ISSN:0975-9646

Incremental MVS based Clustering Method for Similarity Measurement

[1]Vinay C. Warad, [2]B Baron Sam

[1]*Department of Computer Science and Engineering, Sathyabama University Chennai, Tamil Nadu, India*

[2]*Assistant Professor, Department of Computer Science and Engineering, Sathyabama University Chennai, Tamil Nadu, India*

Abstract-As the amount of digital documents has been increasing dramatically over the years as the Internet grows, information management, search, and retrieval, etc., have become practically important problems. Identification of similar and dissimilar attributes is a challenging task. There have been many clustering algorithms published to identify the similarity between the elements in the given data set. The goal of cluster analysis is to partition a data set of N objects into subgroups such that those in each particular group are more similar to each other than to those of other groups this similarity between a pair of objects can be defined either explicitly or implicitly. In this paper, we introduce a novel Multiviewpoint-based similarity measure and traditional K- Means clustering methods. The major difference between a traditional dissimilarity/similarity measure and ours is that the former uses only a single viewpoint, while the latter utilizes multiple viewpoints, which are objects assumed to not be in the same cluster with the two objects being measured. The comparison between the K Means clustering and Incremental multiviewpoint based clustering methods for the similarity or dissimilarity accuracy measurement can be done. We use the java programming language for simulation.The simulation result shows the improved accuracy for the Incremental multiviewpoint based clustering.

Keywords- ***Document clustering, Multiviewpoint, K-Means similarity measure.***

Subject Classification-*Data Mining, Computer Networking and Network Security.*

1 INTRODUCTION

CLUSTERING is one of the most interesting and important topics in data mining. It is "the process of organizing objects into groups whose members are similar in some way". Developing methods to organize large amounts of unstructured text documents into a smaller number of meaningful clusters would be very helpful as document clustering is vital to such tasks as indexing, filtering, automated metadata generation, word sense disambiguation. The principle definition of clustering is to arrange data objects into separate clusters such that the intra-cluster similarity as well as the inter-cluster dissimilarity is maximized. Computation of similarity between categorical data objects in unsupervised learning is an important data mining problem. There are many clustering methods to support the data mining operation. As K Means clustering is one of the Traditional Clustering method it is still one of the top 10 clustering method in data mining [1]. K-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. k-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. It is the most frequently used partition clustering algorithm in practice. Another recent scientific discussion states that k-means is the favourite algorithm that practitioners in the related fields choose to use [2].

An optimal partition is found by optimizing a particular function of similarity among data. Basically, there is an implicit assumption that the true intrinsic structure of data could be correctly described by the similarity formula defined and embedded in the clustering criterion function. Hence, effectiveness of clustering algorithms under this approach depends on the appropriateness of the similarity measure to the data at hand. For instance, the original k-means has sum-of-squared-error objective function that uses Euclidean distance. In a very sparse and high-dimensional domain like text documents, spherical k-means, which uses cosine similarity (CS) instead of euclidean distance as the measure, is deemed to be more suitable [3], [4]. It appears to us that the nature of similarity measure plays a very important role in the success or failure of a clustering method. Our first objective is to derive a novel method for measuring similarity between data objects in sparse and high-dimensional domain, particularly text documents.

1.1 System Architecture

Initially various set of text documents have been chosen for the database. These text documents have different set of information. In the next step keyword identification is performed to choose different keywords.

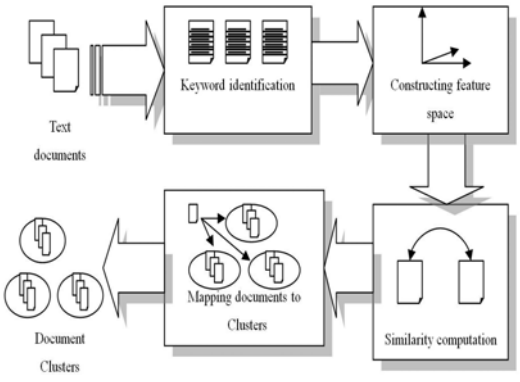

Fig 1.System Architecture

This keyword identification is performed from each text document figure (1). After this feature space is constructed in which feature extraction of all the text documents has been performed. Once the feature of the documents is constructed then based on the features of all text documents similarity computation is performed. In this similarity computation it is determined that how much each text document is related or similar to other text documents. Once the similarity computation is over then the clustering of the documents is performed. In this clustering process documents of similar type is grouped together to form a cluster. Whole of this process is called mapping documents to cluster.

