Density based clustering with crowding differential evolution

Daniela Zaharie
Faculty of Mathematics and Computer Science
West University of Timişoara
bv. V. Pârvan, no. 4, 300223 Timişoara, Romania
Email: dzaharie@info.uvt.ro

Abstract

The aim of this work is to analyze the applicability of crowding differential evolution to unsupervised clustering. The basic idea of this approach, interpreting the clustering problem as a multi-modal optimization one, is similar to that of unsupervised niche clustering proposed by Nasraoui et al.[10] but instead of evolving only the clusters centers and statistically estimating the other parameters (scales and orientation) we evolve both the centers and the parameters of the clusters. Moreover, to simplify the evolutionary process, especially in the case of high-dimensional data, we evolve only hyper-ellipsoids parallel with the axes. In order to model rotated clusters we used a multi-center representation, i.e. the cluster is covered by many normally oriented hyper-ellipsoids. Besides the fact that it simplifies the evolutionary process this multi-center representation allows describing almost arbitrary shaped clusters. Preliminary experimental results suggest that the proposed approach ensures a reliable identification of clusters in noisy data providing in the same time multi-center synthetic descriptions for them.

1 Introduction

The aim of a data clustering task is to identify homogeneous groups of similar data which satisfies at least the following conditions: (i) data in each group are sufficiently similar; (ii) data in different groups are sufficiently dissimilar. Sufficiently similar and sufficiently dissimilar are subjective notions so a unique rigorous definition of natural grouping of data is missing. Moreover, usually we do not have prior information on the data distribution, thus the clustering problem is an ill-posed one. These explains the absence of a perfect clustering method and the existence of a plethora of algorithms which differs by their underlying basic idea, by their complexity and by their applicability potential[5]. Some algorithms provide a partition of the data in clusters, others provide a set of clusters descriptors which are further used in applying data partitioning rules.

A significant class of algorithms is represented by those which have as underlying idea the fact that the clustering process is an optimization one. There are two main directions here. The first one is that of searching for a set of clusters (data partitioning) which optimizes some clustering goodness criteria. The simplest approach here is that of maximizing the intra-cluster similarity while minimizing the inter-cluster similarity. Unfortunately such an approach could lead to degenerate partitions (e.g. each data defines a cluster) so constraints concerning the number of clusters or the minimal number of elements in each cluster should be introduced. These constraints increase the difficulty of the optimization problem and the clustering algorithm should find a trade-off between different criteria. Approaches based on multi-objective optimization have been recently proposed [7],[3].

A second optimization approach in clustering is based on the idea that clusters are high-density regions and identifying them means finding local maxima of a density function. This approach appears in density-based clustering and two of the most representative algorithms of this type are DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [1] and DENCLUE (Density Clustering) [4]. An advantage of density-based algorithms is the fact that they are able to identify the noise in data.

Since the algorithm presented in this work is related, by the density functions, with DENCLUE we shall briefly describe its basic idea. DENCLUE uses a measure of density constructed based on some influence functions which measure the influence that data have in their neighborhood. A common influence function is the gaussian one, i.e. the influence a data x has on a data y is expressed as $f(x,y) = \exp(-d(x,y)^2/(2\sigma^2))$ where $d(x,y)$ is a dissimilarity measure. The density in x can then be defined as $D(x) = \sum_{y \in \mathcal{N}_x} f(x,y)$, $\mathcal{N}_x$ denoting a neighborhood of x. The clusters are identified by the so-called density attractors defined as local maxima of the density function, $D(x)$. For

a given data the corresponding attractor is determined by a local optimization based on gradient information. In order to identify arbitrarily shaped clusters the concept of high density path is used. The decision if the density is high or low is based on a threshold ξ. This threshold is used also to separate the data from noise and as long as its value is adequately chosen it leads to a successful cleaning of data. However the ability of DENCLUE to identify the true clusters is highly dependent on the parameters σ and ξ.

In order to overcome the difficulty of the optimization process associated to a clustering problem many researchers tried to use evolutionary methods. For a review of evolutionary approaches in clustering see [2]. Most of these approaches address the clustering problem by searching in an evolutionary manner for a data partition which optimizes a clustering goodness criterion . Recently, approaches based on evolutionary multiobjective optimization [3] and multimodal optimization [10] have been proposed.

The approach in this paper is related with the last one and is based on the idea that the clusters descriptors (e.g. centers and scales) can be obtained by identifying through evolutionary niching methods all local maxima of a density function. This idea is similar with that of unsupervised niche clustering introduced in [10].

The unsupervised niche clustering (UNC) algorithm evolves a population of cluster centers by using a genetic algorithm combined with a deterministic crowding mechanism [9] in order to allow the identification of all local maxima. A particularity of this approach is the fact that also the scales and orientation parameters of clusters are adjusted. However their adjustment is not ensured by the evolutionary operators but by estimating them using the current values of the centers. Thus the method is a hybridization between a multi-modal evolutionary algorithm and a statistical estimation method. UNC produces a set of clusters descriptors estimating in the same time the number of clusters. It supposes that data are distributed according to normal distributions, thus each cluster will be a hyper-ellipsoid characterized, in the general case, by a mean and a covariance matrix.

