Data Clustering by Particle Swarm Optimization with the Focal Particles

Tarık Küçükdeniz(✉) and Şakir Esnaf

Department of Industrial Engineering, Engineering Faculty,
Istanbul University, Istanbul, Turkey
{tkdeniz,sesnaf}@istanbul.edu.tr

Abstract. Clustering is an important technique in data mining. In unsupervised clustering, data is divided into several subsets (clusters) without any prior knowledge. Heuristic optimization based clustering algorithms tries to minimize an objective function, generally a clustering validity index, in the search space defined by the dimensions of the data vectors. If the number of the attributes of the data is large, then this will decrease the clustering performance. This study presents a new clustering algorithm, particle swarm optimization with the focal particles (PSOFP). Contrary to the standard particle swarm optimization (PSO) approach, this new clustering technique ensures high quality clustering results without increasing the dimensions of the search space. This new clustering technique handles communication among the particles in a swarm by using multiple focal particles. The number of focal particles equals to the number of clusters. This approach simplifies the candidate solution representation by a particle and therefore reduces the effect of 'curse of dimensionality'. Performance of the proposed method on the clustering analysis is benchmarked against K-means, K-means++, hybrid PSO and the CLARANS algorithms on five datasets. Experimental results show that the proposed algorithm has an acceptable efficiency and robustness and superior to the benchmark algorithms.

Keywords: Data clustering · Clustering analysis · High dimensional data · Particle swarm optimization · Focal particles

1 Introduction

Advances in technology has made information easy to capture and inexpensive to store, thus the amount of data stored in various databases increased dramatically. These data contain useful but hidden information that may be critical for the decision-making processes of the enterprises. Data mining is the general name of the techniques that are used to extract information from a very large amount of data [11]. Clustering is a major technique in data mining, which refers to a process of dividing data into several subsets while maintaining maximum similarity among the data within the same cluster and keeping minimum similarity among different clusters. Its applications can be seen in customer segmentation,

P. Pardalos et al. (Eds.): MOD 2015, LNCS 9432, pp. 280–292, 2015.
DOI: 10.1007/978-3-319-27926-8_25

document clustering and information retrieval, web data analysis, image segmentation, anomaly detection, biology, medicine and many other areas. Clustering is an unsupervised process, thus true knowledge about the class that each data object belongs to is not known by the clustering algorithm. If the true class label of data is known to the algorithm and used in the analysis then the method is named classification.

When we look at the history of clustering techniques, we see that many unsupervised clustering algorithms have been developed. K-means is one of the well-known of them. K-means clustering algorithm is easy to implement and very efficient, however suffers from several drawbacks. The objective function of the K-means is not convex hence it may contain many local minima. The outcome of the K-means algorithm is heavily dependent on the initial choice of the centroids [2]. In order to achieve better clustering performance, fuzzy c-means (FCM) clustering algorithm is introduced by Bezdek [4].

Clustering is also an application field in mathematical optimization when it is done by searching for the global minima of a clustering performance function. This approach makes it possible to apply heuristic algorithms to clustering analysis. Particle swarm optimization (PSO) is a population based heuristic algorithm, which maintains a population of particles where each particle represents a potential (candidate) solution to an optimization problem. Merwe and Engelbrecht used PSO in data clustering [22]. They also developed an hybrid approach, which combines PSO and K-means algorithm to achieve better clustering performance.

Merwe and Engelbrecht's original PSO data clustering approach inspired many works. Ji *et al.* clustered mobile networks by applying PSO to weighted clustering algorithm [12]. Correea *et al.* categorized sample types of biological databases with PSO [7]. Chen *et al.* tested PSO clustering algorithm on four different datasets. They analyzed the performance of standard PSO clustering algorithm in their paper [6]. Cui *et al.* applied PSO to the document clustering problem [8]. Attributes of documents defined as the dimensions of the particles. Omran *et al.* applied PSO to the image classification problem [18,19]. Their algorithm is a binary PSO model which dynamically adjusts the number of clusters. Kumar and Arasu proposed a particle swarm optimization based clustering method to medical databases [14]. Their modified particle swarm optimization based adaptive fuzzy K-modes algorithm produces good results in terms of precision and accuracy. Rana *et al.* gives a detailed literature review of PSO applications to data clustering [20]. Readers can also refer to [16] for further literature survey on nature inspired metaheuristic algorithms for data clustering.

