An Improved Clustering Algorithm for Text Mining: Multi-Cluster Spherical K-Means

Volkan Tunali[1], Turgay Bilgin[1], and Ali Camurcu[2]
[1]Department of Software Engineering, Maltepe University, Turkey
[2]Department of Computer Engineering, Fatih Sultan Mehmet Waqf University, Turkey

Abstract: *Thanks to advances in information and communication technologies, there is a prominent increase in the amount of information produced specifically in the form of text documents. In order to, effectively deal with this "information explosion" problem and utilize the huge amount of text databases, efficient and scalable tools and techniques are indispensable. In this study, text clustering which is one of the most important techniques of text mining that aims at extracting useful information by processing data in textual form is addressed. An improved variant of spherical K-Means (SKM) algorithm named multi-cluster SKM is developed for clustering high dimensional document collections with high performance and efficiency. Experiments were performed on several document data sets and it is shown that the new algorithm provides significant increase in clustering quality without causing considerable difference in CPU time usage when compared to SKM algorithm.*

Keywords: *Data mining, text mining, document clustering, SKM.*

Received February 10, 2013; accepted March 17, 2014; published online March 8, 2015

1. Introduction

Data mining is the process of extracting previously unknown, potentially useful and valuable patterns and knowledge from large amounts of data [7, 9]. Data mining studies are focused mostly on structured data stored in relational databases and datawarehouses. However, significant amount of data accessible and available is stored in text databases (i.e., document databases) containing very large collections of documents which consist of resources like news articles, research papers, books, e-mail messages, letters, surveys, reports, presentations and web pages [9]. According to an estimation, 85% of business-related information exists in textual form [10].

Text mining is the application of algorithms and techniques from data mining, statistics, machine learning, natural language processing, information retrieval and knowledge management with the goal of finding solutions to so-called "information explosion/ information overload" problem [8]. Text clustering, a field of text mining, provides effective means of processing and organizing large amounts of textual data automatically. It is defined as unsupervised and automatic grouping of a given document collection into clusters according to document similarities, in such a way that documents belonging to the same cluster are as similar to each other as possible, while documents from different clusters are dissimilar. It is expected that documents in one cluster are of the same or very similar and related topic [2, 8, 14].

In this study, a new and original algorithm called multi-cluster Spherical K-Means (SKM) is developed for clustering high dimensional and large document collections with high performance and efficiency. It is based on SKM algorithm that is widely used due to its high performance and scalability in document clustering tasks. We conducted experiments on several document datasets and show that this new algorithm offers significant increase in clustering quality while keeping computational demands as low as of SKM algorithm.

This paper is organized as follows. Section 2 describes the basic concepts in text clustering. In section 3, some background information about SKM algorithm is provided. In section 4, we explain the details of the design, structure, and operation principles of Multi-Cluster SKM algorithm (MCSKM) we develop. Experiments conducted on several document datasets and experimental results are presented in sections 5 and 6. Finally, section 7 provides some concluding remarks and plans for future work.

2. Fundamental Concepts in Text Clustering

2.1. Document Representation Model

Although, it was first designed for indexing and information retrieval purposes [16] Vector Space Model (VSM) has been a widely used model in data and text mining operations. In this model, documents are represented as m-dimensional vectors. Here, m is the number of unique words (generally referred to as terms) in the document collection after pre-processing. Each document is represented with a feature vector: $d=(w_{d1},...,w_{dm})$. Each component of this vector reflects the degree of relationship between its associated term and the respective document, which is called the weight of the term [2, 3, 19].

Binary term representation is the simplest way of representing documents as vectors. In this approach, however, all terms in a document are of equal weight; that is, their effect to the query or similarity comparison is the same. In general, instead of binary term vectors, a representation approach that considers the weights of terms with respect to documents and to whole collection provides better performance. Therefore, terms that appear often in highly related documents but appear rarely in the whole collection are given more weight [19]. This well-known weighting scheme is the Term Frequency-Inverse Document Frequency (TFIDF) scheme where the importance of a term increases proportionally to the number of times the term appears in the document but is offset by the frequency of the term in the whole corpus. TFIDF value of term w_{dt} is usually calculated as in Equation 1.

$$w_{dt}=(1+log(f_{dt}))\times log(1+n/f_t) \tag{1}$$

Here, f_{dt} is the frequency of term t in document d, n is the total number of documents in collection D, f_t is the number of documents that contain term t.

