A New PSO Based Kernel Clustering Method for Image segmentation

Alya Slimene, Ezzeddine Zagrouba
Team of research SIIVA, RIADI Laboratory,
University of Tunis El Manar, High institute of computer science (ISI),
2 rue abou rayhane bayrouni, 2080, Ariana, TUNISIA
alya.slimene@gmail.com
Ezzeddine.Zagrouba@fsm.rnu.tn

Abstract— In this paper a novel kernel clustering method is proposed. The application of the proposed clustering algorithm to the problem of unsupervised classification and image segmentation task is investigated. The proposed method provides a new scheme for classifying objects of one data set without any prior knowledge on the number of naturally occurring regions in the data or an assumption on clusters shapes. It’s based on the use of Particle Swarm Optimization (PSO) algorithm and the use of core set concept which is commonly used to resolve the Minimum Enclosing Ball (MEB) problem. The performance of the proposed method has been compared with a few state of the art kernel clustering methods over a test of artificial data and the Berkeley image segmentation dataset.

Particle Swarm Optimization; unsupervised learning; Kernel methods; image segmentation.

I. INTRODUCTION

Over the last decades, kernel methods [1] [2] have been received a lot of interest since they have been successfully applied in various machine learning problems. The basic concept underlying the Kernel Methods is the nonlinear mapping of the input data to a high dimensional Feature Space through replacing the inner product with an appropriate Mercer Kernel [3]. Support vector machine (SVM) [27] is the most prominent example of kernel methods applied in a supervised framework. In contrast to an unsupervised learning, the supervised context requires a prior knowledge about the number of clusters presented in data sets. Moreover, it depends upon the presence of labeled (preclassified) patterns to perform a training stage in order to label newly encountered, yet unlabeled pattern. Recently new approaches using kernel methods within unsupervised learning have been investigated which can be grouped into three families. The first family is formed by algorithms [4] based on Metric Kernelising idea, i.e. the metric is computed by means of a Mercer Kernel in a Feature Space. The second family [5] [6] involves Kernel Methods that implements K-Means in the Feature Space. The third family encompasses Kernel Methods based on the support vector data description [7] [8] method. This paper focus on studying, and proposing a Kernel Method for Clustering, that belongs to the third family.

The structure of this paper is as follows: the section two provides a review of the necessary background required to effectively implement our algorithm. Section 3 describes the novel proposed method for kernel clustering. Experimental evaluation and results are presented in section 4. Section 5 concludes the paper and gives some perspectives.

II. RELATED WORK

A. Support Vector Data Description

The Support Vector Data Description (SVDD) [13] is a kernel based method which was applied in the context of outlier detection or one class classification. It consists in defining a model which gives a closed boundary around the data. The model is represented by an hypersphere characterized by a center $a$ and a radius $R > 0$. The basic idea of SVDD is to minimize the volume of the sphere by minimizing $R^2$, and demand that the sphere contains all objects $x_i$ of a data set.

With a mapping function $\varphi$ and in a some high dimensional feature space $F$, the primal formulation of the SVDD problem with a soft margin is expressed as follows:

$$\begin{align*}
\min_{R, a, \xi} & \quad R^2 + C \sum_{i=1}^{m} \xi_i \\
\text{subject to} & \quad \|\varphi(x_i) - a\|_2^2 \leq R^2 + \xi_i, \quad i = 1, \ldots, m \\
& \quad \xi_i \geq 0, \quad i = 1, \ldots, m
\end{align*}$$

(1)

where $C$ provides a way to control outliers percentage, allowing the sphere in feature space to not enclose all points of the target class, $\xi_i$ represents slack variables which aim to relax the constraint and $m$ the size of a data set. The corresponding dual is as follows:

$$\begin{align*}
\max_{\alpha_i} & \quad \sum_{i=1}^{m} \alpha_i \varphi(x_i) - \sum_{i,j=1}^{m} \alpha_i \alpha_j k(x_i, x_j) \\
\text{s.t.} & \quad 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, m
\end{align*}$$

(2)
where $\alpha_i, \alpha_j$ are the Lagrange multipliers, and $k(x_i, x_j)$ is an appropriate kernel function. The primal variables can be recovered from the optimal $\alpha$ as

$$a = \sum_{i=1,m} \alpha_i \varphi(x_i)$$  \hspace{1cm} (4)

$$R = k(x_k \cdot x_k) - 2 \sum_i \alpha_i k(x_i \cdot x_k) + \sum \alpha_i \alpha_j k(x_i \cdot x_j)$$ \hspace{1cm} (5)

Where $x_k \in S$, the set of data points lying on a sphere boundary called support vectors and which have $\alpha_k < C$.

