JCSE International Journal of Computer Science and Engineering Open Access

Review Paper Volume-2, Issue-4 E-ISSN: 2347-2693

Contrasting and Evaluating Different Clustering Algorithms: A Literature Review

Swati Joshi[1*], Farhat Ullah Khan [2] and Narina Thakur [3]

[1*]*Department of CSE , ASET, Amity University, Noida(Uttar Pradesh), India*
[2] *Department of CSE,ASET, Amity University, Noida(Uttar Pradesh),India*
[3]*Department of CSE , Bharati Vidhyapeeth College of Engg.,New Delhi, India*

www.ijcaonline.org

Received: 12/03/2014 Revised: 22/03/2014 Accepted: 17/04/2014 Published: 30/04/2014

Abstract— Clustering is a practice of splitting data into set of analogous objects; these sets are identified as clusters. Each cluster comprised of points that are alike among them and unalike compared to points of other cluster. This paper is being set to study and put side by side different data clustering algorithms. The algorithms under exploration are: k-means algorithm, hierarchical clustering algorithm, k-medoids algorithm, and density based algorithms. All these algorithms are analyzed on R-tool by taking same dataset under observation.

Keywords— Clustering, K-Means Algorithm, Hierarchical Clustering Algorithm, K-Medoids Algorithm, Density Based Algorithm

I. Introduction

Clustering is a special method in the widely growing field known as data analysis and is widely used in many engineering and scientific research fields such as- robotics, medicine, marketing, aeronautics etc. Cluster analysis analyses the data by hiding the underlying structure, and by dividing the data into groups or in a hierarchy of groups [1]. Then these created groups are used to analyze, if it belong to some predefined ideas or leads to some new experiments. Cluster analysis [2] is a method for exploring the structure of data without the help of any predefined assumptions so it is also called unsupervised learning.

Clustering is the process of dividing data into groups of similar entities. Each group can be called a Cluster, containing entities that are similar to one another and not similar to entities of other groups. As we said Clustering divided the data into clusters or groups and that clusters can be significant or valuable. If they are significant then clusters should grab the natural structure of data, but if it is helpful then its clusters can be used as the starting data for some further analysis. It aims is to group N data points into K clusters so that data points within the same cluster are similar, while data points in different clusters are distinct from each other. Data mining applications faces three complication: - (a) large databases (b) so many attributes (c) attributes of distinct types [3]. This requires getting into rigorous computational requirement. They present actual challenges to traditional Clustering algorithms.

II. Hierarchical Clustering

Hierarchical clustering makes a hierarchy of clusters or tree of clusters, which is known as a Dendogram. Every cluster node have child clusters, sibling clusters divide the position covered by their common ancestor. This method allows exploring data on different level of granularity. Hierarchical clustering techniques are divided into two parts: agglomerative (bottom up) and divisive (top down) [4][5]. An agglomerative approach initiates with a single point (singleton cluster) and iteratively combining two or more of the clusters having highest similarity. A divisive clustering does start with a single cluster containing all the data points and iteratively divide the most suitable cluster [6]. The process halts only when a halting criteria is achieved (generally the defined K number of clusters).

Pros of Hierarchical clustering:

- Flexible with respect to the level of granularity.
- Easy to handle any aspect of similarity or distance.
- Applicable to any type of attribute.

Cons of hierarchical clustering are:

- Vague in stopping criteria.
- Many of the algorithms in hierarchical clustering doesn't re-examine clusters once built.

Moreover hierarchical clustering deals with relatively high computational cost. Single linkage and complete linkage are two popular examples of hierarchical clustering algorithms, and they take O(N2 logN) time. In hierarchical clustering general 'Point by Attribute' representation of data is of secondary importance. In spite of this hierarchical clustering is based on NXN matrix of similarities or dissimilarities between data points and that matrix is sometimes called connectivity matrix [7]. Linkage matrix is built from elements of connectivity matrix [8].