The aim of this work is to analyze the applicability of Crowding Differential Evolution (CDE) [12] in identifying the number and the descriptors of clusters. The present approach starts from the same underlying idea as UNC but these two approaches differ in: (i) the algorithm used in evolving the clusters descriptors; (ii) UNC evolves only the centers while in the proposed algorithm both the centers and the hyper-ellipsoid scales are evolved; (iii) UNC generates one descriptor for each cluster while in our approach a set of descriptors can be associated to the same cluster. The last property is useful especially for clusters which are not necessarily hyper-ellipsoidal.

The reason of trying to use differential evolution in clustering is given by the simplicity and effectiveness of DE algorithms. There are other approaches which use DE in clustering (see for instance [6]) but they are based on interpreting the clustering as a global optimization problem not as a multi-model optimization one.

The rest of the paper is organized as follows. The next section presents some details on applying multi-modal optimization techniques in clustering. In the third section is briefly reviewed the crowding-based differential evolution proposed in [12]. The structure of the proposed algorithm is presented in section four while experimental results on some synthetic and real data are presented in section five. The last section concludes the work.

2 Evolutionary multi-modal optimization in clustering

The key element in the multi-modal approach in clustering is constructing a function whose local maxima corresponds to dense regions allowing the identification of clusters centers.

Let $X = \{x_1, \ldots, x_N\}$ be a set of n-dimensional data, $x_i = (x_i^1, \ldots, x_i^n)$. A natural density function, $f : \mathbb{R}^n \to \mathbb{R}$, is a sum of gaussians:

$$f(x) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{s} \exp\left(-\frac{1}{2} d^2(x, x_i)\right) \qquad (1)$$

where $s > 0$ is a normalization parameter end d is a distance function. Different hypotheses concerning the clusters shapes lead to different distances and normalization parameters, s. The simplest case is when we suppose that all clusters are spherical of radius σ. This case corresponds to the euclidean distance:

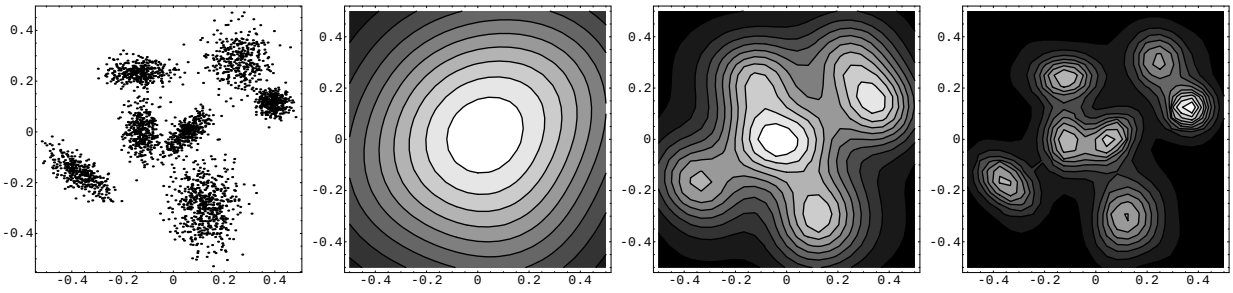
$$d_\sigma^2(x, y) = \frac{1}{\sigma^2} \sum_{j=1}^{n} (x^j - y^j)^2, \qquad s = \sqrt{2\pi}\sigma^{n/2} \qquad (2)$$

When the clusters are considered to be ellipsoids with the same orientation as the axes and having the radii $\sigma_1, \ldots, \sigma_n$ then d is a particular case of the Mahalanobis distance:

$$d_{\sigma_1 \ldots \sigma_n}^2(x, y) = \sum_{j=1}^{n} \frac{(x^j - y^j)^2}{\sigma_j^2}, \quad s = \sqrt{2\pi}(\sigma_1 \sigma_2 \cdots \sigma_n)^{1/2} \qquad (3)$$

The general case corresponds to ellipsoids of arbitrary orientation. In this case the scale and the orientation are described by a covariance matrix, Σ, and d is the general Mahalanobis distance:

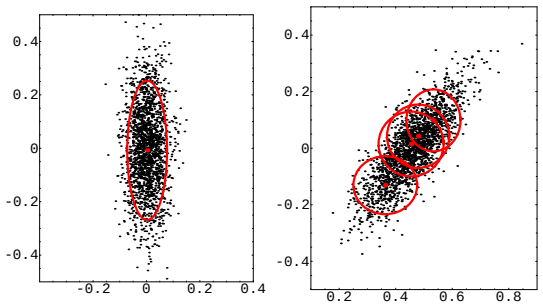
$$d_\Sigma^2(x, y) = (x - y)^T \Sigma^{-1} (x - y), \qquad s = \sqrt{2\pi}(\det\Sigma)^{1/2} \qquad (4)$$



(a) (b) (c) (d)

Figure 1. (a) Normally distributed data; (b),(c) ,(d) Contour plots of the density function for $\sigma = 0.25$, $\sigma = 0.1$, $\sigma = 0.05$, respectively.

