Volume 3, Issue 7, July 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

An Operative Algorithm for *K*-Means Clustering With New Initial Centroids

Beena P*
Department of CSE
MES college of Engineering
Kuttippuram, Kerala, India

Sunil Kumar P V
Department of CSE
MES college of Engineering
Kuttippuram, Kerala, India

Balachandran K P
Department of MCA
MES college of Engineering
Kuttippuram, Kerala, India

Abstract— ***Organising data into sensible groups is the most fundamental way of understanding and learning. Clustering helps to organise data based on natural grouping without any category labels to identify the clusters. One of the most popular and simplest partitional clustering algorithms is the K-Means published in 1955. K-Means algorithm is computationally expensive and the final clusters depend entirely on the initial selection of centroids. This method also insists the selection of number of clusters initially. Several modifications have been proposed for the K-Means clustering method. Some such proposals are summarised and reviewed. A new method is proposed considering both the standard deviation and mean of the attributes of the dataset.***

Keywords— ***Clustering algorithms, K-means algorithm, Initial centroid, Variation Coefficient, Purity***.

I. Introduction

Advances in scientific data collection methods have resulted in the large scale accumulation of scientific data at various data sources. The amount of information available is becoming enormous and tremendous day by day. It is practically difficult to analyse and interpret the data using conventional methods. Effective and efficient data analysis methods are necessary to extract useful information. Cluster analysis is one of the major data mining methods which helps in identifying the natural groupings and interesting patterns from huge data banks. Data clustering is a process of identifying the natural grouping that exists in a given data-set, such that the patterns in the same cluster are more similar and the patterns in different clusters are less similar. Clustering is a main task of explorative data mining, and a common technique for statistical data analysis used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics. Clustering algorithms are broadly divided into two groups, viz. hierarchical and partitional. Hierarchical clustering algorithms finds the clusters in agglomerative (bottom-up) mode or in divisive (top-down) mode recursively where as partitional clustering algorithms find all the clusters simultaneously as a partition of the data-set. Apart from this, the clustering methods can also be categorized into density-based methods, grid-based methods, model-based methods, etc.

II. *K*-Means Clustering

The most popular, the simplest, efficient partitional clustering method is the *K*-Means clustering method. The given set of data is grouped into *K* number of disjoint clusters, where the value of *K* is fixed in advance. The algorithm consists of two separate phases: the first phase defines *K* initial centroids, one for each cluster. The next phase associates each point of the given data set to the nearest centroid. Generally Euclidean distance is used as the measure to determine the distance between data points and the centroids. Euclidean distance between two data points (x_1, x_2,..., x_n), (y_1, y_2,....y_n) is given by (1).

$$\sqrt{(x_1-y_1)^2+(x_2-y_2)^2+......(x_n-y_n)^2} \quad (1)$$

When all the points are included in some clusters, the first step is completed and an early grouping is done. Now the centroids are recalculated and clustering is done with these new centroids. This is repeated till the centroids do not change. This is the convergence criterion for the *K*-Means. The pseudo code for the *K*-Means clustering is outlined in Algorithm 1[2].

Algorithm 1: The *K*-Means Clustering

```
Input:
        D = $d_1, d_2, ....., d_n$ / 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
   2.1. Assign each item $d_i$ to the cluster which has the closest centroid;
   2.2. Calculate the new mean for each cluster;
3. Until convergence criterion is met.
```

Though the algorithm is effective in producing clusters for many practical applications, there are some draw backs. The computational complexity of the original *K*-Means algorithm is very high, especially for large data sets. The complexity is $O(nKl)$ where n is the number of data points, K the number of clusters and l the number of iterations. Also it produces different types of clusters based on the selection of initial centroids. Accuracy of the final clusters heavily depends on the initial centroids selected.

III. RELATED WORK

Anil K. Jain discussed major challenges and key issues in clustering [1]. He provided a brief overview of clustering and summarized well known clustering methods. He also discussed the major challenges and key issues in designing clustering algorithms, and pointed out some of the emerging and useful research directions, including semi-supervised clustering, ensemble clustering, simultaneous feature selection during data clustering, and large scale data clustering Researches are always been conducted to improve the accuracy and efficiency of the *K* -means algorithm. Some of these innovative approaches to *K*-Means clustering are discussed here. Though the time complexity is not improved, these works could fix the initial centroids and reduced the inconsistency in clustering. Euclidean distance is used in all these works to measure the adjacency between the points.

