ISSN 2320 - 2602
Volume 2, No.8, August 2013
International Journal of Advances in Computer Science and Technology
Available Online at http://warse.org/pdfs/2013/ijacst05282013.pdf

A HIERARCHICAL DIVISIVE CLUSTERING BASED MULTI-VIEW POINT SIMILARITY MEASURE FOR DOCUMENT CLUSTERING

B.Amuthajanaki[1], K.Jayalakshmi[2]
[1]Research Scholar, Department of Computer Science
Hindusthan College of Arts and Science, Coimbatore, India
janakikks@gmail.com
[2] Assistant professor, Department of Computer Science
Hindusthan College of Arts and Science, Coimbatore,
India, jay_lakme@yahoo.com

ABSTRACT

As we know a cluster is a collection of similar objects situated together and are divergent to other cluster objects. In this manuscript, we establish divisive based Multi-view point clustering that is based on different similarity measures. With multiple viewpoints, more informative measurement of similarity could be accomplished. Two criterion functions for document clustering are proposed based on this new measure they are, inter cluster and intra-cluster relation between objects. The previous clustering process focused on hierarchical clustering of Multi-view point documents, which are not spotlighted on sparse and high dimensional data. The difficulty this manuscript spotlights on is the classical problem of unsupervised clustering of a data-set. Especially, the bisecting divisive clustering approach is here considered. This advance consists in recursively splitting a cluster into two sub-clusters, starting from the main data-set. We evaluated our approach with previous model on a variety of document collections to validate the advantages of our proposed method.

Key Words: Document Clustering, Hierarchical Clustering, Multi-View point Similarity Measure, Divisive Clustering, Clustering methods

1. INTRODUCTION

Clustering collections data into subsets in such a manner that identical instances are collected together, at the same time as different instances belong to different groups. The occurrences are thereby organized into an efficient depiction that characterizes the populace being sectioned. Clustering of entities is as earliest as the human need for describing the salient characteristics of mean and objects and identifying them with a style. Consequently, it squeezes a choice of scientific regulations: from mathematics and statistics to biology and genetics, the entire of which uses different terms to describe the topologies formed using this analysis. As of 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 contained by it. Because clustering is the grouping of similar instances/entities, a number of measures that can choose whether two objects are similar or dissimilar are entailed. Document clustering methods frequently rely on single term analysis of the document data set, such as the Vector Space Model.

To attain more precise document clustering, more informative features including phrases and their weights are particularly important in such scenarios. Document clustering is mainly constructive in many applications such as automatic categorization of documents, grouping search engine results, building taxonomy of documents, and others. The endeavor of clustering is to discover intrinsic structures in data, and organize them into meaningful subgroups for further study and analysis. There have been many clustering algorithms published every year. Existing schemes acquisitively chooses the next frequent item set which represent the next cluster to minimize the overlapping between the documents that contain both the item set and some remaining item sets. One well-liked advance in document clustering is agglomerative hierarchical clustering. Algorithms in these relations 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 establishes with one-point clusters and recursively merges two or more most suitable clusters. The hierarchical clustering is used to launch cluster taxonomy.

Data partitioning is utilized to construct a set of flat partitions called as non-overlapping clusters. Data group is employed to build a set of flat or overlapping clusters but this clustering method is not spotlighted on sparse and high dimensional data. As a result in proposed work in this paper is stimulated by the facts ascertained by investigation of the above. Particularly similarity measures are believed. As of research judgments it is understood that the nature of similarity measured used in any clustering technique has profound impact on the results. The endeavor of the manuscript is to build up a new method that is used to

cluster text documents that have sparse and high dimensional data objects. Subsequently we originate new clustering criterion functions and corresponding clustering algorithms respectively. Divisive algorithms initiated with just only one cluster that contains all sample data. After that, the single cluster splits into two or more clusters that have higher dissimilarity between them until the number of clusters becomes number of samples or as specified by the user.