Agglomerative Hierarchical:

- Assign each element to a different cluster.
- Calculate all pair wise distances between clusters.
- Make a distance matrix using the distance values.
- Identify the pair of clusters having minimum distance.
- Delete the pair from the matrix and combine them.

Corresponding Author: Swati Joshi[1]
Department of CSE , ASET, Amity University, Noida(Uttar Pradesh), India

- Calculate all distances from this new created cluster to all other clusters, and alter the matrix accordingly.
- Repeat the above steps until the distance matrix is reduced to one single element[9].

III. Partition based Clustering:

Partitional clustering divides a dataset at one attempt only using an objective function. K-means is a popular example of partitional clustering. It uses mean-squared-error as its objective function. Its main pros is that it runs efficiently: its computational complexity is O(NKId), where I is the number of iterations used for union, and d is the dimensionality of the dataset. As K and d are generally so much less than N, then this algorithm runs in a linear time with low dimensional data. But there does not exists a universal objective function that can be used to discover all different fundamental structures of datasets. That is why; partitional clustering generates inaccurate results when the objective function used does not capture the fundamental structure of the data. This is the cause of why partitional clustering algorithms are not capable of dealing with clusters of random shapes, different sizes and densities. Unlike classic hierarchical clustering schemes, in which clusters are not examine again after being built, relocation technique can steadily improve the quality of clusters.

One approach for partitioning the data is to identify a conceptual point of view that recognize a cluster with a specific model, as well as their not known parameters are need to be found. More precisely, probabilistic models presume that data is extracted from a collection of many populations whose distributions and priors are needed to be found. Two main advantaged are there for probabilistic models-(a) built clusters are easily interpretable (b) computation of intra cluster measures is inexpensive.

1) K-Means Algorithm-

K-means describes a prototype in terms of a centroid, which is generally the mean of a collection of data points [10]. It is applicable on objects in a continuous n dimensional space. In k-means algorithm at first K initial centroids are chosen, where K is a parameter given by user, i.e. number of required clusters. Every data points are then allotted to the closest centroid, and each group of data points allocated to a centroid is a cluster only [11]. After this the centroid of every cluster is changed, depending upon the points assigned to the cluster. This process is repeated iteratively until cluster is not changing because of points or in the same way, until the centrorids remain unchanged.

- *Allocating points to the closest centroid*-Generally the similarity measure which is used for K-means is simple as the algorithm reiteratively calculates the similarity of every data point to every centroid. For similarity measures Euclidean distance, Manhattam distance, or Jaccard coefficient is often applied to data points.
- *Centroids and objective functions applied*-In algorithm the re-computation of the centroid of each cluster is a crucial task as the centroids can differ, depending upon the proximity measure for the data points and upon the goal of clustering being done. So the key issue is that: once we have chosen a proximity measure and an objective function, the centroids that should be chosen is likely to be determined by mathematical calculations.
- *Data that exists in Euclidean space:* Here the error of each data point is calculated or we can say Euclidean distance to the nearest centroid is calculated, and then total sum of the squared errors. A set of cluster having smallest the squared error is chosen as it depicts that the prototype of this clustering are showing the better representation of the data points in their cluster.

Time and space complexity:
The space requirements for K-means are small because only the data points and centroid are needed to be stored. Explicitly, the storage required is O((t+K)n where t is the number of points and n is the number of attributes. The time complexity of K-means is also modest, i.e. linear in the number of data points. Explicitly the time needed is O(I*K*t*n), where I is the number of iterations needed for union.

Pros and Cons: K-means is simple and can be applied on vast range of data types. It is also efficient to some level, although multiple runs are usually performed[12]. K-means is not suitable for all type of data. It cannot handle non circular clusters or clusters with different sizes and densities. Though it can usually find chaste subclusters if a big enough number of clusters is specified. K-means also has problem in clustering the data that consist of outliers. In conclusion, K-means is limited to data for which there is a concept of center (centroid) lies.