Depending on the density function type a cluster of center c can be described by (c, σ), $(c, \sigma_1, \ldots, \sigma_n)$ or (c, Σ). If the clusters to be detected are not necessarily spherical then the first descriptor is not able to capture the cluster structure. The most flexible variant is the last one because it can be used to describe arbitrary orientation hyper-ellipsoids. However when n is large the parameters to be estimated $(n(n+1)/2)$ could be too large. A compromise solution is to use the second variant and a multi-center description. This allow a rotated hyper-ellipsoid to be covered by some hyper-ellipsoids having the same orientation as the axes (see Fig.2) whose scale parameters are easier to be estimated than the covariance matrix Σ.



(a) (b)

Figure 2. Covering normally distributed data with ellipses. (a) Ellipse parallel with the axes: single-center descriptor; (b) rotated ellipse: multi-center descriptor

The clustering problem can be formulated as follows: find a set of cluster descriptors, $\{(c_k, \sigma_k)\}_{k=\overline{1,K}}$, $c_k \in \mathbb{R}^n, \sigma_k \in \mathbb{R}^n$ which approximate the local maxima of the density function. In the following we shall use a density function based on the particular Mahalanobis distance (3). Thus, the density function should be optimized both with respect to the cluster centers, c_k and to their scales, σ_k. Without introducing constraints on the values of the scales the natural tendency will be to identify a single large cluster centered almost in the center of the region to which the data belongs. In order to identify the real clusters the scales values should be constrained to a range $D(\sigma) = [\sigma^1_{min}, \sigma^1_{max}] \times \ldots \times [\sigma^n_{min}, \sigma^n_{max}]$. The most difficult problem is to find the appropriate range. This depends on the number and the size of clusters, information which we do not have apriori. This problem is similar with that of choosing an appropriate common σ (as in the case of DENCLUE) but here the scales values will be further adjusted, thus their range is not so critical as in the case of constant values.

Figure 1 illustrates the first type density functions corresponding to a set of ellipsoidal clusters (a) for different values of the parameter σ ((b),(c) and (d)). As figures 1(b),(c) and (d) suggest, the density function landscape is dependent on the value of σ.

Supposing that the data belong to a domain $D(x) = [x^1_{min}, x^1_{max}] \times \ldots \times [x^n_{min}, x^n_{max}]$ the range for a scale parameter, σ^i will be $[\epsilon, (x^i_{max} - x^i_{min})/\beta]$ with $\epsilon > 0$ the lower bound for the scales range and $\beta > 1$ a parameter controlling the upper bound (an empirical study of the influence of β on the number of detected clusters is presented in section five).

The clustering problem is thus equivalent with that of finding all local maxima (c, σ) of the function

$$g(c, \sigma) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\sqrt{\sigma^1 \ldots \sigma^n}} \exp\left(-\frac{1}{2} d^2_{\sigma^1 \ldots \sigma^n}(c, x_i)\right) \tag{5}$$

subjected to the constraints $c \in D(x)$ and $\sigma \in D(\sigma)$. When N is large the computation of $g(c, \sigma)$ could be extensive. In

such a situation the sum is reduced by taking in consideration only the terms whose values are larger than a threshold. For instance the sum could be only for $i \in A(c)$ where $A(c) = \{x \in D(x) | d_\sigma(c, x) < \theta\}$. In [10] is suggested to choose θ based on the critical values of χ^2 distribution.

In order to identify the local maxima of (5) an evolutionary algorithm which employs a niching mechanism can be used. Such an approach is presented in [10] where a population of centers $\{c_1, \ldots, c_m\}$ is evolved by using a genetic algorithm with gray coding and a deterministic crowding algorithm [9]. After each evolutionary step new values for the parameters $\sigma_1, \ldots, \sigma_m$ are chosen such that they maximizes the density function given in (5). Thus for each population element, c_k, the new values of the parameters $\sigma_k^1, \ldots, \sigma_k^n$ are determined by solving the equations $\partial g(c_k, \sigma)/\partial \sigma^j = 0$ with respect to σ^j. Thus for each population element c_k the new values of the scale parameters are:

$$\sigma_k^j = \sqrt{\frac{2\sum_{i \in A(c)} w_{ik}(x_i^j - c_k^j)^2}{\sum_{i \in A(c)} w_{ik}}} \tag{6}$$

where

$$w_{ik} = \exp\left(-\frac{1}{2} d^2_{\sigma_k^1 \ldots \sigma_k^n}(x_i, c_k)\right). \tag{7}$$

The values of the scale parameters, σ_k^j, used in w_{ik} are the old values, obtained in the previous step. Thus the new values of the scale parameters are obtained from the values c_k corresponding to the current generation and the old values of the scale parameters.

