Document Clustering Analysis Based on Hybrid PSO+K-means Algorithm

Xiaohui Cui, Thomas E. Potok
Applied Software Engineering Research Group, Computational Sciences and Engineering Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831-6085, USA
cuix, potokte@ornl.gov

Abstract: There is a tremendous proliferation in the amount of information available on the largest shared information source, the World Wide Web. Fast and high-quality document clustering algorithms play an important role in helping users to effectively navigate, summarize, and organize the information. Recent studies have shown that partitional clustering algorithms are more suitable for clustering large datasets. The K-means algorithm is the most commonly used partitional clustering algorithm because it can be easily implemented and is the most efficient one in terms of the execution time. The major problem with this algorithm is that it is sensitive to the selection of the initial partition and may converge to a local optima. In this paper, we present a hybrid Particle Swarm Optimization (PSO)+K-means document clustering algorithm that performs fast document clustering and can avoid being trapped in a local optimal solution as well. For comparison purpose, we applied the PSO+K-means, PSO, K-means, and other two hybrid clustering algorithms on four different text document datasets. The number of documents in the datasets range from 204 to over 800, and the number of terms range from over 5000 to over 7000. The results illustrate that the PSO+K-means algorithm can generate the most compact clustering results than other four algorithms.

Keywords: Particle Swarm Optimization, Text Dataset, Cluster Centroid, Vector Space Model

INTRODUCTION

Document clustering is a fundamental operation used in unsupervised document organization, automatic topic extraction, and information retrieval. Clustering involves dividing a set of objects into a specified number of clusters [21]. The motivation behind clustering a set of data is to find inherent structure in the data and expose this structure as a set of groups. The data objects within each group should exhibit a large degree of similarity while the similarity among different clusters should be minimized [3, 9, 18]. There are two major clustering techniques: "Partitioning" and "Hierarchical" [9]. Most document clustering algorithms can be classified into these two groups. Hierarchical techniques produce a nested sequence of partition, with a single, all-inclusive cluster at the top and single clusters of individual points at the bottom. The partitioning clustering method seeks to partition a collection of documents into a set of non-overlapping groups, so as to maximize the evaluation value of clustering. Although the hierarchical clustering technique is often portrayed as a better quality clustering approach, this technique does not contain any provision for the reallocation of entities, which may have been poorly classified in the early stages of the text analysis [9]. Moreover, the time complexity of this approach is quadratic [18].

In recent years, it has been recognized that the partitional clustering technique is well suited for clustering a large document dataset due to their relatively low computational requirements [18]. The time complexity of the partitioning technique is almost linear, which makes it widely used. The best-known partitioning clustering algorithm is the K-means algorithm and its variants [10]. This algorithm is simple, straightforward and is based on the firm foundation of analysis of variances. The K-means algorithm clusters a group of data vectors into a predefined number of clusters. It starts with a random initial cluster center and keeps reassigning the data objects in the dataset to cluster centers based on the similarity between the data object and the cluster center. The reassignment procedure will not stop until

a convergence criterion is met (e.g., the fixed iteration number or the cluster result does not change after a certain number of iterations).

The main drawback of the K-means algorithm is that the cluster result is sensitive to the selection of the initial cluster centroids and may converge to the local optima [16]. Therefore, the initial selection of the cluster centroids decides the main processing of K-means and the partition result of the dataset as well. The main processing of K-means is to search the local optimal solution in the vicinity of the initial solution and to refine the partition result. The same initial cluster centroids in a dataset will always generate the same cluster results. However, if good initial clustering centroids can be obtained using any of the other techniques, the K-means would work well in refining the clustering centroids to find the optimal clustering centers [2]. It is necessary to employee some other global optimal searching algorithm for generating this initial cluster centroids. The Particle Swarm Optimization (PSO) algorithm is a population based stochastic optimization technique that can be used to find an optimal, or near optimal, solution to a numerical and qualitative problem [4, 11, 17]. The PSO algorithm can be used to generate good initial cluster centroids for the K-means. In this paper, we present a hybrid PSO+K-means document clustering algorithm that performs fast document clustering and can avoid being trapped in a local optimal solution. The results from our experiments indicate that the PSO+K-means algorithm can generate the best results in just 50 iterations in comparison with the K-means algorithm and the PSO algorithm.

