ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 4, April 2013

Cogitative Analysis on K-Means Clustering Algorithm and its Variants

Kavitha Karun A[1], Elizabeth Isaac[2]

M.Tech Scholar, Dept of CSE, Rajagiri School of Engineering and Technology, Kochi, India[1]

Asst. Professor, Dept of CSE, Rajagiri School of Engineering and Technology, Kochi, India[2]

Abstract: Rapid advancements in science and technology resulted in the accumulation of enormous amount data. This facilitated the need for new methods for extracting essential data from these huge bulks of data since the old method of query ing proved to be inadequate. As a result many data analysis methods came in to existence and Cluster analysis is one among them. Cluster analysis has found its application in almost all fields especially in Bioinformatics, Image processing, Pattern Recognition etc. Cluster analysis or clustering can be defined as the process of grouping up of data objects in to different sets. It is done in such a way that the objects in the same group exhibit similar properties. There are several clustering algorithms available. The most widely used and popular clustering algorithm is the k-means clustering algorithm. This paper focuses on a survey of k-means clustering algorithm and its variants.

Keywords: Clusters, Cluster Analysis, k-means, k-modes, k-medoids

I. INTRODUCTION

Clustering is the process of assigning objects in to groups such that each object in a group or cluster exhibits some similarities with other members in the same group or cluster. It is considered as an unsupervised learning method. The basic idea behind clustering lies in defining the distance between two data points. Intra-cluster distance denotes the sum of distances between objects in the same cluster and Inter-cluster distance denotes the distances between different clusters. There are different measures for evaluating a cluster. Internal measures of evaluation deals with inter/intra cluster distance where as External measures evaluates the representativeness of the current clusters to true classes or original classes. A good clustering is one in which Intra-cluster distance is minimum and Inter-cluster distance is maximum.

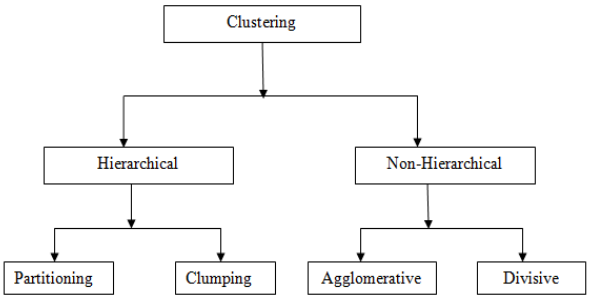

Figure 1. Various Clustering Techniques

Figure 1 shows the various clustering methods. Clustering methods can be broadly classified in to Hierarchical clustering and Non Hierarchical clustering based on the cluster structure produced. The Non Hierarchical methods divide the data set in to different groups with or without overlapping. They are again classified in to Partitioning method and Clumping method. Hierarchical methods produce clusters with nested structure. They are further divided in to Agglomerative method and Divisive method. Categorizations of clustering algorithms are done based on the cluster models produced by them, what constitute a cluster and methods to identify them. Connectivity based clustering, Centroid based clustering, Distribution based clustering and Density based clustering are the major classifications**.**

The rest of the paper is organised as follows: Section II summarise the original k-means algorithm for clustering. Different approaches to improve the k-means are presented in section III. K-modes and K-medoids algorithms are presented in section IV. A discussion on the various techniques and its summary is presented in Section V. The paper is concluded in Section VI.

I. K-MEANS CLUSTERING ALGORITHM

K-means clustering algorithm comes under centroid based clustering where each cluster is represented by a single mean vector which may not necessarily be a member of the data set. In this algorithm the number of clusters to be formed is fixed to 'k'. Then find the k cluster