Although each of these studies provide a number of improvements and innovations for clustering applications of PSO, all of them remains faithful to the Merwe and Engelbrecht's standard particle representation. But this representation creates a disadvantage by increasing the dimensions of the particles by the number of features of a data vector times the number of desired clusters (Fig. 1). Most stochastic optimization algorithms, including particle swarm optimization, suffer from this 'curse of dimensionality', which simply put, implies that their

performance deteriorates as the dimensionality of the search space increases [23]. Bouveyron *et al.* advises dimension reduction or subspace clustering as the primary ways of avoiding the curse of dimensionality [5].

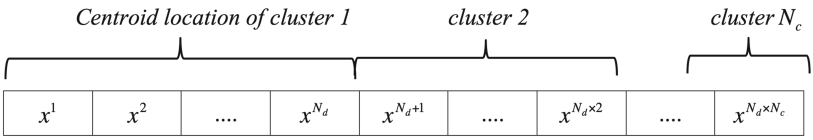

Fig. 1. Particle structure of the standard PSO. Each particle contains the centroids for all clusters.

The proposed method in this study,unlike the standard PSO approach, achieves high quality clustering results without increasing the number of dimensions. To do so, instead of a whole representation of a candidate solution by a particle (including all centroids of all clusters as in Fig. 1), in the proposed method, each particle represents only one centroid in the search space. Therefore, the number of dimensions of a particle equals the number of data vector features. Despite this major change in the particle representation, the proposed version of PSO's adherence to the standard PSO principles is provided by the changes made in the structure of the communication between particles.

One of the main configurational properties of PSO is topology or structure of connections between particles. Several approaches are developed to obtain good performance. In the *gbest* model, each particle is connected to all other particles (Fig. 2a). In the *lbest* model, each particle is connected to a predefined number of other particles (Fig. 2b). In star topology, which is a *lbest* model, one of the particles in the swarm become the focal particle and all other particles are connected to this focal particle (Fig. 2c). Therefore, all communication in the swarm is transmitted through this focal particle.

The proposed PSO variant in this study, addresses a star topology based new PSO clustering method. In this method there are several focal particles in the swarm. Other particles are connected to their nearest focal particle and all communication passes through these focal particles. There are several studies about focal particles in PSO [13,21]. However, we couldn't find any study on multiple focal particles in a swarm with dynamically changing neighborhoods among particles.

In this study, we aim to prove that, by decreasing the number of dimensions with the help of this multiple focal particle topology, our proposed PSO variant achieves high quality clustering results with less computation cost than other heuristics in high dimensional datasets. In the following sections, first data clustering is defined as an optimization problem. Then, in the third section, particle swarm optimization technique is introduced and the method of data clustering with particle swarm optimization is explained. In the fourth section particle swarm optimization with the focal particles method is introduced. This method

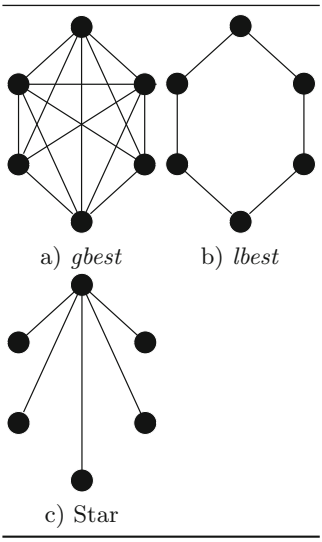

Fig. 2. Swarm topologies: *gbest* topology - Each particle is connected to each other. *lbest* topology - Each particle is connected to a number of other particles. Star topology - Each particle is connected to a focal particle.

is applied on five datasets and, results and the conclusion is given at the end of this study.

