Hybridization of K-means and Harmony Search Methods for Web Page Clustering

Rana Forsati[1] MohammadReza Meybodi [2] Mehrdad Mahdavi [3] AzadehGhari Neiat[4]

[1]Department of Computer Engineering, Islamic Azad University, Qazvin Branch, Qazvin, Iran
[2]Department of Computer Engineering, AmirKabir University of Technology, Tehran, Iran
[3]Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
[4]Department of Computer Engineering, Islamic Azad University, North Tehran Branch, Tehran, Iran
forsati@kiau.ac.ir mmeybodi@aut.ac.ir mahdavi@ce.sharif.edu a_ghariniyat@yahoo.com

Abstract

Clustering is currently one of the most crucial techniques for dealing with massive amount of heterogeneous information on the web, which is beyond human being's capacity to digest. Recent studies have shown that the most commonly used partitioning-based clustering algorithm, the K-means algorithm, is more suitable for large datasets. However, the K-means algorithm can generate a local optimal solution. In this paper we present novel harmony search clustering algorithms that deal with documents clustering based on harmony search optimization method. By modeling clustering as an optimization problem, first, we propose a pure harmony search based clustering algorithm that finds near global optimal clusters within a reasonable time. Contrary to the localized searching of the K-means algorithm, the harmony search clustering algorithm performs a globalized search in the entire solution space. Then harmony clustering is integrated with the K-means algorithm in three ways to achieve better clustering. The proposed algorithms improve the K-means algorithm by making it less dependent on the initial parameters such as randomly chosen initial cluster centers, hence more stable. In the experiments we conducted, we applied the proposed algorithms, K-means clustering algorithm on five different document datasets. Experimental results reveal that the proposed algorithms can find better clusters when compared to K-means and the quality of clusters is comparable and converge to the best known optimum faster than it.

1. Introduction

In recent years as millions of scientific publications, commercial reports and unstructured web pages available on the Internet and Information Retrieval (IR) systems, high-quality document clustering plays more and more important role in the applications such as information retrieval or filtering, Web data mining, and Web data management. In general, Clustering involves dividing a set of documents into a specified number of groups. The documents within each group should exhibit a large degree of similarity while the similarity among different clusters should be minimized. Some of the more familiar clustering methods are: partitioning algorithms based on dividing entire data into dissimilar groups, hierarchical methods, density and grid based clustering, some graph based methods and etc [1,2]. In general, Clustering algorithms can be broadly classified into two categories: hierarchal and partitional algorithms. Hierarchical algorithms used to build tree structure from data [3], while partitioning methods cluster the data in a single level [4,5,6].

Partitioning methods try to partition a collection of documents into a set of groups, so as to maximize a pre-defined fitness value. Although hierarchical methods are often said to have better quality clustering results, usually they do not provide the reallocation of documents, which may have been poorly classified in the early stages of the text analysis[1]. Moreover, the time complexity of hierarchical methods is quadratic. On the other hands, in recent years the partitioning clustering methods are well suited for clustering a large document dataset due to their relatively low computational requirements [4,5,6,8,9]. The time complexity of the partitioning techniques is almost linear, which makes them widely used. The best known method in partitioning clustering is K-means algorithm [7].

Although K-means algorithm is simple, straightforward and easy to be implemented and works fast in most situations, it suffers from some major drawbacks that make it inappropriate for many applications. The first disadvantage is that the number of clusters K must be specified prior to application. Also, since the summary statistic is a mean of the

978-0-7695-3496-1/08 $25.00 © 2008 IEEE
DOI 10.1109/WIIAT.2008.370

values for each cluster, so, the individual members of the cluster can have a high variance and the mean may not be a good summary of the cluster members. In addition, as the number of clusters grow, for example to thousands of clusters, K-means clustering becomes untenable, approaching the O(n^2) comparisons where n is the number of documents. However, for relatively few clusters and a reduced set of pre-selected words, K-means can do well [10]. The other major drawback of K-means algorithm is sensitivity to initialization. Finally, the K-means algorithm converges to local optima and the final clusters may not be the optimal solution. To deal with the limitations that exist in traditional partition clustering methods especially K-means, recently, new concepts and techniques have been entered into web document clustering, with respect to increasing need for the web knowledge extraction. One of these techniques is optimization methods that try to optimize a pre-defined function. So, it can be very useful in web document clustering.

