I.J. Intelligent Systems and Applications, **2013, 02, 41-51**
Published Online January 2013 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2013.02.05

DIMK-means "Distance-based Initialization Method for K-means Clustering Algorithm"

Raed T. Aldahdooh
Computer Engineering Dept., Islamic University of Gaza (IUG), Gaza, Palestine
Raed.Ald@gmail.com

Wesam Ashour
Computer Engineering Dept., Islamic University of Gaza (IUG), Gaza, Palestine
Washour@iugaza.edu.ps

Abstract— Partition-based clustering technique is one of several clustering techniques that attempt to directly decompose the dataset into a set of disjoint clusters. K-means algorithm dependence on partition-based clustering technique is popular and widely used and applied to a variety of domains. K-means clustering results are extremely sensitive to the initial centroid; this is one of the major drawbacks of k-means algorithm. Due to such sensitivity; several different initialization approaches were proposed for the K-means algorithm in the last decades. This paper proposes a selection method for initial cluster centroid in K-means clustering instead of the random selection method. Research provides a detailed performance assessment of the proposed initialization method over many datasets with different dimensions, numbers of observations, groups and clustering complexities. Ability to identify the true clusters is the performance evaluation standard in this research. The experimental results show that the proposed initialization method is more effective and converges to more accurate clustering results than those of the random initialization method.

Index Terms— K-means Algorithm, Clustering Algorithm, Cluster Centroid Initialization, Initializing K-means, K-means Seeding Technique

I. Introduction

In the mid-18th century, in London during cholera outbreak, John Snow had plotted the diseased reported cases using a special map. A key observation, after the creation of the map, was the close association between the density of disease cases and a single well located at a central street. Without the map; it was very difficult to identify the association between the diseased and their locations. This was the first known application of clustering analysis for many researchers [1]. Clustering is an important unsupervised learning technique where a set of patterns, usually vectors in a multidimensional space, are used for identifying group of similar characteristics. Each group, called cluster, consists of vectors that are similar between themselves and dissimilar to vectors of other groups "clusters" [2] [3] [4]. However, cluster analysis is considered to be the most popular tool in statistical data analysis which is widely applied in a variety of scientific areas such as data mining, pattern recognition, geographic information systems, information retrieval, microbiology analysis and so forth [5] [6] [7]. The most challenging task in clustering is lack of prior knowledge. Literature review reveals researchers' interest in the development of efficient clustering algorithms and their application in a variety of real-life situations. We can divide clustering algorithms into three main categories [8]: clustering with overlapping (non-exclusive), partitional, and hierarchical. The last two types have a relationship with each other in that a hierarchical clustering is a nested sequence of hard partitional clustering's, each of which represents a group of the dataset into a different number of mutually disjoint groups. A hard partition of a dataset X = {x1, x2, ..., xN}, composed of n-dimensional attribute vectors xj, is a collection G = {G1, G2, ..., Gk} of k non-overlapping data groups Gi(clusters) such that $G1 \cup G2 \cup ... \cup Gk = X$, & $Gi \neq \emptyset$ and $Gi \cap Gl = \emptyset$ for $i \neq l$. Overlapping methods search for soft or fuzzy partitions by somehow relaxing the mutual disjointness constraints $Gi \cap Gl = \emptyset$. As conclusion; partition-based clustering attempts to analyze data and assembles it in a set of groups separate from each other. The objective function of partition-based clustering algorithm tries to minimize may emphasize the local structure of the data, as by assigning groups to reach its peak in the probability density function, or global structure. Typically the global criteria involve maximizing some measure of dissimilarity between each cluster, while minimizing dissimilarity in the samples within each cluster. Cluster similarity is measured in regard to the mean value of the objects in a cluster, center of gravity, (K-Means [9]) or each cluster is represented by one of the cluster objects located near its center (K-Medoid [10]).

K-means is one of the most famous partition clustering algorithms because of: (i) K-means has been recently elected and listed among the top ten most

influential data mining algorithms; (ii) it is at the same time very simple and quite scalable, as it has linear asymptotic running time with respect to any variable of the problem. K-means clustering is a method of cluster analysis which aims to partition n observations (x1, x2, …, xn), where each observation is a d-dimensional real vector into k clusters in which each observation belongs to the cluster with the nearest mean. In general K-means is one of the most important and best performances of the clustering algorithms. However, there are some drawbacks for K-means algorithm like sensitivity to the initial cluster centroids which is addressed in this paper [11] [12]. Moreover, when the number of data points is large, it takes enormous time to find the global optimal solution [13].

K-means has several limitations which are listed below:

- Scalability: It scales poorly computationally.
- Initial means: The clustering result is extremely sensitive to the initial means.
- Noise: Noise or outliers deteriorates the quality of the clustering result.
- Number of clusters: The number of clusters must be determined before the means clustering begins.
- Local minima: It always converges to local minima.
- Inability to cluster non-linearly separable dataset: It fails to split non-linearly separable datasets in the input space.

In this paper, we suggest techniques to overcome the initial centroid sensitivity drawback in order to get better clustering results. Section 2 of this paper reviews briefly some necessary background on research, which proposed techniques to overcome sensitivity of initialization process in K-means algorithms. Section 3 reviews the basic concepts and steps of K-means algorithm. Section 4 describes the proposed enhanced technique. Section 5 describes datasets and experiments.