2 Data Clustering

When the data clustering problem is treated as an optimization problem, the aim is to find optimal centroids of clusters rather than finding optimal partition of the data vectors [1]. The dataset to be clustered is represented as a set of vectors $D = \{x_1, x_2, ..., x_m\}$ where m is the number of data objects x_i. A data object can have any number of dimensions. These dimensions of data is called attributes or features. A cost function is to be defined for clustering optimization problem. In clustering analysis these cost functions are validity indexes. A comprehensive review of the clustering methods can be found in [15,16,24].

2.1 Validity Indexes

Several validity indexes are defined to assess the performance of the clustering algorithms. In optimization based data clustering, these validity indexes (or similarity indexes) are used to calculate the fitness of the current solution. The most basic validity index is the sum of distances between the data vectors and their assigned cluster centroids in the vector space. This index is called clustering error index [1] and given in the Eq. (1).

$$J_e = \sum_{j=1}^{N_c} [\sum_{\forall x_i \in C_j} d(x_i, o_j)] \tag{1}$$

where d is the distance of the data vector x_i to its assigned centroid. N_c denotes the number of clusters (provided by the user). o_j denotes the centroid vector of cluster C_j.

Another validity index is quantization error (2) from [22].

$$J_q = \frac{\sum_{j=1}^{N_c} [\frac{\sum_{\forall x_i \in C_j} d(x_i, o_j)}{|C_j|}]}{N_c} \tag{2}$$

Here $\mid C_j \mid$ is the number of data vectors belonging to cluster C_j. This quantization error is the average distances of the data vectors to their assigned centroids. The quantization error used in the Eq. (2) allows for division by zero. In our study, if a division by zero was encountered, the fitness of the particle was approximated to infinity.

One another well-known validity index is Silhouette value. The silhouette value for each point is a measure of how similar that point is to points in its own cluster, when compared to points in other clusters. Higher silhouette means a better assignment of data vectors to clusters. Formula for silhouette value is given in (3).

$$S(x_i) = \frac{b(x_i) - a(x_i)}{max\{a(x_i), b(x_,)\}} \tag{3}$$

where $a(x_i)$ is the average distance from the ith point to the other points in the same cluster as i, and $b(x_i)$ is the minimum average distance from the ith point to points in a different cluster, minimized over clusters. Silhouette value is in between -1 and $+1$. There are several other validity indexes for data clustering, a brief list of them can be seen in [16].

The distance parameter in the Eqs. (1) and (2) can be Euclidian, cosine or any other distance metric. In data clustering euclidian distance, given in the Eq. (4), is one of the most frequently used metric. But at some special occasions like document clustering, cosine distance is more suitable [25].

$$d(x_i, o_j) = \sqrt{\sum_{k=1}^{N_d} (x_{ik} - o_{jk})^2} \tag{4}$$

Here N_d is the data dimension, i.e. the number of attributes of each data vector.

3 Particle Swarm Optimization

Swarm optimization algorithms are inspired by the efforts to model the social systems of birds and bees. Particle swarm optimization is developed by Kennedy

and Eberhart in 1995 [9]. In PSO, each particle represents a position in N_d dimensional space. PSO algorithm moves particles through this multi-dimensional search space to search for an optimal solution. A particle's movement is affected by three factors; (1) Particle's own velocity vector, $\vec{v}_i$ - (2) Particle's best position found thus far, $\vec{p}_i$ - (3) Best position found by the particles in the neighborhood of that particle, $\vec{y}_i$.

