Hierarchical Clustering With Multi view point Based Similarity Measure

Vemulapalli V Srivalli[#1] , **R.Ganesh Kumar** [*2] , **Dr. Jitendranath Mungara**[#3]

[#1] M.Tech(CSE) 4th sem ,Dept. of CSE, CMR Institute of Technology, Bengaluru, India.
[*2] Associate Prof. & HOD, Dept. of ISE, CMR Institute of Technology, Bengaluru, India
[#3] Professor & Dean ,Dept. of CSE & ISE, CMR Institute of Technology, Bengaluru , India

Abstract -- A cluster is a group of similar objects placed together and are dissimilar to other cluster objects. In this paper, we introduce Hirarchical Clustering with Multiple view points based on different similarity measures. We use two measures for inter cluster and intracluster relation between objects . The former clustering process focuses on partitional clustering of multiview point documents, which are not focused on sparse and high-dimensional data. Using Hirarchical Multiview point, we can achieve more informative assessment of similarity. We compare our approach with former model on various document collections to verify the advantages of our proposed method.

Keywords **—Hierarchical clustering, document clustering, similarity measure**

1 INTRODUCTION

Clustering groups data into subsets in such a manner that identical instances are collected together, while different instances belong to different groups. The instances are thereby arranged into an efficient depiction that characterizes the populace being sampled. Clustering of objects is as ancient as the human need for describing the salient characteristics of men and objects and identifying them with a type. Therefore, it embraces a choice of scientific disciplines: from mathematics and statistics to biology and genetics, the entire of which uses different terms to describe the topologies formed using this analysis. From biological “taxonomies”, to medical “syndromes” and genetic “genotypes” to manufacturing ”group technology”—the problem is same: forming groups of entities and transfer individuals to the proper groups within it. Since clustering is the grouping of similar instances/entities, a number of measure that can choose whether two objects are similar or dissimilar is required.

A general move towards the clustering problem is to treat it as an optimization process. An best partition is found by optimizing a exacting function of similarity (or distance) among data. Basically, there is an hidden assumption that the true inherent structure of data could be correctly describe by the similarity formula defined and fixed in the clustering decisive factor. Hence, efficiency of clustering algorithms under this approach depends on the accuracy of the similarity measure to the data at hand. For example, the original k-means has sum-of-squared-error intent function that use euclidean distance. In a very sparse and high-dimensional realm like text documents, spherical k-means, which uses cosine similarity (CS) instead of Euclidean distance as the measure, is deemed to be more suitable [1], [2].Correlation coefficient is standardized angular division by centering the coordinates to its mean value. The value between -1 and +1. Correlation measure similarity rather than distance or dissimilarity.

Similarity is quite tricky to measure. Similarity is measure that reflects the strong point of relationship between two objects or two features. This quantity is usually having series of either -1 to +1 or normalize into 0 to 1. If the similarity between attribute i and attribute j is denoted by Sij, we can measure this quantity in numerous ways depending on the level of measurement that we have. Dissimilarity[3] measure
the discrepancy between the two objects based on several features. Variation may also be viewed as measure of chaos between two objects.

Farley and Raftery (1998) propose separating the clustering methods into two main groups: hierarchical and partitioning methods. Hierarchical clustering solutions[20], which are in the shape of trees called dendograms, are of great attention for a number of application domains. They provide a analysis of the data at different levels of abstraction. The solidity of clustering solutions at different levels of granularity allows flat partitions of different granularity to be extracted during data analysis, making them perfect for interactive examination and visualization. In addition, there are various types when clusters have associate clusters, and the hierarchical structure is indeed a normal constrain on the underlying application domain (e.g., biological taxonomies, phylogenetic trees, etc) [4].

One popular approach in document clustering is agglomerative hierarchical clustering (Kaufman and Rousseeuw, 1990). Algorithms in this family build the hierarchy bottom-up by iteratively computing the similarity between all pairs of clusters and then merging the most similar pair. An agglomerative clustering starts with one-point (singleton) clusters and recursively merges two or more most appropriate clusters. the hierarchical agglomerative clustering algorithm CURE (Clustering Using Representatives) has a number of novel features of general significance. It takes special care with outliers and with label assignment stage. It also uses two devices to achieve scalability. The first one is data sampling. The second device is data partitioning in p partitions, so that fine granularity clusters are constructed in partitions first. A major feature of CURE is that it represents a cluster by a fixed number c of points scattered around it. The distance between two clusters used in the agglomerative process is equal to the minimum of distances between two scattered representatives. Single and

average link closeness is replaced by representatives, aggregate closeness. Selecting representatives scattered around a cluster makes it possible to cover non-spherical shapes. The agglomerative process continues until the stopping criterion is satisfied.