Since the partitioning-based clustering algorithms use information about the collection of documents when they partition the dataset into a certain number of clusters, so, the optimization methods can be employed for partitioning. Optimization techniques define a global function and by traversing the search space, try to optimize its value. Regarding to this advantage, K-means can be considered as an optimization method. In addition to the K-means algorithm, many different solutions have been proposed for clustering in the context of other meta-heuristics, In principle, any general purpose optimization method can serve as the basis of this approach.

Harmony Search (HS) [11] is a new meta-heuristic optimization method imitating the music improvisation process where musicians improvise their instruments' pitches searching for a perfect state of harmony. Harmony search algorithm had been very successful in a wide variety of optimization problems.

In fact, in optimization problems, we want to search the solution space and in HS this search can be done more efficiently. Since stochastic optimization approaches are good at avoiding convergence to a locally optimal solution, these approaches could be used to find a globally optimal solution. Typically the stochastic approaches take a large amount of time to converge to a globally optimal partition.

As the behavior of the K-means algorithm mostly is influenced by the number of clusters specified and the random choice of initial cluster centers, in this study, we concentrate on the latter, where the results are less dependent on the initial cluster centers chosen, hence more stabilized, by introducing different algorithms based on HS for clustering documents. The fist algorithm, called Harmony Search Clustering (HSCLUST), which is good at finding promising areas of the search space but not as good as K-means at fine tuning within those areas. To improve the algorithm, hybrid technique using the K-means and the harmony search optimization heuristic proposed to solve clustering problem. The hybrid methods improve the K-means algorithm by making it less dependent on the initial parameters such as randomly chosen initial cluster centers, hence more stable. Three different versions of the hybrid clustering, depending on the stage when we carry out the K-means algorithm, introduced. These methods combine power of the HSCLUST with the speed of a K-means. It seems hybrid algorithms that combine two ideas can result in an algorithm that can outperform either one individually. The advantage of these algorithms over the K-means is that the influence of the improperly chosen initial cluster centers will be diminished after the best solution is chosen and marked with the pheromone over a number of iterations. Therefore it will be less dependent on the initial parameters such as randomly chosen initial cluster centers and more stabilized while it is more likely to find the global solution rather than the local. To demonstrate the effectiveness and speed of HSCLUST and hybrid algorithms, we have applied these algorithms on various standard datasets and got very good results compared to the K-means. The evaluation of the experimental results shows considerable improvements and robustness of hybridized algorithm.

The paper is organized as follows. Section 2 provides a brief review of vector space model for document representation, particularly the aspects necessary to understand document clustering, quality measures that will be used as the basis for our comparison of techniques. Section 3 provides HS-based clustering algorithm named HSCLUST. The hybrid algorithms are explained in section 4. Section 5presents document sets used in our experiments and give the performance evaluation of the proposed algorithms compared to K-means algorithm. Section 6 concludes the paper.

2. Preliminaries

2.1. Document representation

Vector-space model is widely used in document clustering. In the Vector-space model, each document is represented by a vector of weights of n "features" : $d_i = (w_{i1}, w_{i2}, ..., w_{ij}, ..., w_{in})$

Where the weight w_{ij} is the frequency of feature j in document i and n is number of features. The most widely used weighting schema is the combination of

Term Frequency and Inverse Document Frequency (TF-IDF) [18], which is defined in (1):

$$w_{ij} = TFIDF\,(i, j) = tf\,(i, j).(\log \frac{N}{df\,(j)}) \quad (1)$$

Where $tf(i, j)$ the frequency of feature j in a document d_i, N is the number of documents in the whole collection, and $df\,(j)$ is the number of documents where feature j appears.

2.2. Evaluation of cluster quality

There are two kinds of evaluation measure: internal quality measure and external quality measure. An internal quality measure compares different sets of clusters without references to external knowledge; an external quality measure evaluates how well the clustering is working by comparing the groups produced by clustering techniques to known classes. If one clustering algorithm performs better than other clustering algorithms on many of these measures, then we can have some confidence that is truly the best clustering algorithm for the situation being evaluated.