The remainder of this paper is organized as follows: Section 2 provides the methods of representing documents in clustering algorithms and of computing the similarity between documents. Section 3 provides a general overview of the PSO algorithm. The hybrid PSO+K-means clustering algorithms are described in Section 4. Section 5 provides the detailed experimental setup and results for comparing the performance of the PSO+K-means algorithm with the K-means, PSO, and other hybrid approaches. The discussion of the experiment's results is also presented. The conclusion is in Section 6.

PRELIMINARIES

Document representation: In most clustering algorithms, the dataset to be clustered is represented as a set of vectors $X=\{x_1, x_2, ..., x_n\}$, where the vector x_i corresponds to a single object and is called the feature vector. The feature vector should include proper features to represent the object. The text document objects can be represented using the Vector Space Model (VSM) [8]. In this model, the content of a document is formalized as a dot in the multi-dimensional space and represented by a vector *d*, such as $d=\{w_1, w_2, ...w_n\}$, where $w_i(i = 1,2,...,n)$ is the term weight of the term t_i in one document. The term weight value represents the significance of this term in a document. To calculate the term weight, the occurrence frequency of the term within a document and in the entire set of documents must be considered. The most widely used weighting scheme combines the Term Frequency with Inverse Document Frequency (TF-IDF) [8]. The weight of term *i* in document *j* is given in equation 1:

$$w_{ji} = tf_{ji} * idf_{ji} = tf_{ji} * \log_2(n / df_{ji}) \quad (1)$$

where tf_{ji} is the number of occurrences of term *i* in the document *j*; df_{ji} indicates the term frequency in the collections of documents; and *n* is the total number of documents in the collection. This weighting scheme discounts the frequent words with little discriminating power.

The similarity metric: The similarity between two documents needs to be measured in a clustering analysis. Over the years, two prominent ways have been proposed to compute the similarity between documents m_p and m_j. The first method is based on Minkowski distances [5], given by:

$$D_n(m_p, m_j) = \left(\sum_{i=1}^{d_m} | m_{i,p} - m_{i,j} |^n \right)^{1/n} \quad (2)$$

For n =2, we obtain the Euclidean distance. In order to manipulate equivalent threshold distances, considering that the distance ranges will vary according to the dimension number, this algorithm uses the normalized Euclidean distance as the similarity metric of two documents, m_p and m_j, in the vector space. Equation 3 represents the distance measurement formula:

$$d(m_p, m_j) = \sqrt{\sum_{k=1}^{d_m} (m_{pk} - m_{jk})^2 / d_m} \quad (3)$$

where m_p and m_j are two document vectors; d_m denotes the dimension number of the vector space; m_{pk} and m_{jk} stand for the documents m_p and m_j's weight values in dimension *k*.

The other commonly used similarity measure in document clustering is the cosine correlation measure [15], given by:

$$\cos(m_p, m_j) = \frac{m_p^t m_j}{|m_p||m_j|} \quad (4)$$

where $m_p^t m_j$ denotes the dot-product of the two document vectors; |.| indicates the length of the vector. Both similarity metrics are widely used in the text document clustering literatures.

BACKGROUND OF THE PSO ALGORITHM

PSO was originally developed by Eberhart and Kennedy in 1995 [11], and was inspired by the social behavior of a bird flock. In the PSO algorithm, the birds in a flock are symbolically represented as particles. These particles can be considered as simple agents "flying" through a problem space. A particle's location in the multi-dimensional problem space represents one solution for the problem. When a particle moves to a new location, a different problem solution is generated. This solution is evaluated by a fitness function that provides a quantitative value of the solution's utility.

The velocity and direction of each particle moving along each dimension of the problem space will be altered with each generation of movement. In combination, the particle's personal experience, P_{id} and its neighbors' experience, P_{gd} influence the movement of each particle through a problem space. The random values, $rand_1$ and $rand_2$, are used for the sake of completeness, that is, to make sure that particles explore wide search space before converging around the optimal solution. The values of c_1 and c_2 control the weight balance of P_{id} and P_{gd} in deciding the particle's next movement velocity. For every generation, the particle's new location is computed by adding the particle's current velocity, V-vector, to its location, X-vector. Mathematically, given a multi-dimensional problem space, the *ith* particle changes its velocity and location according to the following equations [11]:

$$v_{id} = w * v_{id} + c_1 * rand_1 * (p_{id} - x_{id}) + c_2 * rand_2 * (p_{gd} - x_{id}) \quad (5a)$$

$$x_{id} = x_{id} + v_{id} \quad (5b)$$

where w denotes the inertia weight factor; p_{id} is the location of the particle that experiences the best fitness value; p_{gd} is the location of the particles that experience a global best fitness value; c_1 and c_2 are constants and are known as acceleration coefficients; d denotes the dimension of the problem space; $rand_1$, $rand_2$ are random values in the range of (0, 1).