In the first step of the algorithm, velocity of a particle is calculated as in (5) and then this value is added to the current position of the particle as given in (6). If $\vec{x}_i$ is the current position of the particle, $\vec{v}_i$ is the current velocity of the particle and $\vec{p}_i$ is the personal best position of the particle then the velocity of the particle for the next iteration is;

$$\begin{aligned}\vec{v}_{i,k}(t+1) = w\vec{v}_{i,k}(t) + c_1 r_{1,k}(t)(\vec{p}_{i,k}(t) - \vec{x}_{i,k}(t)) \\ + c_2 r_{2,k}(t)(\vec{y}_{i,k}(t) - \vec{x}_{i,k}(t))\end{aligned} \tag{5}$$

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t+1) \tag{6}$$

where w is the inertia weight, c_1, c_2 are positive constants, called the cognitive and social acceleration factors respectively. $r_{1,k}(t)$, $r_{2,k}(t) \backsim U(0,1)$, and $k = 1, ..., N_d$ [22].

3.1 Data Clustering with Particle Swarm Optimization

In PSO, every particle represents a candidate or potential solution. The model employed by the particle should point a solution of the problem by its own. In Merwe and Engelbrecht's [22] method, a particle is constructed as in 7.

$$\vec{x}_i = (o_{i1}, o_{i2}, ..., o_{ij}, ...o_{iN_c}) \tag{7}$$

where o_{ij} corresponds to the j_{th} centroids represented by the i_{th} particle. Thus, if a data vector consists of N_d dimensions, then a particle will have $N_d \times N_c$ dimensions.

PSO algorithm tries to minimize an objective function iteration by iteration. In data clustering mode, this objective function should be chosen carefully to achieve a good clustering result at the end of the iterations or when a termination criteria for the PSO is reached. Merwe and Engelbrecht [22] have chosen quantization error (2) as the fitness function.

4 Particle Swarm Optimization with the Focal Particles

As it is explained before, in PSO, a particle is a representation of a whole solution, thus a particle should have $N_d \times N_c$ dimensions. This usually yields the so-called 'curse of dimensionality' problem. To overcome this ineffectiveness, we have developed a new clustering approach, namely particle swarm optimization with the focal particles (PSOFP). In this new approach each particle represents only one centroid in the search space. If N_c is the number of clusters, then N_c

number of particles are chosen as the final representatives of clustering solution. These particles are the focal particles to which all other particles in the swarm are connected to their nearest. This neighborhood structure is similar to Fig. 3. This approach results in less dimensionality in particles. Therefore, it is expected to have less computational cost than the standard approaches. In the next section we have benchmarked PSOFP's performance with other clustering algorithms.

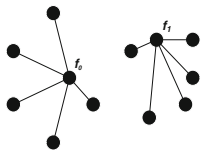

Fig. 3. f_0 and f_1 are the focal particles. There are 13 particles in total. This is an example for a two cluster problem.

In PSOFP, a particle is constructed as in (8).

$$\vec{x}_i = (o_i) \tag{8}$$

where $\vec{x}_i$ is a centroid in the search space. Algorithm 1 displays the pseudo code of PSOFP algorithm. To start PSOFP, a swarm with l particles are initialized with the particle formation given in (8). Swarm initialization of PSOFP is similar to the standard PSO. Then, randomly selected N_c number of these particles are labeled as focal particles. The swarm size should be bigger than N_c. At each iteration, the fitness value of each particle is calculated. To do this calculation, first centroid locations represented by the focal particles are combined together to make a candidate solution. Then, for each non-focal particle, the particle's position vector (the centroid it represents) is overwritten to the corresponding place in the candidate solution. This process is illustrated in the Table 1.