The remaining of this paper is organised as follows: in section 2, we review related literature on similarity and clustering of documents. We then present our proposed system architecture and explanation in section 3. Extensive experiments on real-world benchmark data sets are presented and discussed in Sections 4. Finally, conclusions and potential future work are given in Section 5.

2 RELATED WORK

TABLE 1
Notations

Notation	Description
n	No. of Documents
m	No. of terms
c	No. of classes
k	No. of clusters
d	document vector,‖d‖=1
S={d1,...dn}	set of all documents
Sr	set of documents in cluster r
D =∑di ε S di	composite vector of all documents
Dr =∑ di ε Sr di	composite vector of cluster r
C =D/n	centroid vector of all the documents
Cr =Dr/nr	centroid vector of cluster r,nr =|Sr|

First of all, Table 1 review the fundamental notations that will be used widely throughout this paper to signify documents and associated concepts. Each document in a corpus corresponds to an m-dimensional vector d, where m is the total number of terms that the document corpus has. Document vectors are often subjected to a small number of weighting schemes, such as the regular Term Frequency-Inverse Document Frequency (TF-IDF), and normalized to have unit length. The standard definition of clustering is to organize data objects into separate clusters such that the intracluster similarity with the intercluster dissimilarity is maximized. The problem formulation implies that some forms of measurement are desired to determine such similarity or dissimilarity. Many modern clustering approaches that do not employ any exact form of measurement, for instance, probabilistic model-based method [5], nonnegative matrix factorization [6], information theoretic coclustering [7] and so on.

some of the similarity measures explained briefly below based on single view point:

Euclidean Distance

Euclidean distance is a regular metric for geometrical problems. It is the common distance between two points and can be without difficulty measured with a ruler in two- or three-dimensional space. It is also the default distance measure used with the K-means algorithm.

Euclidean distance is one of the most popular measures: Dist (di, dj) = | |di − dj ||. It is used in the traditional k-means algorithm[8]. The objective of k-means is to minimize the Euclidean distance between objects of a cluster and that cluster's centroid:

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

Particularly, similarity of two document vectors di and dj , Sim(di, dj), is defined as the cosine of the angle between them. For unit vectors, this equals to their inner product:

$$sim(d_i, d_j) = \cos(d_i, d_j) = d_i^t d_j \quad --(2)$$

Cosine Similarity

When documents are represented as term vectors, the similarity of two documents corresponds to the correlation between the vectors. This is quantified as the cosine of the angle between vectors, that is, the so-called cosine similarity. Cosine similarity is one of the most popular similarity measure practical to text documents, such as in various information retrieval applications [9] and clustering too [10]. An important property of the cosine similarity is its independence of document length.

$$\max\sum_{r=1}^{k}\sum_{d_i\in S_r}\frac{d_i^t C_r}{||C_r||} \quad --(3)$$

Jaccard Coefficient

The Jaccard coefficient, which is sometimes referred to as the Tanimoto coefficient, measures similarity as the intersection divided by the union of the objects. For text document, the Jaccard coefficient compares the sum weight of shared terms to the sum weight of terms that are present in either of the two document but are not the shared terms. The formal definition is:

$$Sim_{eJacc}[u_i, u_j] = \frac{u_i^t u_j}{||u_i||^2 + ||u_j||^2 - u_i^t u_j} \quad --(4)$$

Pearson Correlation Measure

Correlation Clustering, introduced by Bansal, Blum and Chawla [11], provides a method for clustering a set of objects into the best possible number of clusters, without specifying that number in proceed. Correlation Clustering that does not require a bound on the number of clusters that the data is partitioned into. Rather, Correlation Clustering[12] divides the data into the optimal number of clusters based on the similarity between the data points. In their paper, [8] Bansal et al. discuss two objectives of correlation clustering: minimizing disagreements and maximizing agreements between clusters.

