**Available Online at www.ijcsmc.com**

**International Journal of Computer Science and Mobile Computing**

**A Monthly Journal of Computer Science and Information Technology**

**ISSN 2320–088X**

***IJCSMC, Vol. 3, Issue. 5, May 2014, pg.871 – 874***

**SURVEY ARTICLE**

# Survey of Document Clustering

**Prof. Hetal Gaudani[1], Khushboo Lakhani[2], Riten Chhatrala[3]**

[1]Associate Professor, G H Patel College of Engineering & Technology, Gujarat, India
[2]Student, G H Patel College of Engineering & Technology, Gujarat, India
[3]Student, G H Patel College of Engineering & Technology, Gujarat, India
[1] hetalgaudani@gcet.ac.in; [2] khushbooml28@gmail.com; [3] ritenchhatrala2@gmail.com

***Abstract— This paper presents the results of an experimental study of common known document clustering algorithms. In essence, there are two main approaches to document clustering. They are agglomerative hierarchical clustering and K-means. (For K-means there are a "standard" K-means algorithm and a variant of K-means, "bisecting" K-means in which K-means is repeated for some finite number of times). Hierarchical clustering, often graphed as the better quality clustering approach, is limited because of its quadratic time complexity. In contrast, K-means and its variant (bisecting K-means) have a time complexity which is linear in the number of documents, but are considered to produce inferior clusters. However, our results indicate that the bisecting K-means approach is better than the standard K-means approach and as good as or better than the hierarchical approaches that we tested for a variety of clusters.***

***Keywords— clustering; document; hierarchical; partitional***

## I. INTRODUCTION

The World Wide Web and the number of other text documents maintained in organizational intranets continue to grow at a surprisingly great speed. Therefore, accessing, managing, browsing and searching large repositories of these text documents require efficient organization of the information present. In dynamic information environments like the World Wide Web, it is usually desirable to apply adaptive methods for document organization such as "clustering".[1]

The topic of clustering has been extensively studied in many scientific disciplines and over the years a variety of different algorithms have been portrayed. These algorithms can be classified based on their underlying methodology, as either agglomerative or partitional algorithms.[2]

Agglomerative hierarchical clustering is often graphed as "better" than K-means, although slower. K-means is used because of its efficiency and agglomerative hierarchical clustering is used because of its quality. Initially we also surmised that agglomerative hierarchical clustering was superior to K-means clustering, as it builds document hierarchies. However, experiments indicated that a simple and efficient variant of K-means, i.e., "bisecting" K-means, can generate 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.[3]

## II. CLUSTERING TECHNIQUE

This section provides a brief overview of hierarchical and partitional (K-means) clustering techniques.

Basically there are two approaches to generate a hierarchical cluster: Hierarchical techniques produce a nested sequence of partition, all inclusive cluster at the top and singleton clusters of individual points of cluster at the bottom level. Each intermediate level can be viewed as combining two clusters from the next lower level or splitting a cluster from the next higher level. The results of a hierarchical clustering algorithm can be graphed as tree, also known as a dendogram. This tree graphically displays the merging process and the intermediate clusters. The dendogram at the shows how six points can be merged into a single cluster.

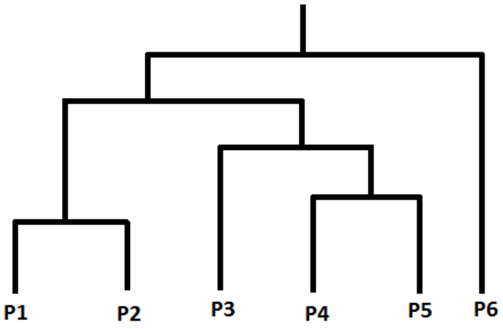

Fig. 1.Shows how six points can be merged into a single cluster

There are two basic approaches to generate a hierarchical cluster:

A. **Agglomerative:** Start with the points as individual clusters at the bottom and, at each step, combine the most similar clusters. This requires a definition of cluster similarity or distance.

B. **Divisive:** Start with one, all-inclusive cluster at the top and, at each step, split a cluster until only singleton clusters of individual points remain. In this case, a decision has to be made as to which cluster to split and how to perform that split at each level.

There are some incremental algorithms that update their cluster hierarchy when new documents arrive, such as DC-tree and IHC. They are based on a tree structure and obtain disjoint set of document hierarchies. On the other hand, there are several static hierarchical algorithms used overlapped clustering of documents, including HFTC, Malik's method and HSTC.