The remaining of this paper is organized as follows: In Section 2, we review related literature on similarity and clustering of documents. We then present our proposal for document similarity measure in Section 3. It is followed by optimized Incremental MVS Clustering algorithms in Section 4. Section 5 presents the simulation results. We conclude the paper in section 6 and also the future scope is introduced.

2 RELATED WORK

There are many state-of-the-art clustering approaches that do not employ any specific form of measurement, for instance, probabilistic model-based method [5], nonnegative matrix factorization [6], information theoretic co-clustering [7] and so on. The problem formulation itself implies that some forms of measurement are needed to determine such similarity or dissimilarity. In this paper, we primarily focus on methods that indeed do utilize a specific measure. In the literature, Euclidean distance is one of the most popular measures

$$\text{Dist}(d_i - d_j) = \|d_i - d_j\| \quad \text{..... (1)}$$

It is used in the traditional k-means algorithm. The objective of k-means is to minimize the Euclidean distance between objects of a cluster and that cluster's centroid

$$\min \sum_{r=1}^{k} \sum_{d_i \epsilon S_r} \|d_i - C_r\|^2 \quad \text{..... (2)}$$

clustering, cosine similarity is more widely used. It is also a popular similarity score in text mining and information retrieval [7]. Particularly, similarity of two document vectors di and dj, $\text{Sim}(d_i,d_j)$, is defined as the cosine of the angle between them. For unit vectors, this equals to their inner product

$$\text{Sim}(d_i,d_j) = \cos(d_i,d_j) = d_i^t d_j \quad \text{..... (3)}$$

Cosine measure is used in a variant of k-means called spherical k-means [3]. The major difference between Euclidean distance and cosine similarity, and therefore between k-means and spherical kmeans, is that the former focuses on vector magnitudes, while the latter emphasizes on vector directions. Besides direct application in spherical k-means, cosine of document vectors is also widely used in many other document clustering methods as a core similarity measurement. Compared with euclidean distance and cosine similarity, the extended Jaccard coefficient takes into account both the magnitude and the direction of the document vectors. If the documents are instead represented by their corresponding unit vectors, this measure has the same effect as cosine similarity. In [8], Strehl et al. compared four measures: euclidean, cosine, Pearson correlation, and extended Jaccard, and concluded that cosine and extended Jaccard are the best ones on web documents.

More related to text data, there are phrase-based and concept-based document similarities. Lakkaraju et al. [9] employed a conceptual tree-similarity measure to identify similar documents. This method requires representing documents as concept trees with the help of a classifier. For clustering, Chim and Deng [10] proposed a phrasebased document similarity by combining suffix tree model and vector space model. They then used Hierarchical Agglomerative Clustering algorithm to perform the clustering task. There are also measures designed specifically for capturing structural similarity among XML documents. They are essentially different from the document-content measures that are discussed in this paper. In general, cosine similarity still remains as the most popular measure because of its simple interpretation and easy computation, though its effectiveness is yet fairly limited. In the following sections, we propose a novel way to evaluate similarity between documents, and consequently formulate new criterion functions for document clustering.

3 MVS SIMILARITY MEASUREMENTS

The cosine similarity in (3) can be expressed in the following form without changing its meaning:

$$\text{Sim}(d_i,d_j) = \cos(d_i\text{-}0,d_j\text{-}0) = (d_i - 0)^t\,(d_j\text{-}0), \quad \text{..... (4)}$$

where 0 is vector 0 that represents the origin point. According to this formula, the measure takes 0 as one and only reference point. The similarity between two documents di and dj is determined w.r.t. the angle between the two points when looking from the origin. To construct a new concept of similarity, it is possible to use more than just one point of reference. We may have a more accurate assessment of how close or distant a pair of points is, if we look at them from many different viewpoints. By standing at various reference points dh to view di, dj and working on their difference vectors, we define similarity between the two documents as

$$\underset{d_i,d_j \epsilon S_r}{Sim(d_i,d_j)} = \frac{1}{n-n_r} \sum_{d_h \epsilon S/S_r} Sim(di - dh, dj - dh) \quad (4)$$

where dh is the third point of observation. As described by the above equation, similarity of two documents di and dj given that they are in the same cluster is defined as the average of similarities measured relatively from the views of all other documents outside that cluster. The two objects to be measured must be in the same cluster, while the points from where to establish this measurement must be outside of the cluster. We call this proposal the Multiviewpoint-based Similarity, or MVS. From this point onwards, we will denote the proposed similarity measure between two document vectors di and dj by MVS(di, dj).

