2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

A Growing Partitional Clustering Based on Particle Swarm Optimization

Nuosi Wu, Zexuan Zhu and Zhen Ji
Shenzhen City Key Laboratory of Embedded System Design,
College of Computer Science and Software Engineering,
Shenzhen University, Shenzhen, China 518060

***Abstract*—This paper proposes a growing partitional clustering method based on particle swarm optimization (PSO) namely PSOGC for handling data with non-spherical or non-linearly separable distribution. Particularly, PSOGC uses PSO to optimize the cluster centers. In each iteration of PSO, the particles encoding candidate cluster centers are evolved according to their social and personal knowledge. Given the candidate cluster centers, a growing strategy increasingly absorbs nearby data samples into the corresponding cluster based on k-nearest neighbor graph. The fitness of each particle is evaluated in terms of intra-cluster connectivity and inter-cluster disconnectivity of the resultant clustering. The combination of PSO and growing strategy ensures the stability of global search and the robustness of partition on data of different non-spherical shapes. Experimental results on six synthetic and three UCI real-world data sets demonstrate the efficiency of PSOGC.**

I. Introduction

CLUSTERING is one of the main tasks of knowledge discovery to identify inherent groups within a certain set of data. Clustering algorithms are mainly designed to group samples such that the similarity of samples in the same cluster are maximized while in different clusters are minimized. Ideally, a clustering algorithm should be simple, efficient and capable of dealing different cluster shapes [1].

K-means [2] is one of the most popular partitional clustering algorithms used in the community. It is easy to implement and efficient in obtaining a single reasonable partition of various data. Yet, K-means still suffers from drawbacks like the demand of a predefined number of clusters, sensitivity to initialization, no defense against irrelevant features, and incapability of handling non-spherical or non-linearly separable data distribution. Many methods have been proposed to solve the drawbacks of K-means. For example, Cheung [3] introduced a rival-penalised mechanism to K-means for penalizing incorrect clustering and deciding the number of clusters automatically. Population-based stochastic optimization techniques of powerful global search capability, such as particle swarm optimization (PSO) [4], [5], have been combined to K-means to alleviate the sensitivity of the clustering to the initialization. Feature weighting methods were also introduced to K-means to distinguish relevant and irrelevant features by assigning larger weights to relevant features [6]. To enable K-means to deal with non-linearly separable data, kernel K-means [7] was proposed to map data points from the original input space to a higher dimensional and more separable feature space through a nonlinear transformation.

In this study, we focus on solving the issues of sensitivity to initialization and non-spherical/non-linearly separable distribution, and accordingly propose a growing partitional clustering method based on PSO namely PSOGC. Particularly, PSO is applied to optimize the locations of cluster centers and a growing strategy based on k-nearest neighbor graph is proposed to generate clusters from the centers encoded in the PSO particles. The fitness of each particle is evaluated with a modularity-like measure in terms of both intra-cluster connectivity and inter-cluster disconnectivity of the resultant clustering. The method is designed to take advantages of the global search capability of PSO and non-linearly clustering capability of the growing strategy. PSOGC is evaluated on six synthetic data sets and three UCI data sets. Experimental results show that it is able to detect various cluster shapes efficiently and robustly.

The remainder of this paper is structured as follows. Section II describes the details of PSO clustering and the proposed PSOGC, Section III presents the experimental results of the algorithm on both synthetic and real-world data sets, and Section IV concludes this study.

II. Methods

Before describing the proposed PSOGC, the basic ideas of PSO and PSO-based clustering are briefly introduced as follows.

A. PSO and PSO-based Clustering

Inspired by the swarm behavior of bird flocking and fish schooling, PSO was first proposed by Kennedy and Eberhart [8] in 1995 as a population-base stochastic algorithm to handle complex optimization problems by exploiting the simulations of social interaction instead of the purely individual cognition.

