

A Kernel Based Clustering Algorithm using Particle Swarm Optimization

Bianca Mogoș and Andrei-Horia Mogoș

Abstract. Unsupervised learning is one of the major research areas in machine learning, while kernel methods provide efficient solutions for various statistical learning problems. In this paper we propose a kernel based clustering algorithm that uses the Particle Swarm Optimization technique and discriminant functions. The method represents a general framework for solving the clustering problem: once an appropriate clustering validation index is chosen for a given class of datasets, the method performs very well in solving the problem. The method automatically detects the clusters in a given dataset and also, automatically estimates the number of clusters. Due to the use of kernel functions, our approach can be used for both linearly separable and linearly non-separable clusters. Since our algorithm uses the Particle Swarm Optimization technique, parallel computation may be used, if necessary. We evaluate our method on various datasets and we discuss its capabilities.

AMS Subject Classification (2000). 00A00 ; 99Z99

Keywords. clustering algorithm; kernel function; Particle Swarm Optimization; discriminant function; emergent behaviour

1 Introduction

Clustering is one of the most important and actual topics in the research area of unsupervised learning. A clustering algorithm can be defined as a method of “unsupervised classification” of similar patterns (observations, data items, or feature vectors) into non-similar groups (clusters) one against the other, according to [14] and [25]. The main approaches in unsupervised classification are based on: statistical pattern recognition [14], data mining [23] and density estimation [22].

On the other hand, kernel functions are very popular, because they offer the possibility to project the data from the initial space into a space with a higher complexity in order to improve the analysis of data. Some of the kernels used in the literature are the Gaussian kernel and the polynomial kernel [1], [5]. In particular, in clustering, kernel functions are useful for dealing with linearly non-separable clusters.

In this paper we propose a general kernel based clustering algorithm that uses an heuristic technique, named Particle Swarm Optimization (PSO) and discriminant functions. The most important features of our algorithm are: 1) automatically estimates the number of clusters; 2) can deal with linearly non-separable clusters; 3) extends the supervised approach related to discriminant functions to the unsupervised one; 4) allows the possibility to use parallel computation.

Our approach, based on Particle Swarm Optimization, can be seen in two ways. First, we can say that we use a distributed approach for solving the problem, because each particle of the population used by PSO can be considered a separate entity that computes a potential solution of the problem. Second, since the particles can be seen as very weak agents, our approach is a good example of emergence in multi-agent systems.

The paper is organized as follows. Section 2 contains a related work analysis. In Section 3 we present several preliminary notions, such as kernel functions, discriminant functions, and the Particle Swarm Optimization technique. Section 4 describes our algorithm and discusses some technical issues about the algorithm. In Section 5, we discuss about two clustering validation indexes proposed in the literature. In Section 6, we present several experimental results. Finally, Section 7 contains the conclusions of the paper.

2 Related Work

In the last years, a new clustering approach – Particle Swarm Optimization – was proposed and used in several algorithms.

By studying the results from the literature related to PSO based clustering, one can observe that most of the results use a K-means approach. There are two main ways of using K-means: explicitly (by calling the K-means method) [6], [17], [21] or implicitly (by assigning the points to the clusters, as used in K-means) [2], [4], [5], [12], [24]. All these methods use standard PSO or improved variants of this algorithm in order to obtain better performances for clustering problems.

A less studied approach for clustering the data is the use of kernel functions [5]. An important feature of the kernel based clustering methods is their ability to classify, with a small computational complexity, the datasets that contain linearly non-separable clusters. The method that we propose will illustrate the importance of combining the PSO technique with kernel functions for solving clustering problems. We mention that the method proposed in [5] represents one of the first attempts of combining the PSO algorithm with a kernel based approach.

Usually, clustering methods requires the number of classes in the dataset in order to perform the classification. Thus, they need apriori information about the dataset (see the methods indicated in this section). On the other hand, our method and the approach used in [5] automatically estimate the number of clusters in a dataset.

For all the analyzed methods, the representation of the PSO particles is based on the centroids of the clusters. Our method propose a new representation of the PSO particles that use discriminant functions. This new approach allows the use of discriminant functions - a flexible tool in supervised classification - for solving unsupervised classification problems.

An advantage of the method we propose with respect to the approach proposed in [5] is that the method can be used for a wide range of various datasets: it represents a general framework that allows the use of different fitness functions by considering, for each particular dataset, an appropriate classification validation index. Furthermore, the method has good classification performances for datasets that contain non-spherical clusters, while the approach used in [5] has not a very good accuracy for this type of datasets, as the authors mention.

3 Preliminary Notions

This section presents several preliminary notions necessary for our approach.