The most important work is to build up a novel hierarchal algorithm for document clustering which provides maximum efficiency and performance. It is mainly spotlighted in studying and making use of cluster overlapping phenomenon to design cluster merging criteria. Recommending a new method to compute the overlap rate in order to improve time efficiency and "the veracity" is mainly concentrated. Multi-view learning algorithms characteristically assume a complete bipartite mapping between the different views in order to exchange information during the learning process. The remaining of this paper is ordered as follows: In section 2, we review related literature on similarity and clustering of documents. We then present our proposed system methods in section 3. Extensive experiments on real world benchmark data sets are presented and discussed in Sections 4. Finally, the conclusions and future work are given in Section 5.

2. RELATED WORK

Document clustering has been examined for utilize in a number of different areas of text mining and information retrieval. At first, document clustering was explored for improving the precision or recall in information retrieval systems [3] and as an efficient way of finding the nearest neighbors of a document [4]. More recently, clustering has been suggested for exploit in browsing a collection of documents [5] or in organizing the results returned by a search engine in response to a user's query [6]. Document clustering has also been utilized to mechanically generate hierarchical clusters of documents [7]. Agglomerative hierarchical clustering and K-means are two clustering techniques that are usually exercised for document clustering. Agglomerative hierarchical clustering is frequently portrayed as "better" than K-means, although slower. An extensively known revision, discussed in [8], indicated that agglomerative hierarchical clustering is superior to K-means, although we stress that these results were with non-document data. In the document field, Scatter/Gather [5], a document browsing system based on clustering, uses a hybrid approach involving both K-means and agglomerative hierarchical clustering.

K-means is employed because of its efficiency and agglomerative hierarchical clustering is used because of its quality. Modern exertion to generate document hierarchies [9] uses some of the clustering techniques from [5] and presents a result that indicates that agglomerative hierarchical clustering is better than K-means, although this result is just for a single data set and is not one of the major consequences of the manuscript. To begin with we also believed that agglomerative hierarchical clustering was superior to K-means clustering, especially for building document hierarchies, and we sought to find new and better hierarchical clustering algorithms. Still, throughout the course of our experiments we discovered that a simple and efficient variant of K-means, "bisecting" K-means, can produce clusters of documents that are better than those produced by "regular" K-means and as good as or better than those produced by agglomerative hierarchical clustering techniques. Our investigational results also showed that divisive algorithms always generate better hierarchical clustering solutions by repeated bisection than agglomerative algorithms for all the criterion functions. The experiential superiority of divisive algorithms suggests that divisive clustering algorithms are well-suited for clustering large document datasets due to not only their relatively low computational requirements, but also comparable or better clustering performance.

3. PROPOSED METHOD

Divisive algorithms commence with just only one cluster that contains all sample data. After that, the single cluster splits into two or more clusters that have higher dissimilarity between them until the number of clusters becomes number of samples or as specified by the user. There are two main criterion is determined. First is Intra-Cluster Similarity Technique (IST): This hierarchical technique seems at the similarity of all the documents in a cluster to their cluster centric and is defined by $\mathbf{Sim(X)} = \sum_{\mathbf{d}\in\mathbf{X}} \mathbf{cosine(d,c)}$, where d is a document in cluster, X, and c is the centric of cluster X. The predilection of which pair of clusters to merge is made by determining which pair of clusters will escort to smallest decrease in similarity. As a result, if cluster Z is formed by merging clusters X and Y, then we select X and Y so as to maximize $\mathbf{Sim(Z) - (Sim(X) + Sim(Y))}$ that is non-positive. Second is Intra cluster Similarity Technique: This hierarchical method defines the similarity of two clusters to be the cosine similarity between the centroids of the two clusters. It explains the cluster relationship as follows, $\mathbf{similarity\ (c1,c2)} = \frac{\sum_{\mathbf{d_1}\in\mathbf{c1}} \mathbf{cosine(d_1,d_2)}}{\mathbf{d_2}\in\mathbf{c2}} / \mathbf{S(c1)\times S(c1)}$ where d_1 and d_2 is, documents, respectively, in cluster1 and cluster2.