This work was supported in part by the National Natural Science Foundation of China Joint Fund with Guangdong, under Key Project U1201256, the National Natural Science Foundation of China, under Grants 61171125 and 61205092, the Guangdong Foundation of Outstanding Young Teachers in Higher Education Institutions, under grant Yq2013141, Guangdong Natural Science Foundation under grant S2012010009545, the Scientific Research Foundation for the Returned Overseas Chinese Scholars, Ministry of Education of China, under grand 20111568, and Shenzhen Scientific Research and Development Funding Program under grants ZYC201105170243A, KQC201108300045A and JCYJ20130329115450637. All correspondence should be addressed to Prof. Zhen Ji (Email: jizhen@szu.edu.cn, Tel: +86 755 2655 7413).

In a conventional PSO, a swarm of individuals, namely particles, cooperate and evolve on the social behavior to search for the global optimum in the D-dimensional search space. Each particle records three pieces of information including the current position $x_i = (x_{i,1}, x_{i,2}, \cdots, x_{i,D})$, the velocity $v_i = (v_{i,1}, v_{i,2}, \cdots, v_{i,D})$, and the previous best position $p_i = (p_{i,1}, p_{i,2}, \cdots, p_{i,D})$. During the evolution, each particle is updated with the core formulae as follows:

$$v_{i,d}^{t+1} = \omega \times v_{i,d}^{t} + c_1 \times r_1 \times (p_{i,d}^{t} - x_{i,d}^{t}) + c_2 \times r_2 \times (p_{g,d}^{t} - x_{i,d}^{t}) \quad (1)$$

$$x_{i,d}^{t+1} = x_{i,d}^{t} + v_{i,d}^{t+1} \quad (2)$$

where the superscript t indicates the current number of iteration; $v_{i,d}^{t}$ and $x_{i,d}^{t}$ are the d-th dimensions of v_i and x_i, respectively; $p_{i,d}^{t}$ is the d-th dimension of the personal best ($pbest$) of particle i; $p_{d,g}^{t}$ is the d-th dimension of the global best ($gbest$); ω is an inertia weight introduced by Shi et al. [9] to balance the exploration and exploitation of PSO; r_1 and r_2 are uniformly distributed random numbers in [0,1]; c_1 and c_2 are acceleration coefficients to determine the balance between the influence of $pbest$ and $gbest$. PSO has attracted increasing interest of scientists from various research areas including clustering due to the simplicity of implementation, few parameters, and efficiency [10], [11].

To use PSO for clustering, the key issues are particle encoding and fitness evaluation. Let $S = \{s_1, \cdots, s_l, \cdots, s_n\}$ be a set of n objects to be clustered, m denote the dimension of each object, i.e., the length of each data vector, N_c be the number of clusters to be formed, and $C_{i,j}$ denote the j-th cluster in i-th particle. Each single particle is designed to represent the candidate N_c cluster centroids, i.e., each particle x_i is constructed as $x_i = (z_{i,1}, \cdots, z_{i,j}, \cdots, z_{i,N_c})$, where $z_{i,j}$ refers to the cluster centroid vector of $C_{i,j}$. The fitness of a particle x_i can be easily measured with the mean square error:

$$MSE = \frac{\sum_{j=1}^{N_c} [\sum_{\forall s_l \in C_{i,j}} d(s_l, z_{i,j})/|C_{i,j}|]}{N_c} \quad (3)$$

where $d(\cdot,\cdot)$ is a distance metric of two vectors and $|C_{i,j}|$ is the number of objects belonging to cluster $C_{i,j}$, i.e. the frequency of that cluster.

B. The Growing Clustering Based on PSO

Traditional K-means and PSO clustering work under the assumption of mixture of "spherical" Gaussian distribution model to ensure linear separableness. Pointing at nonlinearly separable data sets, a few solutions have been proposed, e.g., the kernel-based method [12], multi-exemplar clustering [13], and graph-based method [14], [15], [20]. These methods could somehow solve the problem, but most of them need prior knowledge specific for the target data sets. For example, one should learn the knowledge of a data set to choose a suitable kernel function in kernel-based clustering method. In multi-exemplar clustering, domain knowledge would be needed to tune the parameters crucial to build the multi-exemplar-model. Spectral clustering, based on graph theory, may have some challenges in calculating the eigenvalues and eigenvectors when the number of samples is large.

