Journal of Engineering and Applied Sciences 12 (24): 7501-7507, 2017
ISSN: 1816-949X

A Review on Hierarchical Clustering Algorithms

Vijaya, Aayushi Sinha and Ritika Bateja
Department of Computer Science and Engineering,
Manav Rachna International University, Faridabad, 121004 Haryana, India

Abstract: Clustering is the process of grouping the datasets into various clusters such that the variations within the clusters are very small but between the clusters are remarkable. Clustering has a wide application field like data concept construction, simplification, pattern recognition, etc. Clustering methods are mainly classified into two groups, hierarchical and partitioning. The hierarchical clustering method defines the hierarchy of clusters by splitting and merging them whereas partitioning method involves defining partitions and their evaluation based on some criteria. Thus, clustering algorithms chosen need to be efficient. This study focuses on different types of hierarchical clustering algorithms as well as various advanced clustering algorithms based on hierarchical clustering. It also discusses their strengths and weaknesses in detail.

Key words: Clustering, data mining, hierarchical clustering algorithms, partitioning, construction, splitting

INTRODUCTION

With a natural evolution in information technology, the amount of data increases explicitly with an imminent need of converting that data into useful information. Thus, the concept of data mining came into existence. Since, data mining is the extraction of interesting (not known) patterns from huge amount of data, it has attracted great attention and has become considerably important and necessary. It is the process of examining large pre-existing databases in order to extract new information by analysing a database or collection of records and ending up summarising it into useful information. It is the most essential requirement of today's as it help companies to extract information of their interest from their database warehouse. Data mining is one of the essential features of knowledge data discovery. There are certain phases defined for data mining process such as problem definition phase, data exploration phase, data pre-processing step, data modeling, model, evaluation phase and deployment phase. Each phase has its own work related to data mining. A large repository or storage space is used to store the data (current and historical data) and then various data mining techniques such as clustering, classification, decision tree, pattern evaluation, prediction, etc. are applied on that data and the model is prepared. There are different kind of data such as spatial data, transactional data, analytical data, etc., on which these mining techniques can be applied (Han *et al*., 2011). In fact now a days, data mining has exploited each and every field of human life. One of the real life example of data mining is customer churn prediction (say in telecom industry). The main motive of predicting customer churn is to identify how willingly the customer is to unsubscribe the services provided by them. After predicting the churn rate the telecom company might provide some exiting offers to the customer to continue their service with that customer. There are certain parameters which the company considers such as recency, frequency, etc., to calculate the churn rate. Data mining has also been used in business management, production control, market analysis for strategy formulation and exploration, risk analysis, cross-selling, decision support, fraud detection, forecasting, understanding customer behaviour, etc. In spite of all the advantages there are certain issues related with data mining such as mining methodology and user interaction, performance related issues and diverse data type issues, etc. as different users may be interested in different kinds of knowledge, so, it is necessary for data mining to cover a broad range of data mining tasks. Many data mining techniques are also affected by noise or incompleteness of data. Thus, the quality of information is adversely affected. With an increase in the data set size, the complexity of the data mining algorithms or techniques also increases, thus, putting an overhead on user as well as the system. There are different types of

Corresponding Author: Vijaya, Department of Computer Science and Engineering, Manav Rachna International University, Faridabad, 121004 Haryana, India

data mining techniques such as classification, association rule learning, regression, clustering (Arockiam *et al*., 2012), etc. The objective of this study is to discuss clustering.

MATERIALS AND METHODS

Clustering: Clustering is an unsupervised learning method. It deals with the data structure partitioned in an unknown area and is the basis of further learning. It is a technique defined for grouping up of a set of objects, called clusters on the basis of their characteristics and thus finding structures in unlabelled data. Clustering is finding its huge applications in computational biology, Micro array data analysis, bioinformatics, medical imaging, business and marketing, social network analysis, software evaluation and agriculture (Arockiam *et al*., 2011). One can also say that clustering is now a part of each and every field of regular life. To illustrate clustering let us consider an example of hospital management system where each patient is a data object. Now patients with similar symptoms can be placed in one common ward as it might be possible that they require a common treatment. Thus, an ease for doctors to treat large number of patients in small time. Different wards can be compared with different clusters with patients in one ward as the data item for one cluster. The basic idea of clustering is to partition large set of data into various similarity restricted domains to make the study and analysis, a simpler thing as the clusters are formed in such a way that their inter cluster similarity (similarity between the clusters) is minimum and intra cluster similarity (similarity within the cluster) is maximum.