II. Related Work

Clustering is one of the fundamental problems in machine learning. K-means clustering algorithm has a very rich history because of its observed speed and simplicity, in this work the focus is on improving its accuracy. The initial location of the cluster centroid has major impact on the performance of K-means algorithm. This effect will be discussed in the next section. The following are some methods proposed by different researchers which decrease the sensitivity and increase accuracy of k-means, through selection of the best centroid locations within the existing dataset.

Method 1: [14] In 2007; David Arthur and Sergei Vassilvitskii published research titled "k-means++: The Advantages of Careful Seeding" where they proposed a specific way of choosing initial centroids. In their research; initial centroids are chosen consecutively with probability proportional to the distance to the nearest centroid as follows:

1. Choose an initial centroid c1 =x randomly from X.
2. Set D(x) as the shortest Euclidean distance from a data point x to the closest centroid.
3. Choose the next centroid ci, selecting ci = $x' \in X$ with probability

$$\frac{D(x')^2}{\sum D(x)^2} \quad (1)$$

4. Repeat steps 2 and 3 until we have chosen a total of K centroids.
5. Proceed as with the standard K-means algorithm.

They have presented a new way to seed the K-means algorithm that is O(logk) competitive with the optimal clustering. Furthermore, there seeding technique is as fast and as simple as the K-means algorithm itself, which makes it attractive in practice. There experiments show that augmentation improves both the speed and the accuracy of k-means, often quite dramatically.

Method 2: [15] Initializing Partition-Optimization Algorithms proposes a staged approach to specifying initial values by finding a large number of local modes and then obtaining representatives from the most separated ones. The steps are outlined below:

1. Obtain the singular valued decomposition of the centered data. X=UDV′, Where D is a diagonal matrix of m positive singular values.
2. For each coordinate in the reduced space, they obtain an appropriate number of local modes. For a chosen m*< m keep the first m*dimensions of the U and continue working in the reduced space.
3. Eliminate all those candidates from the product set which are not closest to any observation in U^*, Choose kj modes where kj is chosen appropriately.
4. Obtain the k* local modes of the dataset using the K-means algorithm with the starting points provided from above. Also, classify the observations, and obtain the corresponding group means in the original domain. Eliminate all candidates that are not close to any observation leaving k*modes.
5. Use hierarchical clustering with single-linkage on the k*modes to reduce the number of modes to the desired k.

Method 3: [16] Cluster Center Initialization Method for K-means Algorithm Over Datasets with Two Clusters defines nearest neighbor pair and puts forward four assumptions about nearest neighbor pairs, based on which a centroid initialization method for K-means algorithm over datasets with two clusters is build. The steps of research are outlined below:

Supposing that X={x1,x2,...,xn} is a dataset, where xj={x1j,x2j,...,xmj}T .

1. Compute the dissimilarity between any pair of data points in X using formula:

$$d(x_j, x_k) = \sqrt{(x_j - x_k)^T (x_j - x_k)} \quad (2)$$

2. For any datum point x in X find its nearest neighbor xNN using formulae:

$$x_{NN} = \arg\min_{y \epsilon X - \{x\}} \{d(x, y)\} \quad (3)$$

and constitute a set B of nearest neighbor pairs

3. Find two most dissimilar nearest neighbor pairs, (x1, x1,NN) and (x2, x2,NN), using formulae:

$$d = \big((x, x_{NN}), (y, y_{NN})\big)$$
$$d = \min\{d(a, b) | a \in \{x, x_{NN}\}, b \in \{y, y_{NN}\}\} \quad (4)$$

$$d' = \Big((x_1, x_{1,NN}), (x_2, x_{2,NN})\Big)$$
$$d' = \max\left\{d\begin{pmatrix}(x, x_{NN}),\\ (y, y_{NN})\end{pmatrix} \middle| \begin{matrix}(x, x_{NN}) \in B,\\ (y, y_{NN}) \in B\end{matrix}\right\} \quad (5)$$

4. Find the third most dissimilar nearest neighbor pairs (x3, x3,NN).
5. Find the fourth most dissimilar nearest neighbor pairs (x4, x4,NN).
6. Find the nearest neighbor pair (x5, x5,NN) on the overlapping of two clusters..
7. Select two initial cluster centroids according to some assumptions.

Method 4: [17] Hierarchical K-means: an algorithm for centroids initialization for K-means, a new approach to optimize the initial centroids for K-means proposed. It utilizes all the clustering results of K-means in certain times, even though some of them reach the local optima. Then, transform the result by combining with Hierarchical algorithm in order to determine the initial centroids for K-means. The execution steps of the proposed Hierarchical K-means algorithm to determine initial centroids for K-means are described as follows:

1. Set X={xi | i=1,….,r} as each data of A, where A={ai |,i=1,...,n} is attribute of n-dimensional vector.
2. Set K as the predefined number of clusters.
3. Determine p as numbers of computation
4. Set i=1 as initial counter
5. Apply K-means algorithm.
6. Record the centroids of clustering results as Ci= {cij | j=1,…,K}
7. Increment i=i+1
8. Repeat from step 5 while i<p.
9. Assume C = { Ci| i=1,…,p} as new dataset, with K as predefined number of clusters
10. Apply hierarchical algorithm
11. Record the centroids of clustering result as D = {di | i=1,…,K}

Then, apply D = {di | i=1… K} as initial cluster centroids for K-means clustering. The experiment results reflect the accuracy of the method.

Method 5: [18] Efficiency issues of evolutionary K-means method suggests that evolutionary techniques conceived to guide the application of K-means can be more computationally efficient than systematic (i.e., repetitive) approaches that try to get around the K-means drawbacks by repeatedly running the algorithm from different configurations for the number of clusters and initial positions of prototypes. To do so, a modified version of a (K-means-based) fast evolutionary algorithm for clustering is employed.