3.1 Construction of the Kernel Space

According to the definition, a way of verifying if a function $k : X \times X \rightarrow R$ is a kernel consists in creating a feature space F (a space with inner product) and an embedding mapping in this space, $\phi : x \mapsto \phi(x) \in F$, such that k is expressed as the inner product between any two projections of the initial points through function ϕ , i.e. $k(x, z) = \langle \phi(x), \phi(z) \rangle \forall x, z \in X$.

A general method of creating a kernel space is presented in [25] as follows. Let $k : X \times X \rightarrow R$ be a continuous function (or having a countable domain) and finitely positive semi-definite. From the proof of the kernel characterization theorem results that exists a vector space of functions, with inner product, \mathcal{F} , defined by

$$\mathcal{F} = \left\{ \sum_{i=1}^l \alpha_i k(x_i, \cdot) \mid l \in N, x_i \in X, \alpha_i \in R, i = 1, \dots, l \right\}$$

The space \mathcal{F} can be extended to a Hilbert space F_k , by including all the functions defined as limits of the Cauchy sequences from \mathcal{F} . Consequently, the completeness of the space F_k is satisfied. At the same time, the separability of F_k is guaranteed by the continuous hypothesis of function k or by the countability of the domain. Next, it is considered that the Hilbert space F_k is the feature space and the embedding map ϕ is defined by

$$\phi : x \in X \mapsto \phi(x) = k(x, \cdot) \in F_k \quad (3.1)$$

From the definition of function ϕ and of the inner product follows that

$$\langle \phi(x), \phi(z) \rangle = \langle k(x, \cdot), k(z, \cdot) \rangle = k(x, z).$$

This finishes the construction of the feature space defined by the kernel function k .

3.2 Supervised learning of data using discriminant functions

So far, the discriminant functions approach is integrated as a part of a supervised learning algorithm (see, for example, [26]) and it is discussed in the geometrical methods framework of discriminant analysis [10]. Let $S = \{x_1, x_2, \dots, x_l\}$ be a sample data of size l and let q be the number of the

true classes, denoted by C_1, C_2, \dots, C_q . From a geometrical point of view, the classification consists in partitioning the data space into regions associated to the classes, called discriminant regions.

We define q discriminant functions:

$$g_j(x) : X \rightarrow R, g_j(x) = \langle w_j, \phi(x) \rangle, \forall x \in X, \quad (3.2)$$

$$j = 1, \dots, q$$

where $\phi : x \mapsto \phi(x) = k(x, \cdot) \in \mathcal{F}$ represents the function that project the point $x \in X$ to the feature space \mathcal{F} , defined in Section 3.1, and the weight vector w_j is an element of the feature space \mathcal{F} , of the form $w_j = \sum_{i=1}^l \alpha_i^j k(x_i, \cdot)$.

Thus, the definition of discriminant function, from (3.2), can be rewritten as follows:

$$g_j(x) = \langle w_j, \phi(x) \rangle = \left\langle \sum_{i=1}^l \alpha_i^j k(x_i, \cdot), k(x, \cdot) \right\rangle = \quad (3.3)$$

$$= \sum_{i=1}^l \alpha_i^j k(x_i, x), \forall x \in X, j = 1, \dots, q$$

We have two ways of describing the discriminant functions. The first way, defined by equation (3.2), depends on the weight vector w_j and it is known as the primal method. The second way, illustrated by equation (3.3), describes the discriminant functions through linear combinations of the sample data. It is called the dual method, and the parameters α_i dual variables [25]. The main advantage of the second variant is that the representation of the discriminant functions depends only on the information provided by the kernel matrix K . It is not necessary to compute the inner products in the projection space defined by the function ϕ for each new x data.

The membership of a point x to a cluster $C_i, i = 1, \dots, q$ is expressed as follows [10]:

$$x \in C_i \text{ if and only if } g_i(x) > g_j(x), \quad (3.4)$$

$$\forall j \neq i, j = 1, \dots, q$$

3.3 Particle Swarm Optimization Technique

Particle swarm optimization (PSO) represents an optimization technique based on a population of particles. The approach was proposwd by R. C.

Eberhart and J. Kennedy for solving optimization problems [8] and [15], having as inspiration source the social behaviour of birds and fish.

The optimization problem, that PSO technique wants to solve, can be formulate in the following way [20]:

We search for $X^* \subseteq X \subseteq R^n$ such that

$$X^* = \{\mathbf{x}^* \in X | f(\mathbf{x}^*) \leq f(\mathbf{x}), \forall \mathbf{x} \in X\} \quad (3.5)$$

In this paper, we use the following formulas for updating the velocity and the position of a PSO particle [20]:

$$\begin{aligned} \mathbf{v}_i^{t+1} &= \mathbf{v}_i^t + \varphi_1 \mathbf{U}_1^t (\mathbf{pb}_i^t - \mathbf{x}_i^t) + \varphi_2 \mathbf{U}_2^t (\mathbf{gb}_i^t - \mathbf{x}_i^t) \\ \mathbf{x}_i^{t+1} &= \mathbf{x}_i^t + \mathbf{v}_i^{t+1} \end{aligned} \quad (3.6)$$

where t is the current iteration, \mathbf{x}_i^t is the current position of the particle i , \mathbf{v}_i^t is the particle velocity, \mathbf{pb}_i^t and \mathbf{gb}_i^t are the personal best of the particle and the global best, respectively; \mathbf{U}_1^t and \mathbf{U}_2^t are two random numbers in the range (0,1), and φ_1 and φ_2 are the learning factors. The algorithm repeats until a maximum number of iteration has been exceeded or the obtained error is sufficiently small.