The inertia weight factor w provides the necessary diversity to the swarm by changing the momentum of particles to avoid the stagnation of particles at the local optima. The empirical research conducted by Eberhart and Shi [7] shows improvement of search efficiency through gradually decreasing the value of inertia weight factor from a high value during the search.

Equation 5a requires each particle to record its current coordinate X_{id}, its velocity V_{id} that indicates the speed of its movement along the dimensions in a problem space, and the coordinates P_{id} and P_{gd} where the best fitness values were computed. The best fitness values are updated at each generation, based on equation 6,

$$P_i(t+1) = \begin{cases} P_i(t) & f(X_i(t+1)) \le f(X_i(t)) \\ X_i(t+1) & f(X_i(t+1)) > f(X_i(t)) \end{cases} \quad (6)$$

where the symbol f denotes the fitness function; $P_i(t)$ stands for the best fitness values and the coordination where the value was calculated; and t denotes the generation step.

It is possible to view the clustering problem as an optimization problem that locates the optimal centroids of the clusters rather than finding an optimal partition. This view offers us a chance to apply PSO optimal algorithm on the clustering solution. In [6], we proposed a PSO document clustering algorithm. Contrary to the localized searching in the K-means algorithm, the PSO clustering algorithm performs a globalized search in the entire solution space [4, 17]. Utilizing the PSO algorithm's optimal ability, if given enough time, the PSO clustering algorithm we proposed could generate more compact clustering results from the document datasets than the traditional K-means clustering algorithm. However, in order to cluster the large document datasets, PSO requires much more iteration (generally more than 500 iterations) to converge to the optima than the K-mean algorithm does. Although the PSO algorithm is inherently parallel and can be implemented using parallel hardware, such as a computer cluster, the computation requirement for clustering extremely huge document datasets is still high. In terms of execution time, the K-means algorithm is the most efficient for the large dataset [1]. The K-means algorithm tends to converge faster than the PSO, but it usually only finds the local maximum. Therefore, we face a dilemma regarding choosing the algorithm for clustering a large

document dataset. Base on this reason, we proposed a hybrid PSO+K-means document clustering algorithm.

HYBRID PSO+K-MEANS ALGORITHM

In the hybrid PSO+K-means algorithm, the multi-dimensional document vector space is modeled as a problem space. Each term in the document dataset represents one dimension of the problem space. Each document vector can be represented as a dot in the problem space. The whole document dataset can be represented as a multiple dimension space with a large number of dots in the space. The hybrid PSO+K-means algorithm includes two modules, the PSO module and K-means module. At the initial stage, the PSO module is executed for a short period to search for the clusters' centroid locations. The locations are transferred to the K-means module for refining and generating the final optimal clustering solution.

The PSO module: A single particle in the swarm represents one possible solution for clustering the document collection. Therefore, a swarm represents a number of candidate clustering solutions for the document collection. Each particle maintains a matrix $X_i = (C_1, C_2, ..., C_i, .., C_k)$, where C_i represents the *ith* cluster centroid vector and k is the cluster number. At each iteration, the particle adjusts the centroid vector' position in the vector space according to its own experience and those of its neighbors. The average distance between a cluster centroid and a document is used as the fitness value to evaluate the solution represented by each particle. The fitness value is measured by the equation below:

$$f = \frac{\sum_{i=1}^{N_c} \{ \frac{\sum_{j=1}^{p_i} d(o_i, m_{ij})}{p_i} \}}{N_c} \qquad (7)$$

where m_{ij} denotes the *jth* document vector, which belongs to cluster i; O_i is the centroid vector of ***i***th cluster; $d(o_i, m_{ij})$ is the distance between document m_{ij} and the cluster centroid O_i.; P_i stands for the document number, which belongs to cluster C_i; N_c stands for the cluster number.