Method 6: [19] A Deterministic Method for Initializing K-means Clustering by Ting Su and Jennifer Dy motivate theoretically and experimentally the use of a deterministic divisive hierarchical method, which they refer to as PCA-Part (Principal Components Analysis Partitioning) for initialization. The researchers proposed sorting data instances on a single variable then performed the initial partition. These partitions are used only in one dimension. An alternative method is to partition the sample space hierarchically. Starting with one cluster, then cut it into two. Pick the next cluster to partition, and so on. PCA-Part uses the latter approach.

III. Basic Concepts

In this subsection, we review some necessary background and basic concepts.

3.1 K-means Algorithm

A partition clustering algorithm splits the data points into k partitions, where each partition represents a cluster. The partitioning is done based on certain objective function. One of the criterion functions is minimizing square error criterion which is computed as shown by formula:

$$E = \sum \sum ||p - \mu_i||^2 \quad (6)$$

Where p is the point in a cluster and μi is the centroid of the cluster. Each cluster must have at least one point and each point must be in one and only one cluster.

K-means is one of the most widely used partition-based clustering algorithms in practice. It is simple, easy, understandable, scalable, and can be adapted to deal with streaming data and very large datasets [20]. K-means algorithm divides a dataset X into k disjoint clusters based on the dissimilarities between data objects and cluster centroids. Let $\bar{\mu}_i$ be the centroid of cluster Ci and the distances between Xj that belong to Ci and $\bar{\mu}_i$ is equal to d(Xj, $\bar{\mu}_i$). Then, the objective function minimized by K-means is given by:

$$\min_{\bar{\mu}_1, .., \bar{\mu}_k} E = \sum_{i=1}^{k} \sum_{x_j \in C_i} d(x_j, \bar{\mu}_i) \quad (7)$$

Where'd' is one of distance function.

Typically d is chosen as the Euclidean or Manhattan distance.

The Euclidean distance between points X and Y is the length of the line segment connecting them ($\overline{XY}$). If X and Y are n-dimensional vectors where X= (x1, x2,..., xn) and Y = (y1, y2,..., yn), then the Euclidean distance from X to Y, or from Y to X is given by:

$$\begin{Bmatrix} d(X,Y) \\ d(Y,X) \end{Bmatrix} = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \qquad (8)$$

The Manhattan distance between two points measured along axes at right angles where distance that would be traveled to get from one data point to the other if a grid-like path is followed. In a plane with X at (x1, x2) and Y at (y2, y2), it is |x1 - y1| + | x2 – y2|. The Manhattan distance between two n-dimensional vectors is the sum of the differences of their corresponding components.

$$d(X,Y) = \sum_{i=1}^{n} |x_i - y_i| \qquad (9)$$

Where n is the number of variables, and Xi and Yi are the values of the ith variable, at points X and Y respectively.

Usually choose the method of calculating the distance between points in the K-means algorithm based on the nature of the data.

K-means algorithm working process can be summarized as follows:

- **Step1** Determine the number of clusters (k parameters in k-means).
- **Step2** K-means selects randomly k cluster centroids.
- **Step3** Assign objects to clusters based on distance function.
- **Step4** When all objects have been assigned, Re-compute new cluster centroids by averaging the observations assigned to a cluster.
- **Step5** Repeat (3-4) until convergence criterion is satisfied.

Pseudo code for K-means algorithm:

Require: $k \geq 2$ *and* $t \geq 1$ $\begin{cases} k: number\ of\ cluster, \\ t: max\ number\ of\ iteration. \end{cases}$

Select initial cluster centroids $\overline{\mu_1}, \overline{\mu_2}, .., \overline{\mu_k}$.
Repeat
For each point xj in a dataset do
For all $\overline{\mu_i}$ *do*
Compute the dissimilarity $d(x_j, \overline{\mu_i})$;
End for.
assign point xj to closest cluster Ci;
End for.
For all $\overline{\mu_i}$ *do*
Update $\overline{\mu_i}$ *as the centroid of cluster Ci;*
End for.
Until convergence criterion is satisfied or the number of iterations exceeds a given limit t.

The number of clusters found is equal to the number of the initial starting points which are specified as input parameters to the clustering algorithm.

3.2 The Effect of Random Selection of the Initial Cluster Centroids

Starting points in K-means algorithm has significant impact on the results. In this sub section we will illustrate by examples that choosing different starting point values lead to different clusters with different error values "effect of K-means initialization process".

Figure 1 shows the results of running the K-means clustering algorithm on dataset with input parameter (k=2). This simple example shows that the position of starting point "initial cluster centroids" is important when trying to determine the best representation of clusters. when comparing figures 1.1 and 1.2 we can determine which of the two clustering is "better", clusters in second case "figure 1.2" has a better result because of lower values of objective function E than the first clustering result in "figure1.1". By this example, we illustrate the importance of the initialization process on K-means algorithm results.

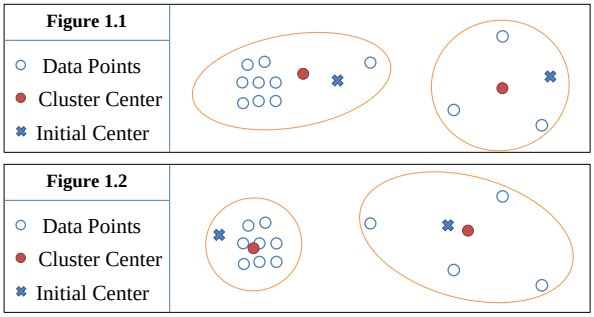

Fig. 1: Initial centroid effects on K-means result.