Hierarchical divisive is a top-down technique of clustering which generates clusters by sub-dividing the single cluster containing the entire network at first. To build a new concept of similarity, it is feasible to use more than one point of reference. We define similarity between the two documents as $\mathbf{Sim_{d_i,d_j\in S_r}} = \frac{1}{\mathbf{n-n_r}} \sum_{\mathbf{d_h}\in\mathbf{S\setminus S_r}} \mathbf{Sim(d_i - d_h, d_j - d_h)}$. Seeing that explained by the above equation, similarity of two documents $\mathbf{d_i}$ and $\mathbf{d_j}$, 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 entities to be computed must be in the same cluster, while the points from where to establish this measurement must be outside of the cluster. We describe this proposal the Multi viewpoint-based Similarity, or MVS. Intended for this MVS Clustering functional in our proposed system, we will denote the proposed similarity measure between two document vectors d_i and d_j by $\mathbf{MVS}(\mathbf{d_i}, \mathbf{d_j} | \mathbf{d_j} \in \mathbf{S_r})$, or occasionally $\mathbf{MVS}\ (\mathbf{d_i}, \mathbf{d_j})$ for short. The ultimate form of MVS depends on particular formulation of the individual similarities inside the sum. If the proportional similarity is defined by dot-product of the distinction vectors, we have

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

$$= \frac{\mathbf{1}}{\mathbf{n} - \mathbf{n_r}} \sum_{\mathbf{d_h}} \mathbf{cos}(\mathbf{d_i} - \mathbf{d_h}, \mathbf{d_j} - \mathbf{d_h}) \|\mathbf{d_i} - \mathbf{d_h}\| \|\mathbf{d_j} - \mathbf{d_h}\|$$

The relationship between two points d_i and d_j 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 d_i and d_j looking from d_h and the correlation distances from d_h to these two points. It is able to be seen that this technique offers more informative assessment of similarity than the single origin point-based similarity measure. In our projected technique, 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 structural design of the Divisive MVS is shown in the Figure.1. Set of documents are in use as input from the user, then each block performs the operations on the documents to form the final hierarchical divisive clustering.

Preprocessing is completed by two steps they are removal of stop words and stemming. Stop-words are very ordinary 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 procedure of converting different forms of a word into one canonical form. Words like “compute”, “computing”, “computer” is all changed to a single word “compute”. This is essential to keep away from treating different variations of a word manifestly. Word stemming was done using the popular Porter stemming algorithm. The two appraises are used to make sure the similarity and dissimilarity between the documents. With using the multiple measure point, we will obtain most appropriate clustered documents. The two measures we used are cosine similarity and correlation similarity. The associations between vectors A and B are defined as follows:

$$\mathbf{r}(\mathbf{A}, \mathbf{B}) = \frac{\frac{\mathbf{1}}{\mathbf{n}} \sum_i \mathbf{A_i} \mathbf{B_i} - \boldsymbol{\mu}\mathbf{A}\boldsymbol{\mu}\mathbf{B}}{\boldsymbol{\sigma}\mathbf{A}\boldsymbol{\sigma}\mathbf{B}}$$

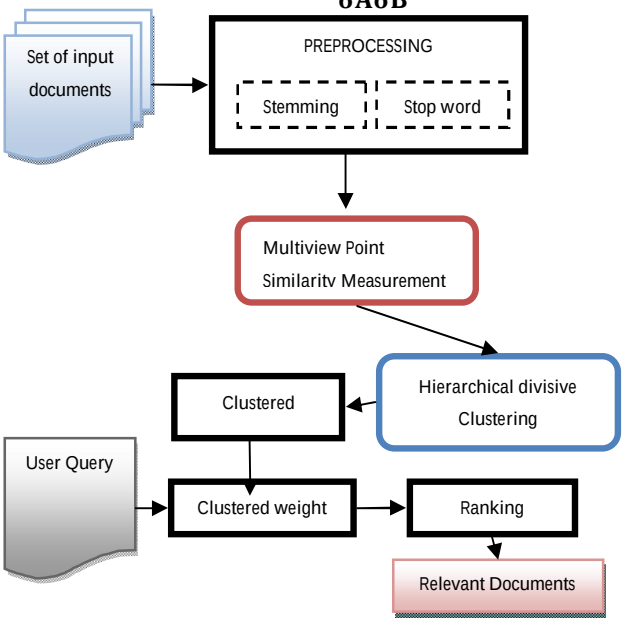

Figure 1: Proposed Architecture for Hierarchical Divisive MVS

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

$$\mathbf{a}.\mathbf{b} = ||\mathbf{a}||.||\mathbf{b}||\mathbf{cos}\boldsymbol{\theta}$$

