Shrinking Number of Clusters by Multi-Dimensional Scaling

Tae-Chang Jee, Hyunjin Lee, and Yillbyung Lee

***Abstract*— Clustering is to divide given data and then, automatically find out the meanings hidden in the data. It analyzes data, which are difficult for people to check in detail, and then, makes several clusters consisting of data with similar characteristics. Clustering, which is used in various fields, is automatically done without human interference, but the number of clusters should be decided by men in advance. The number of clusters is a very important element because the result of clustering can be different, depending on the number of clusters. Therefore, this paper proposed a method of deciding the number of clusters, which is projecting the center of a cluster on the two-dimensional plane by use of Multi-Dimensional Scaling, and then, combining the clusters. As a result of experimenting this method with real data, it was found that clustering performance became better.**

***Index Terms*— Document Clustering, Number of Clusters, K-Means, Multidimensional Scaling**

I. Introduction

Along with the development of computer and web, many data are rapidly coming out. Representative examples of it are: as Internet environment is being extensively spread, various web documents are on rapid rise; and as each college encourages its faculty to do research to strengthen its competitiveness, many papers of researches and studies are being increasingly published. In order to search necessary information out of a large quantity of documents, it is necessary to develop efficient analysis methods for these data, the researches are being carried out in the fields of statistics and machine learning of artificial intelligence. This paper handled the technique of clustering out of various data analysis methods. Clustering is to divide given data or objects into clusters and then, automatically finds out significant information hidden in the data. It corresponds to unsupervised learning of machine learning. Document clustering is aimed at helping users' in-depth analysis by laying up the documents with similar characteristics, not by accurately separating all the documents.

Classification needs no more worrisome because the subjects to be classified are clear and the number of classifications is same as that of the subjects to be classified. Clustering allows a user to set up the number of clusters and its result has its own meaning. However, from the viewpoint of general users, it may be felt inconvenient that the result of clustering depends on the number of clusters and that the best optimal number of clusters can be obtained after users test with various numbers of clusters. Accordingly, a method to automatically decide the number of clusters is necessary, but there is still a lack of the researches compared with the researches of the clustering algorithm.

People want to get a result in a very short time in information retrieval system. Besides the clustering performance is superior, it will be inconvenient system if the clustering takes too long time. Shrinking the number of clusters is aim to enhance the clustering performance, but if it takes too long time, it will be meaningless task after all. Therefore, this paper studied the method of automatically deciding the number of clusters without users' repetitive tests and without giving great influence on the clustering time.

This paper is composed of: descriptions of the existing techniques of deciding the number of clusters (chapter 2); explanation of how to reduce the number of clusters (chapter 3); usefulness of the method proposed in chapter 3 through analysis of test results (chapter 4); and suggestions for future studies (chapter 5).

II. Related Works

There have been research efforts that strive to provide the model selection capability to the K-menas methods. Pelleg and Moore (2000) proposed X-means which is an extension of K-means with an added functionality of estimating the number of clusters to generate [14]. The Baysian Information Criterion (BIC) is employed to determine whether to split a cluster or not. The splitting is conducted when the information gain for splitting a cluster is greater than the gain for keeping that cluster.

Liu and Gong (2002) proposed another approach for realizing the model selection capability based on the hypothesis that, if one searches for solutions in an incorrect solution space, result obtained from each run of the document clustering will be quite randomized because the solution does not exist.

Manuscript received October 27, 2006. This work was supported as a WIPS and a Brain neuroinformatics Research program by Korean Ministry of Commerce, Industry, and Energy.

Tae-Chang Jee is with the Department. of Computer Science, Yonsei University, Seoul, Korea (corresponding author to provide phone: +82-19-335-5299; fax: +82-2-6363-3499; e-mail: garura@csai.yonsei.ac.kr).

Hyunjin Lee is with the Department of Computer, Information & Communication, Korea Cyber University, Seoul, Korea (e-mail: hjlee@mail.kcu.ac).