B. Support Vector Clustering

Support vector clustering (SVC) is a kernel clustering method based on the support vector concept of the well known SVM method [2] [27]. The main quality of SVC consists, unlike most clustering algorithms published in the literature, in the absence of any assumptions regarding the number or shapes of clusters. Clustering with SVC is done in two stages: the first, called cluster description uses SVDD algorithm to compute the minimum enclosing sphere. The second stage which deals with cluster labeling or assignment uses the computed sphere to differentiate between points that belong to different clusters. This can be performed by the state of art method referred as Complete Graph (CG) [7]. The main idea underlying the CG strategy is to assign two data points $x_j$ and $x_i$ to different clusters if any path that connects them exits from the sphere in the feature space. For this purpose, an adjacency matrix $A_{ij}$ between is built as follows:

$$A_{ij} = \begin{cases} 1 & \text{if } \forall y \text{ on the line segment } x_i x_j, d(y, c) < R \\ 0 & \text{Otherwise} \end{cases}$$  \hspace{1cm} (6)

where $d(y, c)$ represents the distance of a point $y$ from the center $c$ of the sphere in the feature space $F$. Due to the highly intensive complexity of CG strategy which is equal to $O(m^2 p)$ ($p$ represent the number of sampling points on the line segment), it is inefficient to large scale problem. Therefore, different methods to cluster labeling have been developed in order to reduce the complexity of the original one. Among those methods we note Support Vector Graph [7], The Proximity Graph [17], Spectral Graph Partitioning [18], Gradient Descent [19] and Cone Cluster Labeling [20] which is not relied on the path sampling idea to decide about the adjacency of two points.

C. Core-Sets concept for Approximating the Minimum Enclosing Ball (MEB)

The minimum enclosing ball (MEB) problem consists in computing a ball of minimum radius enclosing all the objects of a given dataset. As mentioned in [15], MEB computation has been successfully applied to solve a wide range of application, including gap tolerant classifiers [8] in Machine Learning, tuning Support Vector Machine parameters [10], Support Vector Clustering [4] [3], doing fast farthest neighbor query approximation [17], k-center clustering [5], computation of spatial hierarchies (e.g., sphere trees [18]), and other applications [13]. For a given subset $S$ of objects and a parameter $\varepsilon \in [0..1]$, the corresponding MEB for $S$ is approximated on a small subset $X$, called core-set, such that $X \subseteq S$ and the radius of the MEB(S) is at most $(1 + \varepsilon)$ times the radius of MEB(X). In other Words, X is a core-set if an expansion by factor $(1 + \varepsilon)$ of its MEB contains $S$. Different implementations of MEB computation are proposed in the literature where the estimation of optimal core-set elements and size is also investigated. The core-set size is independent of dimension $d$ and as noted in [16] is at most $2/\varepsilon$ but in the worst-case lower bound is only $1/\varepsilon$.

D. Particle Swarm optimization algorithm

Particle Swarm Optimization (PSO), was introduced by Kennedy and Eberhart [9] [10] by simulating the social behavior of bird flocking or fish schooling. These recent years, PSO has become very popular as an efficient search and optimization technique. In PSO paradigm, the so-called swarm is composed of a set of conceptual ‘particles’ $i$ initialized with position $x_i$ and velocity $v_i$. The position $x_i$ of one particle represents a solution candidate to resolve the considered optimization problem represented by an objective function $f$. The corresponding position and velocity of one particle are adjusted, and the function is evaluated with the new coordinates at each time-step $t$. The basic update equations for the $d$th dimension of the $i$th particle in PSO may be given as

$$v_i^{t+1} = \omega v_i^t + C_1 r_1 (P_i - x_i^t) + C_2 r_2 (P_b - x_i^t)$$  \hspace{1cm} (7)

$$x_i^{t+1} = x_i^t + v_i^{t+1}$$  \hspace{1cm} (8)