The PSO module can be summarized as:

(1) At the initial stage, each particle randomly chooses k numbers of document vectors from the document collection as the cluster centroid vectors.

(2) For each particle:

(a) Assigning each document vector in the document set to the closest centroid vector.

(b) Calculating the fitness value based on equation 7.

(c) Using the velocity and particle position to update equations 5a and 5b and to generate the next solutions.

(3) Repeating step (2) until one of following termination conditions is satisfied.

(a) The maximum number of iterations is exceeded

or

(b) The average change in centroid vectors is less than a predefined value.

The K-means module: The K-means module will inherit the PSO module's result as the initial clustering centroids and will continue processing the optimal centroids to generate the final result. The K-means module can be summarized as:

(1) Inheriting cluster centroid vectors from the PSO module.

(2) Assigning each document vector to the closest cluster centroids.

(3) Recalculating the cluster centroid vector c_j using equation 8.

$$c_j = \frac{1}{n_j} \sum_{\forall d_j \in S_j} d_j \qquad (8)$$

where d_j denotes the document vectors that belong to cluster S_j; c_j stands for the centroid vector; n_j is the number of document vectors belong to cluster S_j.

(4) Repeating step 2 and 3 until the convergence is achieved.

In the PSO+K-means algorithm, the ability of globalized searching of the PSO algorithm and the fast convergence of the K-means algorithm are combined. The PSO algorithm is used at the initial stage to help discovering the vicinity of the optimal solution by a global search. The result from PSO is used as the initial seed of the K-means algorithm, which is applied for refining and generating the final result.

EXPERIMENTS AND RESULTS

Datasets: We used four different document collections to compare the performance of the K-

means, PSO and hybrid PSO+K-means algorithms with different combination models. These document datasets are derived from the TREC-5, TREC-6, and TREC-7 collections [19]. A description of the test datasets is given in Table 1. In those document datasets, the very common words (e.g. function words: "a", "the", "in", "to"; pronouns: "I", "he", "she", "it") are stripped out completely and different forms of a word are reduced to one canonical form by using Porter's algorithm [13]. In order to reduce the impact of the length variations of different documents, each document vector is normalized so that it is of unit length. The document number in each dataset ranges from 204 to 878. The term numbers of each dataset are all over 5000.

Table 1: Summary of text document datasets

Data	Number of documents	Number of terms	Number of classes
Dataset1	414	6429	9
Dataset2	313	5804	8
Dataset3	204	5832	6
Dataset4	878	7454	10

Experimental setup: The K-means, PSO and PSO+K-means clustering approaches are applied on the four datasets, respectively. The Euclidian distance measure and cosine correlation measure are used as the similarity metrics in each algorithm, respectively. Some researchers [12, 20] proposed using the K-means+PSO hybrid algorithm for clustering low dimension datasets. They argue that the K-means algorithm tends to converge faster than other clustering algorithms, but usually with a less accurate clustering. The performance of the clustering algorithm can be improved by seeding the initial swarm with the result of the K-means algorithm. To compare the performance of different kinds of hybrid algorithms, we applied K-means+PSO and K-means+PSO+K-means algorithms on the four datasets, respectively.

We have noticed that K-means clustering algorithms can converge to a stable solution within 20 iterations when applied to most document datasets. The PSO usually needs to repeat for more than 100 iterations to generate a stable solution. For an easy comparison, the K-means and PSO approaches run 50 iterations. In the K-means+PSO approach, the K-means algorithm is first executed for 25 iterations. The result of the K-means algorithm is then used as the initial cluster centroid in the PSO algorithm, and the PSO algorithm executes for another 25 iterations to generate the final result. The PSO+k-means approach has the same execution procedure, except that it first executes the PSO algorithm for 25 iterations and uses the PSO result as the initial seed for the K-means algorithm. In the K-means+PSO+K-means approach, the K-means algorithm is first executed for 25 iterations. The result of the K-means algorithm is then used as the initial cluster centroid in the PSO algorithm and the PSO algorithm executes for 25 iterations, the global best solution result from the PSO algorithm is used as the initial cluster centroid of the K-means algorithm, and the K-means algorithm executes for another 25 iterations to generate the final result. In these five different algorithms, the total executing iteration number for K-means, PSO, K-means+PSO, and PSO+K-means is 50. The total executing number for the K-means+PSO+K-means iteration approach is 75.

