IOSR Journal of Engineering (IOSR JEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
PP 34-39

A Study of Hierarchical Clustering Techniques

Susan Tony[1], Pooja Sonawane[2],Ruchi Chauhan[3],Shikha Malik[4]
[1](EXTC, Atharva college of Engineering/ Mumbai University, India)
[2](EXTC, Atharva College of Engineering/ Mumbai University, India)
[3](EXTC, Atharva College of Engineering/ Mumbai University, India)
[4](EXTC, Atharva College of Engineering/ Mumbai University, India)

Abstract: *Clustering algorithms classify data points into meaningful groups based on their similarity to exploit useful information from data points. They can be divided into categories: Hierarchical clustering and Partition clustering algorithms, Clustering algorithms based on cost function optimization and others. In this paper, we discuss some hierarchical clustering algorithms and their attributes.*

Keywords: *Agllomerative, Divisive, Dendrogram*

I. Introduction

Clustering is a data mining technique to group the similar data into a cluster and dissimilar data into different clusters. Clustering is the unsupervised classification of data into groups/clusters [1] [22] (observations, data items, or feature vectors) into groups (clusters). A clustering algorithm partitions a data set into several groups such that the similarity within a group is larger than among groups. Moreover, most of the data collected in many problems seem to have some inherent properties that lend themselves to natural groupings [24]. It is widely used in biological and medical applications, computer vision, robotics, geographical data, and so on [2].

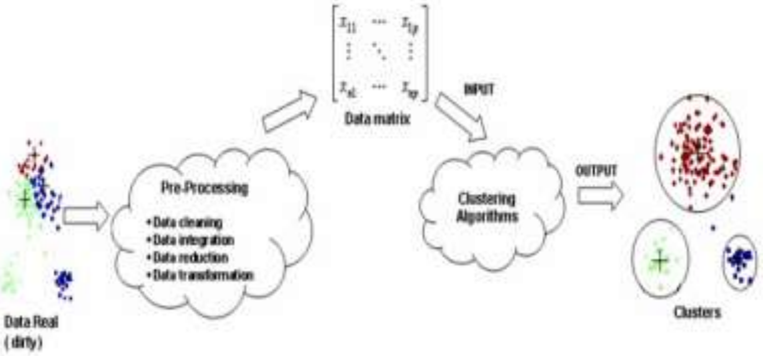
Clustering is the process of organizing objects into groups whose members are similar in some way. A cluster is therefore a collection of objects which are similar between them and are dissimilar to the objects belonging to other clusters. Clustering algorithms are used extensively not only to organize and categorize data, but are also useful for data compression and model construction.

To date, many clustering algorithms have been developed. They can organize a data set into a number of groups /clusters. Clustering algorithms may be divided into the following major categories: Sequential algorithms, Hierarchical clustering algorithms, Clustering algorithms based on cost function optimization and etc [3]. In this paper, we only focus on hierarchical clustering algorithms. At first, we introduce some hierarchical clustering algorithms.

II. Clustering process

Clustering is the unsupervised classification of data into groups/clusters [1]. The input for a system of cluster analysis is a set of samples and a measure of similarity (or dissimilarity) between two samples. The output from cluster analysis is a number of groups /clusters that form a partition, or a structure of partitions, of the data set (Fig. 1)The ultimate goal of clustering can be mathematically described as follows [22]:
X=C1….. Ci U Cn; $Ci \cap Cj = \phi(i \neq j)$ Where X denotes the original data set, Ci, Cj are clusters of X, and n is the number of clusters [5].

Figure no 1 Clustering Process



III. Clustering Algorithms

Clustering algorithms can be broadly classified into the following major categories [3] [22]:

Sequential Algorithms:These algorithms produce a single clustering. They are quite straightforward and fast methods. In most of them, all the feature vectors are presented to the

Algorithm once or a few times (typically no more than five or six times). The final result is, usually, dependent on the order in which the vectors are presented to the algorithm.