The variables $r_1$ and $r_2$ are random positive numbers, drawn from a uniform distribution and taken values in the interval $[0,1]$. $C_1$ and $C_2$ are called acceleration coefficients whereas $\omega$ represent the inertia weight. In addition to that it controls the impact of previous velocities, [12] assert that a large inertia value stimulate an exploration behavior whereas a small inertia weight contribute to exploitation. $P_i$ represents the local best solution found so far by the $i^{th}$ particle, while $P_b$ represents the positional coordinates of the fittest particle found so far in the entire community or in some neighborhood of the current particle. As reported by Van den Begh [11], the inertia weight and acceleration constants are strongly related with the purpose of ensuring the algorithm convergence. They should satisfy the following equation:
\[
\frac{C_1 + C_2}{2} - 1 \leq \omega
\]  

(9)

Fig. 1 gives an illustrative representation of PSO exploration space procedure.

The basic Pseudocode of the PSO algorithm can be given as follows:

**Algorithm 1** PSO basic Pseudocode [9]

Initialize particles population

**while** maximum iteration or required fitness is not attained **do**

- Calculate the fitness of each particle \( i \)
- Update \( P_i \) if the current fitness is better than before
- Determine \( P_n \) from the neighbors

**for** each particle \( i \) **do**

- calculate \( v_i \) according to (7)
- update \( x_i \) according to (8)
- update the best global solution

**end** **for**

**end** While

After the stopping criteria of the PSO algorithm is attained, most of the particles are expected to converge to near global optima of the search space. This convergence is highly dependent to the choice of neighborhood topology. Indeed as mentioned in [33] [32], the PSO topology affects the swarms exploitation and exploration behavior and hence the performance and the convergence of PSO. It exist two topologies structures: the static and the dynamic one. A static structure demand that neighbour’s positions remain fixed throughout iterations. Also the topology structure takes into account the spatial layout of neighbours which can be a spatially random topology, when no connection between a particle’s position and the corresponding neighbour’s positions can be founded. A related spatial topology can take a ring, a star, or a Von Neumann disposition.

### III. THE HIERARCHICAL PSO-SVDD PROPOSED METHOD

However, different methods have been developed to reduce the running time complexity of both SVC cluster description and cluster labeling strategies, the use of SVC to large scale problem and so to image segmentation is bottleneck. From this computational point of view, we propose a novel approach based on PSO that try to overcome the drawbacks of SVC which aim at speeding up the two strategies noted above. Fig. 2 displays the block diagram of the proposed algorithm. It consists of two modules. The first module implements a cluster description strategy to produce the sphere. The second module utilizes the approximated sphere to attribute labels to the sphere interior points.

**A. Proposed Cluster description strategy**

The proposed cluster description strategy provides a new scheme for computing the minimum enclosing ball \( B \). It relies on an incremental expansion of the core set elements by including, at each iteration, a particular point considered as the best point founded during the exploration process. Although, this point can be the furthest away from the current ball’s center as proposed in [21] or be the closest to the expansion of current ball’s by \((1 + \varepsilon)\) factor as done in [14], we suggest using the PSO basic concept for searching point to set the position of this point. In other words, we use the PSO fundamental equations (7) and (8) to get a candidate solution to the optimization problem. Neighbouring particles take a random spatial layout. Noting by \( c \) and \( r \) respectively the center and the radius of \( B \).

Core sets based kernel clustering algorithms have an important step of core set’s initialisation that affects the system performance. For this purpose, [14] and [28] uses a small subset of data referenced as Voronoi sets [35] to initialize the core sets. The proposed cluster description method uses the core sets initialisation strategy presented in [22] which has been proposed to approximate the MEB problem: from a randomly chosen point \( p \), the furthest point, in the feature
space $F$, $p_1$ is computed and added to the empty core set $Q$. The next step require to recomputed in $F$, the furthest point $p_2$ from $p_1$ that will be appended to $Q$. In the sequel, $Q$ contain two points $p_1$ and $p_2$.

The level termination criterion of the proposed method is attained when no expansion of the sphere is detected.

The pseudo code of the resulting cluster description strategy is given in Algorithm 2.