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 4, April 2013

centers. After that, assign the objects to the nearest cluster center in such a way that each object will be assigned to one and only one cluster. It gives an approximation to an NP-hard combinatorial optimization problem. It is an unsupervised algorithm. Input given to the algorithm is "k" which stands for the number of clusters the user intended to have. From a set of data points or observations, k-means attempts to classify them into k clusters. The algorithm is iterative in nature. If $d_1,\ldots d_n$ are data points or vectors or observations, then by k-means each observation will be assigned to one and only one cluster. X (i) denotes cluster number for the i^{th} observation. Distance between the data points and centroids are calculated by Euclidean distance. K-means minimizes within-cluster point scatter.
The k-means clustering algorithm consists of (i) finding the initial centroids and (ii) assigning each data point or observation to an appropriate cluster. In the original k-means algorithm the centroids for each cluster is selected randomly. After fixing the centroids, then assign each observation to the nearest centroid based on the distance between the data points and the initial centroids. This forms the initial partition. The centroids for each cluster are then calculated since the centroids may change due to the inclusion of new data points. Same procedure is repeated for including the datapoints. The process is continued until the centroids will never change. This is the convergence criterion for k-means.

Pseudo code for the k-means clustering algorithm is listed as follows [1,7]

Algorithm 1: The k-means clustering algorithm

Input:
D = {d1, d2,.....,dn} //set of n data items.
k // Number of desired clusters

Output:
A set of k clusters.

Steps:
1. Arbitrarily choose k data-items from D as initial centroids;

2. **Repeat**

Assign each item di to the cluster which has the closest centroid;
Calculate new mean for each cluster;

Until convergence criteria is met.

The k-means clustering algorithm is cons
to be an efficient algorithm in clustering. But it has
drawbacks. The major problem lies in the process of randomly choosing the initial centroids. Quality of final clusters is greatly influenced by the choice of these initial centroids. So several attempts were made to increase the efficiency of k-means algorithm.

II. MODIFIED APPROACH TO K-MEANS

This section summarizes different approaches to enhance the efficiency of k-means. In the first approach the two phases of original k-means algorithm is enhanced to improve the overall performance. In the first phase instead of randomly choosing initial centroids they are calculated by using a systematic approach [1,12]. In the second phase, a different version of the approach in [1,4] is used. In this, the data points are assigned to the nearest clusters based on their relative distance and they are subsequently fine tuned using a heuristic approach.

Pseudo code for the modified k-means algorithm is listed as Algorithm 2 [1].

Algorithm 2: The enhanced method

Input:
D = {d1, d2,.....,dn} // set of n data items
k // Number of desired clusters

Output:
A set of k clusters.

Steps:
Phase 1: Determine the initial centroids of the clusters by using Algorithm 3[1].

Phase 2: Assign each data point to the appropriate clusters by using Algorithm 4[1].

Algorithm 3: Finding the initial centroids

Input:
D = {d1, d2,.....,dn} // set of n data items
k // Number of desired clusters

Output:
A set of k initial centroids .

Steps:

1. Set m = 1;

2. Compute the distance between each data point and all other data- points in the set D;

3. Find the closest pair of data points from the set D and form a data-point set Am (1<= m <= k) which contains these two data- points, Delete these two data points from the set D;

4. Find the data point in D that is closest to the data point set Am, Add it to Am and delete it from D;

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 4, April 2013

5. Repeat step 4 until the number of data points in Am reaches 0.75*(n/k);

6. If m<k, then m = m+1, find another pair of data points from D between which the distance is the shortest, form another data-point set Am and delete them from D, Go to step 4;

7. For each data-point set Am (1<=m<=k) find the arithmetic mean of the vectors of data points in Am, these means will be the initial centroids.