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

$$\mathbf{Similarity} = \mathbf{cos}(\boldsymbol{\theta}) = \frac{\mathbf{A}.\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{\mathbf{i}=\mathbf{1}} \mathbf{A_i} * \mathbf{B_i}}{\sqrt{\sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{n}} (\mathbf{A_i})^2} \sqrt{\sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{n}} (\mathbf{B_i})^2}}$$

4. EXPERIMENTAL RESULTS AND DISCUSSION

The performance is examined based on the running time needed to execute agglomerative and divisive algorithm depending on the nature of the field and the number of records. As of the database, one field in each type of data is taken for assessment. Consequently, for binary data type sex field is selected, for numeric type age field is selected and for string, province field is selected. Additionally, two fields are combined together and the performance of the algorithm is compared as a special category. For that Sex and Injured/Dead fields are selected. FScore is an evenly weighted combination of the precision (P) and recall (R) values used in information retrieval. Lastly, Accuracy determines the fraction of documents that are correctly labels, assuming a one-to-one correspondence between true classes and assigned clusters.

4.1 Precision Rate

It is shown from figure 2 that the precision rate for clustering the database based on a binary field using the agglomerative algorithms are more or less equal and the precision rate decreases as the size of the database increases. However, divisive algorithm produces high precision rate than agglomerative algorithms when the size of the database increases. In that figure X axis represents the datasets and Y axis represents the precision rate.

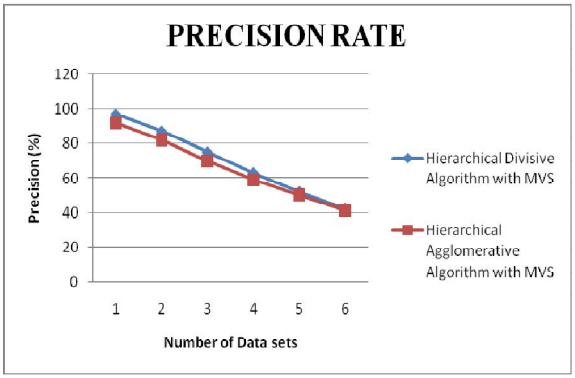

Figure 2: Precision rate

4.2 Recall Rate

It is shown from figure 2 that the recall rate for clustering the database based on a binary field using the agglomerative algorithms are more or less equal and the recall rate decreases as the size of the database increases. However, divisive algorithm produces high recall rate than agglomerative algorithms when the size of the database increases. In that figure X axis represents the datasets and Y axis represents the recall rate.

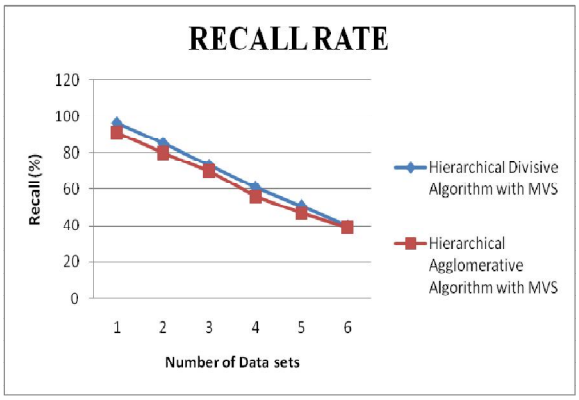

Figure 3: Recall Rate

4.3 F-Score

It is shown from figure 2 that the F-Score rate for clustering the database based on a binary field using the agglomerative algorithms are more or less equal and the F-Score rate decreases as the size of the database increases. However, divisive algorithm produces high F-Score rate than agglomerative algorithms when the size of the database increases. In that figure X axis represents the datasets and Y axis represents the F-Score rate.