IV. Proposed method

Selection process of the first centroids when they are far apart and each centroid follows to different cluster has several benefits: [i] decrease amount of computation, [ii] Optimize algorithm performance by minimizing the objective function of K-means algorithm, which leads to better results.

This research proposes a new method for the initialization process in k-means; this method starts by choosing random initial centroids. After the process of selecting the start point randomly; some calculations are performed to guess whether the point is suitable to be considered as a first initial centroid or not. Such decision is based on the process of computing distances between the selected centroid and other points within the dataset. This paper uses two types of measuring distances between points "Euclidian or Manhattan"

Because of the different nature of the data. Figure 2 shows the proposed method to calculate the initial centroids for K-means algorithm.

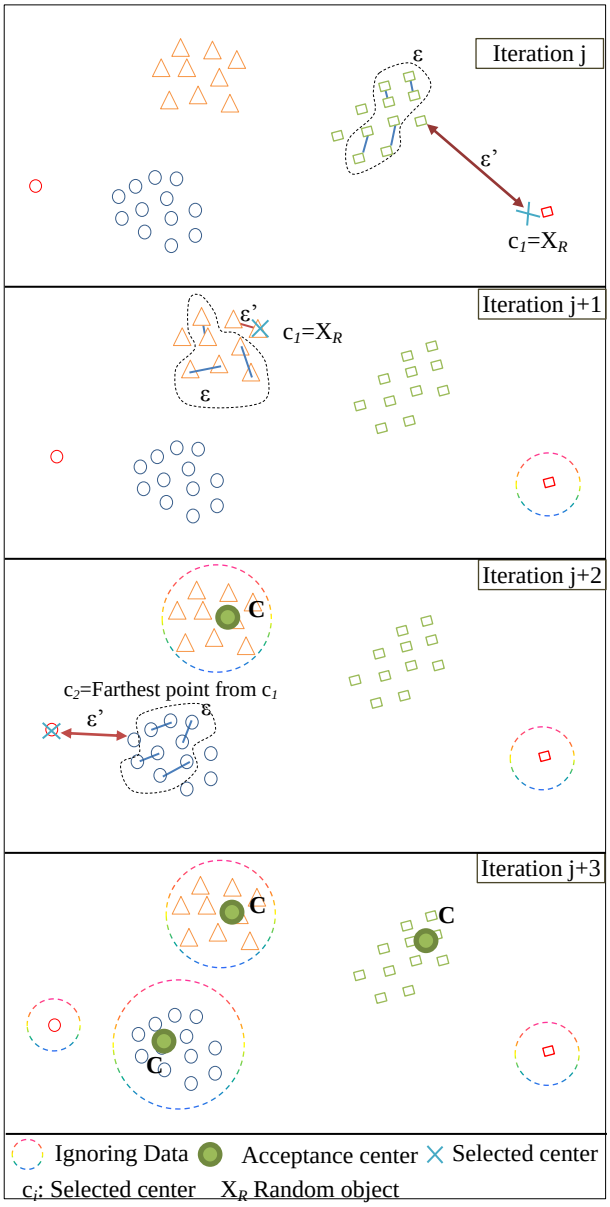

Fig. 2: show the operation of selection candidates of the initial cluster center with proposed method.

Assumption I: The number of objects in a cluster is close or equal to the number of objects in other clusters.

This assumption is based on the fact that K-means algorithm always get better results with datasets which are similar in density and close in the objects number in each cluster. So, this assumption is valid for a large number of datasets.

Point selected to be centroid, which in turn should be tested if it is noise or not. Then, the mean value of N number of the closest point to the current centroid is saved as the first accepted centroid which is ignored in the following computations. This method is repeated until the required number of centroids is identified.

The process of computing the distances between the selected point and the remaining points is the backbone of this method, because the distance values between the selected centroid and its nearest point is used to calculate the value of ε' and is compared with the value of ε which is equal to the mean value of the distances between each pair of N points.

Determine the number of closest points to the selected centroid depending on the Assumption I, where the number is equal to 80% to 90% of the number computed from dividing the total number of dataset objects on number of clusters given by the user.

If the first selected point was noise; i.e. ε'> ε; this point is ignored and another point should be selected randomly as initial centroid until the first centroid is found. Then; the next centroid should be selected as the farthest points from the first centroid. If the second selected point was noise; it is ignored and its closest

4.1 The proposed algorithm is described as follows:

Assume X={x1,x2,..,xn} which is a dataset with n number of objects, and k is an input parameter equal to number of clusters.

1. Choose an initial centroid ci = xr, where 0< i ≤ k and xr random from X.
2. Compute the distance between selected centroid ci and each point in X, and then sort the data points based on the resulted distances.

$$D = d(c_i, p_j) \tag{10}$$

Where D: typically is chosen as the Euclidean distance, 0 < j ≤ n.

3. Get a subset of the sorted data with a number of points equal to N

$$N = \text{CeilEven}\left(\frac{n/k}{\sigma}\right) \tag{11}$$

Where N: number of data most close to the selected centroid ci, σ is a double number 1< σ ≤2, and CeilEven is a function that rounds a double number up to the nearest even integer.

