ISSN: 2379-3686 International Journal of Science Research and Technology
Volume 2 Issue 1, p p 31-36, 25th March 2016

HIERARCHICAL CLUSTERING TECHNIQUES IN DATA MINING: A COMPARATIVE STUDY

Rani Geetika
IT Department, DAV Institute of Engineering and Technology
Kabir Nagar, Jalandhar, Punjab, India

***Abstract*—Clustering is an important data mining technique which has gained a tremendous importance in recent times due to its inherent nature of capturing the hidden structure of the data. In Clustering, different objects that have some similarity based on their characteristics are brought together into a group. Hierarchical Clustering Analysis is one of the clustering techniques which play a significant role in many applications in real world such as text mining, data analysis, market research, pattern recognition, and many more. This paper focuses on bringing out the comparison among different hierarchical clustering techniques.**

Keywords—Data Ming, Data Mining Techniques, Clustering, Hierarchical Methods.

I. INTRODUCTION

Clustering is one of the unsupervised learning in which there are no predefined set of classes [1]; hence resultant clusters are not known beforehand. In clustering, the different objects are grouped in such a way that objects in the same group have more similarity to each other as compared to the ones in other groups. This grouping of objects results in clusters. We can also say that clustering is a technique which divides data into groups of similar objects as shown in Figure 1. However, representing a large dataset by fewer clusters results in loss of certain fine details. A clustering method is considered to be of good quality if it produces clusters which have high intra-class similarity and low inter-class similarity. The goodness of a clustering technique depends on the similarity measure used and its ability to find out some or all of the hidden patterns. A number of distance functions are available, however the most commonly used for finding similarities are Minkowski, Manhattan and Euclidean distance. In data mining, there are some requirements for data clustering. These requirements are Scalability, Ability to deal with different types of attributes, Ability to handle dynamic data, Discovery of clusters with arbitrary shape, Minimal requirements for domain knowledge to determine input parameters, Able to deal with noise and outliers, Insensitive to order of input records, High dimensionality, Incorporation of user-specified constraints, Interpretability and usability[1].

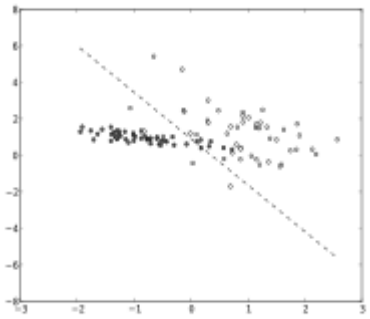


Fig. 1 CLUSTERING

A number of clustering techniques are available in data mining [4], which results in different dataset clustering. The main categories of clustering methods are [2]:

- Partitioning Methods
- Hierarchical methods
- Density-based methods
- Grid-based methods
- Model-based methods
- Constraint based methods

In this paper, our main focus is on Hierarchical clustering techniques. The study includes discussion and comparison of Hierarchical algorithms. The paper is organized as follows: Section 2 describes hierarchical clustering process. Section 3 denotes categories of Hierarchical clustering algorithms. Comparison between algorithms present in section 4 and finally paper concludes in section 5.

II. Related Study

Hierarchical methods are well known clustering technique used in data mining tasks. In hierarchical clustering, the dataset is not partitioned into particular clusters in single step. It takes a series of partitioning steps, in which a single cluster is partitioned to a number of clusters each of which contain a single object. We can also say that a hierarchical clustering scheme produces a nested sequence of clusters in which each clustering is nested into the next clustering in the sequence [6]. It uses the distance matrix criteria for clustering the data. Typically, each iteration involves merging or splitting a pair of clusters based on a certain criterion, often measuring the proximity between clusters. The merging or splitting stops once the desired number of clusters has been formed.

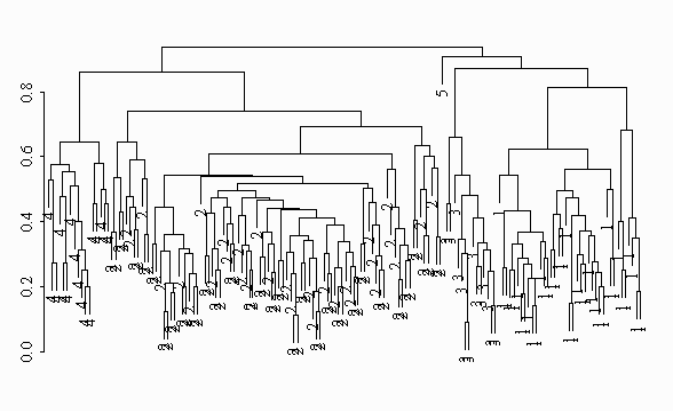


Fig. 2 Hierarchical Clustering

A hierarchical clustering algorithm is of two types: An agglomerative clustering takes single clusters and keeps on merging them recursively. The agglomerative hierarchical clustering is also known as AGNES. A divisive clustering takes one cluster consisting of all data and then keep on splitting them into appropriate cluster recursively until a stopping condition (frequently, the requested number k of clusters) is fulfilled. The process of AGNES and DIANA is depicted in Figure 3 below:

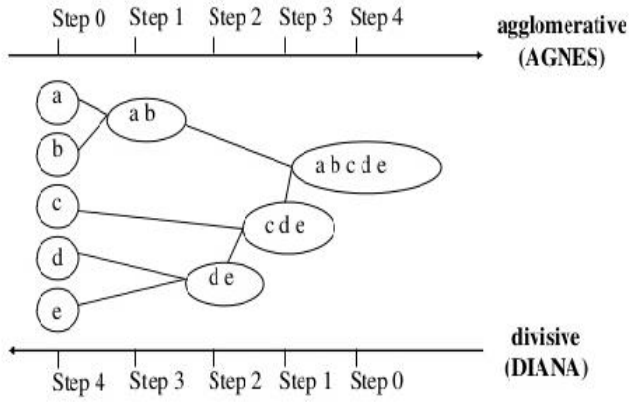


Fig.3 Hierarchical Clustering

A hierarchical clustering is a greedy search algorithm based on a local search, the merging decision made early in the agglomerative process are not necessarily the right ones. One possible solution to this problem is to refine a clustering produced by the agglomerative hierarchical algorithm to potentially correct the mistakes made early in the agglomerative process.

III. Hierarchical Clustering Techniques

The examples of hierarchical clustering algorithms include ROCK, CURE, CHAMELEON and BIRCH. All of these techniques are based on agglomerative hierarchical clustering. ROCK and CURE uses a static model to determine the most similar cluster to merge in the hierarchical clustering whereas BIRCH algorithm proposed by Zhang, Ramakrishnan, and Livny [11], first performs hierarchical clustering with a clustering features (CF) tree before applying other techniques to refine the clusters. CHAMELEON uses dynamic modeling to measure the similarity between two clusters.

Further details about these techniques have been given in coming subsections.

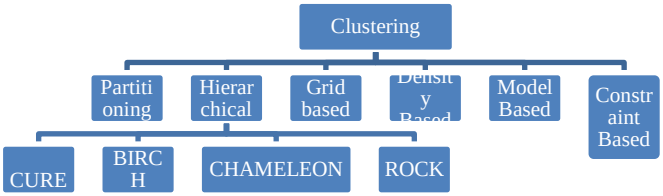
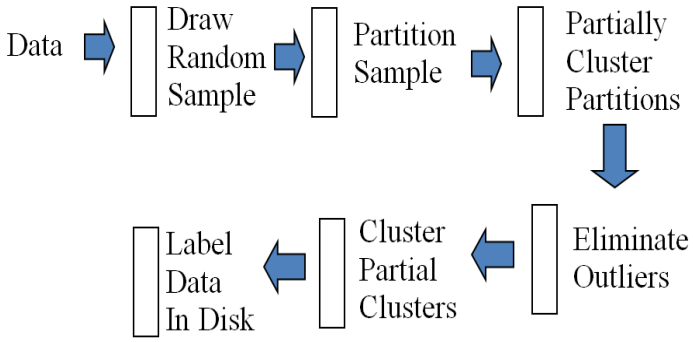


Fig. 4 Clustering Classification

A. CURE (Clustering Using Representative)

It is a hierarchical clustering algorithm based on partitioning which can identify clusters of any shape. The objects are clustered together based on how close are the representative points of different clusters [8], without considering the internal closeness (density or homogeneity) of the two clusters involved [7]. A constant number c of well scattered points in a cluster are chosen, and then shrunk toward the center of the cluster by a specified fraction alpha. The clusters with the closest pair of representative points are merged at each step and stops when there are only k clusters left, where k can be specified. It helps in avoiding noise. It cannot be applied to large data sets [3]. The CURE process is presented in Fig. 3.

Fig. 5 CURE PROCESS [7]



B. BIRCH (Balanced Iterative Reducing and Clustering Using Hierarchies)

Balanced Iterative Reducing and Clustering Using Hierarchies (BIRCH) is an efficient data clustering method proposed by Charikar et al. in 1997[9]. It is an agglomerative hierarchical clustering algorithm especially designed for very large databases. The algorithm scan data and build balanced trees representing data clusters, each point is inserted into a cluster based on a local decision, and may cause the cluster to split/merge. The main goals of BIRCH algorithm is to minimize running time and space requirements by minimizing the number of I/O operations [3] and handle "noise" (outliers of clusters) well. It introduces a novel hierarchical data structure, CF-tree, for compressing the data into many small sub-clusters and then performs clustering with these summaries rather than the raw data. A Clustering Features Tree (CF-tree) is a hierarchical data structure for multiphase clustering. For each successive data point, the CF-tree is traversed to find the closest cluster to it in the tree, and if the point is within a threshold distance of a closest cluster, it is absorbed into it. Otherwise, it starts its own cluster in the CF-tree.

