International Journal of Scientific Engineering and Technology (ISSN : 2277-1581)
Volume No.2, Issue No.9, pp : 861-865 1 Sept. 2013

Document Retrieval using Hierarchical Agglomerative Clustering with Multi-view point Similarity Measure Based on Correlation: Performance Analysis

J. Sankari, Dr. R. Manavalan
Department of Computer Science and Applications
K. S. Rangasamy College of Arts and Science, Tiruchengode**,** India.
[1] jgksankari@gmail.com, [2] manavalan_r@rediffmail.com

Abstract*: Clustering is one of the most interesting and important tool for research in data mining and other disciplines. The aim of clustering is to find the relationship among the data objects, and classify them into meaningful subgroups. The effectiveness of clustering algorithms depends on the appropriateness of the similarity measure between the data in which the similarity can be computed. This paper focus on performance analysis of Agglomerative clustering with Multi Viewpoint based on Cosine similarity and Correlation similarities for finding the relationship between different documents and clustering them. The experiment is conducted over fifteen text documents and the performance of the proposed method is analyzed thoroughly and compared to Hierarchical Agglomerative clustering with Multi Viewpoint that is based on cosine similarity. The experimental results clearly shows that the proposed model Hierarchical Agglomerative clustering with Multi Viewpoint, based on correlation similarity perform quite well for document retrieval.*

Keywords-Hierarchical Agglomerative Clustering, Document retrieval, Multi Viewpoint similarity measure, cosine similarity, correlation similarity.

I. INTRODUCTION

Clustering is used to group fundamental structures in data and classify them into meaningful subgroup for further analysis. It also makes search mechanism too easy and reduces the bulk of operations and computational cost. Many clustering algorithms have been published for developing various techniques and applications [iii] [iv].Text document clustering, groups similar documents to form a cluster, while documents that are different separated apart into different clusters. Accurate clustering requires a precise definition of the closeness between a pair [vi].

A common approach to the clustering problem is to treat it as an optimization process. An optimal partition is found by optimizing a particular function of similarity among data. The set of terms shared between a pair of documents is typically used as an indication of the similarity of the pair. The similarity measure plays a very important role in the success or failure of a clustering method [vii].A hierarchical clustering algorithm creates a hierarchical decomposition of the given set of data objects. Depending on the decomposition approach, hierarchical algorithms are classified as agglomerative (merging) or divisive (splitting). The accuracy of clustering approach is determined based on the similarity or distance measures .A variety of similarity measures have been proposed so far and widely used measures are cosine similarity, The Jaccard coefficient and correlation coefficient.

To improve the accuracy of document clustering, Correlation similarity measure is integrated to Hierarchical Agglomerative Clustering with Multi Viewpoint Similarity Measure. The proposed work is motivated by research of similarity measures in document clustering. Similarity measures play a vital role in clustering the documents. The overview of proposed model is shown in Fig 1.

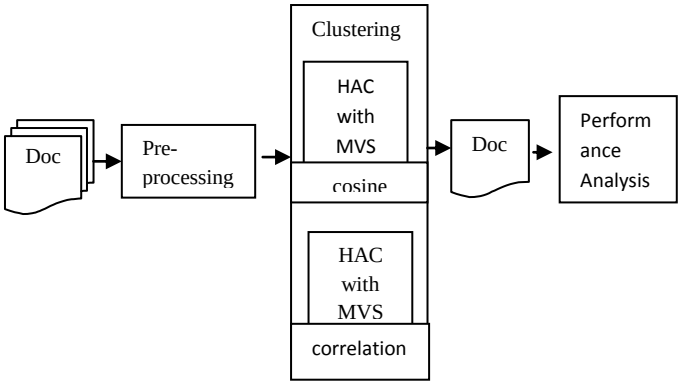

Fig. 1: Overview of Proposed Model

Remaining of this paper is framed as follows: Related work for similarity measures and clustering are reviewed in section ii. Preprocessing and its steps are explained in section iii. In Section iv, Multi Viewpoint clustering based similarity measures is exposed. The Agglomerative clustering with Multi view point based on correlation similarity is discussed in section v. The experiments on fifteen text documents are presented in section vi. Finally, this work is concluded and the extension of the future work is discussed in section vii.

II. RELATED WORK

The fundamental notations for the signify documents and associated concepts are given in Table 1. Each document in a corpus corresponds to an m-dimensional

vector d, where m is the total number of terms. The standard definition of clustering is to organize data objects into separate clusters such that the intra cluster similarity with the inter cluster dissimilarity is maximized. The problem formulation implies that some forms of measurement are desired to determine such similarity or dissimilarity [i].