Hierarchical Clustering:These schemes are further divided into:

- Agglomerative Algorithm (Bottom-up,Merging) These algorithms produce a sequence of clustering of decreasing number of clusters, *m*, at each step. The clustering produced at each step results from the previous one by merging two clusters into one.
- Divisive Algorithm (Top-down, Splitting)

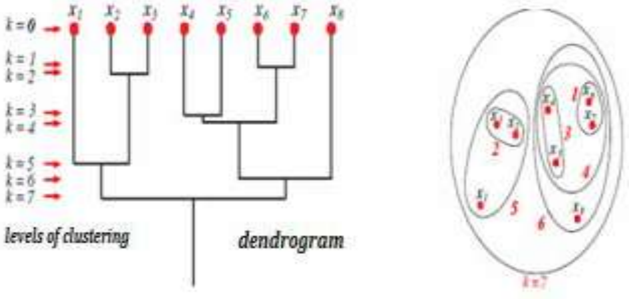
that is, they produce a sequence of clustering of increasing m at each step. The clustering produced at each step results from the previous one by splitting a single clusters into two.

Clustering Algorithms Based on Cost Function Optimization: This category contains algorithms in which“sensible” is quantified by a cost function, J, in terms of which a clustering is evaluated. Usually, the number of clusters m is kept fixed. Most of these algorithms use differential calculus concepts a produce successive clustering while trying to optimize J. Algorithms of this category are also called iterative function optimization schemes. This category includes: Hard or crisp clustering algorithms, Probabilistic clustering algorithms, Fuzzy clustering algorithms, Probabilistic clustering algorithms and Boundary detection algorithms

Hierarchical clustering Algorithm

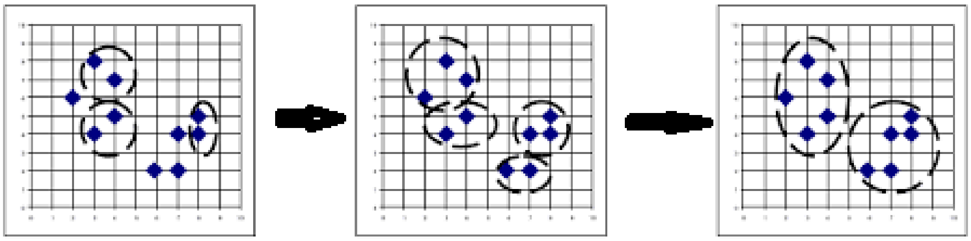
Hierarchical clustering is a method of cluster analysis which seeks to build a hierarchy of clusters. The hierarchical methods group training data into a tree of clusters. This tree structure called dendrogram (Fig. 2). It represents a sequence of nested cluster which constructed top-down or bottom-up. The root of the tree represents one cluster, containing all data points, while at the leaves of the tree, there are n clusters, each containing one data point. By cutting the tree at a desired level, a clustering of the data points into disjoint groups is obtained [6] [22].

Figure no 2. Tree structure of training data (dendrogram)



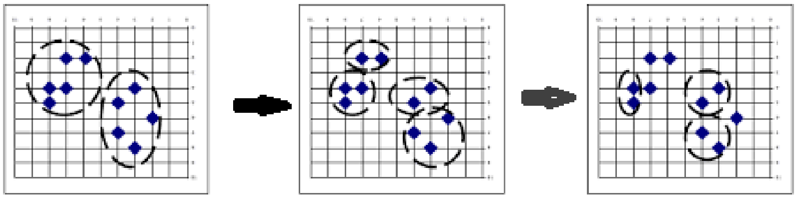
Hierarchical clustering algorithms divide into two categories: Agglomerative and Divisive.Agglomerative clustering executes in a bottom–top fashion, which initially treats each data point as a singleton cluster and then successively merges clusters until all points have been merged into a single remaining cluster (Fig 3).

Figure no 3. Agglomerative Hierarchical Clustering



Divisive clustering, on the other hand, initially treats all the data points in one cluster and then split them gradually until the desired number of clusters is obtained(Fig.4)

Figure no 4. Divisive Hierarchical Clustering



The quality of a pure hierarchical clustering method suffers from its inability to perform adjustment, once a merge or split decision has been executed. Then it will neither undo what was done previously, nor perform object swapping between clusters. Thus merge or split decision, if not well chosen at some step, may lead to some-what low-quality clusters. One promising direction for improving the clustering quality of hierarchical methods is to integrate hierarchical clustering with other techniques for multiple phase clustering [23][25].Many agglomerative clustering algorithms have been proposed, such as CURE, ROCK, CHAMELEON, BIRCH, single-link, complete-link, average-link,Leaders-Subleaders. One representative divisive clustering algorithm is the bisecting k-means method. So in this paper, we describe a few improved hierarchical clustering algorithms that overcome the limitations that exist in pure hierarchical clustering algorithms.