No parameter needs to be set up for the K-means algorithm. In the PSO clustering algorithm, because of the extremely high dimensional solution space of the text document datasets, we choose 50 particles for all the PSO algorithms instead of choosing 20 to 30 particles recommended in [4, 17]. In the PSO algorithm, the inertia weight w is initially set as 0.72 and the acceleration coefficient constants c1 and c2 are set as 1.49. These values are chosen based on the results of [17]. In the PSO algorithm, the inertia weight will reduce 1% in value at each iteration to ensure good convergence. However, the inertia weight in all hybrid algorithms is kept constant to ensure a globalized search.

Results: The fitness equation 7 is used not only in the PSO algorithm for the fitness value calculation, but also in the evaluation of the cluster quality. It indicates the value of the average distance between documents and the cluster centroid to which they belong (ADVDC). The smaller the ADVDC value, the more compact the clustering solution is. Table 2 demonstrates the experimental results by using the K-means, PSO, K-means+PSO, PSO+K-means and K-means+PSO+K-means respectively. Ten simulations are performed separately. The average ADVDC values and standard division are recorded in Table 2. To illustrate the convergence behavior of different clustering algorithms, the clustering ADVDC values at each iteration are recorded when these five algorithms are applied on datasets separately.

As shown in Table 2, the PSO+K-means hybrid clustering approach generates the clustering result that has the lowest ADVDC value for all four datasets using the Euclidian similarity metric and the Cosine

Table 2: Performance comparison of K-means, PSO, K-means+PSO, PSO+K-means and K-means+PSO+K-means

		ADVDC value				
		K-means	PSO	PSO+K-means	K-means+PSO	K-means+PSO +K-means
Dataset 1	Euclidian	8.238±0.090	6.759±0.956	4.556±1.405	8.244±0.110	8.138±0.138
	Cosine	8.999±0.150	10.624±0.406	7.690±0.474	9.083±0.118	8.910±0.231
Dataset 2	Euclidian	7.245±0.166	6.362±1.032	4.824±1.944	7.243±0.114	7.292±0.086
	Cosine	8.074±0.200	9.698±0.435	7.676±0.172	8.126±0.072	8.140±0.152
Dataset 3	Euclidian	4.788±0.089	4.174±0.207	2.550±0.746	4.662±0.315	4.799±0.247
	Cosine	5.093±0.120	5.750±0.395	4.355±0.252	5.078±0.213	4.985±0.267
Dataset 4	Euclidian	9.09±0.097	9.311±1.010	6.004±2.666	8.936±0.285	8.861±0.363
	Cosine	10.22±0.402	12.874±0.593	9.547±0.237	10.363±0.132	10.221±0.343

correlation similarity metric. The results from the PSO approach have improvements compared to the results of the K-means approach when using the Euclidian similarity metric. However, when the similarity metric is changed with the cosine correlation metric, the K-means algorithm has a better performance than the PSO algorithm. The K-means+PSO and K-means+PSO+K-means approaches do not have significant improvements compared to the result of the K-means approach.

Figure 1 illustrates the convergence behaviors of these algorithms on the document dataset 1 using the Euclidian distance as a similarity metric. In Figure 1, the K-means algorithm converges quickly but prematurely with high quantization error. As shown in figure 1, the ADVDC value of the K-means algorithm is sharply reduced from 11.3 to 8.2 within 10 iterations and fixed at 8.2. In Figure 1, it is hard to separate the curve lines that represent the K-means+PSO and K-means+PSO+K-means approaches from the K-means approach. The three lines nearly overlap each other, which indicates these three algorithms have nearly the same convergence behavior. The PSO approach's ADVDC value is quickly converged from 11.3 to 6.7 within 30 iterations. The reduction of the ADVDC value in PSO is not as sharp as in K-means and becomes smoothly after 30 iterations. The curvy line's tendency indicates that if more iterations are executed, the distance average value may reduce further although the reduction speed will be very slow.

The PSO+K-means approach's performance significantly improves. In the first 25 iterations, the PSO+K-means algorithm has similar convergence behavior because within 1 to 25 iterations, the PSO and the PSO+K-means algorithms execute the same PSO optimal code. After 25 iterations, the ADVDC value has a sharp reduction with the value reduced from 6.7 to 4.7 and maintains a stable value within 10 iterations.