Therefore, it helps to find structures that are hidden in large dataset. The entire clustering process can be broadly divided into four steps: feature extraction and selection which involves selecting the specific data from the entire database based on some features on which clustering is to be performed, clustering algorithm design which defines the algorithm to be used for making the clusters, result evaluation stage evaluates the clusters that whether they meet the organisations need or not and result explanation in which clusters are ready for analysis and explanation. There are several methods for the clustering process. However, most of the methods are part of two broadly defined categories for clustering methodologies, hard clustering and soft clustering. Hard clustering is a technique in which each data elements of a data set can be a part of only one cluster. It is also called as exclusive clustering as there will be no overlapping of clusters. One of the most common method of hard clustering is K-means method. The major drawback of the hard clustering method is inflexibility in defining the clusters as any data element is strictly restricted to only one cluster, so it does not work well when data are not properly separated. To sort out this inflexible approach soft clustering concept is often used in the existing literature. Soft clustering, also known as overlapping clustering is the form of partitioning in which data can be a part of more than one cluster. In this clustering data objects, lying on the boundaries of two clusters are not forced to exactly belong to only one cluster but with each data a grade point or membership degree (between 0 and 1) can be associated which explains up to what extent the data element is part of any cluster. One of the most famous techniques of soft clustering is fuzzy c means (Bora and Gupta, 2014).

Hierarchical clustering algorithm: Hierarchical clustering algorithm produces hierarchical clusters and the classification of clusters depends on whether the hierarchical decomposition is formed in a bottom-up or top-down style. It produces nested clusters starting with a single object as a cluster to a single cluster having all the objects or vice versa. In general, the merges and split are done inferring minimum cost. In existing literature Hierarchical clustering algorithm is often represented by dendrogram, a simple pictorial representation which shows different stages/order at which clusters are divided or combined (Rani and Rohil, 2013). One of the most attracting features of hierarchical clustering is that the total number of clusters to be formed is not fixed at the initial level. Therefore, any desired number of clusters can be formed in this clustering which makes it a flexible approach. However, the main disadvantage is that once a merge or split process has occurred, it is difficult to make adjustments in a cluster. So, if the processes are not selected in a good way it may led to low quality clusters. Another issue with this type of clustering is absence of any globally defined function to create the clusters. Since, the cluster formulation is a local step which is decided at every step, so there is no defined function which can led to exact cluster formation. Hierarchical clustering supports the storage of data points in the form of proximity matrix. It requires computation and storage of m^2 proximities for m data objects. Thus, for m data objects the space complexity of hierarchical clustering is m^2 whereas it's time complexity is $O(m^2 \log m)$. There are two types of hierarchical clustering algorithm named as: agglomerative clustering and divisive clustering. The objective of this study is to discuss both the types in details. Now, let us move towards agglomerative clustering.

Agglomerative clustering: As the name suggests agglomerative clustering includes summing up of clusters. In this clustering, the process generally starts with each object as a single and separate cluster. At the next step, distance between all the clusters is compared (one can use a distance matrix) and those two clusters which have minimum distance between them are treated as the most

similar clusters and they are merged together. Then, the distance matrix is updated correspondingly and the process continues until only one cluster remains. The most attracting feature of agglomerative clustering is that it produces better quality of clusters and further it can be used to create taxonomy. However, there are certain issues related with this clustering such as it is quite expensive in terms of storage and computational time requirement and also its efficiency is affected with an increase in the size of the data set as it's run time complexity is $O(m^2 logm)$ (Firdaus and Uddin, 2015). Another problem encountered with this technique is the problem of data quality as merging of data can lead to noisy high dimensional data. The basic step of agglomerative clustering algorithm is determination of proximity between the clusters and the differences in proximity calculation leads to different agglomerative clustering methods. In some cases, proximity can be defined as the proximity between the two farthest points of the clusters. This defines the max. technique. For defining the min. technique, the proximity is calculated as the proximity between the two closest pair of data points between the clusters. Further, AVG technique lies in between the min. and max. technique which calculates average of proximities between all the pairs of data points between the clusters.

