Document Clustering using Particle Swarm Optimization

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

Abstract

Fast and high-quality document clustering algorithms play an important role in effectively navigating, summarizing, and organizing information. Recent studies have shown that partitional clustering algorithms are more suitable for clustering large datasets. However, the K-means algorithm, the most commonly used partitional clustering algorithm, can only generate a local optimal solution. In this paper, we present a Particle Swarm Optimization (PSO) document clustering algorithm. Contrary to the localized searching of the K-means algorithm, the PSO clustering algorithm performs a globalized search in the entire solution space. In the experiments we conducted, we applied the PSO, K-means and hybrid PSO clustering algorithm on four different text document datasets. The number of documents in the datasets ranges from 204 to over 800, and the number of terms ranges from over 5000 to over 7000. The results illustrate that the hybrid PSO algorithm can generate more compact clustering results than the K-means algorithm.

1. 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. The motivation behind clustering a set of data is to find inherent structure in the data and to 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, 23, 25]. There are two major clustering techniques: "Partitioning" and "Hierarchical" [9]. Most document clustering algorithms can be classified into these two groups. The 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 [23].

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 [23, 25]. 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 [11]. This algorithm is simple, straightforward and is based on the firm foundation of analysis of variances. In addition to the K-means algorithm, several algorithms, such as Genetic Algorithm (GA) [10, 18] and Self-Organizing Maps (SOM) [14], have been used for document clustering. Particle Swarm Optimization (PSO) [13] is another computational intelligence method that has already been applied to image clustering and other low dimensional datasets [15, 16]. However, to the best of the author's knowledge, PSO has not been used to cluster text documents. In this study, a document clustering algorithm based on PSO is proposed.

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 K-means and PSO optimal algorithm. The PSO clustering algorithms are described in Section 4. Section 5 provides the detailed experimental setup and results for comparing the

0-7803-8916-6/05/$20.00 ©2005 IEEE

performance of the PSO algorithm with the K-means approaches. The discussion of the experiment's results is also presented. The conclusion is in Section 6.

2. Preliminaries

2.1 Document representation

In most clustering algorithms, the dataset to be clustered is represented as a set of vectors $X=\{x_1, x_2, \ldots, x_n\}$, 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 $d=\{w_1, w_2, \ldots..w_n\}$, where w_i $(i = 1,2,\ldots,n)$ 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, 19]. The weight of term i in document j is given in equation 1:

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

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.

2.2 The similarity metric

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

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

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, most algorithms use 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:

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

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 [19, 20], given by

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

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.

3. Background

3.1 K-means Clustering Algorithm

The K-means algorithm is simple, straightforward and is based on the firm foundation of analysis of variances. It clusters a group of data vectors into a predefined number of clusters. It starts with randomly initial cluster centroids and keeps reassigning the data objects in the dataset to cluster centroids based on the similarity between the data object and the cluster centroid. 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 K-means algorithm can be summarized as:

(1) Randomly select cluster centroid vectors to set an initial dataset partition.
(2) Assign each document vector to the closest cluster centroids.
(3) Recalculate the cluster centroid vector c_j using equation 5.

$$c_j = \frac{1}{n_j} \sum_{\forall d_j \in S_j} d_j \qquad (5)$$

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 that belong to cluster S_j.

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

The main drawback of the K-means algorithm is that the result is sensitive to the selection of the initial cluster centroids and may converge to the local optima [21]. Therefore, the initial selection of the cluster centroids affects the main processing of the K-means and the partition result of the dataset as well. The 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 other techniques, the K-means would work well in refining the clustering centroids to find the optimal clustering centers [2].

3.2 PSO Algorithm

PSO was originally developed by Eberhart and Kennedy in 1995 [13], and was inspired by the social behavior of a flock of birds. 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 a 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. At every generation, the particle's new location is computed by adding the particle's current velocity, v_{id}, to its location, x_{id}. Mathematically, given a multi-dimensional problem space, the ***ith*** particle changes its velocity and location according to the following equations [13]:

$$v_{id} = w \times v_{id} + c_1 \times rand_1 \times (p_{id} - x_{id}) + c_2 \times rand_2 \times (p_{gd} - x_{id}) \quad (6a)$$

$$x_{id} = x_{id} + v_{id} \quad (6b)$$

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).