Yillbyung Lee is with the Department. of Computer Science, Yonsei University, Seoul, Korea (e-mail: yblee@csai.yonsei.ac.kr).

Otherwise, results obtained from multiple runs must be very similar assuming that there is only one genuine solution in the solution space. Translating this into the model selection problem, it can be said that, if one guesses on the number of clusters is correct, each run of the document clustering will produce similar sets of document clusters; otherwise, clustering result obtained from each run must be unstable, showing a large disparity [11].

Yu (1998) proposed a method to automatically determining number of clusters by using of BIC [15]. An experiment on EM clustering of 1-d/2-d data are presented and shows a good result. However, BIC measure can't be easily extended to text clustering case.

Salvador (2004) proposed and another algorithm, the L method the finds the "knee" in a '# of clusters vs. clustering evaluation metric' graph [13]. It showed work reasonably well in determining the number of clusters of segments for clustering/segmentation algorithms. But this method is limited to hierarchical algorithm only.

Lu (2005) proposed a new evolutionary algorithm to address estimation the optimal number of clusters [12]. The proposed evolutionary algorithm defines a new entropy-based fitness function, and three new genetic operators for splitting, merging and removing clusters. It can exactly estimate the optimal number of clusters for a set of data.

Boutsinas (2006) presented the z-windows clustering algorithm, which aims to address determining how many clusters are present in a given set of patterns using a windowing techniques [3]. The key idea is to use a sufficiently large number of initial windows, which are properly merged during the algorithm.

III. Proposed Methods

A. System Architecture

The composition of document clustering is presented in Fig. 1 First, the entire document or searched document is needed. Then document-feature vector is created through applying morphological analysis and parsing on the document. The form of the document-feature vector is illustrated in TABLE I. By inputting clustering algorithm to the document-feature vector, clustering is achieved. In this paper, K-Means was used as clustering algorithm. After clustering is achieved through K-Means, information is produced indicating a center point for each cluster and each document's degree of membership to individual clusters. Using this information, a visualization process is carried out.

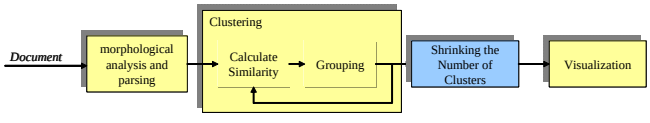

Fig. 1. Document Clustering System Architecture

Applying morphological analysis and parsing on each document, we have document-feature vector like TABLE I, where $D_i\,(i=1...n)$ is the i^{th} document, n is the number of documents, $T_j\,(j=1...m)$ is the j^{th} feature, m is the number of features, and t_{ij} is the number of j^{th} feature in i^{th} document. We restrict the feature to a noun and pronoun only.

TABLE I
EXAMPLE OF DOCUMENTS-FEATURE VECTOR

	T_1	T_2	…	T_m
D_1	t_{11}	t_{12}	…	t_{1m}
D_2	t_{21}	t_{22}	…	t_{2m}
…	…	…	…	…
D_n	t_{n1}	t_{n2}	…	t_{nm}

The 'Shrinking the Number of Centers' in Fig. 1 is the object of this paper. Usually the clustering is ended there is no other job to do and go to Visualization step directly. In this paper, after the clustering is done the shrink the number of centers step is done. Therefore the number of clusters is changed to more reliable numbers.

B. K-Means Clustering

The object of this study document clustering is one of the clustering algorithm applications. Clustering starts to analyze without prior knowledge of data structure so it can be seen as the process of 'data exploration' or 'data excavation'. It clusters a great quantity of data into a small number of homogeneous groups so it can be seen as the process of 'integration of data' or 'simplification of data' through minimum data loss. The result of clustering becomes related with the step of 'formation of hypotheses' as it derives the information on the structural characteristics of a population [1][8][9].