2.2. Similarity Measure

In order to, group data objects meaningfully, a suitable similarity measure must be defined between the objects. Therefore, similarity measure is extremely important for cluster analysis and a good choice of similarity measure is directly related to the performance of clustering.

The cosine measure is one of the most popular document similarity measures due to its sensitivity to document vector and to its performance. The cosine measure is defined as the cosine of the angle between two feature vectors. The cosine similarity between two documents d_i and d_j is calculated as in Equation 2.

$$sim(d_i,d_j)=\cos(d_i,d_j)=\frac{\sum_{k=1}^{m} w_{ik}w_{jk}}{\sqrt{\sum_{k=1}^{m} w_{ik}^2 \cdot \sum_{k=1}^{m} w_{jk}^2}} \tag{2}$$

Where $i, j = 1,..., n$

The larger the cosine value, the larger the similarity between two documents [19].

3. SKM Algorithm

The K-means is a very well-known and commonly used partitioning clustering algorithm. It partitions a document collection D of n documents into k clusters so that a global criterion function is either maximized or minimized [9]. This global clustering criterion J_{SKM} is defined as:

$$J_{SKM}(D,C)=\sum_{j=1}^{k}\sum_{d_i\in C_j} f(d_i,c_j) \tag{3}$$

In Equation 3, C represents the whole clustering, c_j represents the centroid of cluster C_j, for j=1,..., k and $f(d_i,c_j)$ is the similarity function between the document vector d_i and centroid vector c_j. When the cosine similarity measure is used as the similarity function, document d_i is assigned to the cluster c_j represented with the most similar centroid c_j and the global criterion function is maximized as a result.

There are several weighting schemes for weights of terms in the document vector. The well-known one is the TFIDF where the importance of a term increases proportionally to the number of times the term appears in the document but is offset by the frequency of the term in the whole corpus. However, this scheme is vulnerable that long documents are favored over short ones since they contain more terms. For this reason, a normalization factor is used in order to discount the contribution of long documents [19]. One possible normalization factor is computed as in Equation 4.

$$NF(d_i)=\sqrt{\sum_{j=1}^{m} w_{ij}^2} \tag{4}$$

This normalization implies that $||d_i||=1$; that is, each document vector lies on the surface of the unit sphere in R^m. Like document vectors, centroid vectors of the clusters also lie on the surface of the high-dimensional sphere. The variant of the K-means algorithm that uses document vectors of unit length along with cosine similarity measure is called SKM algorithm [6]. An important computational advantage of using vectors of unit length is that calculation of cosine similarity measure becomes simply the dot product of two vectors. That is, the division is eliminated from Equation 2 since, the denominator is always 1.

4. MCSKM Algorathim

SKM algorithm is a partitioning clustering algorithm that performs hard clustering; that is, finally each document is assigned to only one cluster. In the clustering algorithm that we call MCSKM, we modified SKM algorithm in a way that documents are allowed to be assigned to more than one cluster inside the K-means loop. As a result, we obtained significant increase in clustering quality while keeping computational demands as low as of SKM algorithm. Besides clustering quality, MCSKM algorithm satisfies one of the desired properties of document clustering tasks that one document can be assigned to more than one clusters (i.e., overlapping clusters). Assuming that documents belong only to their most similar clusters, MCSKM algorithm can also be utilized as a hard clustering algorithm.

Fuzzy C-Means (FCM) is the most popular soft clustering algorithm where each data object belongs to all clusters with a degree of membership [4]. Original FCM uses Euclidean distance measure and this measure is not suitable for high-dimensional document vectors. Unlike K-means, there is no straightforward way to transform the FCM to make it applicable on document vectors such as simply changing the distance measure with the cosine similarity measure.

In [15] FCM is modified to work with document vectors, and a variant of FCM called Hyperspherical Fuzzy C-Means (H-FCM) algorithm is developed. H-FCM is shown to present better clustering performance in most cases. In some FCM and H-FCM related studies on document clustering, it is shown that computational complexity and requirements of these algorithms are very high to be practical for very large document collections [11, 13].

Underlying idea of MCSKM is very intuitive that documents can be exact (crisp) member with full membership of more than one cluster as long as some membership condition is satisfied. In FCM-based algorithms, however, documents belong to all clusters with a degree of membership. From the perspective of the user, this makes it difficult and unintuitive to grasp the structure of the document collection being explored. Furthermore, working mechanism and implementation of MCSKM is very simple and effective like its ancestor SKM algorithm. In conclusion, we focused on developing a clustering algorithm producing exact clustering with a soft clustering approach instead of a FCM-based approach in this study.