4. Compute the average distance between each pair of N points

$$\varepsilon = \left(\sum_{\substack{m=0,\\ m=m+2}}^{N} d(p_{m+1}, p_{m+2})\right) \Big/ \frac{N}{2} \tag{12}$$

$$\varepsilon' = d(c_i, p_1) \tag{13}$$

Where p represent the closest data points to ci, while m is incremented by 2, and p_1 is the closest point to ci.

5. If { $\varepsilon' > \varepsilon$ and i=1}; ignore ci and go to step 1 to select a new ci
6. If { $\varepsilon' > \varepsilon$ and i > 1}; ignore ci, select a new ci with value equal to the closest point to the previous ci; and go to step 2.
7. Choose the next centroid ci+1 to be the farthest point from ci.
8. The mean value of N points closest to ci is identified as the centroid and is saved as "acceptance centroid Ci ".

$$C_i = \left(\sum_{m=0}^{N} p_m\right) \Big/ N \qquad (14)$$

Where Ci: represent the mean value of the closest points to ci.

9. Ignore N points which are the closest to ci.
10. Go to step 2 with value of ci = ci+1.
11. Repeat steps until a total of K centroids are chosen.
12. End

V. Performance Evaluation

To test the performance of the new algorithm models "DIMK-means", firstly, datasets using to test the new model introduced; then, experiment results of the new algorithm compared with standard k-mean algorithm.

5.1 Datasets Selection

The performance evaluation of the proposed initialization method is applied on five different artificial and real-world datasets. Furthermore, the performance of K-means algorithm with the proposed initialization method is evaluated using popular evaluation methods such as Sum of Square Errors (SSE), Akaike Information Content (AIC) and The Bayesian Information Criterion (BIC). The results of such evaluation are compared with K-means algorithm with random initialization method in order to identify the differences.

5.1.1 Artificial datasets

The Artificial datasets used in the experiment are:

1. **The Ruspini dataset: Ruspini dataset** [21], is a collection of 75 points, arranged in 4 groups, in the Euclidean plane. It is widely used to illustrate the effectiveness of clustering methods. The following section shows the results of applying the proposed initialization method to this dataset.
2. **The Rfivec dataset**: Artificial datasets generated by the researcher with two dimensions of feature, this dataset was designed in a way that is sensitive to centroid initialization. This dataset contains 5 clusters as follows: cluster 0 from 1 to 21, cluster 1 from 22 to 52, cluster 2 from 53 to 78, cluster 3 from 79 to 87 and cluster 4 from 88 to 135. Values of the generated artificial dataset are used to assess the level of the algorithm accuracy and ability to identify true clusters.

5.1.2 Real datasets

The real datasets used in the experiment are:

1. **IRIS Dataset**: This is perhaps the best known database to be found in the pattern recognition and clustering literature. The dataset contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are not linearly separable from each other.
2. **Wine recognition Dataset**: These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines.
3. **Libras Movement Dataset**: The dataset contains 15 classes of 24 instances each, where each class references to a hand movement type in LIBRAS. In the video pre-processing, time normalization is carried out selecting 45 frames from each video, in according to a uniform distribution. In each frame, the centroid pixels of the segmented objects (the hand) are found, which compose the discrete version of the curve F with 45 points. All curves are normalized in the unitary space. In order to prepare these movements to be analyzed by algorithms, a mapping operation is carried out, that is, each curve F is mapped in a representation with 90 features, with representing the coordinates of movement. Some sub-datasets are offered in order to support comparisons of results.

These datasets and more can be found in UCI Machine Learning Repository [22] which is a collection of databases, domain theories, and data generators used by the machine learning community for the empirical analysis of machine learning algorithms.

5.2 Experiment Results

The clustering results achieved by K-means algorithm which uses random initial centroids and K-means with initial centroids derived by the proposed algorithm were compared. The clustering results of K-means using random initial centroids are the mean results of over 30 runs since each run gives different results. The experiment implementation is summarized in coding K-means and enhancing the process of selecting initial centroids of clusters using Java

programming language and thus, the algorithm runs on any platform. Moreover, the algorithm allows the user to load many datasets file formats such as csv, data or txt files.

The effectiveness of a clustering algorithm can be evaluated by different measurement algorithms such as:

Sum of Square Errors (SSE): [23] SSE is the simplest and most widely used criterion measure for clustering. For a given cluster; SSE is computed as follows: for each instance in the cluster; summing the square differences between each attribute value and the corresponding one in the cluster centroid. These are summed up for each instance in the cluster and for all clusters. The formula for SSE for one cluster is:

$$\mathrm{SSE} = \sum_{i=1}^{n} (x_i - x_c) \qquad (15)$$

Where n is the number of observations xi is the value of the ith observation and xc is the mean of all the observations.

Akaike Information Content (AIC) score: [24] [25] AIC measures the log-likelihood of the model penalized by the number of parameters in the model. A clustering result with small k and small variance of each cluster will have a relatively low AIC score, which means the clustering result is good. In the general case, the formula for AIC is:

$$\mathrm{AIC} = 2k - 2\ln(L) \qquad (16)$$

Where k is the number of parameters in the statistical model, and L is the maximized value of the likelihood function for the estimated model.

The Bayesian Information Criterion (BIC): [26] BIC proposed by Schwarz (1978) is a popular method for model selection. BIC evaluates candidate models with different number of basic functions, and the optimal number is chosen from the best model in terms of BIC score. The formula for the BIC is:

$$\mathrm{BIC} = 2 * \ln(k) + k * \ln(n) \qquad (17)$$

Where n is equal to sample size, k is the number of parameters in the statistical model, and L is the maximized value of the likelihood function for the estimated model.