4 The Kernel Based Clustering Algorithm

In this section we describe the main steps of our algorithm and several important technical details.

4.1 Particle Representation

At the beginning of our simulation we consider an initial number of clusters, $q_0, q_0 \geq 2$. Then we define q_0 discriminant functions as follows:

$$g_j(x) = \sum_{i=1}^l \alpha_i^j k(x_i, x) \quad (4.1)$$

Using these functions, each particle p of the population used in PSO, has the form:

$$p = [\alpha_1^1 \alpha_2^1 \dots \alpha_l^1 \dots \alpha_1^{q_0} \alpha_2^{q_0} \dots \alpha_l^{q_0}] \quad (4.2)$$

The parameters $\{\alpha_i^j, i = 1, \dots, l\}$ represents the coefficients of the j discriminant function, for each $j = 1, \dots, q_0$. Thus, during the run of the PSO

algorithm, each particle proposes its own classification of the initial dataset. In the end, due to the abilities of the PSO technique, the algorithm will return the best classification, given by one of the particles of the population.

4.2 A Brief Overview of the Algorithm

The algorithm contains several steps that will be discussed in this subsection.

Step 0: Parameters definition: initial number of clusters q_0 , number of particles, maximum number of iterations $Kmax$, learning factors (ϕ_1, ϕ_2) , maximum inertia ($vmax$)

Repeat for No times, Step 1 - Step 2 and memorize the obtained solution (the particle global best) for each iteration:

Step 1: Particles initialization: initialize the elements of the particles with values in the range $[lmin, lmax]$

Step 2: Applying the Particle Swarm Optimization technique:

2.1: For each particle, initialize personal best value. Initialize global best value for the entire population.

Repeat for $Kmax$ times, steps 2.2 - 2.3:

2.2: For each particle :

2.2.1: Update, according to equations (3.6) of Section 3.3 the velocity and the position of the particle.

2.2.2: Update, if necessary, particle personal best value.

2.3: Update global best value.

Step 3: Of the No solutions obtained above, select the best solution, i.e. the solution that minimizes the value of the fitness function.

Step 4: Find the classification provided by the particle solution returned at the Step 3.

As we can see, the main step of the algorithm is Step 2 that actually uses PSO in order to automatically estimate the number of clusters and perform the classification. The details related to Step 1 are presented in Subsection 4.3. Step 3 is responsible for the quality of the solution: we choose the best solution from the No solutions proposed by the algorithm. This step is

necessary, because it's possible that, on a single run, PSO find only a local minimum.

Fitness Function Algorithm

Let be $p = [\alpha_1^1 \alpha_2^1 \dots \alpha_l^1 \dots \alpha_1^{q_0} \alpha_2^{q_0} \dots \alpha_l^{q_0}]$ a particle of the PSO population.

For each point $x_i \in X, i = 1, \dots, l$ the corresponding cluster can be computed as follows: first, in equation (4.3)

$$\begin{pmatrix} g_1(x_i) \\ g_2(x_i) \\ \vdots \\ g_{q_0}(x_i) \end{pmatrix} = \begin{pmatrix} \alpha_1^1 & \alpha_2^1 & \dots & \alpha_l^1 \\ \alpha_1^2 & \alpha_2^2 & \dots & \alpha_l^2 \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{q_0} & \alpha_2^{q_0} & \dots & \alpha_l^{q_0} \end{pmatrix} \cdot \begin{pmatrix} k(x_1, x_i) \\ k(x_2, x_i) \\ \vdots \\ k(x_l, x_i) \end{pmatrix} \quad (4.3)$$

we show how the discriminant functions will be computed; next, in equation (4.4)

$$C(x_i) = \operatorname{argmax}_{j=1, \dots, q_0} g_j(x_i) \quad (4.4)$$

we compute the cluster $C(x_i)$ of the point x_i . In the end, for each particle, a classification of the initial dataset will be obtained.

Based on the classification $\{C(x_i)\}_{i=1, \dots, l}$ the fitness function is computed by applying a clustering validation index. We will discuss about two such indexes, in Section 5.