2) K-MedoidsAlgorithm-

In K-medoids algorithm a cluster is depicted by one of its data points. As we have seen that with K-means problem lies in outliers and in covering any attribute type ,but this K-medoids is an easy solution because it can cover any attribute type and medoids are not sensitive to outliers because secondary cluster points do not affect them. When selection of medoids takes place, clusters can be defined as subsets of points near to respective medoids, and the objective function is described as the average distance or another similarity and dissimilarity measure between a data point and its corresponding medoid.

Two classic versions of K-medoid algorithm are the methods PAM (Partitioning Around Medoids) and the algorithm CLARA (Clustering Large Applications)[13][14]. PAM is a iterative optimization technique which does combination of relocation of data points between out looked clusters by again nominating the data points as potential medoids. The leading principle of process is to observe the effect on an objective function. And this is obviously a costly strategy. CLARA uses several samples, each sample with 40k points, and each are subjected to PAM. The whole set of data is assigned to producing medoids, the objective function is calculated, and the best structure of medoids is kept.

Further progress is done in algorithm CLARANS [15] (Clustering Large Applications based upon Randomized Search) in the perspective of clustering in spatial databases. In CLARANS a graph is considered whose nodes are the sets of k medoids and if two nodes are distinct by exact one medoid then an edge connects these two nodes. While CLARA compares small number of neighbors belonging to a fixed small sample, CLARANS does random search to produce neighbors by starting with a random node and randomly checking neighbors with the max node value.

Pros and Cons:

It is more robust than k-means in the existence of noise and outliers, because a medoid is less subjective to outliers or to other extreme values than a mean. If we see its dark side, it is relatively more costly. Its complexity is O(I K (N-K)2), where I is the total number of iterations, K is the total number of clusters, and N is the total number of objects.-Relatively it is not so much efficient. In advance total number of clusters K is need to be specified. Result and total processing time depends upon initial condition.

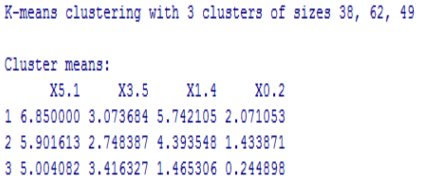
IV. Density-Based Partitioning

Here a cluster, described as a connected dense component, can grow in any direction that density leads. That is why density-based algorithms are capable of discovering clusters of arbitrary shapes. It also provides an inbuilt protection against outliers. They are scalable. These exceptional properties are tempered with some inconveniences. One drawback is that a single dense cluster having two adjacent areas with considerably different densities is not very enlightening. Another drawback is a deficiency of interpretability. Because density-based algorithms need a metric space, the real setting for them is spatial data. In order to make computations feasible, some index of data is build (such as R-tree). Index building is a topic of lively research. Traditional indices were efficient only with logically low dimensional data. There are two major approaches for density-based techniques. The first approach fastens density to a training data point. Representative algorithms contain DBSCAN, GDBSCAN, OPTICS, and DBCLASD. The second approach fastens density to a point in the attribute space. It is described by the algorithm DENCLUE that is much less affected by data dimensional.

V. Comparing Algorithms Using RTOOL

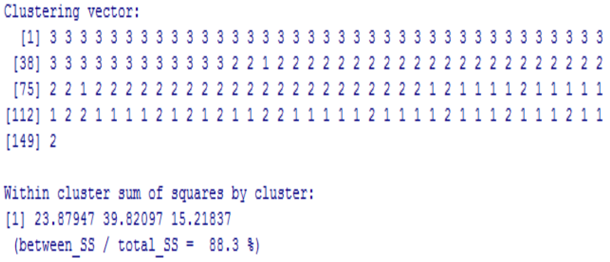
We got IRIS data from web and used R-tool to visualize the effect of algorithms.