Table 1 shows the comparison of K-means performance results using the random and the proposed method for initial cluster centroids which was applied on the artificial datasets described in 5.1.1 subsection.

Table 1: The K-Means Clustering Mean Results of Artificial Datasets over 30 Runs
(K *is an input parameter obtained from user, which represent the number of clusters*).

Dataset	Method	K	SSE	AIC	BIC
Ruspini	Random	4	69878.82	2989.233	2989.108
	Proposed method		25712.90	2948.02	2947.895
Rfivec	Random	5	630730.3	6583.498	6583.628
	Proposed method		473906.4	6567.404	6567.535

Table 2: The K-Means Clustering Mean Results of Real Datasets over 30 Runs
(K *is an input parameter obtained from user, which represent the number of clusters*)

Dataset	Method	K	SSE	AIC	BIC
IRIS	Random	3	223.3735747	9926.572673	9926.748764
	Proposed method		171.1536034	9903.996737	9904.172828
Wine recognition	Random	3	4953073.037	398451.3311	398451.5791
	Proposed method		4846459.155	398308.174	398308.422
Libras Movement	Random	15	678.0630313	2109578.518	2109577.961
	Proposed method		649.1821375	2087765.609	2087765.052

Table 2 also shows the comparison of initial cluster centroids computed using the proposed algorithm and the random initialization method, which were applied on the real datasets described before.

It is observed that the proposed method initialization centroids selection scored smaller values for each type of performance measurement algorithms (SSE, AIC, or BIC) than the results of the random initialization method. In addition, usually the proposed method leads to SSE values close to the minimum SSE values obtained from the random initialization method. Moreover, the difference between the values of the worst and the best case reached by the K-means algorithm when initialized with the random method is very high, confirming the need for a stable initialization method, while in the proposed method, the gap between the worst and best case is kept to minimum. This proves

that the proposed method is more stable than the random method and has better results.

To prove the efficiency of the proposed method, the graph of datasets was implementing to make a comparison between the results of the random initialization and the proposed methods.

Figures 3 and 4 show the results of running the K-means algorithm with Ruspini dataset which consists of 4 clusters.

- Figure 3: results of K-means algorithm using random initialization.
- Figure 4: results of K-means algorithm using the proposed initialization method.

Each identified cluster identified by a different plotting character and color. Note the widely divergent results.

It is observed that the random method for initialization get inefficient results as it merges 2 clusters together (which are plotted as red square in figure 3) and split one of the true clusters to two different clusters (plotted with blue circles and green dots in figure 3). While the proposed method for initialization is more efficient and accurate in identifying each cluster very close to true ones.

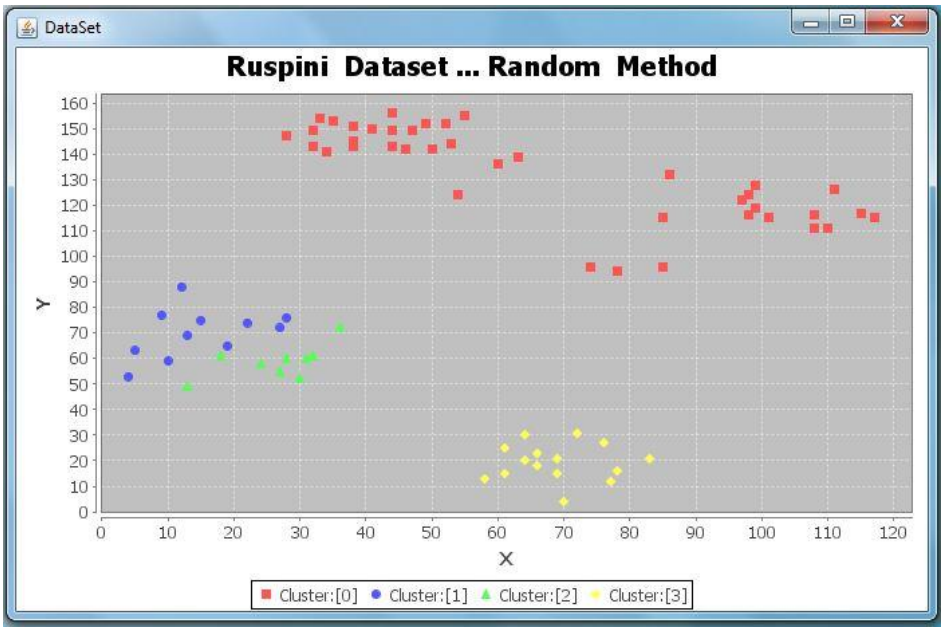

Fig. 3: Results of running the K-means with k=4 and using 4 different starting points, each randomly chosen from the dataset