Fahim et al. [3] proposed an efficient enhanced *K*-means algorithm which refines the second phase of the algorithm. Fahim's approach makes use of a distance function based on a heuristics to reduce the number of distance calculations. As the initial centroids are determined randomly, there is no guarantee for the accuracy of the final clusters.

K. A. Abdul Nazeer et al. [4, 5] proposed an enhanced algorithm by considering relative distance of each point. In the first phase, the initial centroids are determined systematically to produce clusters with better accuracy. The second phase makes use of a variant of the clustering method discussed in [3] that suits for spherical shaped clusters. Though this algorithm produces clusters with better accuracy and efficiency compared to original *K*-means, it is also computationally expensive.

K. A. Abdul Nazeer et al. [6] modified his work in [5] by a heuristic method for finding better initial centroids. He used the second phase of the original *K*-Means without any modification. This method selects an attribute based on which the data points are to be clustered. This attribute has the maximum range among all other attributes.

K. A. Abdul Nazeer et al. [7] refined his work in [6] with a variant of the method in [2] for the second phase. This definitely improved the accuracy, efficiency and complexity.

R.Sumathi et al. [8] suggested a weighted ranking algorithm. Weights to the attributes of data points are assigned by experts. The work is an extension of [7] and produced meaningful clusters. Murat Erisoglu et al. [9] proposed a new method for finding the initial centroids which are well separated. It selects the two attributes that best describe the data set with the help of variation coefficients and correlation coefficients. The second phase of the original *K*-means was used without modification for the clustering. The method produced an improved and consistent cluster structures. The method suggested by Damodar Reddy et al. [10] selects initial centroids with the help of voronoi diagram constructed with the data set. The initial centroids are those points that lie on the boundary of higher radius voronoi circles. The centroids thus generated are the input to the second phase of original *K*-Means.

T.Hithendra Sarma et al. [11] proposed which is a prototype based hybrid approach to speed up the clustering. The data set is partitioned into small clusters each represented by a prototype. These prototypes become the candidate for clustering. A correction is also proposed for the final clusters generated. The method is suitable for high dimensional large data sets.

IV. PROPOSED WORK

The proposed method modifies both phases of the original *K*-Means and is suitable for multivariate data set. The concept of Variation Coefficient [15] is introduced in the process of *K*-Means clustering. Variation Coefficient is a statistical measure of the dispersion of data points in a data series around the mean. It is the standard deviation normalized by the mean.It is calculated by the equation (2).

$$CV = \left| \frac{SX}{\overline{X}} \right| \tag{2}$$

where X denotes the data point, SX the standard deviation of data set and $\overline{X}$ the mean. Earlier the method proposed in [9] used this measure along with correlation coefficient for the initialization process.

This proposal selects the attribute with highest Variation Coefficient based on which clustering is performed. This is a modification to the method proposed in [7]. The proposed Operative *K*-Means is outlined in Algorithm 2. Algorithm 3 initialises the centroids and Algorithm 4 performs the clustering.

Algorithm 2: The Operative Clustering Algorithm

```
Input:
        D = d_1, d_2, …, d_n// set of n data items.
        K // Number of desired clusters.
Output:
        A set of K clusters.
Steps:
        1. Determine the initial centroids of the clusters by using Algorithm 3;
        2. Assign the data points to the clusters by using Algorithm 4.
```

Algorithm 3 selects the attribute based on which clustering is to start with. It selects the attribute with maximum Variation Coefficient and the data set is sorted based on the selected attribute. Sorting can be done with any of the time effective method. Any suitable data structure can also be used as in [12].The sorted dataset is partitioned into *K* equally sized sets. The mean of these *K* sets form the initial centroids. The pseudo code for Algorithm 3 is given below.

Algorithm 3: Finding the initial centroids

```
Input:
        D = d_1, d_2… , d_n // set of n data items.
        K // Number of desired clusters.
Output:
        A  set of K initial centroids.
Steps:

        1. For each column of the data set, determine the Variation Coefficient.
        2. Identify the column with the maximum Variation Coefficient;
        3. Sort the entire data set in non-decreasing order based on the column with maximum Variation Coefficient;
        4. Partition the sorted data set into K equal parts;
        5. Determine the mean of each part obtained in Step 4; these means will be the initial centroids.
```

Algorithm 4 [7] performs the second phase of clustering with the initial centroids obtained from Algorithm 3. This algorithm assigns data points to the cluster with the most adjacent centroid. The cluster no and distance associated with the data point are stored as an attribute. In the succeeding iterations this attribute becomes a measure to shift the data point from one cluster to another. In each iteration, if the distance of the data point from the closest cluster centroid is less than the distance attribute, the data point is moved and the relevant attributes are updated. The process is repeated until the convergence is met. The algorithm is outlined as Algorithm 4.