4.1. General Structure

Before explaining the details of the MCSKM algorithm, we need to provide some definitions. In addition to the cluster number *k* parameter of K-means algorithm, there are two new parameters for MCSKM algorithm.

- Max Assignable Clusters-MAC: In the classical SKM loop, documents are assigned to only one cluster they are most similar and then cluster centroids are recalculated. In MCSKM algorithm, however, MAC parameter determines the maximum number of clusters a document is allowed to be assigned to as long as the condition specified by the Similarity Ratio Limit (SRL) parameter is satisfied. MAC value must be greater than 1. If it is equal to 1, then MCSKM algorithm is equivalent to SKM algorithm. Clustering quality of MCSKM varies according to the MAC value and based on experimental results, we can make a suggestion for MAC value, depending on the *k* parameter, as in Equation 5.

$$\frac{k}{4} \leq MAC \leq \frac{k}{2} \tag{5}$$

- SRL: Let the most similar cluster of any document be C_{best}. SRL is the parameter that determines how close the similarity between a cluster (other than C_{best}) and a document must be to the similarity between C_{best} and the document so that the document can be assigned to that cluster (up to the number of clusters allowed by MAC parameter). For example, document *d* is most similar to C_1 with a similarity value Sim_1=$Sim(d, C_1)$=0.65 and the second cluster that *d* is most similar is C_3 with a similarity value Sim_2=$Sim(d, C_3)$=0.58. Let SRL=0.15.

$$SimRatio = \frac{Sim_1}{Sim_2} - 1 = \frac{0.65}{0.58} - 1 = 0.12 \tag{6}$$

When we subtract 1 from the ratio of similarities, we get 0.12 value. If this value is less than the SRL parameter, then document *d* is allowed to be assigned to the cluster C_3. In the example above, because SRL=0.15, *d* can be assigned to C_3. The same test is applied to the other clusters that document *d* is similar in the order of similarity from highest to lowest. Document *d* is assigned to all clusters satisfying the SRL condition, limited to the number of clusters determined by MAC.

The lower the SRL parameter, the lower the probability documents are assigned to the other clusters. The higher the SRL parameter, the higher the probability documents are assigned to the other clusters. Usually dependent on the characteristics of the dataset clustered, based on experimental results, we can make a suggestion for SRL value as in Equation 7.

$$0.1 \leq SRL \leq 0.2 \tag{7}$$

4.2. Algorithm and Clustering Criterion Function

Steps of MCSKM algorithm are as follows:

1. Arbitrarily choose *k* documents from *D* as the initial cluster centroids.
2. For each document *d*.
   a. Calculate similarities between *d* and *k* clusters.
   b. Sort similarities highest to lowest.
   c. Assign document *d* to cluster with the highest similarity (C_{best}).
   d. For the next MAC-1 clusters, test the SRL condition and assign document *d* to eligible clusters.
3. Recalculate *k* centroids based on the documents assigned to them.
4. Repeat steps 2 and 3 until convergence (i.e., no change in the value of the global criterion function).

Although, MCSKM algorithm employs a soft clustering approach, it tries to maximize the global clustering criterion function used in SKM algorithm with a hard clustering approach. In other words, even if a document is assigned to more than one cluster, only the similarity to the most similar cluster to that document (C_{best}) is used in the global clustering criterion function. Therefore, fundamental optimization approach remains the same and the global clustering criterion function J_{MCSKM} is defined as:

$$J_{MCSKM}(D,C) = \sum_{i=1}^{n} f(d_i, c_{d_i}^{best}) \tag{8}$$

In Equation 8, C represents the whole clustering, $c_{d_i}^{best}$ represents the centroid of the cluster that document d_i is assigned to with highest similarity. $f(d_i, c_{d_i}^{best})$ is the similarity function between document d_i and centroid vector $c_{d_i}^{best}$. When MAC=1, MCSKM algorithm is equivalent to SKM and similarly, clustering criterion function J_{MCSKM} is equivalent to clustering criterion function J_{SKM}.

4.3. Operation Principle

K-means algorithm is an iterative optimization algorithm and its clustering results are extremely dependent on the initial conditions. The better the initial clustering the algorithm starts with, the better the local maximum point is reached. Moreover, the result of an iteration becomes the initial condition of the following iteration. We observe that MCSKM algorithm has a positive effect to improve the initial conditions of each K-means iteration, thus forcing the algorithm towards a higher local maximum point in terms of optimization.