The normalized Pearson correlation is defined as:

$$S(x_i, x_j) = \frac{(x_i - \bar{x_i})^T.(x_j - \bar{x_j})}{||x_i - \bar{x_i}||x_j - \bar{x_j}||} \quad --(5)$$

where $\bar{x_i}$ denotes the average feature value of x over all dimensions.

3. Hirarchical MVS Similarity

Our Novel Similarity Measure

To construct a new concept of similarity, it is possible to use more than one point of reference.

we define similarity between the two documents as

$$Sim_{d_i,d_j\in S_r}(d_i, d_j) = \frac{1}{n - n_r}\sum_{d_h\in S\setminus S_r} Sim(d_i - d_h, d_j - d_h) \quad --(6)$$

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 Multi viewpoint-based Similarity[13], or MVS. For these MVS Clustering applied in our proposed system. we will denote the proposed similarity measure between two document vectors di and dj by $MVS(d_i, d_j | d_i, d_j \in S_r)$, or occasionally MVS(di, dj) for short.

The final form of MVS[14] depends on particular formulation of the individual similarities inside the sum. If the comparative similarity is defined by dot-product of the distinction vectors, we have

$$MVS(d_i, d_j | d_i, d_j \in S_r) = \frac{1}{n-n_r}\sum_{d_h \in S \setminus S_r}(d_i - d_h)^t(d_j - d_h) = \frac{1}{n-n_r}\sum_{d_h}\cos(d_i - d_h, d_j - d_h)||d_i - d_h||||d_j - d_h|| \quad --(7)$$

The similarity between two points di and dj inside cluster S_r, viewed from a point d_h, which is outside this cluster is equal to the product of the cosine of the angle between di and dj looking from dh and the correlation distances from dh to these two points. It can be seen that this method offers more informative assessment of similarity than the single origin point-based similarity measure.

Proposed System

In our proposed method, we are using correlation similarity and cosine similarity to measure the similarity between objects in the same cluster and dissimilarity between objects in the different cluster groups.

Proposed Architecture of the Hirarchical MVS is shown in the Fig.1. Set of documents are taken as input from the user, then each block performs the operations on the documents to form the final hierarchical clustering. Blocks are explained below.

Preprocessing

Preprocessing done in two steps i.e removal of stopwords and stemming the document. Stop-words are very common words that do not provide any useful information to us, such as “and”, “the”, “which”, “is”, etc.. It is often useful to get rid of these words otherwise they might mislead the clustering process by including frequent terms that are not informative to us. Word stemming is the process of converting different forms of a word into one canonical form. Words like “compute”, “computing”, “computer” are all changed to a single word “compute”. This is necessary to avoid treating different variations of a word distinctly. Word stemming was done using the popular Porter stemming algorithm [15].

Multiview Point Similarity Measure

The two measures are used to make sure the similarity and dissimilarity between the documents. By using the multiple measure point, we will obtain most appropriate clustered documents. The two measures we used are cosine similarity and correlation similarity.

The correlation between vectors A and B are defined as follows:

$$r(A,B) = \frac{\frac{1}{n}\sum_i AiBi - \mu A \mu B}{\sigma A \sigma B} \quad --(8)$$

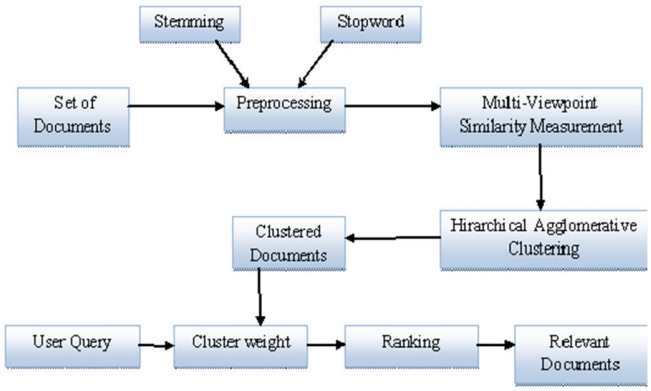

Fig 1: Proposed Architecture for Hirarchical MVS

The cosine of two vectors can be derived by using the Euclidean dot product formula:

$$a.b = ||a||.||b||\cos\theta \quad --(9)$$

Given two vectors of attributes A and B, the cosine similarity θ, is represented using a dot product and magnitude as

$$similarity = \cos(\theta) = \frac{A.B}{||A||||B||} = \frac{\sum_{i=1} A_i * B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} * \sqrt{\sum_{i=1}^{n}(B_i)^2}} \quad --(10)$$

Incremental Mining

In the incremental optimization algorithm[18][19], we have two major steps Initialization and Refinement.