The final form of MVS in (4) depends on particular formulation of the individual similarities within the sum. If the relative similarity is defined by dot-product of the difference vectors, we have

$$\text{MVS}(d_i, d_j | d_i, d_j \in S_r) = \frac{1}{n-n_r}\sum_{d_h \in S/S_r}(d_i - d_h)^t(d_j - d_h)$$
$$= \frac{1}{n-n_r}\sum_{d_h} \cos(d_i - d_h, d_j - d_h)\ \|d_i - d_h\|\,\|d_j - d_h\| \quad (5)$$

The similarity between two points di and dj inside cluster Sr, viewed from a point dh outside this cluster, is equal to the product of the cosine of the angle between di and dj looking from dh and the euclidean distances from dh to these two points. This definition is based on the assumption that dh is not in the same cluster with di and dj. The smaller the distances kdi _ dhk and kdj _ dhk are, the higher the chance that dh is in fact in the same cluster with di and dj, and the similarity based on dh should also be small to reflect this potential. Therefore, through these distances, (5) also provides a measure of inter-cluster dissimilarity, given that points di and dj belong to cluster Sr, whereas dh belongs to another cluster. The overall similarity between di and dj is determined by taking average over all the viewpoints not belonging to cluster Sr. It is possible to argue that while most of these viewpoints are useful, there may be some of them giving misleading information just like it may happen with the origin point.

procedure BUILDMVSMATRIX(A)
  for $r \leftarrow 1 : c$ **do**
    $D_{S\setminus S_r} \leftarrow \sum_{d_i \notin S_r} d_i$
    $n_{S\setminus S_r} \leftarrow |S \setminus S_r|$
  end for
  for $i \leftarrow 1 : n$ **do**
    $r \leftarrow$ class of d_i
    for $j \leftarrow 1 : n$ **do**
      if $d_j \in S_r$ **then**
        $a_{ij} \leftarrow d_i^t d_j - d_i^t \frac{D_{S\setminus S_r}}{n_{S\setminus S_r}} - d_j^t \frac{D_{S\setminus S_r}}{n_{S\setminus S_r}} + 1$
      else
        $a_{ij} \leftarrow d_i^t d_j - d_i^t \frac{D_{S\setminus S_r} - d_j}{n_{S\setminus S_r} - 1} - d_j^t \frac{D_{S\setminus S_r} - d_j}{n_{S\setminus S_r} - 1} + 1$
      end if
    end for
  end for
  return $A = \{a_{ij}\}_{n \times n}$
end procedure

Fig 2. Procedure to build Similarity Matrix

However, given a large enough number of viewpoints and their variety, it is reasonable to assume that the majority of them will be useful. Hence, the effect of misleading viewpoints is constrained and reduced by the averaging step. It can be seen that this method offers more informative assessment of similarity than the single origin point-based similarity measure. The procedure for building MVS matrix is described in Fig. 2.

3.1 Proposed System

From a more technical viewpoint, our datasets consist of unlabeled objects—the classes or categories of documents that can be found are a priori unknown. The rationale behind clustering algorithms is that objects within a valid cluster are more similar to each other than they are to objects belonging to a different cluster. Thus, once a data partition has been induced from data, the expert examiner might initially focus on reviewing representative documents from the obtained set of clusters. Then, after this preliminary analysis, (s) he may eventually decide to scrutinize other documents from each cluster. By doing so, one can avoid the hard task of examining all the documents (individually) but, even if so desired, it still could be done. To solve the problems associated with the conventional clustering process we proposed an incremental MVS algorithm is as shown in the figure (3). This proposed module consists of many functional sub-modules such as Pre-Processing Module, Term Frequency, Calculating the number of clusters, incremental MVS Clustering techniques and Query Processing.