Inspired by the idea of graph based-method but avoiding complicated eigenvector calculation, we propose the PSOGC algorithm which is a combination of PSO-based clustering and a novel cluster growing strategy. The details of PSOGC are provided as follows.

As the intuitive goal of clustering is to divide the data samples into groups such that samples in the same group are similar and samples in different groups are dissimilar to each other, similarity graph $G = (V, E)$ is a suitable form to represent the relevance between samples, especially in nonlinear separable cases. Each vertex in this graph represents a data point s_i. Two vertices are connected if the similarity s_{ij} between the corresponding data points s_i and s_j is positive or greater than a certain threshold, and the edge is weighted by s_{ij}.

There are several popular constructions to transform a given set $\{s_1, \cdots, s_n\}$ of data points with pairwise similarities s_{ij} or pairwise distances d_{ij} into a graph [15]. In this study, the k-nearest neighbor graph is used. Firstly, an affinity matrix $A \in \mathbb{R}^{n \times n}$ is defined with element $a_{ij} = d(s_i, s_j)$ reflecting the similarity between s_i and s_j, where $d(\cdot,\cdot)$ is the widely used Euclidean distance. Then the k smallest elements in each column are set to 1 (forming a directed edge from s_i to s_j), leaving others to 0. In this way, the affinity matrix degenerates to a binary k-nearest neighbor graph-based sparse matrix B.

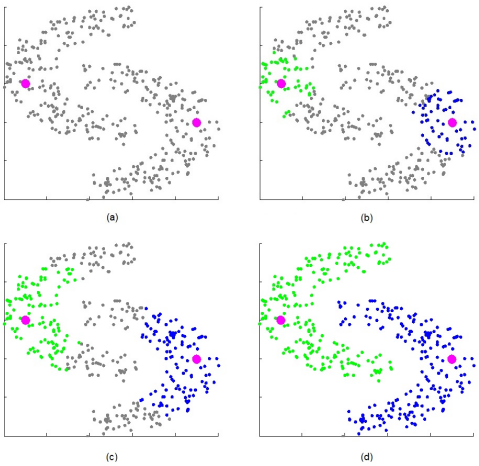


Fig. 1. Process of growing clustering.

Based on the matrix B and a set of candidate cluster centers, a growing method kicks in to group the samples into corresponding clusters. As shown in Fig. 1, at the beginning of the growing method, the N_c samples closest to the candidate cluster centers (one for each center) are selected as seedlings to grow their corresponding clusters. Then the k data samples nearest to each seedling are categorized to the corresponding cluster. In this step, at most $(k+1)N_c$ samples are affected. Afterward, the method searches in the sparse matrix B for new

samples with direct connection to the samples that have been clustered, and then each identified new sample is grouped to the corresponding cluster that has the most edges connected to it. Equivalently, for a non-grouped sample s_i, we calculate its connections to a cluster C_j using (4):

$$L(s_i, C_j) = \sum\nolimits_{c \in C_j} B_{s_i,c}, j = 1 \dots N_c \tag{4}$$

Find out the cluster C_{jmax} such that $jmax = \arg\max L(s_i, C_j)$ and let s_i join C_{jmax}. The procedure repeats until all samples are grouped. The cluster growing procedure is summarized in Algorithm 1.