4. Description of the PSO Clustering Algorithm

In the past several years, PSO has been proven to be both effective and quick to solve some optimization problems [13]. It was successfully applied in many research and application areas [4, 7, 13]. In document clustering research area, it is possible to view the clustering problem as an optimization problem that locates the optimal centroids of the clusters rather than to find an optimal partition. This view offers us a chance to apply PSO optimal algorithm on the clustering solution.

Similar to other partitional clustering algorithms, the objective of the PSO clustering algorithm is to discover the proper centroids of clusters for minimizing the intra-cluster distance as well as maximizing the distance between clusters. The PSO algorithm performs a globalized searching for solutions whereas most other partitional clustering procedures perform a localized searching. In localized searching, the solution obtained is usually located in the vicinity of the solution obtained in the previous step. For example, the K-means clustering algorithm uses the randomly generated seeds as the initial clusters' centroids and refines the position of the centroids at every iteration. The refining process of the K-means algorithm indicates the algorithm only explores the very narrow vicinity surrounding the initial randomly generated centroids.

The whole clustering behavior of the PSO clustering algorithm can be classed into two stages: a global searching stage and a local refining stage. At the initial iterations, based on the PSO algorithm's particle velocity updating equation 6a, the particle's initial velocity v_{id}, the two randomly generated values ($rand_1$, $rand_2$) at each generation and the inertia weight factor w provide the necessary diversity to the particle swarm by changing the momentum of particles to avoid the stagnation of particles at the local optima. Multiple

particles parallel searching, using multiple different solutions at a time, can explore more area in the problem space. The initial iterations can be classified as the global searching stage. After several iterations, the particle's velocity will gradually reduce and the particle's explore area will shrink while the particle will approach the optimal solution. The global searching stage gradually changes to the local refining stage. By selecting different parameters in the PSO algorithm, we can control the shift time from the global searching stage to the local refining stage. The later the particle shift from the global searching stage to local refining stage, greater the possibility that it can find the global optimal solution.

4.1 The Basic PSO Clustering Algorithm

In the PSO document clustering 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.

One 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 $X_i = (C_1, C_2, \ldots, C_i, .., C_k)$, where C_i represents the ith cluster centroid vector and k is the number of clusters. According to its own experience and those of its neighbors, the particle adjusts the centroid vector' position in the vector space at each generation. The average distance of documents to the cluster centroid (ADDC) 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 the ith cluster; $d(o_i, m_{ij})$ is the distance between document m_{ij} and the cluster centroid O_i; P_i stands for the number of documents, which belongs to cluster C_i; and N_c stands for the number of clusters.

The PSO algorithm can be summarized as:

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

(2) For each particle:

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

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

(c) Using the velocity and particle position to update equations 6a and 6b and to generate the next solutions.

(3) Repeat step (2) until one of the 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.

4.2 The Hybrid PSO clustering

Merwe's research [15] indicates that utilizing the PSO algorithm's optimal ability, if given enough time, the PSO clustering algorithm could generate more compact clustering results from the low dimensional dataset than the traditional K-means clustering algorithm. However, when clustering large document datasets, the slow shift from the global searching stage to the local refining stage causes the PSO clustering algorithm to require many more iterations to converge to the optima in the refining stage than the K-means algorithm requiring. Although the PSO algorithm is inherently parallel and can be implemented using parallel hardware, such as a computer cluster, the computation requirement for clustering large document dataset is still high. In our experiments, it needs more than 500 iterations for the PSO algorithm to converge to the optimal result for a document dataset that includes 800 documents. The K-means algorithm only requires 10 to 20 iterations.

Although the PSO algorithm generates much better clustering result than the K-means algorithm does, in terms of execution time, the K-means algorithm is more efficient for large datasets [1]. For this reason, we present a hybrid PSO approach that uses K-means algorithm to replace the refining stage in the PSO algorithm. In the hybrid PSO algorithm, the algorithm includes two modules, the PSO module and the K-means module. The global searching stage and local refine stage are accomplished by those two modules, respectively. In the initial stage, the PSO module is executed for a short period (50 to 100 iterations) 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. The whole approach can be summarized as:

(1)Start the PSO clustering process until the maximum number of iterations is exceeded
(2) Inherit clustering result from PSO as the initial centroid vectors of K-means module.
(3)Start K-means process until maximum number of iterations is reached.