Algorithm 4: Assigning data points to clusters

```
Input:
        D = d_1, d_2… , d_n // set of n data items.
        C = c_1, c_2, … , c_k // set of K centroids.
Output:
        A set of K clusters.
Steps:
        1. Compute the distance of each data point to all the centroids.
        2. For each data point d_i, find the closest centroid c_j and assign d_i to cluster j;
        3. Set ClusterId[i]= j; // j:Id of the closest cluster;
        4. Set NearestDist[i]= d(d_i, c_j);
        5. For each cluster recalculate the centroids
        6. Repeat
        7. For each data point d_i,
               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 compute the distance d(d_i, c_j);
                       7.2.2   Assign the data point d_i to the cluster with the nearest centroid c_j;
                       7.2.3   Set ClusterId(i)= j;
                       7.2.4   Set NearestDist[i]= d(d_i, c_j);
        8. Endfor;
        9. For each cluster recalculate the centroids;
        10. Until the clusters do not change.
```

V. Experiments and Results

The proposed algorithm along with the works mentioned in [2, 6, 7, 9] were implemented, tested and compared. Implementations were done in Java. The data sets available in the UCI data repository were used for testing. Iris is Iris plants database. Spambase is a spam e-mail database. BCW is the breast cancer Wisconsin (original) data set. Leaf data is leaf species data set. Details of the data sets used are summarized in Table 1.

TABLE I
DATA SETS

Dataset	No of	No of Attributes	No of
Iris	150	4	3
Spambase	4601	57	2
BCW	699	9	2
LeafDataMar	1600	64	100

Purity[16] of the clusters is used as the measure for testing the efficiency. It's a measure of correctly classified data points. Purity of a cluster is defined by equation (3)

$$Purity(C_j) = \frac{1}{|C_j|} \max |C_j|_{class=i} \tag{3}$$

where $|C_j|_{class=i}$ denotes the number of items of class i assigned to cluster j. Overall cluster purity is given by equation(4).

$$Purity = \sum_{j=1}^{k} \frac{|C_j|}{|D|} Purity(C_j) \tag{4}$$

The results vary with data sets and are tabulated in table 2. For the original *K*-Means results of three executions are averaged and tabulated. The results show that the proposed Operative *K*-Means fixes the initial centroids in an efficient way. Fig. 1 shows the graphical view of the results.

TABLE II
COMPARISON

Algorithm	Purity			
	Iris	Spambase	BCW	Leaf Data Mar
K –Means[2]	81.56	63.6	96.1	59.9
K -Means with Heuristic[6]	88.67	67.88	97.5	58.19
Heuristic *K* -Means [7]	92	63.6	96.19	36.38
Variation and Correlation Coefficient *K*-	88.67	63.6	96.05	51.69
Operative *K*-means	94	67.94	97.5	69.75