At Initialization, k arbitrary documents are chosen to be the seeds from which primary partitions are formed. Refinement is a process that consists of a number of iterations. In each iteration, the n documents are randomly visited one by one. Each document is verified, if its move to another cluster results in progress of the objective function. Then the document is moved to the cluster that leads to the highest improvement. If no cluster is 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. The incremental clustering algorithm updates instantly whenever each document is moved to new cluster. Since every move results increases the objective function value, convergence to a local optimum is guaranteed.

Hierarchical Agglomerative Clustering

Hierarchical clustering methods are categorized into agglomerative (bottom-up) and divisive (top-down). An agglomerative clustering starts with one-point (singleton) clusters and recursively merges two or more most appropriate clusters. Advantages of hierarchical clustering are Embedded flexibility regarding the level of granularity, Ease of handling of any forms of similarity or distance and therefore, applicability to any attribute types. The classic approaches to hierarchical clustering are presented in the sub-section Linkage Metrics. Hierarchical clustering are based on linkage metrics that results in clusters of appropriate (convex) shapes.

Necessitate to define distance between clusters:

Single-linkage: distance between closest pair of points
Average-linkage: distance between centroids of two clusters
Complete-linkage: distance between farthest
Steps in Agglomerative Hierarchical Clustering

1. Start with N clusters, each containing a single entity, and an N × N symmetric matrix of distances (or similarities) Let dij = distance between item i and item j.
2. Search the distance matrix for the nearest pair clusters (i.e., the two clusters that are separated by the smallest distance). Denote the distance between these most similar clusters A and B by dAB .
3. Merge clusters A and B into a new cluster, labeled S. Update the entries in the distance matrix by
   a. Deleting the rows and columns corresponding to clusters A and B, and
   b. Adding a row and column giving the distances between the new cluster S and all the remaining clusters.
4. Repeat steps (2.) and (3.) a total of N-1 times.

Use distance matrix as clustering criteria. This method does not require the number of clusters k as an input, but needs a termination condition. In our proposed system we are using average linkage method for HAC clustering.

Average Linkage Method

The distance between clusters is the average distance between pairs of observations. Average linkage tends to join clusters with small variances, and it is slightly biased toward producing clusters with the same variance.

• Because it considers all members in the cluster rather than just a single point, however, average linkage tends to be less influenced by extreme values than other methods.

The average linkage between K and L Clusters are find by using:

$$D_{KL} = \frac{1}{n_K n_L} \sum_{i \in C_K} \sum_{j \in C_L} d(x_i, x_j) \quad \text{---(11)}$$

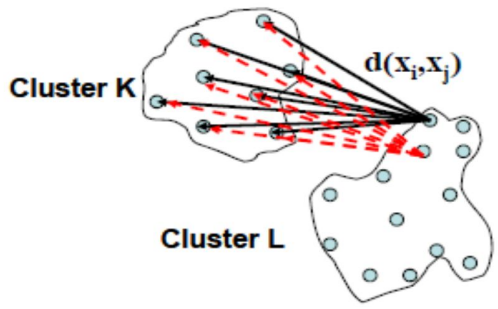

Fig 2: Average Linkage Method

4. PERFORMANCE EVALUATION

To verify the advantages of our proposed methods, we assess their performance in experiments on document data. The objective of this section is to compare Hierarchical MVSC with the existing algorithms that also use specific similarity measures and criterion functions for document clustering. The similarity measures to be compared includes euclidean distance, cosine similarity, and partitional MVS.

Document Collections

The data corpora that we used for experiments consist of 20 benchmark document data sets. we included 20 text collections so that the examination of the clustering methods is more methodical and extensive. These data sets are provided together with CLUTO by the toolkit's authors [16]. They had been used for experimental testing in earlier papers, and their source and origin had also been described in details [9], [17]. Table 2 summarizes their characteristics. The corpora present a assortment of size, number of classes and class balance. They were all preprocessed by normal procedures, including stop-word removal, stemming, removal of too rare as well as too frequent words, TF-IDF weighting and normalization.