5. Experiments and Results

5.1Datasets

We used four different document collections to compare the performance of the K-means and PSO algorithms. These document datasets are derived from Text REtrieval Conference (TREC) collections [24]. 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 [17]. 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.

5.2 Experimental setup

The K-means, PSO and hybrid PSO 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. For an easy comparison, the K-means and PSO approaches run 100 iterations in each experiment. In the hybrid PSO approach, it first executes the PSO algorithm for 90 iterations and uses the PSO result as the initial seed for the K-means module and the K-means module executes for 10 iterations to generate the final result. The total iterations of hybrid PSO is same as K-means and PSO.

No parameter needs to be set up for the K-means algorithm. In the PSO clustering algorithm, we choose 50 particles for all the PSO algorithms instead of choosing 20 to 30 particles recommended in [4, 22]. Because the text document datasets is a high dimensional problem space, increasing the particle number in the algorithm can increase the chance for finding the optimal solution. 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 [22]. In the PSO approach, the inertia weight w is reduced by 1% at each generation to ensure good convergence.

Table 1. Summary of text document datasets

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

5.3 Results and Discussions

The fitness equation 7 is used not only in the PSO algorithm for fitness value calculation, but also in the evaluation of the cluster quality. It indicates the value of the average distance (ADDC) between documents and the cluster centroid to which they belong. The smaller the ADDC value, the more compact the clustering solution is. Table 2 demonstrates the experimental results by using the K-means, PSO, hybrid PSO respectively. Ten simulations are performed for each algorithm. The average ADDC values are recorded in Table 2.

As shown in Table 2, the hybrid PSO approach generates the clustering result with the lowest ADDC value for all four datasets using the Euclidian similarity metric and the Cosine correlation similarity metric. Because 100 iterations is not enough for the PSO algorithm to converge to an optimal solution, the result values in the table 2 indicate that 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 to the cosine correlation metric, the K-means algorithm has a better performance than the PSO algorithm.

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. As shown in Figure 1, the ADDC value of the K-means algorithm is sharply reduced from 11 to 8.2 within 10 iterations and fixed at 8.2. The PSO approach's ADDC value is quickly converged from 11 to 8.1 within 30 iterations. The reduction of the ADDC value in PSO is not as sharp as in K-means and

becomes smooth after 30 iterations. The curvy line's indicate that if more iterations are executed, the average distance value may reduce further although the reduction speed will be very slow. The hybrid PSO approach's performance significantly improves. In the first 90 iterations, the hybrid PSO approach has similar convergence behavior as PSO approach because within 1 to 90 iterations, the PSO and the hybrid PSO algorithms execute the same PSO optimal code. After 90 iterations, the ADDC value has a sharp reduction with the value reduced from 8.1 to 6.4 and maintains a stable value within 10 iterations.

Table 2: Performance comparison of K-means, PSO, hybrid PSO algorithms

		ADDC value		
		K-means	PSO	Hybrid PSO
Dataset1	Euclidian	8.17817	8.11009	6.38039
	Cosine	8.96442	10.41271	8.14551
Dataset2	Euclidian	7.26175	6.25172	4.51753
	Cosine	8.07653	9.57786	7.21153
Dataset3	Euclidian	4.59539	4.14896	2.25961
	Cosine	4.97171	5.71146	4.00555
Dataset4	Euclidian	9.08759	8.62794	6.37872
	Cosine	10.1739	12.8927	9.5379

The hybrid PSO algorithm generates the highest clustering compact result from the experiments. The average distance value is the lowest. In the hybrid PSO approach experiment, 90 iterations are not enough for the PSO module to discover the optimal solution, however, there is a high possibility that one particle's solution is located in the vicinity of the global solution or near a global solution. The result of the PSO module is used as the initial seed of K-means module and the K-means module can quickly locate the optima with a low ADDC value.

