

Enhancing K-Means Clustering Algorithm

¹M.V.B.T.Santhi, ²V.R.N.S.S.V.Sai Leela, ³P.U.Anitha, ⁴D.Nagamalleswari

¹Dept. of IST, KLUUniversity, India

²Dept. of CSE, GITAM University, India

^{3,4}Dept. of CSE, KLUUniversity, India

Abstract

Clustering is the process of organizing data objects into a set of disjoint classes called clusters. Clustering is an example of unsupervised classification. Cluster analysis seeks to partition a given data set into groups based on specified features so that the data points within a group are more similar to each other than the points in different groups. Cluster analysis is one of the primary data analysis methods and k-means is one of the most well known popular clustering algorithms.

The k-means algorithm is one of the frequently used clustering methods in data mining, due to its performance in clustering massive data sets. The final clustering result of the k-means clustering algorithm greatly depends upon the correctness of the initial centroids, which are selected randomly.

A new method is proposed for finding the better initial centroids and to provide an efficient way of assigning the data points to suitable clusters with reduced time complexity. The proposed algorithm has the more accuracy with less computational time comparatively original k-means clustering algorithm.

Keywords

Clustering, k-means, time complexity, centroid, data sets

I. Introduction

A. Overview of the Cluster Analysis

Data mining is the process of extracting patterns from data. Data mining is seen as an increasingly important tool by modern business to transform data into business intelligence giving an informational advantage. It is currently used in a wide range of profiling practices, such as marketing, surveillance, fraud detection, and scientific discovery.

B. Cluster Analysis

Cluster analysis or clustering is the assignment of a set of observations into subsets (called clusters) so that observations in the same cluster are similar in some sense. Clustering is a method of unsupervised learning, and a common technique for statistical data analysis used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics.

Stages in Clustering

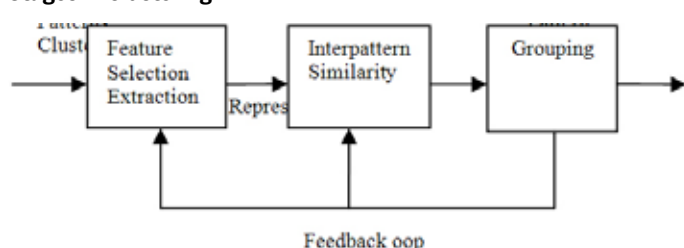


Fig. 1: Different stages in clustering

C. Types of clustering

1. Hierarchical algorithms find successive clusters using previously established clusters. These algorithms usually

are either agglomerative ("bottom-up") or divisive ("top-down"). Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters. Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters.

2. Partitional algorithms typically determine all clusters at once, but can also be used as divisive algorithms in the hierarchical clustering.
3. Density-based clustering algorithms are devised to discover arbitrary-shaped clusters. In this approach, a cluster is regarded as a region in which the density of data objects exceeds a threshold. DBSCAN and OPTICS are two typical algorithms of this kind.
4. Subspace clustering methods look for clusters that can only be seen in a particular projection (subspace, manifold) of the data. These methods thus can ignore irrelevant attributes. The general problem is also known as Correlation clustering while the special case of axis-parallel subspaces is also known as Two-way clustering, co-clustering or bi-clustering: in these methods not only the objects are clustered but also the features of the objects, i.e., if the data is represented in a data matrix, the rows and columns are clustered simultaneously. They usually do not however work with arbitrary feature combinations as in general subspace methods. But this special case deserves attention due to its applications in bioinformatics.

D. Common distance functions

1. The Euclidean distance (also called distance as the crow flies or 2-norm distance). A review of cluster analysis in health psychology research found that the most common distance measure in published studies in that research area is the Euclidean distance or the squared Euclidean distance.
2. The Manhattan distance (aka taxicab norm or 1-norm)
3. The maximum norm (aka infinity norm)
4. The Mahalanobis distance corrects data for different scales and correlations in the variables
5. The angle between two vectors can be used as a distance measure when clustering high dimensional data. See Inner product space.
6. The Hamming distance measures the minimum number of substitutions required to change one member into another.