**Algorithm 2** proposed cluster description strategy: PSO_SVDD

Initialization of Core sets $Q$

Repeat

- Compute $B_{c,r} = \text{MEB}(Q)$.
- Assign to current_position the farthest point from $c$ outside the ball $B$.
- Update the particle’s best position ($P_i$) such that will take the last added point to core set $Q$
- Determine the best neighbor position $B_n$ yielding to the ball enclosing the maximum number of points.
- Calculate $V_i$ according to (7).
- Calculate $X_i$ according to (8).
- Find $z$, the nearest point to $X_i$ in the feature space.
- $Q \leftarrow Q \cup \{z\}$

Until Convergence

**B. Proposed Cluster labeling strategy**

For cluster labeling and for a user-specified maximum number of levels $L_{\text{Max}}$, we adopt an incremental scheme that executes the cluster description strategy until the $L_{\text{Max}}$ is achieved or the convergence criterion is attained. This convergence is reached when the number of training point falling outside the ball is less than $C \times m$ where $m$ represents the size of data set and $C$ is the trade off parameter. This strategy provides a way to distinguish the examples belonging to a single class from the examples in all remaining classes. For example, if we set the $L_{\text{Max}}$ to 1, this allows for a binary classification or an extraction of foreground from background in image segmentation context. Indeed, from one level to an other the complexity time is reduced as soon as the number of examples is scaled down. The resulting cluster labeling is given in Algorithm 3.

**Algorithm 3** proposed cluster labeling strategy

Repeat

- PSO_SVDD
- Assign points that are inside the sphere to cluster L
- Update the set of remaining data points to contain those which are outside the sphere
- Increment $L$

Until $L \geq L_{\text{Max}}$ or stopping criterion

**C. Time complexity analysis**

As illustrated in Fig. 3, $r_0$ and $c_0$ denote respectively the radius and the center of the first approximated sphere $S$. $q$ represent the last point added to core set $Q$. $p_2$ and $p_1$, which are outside the sphere $S'$, represent respectively the closest point to $S'$ and the best neighbor position to point $q$ among the n-candidates. $S'$ is the $(1+\varepsilon)$ expansion of $S(c_0, r_0)$. Let $q_a$ be the point such that $qq_a = qp_1 + qp_2$.

![Figure 3. Best point exploration procedure](image)

At the first iteration of and by the triangle inequality, we have

$$\|q p_2\| \geq \|c_0 p_1\| - \|c_0 q\| \geq (1+\varepsilon) r_0 - r_0 \geq \varepsilon r_0$$

From other hand we have $\|q p_2\| \geq (1 + \varepsilon) r_0$. So by triangle inequality we have:

$$\|q p_a\| = \|q p_2\| + \|q p_1\| \geq (1 + 2 \varepsilon) r_0 \geq (1 + \varepsilon) r_0$$

Likewise it can be proved that the radius of one computed sphere at iteration $i$ is dependent of the one computed at previous iteration. So $r_{i+1} = (1+\varepsilon)r_i$. At the second iteration,

$$\|q p_a\| \geq (1 + \varepsilon) r_i$$

then $\|q p_a\| \geq (1 + \varepsilon)(1 + \varepsilon) r_i$.

In the sequel, at the $i^{th}$ iteration we have $\|q p_a\| \geq (1 + \varepsilon)^i r_0$.

So we can deduce that the time complexity $C$ of the final core set construction to approximate the sphere for the $N$ points is $O(i)$.

Thus by variable substitution, we note that

$$i = \frac{\log \left( \frac{N}{r_0} \right)}{\log (1 + \varepsilon)} \simeq \log (\log (N))$$. Hence $O(i) = O(\log (\log (N)))$.

For one particle the complexity of the proposed algorithm is detailed as follows:

The time complexity of the core set initialization step is $O(N)$, the time complexity of the final core set construction is
and time complexity of SVDD is $O\left(\frac{1}{\varepsilon^4}\right)$ [28].
So the time complexity $T$ of the proposed algorithm is

$$T = O(N) + O\left(\log\left(\frac{N}{\varepsilon}\right)\right) + O\left(\frac{1}{\varepsilon^4}\right)$$

whereas it’s equal to

$$O\left(\frac{1}{\varepsilon^2}N + \frac{1}{\varepsilon^4}\right)$$
in scaled-up KG [28] [14].