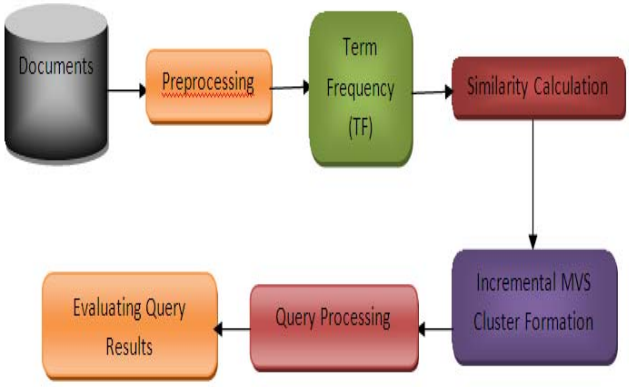

Fig 3. The Proposed Block Diagram

3.1.1 Pre-Processing Module:

Before running clustering algorithms on text datasets, we need to performe some pre-processing steps. In particular, stop-words (prepositions, pronouns, articles, and irrelevant document metadata) have been removed. Then, we adopted a traditional statistical approach for text mining, in which documents are represented in a vector space model. In this model, each document is represented by a vector containing the frequencies of occurrences of words, which are defined as delimited alphabetic strings, whose number of characters is between 4 and 25. We also used a dimensionality reduction technique known as Term Variance (TV) that can increase both the effectiveness and efficiency of clustering algorithms. TV selects a number of attributes (in our case 100 words) that have the greatest variances over the documents. In order to compute distances between documents, two measures have been used, namely: cosine-based distance and Levenshtein-based distance. The later has been used to calculate distances between file (document) names only.

3.1.2 Calculating the number of Clusters:

In order to estimate the number of clusters, a widely used approach consists of getting a set of data partitions with different numbers of clusters and then selecting that particular partition that provides the best result according to a specific quality criterion (e.g., a relative validity index). Such a set of partitions may result directly from a hierarchical clustering dendrogram or, alternatively, from multiple runs of a partitional algorithm (e.g., K-means) starting from different numbers and initial positions of the cluster prototypes.

4 Incremental MVS Based Clustering Technique

The clustering algorithms adopted in our study such as the partitional K-means and K-medoids, the hierarchical Single/Complete/Average Link, and the cluster ensemble based algorithm known as CSPA are popular in the machine learning and data mining fields, and therefore they have been used in our study. We assess a simple approach to remove outliers. This approach makes recursive use of the silhouette. Fundamentally, if the best partition chosen by the silhouette has singletons (i.e., clusters formed by a single object only), these are removed. Then, the clustering process is repeated over and over again until a partition without singletons is found. At the end of the process, all singletons are incorporated into the resulting data partition (for evaluation purposes) as single clusters. We denote our clustering framework by MVSC, meaning Clustering with Multiviewpoint-based Similarity. Subsequently, we propose an incremental MVS Clustering, which is MVSC with criterion of incremental function. The main goal is to perform document clustering by optimizing incremental function (I_R or I_V). the incremental k-way algorithm, a sequential version of k-means is employed. Considering that the expression which depends only on nr and Dr, can be written in a general form

$$I_V = \sum_{r=1}^{k} I_r(n_r, D_r) \quad \text{..... (6)}$$

Where $I_r(n_r, D_r)$ corresponds to the objective value of cluster r. With this general form, the incremental optimization algorithm, which has two major steps Initialization and Refinement, is described in Fig. 4.

At Initialization, k arbitrary documents are selected to be the seeds from which initial partitions are formed. Refinement is a procedure that consists of a number of iterations. During each iteration, the n documents are visited one by one in a totally random order. Each document is checked if its move to another cluster results in improvement of the objective function. If yes, the document is moved to the cluster that leads to the highest improvement. If no clusters are better than the current cluster, the document is not moved. The clustering process terminates when an iteration completes without any documents being moved to new clusters. Unlike the traditional k-means, this algorithm is a stepwise optimal procedure. While k-means only updates after all n documents have been reassigned, the incremental clustering algorithm updates immediately whenever each document is moved to new cluster. Since every move when happens increases the objective function value, convergence to a local optimum is guaranteed.