II. Related Work

A. Existing System

The k-means algorithm takes the input parameter, k, and partitions a set of n objects into k clusters so that the resulting intra-cluster similarity is high but the inter-cluster similarity is low. Cluster similarity is measured in regard to the mean value of the objects in a cluster, which can be viewed as the cluster's centroid or center of gravity.

The k-means algorithm proceeds as follows.

Algorithm : The k-means clustering algorithm

Require: $D = \{d_1, d_2, d_3, \dots, d_i, \dots, d_n\}$ // Set of n data points.

k // Number of desired clusters

Ensure: A set of k clusters.

Steps:

1. Arbitrarily choose k data points from D as initial centroids;
2. Repeat

Assign each point d_i to the cluster which has the closest centroid;

Calculate the new mean for each cluster;

Until convergence criteria is met.

Advantages

Simplicity

Speed

Disadvantages

Does not yield the same result with each run, since the resulting clusters depend on the initial random assignments

Requirement for the concept of a mean to be definable which is not always the case

B. Proposed System

The aim of the proposed algorithm is to improve the computational efficiency of the K Means algorithm. The algorithm involves initial centroid selection, which is done randomly in existing algorithm. Hence we propose an algorithm which selects initial centroids based on the distances calculated from the origin.

One of the most popular clustering algorithms is k-means clustering algorithm, but in this method the quality of the final clusters rely heavily on the initial centroids, which are selected randomly. Moreover, the k-means algorithm is computationally very expensive also. The proposed algorithm is found to be more accurate and efficient compared to the original k-means algorithm. This proposed method finding the better initial centroids and provides an efficient way of assigning the data points to the suitable clusters. This method ensures the total mechanism of clustering in $O(n \log n)$ time without loss the correctness of clusters. This approach does not require any additional inputs like threshold values. The proposed algorithm produces the more accurate unique clustering results. The value of k , desired number of clusters is still required to be given as an input to the proposed algorithm.

III. System Design

A. Overview of the Enhanced Clustering Algorithm

In this algorithm, two methods are used, one method for finding the better initial centroids. And another method for an efficient way for assigning data points to appropriate clusters. The method used for finding the initial centroids computationally expensive. We proposed a new approach for finding the better initial centroids with reduced time complexity.

In the proposed algorithm first we are checking, the given data set contain the negative value attributes or not. If the data set contains the negative value attributes then we are transforming the all data points in the data set to the positive space by subtracting the each data point attribute with the minimum attribute value in the given data set. Here, the transformation is required, because in the proposed algorithm we calculate

the distance from origin to each data point in the data set. So, for the different data points. we will get the same Euclidean distance from the origin. This will result in incorrect selection of the initial centroids.

To overcome this problem all the data points are transformed to the positive space. Then for all the data points, we will get the unique distances from origin. If data set contains the all positive value attributes then the transformation is not required. In the next step, for each data point we calculate the distance from origin. Then, the original data points are sorted accordance with the sorted distances. After sorting partition the sorted data points into k equal sets. In each set take the middle points as the initial centroids. These initial centroids lead to the better unique clustering results. Next, for each data point the distance calculated from all the initial centroids. The next stage is an iterative process which makes use of a heuristic approach to reduce the required computational time. The data points are assigned to the clusters having the closest centroids in the next step. ClusterId of a data point denotes the cluster to which it belongs. NearestDist of a data point denotes the present nearest distance from closest centroid.

Next, for each cluster the new centroids are calculated by taking the mean of its data points. Then for each data point the distance calculated from the new centroid of its present nearest cluster. If this distance is less than or equal to the previous nearest distance, then the data point stays in the same cluster, otherwise for each data point we need to calculate the distance from all centroids. After calculated the distances, the data points are assigned to the appropriate clusters and the new ClusterId's are given and new NearestDist values are updated. This reassigning process is repeated until the convergence criterion is met.

Algorithm : The Enhanced Method

Require: $D = \{d_1, d_2, d_3, \dots, d_i, \dots, d_n\}$ // Set of n data points.
 $d_i = \{x_1, x_2, x_3, \dots, x_i, \dots, x_m\}$ // Set of attributes of one data point.

k // Number of desired clusters.