a) K-means clustering algorithm-



```
K-means clustering with 3 clusters of sizes 38, 62, 49

Cluster means:
      X5.1     X3.5     X1.4     X0.2
1 6.850000 3.073684 5.742105 2.071053
2 5.901613 2.748387 4.393548 1.433871
3 5.004082 3.416327 1.465306 0.244898
```



```
Clustering vector:
  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [38] 3 3 3 3 3 3 3 3 3 3 3 3 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [75] 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 1 1 1 2 1 1 1 1 1
[112] 1 2 2 1 1 1 1 2 1 2 1 2 1 1 2 2 1 1 1 1 1 2 1 1 1 1 2 1 1 1 2 1 1 1 2 1 1
[149] 2

Within cluster sum of squares by cluster:
[1] 23.87947 39.82097 15.21837
 (between_SS / total_SS =  88.3 %)
```

Figure1: K-means clustering in R-Tool

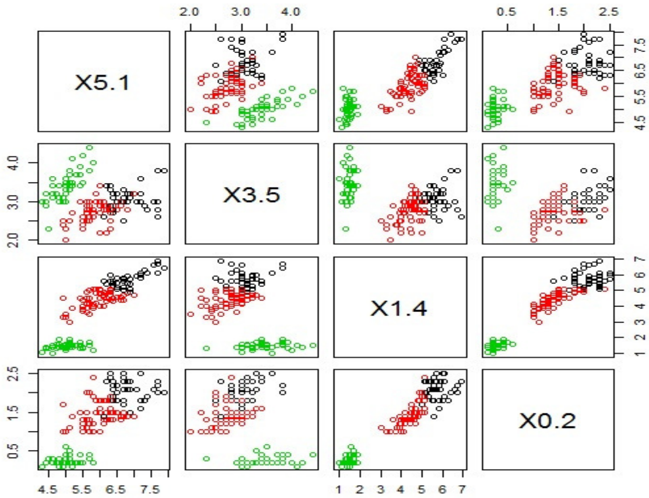
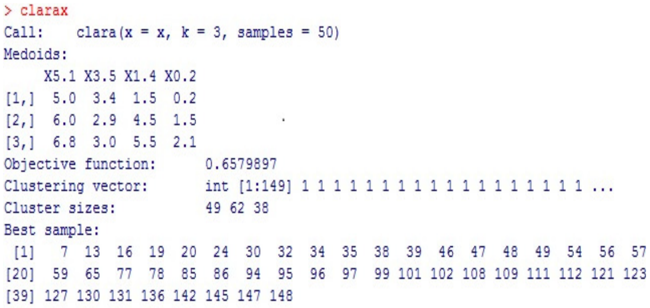


Figure 2: PLOT in R-Tool of K-means clustering

b)K-Medoids clustering algorithm-



```
> clarax
Call:    clara(x = x, k = 3, samples = 50)
Medoids:
     X5.1 X3.5 X1.4 X0.2
[1,]  5.0  3.4  1.5  0.2
[2,]  6.0  2.9  4.5  1.5
[3,]  6.8  3.0  5.5  2.1
Objective function:      0.6579897
Clustering vector:       int [1:149] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ...
Cluster sizes:           49 62 38
Best sample:
 [1]   7  13  16  19  20  24  30  32  34  35  38  39  46  47  48  49  54  56  57
[20]  59  65  77  78  85  86  94  95  96  97  99 101 102 108 109 111 112 121 123
[39] 127 130 131 136 142 145 147 148
```