Operation of MCSKM algorithm in comparison with SKM algorithm is shown in Figure 1. In the figure, bold arrows show which clusters the documents are assigned to and the figures over the arrows show the similarity values. Dashed arrows point to the second most similar clusters of the documents.

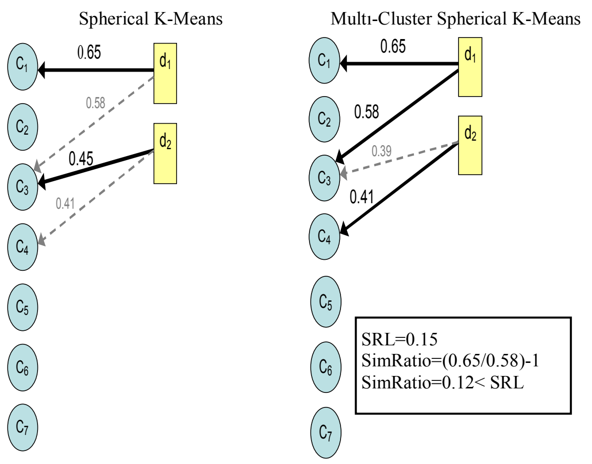

Figure 1. Operation of MCSKM algorithm in comparison with SKM.

Considering the example in Figure 1, in classical SKM, document d_1 is assigned to cluster C_1 and document d_2 is assigned to cluster C_3. In MCSKM, however, document d_1 is assigned to C_1 and also to C_3 within the SRL condition. Therefore, while document d_2 is expected to be assigned to C_3, it is assigned to C_4 because the existence of d_1 in C_3 makes C_3 unsuitable for d_2; that is, document d_2 is more similar to C_4 than C_3. Here, assignment of d_1 to C_3 forces d_2 to be assigned to C_4 which is a better choice for the next K-means iteration. When this approach is applied to all documents, documents are assigned to more relevant clusters at each K-means iteration, thus resulting in a better optimization and higher clustering quality with respect to SKM.

In order to, use MCSKM algorithm, the k parameter must be at least 5 because if there are less than 5 clusters, assigning documents to 2 clusters causes other document to incorrectly be assigned to inappropriate remaining clusters. This produces extremely poor clustering.

4.4. Scalability Analysis

SKM algorithm is one of the most preferred clustering algorithms for document clustering tasks due to its low computational complexity and high scalability. The most time consuming fundamental operation in SKM algorithm is the cosine similarity calculation between documents and cluster centroids. Assume that unit operation time for calculating cosine similarity is F, unit operation time for updating cluster centroids is G, number of documents in the collection is n, number of clusters is k and iteration count of K-means loop is L. Therefore, essentially the following computations are performed for each and every K-means iteration:

- $n.k$ times cosine similarity calculation.
- k times cluster centroid update.

As a result, time complexity of SKM can be expressed as:

$$T_{SKM} = (F \cdot n \cdot k + G \cdot k) \cdot L \tag{9}$$

Since, $n>>k$ and $F>>G$, overall time complexity of SKM algorithm is found $O(n)$, which means that computational complexity is proportional to number of documents and SKM is very scalable in terms of number of documents to be clustered.

MCSKM algorithm adds one extra step to the basic steps in SKM algorithm. After the similarity calculation of each document to all cluster centroids, those k similarity values are sorted from highest to lowest. Therefore, for each K-means iteration, the following computational overhead is added:

- n times sorting of k elements.

Computational complexity of sorting algorithms that are usually preferred for sorting small number of elements (like Selection Sort or Insertion Sort) is $O(n^2)$ and assuming that unit comparison operation in sorting requires H unit time, time complexity of MCSKM algorithm can be expressed as:

$$T_{MCSKM} = (F \cdot n \cdot k + G \cdot k + H \cdot n \cdot k^2) \cdot L \tag{10}$$

Since, $F>> H$ and $n>>k$, overhead of sorting operation is so small when compared to the similarity computations that it is negligible. Consequently, overall time complexity of MCSKM algorithm is found $O(n)$, meaning that its computational demand is linear to the number of documents like SKM algorithm and MCSKM algorithm is also, highly scalable.

5. Experiments

In order to, evaluate the performance of MCSKM algorithm on clustering high dimensional document collections, we conducted experiments on common benchmark document datasets.