Ensure: A set of k clusters.

Steps

- 1: In the given data set D , if the data points contains the both positive and negative attribute values then go to step 2, otherwise go to step 4.
- 2: Find the minimum attribute value in the given data set D .
- 3: For each data point attribute, subtract with the minimum attribute value.
- 4: For each data point calculate the distance from origin.
- 5: Sort the distances obtained in step 4. Sort the data points accordance with the distances.
- 6: Partition the sorted data points into k equal sets.
- 7: In each set, take the middle point as the initial centroid.
- 8: Compute the distance between each data point d_i ($1 \leq i \leq n$) to all the initial centroids c_j ($1 \leq j \leq k$).
- 9: Repeat
- 10: For each data point d_i , find the closest centroid c_j and assign d_i to cluster j .
- 11: Set ClusterId[i]=j. // j:Id of the closest cluster.
- 12: Set NearestDist[i]= $d(d_i, c_j)$.
- 13: For each cluster j ($1 \leq j \leq k$), recalculate the centroids.
- 14: For each data point d_i ,
- 14.1 Compute its distance from the centroid of the present nearest cluster.

14.2 If this distance is less than or equal to the present nearest distance, the data point stays in the same cluster.

Else

14.2.1 For every centroid c_j ($1 \leq j \leq k$) compute the distance $d(d_i, c_j)$.

End for;

Until the convergence criteria is met.

IV. Screen Shots

The algorithm can be applied on the datasets which contain real data items. The non real data items are ignored and skipped while executing the algorithm on the dataset. The various datasets on which this algorithm is applied are

- Iris Data
- Height Weight Data
- Bank Data

The datasets like iris, bank data contains both real and non real data. In this case ,real data like sepal length, petal length is used in the algorithm execution and the non real data like class names is ignored. The paper aim is to cluster the data efficiently in a less time possible. The proposed algorithm also can be applied on the same data items on which the existing kmeans is applied. Thus both the algorithms are applied on the similar datasets. The notable point in this paper is the number of clusters, the data sets needed to be clustered. This value plays key role in deciding the time taken for the clustering to be completed. The observations are done on the datasets by applying both algorithms and clustering them for the same number of clusters. This allows accuracy in the results which are considered.