D. Applications

The proposed algorithm discussed in the previous section is independent of the domain on which can be applied and hence can be used in wide variety tasks of vision’s domain. Fig. 4 displays the bloc diagram of the proposed method application to image segmentation task.

![Bloc diagram of the hierarchical PSO-SVDD based image segmentation algorithm](image)

The region cleaning module which is an optional step aims at removing small regions containing less than $M$ pixels. The main objective of the two optional modules is to reduce the number of small regions and therefore to counteract the effect of over segmentation results.

IV. EXPERIMENTS

In order to show the effectiveness of our algorithm, experiments have been carried out on synthetic and natural images selected from the Berkeley image segmentation database [26]. The scaled-up KG method [14] was used to provide a basis of comparison for the hierarchical PSO-SVDD method. The proposed method was implemented in Matlab. For parameter tuning of the hierarchical PSO-SVDD we set the value of cognitive acceleration coefficient to 2.8 and social acceleration coefficient to 1.3 as recommended in [27]. The parameter $C$ is empirically set to 0.2. We use the Gaussian kernel as a mapping function to high dimensional space as recommended in [8].

A. Experiments on synthetic data

As synthetic data, the delta set [25] is used. It consists of 424 points represented in two dimensional space. The main characteristic of the delta set is that’s have a two nonlinearly separated classes represented both in semi spherical shape with same center and different radius.

Two Experiments have been conducted on delta set: the first carried out on the original dataset size and the second on a scaled up dataset size. The first performed proves that hierarchical PSO-SVDD gives the same accuracy as scaled-up KG with less computational time. The second experiment aims at studying the accuracy and the CPU time when various delta set size and different value of $\varepsilon$ are explored. The size of the data sets was varied from 4240 to 42400. The points were randomly drawn around the two semi-circles. The radiuses of the semi-circles were set as 0.5 and 1.0; respectively. The value of $\varepsilon$ varies from 0.1 to 0.5. The sigma parameter of Gaussian kernel was set to 0.5.

As shown in Fig. 5, hierarchical PSO-SVDD is significantly faster than scaled-up KG and is particularly insensitive to $\varepsilon$ parameter. Indeed, for a delta set of size 42400, the execution time of scaled-up KG for the best value of $\varepsilon$ is about 90s whereas it doesn’t exceed 2s with the proposed approach. An other quality of the hierarchical PSO-SVDD is that unlike scale-up KG the accuracy is non dependent to the value of $\varepsilon$. Besides, for different value of $\varepsilon$, the accuracy rate is still unchanged.

![CPU time vs size of data sets comparison for scaled-up KG (right) and hierarchical PSO-SVDD (left) for different value of $\varepsilon$](image)
B. Experiments on Berkeley Segmentation Dataset

For image segmentation, the Berkeley Segmentation Dataset [26] is used to evaluate the algorithm’s performance. The dataset contains 300 color images organised in two separated sets, a training set composed of 200 images which is frequently used to tune the parameters of a segmentation algorithm, and a testing set containing the remaining 100 images on which the final performance evaluations should be performed. The dataset was also provided with a set of human segmented images. Each pixel of an input image has been transformed into 3D vector in the CIE \( L^*u^*v^* \) color space. The LUV color space was chosen since from one hand it was designed as the best one that approximate the human perception and from other hand it’s characterized by a linear mapping property [34]. Fig 6 displays for some selected image, the different levels of a target image segmentation with the proposed method.

Fig. 7 provides a comparison of segmentation results of the proposed method and scaled-up KG [14]. The segmented images with the proposed method are generally well segmented into regions of homogeneous color and which are perceptually meaningful to human’s vision. In the sequel, the hierarchical PSO-SVDD algorithm has been able to achieve a good segmentation and can detect, automatically, very well the number of region.