5.1. Software and Hardware Platform

We compared MCSKM algorithm with SKM and Bisecting K-Means (BKM) algorithms. We implemented MCSKM and SKM algorithms in C++ programming language. We used Clustering Toolkit (CLUTO) software for BKM. Cluto is written in ANSI C by Karypis and available as a standalone application at the web site of its author [5, 17].

Since, SKM and MCSKM algorithms are very dependent on initial conditions, we ran both algorithms 10 times with 10 different random initial conditions and then averaged the results in order to minimize the effect of their dependency on initial conditions for fair comparison. CLUTO does not provide an option for specifying initial conditions and it produces the same result everytime it runs, so we did not use this approach for BKM algorithm.

In order to, observe the net overhead introduced by MCSKM algorithm over the baseline SKM algorithm experimentally, we measured the unit iteration time of the K-means loops inside both algorithms instead of total run times of the algorithms. Initially, how many iterations it will take for K-means based algorithms to converge a local optimum point is not known. Therefore, it is improper to compare the total run times of these algorithms. Then, we compared the average run times of K-means iterations instead.

All experiments were performed on Windows Vista Business platform installed on a PC with single core Mobile AMD Sempron 3600+ 2 GHz CPU and 2 GB main memory.

5.2. Datasets

We used 3 different document datasets in order to compare the clustering performances of MCSKM, SKM and BKM algorithms. Two of them are very common benchmark datasets used in text mining and information retrieval research very often, and are composed of documents in English. The third one is a dataset consisting of Turkish documents.

20 News Groups (20NG) is a well-known test dataset frequently used in text mining research. It consists of about 20000 documents compiled from 20 Usenet groups as 1000 messages from each group [1].

Wap dataset is composed of 1560 web pages from 20 different categories, collected from Yahoo! topic hierarchy within the WebACE project [18]. Categories that form this dataset are highly skewed that there are 5 documents in the smallest category, 341 documents in the largest category and average category size is 78 documents. Additionally, most of the categories of this dataset are very close to each other.

Milliyet dataset consists of 1455 Turkish news articles published on the web site of Milliyet newspaper in [12]. It contains articles from 3 different categories and there are 485 documents in each category. Categories of this dataset are economics, politics and sport.

Properties of the datasets used in the experiments are shown in Table 1.

Table 1. Properties of the datasets used in the experiments.

Dataset	Document Count	Category Count	Smallest Category	Largest Category	Category Average	Dimension
20NG	19997	20	1000	1000	1000	34905
Wap	1560	20	5	341	78	7977
Milliyet	1455	3	485	485	485	7609

5.3. Clustering Validity Measures

In the experimental results, we used Purity, Entropy, and Normalized Mutual Information (NMI) as clustering quality measures.

Suppose there are *c* categories (or classes) and *k* clusters. Let n_l be the number of objects in cluster C_l, and $n_l^{(h)}$ be the the number of objects in cluster C_l that belong the class K_h, where h=1,…, *c*.

Purity is the ratio of the dominant class size in the cluster to the cluster size itself. A high purity implies that the cluster is a pure subset of the dominant class. Purity of cluster C_l is defined as:

$$purity\,(C_l) = \frac{1}{n_l}\max_h(n_l^{(h)}) \tag{11}$$

Purity of the entire collection of clusters is evaluated as a weighted sum of the individual cluster purities and is defined as:

$$purity\,(C) = \sum_{l=1}^{k}\frac{n_l}{n}\,purity\,(C_l) = \frac{1}{n}\sum_{l=1}^{k}\max_h(n_l^{(h)}) \tag{12}$$

Entropy is a more comprehensive measure than purity. It considers the distribution of classes in a cluster. Note that, we use normalized entropy which takes values between 0 and 1. An entropy value of 0 means the cluster is comprised entirely of one class, while an entropy value close to 1 is considered bad because it implies that the cluster contains a uniform mixture of all classes. Entropy of cluster C_l is defined as:

$$entropy\,(C_l) = -\frac{1}{\log(c)}\sum_{h=1}^{c}\frac{n_l^{(h)}}{n_l}\log(\frac{n_l^{(h)}}{n_l}) \tag{13}$$

Entropy of the entire collection of clusters is evaluated as a weighted sum of the individual cluster entropies and is defined as:

$$entropy\,(C) = \sum_{l=1}^{k}\frac{n_l}{n}\,entropy\,(C_l) \tag{14}$$

NMI is an information-based clustering validity measure. A high NMI value implies that clustering and true class label match well. NMI is defined as:

$$NMI(C) = \frac{2}{n}\sum_{l=1}^{k}\sum_{h=1}^{c} n_l^{(h)} \log_{k\cdot c}\left(\frac{n_l^{(h)} n}{n^{(h)} n_l}\right) \qquad (15)$$

6. Experimental Results

In this section we present the experimental results and our evaluations of the results.