Divisive clustering: The oldest type of hierarchical clustering is divisive clustering which is just an opposite of agglomerative clustering. In this clustering, the process starts with all the objects or datum in one single cluster and at each step the cluster divides into sub-clusters until all datum exist as single clusters. This clustering involves mainly three major steps (Firdaus and Uddin, 2015). The first step performs a splitting procedure for the subdivision of clusters into two sub clusters. On the second step, a local evaluation of the bipartitions resulting from the splits of clusters is done and on the last step formula for determining the nodes levels of the resulting dendrogram is generated. After the selection of cluster to be split next, a simple approach based on distance matrix can be used to carry out the splitting procedure. A distance matrix ($n \times n$ matrix that contains distance of n data object from all the other data objects (including itself)) is calculated and then two most distant objects are chosen as the centre of new clusters and further process is carried out. Divisive clustering is also called top-down clustering (Roux, 2015) as it starts with a single cluster at the top which contains all the documents and then it get further divided into subparts using a flat algorithm. This clustering is more conceptually complex than agglomerative clustering as it requires a subroutine or another flat algorithm. However, it is faster than agglomerative clustering. In fact, agglomerative clustering method makes the decision on some local pattern without involving the whole information of the distribution of the entire dataset whereas in divisive clustering it has the complete information about the global distribution to proceed for the initial division. On the basis of number of variables considered to proceed for the analysis task, divisive clustering can be divided into two parts, monothetic and polythetic.

Monothetic: In this divisive type algorithm, splitting of clusters is done following only one attribute which is the most varying one so that, the resulting clusters are similar with respect to that variable (Firdaus and Uddin, 2015). Monothetic clustering is easy while handling huge dataset (Roux, 2015). By this type it is possible to construct the hierarchy on a sample of the dataset and use classification rules to assign the rest of the objects. Moreover, dividing a cluster according to a single variable can also be a deficiency in some situations. Monothetic method has also given good results on the Fisher's iris dataset and on other real applications where it has been compared with the dynamical clustering method and the ward agglomerative hierarchical method (Chavent, 1998). Selection of a monothetic clustering algorithm is preferable as it is well suited for generating concept hierarchies (Kummamuru *et al*., 2004).

Polythetic: In this algorithm, clusters are split by considering all the attributes. It involves analysis across several dimensions. Some of the steps included in polythetic algorithm are given (Firdaus and Uddin, 2015):

- Measure the distance between two objects. Also decide a threshold distance
- Create a distance matrix by calculating distances between all the objects
- The two objects having the maximum distance between them are the most dissimilar one
- Use the objects as the core for k-means method to develop two new clusters
- Continue until one cluster contain only one object

One of the major issues related with divisive clustering is the computational cost which is $O(2^n)$ for n objects which makes it worse in some cases.

RESULTS AND DISCUSSION

Advanced hierarchical clustering algorithms: The objective of this study is to present some common

hierarchical clustering algorithms which are advanced enough to deal with the issues of the traditional hierarchical methods discussed in the previous study.

CURE: CURE is basically an agglomerative clustering method which uses partitioning technique for creating clusters. Actually it is a combination of both sampling and partitioning. This clustering technique is basically used when the dataset is too large to be analysed. It also takes under consideration the concept of outliers because its shrinking property allows it to compensate the effect of outliers. CURE is basically used to deal with the problem of outliers and handle clusters of non-uniform shape and size as it allows more than one representative per clusters. It uses hierarchy based algorithm which is a midway between centroid based and all point based approach. CURE handles the disadvantages of both the method one is centroid based methods which considers only one point as the representative of the clusters (the centroid) thus, creating a problem as in large clusters, the centroids of the sub clusters can be far apart leading to the splitting of the cluster and also the other one is all points based method which considers all points as the representative of the clusters, thus, CURE method becomes purely centroid based method while if it is 0 then, CURE method becomes all-points based method.

For n input size, the run time complexity for CURE algorithm is $O(n^2 \log n)$ which makes it ineffective for large databases. Talking about lower dimensions the complexity reduces to $O(n^2)$. Studies have revealed that this is a quite suitable technique for large datasets (Rani and Rohil, 2013) making it extremely sensitive to outliers. In CURE method a well-defined number of points (say n) are selected and they are shrunk towards the centroid of the cluster by a fraction (say a), these scattered points after they shrunk are used to represent the clusters. At the next level two clusters with nearest pair of representatives are merged together. CURE method can identify different types of clusters based on the value of the shrinking factor (say a) which lies between 0 and 1 (Guha *et al*., 1998). Limitation associated with CURE method is that, it completely ignores the aggregate interconnectivity information of objects in the two considering clusters.