4.3 Technical Details Related to the Algorithm

4.3.1 PSO Parameter Selection

In ([16], page 314) the authors recommend to use a population dimension between 10 and 50 particles. As used in an example from ([16], page 318) and in [5] we use a population of 40 particles. As maximum number of iterations $Kmax$ we chose 50 iterations. We observed that this value of $Kmax$ is sufficiently large for the complexity of the clustering problem.

The learning factors have the same value, $\phi_1 = \phi_2 = 2$, as specified in [9] or we may choose $\phi_1 = \phi_2 = 0.5$ if we want a better exploration. The maximum inertia was set to 0.5, because we wanted to encourage a medium modification of the particles positions, and thus to obtain a better exploration.

4.3.2 Automated Estimation of the Number of Clusters

In our simulations we start with a sufficiently large initial number of clusters $q_0 = 10$. During a simulation, in each iteration, each particle is updated and

consequently, some of the discriminant functions (associated to this particle) lower their influence in the clusters discrimination. Thus, the number of clusters is decreased using the following mechanism: when a discriminant function g_j associated to the global best value has the property that

$$\forall x \in S \exists i \in 1, \dots, q_0, i \neq j \text{ such that } g_j(x) < g_i(x) \quad (4.5)$$

then we can consider, that, for the current moment, the cluster corresponding to that function is empty. After the end of the simulation, using the same reasoning, for all discriminant functions with the property (4.5), associated to the final global best, the corresponding clusters will be empty.

The number of clusters is changed during the execution of the PSO algorithm in the sense that, at any step, the number of active classes, denoted by $q_a \leq q_0$, is equal to the number of discriminant functions $\{g_i\}_{i=1, \dots, q_a}$ that satisfy the condition: for any $g_i, i = 1, \dots, q_a$ exists $x \in S$ such that $g_i(x) > g_j(x), \forall j = 1, \dots, q_0, j \neq i$. We mention that the size of the particle remains unchanged, namely q_0 , during the iterations. Our method estimates as the right number of clusters the result obtained after a sufficiently high number of iterations. We validate the precision of our method using the experimental data introduced in Section 6.

4.3.3 Kernel Function - Rational Kernel

During our simulation experiments we have tested several kernel functions: Gaussian kernel, polynomial kernel, all subsets kernel, and rational kernel. We concluded that the most appropriate kernel function for our classification problem is the rational kernel.

One important decision that must be taken when using the rational kernel is how to choose the value of the θ parameter. We searched for a similarity function $k_\theta(\cdot, \cdot)$ with a medium degree of smoothness with respect to the parameter θ . Consequently, for our specific problem, we studied several graphical representations of the rational kernel function for $\theta = 1, 4, 7$, and 10, as seen in Figure 1, and we chose $\theta = 7$.

5 Two Clustering Validation Indexes

Due to the accurate way of measuring the quality of the clusterization and to the high flexibility, we chose two clustering validation indexes, for showing how our algorithm actually works.

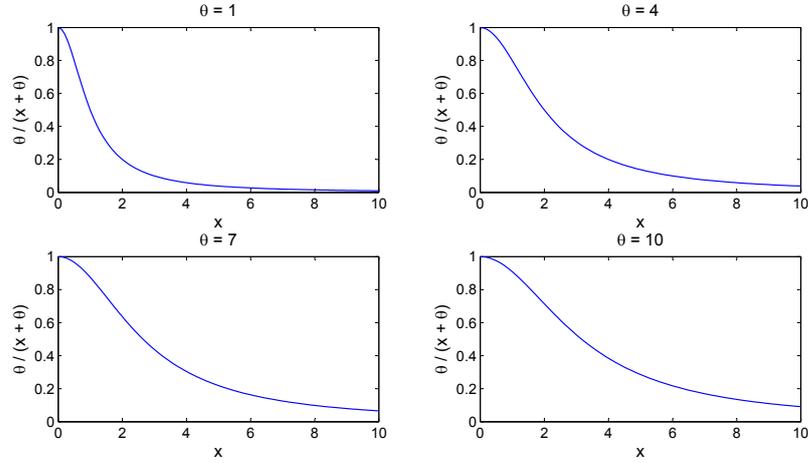


Figure 1: Several graphical representation of the rational kernel for different values of the parameter θ

5.1 Dunn Index

Dunn index represents a well known validity measure for clusterization. Studies related to this index can be found in [13], [18] and [19]. According to [18], this index was first introduced in [7].

Dunn index is defined as follows:

$$D(q) = \frac{\max_{k=1,\dots,q} (diam(C_k))}{\min_{i=1,\dots,q} \left\{ \min_{j=i+1,\dots,q} (d(C_i, C_j)) \right\}} \quad (5.1)$$

where

$$\begin{aligned} diam(C_k) &= \max_{x,y \in C_k} d(x,y) \text{ and} \\ d(C_i, C_j) &= \min_{x \in C_i, y \in C_j} d(x,y). \end{aligned} \quad (5.2)$$

It is considered that the partition $C(q)$ of the data is optimal with respect to the index defined by equation (5.1) if it minimizes the clustering validation index $D(q)$.