A simpler adjusting relation can be obtained by maximum log-likelihood estimation. Let c_k be a population element and $A(c_k) \subset X$ the subset of data which satisfy

$$A(c_k) = \{x \in X | d_{\sigma_k^1 \ldots \sigma_k^n}(x, c_k) < \theta\} \tag{8}$$

In order to obtain new values for $\sigma_k^1, \ldots \sigma_k^n$ depending on c_k and the current values of σ_k^j we consider the log-likelihood function:

$$\begin{aligned} &L(\sigma_k^1, \ldots, \sigma_k^n) \\ &= \log \prod_{i \in A(c_k)} \frac{1}{\sqrt{2\pi}\sqrt{\sigma_k^1 \ldots \sigma_k^n}} \exp\left(-\tfrac{1}{2}\sum_{j=1}^n \left(\frac{x_i^j - c_k^j}{\sigma_k^j}\right)^2\right) \\ &= \log \frac{1}{(\sqrt{2\pi}\sqrt{\sigma_k^1 \ldots \sigma_k^n})^{\mathrm{card}_{A(c_k)}}} - \tfrac{1}{2}\sum_{i \in A(c_k)} \sum_{j=1}^n \left(\frac{x_i^j - c_k^j}{\sigma_k^j}\right)^2 \end{aligned} \tag{9}$$

Thus

$$\frac{\partial L(\sigma_k^1, \ldots, \sigma_k^n)}{\partial \sigma_k^j} = -\frac{\mathrm{card} A(c_k)}{2\sigma_k^j} + \sum_{i \in A(c_k)} \frac{(x_i^j - c_k^j)^2}{(\sigma_k^j)^3} \tag{10}$$

and the maximum likelihood estimation of σ_k^j is:

$$\sigma_k^j = \sqrt{\frac{2\sum_{i \in A(c_k)}(x_i^j - c_k^j)^2}{\mathrm{card} A(c_k)}}. \tag{11}$$

This adjusting relation is similar with the binarized version proposed in [10] while θ used in (8) is related to the threshold parameter used in [10]. However in [10] no maximum likelihood arguments are given. We shall use the relation (11) to adjust the scale parameters in order to compare the hybrid CDE (similar to UNC) with the simpler fully evolutionary variant.

3 Crowding differential evolution

Differential evolution has been proposed in [11] as an heuristic, inspired by simplex methods, able to efficiently solve difficult optimization problems on continuous domains.

Its particularity consists in the search operator based on an internal perturbation scheme not on an external one as is usual in classical mutation operators. To summarize the particularities of the DE algorithm let us consider a simple unconstrained maximization problem of a function $f : D \subset \mathbb{R}^n \to \mathbb{R}$: find $x^* \in D \subset \mathbb{R}^n$ such that $f(x^*) \geq f(x)$ for all $x \in D$. Let us denote by $X = (x_1, \ldots, x_m)$ the current generation and by y_i the offspring corresponding to x_i.

Different schemes of constructing $y_i = (y_i^1, \ldots, y_i^n)$ starting from the elements of X have been proposed. The most frequently used is:

$$y_i^j = \begin{cases} x_{r_3}^j + F \cdot (x_{r_1}^j - x_{r_2}^j), & \text{with probability } p_c \\ x_i^j, & \text{with probability } 1 - p_c \end{cases} \tag{12}$$

for $j \in \{1, \ldots, n\}$. In eq.(12) r_1, r_2, r_3 are distinct indices randomly selected from $\{1, \ldots, m\}$, $F \in (0, 2)$ is a parameter which controls the magnitude of the perturbation and $p_c \in [0, 1]$ is a probability value which controls the ratio of new components in the offspring. The selection step of classical DE is a simple one: an offspring, y_i, replaces its parent, x_i, only if it is better ($f(y_i) \geq f(x_i)$).

Recently, some DE variants for multi-modal optimization problems have been proposed [12], [14]. The simplest of them is the crowding-based differential evolution (CDE) proposed by Thomsen in [12]. The basic idea of (CDE) is to modify the classical DE by introducing a crowding mechanism in the selection step. The general structure of CDE algorithm is described in algorithm 1. The unique difference between CDE and the classical sequential differential evolution is only in the selection step where the new generated element, y_i doesn't replace x_i but the element in the population which is most similar to y_i. This means that CDE is based on a global crowding mechanism which implies the computation at each step of the distances between the new generated offspring and all the population elements. This global computation is a significant disadvantage only when working with large populations. The effect of this crowding

Algorithm 1 CDE algorithm

```
initialize the population X
repeat
    for all i ∈ {1, ..., m} do
        construct y_i using (12)
        find the element z most similar to y_i
        if f(y_i) ≥ f(z) then
            replaces z with y_i
        end if
    end for
until a stopping condition is satisfied
```

mechanism is that a fixed-size population concentrates on the local maxima of the objective function. In order to extract the approximation of local maxima a post-processing step should be applied. This post-processing step can be based either on computing distances between population elements or on an heuristic valley detection [13].