BIRCH: BIRCH (Balanced Iterative Recursive and Clustering using Hierarchies) is an integrated agglomerative hierarchical clustering method suitable for large datasets. During the clustering process, the dataset to be clustered is very large but the memory available is limited (much smaller than the provided dataset). To deal with such memory related issues BIRCH clustering method is adopted. BIRCH method handles the clustering of large datasets by generating a compact summary of the provided dataset which contains as much distribution information as possible as performing clustering process on the summary instead of the entire dataset. This method reduces the number of input output operations. In this method objects are partitioned hierarchically using tree structure and the other clustering algorithms are used to further improve them. Thus, it supports parallel and concurrent clustering (Zhang *et al*., 1997). Since, BIRCH method is based on summarising the original dataset into sub-clusters to reduce the size of the dataset there are certain things which are taken into consideration while defining the summary such as how much information each sub-cluster should contain, how the information in the sub-clusters should be organised and how effective is the organisation. Considering all these issues and to maintain memory efficiency, BIRCH clustering method uses the concept of CF and CF-tree. A CF (Clustering Feature) is triplet containing information about sub-clusters of data whereas CF-Tree (Clustering Feature Tree) is a height balanced tree with two parameters, branching factor and threshold which is used or designed to store the clustering feature for hierarchical clustering.

In BIRCH method, at first, a CF tree is built by scanning all the data using all the memory available for a sub cluster. Proceeding forward, each sub-cluster can be treated as a single point and different clustering algorithms can be applied on them as the details that their CF vector holds is sufficient for most of the distance and quality metrics calculation. After performing this step a set of clusters holding most of the distribution pattern of data, are obtained. Lastly, additional passes (cost ineffective) are performed to get more refined and accurate cluster. The most important advantage of BIRCH method is its ability to produce the best quality clusters by incrementally and dynamically clustering the incoming multidimensional data points. Most of the times a single scan of database is sufficient but more passes can be used to improve the quality of clusters generated. It considers time and memory usage while making clusters. It is local and incremental in the sense that each clustering decision is made using the existing clusters without scanning the whole data points and thus, it does not require whole data in advance. It treats every data points with different importance. The region where data points are densely collected is recognised as a single cluster by the BIRCH method and it discards the sparse points by defining them as outliers (Zhang *et al*., 1996).

ROCK: ROCK (Robust Clustering using Links) is also an agglomerative type of clustering method which uses links rather than distance factor to form clusters. It is generally,

used to cluster large datasets having categorical or Boolean data objects. It generates better quality of clusters and also is more effective (Rani and Rohil, 2013). In case of the clusters which are not well separated from each other, clustering methods, considering only the closeness between the data points are not able to differentiate between them properly. In such cases even if two data points (say p_i and p_j) which belongs to different are neighbours, it is quite possible that they do not have large number of common neighbours. But, since, the similarity or closeness based methods only emphasise on the characteristics of the points themselves, it follows a local approach which is more prone to errors as two distinct clusters may contain few data points or outliers that are quite close or similar following only a similarity based approach would cause these two clusters to be merged which in turn will affect the quality of clusters. To deal with such issues, a more robust clustering method with global approach, ROCK is adopted (Guha *et al*., 1998). ROCK methodology involves the concept of links and neighbours.

If two points are similar enough with each other they are neighbours and the number of common neighbours is the link for that pair of points. Since, link contains global knowledge about other points existing in the neighbourhood of two points, thus, larger the link larger the possibility that this pair of points lie in the same cluster. There are certain steps defined for the ROCK method of data clustering. At first, number of links between the data objects is computed, then each cluster is treated as a singleton data point and then the merging of these data points is performed based on some goodness measure (Guha *et al*., 1999). In fact, there are two specified criteria defined for the termination of the merging phase of the ROCK method. First one is merging is done till the required number of clusters are not obtained and second is no links are available between the clusters. Since, ROCK uses the random sampled data for forming the clusters in the last phase, data labelling, ROCK assigns the remaining data points to the generated clusters. For this, firstly, a set of points from each cluster is taken out which are used for labelling of the remaining data points the original dataset. To perform labelling of data, at first, a fraction of clusters from each cluster is obtained. Any unlabelled data point from the original data set is assigned to that cluster in which it has maximum neighbours. For this, it is compared with different groups that are initially generated out from the defined clusters and it is provided to that cluster with which it is most connected (Dutta *et al*., 2005). Outliers can easily be handled using ROCK method as outliers will probably have very few or no neighbours and thus they will not take part in clustering. Random samping results in very less impact of large data size on the execution time of ROCK. Time and space complexity of ROCK algorithm is $O(n^2 \log n)$.