Dunn index is an appropriate validation measure for the classification of datasets well separated where the distances between clusters are, usually, large, and the clusters diameters are small. As specified in [18], the main disadvantage of the Dunn index is the fact that it isn't robust to datasets affected by noise, because this type of datasets may contain clusters with large diameters.

5.2 Clustering Validation Index Based on the Symmetry of the Data

A second index used in this paper was proposed in [3] and is based on a symmetry measure. We will call this index, the Symmetry index.

The center of the cluster C_i , denoted by v_i , is defined by the equation

$$v_i = \frac{1}{n_i} \sum_{x \in C_i} x \quad (5.3)$$

where n_i is the number of elements of the cluster C_i .

The Symmetry index is defined as follows [3]:

$$PS(q) = \frac{1}{q} \sum_{i=1}^q \left[\frac{1}{n_i} \sum_{x_j \in C_i} \frac{d_s(x_j, v_i) d_e(x_j, v_i)}{d_{min}} \right] \quad (5.4)$$

where $d_e(\cdot, \cdot)$ represents the euclidian distance between two points, d_{min} is computed using the formula

$$d_{min} = \min_{m, n=1, \dots, q, m \neq n} \{d_e(v_m, v_n)\}$$

and the symmetry distance $d_s(x_j, v_i)$ is defined by

$$d_s(x_j, v_i) = \min_{k=1, \dots, n_i, k \neq j} \{E(x_j, v_i, x_k)\} \quad (5.5)$$

where

$$E(x_j, v_i, x_k) = \frac{\|(x_j - v_i) + (x_k - v_i)\|}{\|(x_j - v_i)\| + \|(x_k - v_i)\|}. \quad (5.6)$$

It is considered that the partition $C(q)$ of the data is optimal with respect to the index defined in equation (5.4) if it minimizes the clustering validation index $PS(q)$.

Limitations of the Symmetry index appear in the following situations:

- when the clusters don't have symmetric structures
- if the dataset contains symmetric clusters with respect to a surface.

6 Experimental Results

For testing our method we use four experimental datasets¹. Experimental dataset 1 is inspired from [5], 2 and 3 from [3], while experimental datasets

¹For a detailed description of the four datasets used in this paper consult <http://aimas.cs.pub.ro/~andrei.mogos/datasets/datasets2012.pdf>

4 are proposed by us in this paper. In Table 1, we present a brief description of the four datasets. We use UD for uniform distribution and ND for normal distribution.

Table 1: Description of the experimental datasets

Dataset	Type	Dataset Dimension	Dataset Structure
Dataset 1	linearly separable	400	100 data, UD
			100 data, UD
			100 data, UD
			100 data, UD
Dataset 2	linearly separable	400	100 data, ND
			100 data, ND
			200 data, ND
Dataset 3	linearly separable	540	100 data, ND
			100 data, ND
			200 data, ND
			70 data, UD
			70 data, UD
Dataset 4	linearly non-separable	250	50 data, UD
			200 data, UD

Table 2 contains the confusion matrices related to the classification obtained using Dunn and Symmetry validation indexes. The element of the confusion matrix from line i and column j indicates the number of data belonging to the class C_i that were predicted in the class C_j . Also, when a validation index produces a correct classification of the dataset, we pointed out that the corresponding confusion matrix is a diagonal matrix.

Several simulation results obtained using the two validation indexes on Datasets 1 – 4 are presented in Figures 2 – 5.

The fourth simulation is made on Dataset 4, that contains linearly non-separable clusters. For this dataset, only Dunn index may be used by our method. Due to the fact that it uses kernel functions, our method can perform the classification of datasets that contain linearly non-separable clusters.

6.1 An Analysis of the Behaviour of our Approach

We present a brief analysis of how our method works on two of the simulations presented above: the third and the fourth simulations. For each of the two simulations we describe how the value of the fitness function applied in gbest

Table 2: The confusion matrices related to the classification obtained using Dunn and Symmetry validation indexes

Dataset\Index	Dunn Index	Symmetry Index
Dataset 1	<i>Diagonal matrix</i>	$\begin{pmatrix} 100 & 0 & 0 & 0 \\ 100 & 0 & 0 & 0 \\ 100 & 0 & 0 & 0 \\ 100 & 0 & 0 & 0 \end{pmatrix}$
Dataset 2	$\begin{pmatrix} 100 & 0 & 0 \\ 100 & 0 & 0 \\ 0 & 0 & 200 \end{pmatrix}$	$\begin{pmatrix} 98 & 0 & 2 & 0 & 0 \\ 0 & 99 & 0 & 1 & 0 \\ 0 & 0 & 199 & 0 & 1 \end{pmatrix}$
Dataset 3	$\begin{pmatrix} 100 & 0 & 0 & 0 & 0 \\ 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 200 & 0 & 0 \\ 70 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 70 \end{pmatrix}$	<i>Diagonal matrix</i>
Dataset 4	<i>Diagonal matrix</i>	$\begin{pmatrix} 196 & 0 & 2 & 2 \\ 0 & 50 & 0 & 0 \end{pmatrix}$

decreases and how *gbest* is modified during the simulation. The analysis of the third simulation can be observed in Figure 6, while the analysis of the fourth simulation can be seen in Figure 7.