Discussion: Using hybrid algorithms for boosting the clustering performance is not a novel idea. However, most of hybrid algorithms use K-means algorithm for generating the initial clustering seeds for other optimal algorithms. To the best of the author's knowledge, there no hybrid algorithm that uses PSO optimal algorithm generating initial seed for K-means clustering. In [20], Merwe and Engelbrecht argued that the performance of the PSO clustering algorithm could be improved by seeding the initial swarm with the result of the K-means algorithm. They conducted simulations on some low dimension datasets with 10 particles and 1000 iterations. However, from the experimental results in Table 2, we noticed the K-means +PSO algorithm does not show any improvement in the large document datasets. The differences between the experiments in the present research and the experiment in [20] are (a) the datasets used here are document data with more than 5000 terms, which is also the dimensional number of the PSO searching space; (b) The iterations running in each simulation are no more than 75. In the K-means+PSO approach, the K-means algorithm generates a local optima result, which is used as the initial status of one particle in the PSO algorithm. The high fitness value of this initial particle's value will directly force the particle to start the optimal refining stage. Although other particles are randomly deployed in the searching space, the high average distance value

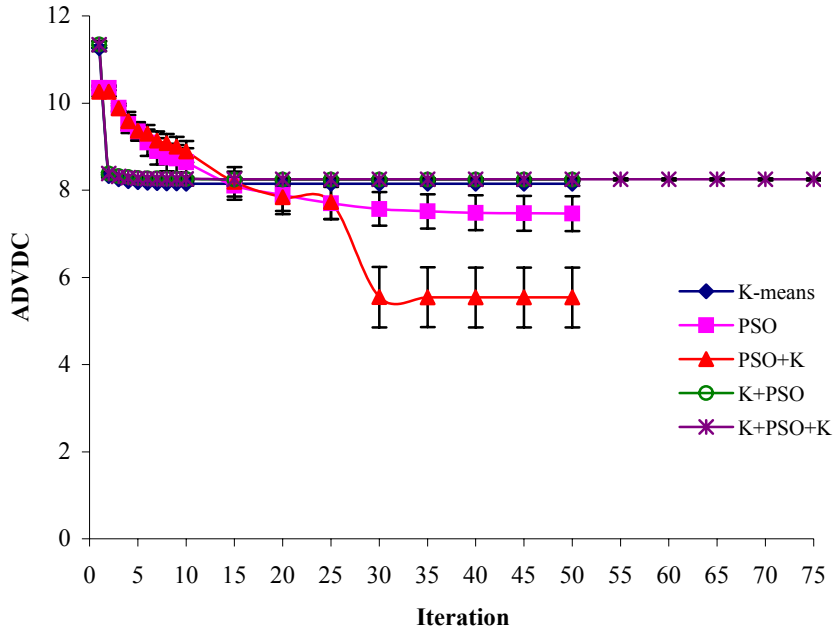

Figure 1: The convergence behaviors of different clustering algorithm (PSO+K: the PSO+K-means Algorithm, K+PSO: the K-means + PSO algorithm, K+PSO+K: the K-means +PSO+K-means algorithm)

of the initial solution that the PSO algorithm inherited from the K-means algorithm will attract other particles to converge quickly into the vicinity of the local optima. The performance results in Table 2 and the convergence behaviors of the K-means+PSO approach in Figure 1 illustrate this.

The PSO+K-means algorithm generates the highest clustering compact result in the experiments. The average distance value is the lowest. In the PSO+K-means algorithm clustering experiment, although 25 iterations is not enough for the PSO to discover the optimal solution, it has a high possibility that one particle's solution is located in the vicinity of the global solution. The result of the PSO is used as the initial seed of the K-means algorithm and the K-means algorithm can quickly locate the optima with a low distance average value. Comparison of the performance of these five approaches in our experiments illustrates that the sequence of hybrid K-means algorithms is very important.