1: **procedure** INITIALIZATION
2: Select k seeds $s_1, \ldots, s_k$ randomly
3: $cluster[d_i] \leftarrow p = \arg\max_r \{s_r^t d_i\}, \forall i = 1, \ldots, n$
4: $D_r \leftarrow \sum_{d_i \in S_r} d_i$, $n_r \leftarrow |S_r|$, $\forall r = 1, \ldots, k$
5: **end procedure**
6: **procedure** REFINEMENT
7: **repeat**
8: $\{v[1:n]\} \leftarrow$ random permutation of $\{1, \ldots, n\}$
9: **for** $j \leftarrow 1 : n$ **do**
10: $i \leftarrow v[j]$
11: $p \leftarrow cluster[d_i]$
12: $\Delta I_p \leftarrow I(n_p - 1, D_p - d_i) - I(n_p, D_p)$
13: $q \leftarrow \arg\max_{r, r \neq p} \{I(n_r + 1, D_r + d_i) - I(n_r, D_r)\}$
14: $\Delta I_q \leftarrow I(n_q + 1, D_q + d_i) - I(n_q, D_q)$
15: **if** $\Delta I_p + \Delta I_q > 0$ **then**
16: Move d_i to cluster q: $cluster[d_i] \leftarrow q$
17: Update D_p, n_p, D_q, n_q
18: **end if**
19: **end for**
20: **until** No move for all n documents
21: **end procedure**

Fig 4. Algorithm: Incremental clustering.

During the optimization procedure, in each iteration, the main sources of computational cost are .

Searching for optimum clusters to move individual documents to: O (nz.k)

Updating composite vectors as a result of such moves: O(m.k).

where nz is the total number of nonzero entries in all document vectors. Our clustering approach is partitional and incremental; therefore, computing similarity matrix is absolutely not needed. If τ denotes the number of iterations the algorithm takes, since nz is often several tens times larger than m for document domain, the computational complexity required for clustering with IR and O(nz . τ .k).

5 Simulation Results

The objective of this section is to compare our proposed incremental MVS Clustering with the existing K-Means algorithm which uses the specific similarity measures and criterion functions for document clustering. The similarity measure to be compared includes Euclidean distance and cosine similarity. Our simulation programs are implemented in Java. Data set selected for the simulation is the text data as shown in the figure 5.

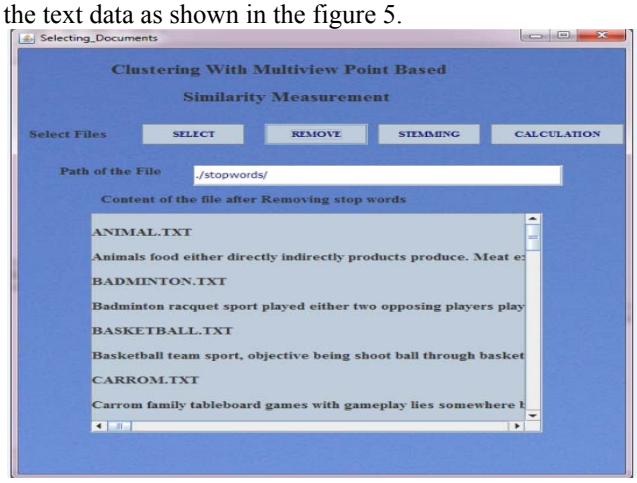

Fig 5. Experimental data set

From a more technical viewpoint, our datasets consist of unlabeled objects—the classes or categories of documents that can be found are a priori unknown. Moreover, even assuming that labeled datasets could be available from previous analyses, there is almost no hope that the same classes (possibly learned earlier by a classifier in a supervised learning setting) would be still valid for the upcoming data, obtained from other computers and associated to different investigation processes. The distance computations from the centroid are shown in the figure 6.