Table 1: Notations and Descriptions

Notation	Description
n	number of documents
m	number of terms
c	number of classes
k	number of clusters
d	document vector, $\| d \| = 1$
$S = \{ d_1,\ldots,d_n\}$	set of all the documents
S_r	set of documents in cluster r
$D = \sum_{di \in S} di$	composite vector of all the documents
$Dr = \sum_{di \in Sr} di$	Composite vector of cluster r
C = D / n	centroid vector of all the documents
$C_r=D_r/n_r$	centroid vector of cluster r, $n_r = s_r$

2.1 Similarity Measures

Accurate clustering requires a precise definition of the closeness between a pair of documents, in terms of either the pair wise similarity or distance. A variety of similarity or distance measures have been proposed and widely applied, such as cosine similarity, Jaccard coefficient, Euclidean distance and Pearson Correlation Coefficient. The description of Cosine similarity and Correlation similarity is given under here.

2.1.1Cosine Similarity

This metric is used when trying to determine similarity between two documents. 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 [viii] [ix]. The cosine similarity is presented in equation (1).

$$\max \sum_{r=1}^{k} \sum_{d_i \in S_r} \frac{d_i^t c_r}{\| c_r \|} \qquad (1)$$

2.1.2 Pearson Correlation Coefficient

Correlation Clustering, introduced by Bansal, Blum and Chawla [x], provides a method for clustering a set of objects into the best possible number of clusters, without specifying that number in proceed. The most common measure of correlation in statistics is the Pearson Correlation (technically called the Pearson Product Moment Correlation or PPMC), which shows the linear relationship between two variables. Two letters are used to represent the Pearson correlation: Greek letter rho (ρ) for a population and the letter "r" for a sample. Results are always lies between -1 and 1. A result of -1 means that there is a perfect negative correlation between the two variables, while a result of 1 means that there is a perfect positive correlation between the two variables. A result of 0 means that there is no linear relationship between the two variables. The Pearson correlation is presented in the following equation.

$$Pearson(x,y) \frac{\sum xy \frac{\sum x - \sum y}{N}}{\sqrt{(\sum x^2 - \frac{(\sum x)^2}{N})(\sum y^2 - \frac{(\sum y)^2}{N})}} \qquad (2)$$

III. PREPROCESSING

A database consists of massive volume of data which is collected from heterogeneous sources of data. Due to this heterogeneity, real world data tends to be inconsistent and noisy. If data is inconsistent, then there is a possibility that mining process can lead to confusion which may give inaccurate results. In order to extract text which is consistent and accurate text, pre-processing is applied on that data Preprocessing is done in two steps i.e. removal of stop word and stemming [xi].

3.1 Stop word removal

The most common words in any text document do not provide meaning of the documents. Those are prepositions, articles, and pronouns etc. These words are treated as stop words. These words are eliminated. Since these words are not necessary for text mining applications. Any group of words can be chosen as the stop word for a given purpose. This process also reduces the text data and improves the system performance.

3.2 Stemming

Stemming or lemmatization is a technique for the reduction of words into their root. Many words in the English language can be reduced to their base form or stem e.g. agreed, agreeing, disagree, agreement and disagreement belong to agree. The stem is useful, because all other inflections of the root are transformed into the same stem. Case sensitive systems could have another problems when making a comparison between a word in capital letters and another with the same meaning in lower case.

IV. MULTI-VIEWPOINT-BASED CLUSTERING WITH SIMILARITY MEASURE

To construct a concept of Multi viewpoint based similarity [i], it is possible to use more than one point of reference. Similarity between the two documents are defined as

$$similarity(d_i, d_j) = \cos(d_i - 0, d_j - 0) = (d_i - 0)^t (d_j - 0) \quad (3)$$

The Multi View Point Similarity matrix is presented in Fig 2.

Algorithm for Multi-View Point Similarity Matrix

Input: N Number of documents.
Output: MVS Similarity Matrix.