Algorithm 1 Procedure of growing method

1: **Input**: a candidate cluster center vector $x_i = (z_{i,1}, z_{i,2}, \cdots, z_{i,N_c})$ and the matrix B
2: Choose N_c samples closest to the candidate cluster centers (one for each center) as seedlings.
3: **for** $i = 1$ to n **do**
4: **if** s_i is not grouped **then**
5: Find $jmax = \arg\max L(s_i, C_j)$ based on (4);
6: Group s_i to C_{jmax};
7: **end if**
8: **end for**

The growing method can generate clusters from candidate cluster centers, yet the cluster centers should be optimized by PSO. We follow the PSO-based clustering introduced in the previous section, except that a new fitness function is defined to evaluate the clustering generated from the candidate centers encoded in each particle. Particlarly, the fitness function is defined as follows:

$$fitness = \frac{\sum_i L(C_i, \overline{C_i})}{n^2} - \frac{\sum_i L(C_i, C_i)}{vol(B)} \tag{5}$$

where $\overline{C_i} = C - C_i$ is the complement of C_i,$vol(B) = \sum_{i,j} B_{i,j}$, $\frac{\sum_i L(C_i, \overline{C_i})}{n^2}$ represents the connection between different clusters and $-\frac{\sum_i L(C_i, C_i)}{vol(B)}$ represents the disconnection between samples within same clusters. Lower fitness indicates higher quality of the particle. The proposed PSOGC algorithm is summarized in Algorithm 2:

Algorithm 2 Procedure of PSOGC

1: **Input**: a data set S, the number of cluster N_c, and k;
2: Randomly initialize the position and velocity of each particle;
3: Generate the matrix B;
4: **while** stopping criteria are not satisfied **do**
5: Update the velocity and position of each particle according to (1) and (2);
6: Perform the growing clustering in **Algorithm 1** for each particle vector;
7: Evaluate the particles based on (5);
8: Update $pbest$ and $gbest$;
9: **end while**

III. Experiment Results and Discuss

A. Data sets

To test the performance of PSOGC, six synthetic data (SD) sets and three real-world UCI data sets are used. The synthetic data sets are plotted in Fig. 2. Among them SD1, SD3, SD4 and SD6 are nonlinearly separable. SD2 is linearly separable, but the non-spherical shape distribution makes it annoying to clustering algorithms. SD5 has four clusters within connection between them. As three of the most widely used UCI data sets, Iris, Wine, and Seeds are considered as the real-world benchmark to test the algorithms. The properties of the all data sets are summarized in Table I.

TABLE I
DATA SETS FOR CLUSTERING

Dataset	Number of samples	Number of attributes	Number of clusters
Iris	150	4	3
Wine	178	13	3
Seeds	210	7	3
SD1	400	2	2
SD2	650	2	4
SD3	266	2	3
SD4	400	2	2
SD5	1000	2	4
SD6	400	3	2

B. Evaluation Measure

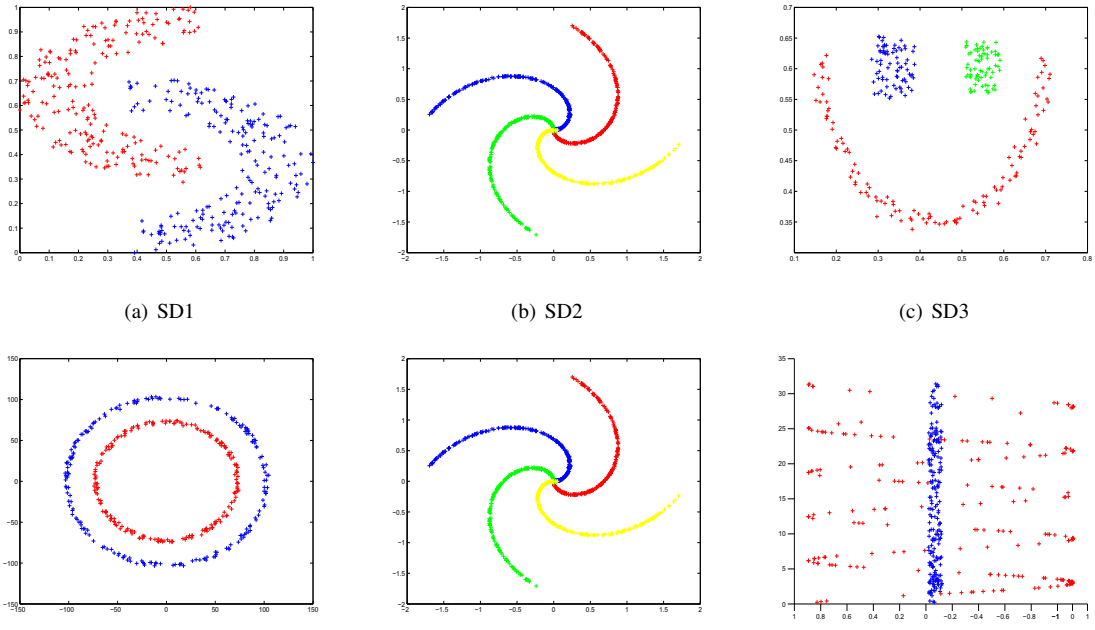
The clustering result can be evaluated in terms of inter- and/or intra- cluster distance, which is also known as internal evaluation. Internal evaluation criteria are indispensable when there is no information of the real clustering. However, when the class labels of the data sets are known, external criteria could provide more accurate evaluation of the clustering result. In the study, one of the most widely used external evaluation measure, Rand Index [16], is used to evaluate the performance of the clustering algorithms. Let $C = \{C_1, C_2, \cdots, C_j\}$ be the set of actual clusters in the data set and $C' = \{C_1', C_2', \cdots, C_j'\}$ be the set of j clusters generated by a clustering algorithm. Given a pair of points (S_i, S_j) in the data set, we refer to it as