There are various methods to cluster subjects, but the basic premise of all the methods is to maximize similarity among the objects in a cluster and to minimize similarity among the clusters. The methods of clustering are Self Organizing Map(SOM), Complete Linkage, and K-Means, etc. [5].

SOM is unsupervised learning based map. It has good performance, but it can't guarantee that learning of different problems of the same dimension can be completed within the given time. Complete Linkage is hierarchical clustering, which compares distances among documents. It has some advantages that it always converges and the calculating time for different problems of the same dimension is always same, but if the document feature vector becomes big, the total calculating time increases by geometric progression [4]. K-Means is partitional clustering. Calculation can be completed within given time and even though the document feature vector becomes big, the calculating time gradually increases, not by geometric progression like Complete Linkage.

K-Means partitions input data of *n* into clusters of *K*. The standards for similarity are Euclidean Distance, Manhattan's Distance, Pearson Correlation Coefficient, and Cosine Coefficient. *K*, which is the number of clusters, should be set

prior to analysis. In case it is difficult to select *K*, analysis will be done for various *K* values and then, a suitable *K* is decided.

K-Means is performed in the following ways:

Step 1: Initializing a cluster representative value of *K*
Step 2: Assigning each input data to the center of the closest cluster
Step 3: Re-calculating the center of each cluster, using the mean value of data in the cluster
Step 4: Repeating the above step 2 through 3 until there is no change in the cluster which each input data belong to

C. Multidimensional Scaling

It is not easy to view the result of clustering at a time. If original data are made in two or three dimensions, they can be expressed on the plane so we can see the result easily. However, if the dimension of the feature vector is big like document data, it is difficult to express them on the plane. Here, we need to know how to project data of a high dimension on the plane of a low dimension. Multi-Dimensional Scaling is one of the methods [2].

From a non-technical point of view, the purpose of Multi-Dimensional Scaling (MDS) is to provide a visual representation of the pattern of proximities (i.e., similarities or distances) among a set of objects [2]. For example, given a matrix of perceived similarities between various brands of air fresheners, MDS plots the brands on a map such that those brands that are perceived to be very similar to each other are placed near each other on the map, and those brands that are perceived to be very different from each other are placed far away from each other on the map.

From a slightly more technical point of view, what MDS does is find a set of vectors in p-dimensional space such that the matrix of Euclidean distances among them corresponds as closely as possible to some function of the input matrix according to a criterion function called stress.

A simplified view of the algorithm is as follows:

Step 1: Assign points to arbitrary coordinates in p-dimensional space.
Step 2: Compute Euclidean distances among all pairs of points, to form the *Dhat* matrix.
Step 3: Compare the *Dhat* matrix with the input *D* matrix by evaluating the stress function. The smaller the value the greater the correspondence between the two points. The stress function S is as follows:

$$S = \left[\frac{\sum_i \sum_j \left(d_{ij} - f\left(d_{ij}\right) \right)^2}{\sum_i \sum_j d_{ij}^2} \right]^{1/2}$$

Step 4: Adjust coordinates of each point in the direction that best maximally stress.
Step 5: Repeat steps 2 through 4 until stress won't get any lower.

This paper used MDS to project the center of a cluster on the low-dimensional plane. Once clustering is completed, cluster centers are made as many as set in the beginning, and it is possible to calculate the distance between cluster centers. The distance between cluster centers shall be applied to MDS to be projected on the low-dimensional plane. TABLE II is a calculation of the distance (we use Cosine Coefficient) among 12 clusters centers, and Fig. 2 is that the value of Table II is expressed on the low-dimensional plane by use of MDS.