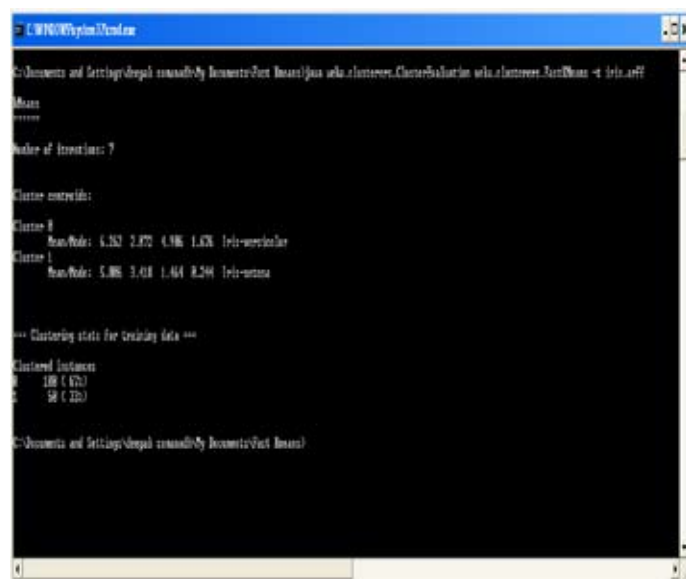


Fig. 2: Proposed k-means on iris.arff



Fig. 3: Proposed k-means on hwt.arff

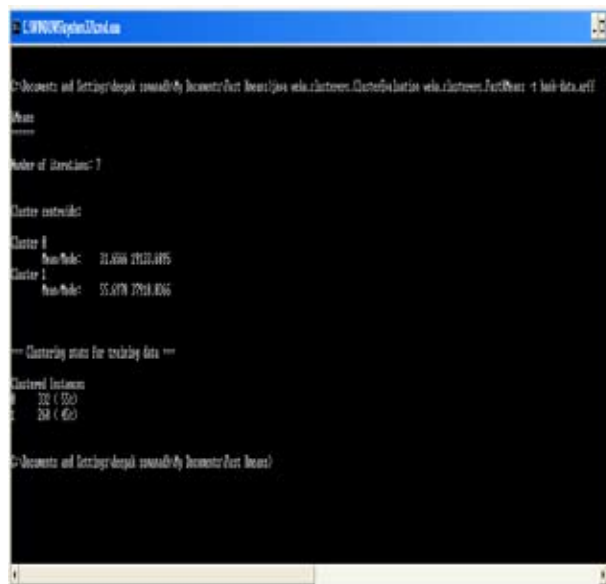


Fig. 4: Proposed k-means on bank-data.arff

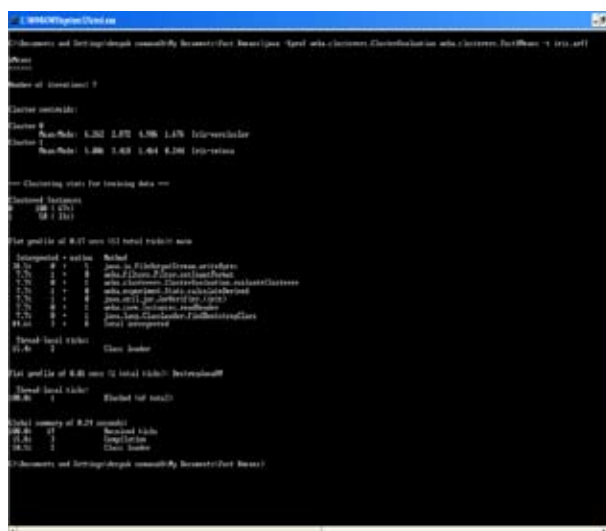


Fig. 5: Proposed Kmeans Time complexity for iris.arff

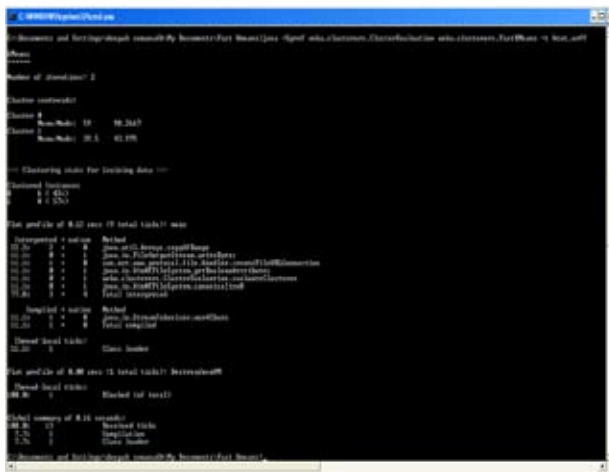


Fig. 6: Proposed Kmeans Time complexity for hwt.arff

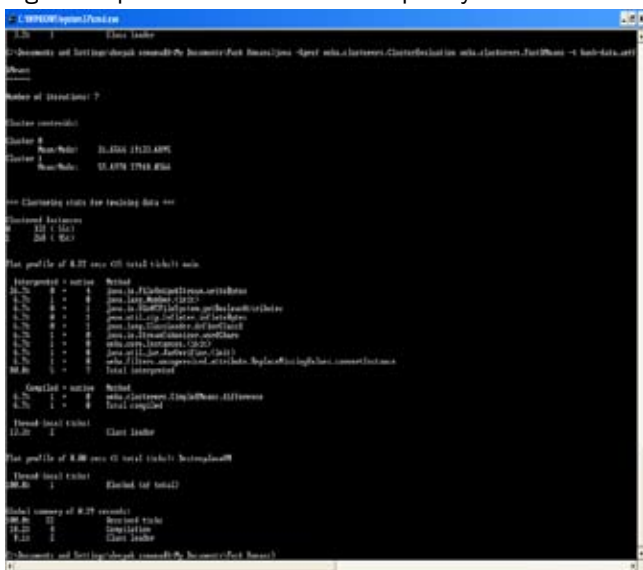


Fig. 7: Proposed Kmeans Time complexity for bank-data.arff

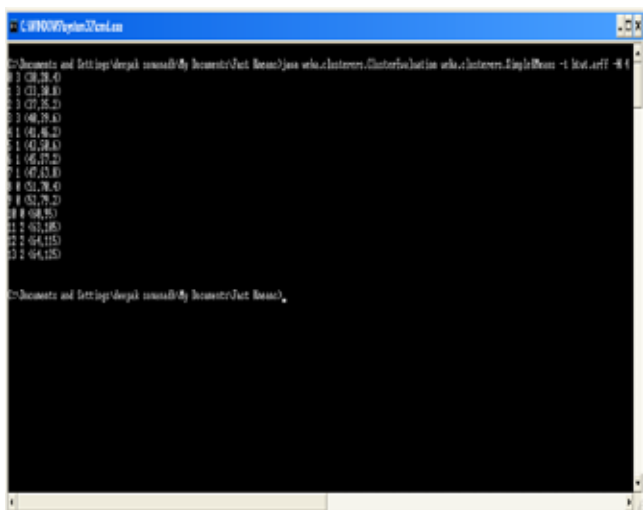


Fig. 8: Four cluster formation using simple kmeans on hwt.arff

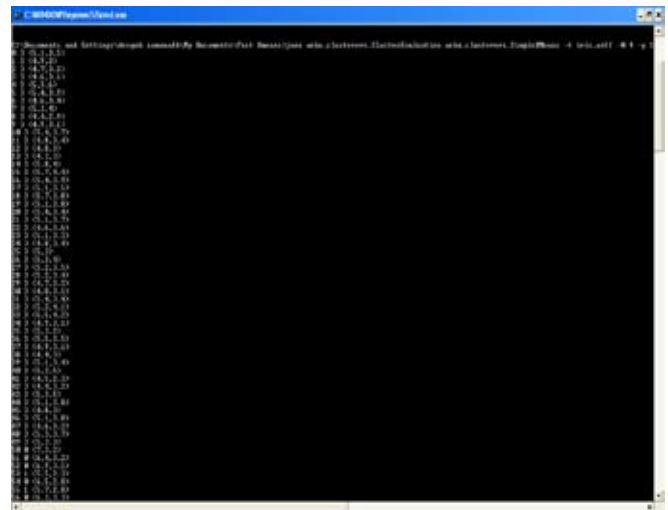


Fig. 9: Four cluster formation using simple kmeans on iris.arff

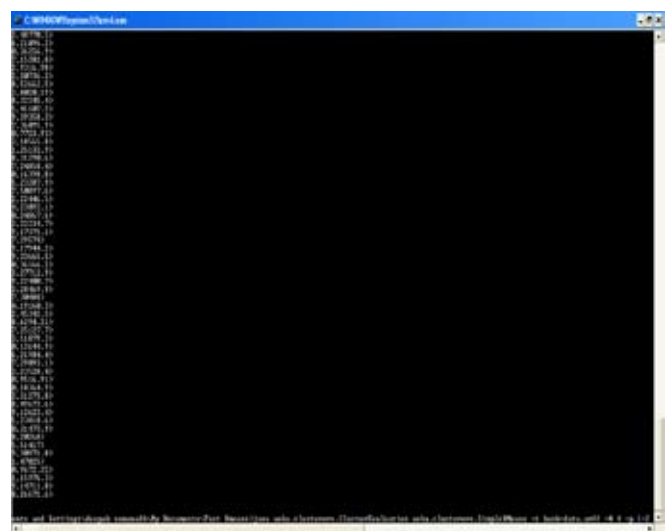


Fig. 10: Four cluster formation using simple kmeans on bank-data.arff

V. Conclusion

One of the most popular clustering algorithms is k-means clustering algorithm, but in this method the quality of the final clusters rely heavily on the initial centroids, which are selected randomly. Moreover, the k-means algorithm is computationally very expensive also. The proposed algorithm is found to be more accurate and efficient compared to the original k-means algorithm. This proposed method finding the better initial centroids and provides an efficient way of assigning the data points to the suitable clusters. This method ensures the total mechanism of clustering in $O(n \log n)$ time without loss the correctness of clusters. This approach does not require any additional inputs like threshold values. The proposed algorithm produces the more accurate unique clustering results. The value of k , desired number of clusters is still required to be given as an input to the proposed algorithm. The proposed algorithm provides better time for execution and with reliable clusters in the dataset clustering.