1) SS if both points appear in the same cluster in C and also in the same cluster in C',
2) DD if the two points appear in two different clusters in C and also in different clusters in C'.

Check all point pairs in a data set, and let $\sum SS$ and $\sum DD$ record the number of the appearances of SS and DD, respectively. The Rand Index is calculated as

$$RI = \frac{2(\sum SS + \sum DD)}{n(n-1)}$$

RI takes value in $[0, 1]$ and a higher value indicates a more accurate clustering.



(a) SD1 (b) SD2 (c) SD3

(d) SD4 (e) SD5 (f) SD6

Fig. 2. Synthetic Data

C. Parameter Setting

A series of experimentations were performed to evaluate the parameter setting of PSOGC and also considering the suggestions presented in the literature, a set of parameters is found to deliver appropriate result. The parameters of PSO are set as follows: swarm size = 15, ω = 0.7, $c_1 = c_2 = 2$, and the maximum number of iterations = 50. The number of neighbors k is a key parameter in the growing method for deciding the speed of budding. A large k leads to a fast speed of budding, but it also tends to give a worse clustering result. In this study, we set k in a range $(\frac{n}{10N_c}, \frac{n}{5N_c})$.

D. Experiment Results

All experiments were performed on a PC configured with Intel Dual Core 2×2.5GHz CPU and 4GB RAM running MATLAB platform. We compare PSOGC to K-means, fuzzy C-means (FCM) [17], kernel K-means (KKM) [18], PSO Clustering (PSOC) [19] and spectral multi-manifold clustering (SMMC) [20]. K-means, FCM and KKM are run with default parameter setting until convergence is reached. PSOC takes the same PSO parameter setting as PSOGC, so that they consume similar computational cost. SMMC use the same parameter k as PSOGC to build up the linkage matrix between sample pairs, and default settings are applied to the other parameters. All algorithms are independently run 20 times on each data set and the average results are reported.

1) Synthetic data sets: Table II shows the performance of all algorithms on synthetic data. It is not surprising that K-means, FCM and PSOC cannot separate these data sets well. On SD4 and SD6, these three methods can only give a random clustering result. Compared to these three methods, KKM does better for it maps the data to a high-dimensional space in which they are linearly separable. However, different ways of mapping are needed for different data sets, so a specific kernel function will not give a good result all the time. SMMC, especially designed for manifolds clustering, shows up as the most distinguished one on synthetic data by transforming the feature space and combining the advantage of graph-based method as well as k-means. PSOGC gives a satisfactory result on all synthetic data sets. It is comparable to SMMC and superior to other algorithms.