TABLE II
DISTANCES OF CENTERS OF CLUSTERS

	0	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***11***
0		0.47	0.49	0.50	0.52	0.46	0.54	0.52	0.43	0.63	0.57	0.35
1	0.4.		0.57	0.63	0.62	0.56	0.64	0.64	0.53	0.54	0.62	0.48
2	0.49	0.57		0.65	0.63	0.62	0.68	0.67	0.58	0.54	0.62	0.53
3	0.50	0.63	0.65		0.69	0.60	0.65	0.60	0.58	0.57	0.65	0.51
4	0.52	0.62	0.63	0.69		0.65	0.71	0.70	0.63	0.59	0.66	0.57
5	0.46	0.56	0.62	0.60	0.65		0.51	0.59	0.46	0.52	0.58	0.41
6	0.54	0.64	0.68	0.65	0.71	0.51		0.66	0.55	0.60	0.65	0.49
7	0.52	0.64	0.67	0.60	0.70	0.59	0.66		0.57	0.57	0.61	0.54
8	0.43	0.53	0.58	0.58	0.63	0.46	0.55	0.57		0.50	0.57	0.58
9	0.63	0.54	0.54	0.57	0.59	0.52	0.60	0.57	0.50		0.58	0.43
10	0.57	0.62	0.62	0.65	0.66	0.58	0.65	0.61	0.57	0.58		0.51
11	0.35	0.48	0.53	0.51	0.57	0.41	0.49	0.54	0.58	0.43	0.51	

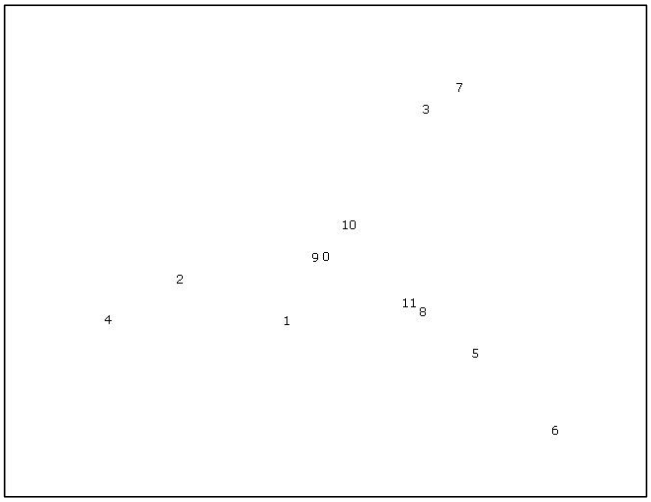

Fig. 2. Centers of Clusters using MDS

D. Shrinking Number of Clusters

In this paper, the method of reducing the number of clusters is based on the low-dimensional geometric structure. Let's look at the cluster centers projected on the low-dimensional plane by use of MDS (see Fig. 2). Some cluster centers are in appropriate distance from other cluster centers, but like 0-9 and 8-11, some cluster centers are closely adjacent to each other(see Fig. 2). The original data of Table I show that the distance among cluster centers are distributed from 0.35 to 0.71. For combining close clusters, the relationship with other clusters should be considered. Therefore, it is impossible to confidence certain of that clusters, which are in the closest distance, can be combined. In addition, different data can make the distance different. In this case, it becomes difficult to determine what clusters (in what distance) should be combined. However, if they are projected on the low-dimensional plane like Fig. 2, the

relationship with other clusters is reflected so it is possible to combine the clusters in the closest distance. And further, the distance is regular so it becomes easy to decide the threshold of the distance for combination.

A simplified view of the algorithm is as follows:

Step 1: Compute threshold T.

$$T = \frac{1}{\left\langle \sqrt{NumC} \right\rangle} \times \alpha \qquad (1)$$

where $NumC$ is the Number of Clusters and $\alpha \in [0,1]$ is the weight to calculate the tolerance of cluster distances.

Step 2: Calculate MDS by use of the algorithm of Chapter 3.2.

Step 3: Use MDS result to calculate distance matrix Dist among clusters.

Step 4: Combine two clusters that belong to the minimum value of Dist if the minimum value of Dist is smaller than T.