Specific Algorithms

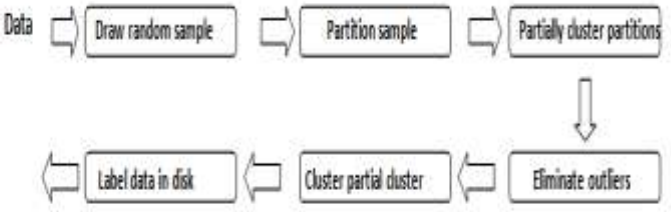
We focus on hierarchical clustering algorithms. At first, we introduce these algorithms in subsections and then compare them by different criteria

CURE (Clustering Using REpresentatives)

In this section, we present CURE's agglomerative hierarchical clustering algorithm[27]. It first partitions the random sample and partially clusters the data points in each partition. After eliminating outliers, the pre clustered data in each partition is then clustered in a final pass to generate the final clusters. Fig 5 is an overview of CURE [22]. CURE algorithm salient features are:

(1) the clustering algorithm can recognize arbitrarily shaped clusters (e.g., ellipsoidal)
(2) the algorithm is robust to the presence of outliers,
(3) the algorithm uses space that is linear in the input size n and has a worst-case time complexity of O(n2 logn). For lower dimensions (e.g., two), the complexity can be shown to further reduce to O(n2). (4) It appropriate for handling large data sets [9].

Figure no 5.CURE process



There is a developed algorithm denoted as CURE that combines centroid and single linkage approaches by choosing more than one representative point from each cluster. At each step of the algorithm, the two clusters with the closest pair of representative points (one in each cluster) are merged [10].

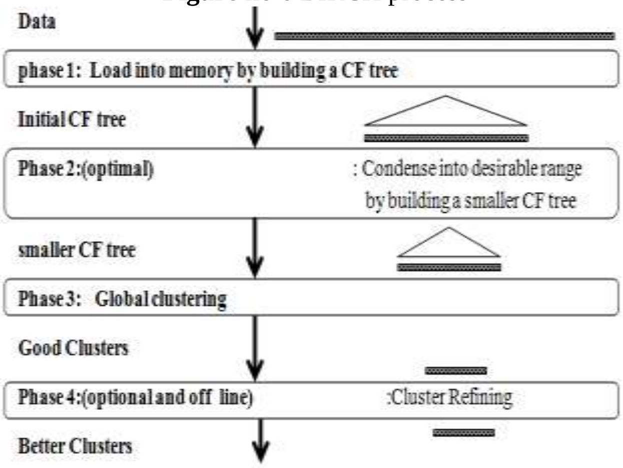
BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)

BIRCH [22] is an agglomerative hierarchical clustering algorithm proposed by Charikar et al. in 1997[11]. It is especially suitable for very large databases. This method has been designed so as to minimize the number of I/O operations [3].

BIRCH incrementally and dynamically clusters incoming multi-dimensional metric data points to try to produce the best quality clustering with the available resources (i. e., available memory and time constraints). BIRCH can typically find a good clustering with a single scan of the data, and improve the quality further with a few additional scans.

BIRCH is also the first clustering algorithm proposed in the database area to handle "noise" (data points that are not part of the underlying pattern) effectively. Fig 6 presents the overview of BIRCH [12].

Figure no 6 BIRCH process

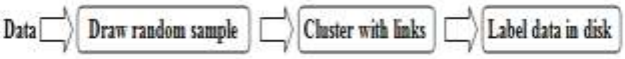

Data
phase 1: Load into memory by building a CF tree
Initial CF tree
Phase 2:(optimal)
: Condense into desirable range by building a smaller CF tree
smaller CF tree
Phase 3: Global clustering
Good Clusters
Phase 4:(optional and off line)
:Cluster Refining
Better Clusters

The data pre-processing algorithm BIRCH groups the data set into compact sub clusters that have summary statistics (called Clustering Features (CF)) associated to each of them. These CF's are computed and updated as the sub clusters are being constructed. The end result is an ``in-memory" summary of the data, where ``local" compact sub clusters are represented by appropriate summary statistics [13]. BIRCH can achieve a computational complexity of O(n). Two generalizations of BIRCH, known as BUBBLE and BUBBLE-FM algorithms [3].this algorithm can find approximate solution to combinatorial problems with very large data sets [13].