The agglomerative algorithm comprises two specific techniques: Dynamic Hierarchical Compact and Dynamic Hierarchical Star. The former creates disjoint hierarchies of clusters, while the latter produces hierarchies that are overlapped. The experimental results on several benchmark text collections show that these methods are not only suitable for yielding hierarchical clustering solutions in dynamic environments effectively and efficiently, but they also offer hierarchies easier to browse than the traditional algorithms. [1]

Unlike hierarchical techniques, partitional clustering algorithms create a one-level (un-nested) partitioning of the data points. If the desired number of clusters is K, then partitional approach typically finds all the K clusters at once. Unlike K-means, traditional hierarchical schemes bisect a cluster to get two clusters or combine two clusters to get one. Of course, a hierarchical approach is used to generate a flat partition of K clusters, and similarly, the iterative application of a partitional technique can provide a hierarchical clustering. The variant of K-Means, i.e., bisecting K-means algorithm that we present later is such an approach.

Out of a number of partitional techniques, we shall only describe the K-means technique which is widely used in document clustering. K-means is based on the idea that a centre point, which can represent the whole cluster. In essence, in K-means we use the notion of a centroid, which can be defined as the mean or median point of a group of points. [3]

## III. HIERARCHICAL AGGLOMERATIVE CLUSTERING ALGORITHMS

Hierarchical agglomerative algorithms find the clusters by initially assigning each object to its own cluster and then repeatedly combining pairs of clusters until a certain stopping criterion is met. [2]

β-similarity is the undirected graph whose vertices are the clusters and there is an edge from vertex i to vertex j, if the cluster j is β-similar to the cluster i. Two clusters are said to be a β-similar if their similarity is greater or equal to β, where β is a user-defined parameter. Certainly, i is said to be a β-isolated cluster if its similarity with cluster j is lesser than β.The clustering algorithms based on graphs involve two main tasks: the construction of a certain graph and a cover routine of this graph that determines the clusters. In this context, a cover for a graph G = (V, E) is a collection $V_1$, $V_2$..... $V_k$ of subsets of V each representing a different cluster.

Our hierarchical clustering algorithm is an agglomerative method based on graph too. Multi-layered clustering is used by it to produce the hierarchy. The granularity increases as the level of the hierarchy decreases, with the top layer being the most general while the leaf nodes being the most specific. At each successive level of the hierarchical cluster, vertices represent subsets of their parent clusters. The process for each layer has two steps: the construction of a graph and a cover routine of this graph. The general framework of dynamic hierarchical agglomerative is shown in Figure 1.

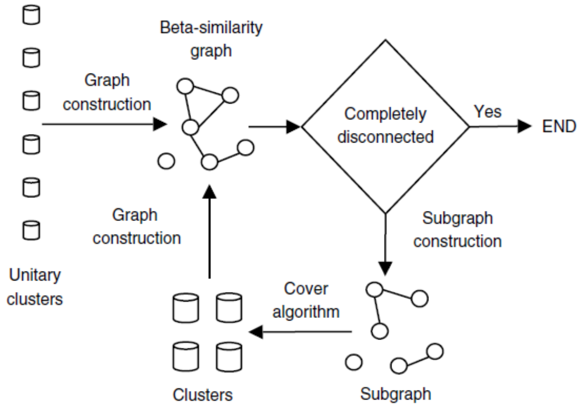

Fig. 2. Dynamic hierarchical agglomerative framework [4]

In our framework, a similarity measure to compare various objects and an inter cluster similarity measure are required. The algorithm starts with each considering each object as a cluster. After that it constructs a sub graph of the β-similarity graph. The set of vertices of this sub graph must equal to the set of vertices of the graph. A cover routine is applied to this sub graph to build the clusters at bottom layer. From the obtained hierarchical clusters, this algorithm constructs a new β-similarity graph and its corresponding sub graph. Then, the cover routine is applied again to obtain the clusters in the next layer. This process is repeated until the β-similarity graph is completely disconnected, i.e., all vertices (clusters) of the graph are β-isolated. In order to ensure the order independence of the framework the cover routine should not depend on the order of the incoming objects. Note that we use the same β value and a unique sub graph type in all levels of the hierarchy. [4]