Sub-clusters are represented by compact summaries, called cluster-features (CF) that are stored in the leafs. The non-leaf nodes store the sums of the CF of their children. A CF-tree is built dynamically and incrementally, requiring a single scan of the dataset. An object is inserted in the closest leaf entry. Two input parameters control the maximum number of children per non-leaf node and the maximum diameter of sub-clusters stored in the leafs. By varying these parameters, BIRCH can create a structure that fits in main memory. Once the CF-tree is built, any partitioning or hierarchical algorithms can use it to perform clustering in main memory. Fig. 4 presents the overview of BIRCH [10].

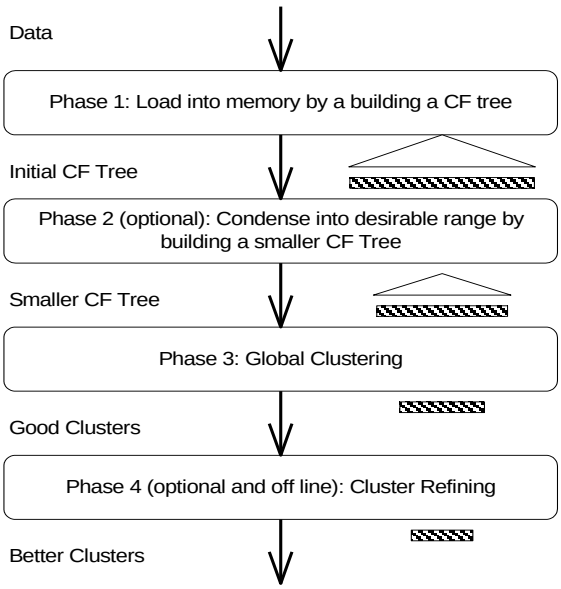


Fig. 6 BIRCH PROCESS [10]

BIRCH is reasonably fast, but has two serious drawbacks: data order sensitivity and inability to deal with non-spherical clusters of varying size because it uses the concept of diameter to control the boundary of a cluster.
BIRCH can typically find a good clustering with a single scan of the data, and improve the quality further with a few additional scans. It can also find approximate solutions to combinatorial problems [11].

C.CHAMELEON

CHAMELEON (Clustering Using Dynamic Model) is a hierarchical clustering algorithms proposed by Karypis et al. [12]. It uses dynamic modeling to determine the similarity between pairs of clusters. It uses a k-nearest-neighbor graph [16] to construct sparse graph. CHAMELEON operates on a sparse graph in which nodes represents data items, and weighted edges represent similarities among the data items. Chameleon uses a graph partitioning algorithm to partition the k-nearest-neighbor graph into a large number of relatively small sub clusters. This process terminates when the larger sub-cluster contains less than a specified number of vertices that is named Minsize. The Minsize parameter essentially controls the granularity of the initial clustering solution. Minsize is used to be in the 1 % and 5 % range.

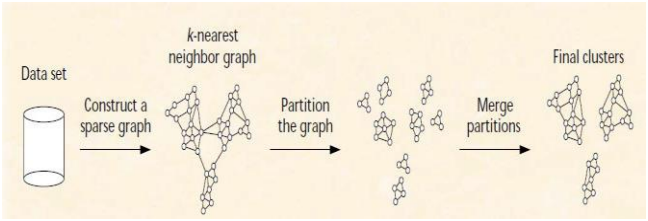


Fig. 7 CHAMELEON PROCESS [16]

It then uses an agglomerative hierarchical clustering algorithm that repeatedly merges sub clusters based on their similarity. In the clustering process, two clusters are merged only if the inter-connectivity and closeness (proximity) between two clusters are high relative to the internal inter-connectivity of the clusters and closeness of items within clusters. Chameleon measures the closeness of two clusters by computing the average similarity between the points in Ci that are connected to points in Cj.

D. ROCK (Robust Clustering using Links)

ROCK is a link based agglomerative hierarchical clustering algorithm [14]. It can handle a large amount of data [3]. ROCK can be considered as a combination of nearest neighbor, relocation, and hierarchical agglomerative methods. In this process, a random number of samples are drawn from the database. Clusters are merged based on the number of points from different clusters that have neighbors in common [15]. Finally the clusters involving only the sample points are used to assign the remaining data points on disk to the appropriate cluster. The process of clustering using ROCK is defined in Fig 6.