4 CDE-based clustering

We shall describe an algorithm for finding clusters descriptors (c_k, σ_k) by applying the crowding differential evolution to the objective function (5). Small clusters having the same orientation as the axes could be described by a single descriptor while large or rotated clusters will be finally described by a set of descriptors. The general structure of the algorithm is presented in algorithm 2.

We consider a population of m elements. The population size should be at least large as the number of clusters we expect to detect. Each element of the population is a pair (c, σ), $c \in D(x)$, $\sigma \in D(\sigma)$. The centers are initialized with elements randomly selected from the set of data while the scales are randomly selected in $D(\sigma)$. Both the centers, c_i, and the scales, σ_i, are evolved based on the DE rule (12). A new generated element is accepted only if it satisfies the constraints. Since the scale parameters should be positive when generating new values for them, the relation $x^j_{r_3} + F \cdot (x^j_{r_1} - x^j_{r_2})$ is replaced with its absolute value. Another difference from the original CDE is the distance used in the crowding mechanism: instead of the euclidean distance we used the particilar case of the Mahalanobis distance 3.

After applying the CDE to the population, the clusters and their descriptors are determined by post-processing the obtained population.

Representatives collecting. The first postprocessing step aims to extract some representatives from the population. Each representative, ρ, has three components: the center, $c(\rho)$, the scale parameters, $\sigma(\rho)$ and the label $L(\rho)$. The basic idea of this step is to iteratively construct a set of labelled representatives, $R = \{\rho_1, \ldots, \rho_K\}$ starting from the empty set. The center and the scale of the first representative are initialized with the first element of the population and its label is set to 1. Then for each element, $x = (c(x), \sigma(x))$, of the population starting with the second one the nearest representative, $\rho^* = (c(\rho^*), \sigma(\rho^*), L(\rho^*))$, is determined. This representative is determined by using the Mahalanobis distance between $c(x)$ and $c(\rho)$ with respect to the parameters $\sigma(\rho)$. Considering the distances $d_1 = d_{\sigma(\rho^*)}(x, \rho)$ and $d_2 = d_{\sigma(x)}(x, \rho)$ the following situations are analyzed:

(i) If $d_1 \leq \delta_1$ or $d_2 \leq \delta_1$ then the representative ρ^* should be modified to include the information given by x: $c(\rho^*) = (c(\rho) + c(x))/2$ and $\sigma^j(\rho^*) = \max\{\sigma^j(\rho^*), \sigma^j(x)\}$ for all $j \in \{1, \ldots, n\}$. The label of ρ^* remains unchanged. This situation is illustrated in the bi-dimensional case in Figure 3 (the dashed ellipse corresponds to the new descriptor obtained).

(ii) If $d_1 > \delta_2$ and $d_2 > \delta_2$ then a new representative is generated. Its center and its scale parameters are those of x and its label is a new one, obtained by incrementing the largest existing label.

(iii) In all the other cases a new representative is generated having the same center and parameters as x but the same label as ρ^*.

The parameters δ_1 and δ_2 are some thresholds which defines the influence area of a center and satisfy $\delta_1 < \delta_2$.

Labels refinement. Due to the iterative manner of constructing the set of representatives it is possible that representatives which should describe the same cluster be differently labelled. To avoid such situations the representatives set is repeatedly scanned and for each pair (ρ_i, ρ_k) for which $d_{\sigma(\rho_i)}(\rho_i, \rho_k) \leq \delta_2$ or $d_{\sigma(\rho_k)}(\rho_i, \rho_k) \leq \delta_2$ and $L(\rho_i) \neq L(\rho_k)$ then the label of the worse representative (with respect to the fitness function) is replaced with the label of the better one. This iterative process should continue until no such pairs are found. However it is possible that this never happens. Let us consider, for instance, three representative, ρ_1,ρ_2 and ρ_3 satisfying $d_{\sigma(\rho_2)}(\rho_1, \rho_2) < \delta_2$, $d_{\sigma(\rho_2)}(\rho_3, \rho_2) < \delta_2$, $L(\rho_1) \neq L(\rho_3)$ and $g(c(\rho_1), \sigma(\rho_1)) > g(c(\rho_2), \sigma(\rho_2))$,$g(c(\rho_3), \sigma(\rho_3)) > g(c(\rho_2), \sigma(\rho_2))$. In this situation when is analyzed the pair (ρ_1, ρ_2) then the label of ρ_2 is changed with that of ρ_1 and when the pair (ρ_2, ρ_3) is analyzed the label of ρ_2 is changed with the label of ρ_3 and so on. To avoid such situations, in the current implementation we stopped the iterative process after a maximal number of cycles. A better approach would be to modify the label of ρ_1 or ρ_3 such that they become identical.