7 Conclusions

The techniques used by the approach proposed in this paper, namely: rational kernel functions, Particle Swarm Optimization, discriminant functions, and cluster validation indexes provide an elegant and flexible general method

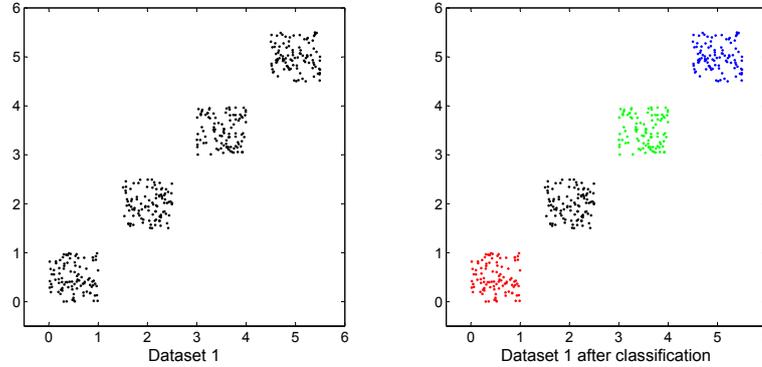


Figure 2: Dataset 1 Classification using Dunn Index

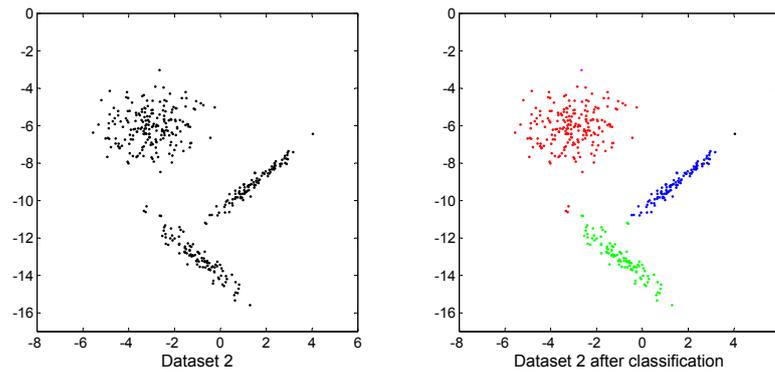


Figure 3: Dataset 2 Classification using Symmetry Index

for the detection of both linearly separable and linearly non-separable clusters.

The method provides a general framework for solving the clustering problem. Once an appropriate fitness function is chosen, in fact the appropriate clustering validation index, the method works well on any dataset that matches with this validation index.

On the other hand, the proposed method automatically estimates the number of clusters. Unlike most of the classification methods, our approach does not require, at the beginning, the actual number of clusters. The algorithm computes, gradually, the number of clusters, starting with an initial number, and then, due to the qualities of the kernel function combined to Particle Swarm Optimization and discriminant functions, it decreases this number, until it