Experimental Setup and Evaluation

To demonstrate how well HMVSCs can perform, we compare them with five other clustering methods on the 20 data sets in Table 2. In summary, the five clustering algorithms are

. Hierarchical MVSC
. MVSC-IR: MVSC using criterion function IR
. MVSC-IV : MVSC using criterion function IV
. k-means: standard k-means with euclidean distance
. Spkmeans: spherical k-means with CS

Our Hierarchical MVSC programs are implemented in Java.

TABLE 2
Document Data Sets

Data	Source	c	n	m	B alance
Fbis	TREC	17	2,463	2,000	0.075
Hitech	TREC	6	2,301	13,170	0.192
Kla	WebAce	20	2,340	13,859	0.018
Reuters	Reuters	7	2, 500	4,977	0.082
Tr45	TREC	10	690	8,260	0.088
Tr23	TREC	6	204	5,831	0.066
Tr12	TREC	8	313	5,799	0.097
Tr11	TREC	9	414	6,424	0.045
Sports	TREC	7	8,580	18,324	0.036
New3	TREC	44	9,556	36,306	0.149
La12	TREC	6	6,279	21,604	0.282
Classic	cacm	4	7,089	12,009	0.323
Wap	webACE	20	1,560	8,440	0.015
Reviews	TREC	5	4,069	23,220	0.099
Tr31	TREC	7	927	10,127	0.0006
Re1	reuters	25	1,657	3,758	0.027
Re0	reuters	13	1,504	2,886	0.018
La2	TREC	6	3,075	15,211	0.274
La1	TREC	6	3,204	17,273	0.290
Klb	webACE	6	2,340	13,859	0.043

C:# of classes, n:# of Documents, M: # of words
Balance=(smallest class size)/(largest class size)

Performance relative to different α values is presented in a later section. The other algorithms are provided by the C library interface which is available freely with the CLUTO toolkit [16]. For each data set, cluster number is predefined equal to the number of true class, i.e., k = c.

None of the above algorithms are guaranteed to find global optimum, and all of them are initialization dependent. Hence, for each method, we performed clustering a few times with randomly initialized values, and chose the best trial in terms of

the corresponding objective function value. In all the experiments, each test run consisted of 10 trials. Moreover, the result reported here on each data set by a particular clustering method is the average of 10 test runs.

After a test run, clustering solution is evaluated by comparing the documents' assigned labels with their true labels provided by the corpus. Three types of external evaluation metric are used to assess clustering performance.

They are the FScore, Normalized Mutual Information (NMI), and Accuracy. FScore is an equally weighted combination of the "precision" (P) and "recall" (R) values used in information retrieval.

$$\text{F Score} = \sum_{i=1}^{k} \frac{n_i}{n} \max_j (F_{i,j}) \quad \text{--(12)}$$

$$\text{where } F_{i,j} = \frac{2 * P_{i,j} * R_{i,j}}{P_{i,j} + R_{i,j}};\ P_{i,j} = \frac{n_{i,j}}{n_j}, R_{i,j} = \frac{n_{i,j}}{n_i}$$

where ni denotes the number of documents in class i, nj the number of documents assigned to cluster j, and ni;j the number of documents shared by class i and cluster j. From another aspect, NMI measures the information the true class partition and the cluster assignment share. It measures how much knowing about the clusters helps us know about the classes

$$\text{NMI} = \frac{\sum_{i=1}^{k}\sum_{j=1}^{k} n_{i,j} \log\left(\frac{n.n_{i,j}}{n_i n_j}\right)}{\sqrt{\left(\sum_{i=1}^{k} n_i \log\frac{n_i}{n}\right)\left(\sum_{j=1}^{k} n_j \log\frac{n_j}{n}\right)}} \quad \text{--(13)}$$

Finally, Accuracy measures the fraction of documents that are correctly labels, assuming a one-to-one correspondence between true classes and assigned clusters. Let q denote any possible permutation of index set f1; . . . ; kg, Accuracy is calculated by

$$\text{Accuracy} = \frac{1}{n} \max_q \sum_{i=1}^{k} n_{i,q(i)}. \quad \text{--(14)}$$

The best mapping q to determine Accuracy could be found by the Hungarian algorithm.2 For all three metrics, their range is from 0 to 1, and a greater value indicates a better clustering solution.