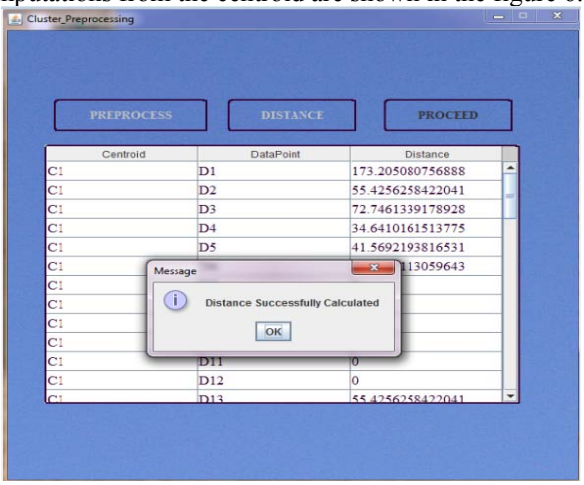

Fig 6. Distances from the centroid

More precisely, it is likely that the new data sample would come from a different population. In this context, the use of clustering algorithms, which are capable of finding latent patterns from text documents found in seized computers, can enhance the analysis performed by the expert examiner. The rationale behind clustering algorithms is that objects within a valid cluster are more similar to each other than they are to objects belonging to a different cluster. Thus, once a data partition has been induced from data, the expert examiner might initially focus on reviewing representative documents from the obtained set of clusters. Here the two types of clusters have been analyzed, viz K-means Clustering and the Incremental MVS Clustering methods. Figure 7 shows different attributes belong to different clusters.

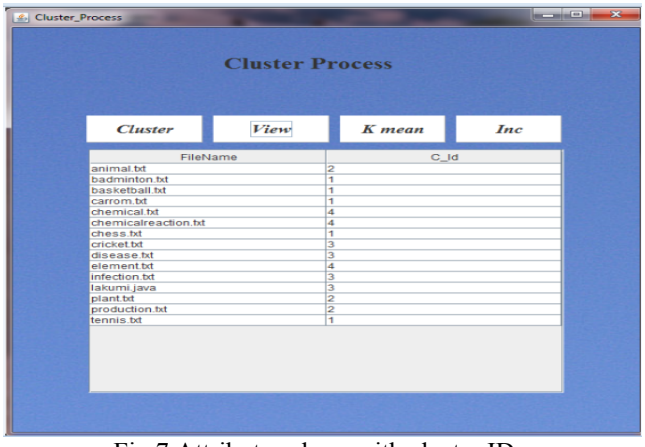

Fig 7 Attributes along with cluster IDs

To form the clusters different attributes have been considered along with the distance. Distances of the each attributes belonging to different clusters have been shown in the figure 8.

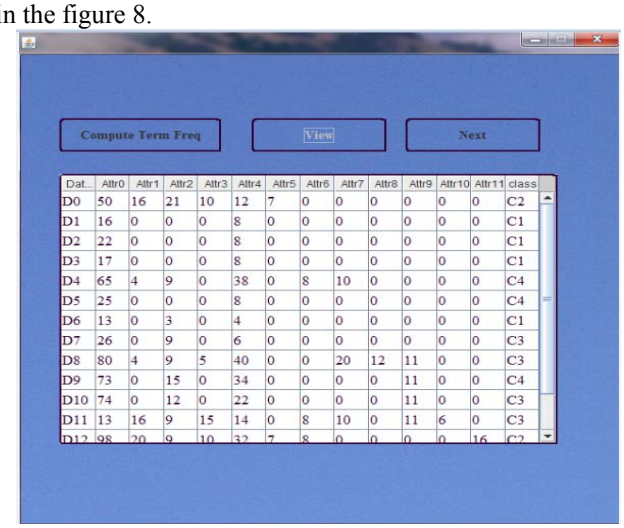

Fig 8 Attribute distances and their classes

Clusters and the different cluster members' distances are shown here in figure 9. This is the visualization results belonging to all the clusters.

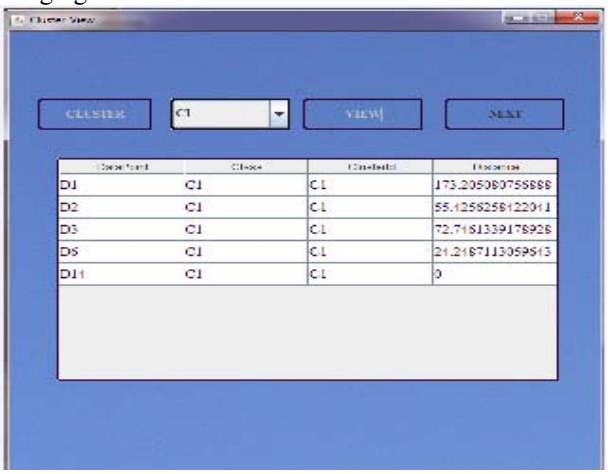

Fig 9 Cluster members and their distances

The similarity measurement for both k- means and Incremental MVS Clustering are calculated using (6). Compare to k-means the proposed an increase in the accuracy of the similarity measurement within the clusters. The experimental value and its corresponding bar chart have been shown in figure 10 respectively.