Step 1: Repeat step 2 until $r \leftarrow 1 : c$
Step 2: compute $D_S \backslash S_r \leftarrow \sum d_i \notin S_r^{di}$
$n_S \backslash S_r \leftarrow |S \backslash S_r|$
Step 3: Repeat step 4 until $i \leftarrow 1 : n$
Step 4: Assign class of $d_{i\ to\ r}$
Step 5: Repeat step 6 until $j \leftarrow 1 : n$
Step 6: If $d_j \in S_r$ then calculate a_{ij} using

$$a_{ij} \leftarrow d_i^t d_j - d_i^t \frac{D_S \backslash S_r}{n_S \backslash S_r} - d_j^t \frac{D_S \backslash S_r}{n_S \backslash S_r} + 1$$

otherwise calculate $\mathbf{a_{ij}}$ using

$$a_{ij} \leftarrow d_i^t d_j - d_i^t \frac{D_S \backslash S_r - d_j}{n_S \backslash S_r - 1} - d_j^t \frac{D_S \backslash S_r - d_j}{n_S \backslash S_r - 1} + 1$$

Step 7: Return the similarity Matrix

Fig. 2: Procedure for building MVS matrix

It is used to compute the similarity matrix for given n Number of documents. From the above algorithm, when $\boldsymbol{d_i}$ is considered closer to $\boldsymbol{d_j}$, the $\boldsymbol{d_j}$ can still be considered being closer to $\boldsymbol{d_i}$ as per MVS.

Validity Score Calculation Algorithm

Input: Percentage in the range of 0-1.
Output: Validity Score.

Step1: Repeat step 2 until r ← 1 : c
Step 2: Compute q_r using $q_r \leftarrow [\text{percentage} \times n_r]$
Step 3: If q_r is equl to 0 then $q_r = 1$
Step 4: Repeat step 5 and 6 until i ← 1 : n
Step 5: Assign Sort $\{a_{i1}, \ldots, a_{in}\}$ to $\{a_{iv[1]}, \ldots, a_{iv[n]}\}$
Step 6: Assign permute {1, . . . , n} to {v[1], . . .,v[n]}by using this condition $a_{iv[1]} \geq a_{iv[2]} \geq \ldots \geq a_{iv[n]}$
Step 7: Place class of d_i to r
Step 8: Compute the validity using
validity(d_i) ← $|\{d_{v[1]}, \ldots, d_{v[qr]}\} \cap Sr|$
Step 9: Return the validity.

Fig. 3: The algorithm for validity score calculation.

The validity score is always bounded within 0 and 1.The higher validity score of a similarity measure is more useful for the clustering task. The computational procedure for the validity score is shown in Fig 3.

Incremental Clustering Algorithm which has two major steps such as Initialization and Refinement. It is exposed in Fig 4. At Initialization, k-arbitrary documents are selected where as refinement is a procedure that consists of a number of iterations. During the iteration, the n documents are visited one by one in random order. Each document is checked and if it moves to another cluster it results in the 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.

Incremental Clustering Algorithm

Input: K -Arbitrary documents
Output: Clustering documents

Step 1: Select k arbitrary documents $s_1, \ldots, s_k$ randomly.
Step 2: Cluster the documents by using $cluster[d_i] \leftarrow p = \text{argmax}_r\{s_r^t, d_i\}, \forall\ i = 1, \ldots, n$
Step 3: Calculate D_r using $Dr \leftarrow \sum_{di \in Sr} d_i$, $n_r \leftarrow |S_r|$, $\forall\ r = 1, \ldots, k$
Step 4: Repeat step 5 to 9 until no move for all n documents.
Step 5: Assign random permutation of $\{1, \ldots, n\}$ to $\{v[1 : n]\}$
Step 6: Repeat step 7 to 9 until $j \leftarrow 1 : n$
Step 7: Initialize $i \leftarrow v[j]$, $p \leftarrow cluster[d_i]$
Step 8: Compute $\Delta I_p, q, \Delta I_q$ by using

$$\Delta I_p \leftarrow I(n_p - 1, D_p - d_i) - I(n_p, D_p)$$

$$q \leftarrow \arg \max_{r, r \neq p} \{I(n_r + 1, D_r + d_i) - I(n_r, D_r)\}$$

$$\Delta I_q \leftarrow I(n_q + 1, D_q + d_i) - I(n_q, D_q)$$

Step 9: If $\Delta I_p + \Delta I_Z$ is greater than 0
move d_i to cluster q: $cluster[d_i] \leftarrow q$
and update D_p, n_p, D_q, n_q

Fig. 4: Incremental clustering Algorithm

V. HIERARCHICAL AGGLOMERATIVE CLUSTERING WITH MVS BASED ON SIMILARITY MEASURES

Hierarchical clustering methods are categorized into agglomerative (bottom-up) and divisive (top-down).