Step 5: Repeat steps 2 through 4 until minimum of Dist is greater than T.

The result of applying the above algorithm to Fig. 2 having 12 cluster centers is Fig. 3. It can be confirmed that cluster centers are reduced to 6 clusters under the circumstance that $NumC$ is 12 and α is 0.8.

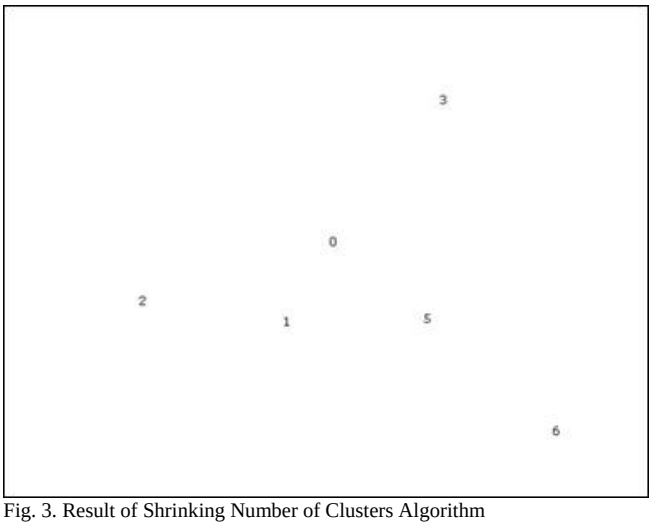

Fig. 3. Result of Shrinking Number of Clusters Algorithm

IV. Experiments and Results

This study implemented at Pentium 4 2.8GHz CPU with C# language. The .Net Framework 1.1 (compiler version) was used.

A. Data Sets

Our experiments adopted a variety of publicly available real-life data sets. These include the Reuter-21578 document collection (Reuters in short)[10], and four databases from the UCI machine learning repository, namely Australian Credit Approval (Australian), Pima Indians Diabetes Database (Diabetes), Heart disease dataset (Heart), Iris Plants Database (Iris)[7]. Table III summarizes the statistics of these five data sets.

TABLE III
Distances of Centers of Clusters

Data Sets	*Num. of instances*	*Num. of features*	*Num. of clusters*
Australian	690	14	2
Diabetes	768	8	2
Heart	270	13	2
Iris	150	4	3
Reuter	2,094	8,031	10

B. Evaluation Metrics

Clustering is unsupervised learning so it can't use the standards for performance evaluation like identification ratio or reliability that are used in supervised learning. Some researches of the methods of evaluating the result of clustering are in progress, one of which is to evaluate with the structural form of clusters [6].

The cluster compactness measure (*Cmp*) evaluates how well the subsets of the input are redistributed by the clustering system, compared with the whole input set. The cluster compactness is based on the variance of a vector data set given by

$$v(X) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} d^2\left(x_i, \bar{x}\right)} \qquad (2)$$

where $d\left(x_i, x_j\right)$ is a distance metric between two vectors x_i and x_j, N is the number of members in X, and $\bar{x} = \frac{1}{N}\sum_i x_i$ is the means of X. A smaller variance value of a data set indicates a higher homogeneity of the vectors in the data set. The cluster compactness for the output clusters generated by a system is then defined as

$$Cmp = \frac{1}{C}\sum_{i}^{C} \frac{v\left(c_i\right)}{v\left(X\right)} \qquad (3)$$

where C is the number of clusters generated on the data set X, $v\left(c_i\right)$ is the variance of the cluster c_i, $v\left(X\right)$ is the variance of the data set X.