ROCK (RObust Clustering using linKs)

ROCK [22] a robust hierarchical-clustering algorithm is an agglomerative hierarchical clustering based on the notion of links [14]. It is appropriate for handling large data sets [3]. ROCK combines, from a conceptual point of view, nearest neighbor, relocation, and hierarchical agglomerative methods. In this algorithm, cluster similarity is based on the number of points from different clusters that have neighbors in common [10]. The steps involved in clustering using ROCK are described in Fig 7. The space complexity of the algorithm depends on the initial size of the local heaps. Therefore space complexity of ROCK's clustering algorithm is O(min {n2, nmmma}), where n is number of input points, ma and mm are the average and maximum number of neighbors for a point, respectively. It has a worst-case time complexity of O(n2 + nmmma + n2 logn). A robust hierarchical clustering algorithm ROCK was develop that employs links and not distances for merging clusters [15]. A quick version of the ROCK algorithm for clustering of categorical data is proposed, it is called QROCK. It has the complexity O(n2). The performance analyses also demonstrate that QROCK is quicker than ROCK [14]

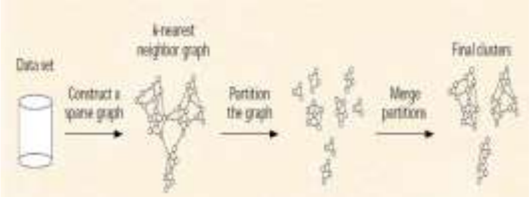
Figure no 7 ROCK Process



Chameleon

Chameleon[22] as a hierarchical agglomerative clustering algorithm can find dynamic modeling. It is based on two phases: at first partitions the data points into sub-clusters, using a graph partitioning, then repeatedly merging sub-clusters, com from previous stage to obtain final clusters. The algorithm is proven to find clusters of diverse shapes, densities, and sizes in two-dimensional space [17]. Chameleon is an efficient algorithm that uses a dynamic model to obtain clusters of arbitrary shapes and arbitrary densities [2]. Fig 8 provides an overview of the overall approach used by Chameleon to find the clusters in a data point [16].

Figure no 8 Chameleon Process



The algorithm is well suited for the applications where the volume of the available data is large. For large n, the worst-case time complexity of the algorithm is O(n(log2 n + m)), where m is the number of clusters formed after completion of the first phase of the algorithm[3].

Linkage algorithms

Linkage algorithms [22] are hierarchical methods that merging of clusters is based on distance between clusters. Three important type of these algorithms are Single-link(S-link), Average-link (Ave-link) and Complete-link (Com-link).They are agglomerative hierarchical algorithms too. The Single-link distance between two subsets is the shortest distance between them,

Average-link the average distance and the Complete-link the largest distance [17]. From [1] Single-link time complexity and space complexity is O(n2log n) and O(n2) respectively.

Table no 1 Linkage Methods or Measuring Association D12 Between Clusters 1 and 2

Single linkage	d12=mini,jd(Xi,Yj)	This is the distance between the closest members of the two clusters.
Complete linkage	d12=maxi,jd(Xi,Yj)	This is the distance between the farthest apart members
Average linkage	d12=$\frac{1}{kl}\sum_{i=1}^{k}\sum_{j=1}^{l} d(Xi,Yj)$	This method involves looking at the distance between all pairs and averages all of these distances

The obvious algorithm for computing the Complete-link clustering takes cubic time. Day and Edelsbrunner showed that it can be reduced to O(n2log n) time by Using priority queues. Murtagh proposed a quadratic-time algorithm. Later, a quadratic-time algorithm based on the (a, b)-tree data structure was developed by KArivBanek. A quadratic-time algorithm that uses linear space was proposed by Defays;