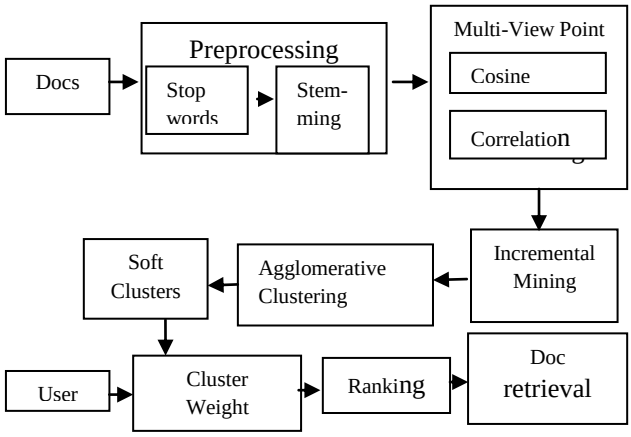

Fig. 5: Block diagram of Agglomerative Clustering with MVS based on Cosine and Correlation similarity Measures

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 block diagram of Agglomerative clustering with MVS based on Cosine and Correlation similarities are shows in fig 5. Each step of this, move towards the two clusters that are the most similar. Thus after each step, the entire number of

clusters is reduced [vii]. The algorithm for Agglomerative clustering is presented in fig 6.

Agglomerative Hierarchical Clustering Algorithm
Input: No of clusters **Output**: Desired Cluster
Step1: Start with N clusters, each containing a single entity, and an N × N symmetric matrix of distances (or similarities) Let d_{ij} = distance between item i and item j. **Step 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 U and V by d_{UV} . **Step 3**: Merge clusters U and V into a new cluster, labelled T. Update the entries in the distance matrix by a. Deleting the rows and columns corresponding to clusters U and V, and b. Adding a row and column giving the distances between the new cluster T and all the remaining clusters. **Step 4**: Repeat steps (2.) and (3.) a total of N-1 times.

Fig. 6: Hierarchical Agglomerative Clustering Algorithm

VI. EXPERIMENTAL ANALYSIS

To substantiate the merits of this proposed method, the experiments are enacted in distinct data sets. The main intention of this work is comparison of cosine similarity and correlation similarity in Agglomerative Clustering with MVS. Fifteen documents from different category are used to analyze the experiment. The collection of fifteen text document is exhibited in Table 2.

Table 2: Documents and its clusters

Common	Sports	Health	Science
Animal.txt Plant.txt Production.txt Lakumi.txt Element.txt	Badminton.txt Basketball.txt Carrom.txt Chess.txt Cricket.txt Tennis.txt	Disease.txt Infection.txt	Chemical.txt ChemicalReaction.txt

These documents are pre-processed by applying stop word removal and stemming. After pre-processing, the similarity is calculated using Cosine similarity and Correlation similarity measure. Table 3 presents the Cosine and correlation similarity for each cluster and the same is flashed in fig 7.

Table 3: Similarity Measure for Clusters

Similarity Measure	Common	Sports	Health	Science
Cosine similarity	0.780	0.944	0.798	0.833
Correlation similarity	0.977	0.857	0.886	0.972

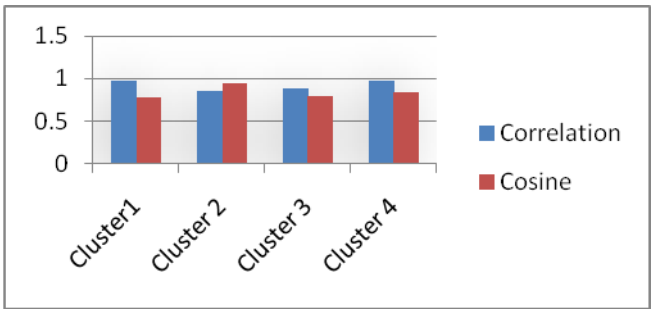

Fig. 7: Performance Analysis of similarity Measures

Agglomerative clustering algorithm is applied for moving dissimilarity document to the applicable cluster. Finally the soft clusters are computed and it is exposed in Table 4.

Table 4: Soft clusters after performing HAC with MVS based Correlation similarity

Cluster 1	Cluster 2	Cluster 3	Cluster 4
Animal.txt Plant.txt Production.txt	Badminton.txt Basketball.txt Carrom.txt Chess.txt Cricket.txt Tennis.txt	Disease.txt Infection.txt Lakumi.txt	Chemical.txt ChemicalReaction.txt Element.txt

The soft clusters for Agglomerative clustering with Multi Viewpoint similarity based on cosine similarity are presented in table 5.

Table 5: Soft clusters after performing HAC with MVS based Cosine similarity

Cluster 1	Cluster 2	Cluster 3	Cluster 4
Badminton.txt Basketball.txt Carrom.txt Chess.txt Cricket.txt Tennis.txt	Animal.txt Production.txt	Disease.txt Infection.txt Lakumi.txt	Chemical.txt ChemicalReaction.txt Plant.txt Element.txt

The average result of Cosine similarity and Correlation similarity in HAC with MVS is presented in table 6 and it is exposed in Fig 8.