Chameleon method: It is a hierarchical clustering method following the concept of dynamic modelling (Firdaus and Uddin, 2015) to calculate the similarity between two data objects. This method is applicable to the data set having clusters of different shape, sizes, densities, etc. The most important property of this method is that it considers both the factors, interconnectivity and closeness of the sub clusters (Karypis *et al*., 1999) to determine the pair of most similar sub-clusters in contrast of other methods which takes into account either absolute inter-connectivity or absolute closeness. Chameleon uses sparse graph for the modelling of data in which each node represents one data item and weighted edges shows the similarities among them. Use of sparse data graph helps chameleon method to deal with large data sets. For designing a sparse graph chameleon use the k-nearest neighbour graph method, where each vertex represents a data item and an edge is provided between two vertices if they are most similar one. Working of chameleon clustering method can be divided into two phases.

Phase 1; Initial sub-cluster formation: A graph partitioning algorithm is used to partition k-nearest neighbour graph. In k-nearest graph each edge represents the similarity among data points is used to determine the minimum edge value of the data set into several partitions to minimize the edge cut value, i.e., the value of the sum of the weight of the edges that straddle partitions. Thus, a multilevel graph partitioning algorithm that minimises the edge cut value also minimises the similarity between the data points across the partition and hence, clusters are partitioned into sub-clusters.

Phase 2; Dynamic merging of the sub-clusters: After the formation of sub-clusters chameleon moves to an agglomerative hierarchical clustering that combines various sub-clusters formed in Phase 1. Since, the basic concept of agglomerative clustering is to combine the most similar clusters. It is used to join the sub clusters. This phase uses the dynamic concept of relative closeness and relative inter-connectivity to select the most similar pair of clusters. Time complexity of Chameleon two-phase clustering algorithm is $O(nm+n \log n+m^2 \log m)$ where the complexity of defining the sub clusters or of Phase 1 is $O(nm)$ whereas the complexity of merging the sub clusters or of second phase is $O(n\log n+n\log n+m^2 \log m)$. One of the major disadvantages

of chameleon method is its ability to be applicable only in low dimensional spaces. It is not suitable for high dimensions because its complexity in high dimensions is $O(n^2)$ (Karypis *et al*., 1999; Yousria *et al*., 2009).

Linkage algorithm: Linkage agglomerative clustering algorithm uses the distance concept, i.e., the distance between clusters to combine the clusters together (Rani and Rohil, 2013). There are three most important linkage algorithms, Single link (S-link), Average link (Ave-link) and Complete link (Com-link). In single linkage the distance between two clusters is the minimum distance between them (Gower and Ross, 1969). In single linkage hierarchical clustering method the distance between two clusters is defined as the minimum of the distances between all pair of data points drawn from the two considering clusters. Single linkage clustering method is adversely affected by chaining effect, i.e., in order of merging two groups it looks only for one pair of points to be close, neglecting the others. Thus, the clusters generated are not compact and too much spread out. It produces straggled or elongated clusters due to the effect of noise (unwanted data) present in the data set. Another issue associated with single linkage method is that it tends to produce long thin clusters.

This leads to difficulty in defining classes that would sub-divide the data. The efficiency of single linkage is limited due to the presence of outliers and also, it is difficult for this algorithm to deal with clusters of different density. In complete linkage method the distance between two clusters is defined as the maximum of the distances between all pair of data points drawn from the two considering clusters. The clusters generated by this method are compact or tightly bounded in nature. Complete linkage method avoids chaining but allows crowding because it is based on worst case or maximum dissimilarity, thus, between pairs, a data point can be closer to data point of other cluster than to its own cluster. Advantage of complete linkage algorithm lies in the factor that it is not affected by outliers but it has trouble dealing with convex shaped clusters and also it breaks down the large clusters. In average linkage the distance between two clusters is the average of the distances between all pair of data points drawn from the two considering clusters. Average linkage algorithm is affected by shape and size of the clusters.

Leader-subleader algorithm: Leader-subleader is an efficient for dealing with large datasets. This algorithm uses incremental clustering principles to generate a hierarchical structure for forming the sub clusters. It is an extension of leader algorithm in which L leaders are used to represent L clusters formed using a threshold value. Two most important features of this algorithm is effective clustering and prototype selection for pattern classification. This algorithm involves generating subleaders for the each cluster represented by leaders using a suitable sub-threshold value (Vijaya, 2004).