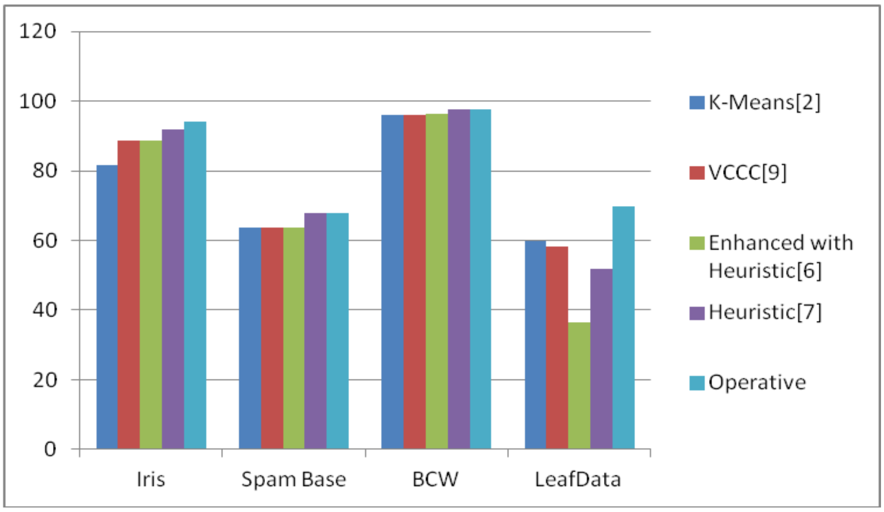

Fig. 1 Graph of Comparison

VI. CONCLUSION AND FUTURE WORK

The experimental results conclude that the proposed method produced good results. The purity of the clusters produced by the proposed method is more compared with other works tested. These algorithms eliminated the inconsistency with initial centroid selection. The results revealed that the mode of selection of initial centroids determines the cluster purity. There is still scope for research. Researches in the field of improving the performance of *K*-Means could not develop a widely accepted version. Researches can address the following issues to improve the performance.

- K - the number of clusters should be fixed beforehand.
- The computational complexity is very high.

REFERENCES

[1] Anil K. Jain. Data Clustering: 50 years beyond K-means. Pattern Recognition Letters Elsevier, 31(8): 651–666, 2010.

[2] Margaret H. Dunham, Data Mining- Introductory and Advanced Concepts, Pearson Education, 2012

[3] Fahim A.M, Salem A.M, Torkey F.A and Ramadan M.A. An efficient enhanced k-means clustering algorithm. Journal of Zhejiang University SCIENCE ISSN 1009-3095 (Print); ISSN 1862-1775 (Online), www.springerlink.com, 7(10), 2006

[4] K.A.A. Nazeer and M.P. Sebastian. Clustering Biological Data Using enhanced k-Means Algorithm. Springer Netherlands, First edition, 2010.

[5] M.P.Sebastian and K.A. Abdul Nazeer. Improving the accuracy and efficiency of the k-means clustering algorithm. In Proceedings of the World Congress on Engineering 2009, Vol I July 2009.

[6] M.P.Sebastian, K.A. Abdul Nazeer and S.D.Madhu Kumar. Enhancing the k-means clustering algorithm by using a O(n logn) heuristic method for finding better initial centroids In Second International Conference on Emerging Applications of Information Technology IEEE , Feb 2011 pp. 261 –264.

[7] M.P.Sebastian, K.A. Abdul Nazeer and S.D.Madhu Kumar. A Heuristic k-Means Algorithm with Better Accuracy and Efficiency for Clustering Health Informatics Data American Scientific Publishers Journal of Medical Imaging and Health Informatics , 1:66–71, 2011.

[8] R.Sumathi and E.Kirubakaran. Enhanced Weighted K-Means Clustering Based Risk Level Prediction for Coronary Heart Disease European Journal of Scientific Research ,ISSN 1450-216X 71(4):490–500 2012.

[9] Murat Erisoglu, Nazif Calis and Sadullah Sakalli- ogl. A new algorithm for initial cluster centers in k-means algorithm. Pattern Recognition Letters Elsevier, 32(14):1701–1705, October 2011.

[10] Damodar Reddya and Prasanta K. Janaa. Initialization for K-means clustering using Voronoi diagram. Proscenia Technology Elsevier, 4:395–4

[11] T. Hitendra Sarma, P. Viswanath and B. Eswara Reddy A hybrid approach to speed-up the k-means clustering method Springer-Verlag 2012.

[12] JASILA, E K; NAZEER, K A Abdul. Microarray Gene Expression Data Clustering Using Red Black Tree Based K-Means Algorithm. International Journal of Management & Information Technology, [S.l.], v. 1, n. 3, p. 54-58, sep. 2012. ISSN 22785612. Available at: <http://cirworld.ijssronline.com/index.php/ijmit/article/view/13-MIT-216>

[13] O. L. Mangasarian and W. H. Wolberg: "Cancer diagnosis via linear programming", SIAM News, Volume 23, Number 5, September 1990, pp 1 & 18.

[14] The UCI Repository website. [Online]. Available: http://archive.ics.uci.edu/

[15] Introductory Statistics Lectures Measures of Variation Descriptive Statistics III: Anthony Tanbakuchi Department of Mathematics Pima Community College.

[16] Christopher D. Manning, Prabhakar Raghavan & Hinrich Schütze; : Introduction to Information Retrieval ; Website:http: //informationretrieval.org/ Cambridge University Press © 2008 Cambridge University Press.

[17] Bryan Bergeron, Bioinformatics Computing “in PHI.