F measure[12] combines the precision and recall ideas with equal weight on each from and evaluates whether the clustering can remove the noise pages and generate clusters with high quality. Precision and recall compare each cluster i with each class j in the classification, precision and recall are obtained by (2):

$$p_{ij} = \frac{n_{ij}}{n_i} \qquad r_{ij} = \frac{n_{ij}}{n_j} \quad (2)$$

Where n_{ij} is the number of members of class j in cluster i, n_i is the number of members of cluster i and n_j is the number of members in class j. The F measure for a cluster i and class j is then given by (3):

$$F\,(i, j) = \frac{2(P(i, j).R(i, j))}{(P(i, j) + R(i, j)} \quad (3)$$

The F-measure of the whole clustering is:

$$F = \sum_j \frac{n_j}{n} \max\{F(i, j)\} \quad (4)$$

3. HSCLUST: Harmony Search clustering algorithm

In order to cluster documents using harmony search algorithm, we must first model the clustering problem as an optimization problem that locates the optimal centroids of the clusters rather than to find an optimal partition. This model offers us a chance to apply HS optimization algorithm on the optimal clustering of a collection of documents. The following subsections describe harmony operators' accord to clustering.

3.1. Representation of solutions

Let $\{d_i\ , i = 1,2,..., \ n\}$ be the set of documents. Let d_{ij} denote the weight of jth feature of document d_i. Also, define a_{ij} for $i = 1,2,...,K$ and $j = 1,2,\cdots,n$,

$$a_{ij} = \begin{cases} 1, \text{ if j}^{\text{th}} \text{ document belongs to i}^{\text{th}} \text{ cluster,} \\ 0, \text{ otherwise.} \end{cases} \quad (5)$$

Then, the assignment matrix A= $[a_{ij}]$ has the properties that each $a_{ij} \in \{0,1\}$ and each document must assigned exactly to one cluster (e.g. $\sum_{i=1}^{K} a_{ij} = 1$ for $j = 1,2,...,n$). An assignment that represents K nonempty clusters is a legal assignment. Each assignment matrix corresponds to a set of K centroids $C = (c_1, c_2, ..., c_i, ..., \ c_k)$. So, the search space is the space of all A matrices that satisfy constraint in which each document must be allocated to exactly one cluster and there is no cluster that is empty. A natural way of encoding such A into a string, s, is to consider each row of HM of length n and allow each element to take the values from $\{1, 2, \cdots, K\}$. In this encoding, each element corresponds to a document and its value represents the cluster number to which the corresponding document belongs.

3.2. Initialization of harmony memory

Harmony memory must be initialized with randomly generated feasible solutions. Each row of harmony memory corresponds to a specific clustering of documents in which, the value of the ith element in each row is randomly selected from the uniform distribution over the set $\{1, 2, \cdots, K\}$ and indicates the cluster number of ith document. Such randomly generated solutions may not be legal in which no document is allocated to some clusters. This is avoided by assigning $\left\lfloor \frac{n}{K} \right\rfloor$ randomly chosen documents to each cluster and the rest of documents to randomly chosen clusters.

3.3. Improvise a new clustering

In improvising step, we need a technique which generates one solution vector, NHV, from all HMS solution vectors exists in HM .The new generated harmony vector must inherit as much information as possible from the solution vectors that are in the HM. If the generated vector, which is corresponds to a new clustering, consists mostly or entirely of assignments found in the vectors in HM, it provides good heritability.

The cluster number of each document in the new solution vector is selected from harmony memory with probability HMCR and with probability (1– HMCR) is randomly selected from set $\{1, 2, \ldots, K\}$. After generating the new solution, the PAR process is applied. PAR is originally the rate of allocating a different cluster to a document. To apply pitch adjusting process to document d_i the algorithm proceeds as follow. The current cluster of d_i is replaced with a new cluster chosen randomly from the following distribution:

$$p_j = \Pr\{cluster\ j\ \ is\ selected\ as\ new\ cluster\} = \frac{D_{\max} - D(NHV\ ,c_j)}{\sum_{j=1}^{K}(D_{\max} - D(NHV\ ,c_i))}$$

(6)

Where $D_{\max} = \max_i\{D(NHV, c_i)\}$. This novel PAR process exempts the algorithm from the bw parameter.

3.4. Evaluation of solutions

Each row in HM corresponds to a clustering with assignment matrix A. Let $C = (c_1, c_2, \ldots, c_i, \ldots, c_k)$ is set of K centroids for assignment matrix A. The centroid of the kth cluster is $c_k = (c_{k1}, c_{k2}, \ldots, c_{kt})$ and is computed as follows:

$$c_{kj} = \frac{\sum_{i=1}^{n}(a_{ki})d_{ij}}{\sum_{i=1}^{n} a_{ki}} \tag{7}$$

Our objective function is to discover the proper centroids of clusters for maximize intra-cluster similarity (minimizing the intra-cluster distance) as well as minimizing the inter-cluster similarity (maximizing the distance between clusters). Fitness value of each row, which corresponds to one potential solution, is determined by average distance of documents to the cluster centroid (ADDC) represented by that row. This value is measured by equation (8):

$$f = \frac{\sum_{i=1}^{k}\{\frac{\sum_{j=1}^{n_i} D(c_i, d_{ij})}{n_i}\}}{K} \tag{8}$$

where K is the number of clusters, n_i is the numbers of documents in cluster i (e.g. $n_i = \sum_{j=1}^{n} a_{ij}$), D is distance function, and d_{ij} is the jth document of cluster i.

