans algorithm (KM) is an unsupervised classification algorithm based on this approach [2,3,4], it is widely used for classification problems.

However the KM algorithm suffers from two difficulties which are the initialization phase and the local optimums [5,6,7,8]: this algorithm converges in a finite number of iterations but the obtained solution depends on the initial values chosen for the algorithm, if indeed, we reinitialize the algorithm with a set of other values, it will converge to an other local solution which is entirely different from the first one.

We present in this work some improvements to this algorithm based on the evolutionary strategies. The purpose is to get around the two difficulties shown by the KM algorithm.

We have designed a new evolutionist kmeans algorithm (EKM) which has so 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 unique 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, to a Gaussian perturbation, which makes it then possible to avoid the local solutions.

We have proposed a new mutation operator in order to be able to control the Gaussian disturbance level and to reduce the computation time required to converge towards the global solution.

In section 2, we introduce evolutionary strategies. Then, in section 3, we give some definitions, and we recall the kmeans algorithm. We describe in section 4 our evolutionist kmeans algorithm. While in section 5, the performances of this new method are evaluated by some experimental results. Finally, we give a conclusion.

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 [3].

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]:

<table>
<thead>
<tr>
<th>Random generation of the initial population</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fitness evaluation of each chromosome</td>
</tr>
<tr>
<td>Repeat</td>
</tr>
<tr>
<td>Select the parents</td>
</tr>
<tr>
<td>Update the genes by mutation</td>
</tr>
<tr>
<td>Select the next generation</td>
</tr>
<tr>
<td>Fitness evaluation of each chromosome</td>
</tr>
<tr>
<td>Until Satisfying the stop criterion</td>
</tr>
</tbody>
</table>

Figure 1: Standard ES algorithm.

3. Kmeans classification

3.1 Descriptive elements

Let us consider a set of maxobs objects \( \{O_1, O_2, \ldots, O_i, \ldots, O_{\text{maxobs}}\} \) characterized by \( N \) attributes, grouped in a line vector form \( V = (a_1, a_2, \ldots, a_j, \ldots, a_0) \). Let \( R_i = (a_j)_{1 \leq j \leq N} \) be a line vector of \( R^N \) where \( a_j \) is the value of the attribute \( a \) for the object \( O_i \). Let \( \text{mat} \_ \text{obs} \) be a matrix of \( \text{maxobs} \) lines (representing the objects \( O \)) and \( N \) columns (representing the attributes \( a \)), defined by:

\[
\text{mat} \_ \text{obs} = (a_j)_{1 \leq j \leq \text{maxobs}}
\]

\( V \) is the attribute vector, \( R_i \) is the observation associated with the object \( O_i \) or the realization of the attribute vector \( V \) for this object, \( R^N \) is the observations space and \( \text{mat} \_ \text{obs} \) is the observation matrix associated with \( V \). The ith line of \( \text{mat} \_ \text{obs} \) is the observation \( R_i \). Each \( R_i \) belongs to a class \( CL_s, s=1, \ldots, nbc \).

From a geometric point of view, if we represent each observation by a point in the observations space. From a geometric point of view, if we represent each object by a point in the observations space.

3.2 Kmeans algorithm

The \textit{kmeans} algorithm is one of the most common algorithms used for the classification. We are given \( \text{maxobs} \) observations \( (R_i)_{1 \leq i \leq \text{maxobs}} \) which must be associated with \( nbc \) classes \( (CL_s)_{1 \leq s \leq nbc} \). The centers \( (g_s)_{1 \leq s \leq nbc} \) are line vectors of \( N \) dimension.

The \textit{kmeans} is based on the minimization of the optimization criterion given by [2,3,4]:

\[
J = \frac{1}{2} \sum_{i=1}^{\text{maxobs}} \sum_{s=1}^{nbc} \| R_i - g_s \|^2
\]

where \( \| . \| \) is a distance which is generally supposed to be Euclidean.

The \textit{KM} algorithm supposes that the number of classes \( nbc \) is known a priori.

Figure 2 gives the \textit{KM} algorithm flowchart [3, 4].

1. Fix the number of classes \( nbc \).
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, \textit{if not go to 3}.

Figure 2: Flowchart of the KM algorithm.