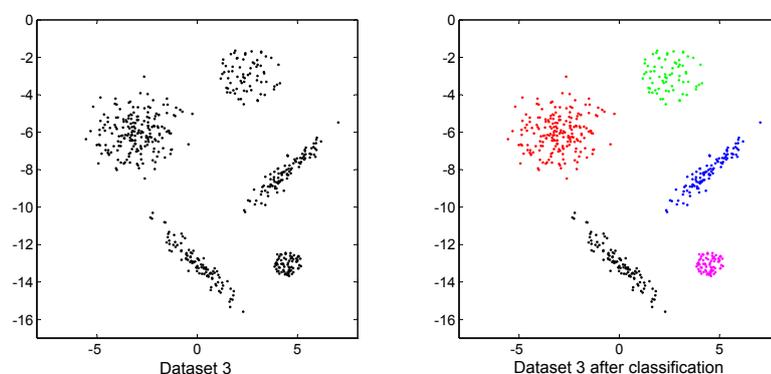


Figure 4: Dataset 3 Classification using Symmetry Index

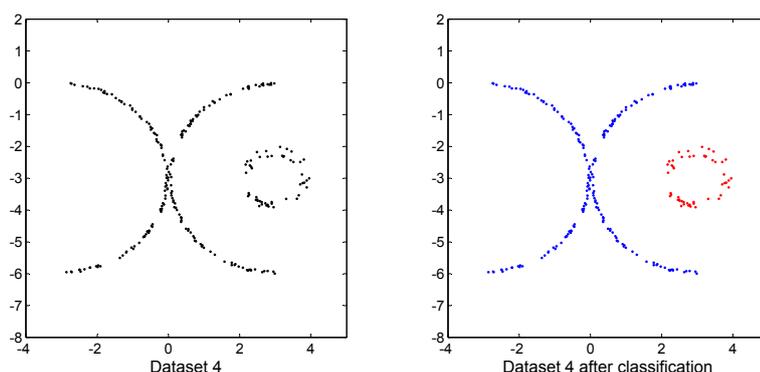


Figure 5: Dataset 4 Classification using Dunn Index

reaches the actual number of clusters. During this process, the approach, also finds the clusters of the dataset.

Furthermore, the developed method performs well on linearly non-separable clusters. Due to the use of a kernel function – we use the rational kernel function – the algorithm can also be used for linearly non-separable clusters. An interesting feature of the method is that it extends discriminant functions based supervised learning to the case of unsupervised learning. By combining the discriminant functions – a powerful tool in supervised learning – with the Particle Swarm Optimization technique, we can use these functions for clustering, thus for an unsupervised learning problem.

From a computational point of view, the use of parallel computation represents another advantage of the proposed method. As specified in ([16], page

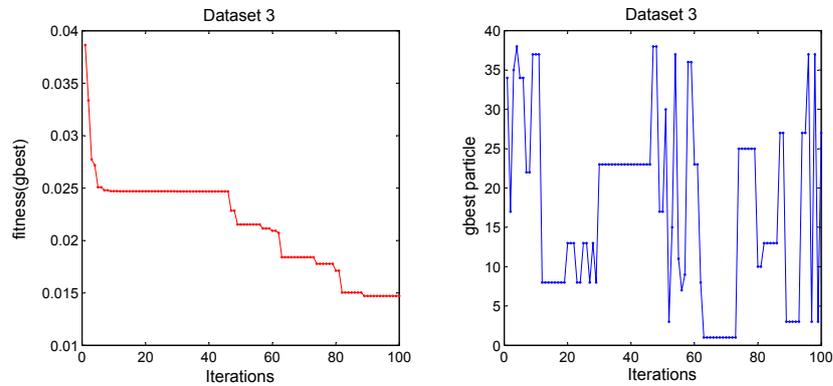


Figure 6: An analysis of the behaviour of the proposed approach for Symmetry index and Dataset 3

314), usually it's sufficient to choose the population dimension, used in Particle Swarm Optimization, in the range 10 to 50. Thus, the required number of processors is at most 50, an accessible number of processors.

As future work, we aim to analyze other clustering validation indexes. The final objective would be to create a library of such indexes that will cover all the possible structures of the initial datasets.

References

- [1] **B. T. Chen and M. Y. Chen**, Applying particles swarm optimization for support vector machines on predicting company financial crisis, *2010 International Conference on Business and Economics Research*, **1**, (2011), 301–305
- [2] **C. Y. Chen and F. Ye**, Particle Swarm Optimization Algorithm and Its Application to Clustering Analysis, *Proceedings of the 2004 IEEE International Conference on Networking, Sensing & Control*, (2004), 789–794
- [3] **C. H. Chou, M. C. Su, and E. Lai**, Symmetry as A New Measure for Cluster Validity, *Proc. 2nd WSEAS Int. Conf. Sci. Comput. Soft Comput.*, (2002), 209–213
- [4] **L. Y. Chuang, Y. D. Lin, and C. H. Yang**, An Improved Particle Swarm Optimization for Data Clustering, *Proceedings of the International MultiConference of Engineers and Computer Scientists*, **I**, (2012), 440–445
- [5] **S. Das, A. Abraham, and A. Konar**, Automatic kernel clustering with a Multi-Elitist Particle Swarm Optimization Algorithm, *Pattern Recognition Letters*, **29**, (2008), 688–699

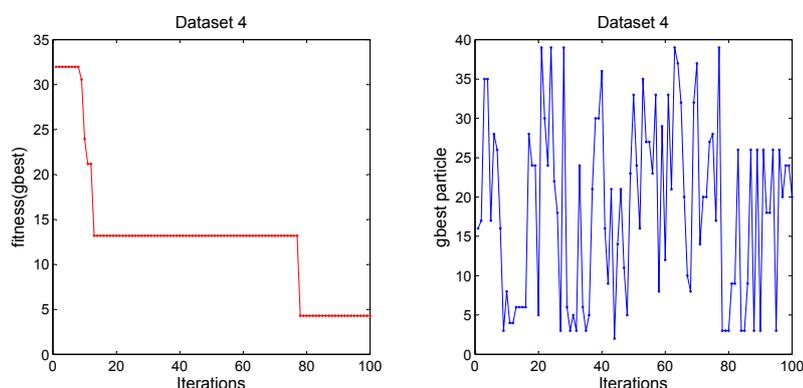


Figure 7: An analysis of the behaviour of the proposed approach for Dunn index and Dataset 4