A Parallel implementation of Complete-link clustering has also been developed, but asymptotically the total work was still at least quadratic. Krznaric and Levcopoulos showed that for n points in the Euclidean plane, the

Complete-link clustering can be computed in O(nlog2 n) time and linear space. In addition, they developed an O(n log n + nlog2(1/ε)) time algorithm for constructing a Complete-link ε-approximation that uses O(n) space[18]

Leaders–Subleaders

Leaders-Subleaders [23] is an efficient hierarchical clustering algorithm that is suitable for large data sets. In order to generate a hierarchical structure for finding the subgroups or sub-clusters, incremental clustering principles is used within each cluster. Leaders– Subleaders is an extension of the leader algorithm.

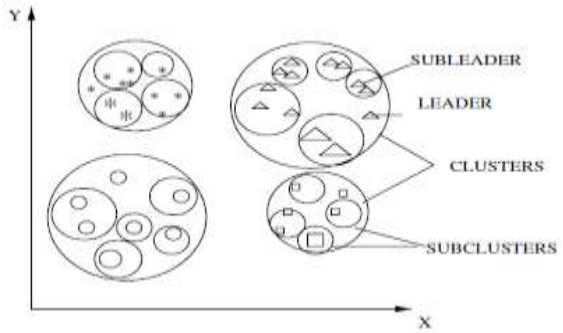
Leader algorithm can be described as an incremental algorithm in which L leaders each representing a cluster are generated using a suitable threshold value. There are mainly two major features of Leaders–Subleaders. First is effective clustering and second is prototype selection for pattern classification.

In this algorithm, after finding L leaders using the leader algorithm, the next step is to generate subleaders, also called the representatives of the sub clusters, within each cluster that is represented by a leader.

This sub-cluster generation process is done by choosing a suitable sub threshold value. Subleaders in turn help in classifying the given new or test data more accurately. This procedure may be extended to more than two levels.

An h level hierarchical structure can be generated in only h database scans and is computationally less expensive compared to other hierarchical clustering algorithms [21].

Figure no. 9. Clusters in Leaders–Subleaders Algorithm



Bisecting k-means

Bisecting k-means (BKMS) [23] is a divisive clustering algorithm. It is proposed by Steinbach et al. (2000) in the context of document clustering. Bisecting k-means always finds the partition with the highest overall similarity, which is calculated based on the pair wise similarity of all points in a cluster. This procedure will stop until the desired number of clusters is obtained. As reported, the bisecting k-means frequently outperforms the standard k-means and agglomerative clustering approaches.

In addition, the bisecting k-means' time complexity is O(nk) where n is the number of items and k is the number of clusters. Advantage of BKMS is low computational cost. BKMS is identified to have better performance than k-means (KMS) agglomerative hierarchical algorithms for clustering large document [7].

IV. Conclusion

This paper presents an overview of improved hierarchical clustering algorithm. Hierarchical clustering is a method of cluster analysis which seeks to build a hierarchy of clusters. The quality of a pure hierarchical clustering method suffers from its inability to perform adjustment, once a merge or split decision has been executed. This merge or split decision, if not well chosen at some step, may lead to some-what low-quality clusters. One promising direction for improving the clustering quality of hierarchical methods is to integrate hierarchical clustering with other techniques for multiple phase clustering. These types of modified algorithm have been discussed in our paper in detail [23].

References

[1]. A.K. Jain, M.N. Murty and P.J. Flynn, Data clustering: A review, ACM Computing Surveys, 31(1999), 264-323.

[2]. N. A. Yousri, M. S. Kamel and M. A. Ismail, A distance-relatedness dynamic model for clustering high dimensional data of arbitrary shapes and densities, Pattern Recognition, 42 (2009), 1193-1209.

[3]. K. Koutroumbas and S. Theodoridis, Pattern Recognition, Academic Press, (2009).

[4]. M. Kantardzic ,Data Mining: Concepts, Models, Methods, and Algorithms, John Wiley & Sons,(2003).

[5]. R. Capaldo and F. Collova, Clustering: A survey, Http://uroutes.blogspot.com, (2008).

[6]. D.T. Pham and A.A. Afify, Engineering applications of clustering techniques, Intelligent Production Machines and Systems, (2006), 326-331.