Algorithm 4: Assigning data-points to clusters

Input:
D = {d1, d2,.....,dn} // set of n data-points.
C = {c1, c2,....., ck} // set of k centroids

Output:
A set of k clusters

Steps:

1. Compute the distance of each data-point di (1<=i<=n) to all the centroids cj (1<=j<=k) as d (di, cj);

2. For each data-point di, find the closest centroid cj and assign di to cluster j.

3. Set ClusterId[i] =j; // j: Id of the closest cluster

4. Set Nearest_Dist[i] = d (di, cj); 5. For each cluster j (1<=j<=k), recalculate the centroids;

5. For each cluster j (1<=j<=k), recalculate the centroids**;**

6. Repeat

7. For each data-point di

7.1 Compute its distance from the centroid of the present nearest cluster;

7.2 If this distance is less than or equal to the present nearest distance, the data-point stays in the cluster;

Else

7.2.1 For every centroid cj (1<=j<=k) Compute the distance d (di, cj);
Endfor;

7.2.2 Assign the data-point di to the cluster with the nearest centroid cj

7.2.3 Set ClusterId[i]=j;

7.2.4 Set Nearest_Dist[i]= d(di, cj);
Endfor;

8. For each cluster j (1<=j<=k), recalculate the centroids;

Until the convergence criteria is met.

In Algorithm 3, the centroids are determined in a systematic way. First step is to calculate the distance between each data-point or observation to all other observations in the same set. The second step is to find the closest pair in the set by using the calculated distance in the step 1.This pair forms the set A1. Then find an observation that is closest to A1 in the same way and add it to A1. Continue adding observations or data points to A1 until the number of observations in it reaches threshold value ie.0.75 *(n/k). Again go to step 2 and find another closest pair A2. Continue the same procedure and add observations to A2 until the threshold. The same procedure is repeated until k such sets of observations are formed. The last step in the algorithm is to find the initial centroids and the arithmetic mean of observations in each set forms the initial centroid. Thus' k' initial centroids are obtained. Euclidean distance is used for distance computation.

In Algorithm 4, the first step is to find out the distance between each data-point and the initial centroids of all the clusters determined in Algorithm 3. Initial grouping of the data points are done by assigning the data points to clusters having the nearest centroid. This 'nearness' is found by using Euclidean distance. While assigning the data-point to a cluster the ClusterId i.e. the cluster to which it is assigned and Nearest_Dist i.e the distance of the data point from its nearest cluster centroid is noted. The centroids are recalculated by taking the mean of data points since the inclusion of data-points may change the centroids. Then a heuristic approach is used to improve the efficiency. The distance between each data-point and new centroid of its present cluster is noted. If this distance is less than the Nearest_Dist ,it shows that data-point stays in the cluster itself and no need to compute the distance between other cluster centroids. On the other hand if this distance is large compared to Nearest_Dist then it needs to compute distance between the data-point and all other cluster centroids. Then reassign the data point to the cluster having the nearest centroid. This procedure is repeated until no more data-points move to other clusters, this indicates the convergence criterion. This heuristic approach improves the efficiency by the significant reduction in the number of computations.

III. K-MEDOIDS AND K-MODES ALGORITHMS

A. The k-medoids algorithm:

The k-medoids algorithm is a clustering algorithm which is related to the k-means algorithm and the medoid shift algorithm. K-medoids algorithm is based on partition

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 4, April 2013

clustering. In k-medoids also the value of 'k'(the number of clusters needed) should be given in advance as in k-means. K-medoids uses objects called medoids to represent a cluster. A medoid can be defined as the most centrally located data object in a cluster. This can be achieved by finding the distance between objects. In k-medoids algorithm k data objects are selected randomly as medoids for k clusters. Then the other data objects are assigned to clusters having medoid nearest to that data object. After this initial partition new medoids are determined such that it can represent the cluster in a better way and the process is repeated. In every iteration the position of this medoid is changed. The entire process is repeated until no medoid will change. The most common realization of k-medoids clustering is Partitioning Around Medoids (PAM) algorithm [8]. The algorithm starts from an initial set of medoids. It then iteratively replaces a medoid by a non-medoid if that swapping improves the total distance of the resultant clustering.

Pseudo code for PAM algorithm is listed as follows [8].