4. Evolutionary kmeans classification

4.1 Proposed coding

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

\[
\text{chr} = (g_s)_{1 \leq s \leq nbc, 1 \leq j \leq N}
\]

\( (g_s)_{1 \leq j \leq N} \) are the components of the gs center:

\[
g_s = (g_{s1}, g_{s2}, g_{s3}, \ldots, g_{snbc})
\]

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

\[
g_{sj} \in [\min a_{1 \leq j \leq \text{maxobs}}, \max a_{1 \leq j \leq \text{maxobs}}]
\]

In the evolutionist kmeans algorithm \textit{EKM}, we must discard any chromosome of the initial population having a gene which 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 \( \text{chr} \) be a chromosome of the population formed by the centers \( (g_s)_{1 \leq s \leq nbc} \), for computing the fitness function value associated with \( \text{chr} \), we define the fitness function \( F \) which expresses the behavior to be optimized (J criterion):

\[
F(\text{chr}) = \frac{1}{2} \sum_{i=1}^{\text{maxobs}} \sum_{s=1}^{nbc} \| R_i - g_s \|^2
\]

The chromosome \( \text{chr} \) 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:

\[ chr^* = chr + \sigma \times N(0,1) \] (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. \( \sigma \) is the strategic parameter. \( \sigma \) is high when the fitness value of \( chr \) is high. When the fitness value of \( chr \) is low, \( \sigma \) 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 \( (g_s)_{1 \leq s \leq nbc} \).

Let \( R_i \in CL_s \) if \( \|R_i - g_s\| = \min_{s \in 1 : \ldots : nbc} \|R_i - g_s\| \). i.e. the class consisting of the \( R_i \) observations that are closest to the center \( g_s \). Let \( g_s^o \) be the center of gravity of CLs (figure 3).

\[
\sum_{R_i \in CL_s} R_i = \frac{g_s^o}{l_s} \text{ where } l_s = \text{card}(CL_s) \] (8)

Figure 3: Illustration example in a two dimensional space.

The mutation operator which we propose in this work consists in generating from the old chromosome a new one by:

\[
g_s^o = g_s + f_m \times (g_s^o - g_s) \times N(0,1) \] (9)

where \( f_m \) is a constant multiplicative factor taken to be between 0.5 and 1. The new strategic parameter proposed \( \sigma' = f_m \times (g_s^o - g_s) \) is low when \( g_s \) gets closer to \( g_s^o \)'s and is high when \( g_s \) is far from \( g_s^o \)'s. The \( \sigma' \) 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.
- \( \sigma' \) 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 \( nbc \).

1.2. Generate randomly the population \( P \):
\[ P = \{chr_1, \ldots, chr_n, \ldots, chr_{maxpop}\} \]

1.3. Verify for each \( chr \) of \( P \) the constraint:
\[ g_{aij} \in [\min a_{ij}, \max a_{ij}], 1 \leq i \leq maxobs \]

1.4. Attribute for each \( chr \) of \( P \), the observations \( R_i \) to the corresponding classes:
\[ R_i \in CL_s \text{ if } \|R_i - g_s\| = \min_{s \in 1 : \ldots : nbc} \|R_i - g_s\| \]

1.5. Update the population \( P \), for each \( chr \) of \( P \) do:
\[ g_s^o = \frac{g_s + \sum_{R_i \in CL_s} R_i}{l_s} \text{ where } l_s = \text{card}(CL_s) \]

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:
\[ R_i \in CL_s \text{ if } \|R_i - g_s\| = \min_{s \in 1 : \ldots : nbc} \|R_i - g_s\| \]

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 (elitist technique):
\[ g_s^o = g_s + f_m \times (g_s^o - g_s) \times N(0,1) \]

2.6. Attribute for each \( chr \) of \( P \) except the first one, the observations \( R_i \) to the corresponding classes:
\[ R_i \in CL_s \text{ if } \|R_i - g_s\| = \min_{s \in 1 : \ldots : nbc} \|R_i - g_s\| \]
2.7. Update the population $P$, for each $chr$ of $P$ except for the first one, do:

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

(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 $Nb_{\text{gen}}$ (generation number) $\leq$ maxgen

---

5. Experimental results and evaluations

5.1 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.

5.2 Test 1

In this test, the number of classes chosen is $nbc=3$ 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>
<thead>
<tr>
<th>Class</th>
<th>$CL_1$</th>
<th>$CL_2$</th>
<th>$CL_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center Vector</td>
<td>6</td>
<td>3</td>
<td>8</td>
</tr>
</tbody>
</table>

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 chropt obtained is:

$$chropt = (5.9641 \ 2.8913 \ 7.9981 \ 5.0404 \ 4.0456 \ 4.9975)$$

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 4: The proposed EKM algorithm.

Figure 5: Repartition of the observations in the space.