[7]. L. Feng, M-H Qiu, Y-X. Wang, Q-L. Xiang, Y-F. Yang and K. Liu, A fast divisive clustering algorithm using an improved discrete particle swarm optimizer, Pattern Recognition Letters, 31 (2010),1216-1225.

[8]. R. Gil-García and A. Pons-Porrata, Dynamic hierarchical algorithms for document clustering, Pattern Recognition Letters, 31 (2010), 469-477.

[9]. S. Guha, R. Rastogi and K. Shim, CURE: An efficient clustering algorithm for large databases,Information Systems, 26 (2001), 35-58.

[10]. J.A.S. Almeida, L.M.S. Barbosa, A.A.C.C. Pais and S.J. Formosinho, Improving hierarchical cluster analysis: A new method with outlier detection and automatic clustering, Chemometrics and Intelligent Laboratory Systems, 87 (2007), 208-217.

[11]. M. Charikar, C. Chekuri, T. Feder and R. Motwani, Incremental Clustering and Dynamic Information Retrieval, Proceeding of the ACM Symposium on Theory of Computing, (1997), 626-634.

[12]. T. Zhang, R. Ramakrishnan and M. Livny, BIRCH: An efficient clustering method for very large databases, Proceeding of the ACM SIGMOD Workshop on Data Mining and Knowledge Discovery,(1996), 103-114.

[13]. J. Harrington and M. Salibián-Barrera, Finding approximate solutions to combinatorial problems with very large data sets using BIRCH, Computational Statistics and Data Analysis, 54(2010), 667.

[14]. M. Dutta, A. KakotiMahanta and A.K. Pujari, QROCK: A quick version of the ROCK algorithm for clustering of categorical data, Pattern Recognition Letters, 26 (2005), 2364-2373.

[15]. S. Guha, R. Rastogi and K. Shim, ROCK: A robust clustering algorithm for categorical attributes, Information Systems, 25 (2000), 345-36.

[16]. G. Karypis, E.H. Han and V. Kumar, CHAMELEON: Hierarchical clustering using dynamic modeling, IEEE Computer, 32 (1999), 68-75.

[17]. Y. Song, S. Jin and J. Shen, A unique property of single-link distance and its application in data clustering, Data & Knowledge Engineering, 70 (2011), 984-1003.

[18]. D. Krznaric and C. Levcopoulos, Optimal algorithms for complete linkage clustering in dimensions, Theoretical Computer Science, 286 (2002), 139-149.

[19]. P.A. Vijaya, M. NarasimhaMurty and D.K. Subramanian, Leaders–Subleaders: An efficient hierarchical clustering algorithm for large data sets, Pattern Recognition Letters, 25 (2004), 505-513.

[20]. V.S. Ananthanarayana, M. NarasimhaMurty and D.K. Subramanian, Rapid and Brief

[21]. PavelBerkhin (2000), Survey of Clustering Data Mining techniques ,Accrue Software, Inc..

[22]. M. Kuchaki Rafsanjani, Z. AsghariVarzaneh, N. EmamiChukanlo / TJMCS Vol .5 No.3 (2012) 229-240

[23]. YogitaRani and Dr. Harish Rohil A Study of Hierarchical Clustering Algorithm International Journal of Information and Computation Technology. ISSN 0974-2239 Volume 3, Number 11 (2013), pp. 1225-1232

[24]. Manish Verma, MaulySrivastava, NehaChack, Atul Kumar Diswar, NidhiGuptaa : A Comparative Study of Various Clustering Algorithms in Data MiningInternational . Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622

[25]. Yogita Rani, Manju& Harish Rohil: Comparative Analysis of BIRCH and CURE Hierarchical Clustering Algorithm using WEKA 3.6.9 The SIJ Transactions on Computer Science Engineering & its Applications (CSEA), Vol. 2, No. 1, January-February 2014

[26]. Dr.A.Senguttuvan,PramodhKrishna,Dr.K.VenugopalRao: Performance Analysis of Extended Shadow Clustering Techniques and Binary Data Sets Using K-Means Clustering.International Journal of Advanced Research in Computer Science and Software Engineering

[27]. "CURE"inhttp://www.sciencedirect.com/science/article/pii/S0306437901000084