Fig. 8 ROCK PROCESS [14]

The initial size of the local heaps decides the space complexity of the algorithm. The space complexity of ROCK's clustering algorithm is O ($\min\{n^2, nm_m m_a\}$), where n is number of input points, ma and mm are the average and maximum number of neighbors for a point, respectively. The worst-case time complexity is $O(n^2 + nm_m m_a + n^2 \log n)$.

IV. Results and findings

A comparative analysis of different hierarchical clustering algorithm, discussed in previous sections has been summed up in tabular form in Table 1 given below. This study has brought some light to the important findings which are discussed further in this section. The algorithms BIRCH and CHAMELEON are based on dynamic model [3] while CURE and ROCK exploit static model. CURE algorithm provides a better execution time as compared to BIRCH's [7]. BIRCH algorithm is found to be incremental and sensitive to order of data [5]. CURE ignores the information about the aggregate inter-connectivity of objects in two clusters which is taken care of in Chameleon algorithm [5]. CHAMELEON is an efficient algorithm which handles two major drawbacks in CURE and ROCK hierarchical clustering algorithms respectively: inter-connectivity of two clusters, which is in CURE algorithm; closeness of two clusters, which is in ROCK algorithm [5].

V. Conclusion

This paper brings together the different kind of hierarchical clustering techniques used in data mining. We included definition, classification of clustering techniques and then focused on algorithms of hierarchical clustering along with their merits and demerits. Finally, a comparative analysis has been tabulated. It has been found that there is no single algorithm which performs best in every domain, henceforth, every algorithm has its own merits and demerits and performs best in their suitable domain.

TABLE I

COMPARISON OF HIERARCHICAL CLUSTERING ALGORITHMS

Sr. No	Algorithm	Merits	Demerits	Time Complexity	**Model** **Static/ Dynamic**	Shapes
1	CURE	• Less sensitive to outliers. • Adjusts well to geometry of non-spherical shapes.	• CURE ignores the information about the aggregate inter-connectivity of objects in two clusters.	$O(n^2 \log n)$	Static	Arbitrary
2	BIRCH	• Scales linearly: finds a good clustering with a single scan and improves the quality with a few additional scans.	• Handles only numeric data, and sensitive to the order of the data record. • Favors only clusters with spherical shape and similar sizes. • Can't distinguish between big and small clusters.	O(n)	Dynamic	Spherical
3	CHA	• Well	• CHAM	$O(n^2)$	Dynami	Arbitrar

	MELEON	suited for large volume of data. • Obtain clusters of arbitrary shapes and densities.	ELEON is known for low dimensional spaces, and was not applied to high dimensions.		c	y
4	ROCK	• Most suitable for clustering data that have boolean and categorical attributes. • Generate better quality clusters. • Highly Scalable.	• Ignores the potential variations in the inter-connectivity of different clusters within the same data set. • Inflexible	O(n2 +m_m m_a+ n^2log n)	Static	Arbitrary

VI. REFERENCES

[1] A.K. Jain, M.N. Murty, P.J. Flyn, "Data Clustering: A Review", ACM Computing.
[2] HAN, J. and KAMBER, M. 2001. Data Mining. Morgan Kaufmann Publishers.
[3] K. Koutroumbas and S. Theodoridis, Pattern Recognition, Academic Press, (2009).
[4] N. Mehta S. Dang "A Review of Clustering Techniques in various Applications for effective data mining" International Journal of Research in Engineering & Applied Science vol. 1, No. 1, 2011.
[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] Guha, S., Rastogi, R., & Shim, K., CURE: An efficient Clustering algorithm for large databases, Proc. Of ACG SIGMOD Intl. Conf. on Management of Data, pp. 73-82, 1998.
[8] 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
[9] 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.
[10] Zhang, T., Ramakrishnan, R., Livny, M., Birch: An efficient data clustering method for very large databases, Proc. Of the ACM SIGMOD Conf. on Management of Data, pp.103-114, Montreal, Canada, 1996.
[11] 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), 655-667.
[12] Karypis, G., E. Han & V. Kumar, CHAMELEON: A hierarchical clustering algorithm using dynamic modeling, IEEE Computer, 32(8), pp.68-75, 1999.
[13] Kaufman, L., Rousseeuw, P.J., Finding Groups in Data: an Introduction to Cluster Analysis, New York: John Wiley &Sons, 1990.
[14] Guha, S., Rastogi, R., & Shim, K., ROCK: A Robust clustering algorithm for categorical attributes, Information Systems, vol. 25, No. 5, pp. 345-366, 2000.
[15] M. Dutta, A. Kakoti Mahanta and A.K. Pujari, QROCK: A quick version of the ROCK algorithm for clustering of categorical data, Pattern Recognition Letters, 26 (2005), 2364-2373.
[16] V. Gaede and O. Gunther, Multidimensional Access Methods, ACM Computing Surveys, Vol. 30, No. 2, 1998.