2) UCI data sets: Fig. 3 shows the performance of the algorithms on the three real-world UCI data sets. PSOGC obtains slightly better average RI values than other algorithms over the three data sets. It also shows good robustness for having small variance. Unlike the case on synthetic data, SMMC does not perform so well as PSOGC on UCI data sets. It tends to obtain lower average RI and larger variance. The reason could be the lack of prior knowledge for the transformation of feature space, which may lead to initialization sensitivity for the following K-means step. On both synthetic and real-world data, PSOGC shows good robustness and efficiency in identifying clusters in various shape distributions. The time cost of PSOGC could be greater than K-means, FCM and KKM, but less than SMMC.

IV. Conclusion and Future Work

In this paper, we presented a growing clustering method based on PSO for clustering non-spherical or non-linearly separable data. The proposed method PSOGC is easy to use and performs robustly on both synthetic and real-world data. The simplicity and robustness of the algorithm make it a

TABLE II
COMPARISON OF THE CLUSTERING ALGORITHMS ON SYNTHETIC DATA SETS IN TERMS OF RAND INDEX.

Data sets	K-means	FCM	Kernel k-means	PSO clustering	SMMC	PSOGC
SD1	0.7622 ± 0.0036	0.7695 ± 0.0000	0.8387 ± 0.0045	0.7766 ± 0.0048	$\mathbf{0.9700 \pm 0.0156}$	0.9675 ± 0.0405
SD2	0.8299 ± 0.0344	0.8799 ± 0.0009	0.8509 ± 0.0149	0.7925 ± 0.0629	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
SD3	0.7579 ± 0.0300	0.7711 ± 0.0000	$\mathbf{1.0000 \pm 0.0000}$	0.7588 ± 0.0430	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
SD4	0.5012 ± 0.0004	0.5000 ± 0.0000	$\mathbf{1.0000 \pm 0.0000}$	0.5004 ± 0.0000	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
SD5	0.6479 ± 0.0111	0.6636 ± 0.0042	0.5057 ± 0.0184	0.6521 ± 0.0276	$\mathbf{0.9752 \pm 0.0012}$	0.9494 ± 0.0097
SD6	0.5057 ± 0.0206	0.5496 ± 0.0000	0.5137 ± 0.0146	0.5070 ± 0.0122	0.8667 ± 0.0323	$\mathbf{0.8983 \pm 0.0678}$

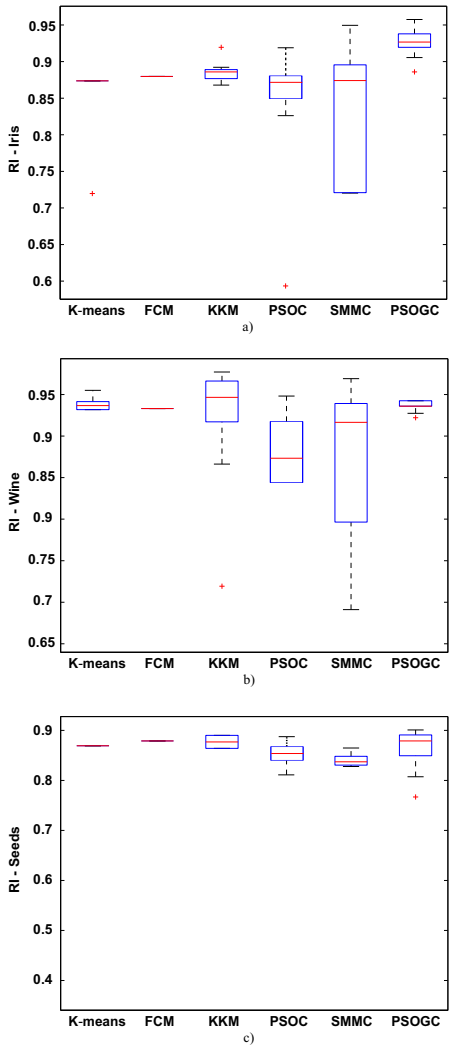


Fig. 3. Comparison of the clustering algorithms on UCI data sets, a) Iirs, b) Wine, c) Seeds

candidate solution for clustering complex data, especially in non real-time cases. Yet, we also note that it still has some drawbacks. For example, in the growing procedure, a smaller cluster may invade into the sphere of larger ones. In the future work, we will consider adding processes to manipulate asymmetric data sets avoiding the invasion of the smaller cluster.