- [6] **L. Dey and A. Mukhopadhyay**, Microarray Gene Expression Data Clustering using PSO based K-means Algorithm, *Proc. Int. Conf. Advanced Computing, Communication and Networks (ICACCN-2011)*, (2011), 587–591
- [7] **J. C. Dunn**, Well Separated Clusters and Optimal Fuzzy Partitions, *Journal of Cybernetica*, **4**, (1974), 95–104
- [8] **R. C. Eberhart and J. Kennedy**, A new optimizer using particle swarm theory, *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, (1995), 39–43
- [9] **R. C. Eberhart and Y. Shi**, Particle Swarm Optimization: Developments, Applications and Resources, *Proc. IEEE International Conference on Evolutionary Computation*, **1**, 81–86
- [10] **D. Enăchescu**, *Statistical Techniques of Data Mining (in Romanian)*, Editura Universității din București, București, 2003
- [11] **M. G. Genton**, Classes of Kernels for Machine Learning: A Statistics Perspective, *Journal of Machine Learning Research*, **2**, (2001), 299–312
- [12] **N. I. Ghali, N. El-Dessouki, A. N. Mervat, and L. Bakrawi**, Exponential Particle Swarm Optimization Approach for Improving Data Clustering, *International Journal of Electrical and Electronics Engineering*, **3**, (2009), 208–212
- [13] **N. Ilc**, Modified Dunns cluster validity index based on graph theory, *Electrical Review, Poland*, **2**, (2012), 126–131
- [14] **A. K. Jain, M. N. Murty, and P. J. Flynn**, Data Clustering: A Review, *ACM Computing Reviews*, (1999)
- [15] **J. Kennedy and R. C. Eberhart**, Particle swarm optimization, *Proceedings of the IEEE International Conference on Neural Networks*, **IV**, (1995), 1942–1948
- [16] **J. Kennedy and R. C. Eberhart**, *Swarm Intelligence*, Academic Press, USA, 2001

- [17] **A. Khan, N. G. Bawane, and S. Bodkhe**, An Analysis of Particle Swarm Optimization with Data Clustering-Technique for Optimization in Data Mining, *International Journal on Computer Science and Engineering*, **2**, (2010), 1363–1366
- [18] **F. Kovács, C. Legány, and A. Babos**, Cluster Validity Measurement Techniques, *Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases*, (2006), 388–393
- [19] **K. Kryszczuk and P. Hurley**, Estimation of the number of clusters using multiple clustering validity indices, *MCS'10 Proceedings of the 9th international conference on Multiple Classifier Systems*, (2010), 114–123
- [20] **M. A. Montes de Oca**, Particle Swarm Optimization. Introduction, <http://iridia.ulb.ac.be/~mmontes/slidesCIL/slides.pdf>, last accessed February 2012, (2007)
- [21] **M. Neshat, S. F. Yazdi, D. Yazdani, and M. Sargolzaei**, A New Cooperative Algorithm Based on PSO and K-Means for Data Clustering, *Journal of Computer Science*, **8**, (2012), 188–194
- [22] **M. Parimala, D. Lopez, and N.C. Senthilkumar**, A Survey on Density Based Clustering Algorithms for Mining Large Spatial Databases, *International Journal of Advanced Science and Technology*, **31**, (2011)
- [23] **P. Rai and S. Singh**, A Survey of Clustering Techniques, *International Journal of Computer Applications (0975 8887)*, **7**, (2010)
- [24] **S. C. Satapathy, G. Pradhan, S. Pattnaik, J. V. R. Murthy, and P. V. G. D. P. Reddy**, Performance Comparisons of PSO based Clustering, *InterJRI Computer Science and Networking*, **1**, (2009), 18–23
- [25] **J. Shawe-Taylor and N. Cristianini**, *Kernel Methods for Pattern Analysis*, Cambridge University Press, New York, 2004
- [26] **S. Theodoridis and K. Koutroumbas**, *Pattern Recognition 3rd Edition*, Academic Press, 2006

Bianca Mogoş

Faculty of Mathematics and Computer Science

University of Bucharest, Academiei no. 14, Bucharest, Romania

E-mail: broman@fmi.unibuc.ro

Andrei-Horia Mogoş

Faculty of Automatic Control and Computers

University Politehnica of Bucharest, Splaiul Independenţei no. 313, Bucharest, Romania

E-mail: andrei.mogos@cs.pub.ro

Submitted: 20.11.2012

Accepted: 15.03.2013