Data classification. After the previous steps each cluster will be described either by one representative (uni-center description) or by many representatives (multi-center description). Anyway in order to classify the data, the nearest representative is determined and its label will be given to the data. However if the distance between the data and the nearest representative is larger than a given value, δ_3 then the data is considered to noise (it does not belong to a

Algorithm 2 CDE-based clustering

initialize the population
repeat
  for all $i \in \{1, \ldots, m\}$ **do**
    construct a new c_i
    construct a new σ_i
    find the element (c, σ) most similar to (c_i, σ_i)
    if $g(c_i, \sigma_i) \geq g(c, \sigma)$ **then**
      replaces (c, σ) with (c_i, σ_i)
    end if
  end for
until a stopping condition is satisfied
{Postprocessing stage:}
Collect representatives $\{\rho_1, \ldots, \rho_K\}$ from the population and label them
Refine the representatives labels
Classify the data
Eliminate small clusters and reclassify the data

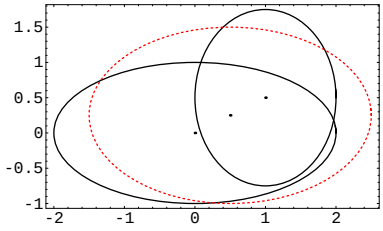


Figure 3. Descriptors merging

cluster). The noisy data could be considered to belong to a separate cluster labelled with a zero value.

Small cluster deletion. The final step is a refinement one used to eliminate some small clusters. The data belonging to clusters which have a number of elements which is less than a given percent of the data number are reassigned to other clusters based on the same idea of the nearest representative.

As the results in the next section illustrate the algorithms behavior is influenced by the values of the parameters δ_1, δ_2 and δ_3 ($\delta_1 < \delta_2 < \delta_3$).

5 Experimental analysis

The aim of the experiments was multiple: (i) to analyze the ability of the CDE-based clustering algorithm to identify clusters and their representatives; (ii) to analyze the influence of the upper bound of scale parameters on the number of identified clusters; (iii) to compare two variants of adjusting the scale parameters: that based on the differential evolution operator (as in algorithm 2) and that based on maximum log-likelihood estimation (in step 5 of algorithm 2 the relation (11) is used).

In the experiments we used both synthetic and real data. The synthetic data are bi-dimensional in order to allow a visual inspection of results. We generated two types of data: (i) based on normal distributions with different covariance matrices (see Figure 4(a)); (ii) based on a uniform distribution in the interior of some geometric figures (see Figure 4(e)). In both cases there are 7 clusters and a uniformly generated noise has been superposed on these clusters.

The real data which we used are classical ones (from UCI Machine Learning Repository, http://www.ics.uci.edu/∼mlearn/MLRepository.html): (i) Iris data (150 data, 4 attributes, 3 classes); (ii) Glass data (214 data, 9 attributes, 6 classes); (iii) Pima data (768 data, 8 attributes, 2 classes); (iv) Thyroid data (215 data, 5 attributes, 3 classes).

Both synthetic and real data have been normalized by dividing each attribute to the difference between the largest and the smallest value corresponding to that attribute.

Even if differential evolution is sensitive to the choice of its control parameters (p and F), preliminary tests suggested that no significant differences are obtained by changing p and F. Thus in all tests we used the values $p = 0.9$ and $F = 0.5$. Concerning the population size and the number of generations most tests have been executed for $m = 20$ and 50 generations. The parameters involved in the postprocessing steps had the following values: $\delta_1 = 1$, $\delta_2 = \sqrt{2}$ and $\delta_3 = \sqrt{5}$.

The results obtained for the synthetic data are illustrated in Figure 4. These results were obtained for both types of synthetic data for $\beta = 20$, $\delta_1 = 1$, $\delta_2 = \sqrt{3}$ and $\delta_3 = \sqrt{5}$. By visual inspection we can see that the CDE-based clustering algorithm is able to identify those seven clusters and associate descriptors to them. However with respect to noise identification the algorithm should be further refined.

One of the most undesirable property of CDE-based clustering (and of UNC algorithm as well) is the dependence of the number of identified clusters on the upper bound of the scale parameters, $(x^i_{max} - x^i_{min})/\beta$. As we would expect for small values of β the algorithm detect a small number of clusters (even one single cluster) and as β becomes larger the number of identified clusters increases. Some results, obtained by 30 independent runs, concerning the dependence of the number of identified clusters on the parameter β are illustrated in Figure 5 for the normal distributed and Iris data. The error bars in the figure indicate, especially in the case of Iris data, that the number of detected clusters has a high variance. In [10] is suggested to use for unsupervised niche clustering $\beta = 2\sqrt{\chi^2_{n,0.995}}/n$ with $\chi^2_{n,0.995}$ a critical value of the chi-squared distribution.