Algorithm 5: The PAM algorithm

Input:
k //number of clusters
n // number of data objects

Output:
A set of k clusters

Steps:

1. Randomly select k of the n data points as the medoids;

2. Associate each data point to the closest medoid.

3. For each medoid m and each data point 'o' associated to m, swap 'm and o' and compute the total cost of the configuration (that is, the average dissimilarity of 'o' to all the data points associated to 'm'). Select the medoid 'o' with the lowest cost of the configuration .

4. Repeat alternating steps 2 and 3 until there is no change in the assignments.

B. The k-modes algorithm:

The k-modes clustering algorithm is a variant of k-means clustering algorithm which replaces the means of clusters by modes. K-modes clustering algorithm closely resembles k-means but it removes the numeric data limitation of k-means while preserving its efficiency. K-modes cam cluster categorical data. It eliminated the limitation imposed by k-means through following modifications [5]:

• For categorical data objects k-modes used a simple matching dissimilarity measure or hamming distance.

• It replaced the means of clusters by their modes.

The dissimilarity measure between two categorical objects X and Y described by m categorical attributes can be defined by the total mismatches of the corresponding attribute categories of the two objects. The objects are said to be more similar if the number of mismatches is less. This measure is often referred to as simple matching [3].

$$d_1(X,Y) = \sum_{k=1}^{m} \delta(x_j , y_j) \quad (1)$$

where

$$\delta(x_j , y_j) = \begin{cases} 0 & x_j = y_j \\ 1 & x_j \neq y_j \end{cases} \quad (2)$$

Pseudo code for k-modes algorithm is listed as follows [3].

Algorithm 6: The k-modes algorithm

Steps:

1. Select k initial modes, one for each cluster.

2. Allocate an object to the cluster whose mode is the nearest to it according to (1). Update the mode of the cluster after each allocation according to Theorem 1[3].

3. After all objects have been allocated to clusters, retest the dissimilarity of objects against the current modes. If an object is found such that its nearest mode belongs to another cluster rather than its current one, reallocate the object to that cluster and update the modes of both clusters.

4. Repeat 3 until no object has changed clusters after a full cycle test of the whole data set.

There is no proof for the convergence of this algorithm. But from its practical applications it is evident that the algorithm always converges [3].

V. DISCUSSION

Section II describes the original k-means algorithm. It is well known for its efficiency in clustering large data sets. But the random selection of initial centroids is a drawback. Another limitation of k-means is that it works only on numerical data. Section III

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 4, April 2013

elaborates the modified approach to k-means where the initial centroids are determined in a systematic way and thereby increasing he accuracy. Moreover, the heuristic approach reduced the amount of computations. Experimental results show that the enhanced method outperforms the original k-means in terms of accuracy.The HCV (hepatitis c virus) data set is used for testing the accuracy and efficiency of the enhanced algorithm. The same data set is given as input to both standard k-means algorithm and the enhanced algorithm. The number of clusters (k) is taken as 2. The original k-means algorithm requires the values of the initial centroids also as input, apart from the input data values and the value of k [1]. The experiment is conducted several times for different sets of values of the initial centroids, which are selected randomly. The enhanced method produces more accurate clusters compared to the original k-means algorithm. In some cases the original k-means algorithm put all the data points in the same cluster. But the enhanced algorithm always produces the same result.

The enhanced algorithm proved better when input data points are normally distributed [6]. In section IV, a brief description of k-modes and k-medoids algorithm is done. K-means is prone to outliers and in some cases set of objects that are close to a particular centroid may be empty, in such cases the centroid cannot be updated. For these reasons k-medoids algorithms are sometimes used instead of k-means. In k-medoids, each cluster is represented by a representative object called a medoid instead of a centroid. The k-means algorithm cannot cluster categorical data and domains that contain mixed numerical and categorical values. The k-modes algorithm removed this limitation of k-means by using a simple dissimilarity measure or hamming distance by preserving the same efficiency as k-means. Table 1 summarises the comparison of the above methods with standard k-means.