The new generated solution is replaced with a row in harmony memory, if the locally optimized vector has better fitness value than those in HM.

4. The hybrid algorithms

The algorithm with the above processes performs a globalize searching for solutions, whereas K-means clustering procedure performs a localized searching. In localized searching, the solution obtained is usually located in the proximity 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 each iteration. The refining process of the K-means algorithm indicates that the algorithm only explores the very narrow proximity, surrounding the initial randomly generated centroids and its final solution depends on these initially selected centroids. So the proposed algorithm is good at finding promising areas of the search space, but not as good as K-means at fine-tuning within those areas, so it may take more time to converge. On the other hand, K-means algorithm is good at fine-tuning, but lack a global perspective. It seems a hybrid algorithm that combines two ideas can results in an algorithm that can outperform either one individually. To improve the algorithm, we propose tree different versions of the hybrid clustering, depending on the stage when we carry out the K-means algorithm.

4.1. The sequential hybridization

In this section, we present a hybrid HS approach that uses K-means algorithm to replace the refining stage in the HSCLUST algorithm. Hybrid algorithm combines the power of the HSCLUST with the speed of a K-means algorithm. In the hybrid HS algorithm, the algorithm includes two modules, the HSCLUST module and the K-means module. The HSCLUST finds the region of the optimum, and then the K-means takes over to find the optimum centroids.

We need to find the right balance between local exploitation and global exploration. The global searching stage and local refine stage are accomplished by those two modules, respectively. In the initial stage, the HSCLUST 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 HSCLUST 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. This algorithm is shown in following pseudo-code.

Algorithm 1: Sequential Hybridization Algorithm for Document Clustering

```
1. /*producing initial clusters*/
   run the HSCLUST process for the maximum
   number of iterations
2. select the best vector from the HM with highest
   fitness
3. calculate cluster centroids using Eq. (7) and set
   as the initial centroid
   vectors of K-means
4. /*refining the clustering*/
   run K-means process until maximum number of
   iterations is reached
5. set A[i][j] to one if the document $d_i$ is assigned
   to cluster j
6. return A
```

4.2. The Interleaved Hybridization

In this hybrid HS algorithm the local method is integrated into the HSCLUST. For instance, after every K iterations, the K-means uses the best vector from the harmony memory (HM) as its starting point. HM is updated if the locally optimized vectors have better fitness value than those in HM and this procedure repeated until stop condition. The whole process of this hybridization is summarized below.

Algorithm 2: Interleaved Hybridization Algorithm for Document Clustering

```
1. while (average change in fitness greater than the
   given threshold) do
   2. run the HSCLUST process for the
      maximum  number of iterations
   3. select the best vector from the HM with
      highest    fitness
   4. calculate cluster centroids using Eq. (7) and
      set    as the initial centroid  vectors of K- means
   5. run K-means process until maximum
      number of  iterations is reached
      6. If the result of K-means has better fitness
         than those in HM then
         7. replace it with a worse solution in
            harmony memory
      8. endif
9. done
10. report best row in HM as cluster centroids
```

4.3. Hybridizing *K*-means as one step of HSCLUST

To improve the algorithm a one-step K-means algorithm is introduced. After that a new clustering solution is generated with applying harmony operations, the following process is applied on new solution. Each run of the algorithm is summarized in algorithm 3.

Algorithm 3: Each step of HS+*K*-means Algorithm for Document Clustering

```
1. Improvise a new solution
2. calculate cluster centroids using Eq. (7) for
   the new solution
3. Use K-means to reassign each document to
   the cluster with the nearest
   centroid
4. If the result of K-means has better fitness
   than those in HM then
    5. replace it with a worse solution in
       harmony memory
6.  endif
7.  done
```

5. Experimental results and analysis

We compare the algorithms according to their quality and speed of convergence using a number of different document sets. In this section the datasets is described and proposed HS based algorithms are compared with K-means algorithm considering speed of convergence and quality of clustering.