To compare the two variants for adjusting the scale parameters: differential evolution operator or maximum log-

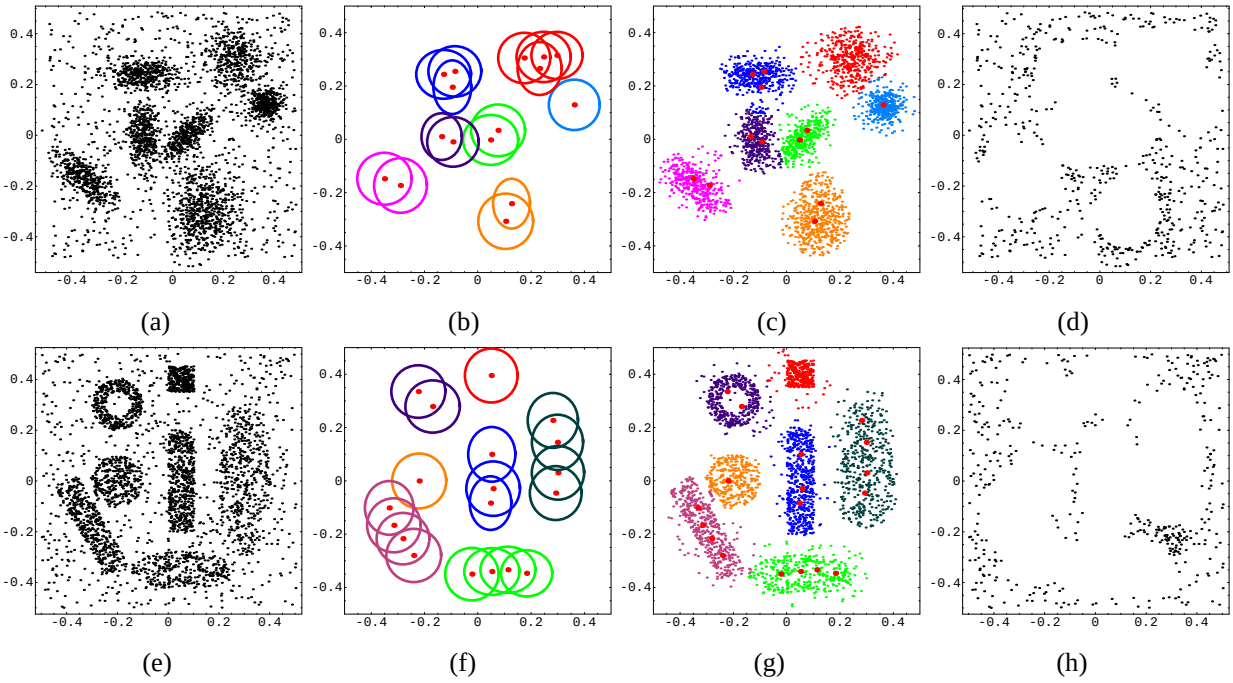


Figure 4. Results for the synthetic data. (a),(e) Initial data; (b),(f) Representatives centers and ellipses corresponding to scale parameters multiplied by δ_2; (c),(g) Classified data; (d),(h) Data classified as noise.

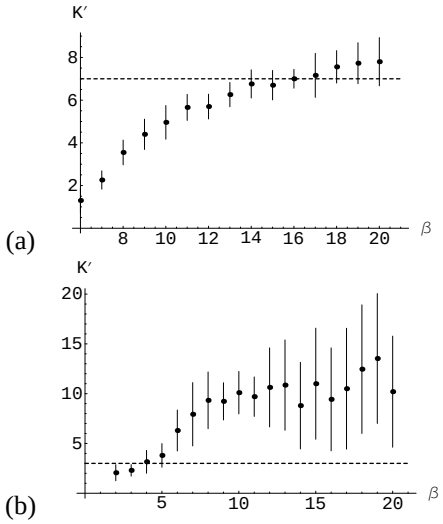


Figure 5. Dependence of the number of detected clusters on the parameter β. (a) Normally distributed data; (b) Iris data.

likelihood estimation we computed the mean and standard deviation of the detected number of clusters and of a classification error for four real data sets (Iris, Glass, Pima and Thyroid). Since for these data the real classification is known we used the error classification error presented in [8]:

$$Err = \frac{2}{N(N-1)} \sum_{i<j} \epsilon_{ij} \tag{13}$$

with

$$\epsilon_{ij} = \begin{cases} 0 & \text{if class}(x_i) = \text{class}(x_j) \text{ and } L(x_i) = L(x_j) \\ & \text{class}(x_i) \neq \text{class}(x_j) \text{ and } L(x_i) \neq L(x_j) \\ 1 & \text{otherwise} \end{cases}$$

Err expresses the ratio of data pairs which are not classified by the algorithm as is expected (data belonging to the same class receives the same label while data belonging to different classes receives different labels). The success ratio in Table 1 expresses the number of cases when the right number of clusters has been detected.