The computation complexity of k-means is O(nkl) where n is the number of input data-points, k is the number of clusters and l is the number of iterations. Computation complexity of k-modes algorithm is O(kn(m+M)) where m is the number of attributes, M is the total number of categories of all attributes, n is the total number of data objects and k is the number of clusters. K-means is more scalable and efficient compared to other algorithms but it is sensible to outliers. K-medoids algorithm is more robust to noise and outliers but it is costly and less efficient than k-means. K-modes algorithm can cluster categorical data with the same efficiency as K-means.

VI. CONCLUSION

K-means clustering algorithm is an efficient algorithm in clustering large amount of data. K-means algorithm and its variants are discussed in this paper. A major limitation observed is that in k-means and in all its variants the number of clusters needed should be given as input. The algorithms will be more efficient if the number of clusters is determined by some statistical methods based on input distribution. This paper compares the different variants of k-means and their features are explained in detail. So it will be of great use to select the appropriate algorithm for different applications.

TABLE I

COMPARISON OF K-MEANS AND ITS VARIANTS

Algorithms	**Computation complexity**	**Characteristics**	**Strength**	**Weakness**
k-means	O(nkl)	Centroid based	Relatively scalable and Efficient. Easy to understand and implement	Sensitive to outliers and can cluster only numerical values
Enhanced k-means	$O(n^2)$	Centroid based	Relatively more accurate	Sometimes takes more time compared to k-means
k-medoids	$O(l\ k(n-k)^{2)}$	Based on medoids	More robust to noise and outliers	Relatively costly and not so efficient
k-modes	O(k n(m+M))	Based on modes	Same efficiency as that of k-means and can cluster categorical data	Not so accurate

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 4, April 2013

REFERENCES

[1] K. A. Abdul Nazeer and M. P. Sebastian, "Improving the Accuracy and Efficiency of The k-means Clustering Algorithm", Proceedings of the World Congress on Engineering , Vol I WCE 2009, London, U.K, July 1 - 3, 2009.

[2] H Park, J S Lee and C H Jun,"A K-means-like Algorithm for k-medoids Clustering and Its Performance",Proceedings of the 36th CIE Conference on Computers and Industrial Engineering,Taiwan, 2006.

[3] Z. Huang, "Extensions to the k-means Algorithm for Clustering Large Data Sets with Categorical Values", Data Mining and Knowledge Discovery, 1998 – Springer.

[4] A.M Fahim, A. M Salem, A Torkey and M. A Ramadan, "An Efficient enhanced k-means clustering algorithm," Journal of Zhejiang University, 10(7):1626–1633, 2006.

[5] S.S Khan and Dr. Shri Kant, "Computation of Initial Modes for K-modes Clustering Algorithm using Evidence Accumulation", International Joint Conference on Artificial Intelligence-07.

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

[7] M. H. Dunham, "Data Mining- Introductory and Advanced

Concepts", Pearson Education, 2006.

[8] E.M. Mirkes, K-means and K-medoids applet. University of Leicester, 2011.

[9] Shalini Singh, N C Chauhan," K-means v/s K-medoids: A Comparative Study", National Conference on Recent Trends in Engineering & Technology.

[10] J .C .A Chaturvedi , P Green, " K-modes clustering" , J classification, (18):35–55, 2001.

Daxin Jiang, Chum Tong and Aidong Zhang, "Cluster Analysis for Gene Expression Data," IEEE Transactions on Data and Knowledge Engineering, 16(11): 1370-1386, 2004.

[11] F Yuan, Z. H Meng, H. X Zhang and C. R Dong, "A New Algorithm to Get the Initial Centroids," Proc. of the 3rd International Conference on Machine Learning and Cybernetics, pages 26–29, August 2004.

BIOGRAPHY

Kavitha Karun A is currently pursuing her Master of Technology in Computer Science and Engineering with Specialization in Information Systems at Rajagiri School of Engineering and Technology, Kochi (India) under M.G University. She has worked as Asst. Professor in Lourdes Matha College of Science and Technology.