6.1. 20NG Dataset

20NG dataset contains 20 natural clusters (categories). Goal of the experiments is to identify these 20 clusters with the highest accuracy. Clustering quality results and unit iteration time of MCSKM algorithm in comparison with SKM and BKM algorithms for different MAC values are shown in Table 2. Experiments with different SRL values within the suggested range in Equation 7 produce very close results, so we report only the results with SRL values with which best results are obtained. Table 2 presents the results for SRL=0.1. Graph of these results is displayed in Figure 2.

A prominent increase in the clustering quality is obtained with MCSKM algorithm for all MAC values with respect to SKM algorithm. In addition, MCSKM algorithm provides better clustering quality than BKM algorithm according to the validity measures except purity. Best clustering result is achieved when MAC=7, which satisfies the MAC condition in Equation 5.

Table 2. Clustering results for 20NG dataset.

Algorithm		Purity	Entropy	NMI	Unit Iteration Time (sec)
BKM		0.768	0.249	0.761	-
SKM		0.660	0.254	0.770	1.138
MCSKM	MAC=2	0.709	0.221	0.797	1.225
	MAC=3	0.758	0.194	0.820	1.228
	MAC=4	0.743	0.203	0.811	1.233
	MAC=5	0.753	0.197	0.818	1.239
	MAC=6	0.740	0.202	0.814	1.243
	MAC=7	**0.760**	**0.190**	**0.825**	**1.244**
	MAC=8	0.739	0.201	0.818	1.248
	MAC=9	0.746	0.196	0.822	1.254
	MAC=10	0.728	0.206	0.813	1.264
	MAC=11	0.748	0.195	0.824	1.319
	MAC=12	0.719	0.213	0.813	1.300

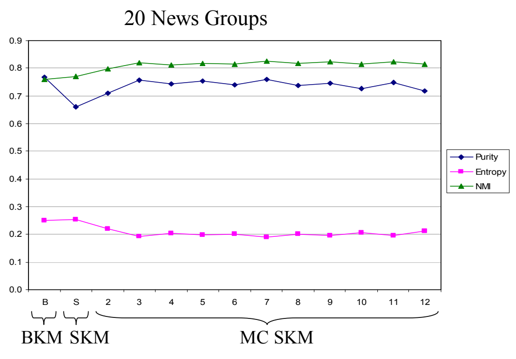

Figure 2. Graph of clustering results for 20NG dataset.

6.2. Wap Dataset

There are 20 natural clusters in Wap dataset. Experiments are conducted to find these 20 clusters with the highest accuracy. Clustering quality results and unit iteration time of MCSKM algorithm in comparison with SKM and BKM algorithms for different MAC values and for SRL=0.175 are shown in Table 3. Graph of these results is shown in Figure 3.

Table 3. Clustering results for wap dataset.

Algorithm		Purity	Entropy	NMI	Unit Iteration Time (sec)
BKM		0.664	0.337	0.576	-
SKM		0.590	0.415	0.495	0.095
MCSKM	MAC=2	0.646	0.363	0.549	0.102
	MAC=3	0.665	0.344	0.566	0.104
	MAC=4	0.671	0.340	0.572	0.104
	MAC=5	0.680	0.331	0.583	0.105
	MAC=6	**0.680**	**0.330**	**0.591**	**0.105**
	MAC=7	0.679	0.332	0.584	0.106
	MAC=8	0.673	0.334	0.586	0.106
	MAC=9	0.670	0.334	0.586	0.107
	MAC=10	0.674	0.332	0.591	0.107
	MAC=11	0.678	0.330	0.590	0.107
	MAC=12	0.679	0.331	0.593	0.107

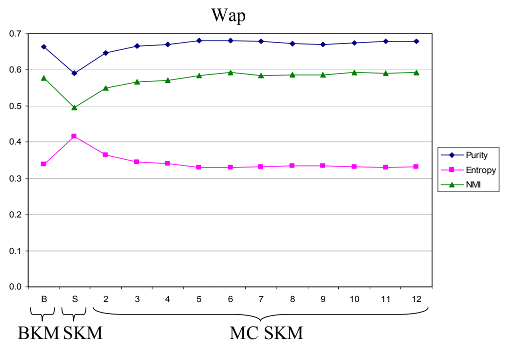

Figure 3. Graph of clustering results for wap dataset.