In this illustrative example, a swarm with 8 particles is initialized. We are trying to cluster our data vectors into three clusters. Thus, the first three particles are assigned as the focal particles. The data vectors are in two dimensions, therefore each particle has two dimensions. To calculate the fitness value of the fourth particle,

- First a candidate solution is built by the focal particles as: {10; 18; 45; 26; 21; 34}. The first two columns of the candidate solution is the centroid of the first cluster and the third and the fourth terms are the centroid of the second cluster, the last part is the centroid of the third cluster.
- Then, we calculate the nearest focal particle to the fourth particle using the selected distance metric. It is the second focal particle in this example.
- In the candidate solution, places belonging to the second focal particle is replaced with the current particle's position: {10; 18 ; 38; 30; 21; 34}. Fitness of the fourth particle is calculated by using this final candidate solution.

Table 1. An illustrative example for the PSOFP fitness calculation process.

Particle Nr.	Focal?	Position vector $(x_1; x_2)$
1	True	10; 18
2	True	45; 26
3	True	21; 34
4	False	38; 30
5	False	12; 22
6	False	5; 52
7	False	15; 42
8	False	45; 22

Another difference from the standard PSO is, focal particles in PSOFP will not have their own inertia weight component. Focal particles are only affected by their own personal best and the best performances of the other particles that are connected to these focal particles. At the end of each iteration, particles, including the focal, move in the search space. When these movements finishes, the neighborhood structure of the swarm is to be updated. Each particle, except focal ones, will be connected to its nearest focal particle. To do this, the distances among focal and non-focal particles are calculated again.

5 Application and Results

Table 2 shows the datasets used for benchmarking. IRIS, WINE, CMC and Gesture Data are from UCI benchmark datasets. RAND1 is a randomly generated dataset which includes $500 \times U(0, 100)$, $1000 \times U(500, 1500)$ and $1000 \times U(2500, 3000)$ values.

Table 2. Benchmark datasets

Name	Data vectors	Data attributes	Clusters
IRIS	150	4	3
WINE	178	13	3
CMC	1473	9	3
Gesture Data	4833	18	5
RAND1	2500	25	3

The following methods are used for benchmarking:

– Standard K-Means Clustering Algorithm

Require:
· Dataset: $D = \{x_1, x_2, ..., x_m\}$
· The number of clusters: N_c
Initialisation:
· Initialize the position $\vec{x}_i$ and velocity $\vec{v}_i$ of $l > N_c$ number of particles randomly. Each particle contains one randomly generated centroid vector (o_i) in the search space.
· Define the set of focal particles S_F, where the number of focal particles equal to N_c

foreach *iteration* **do**
  forall the *particle i* **do**
    · x_i: Position of the particle i
    · f_i: The index of the focal particle that particle i is connected to
    · x_{f_i}: Position of the focal particle that particle i is connected to
    · x_{S_F}: All focal particles' positions
    · generate a candidate solution by replacing the x_{f_i} in the x_{S_F} with the x_i
    · calculate the fitness of particle: $J(x_i)$by a clustering validity index

    `// Compare the particles current fitness with its` *pbest*`:`
    if $J(x_i) < J(p_i)$ **then**
      p_i=x_i
    end
  end
  forall the *particle i* **do**
    · *Define neighborhood*: If i is non-focal then assign i to its nearest focal particle
    · y_i=MIN($p_i \in S^i_{neigh}$) where S^i_{neigh} is the neighborhood of i
    · Change the velocity of the particle i according to the equation (5)
    if $v_i > v_{max}$ **then**
      $v_i = v_{max}$`// Check if the velocity is out of limits`
    end
    Calculate the position of i according to the equation (6)
    if $x_i > x_{max}$ **then**
      $x_i = x_{max}$`// Check if the position is out of limits`
    end
    if $x_i < x_{min}$ **then**
      $x_i = x_{min}$`// Check if the position is out of limits`
    end
  end
end

Algorithm 1. Pseudo code for PSOFP algorithm

– K-means++ Algorithm: Arthur and Vassilvitskii's K-means++ algorithm [3], is an improvement to the standard K-means for choosing better initial values and therefore avoiding poor results.