VI. Future Scope

The input datasets used for this algorithm are arff files. The arff files contains both real and non real data. This paper is mainly intended to real form of data. So, in near future, we can extend this paper on both real and non real forms of data.

References

- [1] A. M. Fahim, A. M. Salem, F. A. Torkey, M. A. Ramadan, "An Efficient enhanced k-means clustering algorithm", journal of Zhejiang University, 10(7): 17271733, 2007.
- [2] K. A. Abdul Nazeer, M. P. Sebastian, "Improving the accuracy and efficiency of the k-means clustering algorithm", in International Conference on Data Mining and Knowledge Engineering (ICDMKE), Proceedings of the World Congress on Engineering (WCE-2009), Vol 1, July 2009, London, UK.
- [3] Chen Zhang, Shixiong Xia, "K-means Clustering Algorithm with Improved Initial center", in Second International Workshop on Knowledge Discovery and Data Mining (WKDD), pp. 790-792, 2009.
- [4] F. Yuan, Z. H. Meng, H. X. Zhang, C. R. Dong, "A New Algorithm to Get the Initial Centroids", proceedings of the 3rd International Conference on Machine Learning and Cybernetics, pp. 27-29, August 2004.
- [5] Koheri Arai, Ali Ridho Barakbah, "Hierarchical K-means: an algorithm for Centroids initialization for k-means", department of information science and Electrical Engineering Politechnique in Surabaya, Faculty of Science and Engineering, Saga University, Vol. 37, No.1, 2007.- 297, 1977.