Figure 3: K-medoids clustering in R-tool

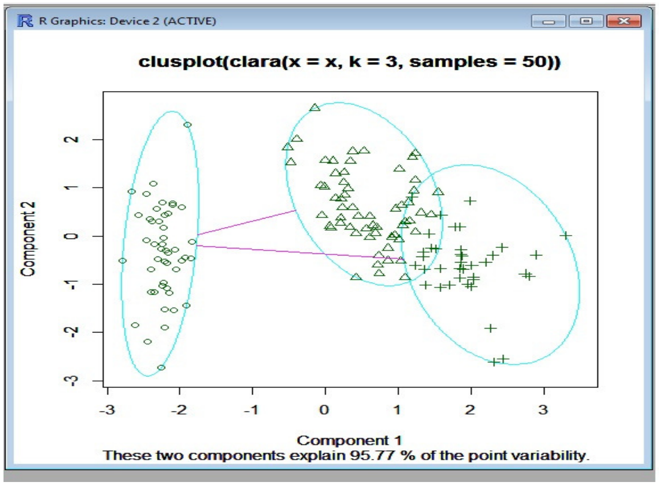


Figure 4: Plot of CLARA in R-tool

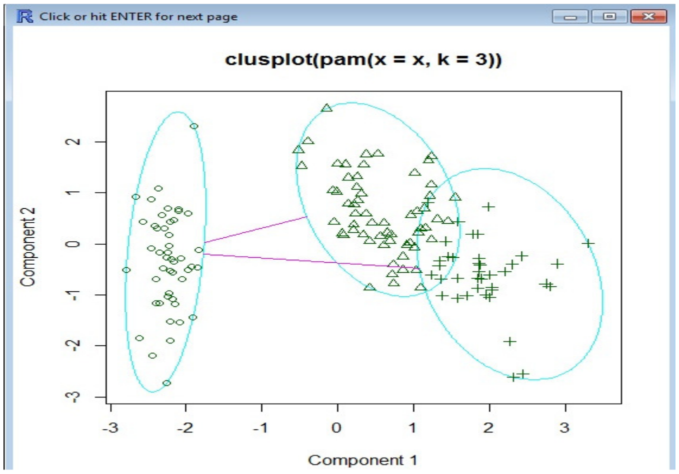


Figure 5: Plot of PAM in R-Tool

Include a great deal of unnecessary information, your paper will likely get rejected or at least be looked upon less favorably.

c)Hierarchical clustering algorithm-

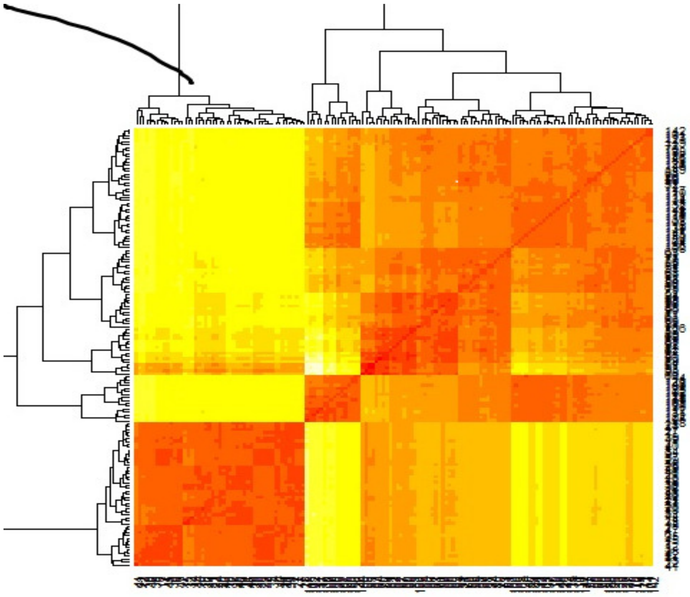


Figure 6: Plot of Hierarchical Clustering in R-Tool

VI. CONCLUSION

In this paper we explored different clustering algorithms. If we lead to winding up then we say Hierarchical clustering algorithm is flexible, efficient with any type of attribute but it doesn't reexamine the clusters once built. K-means is restricted to data for which there is a concept of center lies[16], whereas k-medoids is more robust than k-means in the existence of noise and outliers. Although later is relatively not so much efficient. For both the algorithms: in advance total number of clusters K is needed to be specified. Density-based algorithms are capable of discovering clusters of arbitrary shapes. It also provides an inbuilt protection against outliers. But its inconvenience is the deficiency of interpretability. We got distinct plots when we applied distinct algorithms on the same dataset.