– Merwe's [22] hybrid PSO data clustering method: In hybrid PSO, the result of K-means clustering feed into PSO as a particle, i.e. the solution of K-means algorithm is where the PSO starts.
– CLARANS: Ng and Han [17] introduced the algorithm CLARANS (Clustering Large Applications based upon RANdomized Search) in the context of clustering in spatial databases. Authors considered a graph whose nodes are the sets of k medoids and an edge connects two nodes if they differ by exactly one medoid.

We have paralleled the benchmarking tests on a 16 processor computer. Due to the random nature of k-means and particle swarm algorithms, all methods have been run 160 times. The test computer had 16 Intel Xeon E5 2.90 Ghz processors with 30 GB of RAM. 8 parallel runs are done at the same time. We also tried paralleling the fitness evaluation process in a single run. But due to the high information preprocessing overhead, parallel evaluation of fitness functions in a single run was slower than the serial evaluation. Our test computer was on the Amazon EC2 cloud computing servers. We refer to [10] for a discussion on parallelization in data mining applications.

PSO and PSOFP algorithms are initialized with 100 particles. Permitted maximum iteration count is 4000, but iterations stop when there is less than 0.0001 improvement in the global best value during the last 250 iterations. Equation 5 is used for velocity calculations, $w = 0.90$, $c_1 = c_2 = 2.05$. In standard PSO, the *gbest* model is chosen. Selection of the fitness function is an important process in heuristic optimization. We choose quantization error (2) as the fitness function. Quantization error and Silhouette values of each method is reported in the Table 3. CPU time column is the mean CPU time for 160 runs. *Mean* and *Min.* columns of quantization error represent the average and the best value obtained from 160 runs. *Max.* column of Silhouette value represents the best value achieved among 160 runs for the Silhouette index. *S.D.* column gives the standard deviation of runs.

When we refer to the quantization error, proposed PSOFP algorithm outperforms all other algorithms on the benchmark datasets. The mean value of the quantization error of PSOFP on five datasets is 3.71 %, 4.16 %, 4.06 % and 1.65 % lower than the K-means, K-means++, PSO Hybrid and CLARANS algorithms respectively. When we compare the best valued achieved by each algorithm (minimum values), PSOFP is 7.64 %, 7.59 %, 6.42 % and 7.78 % better than these algorithms. Standard deviation is an indicator of the representation strength of reported average errors. In all datasets, except RAND1, standard deviation of PSOFP is lower than the benchmarking algorithms. This shows that proposed PSOFP is a robust clustering technique. Silhouette value is another useful index to analyze the clustering performance. Values nearer to +1 is better for the Silhouette index. Silhouette values of PSOFP is equal or slightly better than the benchmarking algorithms. Only, in RAND1 dataset CLARANS algorithm is 2.23 % better than the PSOFP on the average.

As the CPU time column of the Table 3 indicates, due to the less number of dimensions of the search space in the PSOFP method, PSOFP is much faster,

at the same time more successful in the term of clustering validity, than the standard PSO. Its computational time is 45.03 %, 5.00 %, 39.66 %, 64.54 % and 9.25 % less than the standard PSO algorithm in WINE, IRIS, CMC, Gesture and RAND1 datasets respectively. Although CLARANS algorithm gives better results than the PSOFP on RAND1 dataset, its computational time in this dataset is 4.3 times higher than the PSOFP.

Table 3. Benchmark results over 160 runs for each method.