The cluster separation measure (*Sep*) evaluates how long the clusters are separated. The cluster separation of a clustering system's output is defined by

$$Sep = \frac{1}{C\left(C-1\right)}\sum_{i=1}^{C}\sum_{j=1, j\neq i}^{C} \exp\left(-\frac{d^2\left(x_{c_i}, x_{c_j}\right)}{2\sigma^2}\right) \qquad (4)$$

where σ is a Gaussian constant, C is the number of clusters,

TABLE IV
THE EXPERIMENTAL RESULTS OF THE FIVE DATA SETS (A = 0.8, B = 0.5)

Data Sets	*Num. of Clusters*	*Cmp*	*Sep*	*Ocq*	Time(msec)
Australia n	Optimal (2)	1.54	0.96	1.25	1,159
	Random (12)	1.63	0.56	1.09	775
	Shrink (3)	***1.46***	***0.22***	***0.84***	1,621
Diabetes	Optimal (2)	1.08	0.39	0.73	896
	Random (10)	1.10	0.33	0.71	131
	Shrink (5)	***1.07***	***0.28***	***0.68***	412
Heart	Optimal (2)	***1.00***	***0.14***	***0.57***	128
	Random (10)	***1.00***	0.15	0.58	31
	Shrink (3)	***1.00***	0.16	0.58	131
Iris	Optimal (3)	***1.02***	0.17	0.60	15
	Random (10)	***1.02***	***0.16***	***0.59***	18
	Shrink (5)	***1.02***	0.17	0.60	59
Reuter	Optimal (10)	1.82	0.87	1.35	6,218
	Random (24)	2.05	0.91	1.48	6,484
	Shrink (5)	***1.76***	***0.58***	***1.17***	11,140

x_{c_i} is the centroid of the cluster c_i, $d\left(x_{c_i}, x_{c_j}\right)$ is the distance between the centroid of c_i and the centroid of c_j.

A smaller cluster compactness indicates a higher average compactness in the output clusters, and a smaller cluster separation indicates a larger overall dissimilarity among the output clusters [6]. Cluster compactness and cluster separation are two standards for comparing performance of clusters so it is difficult to compare the total performance so overall cluster quality (*Ocq*) is defined as:

$$Ocq(\beta) = \beta \cdot Cmp + (1-\beta) \cdot Sep \qquad (5)$$

where $\beta \in [0,1]$ is the weight that balances measures cluster compactness and cluster separation. If this overall cluster quality is small, it means better performance.

C. Clustering Results

Table IV reports the experimental results on five data sets. Here, Optimal is presented as the optimum number of clusters. It follows *Num. Of clusters* of Table III. Random is a randomly selected number and Shrink indicates the random number of clusters that is reduced by the proposed algorithm. Let's see *Cmp*. Better results were made for Australian, Diabetes, and Reuter in case the proposed method was used for reduction than in case the optimal number of clusters or a random was selected, which was same result for Heart and Iris. Let's see *Sep*. Same as *Cmp*, the proposed method showed the better result for Australian, Diabetes, and Reuter except for Heart and Iris. For Heart and Iris, the difference was just 0.02 and 0.01, which is very small compared with others. Because the proposed method showed the similar result in *Cmp* and *Sep*, it showed the better result for the three in terms of *Ocq* while it showed the worse result for the other two, but the difference was very small with 0.01.

The time recorded here doesn't include the learning time of K-Means algorithm. Only the time of the proposed algorithm that is explained in chapter 3.3 is recorded. Optimal and Random are the results of performing only the step 1 and 2 out of the algorithm explained in chapter 3.3. Shrink is the result of performing the steps of 1~5. In Shrink, the steps of 3~5 were added so the time couldn't help being extended more than Random, and the result can be confirmed through Table IV. Average 290% more of time was required. The time for Shrink increased by 170% compared with that for Optimal. The reason why Optimal needed more time than Random lies in the method of getting MDS. Shrink required more time, but in terms of the absolute value, it was just within 1 second (5 seconds for Reuter), which is very small compared with the total clustering time. (In case of Reuter, the total clustering time is about 10 minutes.)