### A. Simple Agglomerative Clustering Algorithm

1. Compute the similarity between all the pairs of clusters, i.e., calculate a similarity matrix whose ij$^{th}$ entry in matrix gives the similarity between the i$^{th}$ and j$^{th}$ pair ofclusters.
2. Combine the most similar two clusters.
3. Update the similarity matrix to reflect the pairwise similarity between the new cluster and the original clusters.
4. Repeat steps 2 and 3 until only a single cluster remains. [3]

## IV. PARTITIONAL CLUSTERING ALGORITHMS

Partitional clustering algorithms compute a k-way clustering of a set of documents. It is done either directly or via a sequence of repeated bisections. A direct k-means clustering is commonly computed as follows. Initially, a bunch of k documents is selected from the collection to act as the seeds of the k required clusters. Then, the similarity for each document to these k seeds is computed. Based on these similarities it is assigned to the cluster corresponding to its most similar seed. This can be said as the initial k-way clustering. Repeated refinement then optimizes the desired clustering criterion function. By recursively applying the above algorithm

2-way clustering can be computed. Initially, the documents are partitioned into two clusters, and then one of these clusters is selected and is further bisected, and so on. This process continues k -1 times, giving k clusters. The key step in this algorithm is the method used to select which cluster to bisect next. Conventionally the largest cluster is selected as it leads to reasonably good and balanced clustering solutions. Experiments show that the clustering solutions obtained via repeated bisections are comparable or better than those produced via direct clustering. As they have to solve a simpler optimization problem at each step, their computational requirements are much smaller.[2]

The basic K-means clustering algorithm steps are as shown below.

A. Basic K-means Algorithm for finding K clusters:
   1. Select K points as the initial centroids.
   2. Assign all points to the closest centroid.
   3. Recompute the centroid of each cluster.
   4. Repeat steps 2 and 3 until the centroids don't change. [3]

B. Basic Bisecting K-means Algorithm for finding K clusters:

   1. Pick a cluster to split.
   2. Find 2 sub-clusters using the basic K-means algorithm. i.e., repeat the Bisecting step
   3. Repeat step 2, the bisecting step, for j times (j is the number of iterations specified) and take the split that produces the clustering with the highest overall similarity.
   4. Repeat steps 1, 2 and 3 until the desired number of clusters is reached. [3]

Note that the bisecting K-means technique can produce either an un-nested (flat) clustering or a hierarchical clustering. We have to "refine" the clusters using the basic K-means algorithms for un-nested clusters, but it is not required for the nested clusters. Strictly speaking, the bisecting K-means algorithm is a divisive hierarchical clustering algorithm. The bisecting K-means has a time complexity which is linear in the number of documents. In case the number of clusters is large and refinement is not to be used, then bisecting K-Means is even more efficient than the regular K-means algorithm. [3]

## CONCLUSION

As K-means algorithm uses a top-down approach it is more efficient than hierarchical clustering algorithms as it uses a bottom-up approach. Also the speed and simplicity of K-means has an upper hand over hierarchical clustering algorithms. But K-means has a disadvantage that if initial centroids are not good whole cluster can be affected. But this can be overcome by bisecting K-means algorithm.

## REFERENCES

*[1] Dynamic hierarchical algorithms for document clustering Reynaldo Gil-García *, Aurora Pons-Porrata Center for Pattern Recognition and Data Mining, Universidad de Oriente, Santiago de Cuba, Cuba.*

*[2] Comparison of Agglomerative and Partitional Document Clustering Algorithms Ying Zhao and George Karypis Department of Computer Science, University of Minnesota, Minneapolis, MN 55455 fyzhao, karypisg@cs.umn.edu*

*[3] A Comparison of Document Clustering Techniques Michael Steinbach George Karypis Vipin Kumar Department of Computer Science and Engineering, University of Minnesota Technical Report #00-034 {steinbac, karypis, kumar@cs.umn.edu}.*

*[4] Dynamic Hierarchical Compact Clustering Algorithm Reynaldo Gil-Garc´ýa1, Jos´e M. Bad´ýa-Contelles, and Aurora Pons-Porrata Center of Pattern Recognition and Data Mining, Universidad de Oriente, Santiago de Cuba, Cuba {gil, aurora}@app.uo.edu.cu Universitat Jaume I, Castell´on Spain badia@icc.uji.es*