Dataset	Algorithm	CPU time	Quantization error			Silhouette		
			Mean	Min	S.D.	Mean	Max	S.D.
WINE	K-Means	0.53	101.58	97.87	3.91	0.726	0.73	0.01
	K-Means++	0.45	99.84	97.87	3.43	0.729	0.73	0.02
	PSO Hybrid	668.75	100.67	97.87	3.73	0.728	0.73	0.01
	CLARANS	14,937.00	99.61	97.15	2.11	0.726	0.74	0.02
	PSOFP	367.61	96.72	95.51	1.59	0.726	0.75	0.14
IRIS	K-Means	0.44	0.65	0.64	0.02	0.724	0.74	0.06
	K-Means++	0.35	0.65	0.64	0.01	0.725	0.74	0.05
	PSO Hybrid	257.98	0.65	0.64	0.01	0.730	0.74	0.04
	CLARANS	356.86	0.65	0.64	0.00	0.730	0.74	0.02
	PSOFP	245.08	0.61	0.53	0.02	0.735	0.74	0.13
CMC	K-Means	11.38	3.83	3.83	0.00	0.645	0.65	0.01
	K-Means++	9.66	3.83	3.83	0.00	0.645	0.65	0.01
	PSO Hybrid	832.80	3.83	3.83	0.00	0.645	0.65	0.01
	CLARANS	654.77	3.83	3.83	0.00	0.645	0.65	0.01
	PSOFP	502.53	3.83	3.82	0.002	0.643	0.65	0.00
Gesture	K-Means	15.54	1.51	1.47	0.021	0.534	0.60	0.001
	K-Means++	12.56	1.50	1.47	0.023	0.523	0.60	0.001
	PSO Hybrid	1,470.71	1.59	1.37	0.046	0.532	0.70	0.003
	CLARANS	867.90	1.54	1.48	0.025	0.535	0.67	0.002
	PSOFP	521.51	1.46	1.19	0.02	0.536	0.71	0.00
RAND1	K-Means	25.11	369.23	334.71	134.11	0.952	0.98	0.11
	K-Means++	24.74	388.58	334.71	160.07	0.905	0.98	0.33
	PSO Hybrid	682.72	360.31	334.71	114.50	0.947	0.98	0.21
	CLARANS	2,664.28	334.80	334.71	0.00	0.978	0.98	0.00
	PSOFP	619.60	354.26	334.71	85.67	0.957	0.98	0.10

6 Conclusions

In this study a new approach is presented for clustering analysis using particle swarm optimization with the focal particles. In standard PSO, each particle

is a representation of the final solution, however, this increases the number of dimensions a particle has. In PSOFP, each particle is a representation of only one point in the search space, therefore the number of dimensions are lower than the standard PSO. We analyzed the performance effect of this dimensionality reduction to the clustering performance. We selected three small and two large datasets and benchmarked proposed PSOFP algorithm with the standard K-means, K-means++, hybrid PSO and CLARANS algorithms. Each algorithm has run 160 times. The Amazon EC2 cloud computing platform is used and 8 parallel runs has been made each time. Also, we tried paralleling the objective function evaluation of particle swarm optimization. This approach didn't accelerate the clustering analysis due to the high information overhead among parallel processes.

Quantization error and Silhouette values are chosen as the performance criteria for benchmark tests. The results indicated that while maintaining better or equal clustering performance with the benchmarking algorithms, PSOFP was faster than the standard PSO algorithm. This shows that the dimensionality reduction approach of the PSOFP is an efficient and robust strategy in heuristic-based data clustering analysis.

As the future work, an improved fully parallel approach for focal particles can be studied. We employed Euclidian distance as the distance metric in our calculations. But cosine metric is also known to be a good representative for the similarity among data objects in high dimensional space. The performances of the algorithms can be compared by using cosine distance metric.