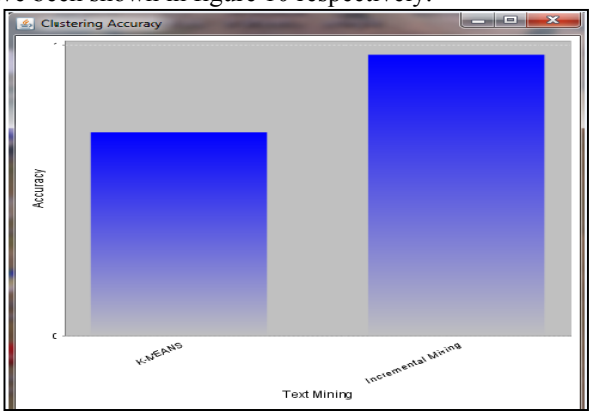

Fig 10 Comparison plot

6 CONCLUSIONS AND FUTURE WORK

This paper proposes a incremental Multiviewpoint-based Similarity measuring method, named incremental MVS. Simulation result shows that incremental MVS is potentially more suitable for text documents than the popular cosine similarity. This paper also focuses on partitional clustering of documents. The key contribution of this paper is the fundamental concept of similarity measure from multiple viewpoints. Future methods could make use of the same principle, but define alternative forms for the relative similarity. In the future, it would also be possible to apply the proposed criterion functions for hierarchical clustering algorithms. Finally, we have shown the application of incremental MVS clustering algorithms for text data. It would be interesting to explore how they work on other types of sparse and high-dimensional data.

REFERENCES

[1] X. Wu, V. Kumar, J.R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G.J.McLachlan, A. Ng, B. Liu, P.S. Yu, Z.-H. Zhou, M. Steinbach, D.J. Hand, and D. Steinberg, "Top 10 Algorithms in Data Mining,", Knowledge Information Systems, vol. 14, no. 1, pp. 1-37, 2007.

[2] I. Guyon, U.V. Luxburg, and R.C. Williamson, "Clustering: Science or Art?", Proc. NIPS Workshop Clustering Theory, 2009.

[3] I. Dhillon and D. Modha, "Concept Decompositions for Large Sparse Text Data Using Clustering," Machine Learning, vol. 42, nos. 1/2, pp. 143-175, Jan. 2001.

[4] S. Zhong, "Efficient Online Spherical K-means Clustering," Proc. IEEE Int'l Joint Conf. Neural Networks (IJCNN), pp. 3180-3185, 2005.

[5] A. Banerjee, I. Dhillon, J. Ghosh, and S. Sra, "Clustering on the Unit Hypersphere Using Von Mises-Fisher Distributions,"J. Machine Learning Research, vol. 6, pp. 1345-1382, Sept. 2005.

[6] W. Xu, X. Liu, and Y. Gong, "Document Clustering Based on Non-Negative Matrix Factorization," Proc. 26th Ann. Int'l CM SIGIR Conf. Research and Development in Informaion Retrieval, pp. 267-273, 2003.

[7] I.S. Dhillon, S. Mallela, and D.S. Modha, "Information-Theoretic Co-Clustering," Proc. Ninth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD), pp. 89-98, 2003.

[8] A. Strehl, J. Ghosh, and R. Mooney, "Impact of Similarity Measures on Web-Page Clustering," Proc. 17th Nat'l Conf. Artificial Intelligence: Workshop of Artificial Intelligence for Web Search (AAAI), pp. 58-64, July 2000.

[9] P. Lakkaraju, S. Gauch, and M. Speretta, "Document Similarity Based on Concept Tree Distance," Proc. 19th ACM Conf. Hypertext and Hypermedia, pp. 127-132, 2008.

[10] H. Chim and X. Deng, "Efficient Phrase-Based Document Similarity for Clustering," IEEE Trans. Knowledge and Data Eng., vol. 20, no. 9, pp. 1217-1229, Sept. 2008.