To ensure an objective evaluation of the quality of our image segmentation algorithm, we have focused on the entire Berkeley data set and not on a limited number of images such in [14]. For this purpose, we have tuned for all the dataset and empirically, the value of Gaussian kernel width to 8.5 and the value of Gaussian filter to 5. As an objective evaluation metric, the Probabilistic Rand (PR) index [29] has been used since it provides a realistic relationship among segmentations i.e can distinguish perfectly between impractical and acceptable segmentations for a given image target and the set of corresponding human-generated segmentations. The unique drawback of the index is to suffer from small ranging values and hence can be difficult to select the best machine-generated segmentation result. The PR index provides a framework to compare an image segmentation result to its corresponding set of manually segmented one. The PR index takes the following formula:

\[
PR\left( S_{test}, \{ S_k \} \right) = \frac{1}{(N^2)} \sum_{i,j} \left[ c_{ij}p_{ij} + (1 - c_{ij})(1 - p_{ij}) \right]
\]

where \( S_{test} \) represent the segmented image to evaluate, \( S_k \) denote the set of manually segmented image corresponding to \( S_{test} \). \( c_{ij} \) is the event that pixels \( i \) and \( j \) have the same label and \( p_{ij} \) its probability. This measure takes values in \([0, 1]\), where 0 imply that \( S_{test} \) and the set of \( \{ S_k \} \) have no similarities and 1 when all segmentations are identical.

<table>
<thead>
<tr>
<th>(a)</th>
<th>(b)</th>
<th>(c)</th>
<th>(d)</th>
<th>(e)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Original image" /></td>
<td><img src="image2" alt="Segmentation stage 1" /></td>
<td><img src="image3" alt="Segmentation stage 2" /></td>
<td><img src="image4" alt="Segmentation stage 3" /></td>
<td><img src="image5" alt="Final segmentation map" /></td>
</tr>
<tr>
<td><img src="image6" alt="Stage 1" /></td>
<td><img src="image7" alt="Stage 2" /></td>
<td><img src="image8" alt="Stage 3" /></td>
<td><img src="image9" alt="Stage 4" /></td>
<td><img src="image10" alt="Stage 5" /></td>
</tr>
<tr>
<td><img src="image11" alt="Stage 1" /></td>
<td><img src="image12" alt="Stage 2" /></td>
<td><img src="image13" alt="Stage 3" /></td>
<td><img src="image14" alt="Stage 4" /></td>
<td><img src="image15" alt="Stage 5" /></td>
</tr>
<tr>
<td><img src="image16" alt="Stage 1" /></td>
<td><img src="image17" alt="Stage 2" /></td>
<td><img src="image18" alt="Stage 3" /></td>
<td><img src="image19" alt="Stage 4" /></td>
<td><img src="image20" alt="Stage 5" /></td>
</tr>
</tbody>
</table>

Figure 6. Segmentation stages: (a) Original image, (b)- (d) different stages of segmentation procedure, (e) final segmentation map.
Fig. 7. Experimental results on selected images from Berkeley image database. (a) Original image. (b) Scaled-up KG. (c) Hierarchical PSO-SVDD, (d) to (g) represent their ground truth.

Fig. 8 shows the distribution of the PRI measure over the 300 images of the Berkeley database for the proposed algorithm. We can observe that a good PRI performance measure can be obtained without any optimization step of an error term and without prior knowledge or constraint on the number of regions. Also, the Fig. 8 shows that our algorithm has a peak at PR score of 0.8; this is can be explained as follows: on average 80% of pairs of pixel labels are correctly classified in the segmentation results. As reported in [30], this implies that there exist a high correlation between the algorithm segmentation result and human hand-segmentations. Besides, in order to provide a basis of comparison for the proposed method, we make use of NCuts [36] and Mean shift [31] segmentation methods. The PRI averages of both methods are presented in [37]. The PRI average obtained with hierarchical PSO-SVDD is 0.75 which is equal to mean Shift and outperforms the PRI of NCuts (0.72).
V. CONCLUSION AND PERSPECTIVE

In this paper, we have presented and tested a new Kernel method for image segmentation. The main idea underlying the proposed method was to investigate a new approach using PSO within SVDD for image segmentation. The fundamental quality of the proposed technique is that it is able to find the optimal number of clusters automatically without a prior knowledge about the number of regions is required. Moreover, it’s contribute to speeding-up cluster description and labeling stages and isn’t sensitive to $\epsilon$ parameter.

Further work could use more specific probabilistic kernels such as Mixture Model Kernel [23] [25] and investigate to develop a mechanism that learns automatically from data the parameter of kernel function.

REFERENCES