Figure 6: Fitness evolution.

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

Table 1: Real centers of the classes.

Table 2: Confusion matrix.
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.

5.3 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 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.

<table>
<thead>
<tr>
<th></th>
<th>Estimated</th>
<th>Estimated</th>
<th>Estimated</th>
</tr>
</thead>
<tbody>
<tr>
<td>CL1</td>
<td>91</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>CL2</td>
<td>4</td>
<td>94</td>
<td>2</td>
</tr>
<tr>
<td>CL3</td>
<td>2</td>
<td>1</td>
<td>97</td>
</tr>
</tbody>
</table>

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

\[
\tau = \frac{18}{300} = 6\% \quad (12)
\]

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.

5.4 Test 3
In this test, we evaluate the performance of the algorithm EKM for a high number of classes, we chose \( nbc = 6 \). 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.

<table>
<thead>
<tr>
<th>Class</th>
<th>CL1</th>
<th>CL2</th>
<th>CL3</th>
<th>CL4</th>
<th>CL5</th>
<th>CL6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center Vector</td>
<td>6</td>
<td>3</td>
<td>8</td>
<td>5</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>7</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>
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 $\text{chr}_{\text{opt}}$ is obtained:

$$
\text{chr}_{\text{opt}} = (5.9771, 2.9653, 8.0050, 4.9850, 7.9742, 7.0039, 4.0437, 6.9720, 3.9240, 5.0007, 6.0951, 6.0457) \tag{13}
$$

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

![Figure 11: Repartition of the observations in the space.](image)

![Figure 12: Fitness evolution.](image)

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

$$
\tau = \frac{30}{600} = 5\% \tag{14}
$$

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

5.5 Test 4

For this test, the same class centers are taken as for test 3 however, 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 $\text{CL}_i$s which are more close to the center of an other class $\text{CL}_j$s than they are to their own center (figure 14). These observations are generally misclassified.

![Figure 13: Optimal classes and centers obtained by the EKM algorithm.](image)

![Figure 14: Repartition of the observations in the space.](image)
The proposed EKM algorithm converges in a small number of generations (not more than 6) towards the global optimum (figure 15). The optimal chromosome $chro_{opt}$ is obtained:

$$chro_{opt} = (6.0089 \ 2.9531 \ 8.0318 \ 4.9672 \ 7.9680 \ 7.0387 \ 4.0193 \ 6.9713 \ 3.8823 \ 4.9214 \ 6.0691 \ 6.0332)$$

(15)

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.

<table>
<thead>
<tr>
<th></th>
<th>$CL_1$</th>
<th>$CL_2$</th>
<th>$CL_3$</th>
<th>$CL_4$</th>
<th>$CL_5$</th>
<th>$CL_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$CL_1$</td>
<td>97</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>$CL_2$</td>
<td>1</td>
<td>88</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>$CL_3$</td>
<td>0</td>
<td>3</td>
<td>91</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>$CL_4$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>95</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>$CL_5$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>87</td>
<td>4</td>
</tr>
<tr>
<td>$CL_6$</td>
<td>0</td>
<td>2</td>
<td>8</td>
<td>1</td>
<td>4</td>
<td>85</td>
</tr>
</tbody>
</table>

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

$$\tau = \frac{57}{600} = 9.5\%$$

(16)

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. Conclusion

The unsupervised classification by the KM algorithm suffers from two difficulties which are the initialization phase and the local optimums.

We have proposed in this work a new approach to get around these two difficulties. The new approach is based on the evolutionary strategies. We have presented a real coding and we have defined an adequate fitness function suitable for the behavior to be optimized. We have proposed a new mutation operator which have permitted to the algorithm to avoid local solutions and to converge rapidly to the global solution.

The proposed EKM algorithm was tested on several simulation examples. The experimental results obtained show the rapidity of convergence and the good performances of the presented classification method. The two problems of initialization and local optimums are discarded.

References


Ahmad EL ALLAOUI works on image processing and pattern recognition in MATSI laboratory of the school of Technology, University of Oujda, Morocco. He has his master in the field of computer sciences at University the Med Benabdellah at Fez. His main field of research interest is classification of texture based on new methods (genetic and evolutionary approaches, neural networks, fuzzy logic).

Mohammed Merzougui works on image processing and pattern recognition in MATSI laboratory of the school of Technology, University of Oujda, Morocco. He has his master in the field of computer sciences at the same University. His main field of research interest is classification of texture images based on new methods (genetic and evolutionary approaches, neural networks, fuzzy logic).