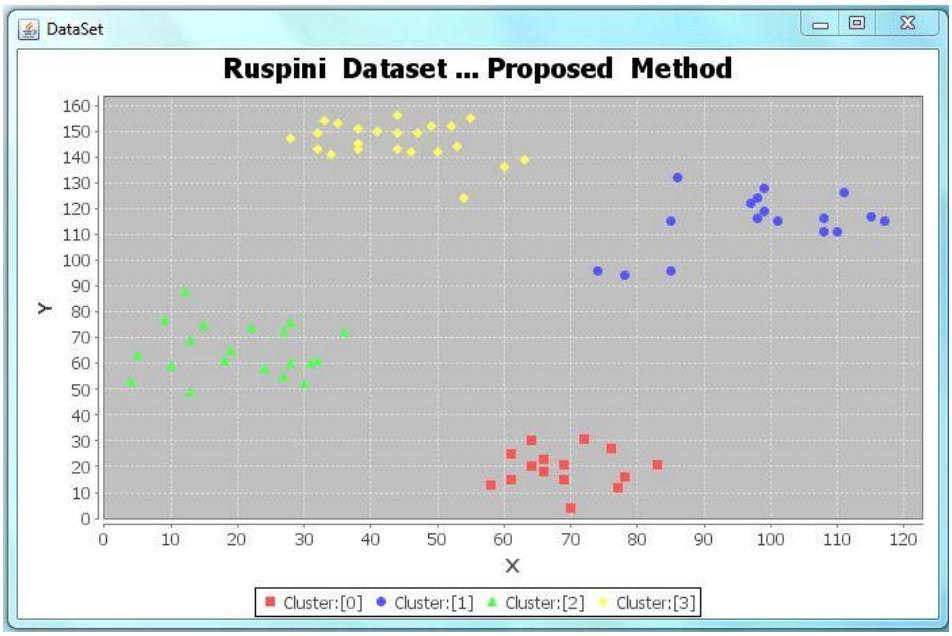

Fig. 4: Results of running the K-means with k=4 and using 4 different starting points, each chosen with the proposed method

Figures 5 and 6 show the results of running the K-means algorithm with Rfivec dataset which consists of 5 clusters.

- Figure 5: results of K-means algorithm using random initialization.
- Figure 6: results of K-means algorithm using the proposed initialization method.

like the previous figures, we observed that the random method for initialization get inefficient results as it merges 2 clusters together (which are plotted as blue circle in figure 5), split one of the true clusters to tow different clusters (plotted with red square and pink oblong in figure 5) and merge subset of cluster with another one (plotted with yellow cube in figure 5). While the proposed method for initialization is more efficient and accurate in identifying each cluster very close to true ones.

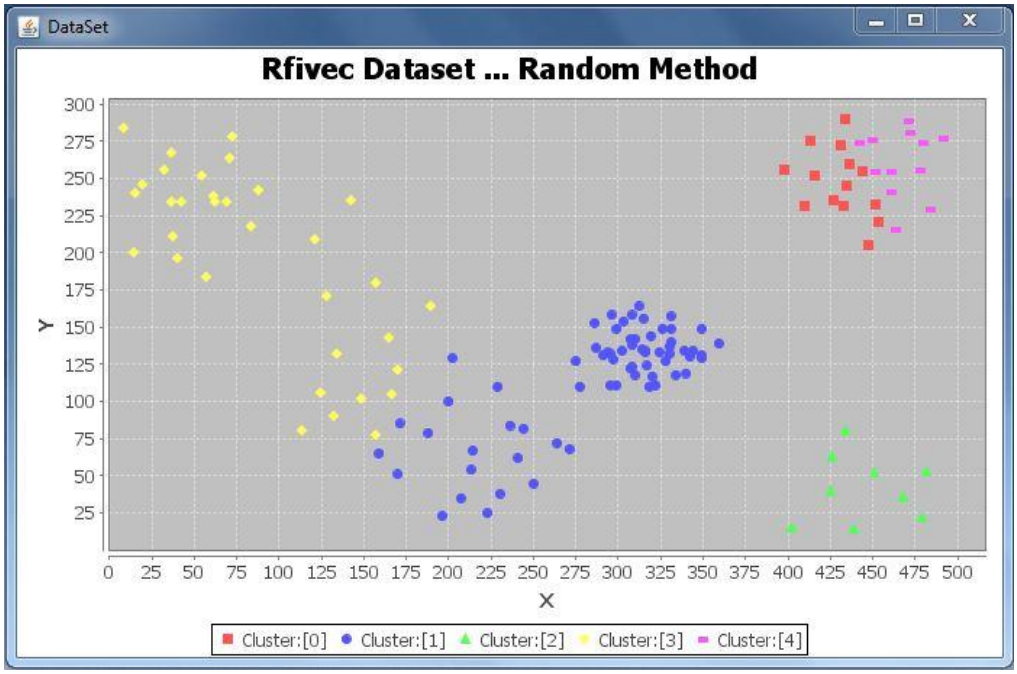

Fig. 5: Results of running the K-means with k=5 and using 5 different starting points, each randomly chosen from the dataset.