Results

Fig 3 shows the Accuracy of the seven clustering algorithms on the 20 text collections. Presented in a different way, clustering results based on FScore and NMI are reported in Tables 3 and 4, respectively. For each data set in a row, the value in bold and underlined is the best result, while the value in bold only is the second to best.

It can be observed that HMVSC perform consistently well. In Fig. 3, 19 out of 20 data sets, except reviews, either both or one of MVSC approaches are in the top two algorithms. The next consistent performer is MVSC-IV. The other algorithms might work well on certain data set. For example, SpKMeans yields outstanding result on wap; KMeans are good on reviews. But they do not fare very well on the rest of the collections.

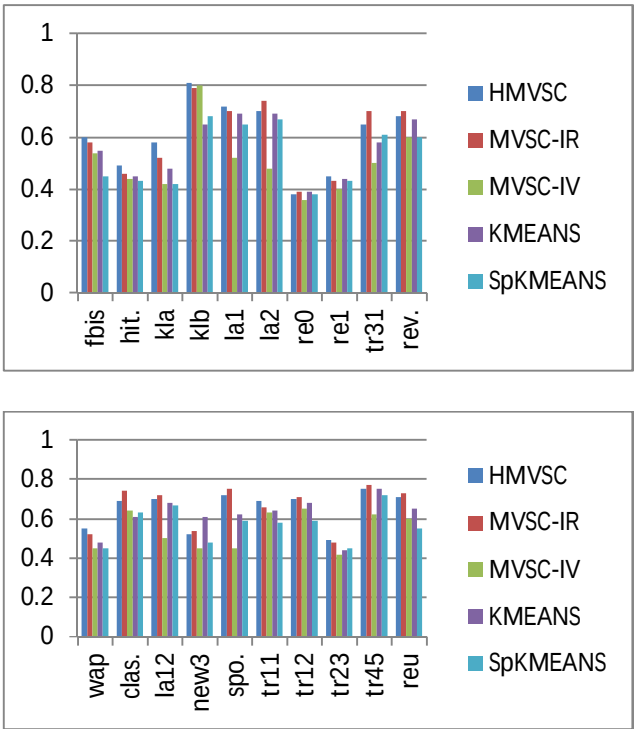

Fig 3: Clustering results in Accuracy. Left-to-right in legend corresponds to left-to-right in the plot.

Data	HMVSC	MVSC-IR	MVSC-IV	k-means	spkmeans
fbis	**0.645**	**0.613**	0.578	0.584	0.482
hitech	**0.512**	**0.528**	0.464	0.494	0.492
kla	**0.62**	**0.592**	0.502	0.545	0.492
klb	**0.873**	0.775	**0.825**	0.729	0.74
la1	**0.719**	0.565	**0.723**	**0.719**	0.689
la2	**0.721**	**0.75**	0.54	0.7	0.65
re0	**0.46**	**0.459**	0.42	0.42	0.45
re1	**0.514**	0.493	0.45	**0.5**	0.468
tr31	**0.728**	**0.79**	0.59	0.67	0.49
reviews	**0.734**	0.655	0.645	**0.745**	0.69
wap	**0.61**	0.575	0.516	0.54	**0.789**
clas.	**0.744**	0.658	**0.71**	0.68	0.513
la12	0.719	**0.735**	0.6	**0.72**	0.7
new3	**0.548**	**0.548**	0.5	**0.558**	0.51
spo.	**0.803**	**0.807**	0.51	0.702	0.69
tr11	**0.749**	**0.724**	0.704	0.72	0.67
tr12	**0.743**	0.69	**0.76**	0.72	0.64
tr23	**0.561**	**0.55**	0.51	0.52	0.53
tr45	0.788	**0.79**	0.71	**0.799**	0.78
reu	**0.775**	**0.776**	0.66	0.72	0.65