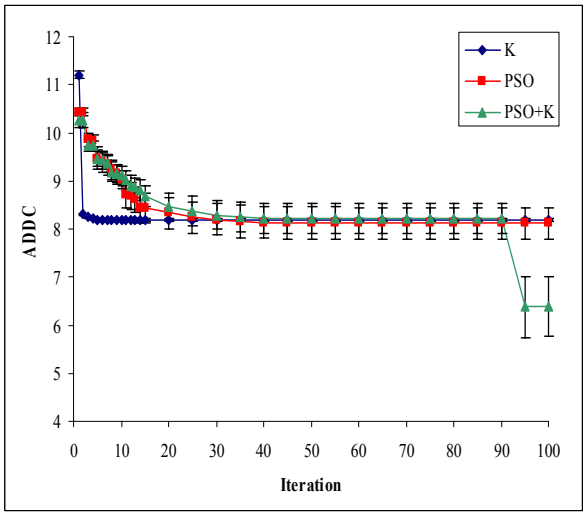

Figure 1: The convergence behaviors of different clustering algorithm (K-means, PSO and hybrid PSO algorithms)

6. Conclusion

In this study, a document clustering algorithm based on the PSO algorithm is proposed. In the PSO clustering algorithm, the clustering behavior can be classified into two stages: the global searching stage and the local refining stage. The global searching stage guarantees each particle searches widely enough to cover the whole problem space. The refining stage makes all particles converge to the optima when a particle reaches the vicinity of the optimal solution. For a large dataset, conventional 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 hybrid PSO algorithm combines the ability of globalized searching of the PSO algorithm and the fast convergence of the K-means algorithm and avoids 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 searching 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 is applied for refining and generating the final result. Our experimental results illustrate that using this hybrid PSO algorithm can generate higher compact clustering than using either the PSO or the K-means alone.

7. 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., Hardin T., Ragade R. K., and Elmaghraby A. S., A Swarm Approach for Emission Sources Localization, The 16th IEEE International Conference on Tools with Artificial Intelligence, 2004, Boca Raton, Florida, USA

[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] Jones, Gareth, Robertson, Alexander M., Santimetvirul, Chawchat and Willett, P., 1995. Non-hierarchic document clustering using a genetic algorithm. Information Research, 1(1).

[11] Hartigan, J. A. 1975. Clustering Algorithms. John Wiley and Sons, Inc., New York, NY.

[12] Hardin T., Cui X., Ragade R. K., Graham J. H., and Elmaghraby A. S., (2004). A Modified Particle Swarm Algorithm for Robotic Mapping of Hazardous Environments, The 2004 World Automation Congress, SEVILLE, Spain

[13] Kennedy J., Eberhart R. C. and Shi Y., 2001. Swarm Intelligence, Morgan Kaufmann, New York.

[14] Merkl D., 2002. Text mining with self-organizing maps. Handbook of data mining and knowledge, pp. 903-910, Oxford University Press, Inc. New York.

[15] Merwe V. D. and 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.

[16] 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.

[17] Porter, M.F., 1980. An Algorithm for Suffix Stripping. Program, 14 no. 3, pp. 130-137.

[18] Raghavan, V. V. AND Birchand, K. 1979. A clustering strategy based on a formalism of the reproductive process in a natural system. Proceedings of the Second International Conference on Information Storage and Retrieval, 10–22.

[19] Salton G., 1989. Automatic Text Processing. Addison-Wesley.

[20] Salton G. and Buckley C., 1988. Term-weighting approaches in automatic text retrieval. Information Processing and Management, 24 (5): pp. 513-523.

[21] 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.

[22] Shi, Y. H., Eberhart, R. C., 1998. Parameter Selection in Particle Swarm Optimization, The 7th Annual Conference on Evolutionary Programming, San Diego, CA.

[23] Steinbach M., Karypis G., Kumar V., 2000. A Comparison of Document Clustering Techniques. TextMining Workshop, KDD.

[24] TREC. 1999. Text Retrieval Conference. http://trec.nist.gov.

[25] Zhao Y. and Karypis G., 2004. Empirical and Theoretical Comparisons of Selected Criterion Functions for Document Clustering, Machine Learning, 55 (3): pp. 311-331.