References

1. Abdel-Kader, R.: Genetically improved PSO algorithm for efficient data clustering. In: Second International Conference on Machine Learning and Computing (2010)
2. Ahmadyfard, A., Modares, H.: Combining PSO and k-means to enhance data clustering. In: International Symposium on Telecommunications, IST 2008, pp. 688–691. IEEE (2008)
3. Arthur, D., Vassilvitskii, S.: k-means++: the advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1027–1035. Society for Industrial and Applied Mathematics (2007)
4. Bezdek, J.C.: Fuzzy Mathematics in Pattern Classification. Cornell University, Ithaca (1973)
5. Bouveyron, C., Girard, S., Schmid, C.: High-dimensional data clustering. Comput. Stat. Data Anal. **52**(1), 502–519 (2007)
6. Chen, C.-Y., Ye, F.: Particle swarm optimization algorithm and its application to clustering analysis. In: 2004 IEEE International Conference on Networking, Sensing and Control, vol. 2, pp. 789–794. IEEE (2004)
7. Correa, E.S., Freitas, A.A., Johnson, C.G.: A new discrete particle swarm algorithm applied to attribute selection in a bioinformatics data set. In: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, pp. 35–42. ACM (2006)
8. Cui, X., Potok, T.E., Palathingal, P.: Document clustering using particle swarm optimization. In: Proceedings of the 2005 IEEE Swarm Intelligence Symposium, SIS 2005, pp. 185–191. IEEE (2005)

9. Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, New York, NY, vol. 1, pp. 39–43 (1995)
10. García-Pedrajas, N., de Haro-García, A.: Scaling up data mining algorithms: review and taxonomy. Prog. Artif. Intell. **1**(1), 71–87 (2012)
11. Hatamlou, A., Abdullah, S., Nezamabadi-pour, H.: A combined approach for clustering based on k-means and gravitational search algorithms. Swarm Evol. Comput. **6**, 47–52 (2012)
12. Ji, C., Zhang, Y., Gao, S., Yuan, P., Li, Z.: Particle swarm optimization for mobile ad hoc networks clustering. In: IEEE International Conference on Networking, Sensing and Control, vol. 1, pp. 372–375. IEEE (2004)
13. Kennedy, J., Mendes, R.: Population structure and particle swarm performance (2002)
14. Kumar, R.S., Arasu, G.T.: Modified particle swarm optimization based adaptive fuzzy k-modes clustering for heterogeneous medical databases. J. Sci. Ind. Res. **74**, 19–28 (2015)
15. Maimon, O.Z., Rokach, L.: Data Mining and Knowledge Discovery Handbook, 1st edn. Springer, US (2005)
16. Nanda, S.J., Panda, G.: A survey on nature inspired metaheuristic algorithms for partitional clustering. Swarm Evol. Comput. **16**, 1–18 (2014)
17. Ng, R.T., Han, J.: Efficient and effective clustering methods for spatial data mining. In: Proceedings of VLDB, pp. 144–155 (1994)
18. Omran, M., Salman, A., Engelbrecht, A.P.: Image classification using particle swarm optimization. In: Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning, Singapore, vol. 1, pp. 18–22 (2002)
19. Omran, M.G., Salman, A., Engelbrecht, A.P.: Dynamic clustering using particle swarm optimization with application in image segmentation. Pattern Anal. Appl. **8**(4), 332–344 (2006)
20. Rana, S., Jasola, S., Kumar, R.: A review on particle swarm optimization algorithms and their applications to data clustering. Artif. Intell. Rev. **35**(3), 211–222 (2010)
21. Reyes-Sierra, M., Coello, C.A.C.: Multi-objective particle swarm optimizers: a survey of the state-of-the-art. Int. J. Comput. Intell. Res. **2**(3), 287–308 (2006)
22. Van der Merwe, D.W., Engelbrecht, A.P.: Data clustering using particle swarm optimization. In: The 2003 Congress on Evolutionary Computation, CEC'03, vol. 1, pp. 215–220. IEEE (2003)
23. van den Bergh, F., Engelbrecht, A.P.: A cooperative approach to particle swarm optimization. IEEE Trans. Evol. Comput. **8**(3), 225–239 (2004)
24. Xu, R., Wunsch, D.: Survey of clustering algorithms. IEEE Trans. Neural Netw. **16**(3), 645–678 (2005)
25. Zhao, Y., Karypis, G.: Comparison of agglomerative and partitional document clustering algorithms. Technical report, DTIC Document (2002)