Fig 4: Clustering Results in FScore

Data	HMVSC	MVSC-IR	MVSC-IV	k-means	spkmeans
fbis	**0.606**	**0.595**	0.584	0.593	0.527
hitech	**0.352**	**0.329**	0.27	0.295	0.28
kla	**0.63**	**0.6**	0.56	0.59	0.537
klb	**0.74**	**0.65**	0.63	**0.65**	0.55
la1	**0.57**	**0.571**	0.4	**0.57**	0.56
la2	**0.56**	**0.61**	0.39	0.4	**0.56**
re0	**0.41**	0.402	0.388	**0.593**	**0.41**
re1	**0.61**	**0.59**	0.532	**0.59**	0.515
tr31	**0.62**	**0.658**	0.49	0.607	0.545
reviews	0.59	**0.603**	0.46	0.6	**0.639**
wap	**0.611**	**0.59**	0.57	0.577	0.575
clas.	**0.584**	**0.655**	0.58	0.57	0.543
la12	**0.59**	0.58	0.378	**0.63**	0.55
new3	0.621	**0.63**	0.57	**0.633**	0.57
spo.	0.67	**0.701**	0.45	**0.671**	0.59
tr11	**0.71**	**0.674**	0.67	0.66	0.66
tr12	**0.69**	**0.69**	**0.65**	0.42	0.64
tr23	0.43	**0.44**	0.36	**0.55**	0.37
tr45	**0.73**	**0.74**	0.65	0.61	0.67
reu	**0.633**	**0.64**	0.512	0.62	0.53

Fig 5: Clustering Results in NMI

5. CONCLUSION

In this paper, we propose Hierarchical Multiview point based similarity measuring method. Theoritical analysis and empirical examples show that Hierarchical MVS is suitable for sparse and High dimensional data. Compared with partitional MVS clusters.

The key contribution of this paper is the fundamental concept of hirarchical clustering from multiple view points. Future based on the same concept using different alternative measures and use other methods to combine the relative similarities according to the different viewpoints.

REFERENCES

[1] 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.

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

[3] D. Lee, J. Lee,—Dynamic dissimilarity measure for support based clusteringl, IEEE Trans. on Knowl. and Data Eng., Vol. 22, No. 6, pp. 900–905, 2010.

[4] R.O. Duda, P.E. Hart, and D.G. Stork. Pattern Classification. John Wiley & Sons, 2001.

[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 ACM 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

[8] 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.

[9] S. Zhong and J. Ghosh, "A Comparative Study of Generative Models for Document Clustering," Proc. SIAM Int'l Conf. Data Mining Workshop Clustering High Dimensional Data and Its Applications,2003.

[10] I.S. Dhillon, "Co-Clustering Documents and Words Using Bipartite Spectral Graph Partitioning," Proc. Seventh ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD), pp. 269-274, 2001.

[11] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. In Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science, pages 238250, Vancouver, Canada, November 2002

[12] Taiping Zhang, Member, IEEE, Yuan Yan Tang, Fellow, IEEE, Bin Fang, Senior Member, IEEE, and Yong Xiang "Document Clustering in Correlation Similarity Measure Space" IEEE Trans. KNOWLEDGE AND DATA ENGINEERING, VOL. 24, NO. 6, JUNE 2012

[13] Duc Thang Nguyen, Lihui Chen, Senior Member, IEEE, and Chee Keong Chan Clustering with Multiviewpoint-Based Similarity Measure IEEE Transactions On Knowledge And Data Engineering, Vol. 24, No. 6, June .

[14] Hamidreza Mirzaei School of Computing Science, Simon Fraser University, Burnaby, Canada hmirzaei@cs.sfu.ca "A Novel Multi-View Agglomerative Clustering Algorithm Based on Ensemble of Partitions on Different Views" 2010 International Conference on Pattern Recognition.

[15] M.F. Porter. An algorithm for suffix stripping. Program, 14(3):130–137, July 1980.

[16] G. Karypis, "CLUTO a Clustering Toolkit," technical report, Dept.of Computer Science, Univ. of Minnesota, http://glaros.dtc.umn.edu/~gkhome/views/cluto, 2003.

[17] Y. Zhao and G. Karypis, "Criterion Functions for Document Clustering: Experiments and Analysis," technical report, Dept. of Computer Science, Univ. of Minnesota, 2002.

[18] Y. Zhao and G. Karypis, "Empirical and Theoretical Comparisons of Selected Criterion Functions for Document Clustering," Machine Learning, vol. 55, no. 3, pp. 311-331, June 2004.

[19] R.O. Duda, P.E. Hart, and D.G. Stork, Pattern Classification, Second ed. John Wiley & Sons, 2001.

[20] G. Karypis, E.H. Han, and V. Kumar. Chameleon: A hierarchical clustering algorithm using dynamic modeling. IEEE Computer, 32(8):68–75, 1999.

.