- [6] A. Bhattacharya, R. K. De, "Divisive Correlation Clustering Algorithm (DCCA) for grouping of genes, detecting varying patterns in expression profiles", bioinformatics, Vol. 24, pp. 1359-1377, 2008.



P.U. Anitha received her B-Tech degree in Computer Science and Information Technology from Jawaharlal Nehru Technological University, Hyderabad, in 2004, M-Tech degree in Software Engineering from Jawaharlal Nehru Technological University, Hyderabad, in 2010, Pursuing Ph.D in Data Mining, JNTUH. She is working as an Assistant Professor in Department of CSE, Koneru Lakshmaiah University, Vaddeswaram, Guntur Dist, AP.



D. Nagamalleswari received her B-Tech degree in Computer Science and Engineering from Jawaharlal Nehru Technological University, Hyderabad, in 2007, M-Tech degree in Computer Science and Engineering from Jawaharlal Nehru Technological University, Kakinada, in 2011. She is working as an Assistant Professor in Department of CSE, Koneru Lakshmaiah University, Vaddeswaram, Guntur Dist, AP.



M.V.B.T. Santhi received her B-Tech degree in Computer Science and Engineering from Jawaharlal Nehru Technological University, Hyderabad, in 2003, M-Tech degree in Computer Science from Archarya Nagarjuna University, in 2010. She is working as an Assistant Professor in Department of Information Science Technology, Koneru Lakshmaiah University, Vaddeswaram, Guntur Dist, AP.



V.R.S.S.N.V. Sai Leela.P received the B.Tech degree in Computer Science and Engineering from Nagarjuna University in 2008, M-Tech degree in Computer Science from Archarya Nagarjuna University, in 2010. She is currently working as an Assistant Professor, with Department of Computer Science and Engineering, GITAM University, Hyderabad, AP.