The results in Table 1 suggest that the variant based on maximum log-likelihood estimation for scale parameters is slightly better. However it involves more computations than the variant based only on DE operators and cannot be applied for non-differential density functions.

Data	No. clusters		Succ.	Error		β
	Mean	Std.dev.		Mean	Std.dev.	
CDE-clustering						
Iris	3.20	1.35	8/30	0.19	0.06	5
Glass	6.53	2.61	6/30	0.42	0.01	5
Pima	5.16	1.86	2/30	0.48	0.01	5
Thyroid	2.43	1.76	5/30	0.35	0.04	10
CDE-clustering + maximum log-likelihood estimation						
Iris	3.03	0.87	13/30	0.22	0.05	5
Glass	6.90	2.97	6/30	0.56	0.04	5
Pima	2.86	1.11	9/30	0.47	0.005	5
Thyroid	1.93	1.09	5/30	0.27	0.03	10

Table 1. Experimental results

6 Conclusions and open questions

Interpreting the clustering problem as a multi-modal optimization one can lead to algorithms able to identify both the number of clusters and their descriptors. Based on the idea similar to that of unsupervised niche clustering [10], the CDE-based clustering analyzed in this work allows identifying clusters of arbitrary shapes by allowing multi-center descriptions for them. Multi-center descriptions of clusters offer the possibility of specifying in a synthetic manner arbitrary shaped clusters.

The CDE-based clustering seems to be effective for bi-dimensional noisy data but its behavior is highly sensitive to the constraints imposed on the scale parameters, especially for high-dimensional data. On the other hand the number and size of detected clusters are influenced by the parameters δ_1, δ_2 and δ_3. The values used in the experiments have been empirically established. Finding rules to determine appropriate values for these parameters is an open question.

Besides the simplicity of the approach applying the differential evolution operator both for centers and for scale parameters has the advantage that it can be easily applied also for non-differential density functions. The CDE-based clustering approach can be applied in a similar manner for hyper-rectangles instead of hyper-ellipsoids. This could improve the clustering quality for arbitrary shaped clusters.

References

[1] Ester, M., Kriegel, H.P., Sander, J. and Xu, X., A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, Proc. of the Intern. Conf. on Knowledge Discovery and Data Mining, AAAI Press, pp. 226-231, 1996.

[2] Freitas, A.A., Data Mining and Knowledge Discovery with Evolutionary Algorithms. Springer-Verlag, 2002.

[3] Handl, J., Knowles, J., Evolutionary Multiobjective Clustering, X. Yao et al (Eds.): PPSN VIII, LNCS 3242, pp. 1081-1091, 2004.

[4] Hinneburg, A. and Keim, D.A., An Efficient Approach to Clustering in Large Multimedia Databases with Noise, Proc. of 4rd Intern. Conf. on Knowledge Discovery and Data Mining, AAAI Press, pp. 58-65, 1998.

[5] Jain, A.K., Murty, M.N., Flynn, P.J., Data Clustering: A Review, ACM Computing Surveys, vol. 31, no. 3, 1999.

[6] Krink, T., Paterlini, S., Differential Evolution and Particle Swarm Optimization in Partitional Clustering, Technical Report no. 446, Dept. of Political Economics, Modena, Italy, 2003.

[7] Law, M.H.C, Topchy, A.P., Jain, A.K., (2004) Multiobjective Data Clustering, Computer Vision and Pattern Recognition, vol.2, pp. 424-430, 2004.

[8] Labroche, N., Monmarché, N. and Venturini, G., A New Clustering Algorithm Based on the Chemical Recognition System of Ants, in Harmelen, F. van (Ed.) Proc. of the 15th European Conference on Artificial Intelligence, Lyon, France, pp. 345-349, 2002.

[9] S.W. Mahfoud; Crowding and Preselection Revisited, in R. Männer and B. Manderick (eds.),*Parallel Problem Solving from Nature*, vol. **2**, Elsevier, pp. 27-36, 1993.

[10] Nasraoui O., Leon E., and Krishnapuram R., Unsupervised Niche Clustering: Discovering an Unknown Number of Clusters in Noisy Data Sets, in A. Ghosh and L. C. Jain (Eds.), Evolutionary Computing in Data Mining, Springer Verlag, 2004.

[11] Storn, K. Price; Differential Evolution; A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, Technical Report TR-95-012, ICSI, 1995.

[12] Thomsen, R., Multimodal Optimization Using Crowding-Based Differential Evolution, *Proc. of the IEEE Congress on Evolutionary Computation*, Portland, June 20-23, 2004.

[13] K. Ursem; Multinational Evolutionary Algorithms, in Proc. of the IEEE Congress of Evolutionary Computation, vol. 3, pp. 1633-1640, 1999.

[14] Zaharie, D., A Multipopulation Differential Evolution Algorithm for Multimodal Optimization, in R. Matoušek, and P. Ošmera (eds.), *Proc. of Mendel'04*, Brno, June 16-18, pp. 17-22, 2004.