V. CONCLUSION AND FUTURE WORK

This paper proposed the method of automatically determining the number of clusters by use of MDS. Clustering, through analysis of a large quantity of data, helps people find out the characteristics of data that they didn't know. However, because of the limitation of unknown data, it is impossible to know how many clusters are suitable to get accurate results. Therefore, we can't help deciding a random number of clusters. Because of the characteristics of clustering, true or false doesn't exist, but for better performance, clustering had better be performed and for this, this paper proposed the method of automatically deciding the number of clusters. This paper applied the proposed method to real data. The result tells that it showed the same or the better performance. It took more time, but the time is acceptable to be applied to the real system.

The method proposed by this paper reduced the number of clusters by merging geometrically closest clusters. The method is executed without re-clustering process, so there is a benefit in time.

It is possible to add the process which divides the big cluster for enhance the clustering performance. However, adding this process is accompanied with re-clustering process. Therefore, it needs to be studied further to check if it's execution time is acceptable to real-time system.

REFERENCES

[1] Michael J. A. Berry and Gordon S. Linoff, “Data Mining Techniques for Marketing, Sales, and Customer Support”, John Wiley & Sons, 1997.

[2] Ingwer Borg, Patrick J. F. Groenen and Stephen P. Borgatti,, “Modern Multidimensional Scaling”, Springer Verlag, 2005.

[3] B. Boutsinas, D. K. Tasoulis and M. N. Vrahatis, “Estimating the number of clusters using a windowing technique”, Journal of Pattern Recognition an Image Analysis, Issue Volume 16, Number 2, April, 2006, pp. 143-154.

[4] Douglass R. Cutting, David R. Karger, Jan O. Pedersen and John W. Tukey, “Scatter/Gather: a cluster-based approach to browsing large document collections”, In *Proc. of the 15th annual international ACM SIGIR*, June, 1992, pp. 318-329.

[5] Earl Gose, Richard Johnsonbugh and Steve Jost, “Pattern Recognition and Image Analysis”, Prentice Hall, 1996.

[6] J. He, A.H. Tan, C.L. Tan, and S.Y. Sung, “On quantitative evaluation of clustering systems”, In Weili We, Hui Xiong, and Shashi Shekhar, editors, Information Retrieval and Clustering. Kluwer Academic Publishers, 2003.

[7] S. Hettich and S. D. Bay, “The UCI KDD Archive [http://kdd.ics.uci.edu]”, Irvine, CA: University of California, Department of Information and Computer Science, 1999.

[8] Anil K. Jain and Richard C. Dubes, “Algorithms for Clustering Data”, Prentice Hall, 1988.

[9] Leonard Kaufman and Peter J. Rousseuw, “Finding Groups in Data an Introduction to Cluster Analysis”, Wiley Series in Probability and Mathematical Statistics, 1990.

[10] D. D. Lewis, “Reuters-21578 text categorization test collection distribution 1.0”, http://www.research.att.com/～lewis, 1999.

[11] A. Liu and Y. Gong, “Document clustering with cluster refinement and model selection capabilities”, In *Proc. of ACM SIGIR 2002*, Tampere, Finland, Aug, 2002, pp. 191-198.

[12] Wei Lu and I. Traore, “Determining the optimal number of clusters using a new evolutionary algorithm”, In *Proc. Of the 17th IEEE International Conference on Tools with Artificial Intelligence(ICTAI 05)*, Nov. 2005, 2 pp..

[13] S. Salvador and P. Chan, “Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms”, In *Proc. Of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 04)*, Nov. 2004, pp. 576-584.

[14] D. Pelleg and A. Moore. “X-means: Extending k-means with efficient estimation of the number of clusters”, In *Proc. of the Seventeenth International Conference on Machine Learning (ICML2000)*, June, 2000, pp. 727-734.

[15] Hua Yu, “Automatically Determining Number of Clusters”, Information Retrieval (CMU CS11-741) Final Report, Apr. 1998, 5 pp..