Table 6: Average value of MVS based Cosine and Correlation Similarity

MVS based Cosine similarity	MVS based correlation similarity
0.83	0.92

From the computational results and evaluation performance chart, it is observed that the proposed approach performs better. The average value for cosine similarity in

HAC with MVS is 0.83 where as 0.92 for correlation similarity in HAC with MVS. The result shows, Agglomerative clustering with MVS based on correlation similarity is 9% greater than the Agglomerative clustering with MVS based on cosine similarity. The Cosine similarity achieves 83% of the accuracy where as 92% of the accuracy is attain by Correlation similarity.

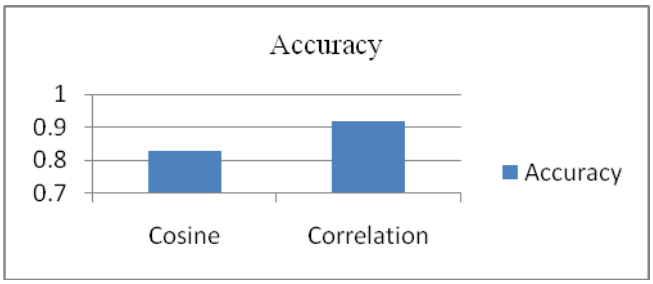

Fig. 8: Performance of HAC with MVS based on Cosine and Correlation similarity

The documents are retrieved from the cluster by applying query which is also used to identify the performance of proposed approach. The sample results for given queries are presented in Table 7. The Accuracy of HAC with MVS based correlation similarity is 9% higher than HAC with MVS based on cosine similarity. It is interesting to note that HAC with MVS based on Correlation similarity produces good results of accuracy. From table 4, it is observed that HAC with MVS based on correlation similarity gives accurate ranking to the documents for given queries.

Table 7: Weight based ranking by query

Query	HAC with MVS based on Cosine		HAC with MVS based on Correlation	
	Weight	Rank	Weight	Rank
Animals	2	1	4	2
Dairy	1	1	1	4
Players	1	1	4	2
Game play	1	1	1	4
Example	1	1	2	3
With	1	1	11	1

Table 7 shows the, Hierarchical Agglomerative Clustering with Multi Viewpoint based correlation similarity is performing better for document retrieval compared to Hierarchical Agglomerative Clustering with Multi Viewpoint based Cosine similarity.

VII. CONCLUSION

In this paper, Cosine similarity and Correlation similarity measures have been constructed and investigated for the task of document retrieval. The experiment is conducted over fifteen text documents. From the computational results, the accuracy of HAC with MVS based on Correlation similarity is compared to HAC with MVS based on Cosine similarity. Hierarchical Agglomerative clustering with Multi Viewpoint based correlation similarity measuring method is analyzed. The experimental results show that Hierarchical Agglomerative clustering with Multi Viewpoint based correlation similarity measure is potentially more suitable for text documents. The experimental result clearly reveals that HAC with MVS based on Correlation similarity produces better accuracy which is 9% higher than HAC with MVS based on Cosine similarity. In future, it would also be possible to apply different similarity measures and different clustering algorithms.

REFERENCES

i. *Duc Thang Nguyen,Lihui Chen and Chee Keong Chan, Senior Member, "Clustering with Multiviewpoint-Based Similarity Measure",IEEE Transactions on Knowledge and Data Engineering,vol 24, No 6, June 2012.*

ii. *Anindya Bhattacharya and Rajat K.De, "Divisive Correlation Clustering Algorithm (DCCA) for grouping of genes:detecting varying patterns in expression profiles," BioInformatics ,volume 24 no,1359-1366,11 2008.*

iii. *Guadalupe J. Torres, Ram B. Basnet, Andrew H. Sung, Srinivas Mukkamala, Bernardete M. Ribeiro"A Similarity Measure for Clustering and Its Applications".*

iv. *Hila Becker "A Survey of Correlation Clustering" Advanced Topics in Computational Learning Theory, May 5, 2005.*

v. *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.*

vi. *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,*

vii. *S. Micciche', F. Lillo AND R. N. Mantegna, "Correlation based Hierarchical clustering in Financial Time Series".*

viii. *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.*

ix. *I. S. Dhillon, —Co-clustering documents and words using bipartite spectral graph partitioning, in KDD, 2001, pp. 269–274.*

x. *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*

xi. *A. Anil Kumar, S.Chandrasekhar, "Text Data Pre-processing and Dimensionality Reduction Techniques for Document Clustering".*