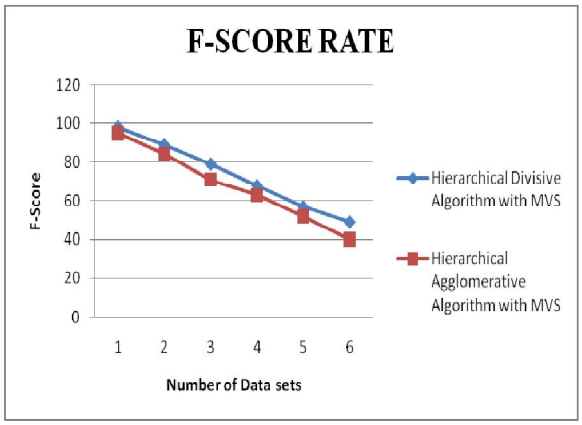

Figure 4: F-Score Rate

4.4 Accuracy Rate

It is shown from figure 2 that the accuracy rate for clustering the database based on a binary field using the agglomerative algorithms are more or less equal and the accuracy rate decreases as the size of the database increases. However, divisive algorithm produces high accuracy rate than agglomerative algorithms when the size of the database increases. In that figure X axis represents the datasets and Y axis represents the accuracy rate.

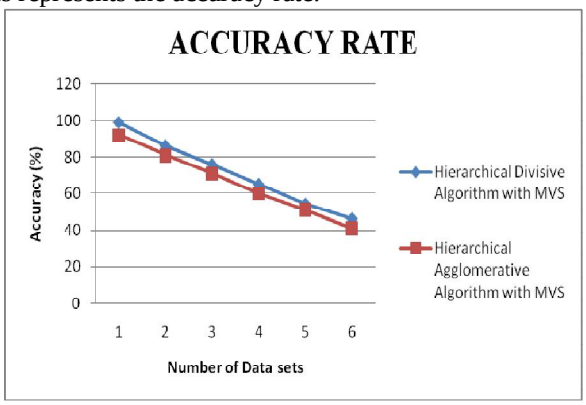

Figure 5: Accuracy Rate

5. CONCLUSION

This paper evaluates the performance of agglomerative and divisive algorithm for various data types. As of this effort it is found that the divisive algorithm works as twice as fast as that of agglomerative algorithm. It is also initiated that the time required for string data type is high when compared to the other. It is as well found that the running time get increased on an average of six times when the number of records get twice over. Furthermore the run time for all the agglomerative algorithms for same type of data and for same amount of records is more or less equal. Evaluated with other state-of-the-art clustering techniques

that use different types of similarity measure, on a large number of document datasets and under different evaluation metrics, the proposed algorithms show that they could give significantly improved clustering performance. The main contribution of this paper is the fundamental concept of similarity measure from multiple viewpoints. Future techniques could formulate employ of the same principle, but define alternative forms or the relative similarity.

REFERENCES

[1] C. J. van Rijsbergen, "**Information Retrieval**", Buttersworth, London, second edition(1989),.

[2] Chris Buckley and Alan F. Lewit,' **Optimizations of inverted vector searches**", SIGIR '85, Pages 97-110, 1985.

[3] Douglass R. Cutting, David R. Karger, Jan O. Pedersen, and John W. Tukey, "**Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections**", SIGIR '92, Pages 318 – 329, 1992.

[4] Oren Zamir, Oren Etzioni, Omid Madani, Richard M. Karp, "**Fast and Intuitive Clustering of Web Documents**", KDD '97, Pages 287-290, 1997.

[5] Daphe Koller and Mehran Sahami, "**Hierarchically classifying documents using very few words**", *Proceedings of the 14th International Conference on Machine Learning (ML)*, Nashville, Tennessee, July 1997, Pages 170-178.

[6] Richard C. Dubes and Anil K. Jain, "**Algorithms for Clustering Data**", Prentice Hall, 1988.

[7] Bjorner Larsen and Chinatsu Aone, "**Fast and Effective Text Mining Using Linear-time Document Clustering**", KDD-99, San Diego, California, 1999.

[8] Sergio M. Savaresi, Daniel L. Boley, Sergio Bittanti and Giovanna Gazzaniga,"**Cluster selection in divisive clustering algorithms**", published on 2002