MCSKM algorithm clearly outperforms SKM algorithm as well as BKM algorithm for all MAC values in terms of clustering quality. Best clustering result is achieved when MAC=6, which satisfies the MAC condition in Equation 5.

6.3. Milliyet Dataset

There are 3 natural clusters in Milliyet dataset which consists of documents in Turkish language. In order for MCSKM algorithm to be applicable, cluster count parameter k must be greater than or equal to 5 ($k \geq 5$). Therefore, experiments on this dataset were conducted for k=9 and clusters were expected to match the known 3 categories with the highest accuracy possible. Clustering quality results and unit iteration time of MCSKM algorithm in comparison with SKM and BKM algorithms for different MAC values and for SRL=0.15 are shown in Table 4. Graph of these results is given in Figure 4.

Table 4. Clustering results for Milliyet dataset.

Algorithm		Purity	Entropy	NMI	Unit Iteration Time (sec)
BKM		1.000	0.000	0.699	-
SKM		0.992	0.018	0.729	0.046
MCSKM	MAC=2	0.994	0.012	0.693	0.052
	MAC=3	0.999	0.003	0.740	0.057
	MAC=4	0.998	0.004	0.822	0.059
	MAC=5	1.000	0.001	0.867	0.062
	MAC=6	**1.000**	**0.000**	**0.928**	**0.062**
	MAC=7	0.952	0.080	0.853	0.063

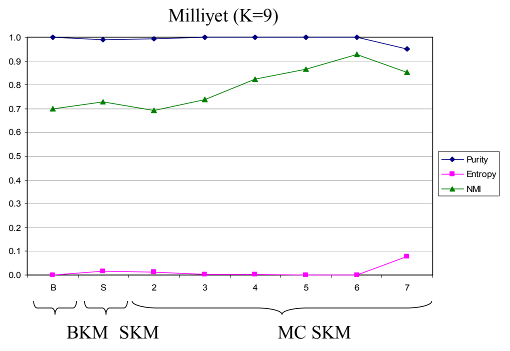

Figure 4. Graph of clustering results for Milliyet dataset.

In terms of clustering quality, MCSKM algorithm exhibits higher performance than SKM and BKM algorithms for all MAC values. Best clustering result is achieved when MAC=6, which is not within the suggested range of MAC values in Equation 5. However, results obtained for MAC values within our suggested range are also superior to the results of SKM and even BKM.

6.4. Evaluation of Results

We show that MCSKM algorithm provides significant increase in clustering quality with respect to SKM algorithm that MCSKM is based on according to the external validity measures of purity, entropy and NMI. Additionally, results obtained are comparable to and even better than the results of BKM algorithm which is well known to be very effective in text clustering and is the most referred algorithm for comparison purposes.

Overhead of CPU time usage that MCSKM algorithm imposes over SKM algorithm is also shown in the experimental results. Accordingly, considering the higher clustering quality, the increase in the unit iteration time caused by MCSKM algorithm remains within acceptable limits. Therefore, MCSKM algorithm with computational complexity of *O(n)* according to the scalability analysis is applicable for clustering very large document collections.

7. Conclusions and Future Work

In this study, we applied the idea of overlapping clusters in soft clustering approaches to SKM algorithm that performs hard clustering. We developed the MCSKM algorithm by altering SKM algorithm in a way that documents are allowed to be assigned to more than one cluster inside the K-means loop. Conducting experiments on several document datasets, we show that our new algorithm provides significant increase in clustering quality with very slight difference in CPU time usage, and that it is feasible for clustering large document collections while keeping the high scalability of SKM algorithm.

As a future work, we plan to develop a method for determining the ideal MAC and SRL parameters by analyzing the datasets first. Moreover, we plan to extend our research towards applying our MCSKM algorithm on real-time streaming text datasets.

References

[1] 20NG, 20 News Groups Dataset, U.C.I. Machine Learning Repository., available at: http://archive.ics.uci.edu/ml/databases/20newsgroups/, last visited 2010.

[2] Aliguliyev M., "Clustering of Document Collection-A Weighting Approach," *Expert Systems with Applications,* vol. 36, no. 4, pp. 7904-7916, 2009.