CONCLUSION

In this study, we presented a document clustering algorithm, the PSO+K-means algorithm, which can be regarded as a hybrid of the PSO and K-means algorithms. In the general PSO algorithm, PSO can conduct a globalized searching for the optimal clustering, but requires more iteration numbers and computation than the K-means algorithm does. The K-means algorithm tends to converge faster than the PSO algorithm, but usually can be trapped in a local optimal area. The PSO+K-means algorithm combines the ability of the globalized searching of the PSO algorithm and the fast convergence of the K-means algorithm and can avoid the drawback of both algorithms. The algorithm includes two modules, the PSO module and the K-means module. The PSO module is executed for a short period at the initial stage to discover the vicinity of the optimal solution by a global search and at the same time to avoid consuming high computation. The result from the PSO module is used as the initial seed of the K-means module. The K-means algorithm will be applied for refining and generating the final result. Our experimental results illustrate that using this hybrid PSO+K-means algorithm can generate higher compact clustering than using either PSO or K-means alone. The results from the three different hybrid K-means algorithms illustrates that performing the K-means in advance of the PSO module in the hybrid algorithm will reduce the globalized searching ability of the PSO algorithm and lower the whole algorithm's performance.

ACKNOWLEDGMENT

Oak Ridge National Laboratory is managed by UT-Battelle LLC for the U.S. Department of Energy under contract number DE-AC05_00OR22725.

REFERENCES

1. Al-Sultan, K. S. and Khan, M. M. 1996. Computational experience on four algorithms for the hard clustering problem. Pattern Recogn. Lett. 17, 3, 295–308.
2. Anderberg, M. R., 1973. Cluster Analysis for Applications. Academic Press, Inc., New York, NY.
3. Berkhin, P., 2002. Survey of clustering data mining techniques. Accrue Software Research Paper.
4. Carlisle, A. and Dozier, G., 2001. An Off-The-Shelf PSO, Proceedings of the 2001 Workshop on Particle Swarm Optimization, pp. 1-6, Indianapolis, IN
5. Cios K., Pedrycs W., Swiniarski R., 1998. Data Mining – Methods for Knowledge Discovery, Kluwer Academic Publishers.
6. Cui X., Potok T. E., 2005. Document Clustering using Particle Swarm Optimization, IEEE Swarm Intelligence Symposium 2005, Pasadena, California.
7. Eberhart, R.C., and Shi, Y., 2000. Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization, 2000 Congress on Evolutionary Computing, vol. 1, pp. 84-88.
8. Everitt, B., 1980. Cluster Analysis. 2nd Edition. Halsted Press, New York.
9. Jain A. K., Murty M. N., and Flynn P. J., 1999. Data Clustering: A Review, ACM Computing Survey, Vol. 31, No. 3, pp. 264-323.
10. Hartigan, J. A. 1975. Clustering Algorithms. John Wiley and Sons, Inc., New York, NY.
11. Kennedy J., Eberhart R. C. and Shi Y., 2001. Swarm Intelligence, Morgan Kaufmann, New York.
12. Omran, M., Salman, A. and Engelbrecht, A. P., 2002. Image classification using particle swarm optimization. Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning 2002 (SEAL 2002), Singapore. pp. 370-374.
13. Porter, M.F., 1980. An Algorithm for Suffix Stripping. Program, 14 no. 3, pp. 130-137.
14. Salton G., 1989. Automatic Text Processing. Addison-Wesley.
15. Salton G. and Buckley C., 1988. Term-weighting approaches in automatic text retrieval. Information Processing and Management, 24 (5): pp. 513-523.
16. Selim, S. Z. And Ismail, M. A. 1984. K-means type algorithms: A generalized convergence theorem and characterization of local optimality. IEEE Trans. Pattern Anal. Mach. Intell. 6, 81–87.
17. Shi, Y. H., Eberhart, R. C., 1998. Parameter Selection in Particle Swarm Optimization, The 7th Annual Conference on Evolutionary Programming, San Diego, CA.
18. Steinbach M., Karypis G., Kumar V., 2000. A Comparison of Document Clustering Techniques. TextMining Workshop, KDD.
19. TREC. 1999. Text Retrieval Conference. http://trec.nist.gov.
20. Van D. M., Engelbrecht, A. P., 2003. Data clustering using particle swarm optimization. Proceedings of IEEE Congress on Evolutionary Computation 2003 (CEC 2003), Canbella, Australia. pp. 215-220.
21. Zhao Y. and Karypis G., 2004. Empirical and Theoretical Comparisons of Selected Criterion Functions for Document Clustering, Machine Learning, 55 (3): pp. 311-331.