CONCLUSION

Clustering algorithms plays very important role in data mining. There are various clustering algorithms developed in the past. In this study, we have presented a detailed account on hierarchical clustering algorithms. Hierarchical clustering which seeks to form hierarchy of clusters may suffer certain cluster quality issues because of its inability to make adjustments once a split or merge has occurred. To deal with such issues a combination of basic hierarchical clustering with various techniques of multiple phase clustering has been adopted. Advanced hierarchical clustering algorithms relying on this concept have also been discussed in detail in the study. In the analysis, we have reached to a solution that every clustering algorithm has its own importance. Apart from this, selection of an efficient clustering algorithm depends upon various parameters like selection of inputs, time complexity, space complexity, presence of outliers, etc.

ACKNOWLEDGEMENTS

Researchers are thankful to the Editor, Associate Editor and Anonymous reviewers for several constructive suggestions. We also would like to express our sincere gratitude to Dr. Sukhdev Singh, Accendere Knowledge Management Services Pvt. Ltd., for his valuable comments that led to substantial improvements on an earlier version of this manuscript.

REFERENCES

Arockiam, L., S. Baskar and L. Jeyasimman, 2011. Overview of clustering techniques in agriculture data mining. Agric. J., 6: 222-225.

Arockiam, L., S.S. Baskar and L. Jeyasimman, 2012. Clustering techniques in data mining. Asian J. Inf. Technol., 11: 40-44.

Bora, D.J. and A.K. Gupta, 2014. A comparative study between fuzzy clustering algorithm and hard clustering algorithm. Int. J. Comput. Trends Technol., 10: 108-113.

Chavent, M., 1998. A monothetic clustering method. Pattern Recognit. Lett., 19: 989-996.

Dutta, M., A.K. Mahanta and A.K. Pujari, 2005. QROCK: A quick version of the rock algorithm for clustering of categorical data. Pattern Recognit. Lett., 26: 2364-2373.

Firdaus, S. and M.A. Uddin, 2015. A survey on clustering algorithms and complexity analysis. Intl. J. Comput. Sci. Issues, 12: 62-85.

Gower, J.C. and G.J.S. Ross, 1969. Minimum spanning trees and single linkage cluster analysis. J. R. Stat. Soc., Appl. Stat., 18: 54-64.

Guha, S., R. Rastogi and K. Shim, 1998. CURE: An efficient clustering algorithm for large databases. SIGMOD Rec., 27: 73-84.

Guha, S., R. Rastogi and K. Shim, 1999. ROCK: A robust clustering algorithm for categorical attributes. Inform. Syst., 25: 345-366.

Han, J., M. Kamber and J. Pei, 2011. Data Mining: Concepts and Techniques. 3rd Edn., Morgan Kaufmann Publishers, USA., ISBN-13: 9780123814791, Pages: 744.

Karypis, G., E.H. Han and V. Kumar, 1999. Chameleon: Hierarchical clustering using dynamic modeling. Computer, 32: 68-75.

Kummamuru, K., R. Lotlikar, S. Roy, K. Singal and R. Krishnapuram, 2004. A hierarchical monothetic document clustering algorithm for summarization and browsing search results. Proceedings of the 13th International Conference on World Wide Web, May 17-20, 2004, ACM New York, USA., pp: 658-665.

Rani, Y. and H. Rohil, 2013. A study of hierarchical clustering algorithm. Intl. J. Inf. Comput. Technol., 3: 1115-1122.

Roux, M., 2015. A comparative study of divisive hierarchical clustering algorithms. Cornell University, Ithaca, New York, USA. https://arxiv.org/abs/1506.08977

Vijaya, P., 2004. Leaders-subleaders: An efficient hierarchical clustering algorithm for large data sets. Pattern Recognit. Lett., 25: 505-513.

Yousria, N.A., M.S. Kamel and M.A. Ismail, 2009. A distance-relatedness dynamic model for clustering high dimensional data of arbitrary shapes and densities. Pattern Recognit., 42: 1193-1209.

Zhang, T., R. Ramakrishnan and M. Livny, 1996. BIRCH: An efficient data clustering method for very large databases. Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data Vol. 25, June 04-06, 1996, ACM, Montreal, Quebec, ISBN:0-89791-794-4, pp: 103-114.

Zhang, T., R. Ramakrishnan and M. Livny, 1997. Birch: A new data clustering algorithm and its applications. Data Mining Knowledge Discovery, 1: 141-182.