REFERENCES

[1] S. Paterlini, T. Krink, "Differential evolution and particle swarm optimisation in partitional clustering." *Computational Statistics & Data Analysis*, vol. 50, no.5 pp. 1220-1247, 2006.

[2] A. K. Jain, "Data clustering: 50 years beyong K-means", *Pattern Recognition Letters*, vol. 31, pp. 651-666, 2010.

[3] Y. M. Cheung, "k*-Means: A new generalized k-means clustering algorithm," *Pattern Recognition Letters*, vol. 24, no. 15, pp. 2883-2893, 2003.

[4] A. Ahmadyfard and H. Modares, "Combining PSO and k-means to enheance data clustering," in *Proc. of 2008 International Symposium on Telecommunications* pp. 688-691, 2008.

[5] F. Yang, T. Sun, and C. Zhang, "An efficient hybrid data clustering method based on K-harmonic means, and Particle Swarm Optimization," *Expert Systems with Applications*, vol. 36, no. 6, pp. 9847-9852, 2009.

[6] J. Z. Huang, M. K. Ng, H. Rong and Z. Li, "Automated variable weighting in k-means type clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 657-668, 2005.

[7] B. Scholkopf, A. Smola, and K. CR. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299-1319, 1998.

[8] J. Kennedy and R. Eberhart "Particle swarm optimization", in *Proc. of 1995 IEEE International Conference on Neural Networks*, vol. 4, no. 2, pp. 1942-1948, 1995.

[9] Y. Shi and R. C. Eberhart, A modified particle swarm optimizer, in *Proc. of 1998 IEEE International Congress on Evolutionary Computation*, pp. 69-73, 1998.

[10] Z. Xinchao, "A perturbed particle swarm algorithm for numerical optimization." *Applied Soft Computing* vol. 10, no.1 pp. 119-124, 2010.

[11] D. W. Van, A. P. Engelbrecht, "Data clustering using particle swarm optimization." in *Proc. of 2003 IEEE Conference of Evolutionary Computation* pp. 215-220, 2003.

[12] B. Scholkopf, A. Smola, K.R. Mller,"Nonlinear component analysis as a kernel eigenvalue problem." *Neural computation* vol. 10, no. 5, pp. 1299-1319, 1998.

[13] M. Liu, X. Jiang and A. C. Kot, "A multi-prototype clustering algorithm," *Pattern Recognition* vol.42, no. 5, p. 689-698, 2009.

[14] S. E. Schaeffer, "Graph clustering," *Computer Science Review* vol. 1, no.1, pp. 27-64, 2007.

[15] U. V. Luxburg, "A tutorial on spectral clustering," *Statistics and computing* vol. 17, no. 4, pp. 395-416, 2007.

[16] W. M. Rand, "Objective criteria for the evaluation of clustering methods." *Journal of the American Statistical association* vol. 66, no. 336, pp. 846-850, 1971.
[17] R. J. Hathaway, J. C. Bezdek, "Fuzzy c-means clustering of incomplete data." *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 31, no. 5, pp 735-744, 2001.
[18] J Shawe-Taylor, N. Cristianini, *Kernel methods for pattern analysis*, Cambridge university press, 2004.
[19] S.C.M. Cohen, and LN. de Castro, "Data clustering with particle swarms," in *Proc. of 2006 IEEE Conference of Evolutionary Computation*, pp. 1792-1798, 2006.
[20] Y. Wang, et al, "Spectral clustering on multiple manifolds," *Neural Networks* vol. 22, no. 7, pp. 1149-1161, 2011.