5.1. Document collections

We used five different dataset to compare the performance of the K-means and proposed HS base algorithms in our experiments. Description of the test datasets is given in Table 1.

5.2. Experimental setup

In next step, the K-means, HSCLUST and hybrid algorithms are applied on the above mentioned data sets, respectively. The cosine correlation measure is used as the similarity metrics in each algorithm. It is to be emphasized at this point that the results shown in the rest of paper is the average over 20 runs of the algorithms (to make a fair comparison). Also, for an easy comparison, the algorithms run 1000 iterations in each run since the 1000 generations are enough to convergence of algorithms. No parameter needs to be set up for the K-means algorithm. For HSCLUST, for each data set the HMS is set 2 times the number of cluster in the data set, HMCRis set to 0.9, PAR $_{min}$ = 0.09 , and PAR $_{max}$ = 0.99 .

Table 1: Summary description of document sets

Document Set	Source	#of documents	#of clusters
DS1	Politics	176	6
DS2	TREC	873	8
DS3	DMOZ	697	14
DS4	NEWSGROUP	9249	10
DS5	WebAce	1560	20

5.3. Results and discussions

We compare the algorithms according to their quality and speed of convergence. For evaluation of the clustering results quality, we use two metrics, namely F-Measure and ADDC. F-Measure expresses the clustering results from an external export view, while ADDC examines how much the clustering satisfies the optimization constraints. The smaller the ADDC value,

the more compact the clustering solution is. Table 2 demonstrates the normalized ADDC of algorithms for cosine similarity measures applied on mentioned document sets. Looking at the Table 2, we can see that the results obtained by Hybrid HS+K-means algorithm are significantly comparable by results obtained by K-means. In order to make a better evaluation of clustering, as a primary measure of quality, we used the widely adopted F-Measure [12]; the harmonic means of precision and recall.

The performances of the algorithms in the document collections considering F-Measure are shown in Fig.1. In comparison, the results for different algorithms, obviously, Hybrid HS+K- means has the best F-Measure among the other algorithms from Fig.1.

Table 2. Comparison of the cosine similarity measures of algorithms considering ADDC values of generated clusters

Document Set	*K*-means	HSCLUST	Sequential Hybridization	Interleaved Hybridization	Hybrid HS+K-means
DS1	0.769091	0.67132	0.651100	0.58135	0.51019
DS2	0.72653	0.67123	0.62010	0.5745	0.49234
DS3	0.4821	0.40192	0.3619	0.32473	0.31400
DS4	0.92046	0.8843	0.83125	0.79120	0.75423
DS5	0.87653	0.82100	0.8101	0.78420	0.76125

This issue is due to the high quality of produced clusters by this algorithm. HSCLUST outperforms K-means algorithm in all of datasets and the lowest value is for K-means. This is due to the fact that it converges to the nearest local maximum having the values of K centriods. As can be noticed, the accuracy obtained using our proposed algorithm is in all the datasets comparable with that obtained from the other methods investigated. The second criterion for evaluating algorithms is their convergence rate to optimal solution. Fig. 2 illustrates the convergence behaviors of HSCLUST and K-means algorithms on the document dataset DMOZ. It is obvious from Fig. 2 that HSCLUST took more time to reach the optimal solution and K-means converges more quickly. This is because the K-means algorithm can be trapped in local optima. Although the K-means algorithm is more efficient than HSCLUST considering execution time, the HSCLUST generates much better clustering than the K-means algorithm. In Fig.3 performance of HSCLUST and hybridized algorithms are compared on document dataset DMOZ. Fig. 3 illustrates that the reduction of ADDC value in HSCLUST follows a smooth curve from its initial vectors to final optimum solution and has not a sharp move. Another noteworthy point in Fig.3 is that the final value of ADDC for Hybrid HS+K-means is the lowest among other algorithms. The sequence of other algorithms with respect to their ADDC values are: Interleaved Hybridization, Sequential Hybridization and HSCLUST. This issue shows that the cluster produced by Hybrid HS+K-means has best quality and results produced by hybrid algorithms have higher quality than HSCLUST. From Fig. 3 we can infer that hybrid algorithms overcome HSCLUST disadvantage by incorporating a two-step hybrid algorithms. In the first step, the algorithm uses harmony search to get close to optimal solution, but since it does not fine-tune this result, thus the obtained result is passed as the initial vector to K-means algorithm and then, K-means fine tunes that.