References

[1]. Caiming Zhong ,Duoqian Miao," Minimum spanning tree based split-and-merge: A hierarchical clustering method", Journal of Information Sciences, Volume 181 Issue 16,August 2011, Elsevier ScienceInc.New York,USA,pages:3397-3410.

[2]. Anuradha Awachar, Rajashree Bairagi, Vijayalaxmi Hegade and Mahadev Khandagale, "An Overview of Ontology Based Text Document Clustering Algorithms", International Journal of Computer Sciences and Engineering, Volume-02, Issue-02, Page No (60-64), Feb -2014.

[3]. Xindong Wu · Vipin Kumar · J. Ross Quinlan," Top 10 algorithms in data mining", International Conference on Data Mining (ICDM) in December 2006.

[4]. A. K. Jain and R. C. Dubes. "Algorithms for Clustering Data." Prentice-Hall, Englewood Cliffs, NJ, 1988.

[5]. L. Kaufman and P.J. Rousseeuw. "Finding Groups in Data: An Introduction to Cluster Analysis." Wiley, New York, 1990.

[6]. S.Anitha Elavarasi and Dr. J. Akilandeswari and Dr. B. Sathiyabhama, January 2011, A Survey On Partition Clustering Algorithms.

[7]. F. Murtagh. A survey of recent advances in hierarchical clustering algorithms.Computer Journal, 26(4):354–359, 1983.

[8]. W. Day and H. Edelsbrunner., "Efficient algorithms for agglomerative hierarchical clustering methods". Journal of Classification, 1(7):7–24, 1984.

[9]. S. Guha, R. Rastogi, and K. Shim, 1998. CURE: An Efficient Clustering Algorithm for Large Databases. Proc. ACM Int'l Conf. Management of Data : 73-84.

[10]. J. Hartigan and M. Wong. Algorithm as136: A k-means clustering algorithm. Applied Statistics, 28:100–108, 1979.

[11]. Kilian Stoffel and Abdelkader Belkoniene "Parallel k/h-Means Clustering for Large Data Sets", P. Amestoy et al. (Eds.): Euro-Par'99, LNCS 1685, pp. 1451{1454, 1999.c Springer-Verlag.

[12]. Zha, H., Ding, C., Gu, M., He, X., & Simon, H. (2002)"Spectral relaxation for K-means clustering." Advances in Neural Information Processing Systems 14 (NIPS'01),1057–1064.

[13]. Raymond T. Ng and Jiawei Han.," CLARANS: A Method for Clustering Objects for Spatial Data Mining. "IEEE Transactions on Knowledge and Data Engineering, 14(5):1003{1016, 2002.

[14]. L. Kaufman and P. J. Rousseeuw. "Finding Groups in Data: an Introduction to Cluster Analysis". John Wiley & Sons,1990

[15]. R. T. Ng and J. Han. ,"Efficient and Effective clustering methods for spatial Data Mining", Proc. of the 20th Int'l Conf.on Very Large Databases, Santiago, Chile, pages 144–155,1994.

[16]. D.Napoleon , P.Ganga Lakshmi," An Enhanced k-means algorithm to improve the Efficiency Using Normal Distribution Data Points ",(IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 07, 2010, 2409-2413.

Author Profile

Swati Joshi, completed B.E. in Computer Science & Engg. in 2005. She is pursuing M.tech in Computer Engg. from Amity University under guidance of Mr. Farhaat Ullah Khan who is II author of this paper.

Farhat Ullah Khan
He is Assistant Professor in ASET,Amity University, Noida(U.P.), India.

Narina Thakur
She is a PhD scholar and working as Associate Professor in Bhartiya Vidhyapeeth Engg. College, New Delhi