[3] Batri K., Murugesh V., and Gopalan N., "Effect of Weight Assignment in Data Fusion Based Information Retrieval," *the International Arab Journal of Information Technology,* vol. 8, no. 3, pp. 244-250, 2011.

[4] Bezdek C., *Pattern Recognition with Fuzzy Objective Function Algorithms*, New York, Plenum Press, 1981.

[5] CLUTO, CLUTO Software for Clustering High-Dimensional Datasets., available at: http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview, last visited 2009.

[6] Dhillon S. and Modha S., "Concept Decompositions for Large Sparse Text Data using Clustering," *Machine Learning,* vol. 42, no. 1-2, pp. 143-175, 2001.

[7] Fayyad U., Piatetsky-Shapiro G., and Smyth P., "From Data Mining to Knowledge Discovery: An Overview," *Advances in Knowledge Discovery and Data Mining*, Menlo Park, pp. 1-34, 1996.

[8] Feldman R. and Sanger J., *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*, New York, Cambridge University Press, 2007.

[9] Han J., Kamber M., and Pei J., *Data Mining: Concepts and Techniques*, 3rd ed., Waltham, MA: Morgan Kaufmann Publishers, 2012.

[10] Hotho A., Nürnberger A., and Paaß G., "A Brief Survey of Text Mining," *LDV Forum-GLDV Journal for Computational Linguistics and Language Technology,* vol. 20, no. 1, pp. 19-62, 2005.

[11] Işık M. and Çamurcu Y., "K-Means ve Aşırı Küresel C-Means Algoritmaları ile Belge Madenciliği," *Marmara Üniversitesi, Fen Bilimleri Enstitüsü Dergisi,* vol. 22, pp. 1-18, 2010.

[12] Işık M., "Bölünmeli Kümeleme Yöntemleri İle Veri Madenciliği Uygulamaları," *MS Thesis*, Marmara University, Istanbul, Turkey, 2006.

[13] Juršič M. and Lavrač N., "Fuzzy Clustering of Documents," *Presented at the Conference on Data Mining and Data Warehouses*, Ljubljana, Slovenia, 2008.

[14] Luo C., Li Y., and Chung M., "Text Document Clustering based on Neighbors," *Data and Knowledge Engineering,* vol. 68, no. 11, pp. 1271-1288, 2009.

[15] Mendes S. and Sacks L., "Evaluating Fuzzy Clustering for Relevance-based Information Access," *in Proceedings of the 12th IEEE International Conference on Fuzzy Systems*, pp. 648-653, 2003.

[16] Salton G., Wong A., and Yang S., "A Vector Space Model for Automatic Indexing," *Communications of the ACM,* vol. 18, no. 11, pp. 613-620, 1975.

[17] Steinbach M., Karypis G., and Kumar V., "A Comparison of Document Clustering Techniques," *in Proceedings of the 6th ACM SIGKDD International Conference on Data Mining, Workshop on Text Mining*, Boston, 2000.

[18] WAP, WAP Dataset, Simon Fraser University, Database and Data Mining Lab., available at: http://ddm.cs.sfu.ca/software.html, last visited 2009.

[19] Witten H., Moffat A., and Bell C., *Managing Gigabytes: Compressing and Indexing Documents and Images*, San Francisco, CA: Morgan Kaufmann Publishers, 1999.

Volkan Tunali received the BSc and MSc degrees in computer engineering from Marmara University, Istanbul in 2001 and 2005 respectively. He received the PhD degree in computer and control education from Marmara University in 2012. He became Assistant Professor of the Software Engineering Department at Maltepe University in 2012. His research interests include data mining and knowledge discovery, text mining, information retrieval, and natural language processing. He is a member of ACM.

Turgay Bilgin received the BSc, PhD degrees in computer and control education from Marmara University, Istanbul in 2001 and 2007 respectively. His doctoral thesis was on the mining of high dimensional datasets. He became Assistant Professor of the Software Engineering Department at Maltepe University in 2008. His research interests are high dimensional data mining, web mining, service oriented architecture and web services. He is a member of ACM.

Ali Camurcu received the PhD degree in computer education from Marmara University, Istanbul in 1996. His current research interests are data mining, intelligent tutoring systems, and medical image processing. He is a professor of Computer Engineering in the Faculty of Engineering and Architecture at Fatih Sultan Mehmet Waqf University. He is a member of ACM.