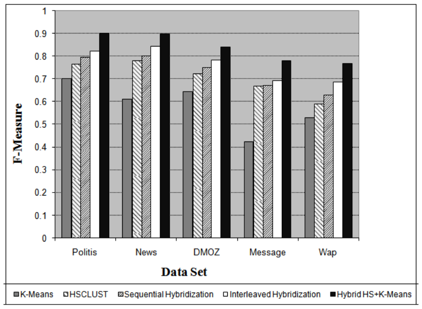

Fig. 1. Comparison of the F-measure for algorithms

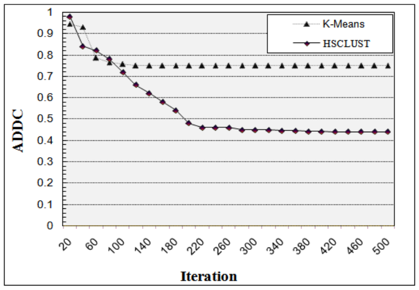

Fig. 2. The convergence behaviors of HSCLUST and *K*-means algorithms on document set DMOZ

7. Conclusion

We considered the problem of finding a globally optimal partition, optimum with respect to ADDC criterion, of a given documents into a specified number of clusters. We proposed some algorithms for this problem. By modeling partitioning problem as an optimization problem, a harmony search based clustering algorithm is proposed. Then the harmony search based algorithm was extended by K-means algorithm through three different hybridization methods. In the experiments, we have used five real life datasets of which characteristics are quite different. We conducted some experiments to test the performance of the hybrid algorithms and compare with the other algorithms. The hybrid algorithms are better than the K-means and the harmony search clustering in terms of the quality of the clustering solutions. From experiments, our methods improve the performance of K-means through calculating F-Measure and ADDC.

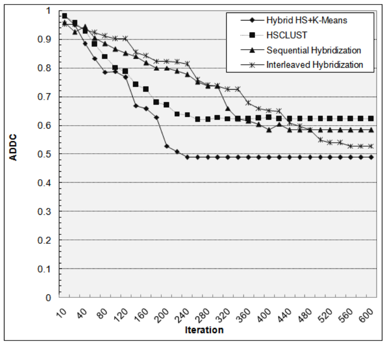

Fig.3. Performance of hybridization algorithms on document set DMOZ

8. References

[1] Jain, A. K., Murty, M. N., and Flynn, P. J. 1999. Data Clustering: A Review. ACM Computing Surveys (CSUR), pp. 264–323.

[2] Grira, N., Crucianu, M., and Boujemaa, N. 2005. Unsupervised and semi-supervised clustering: a brief survey. 7th ACM SIGMM international workshop on Multimedia information retrieval, pp. 9-16.

[3] Zhao, Y., Karypis, G. 2005. Hierarchical clustering algorithms for document datasets”. Data Mining and Knowledge Discovery, 10, 141–168.

[4] Cutting, D.R., Pedersen, J.O., Karger, D.R., and Tukey, J.W. 1992. Scatter/gather: A cluster-based approach to browsing large document collections. In Proceedings of the ACM SIGIR. Copenhagen, pp. 318–329.

[5] Larsen, B. and Aone, C. 1999. Fast and effective text mining using linear-time document clustering. In Proc. of the Fifth ACM SIGKDD Int’l Conference on Knowledge Discovery and Data Mining, pp. 16–22.

[6] Steinbach, M., Karypis, G., and Kumar, V. 2000. A comparison of document clustering techniques. KDD’2000. Technical report of University of Minnesota.

[7] J. McQueen, Some methods for classification and analysis of multivariate observations, proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, (1967), pp. 281-297.

[8] Dhillon, I.S. 2001. Co-clustering documents and words using bipartite spectral graph partitioning. In Knowledge Discovery and Data Mining, pp. 269–274.

[9] Zhao, Y., and Karypis, G. 2004. Empirical and Theoretical Comparisons of Selected Criterion Functions for Document Clustering. Machine Learning, 55 (3).

[10] S. Vaithyanathan, B. Dom, Model selection in unsupervised learning with applications to document clustering, In Proceedings International Conference on Machine Learning, 1999.

[11] K. S. Lee, and Z.W. Geem, A new meta-heuristic algorithm for continues engineering optimization: harmony search theory and practice, Comput. Meth. Appl. Mech. Eng. 194, 2004, pp. 3902–3933.

[12] A. Banerjee, C. Krumpelman, S. Basu, R. Mooney, J. Ghosh, Model based overlapping clustering, In: Proceedings of International Conference on Knowledge Discovery and Data Mining (KDD), 2005.