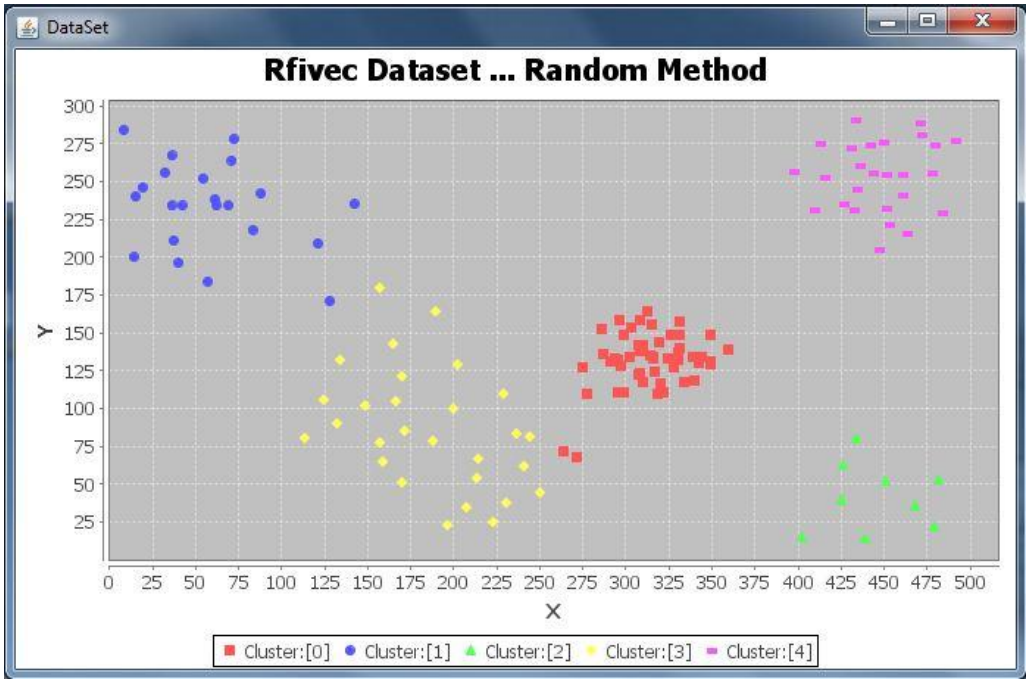

Fig. 6: Results of running the K-means with k=5 and using 5 different starting points, each chosen with the proposed method.

VI. Conclusion

Clustering is used in many fields such as data mining, knowledge discovery, statistics and machine learning. A good clustering algorithm produces high quality clusters to yield low inter cluster similarity and high intra cluster similarity. This paper presents a new way to select initial centroids in K-means algorithm. This initialization method is as fast and as simple as the K-means algorithm itself, which makes it attractive in practice. The main reason of this enhancement is to make K-means less sensitive to the initialization process and to get consistent results every time algorithm runs. Experimental results demonstrate that the modification appears to give efficient performance when dealing with several virtual and real-world datasets, and it is observed that the proposed method has substantially outperformed the standard K-means in terms of both speed and accuracy.

Acknowledgments

The authors would like to thank the anonymous reviewers for their careful reading of this paper and for their helpful comments. An extra special thanks goes to Walid Alnabahin who did not spare any effort to review and audit this paper linguistically.

References

[1] wikipedia. (2012, April) wikipedia. [Online]. http://en.wikipedia.org/wiki/1854_Broad_Street_cholera_outbreak

[2] T. Abraham and J. F. Roddick, "Survey of Spatio-Temporal Databases," *GeoInformatica*, vol. 3, March 1999.

[3] D. Birant and A. Kut, "ST-DBSCAN: an algorithm for clustering spatial–temporal data," *Data & Knowledge Engineering*, vol. 60, pp. 208-221, 2007.

[4] J. Han and M. Kamber, "Data Mining Concepts and Techniques, Morgan Kaufmann Publishers," *San Francisco, CA*, pp. 335–391, 2001.

[5] J. Han and M. Kamber, "Data Mining Concepts and Techniques," *Morgan Kaufmann Publishers, San Francisco, CA*, pp. 335–391, 2001.

[6] J. Han, M. Kamber, and A.K.H. Tung, "Spatial clustering methods in data mining," *Geographic Data Mining and Knowledge Discovery, Taylor & Francis, London*, 2001.

[7] M. Sadaak, Y. Endo, S. Hayakawa, and E. Kataoka, "Classification and clustering of information objects based on fuzzy neighborhood system," *IEEE Internat.i, Conf. on Systems, Man and Cybernetics, Hawai*, pp. 3210-3215, 2005.

[8] A. Jain and R. Dubes, "Algorithms for Clustering Data," *Prentice Hall*, 1988.

[9] J. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297, 1967.

[10] H. Vinod, "Integer programming and the theory of grouping," *Journal of the American Statistical Association*, vol. 64, pp. 506–519, 1969.

[11] J.T. Tou and R.C. Gonzalez, "Pattern Recognition Principles," *Addison-Wesley, Reading, MA*, 1974.

[12] S.Z. Selim and M.A. Ismail, "K-means type a lgorithms: a generalized convergence theorem and characterization of local optimality," *IEEE Trans. Pattern Anal*, vol. 6, pp. 81–87, Mach 1984.

[13] H. Spath, "Cluster Analysis Algorithms," *Ellis H orwood, Chichester, UK*, 1989.

[14] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," *ACM-SIAM Symposium on Discrete Algorithms (SODA 2007) Astor Crowne Plaza, New Orleans, Louisiana*, pp. 1–11, 2007.

[15] R. Maitra, "Initializing partition-optimization algorithms," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 6, pp. 144–157, 2009.

[16] Chun Sheng Li, "Cluster Center Initialization Method for K -means Algorithm Over Data Sets with Two Clusters," *2011 International Conference on Advances in Engineering*, vol. 24, pp. 324 – 328, 2011.

[17] Kohei Arai and Ali Ridho Barakbah, "Hierarchical K-means: an algorithm for centroids initialization for K-means ," *Rep. Fac. Sci. Engrg, Saga Univ.* , vol. 36, 2007.

[18] M.C. Naldi, R.J.G.B. Campello, E.R. Hruschka, and A.C.P.L.F. Carvalho, "Efficiency issues of evolutionaryk-means," *Applied Soft Computing*, vol. 11 , pp. 1938–1952, (2011).

[19] Ting Su and Jennifer Dy, "A Deterministic Method for Initializing K-means Clustering," *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference*, pp. 784 - 786 , Nov 2004.

[20] S. Kalyani and K.S. Swarup, "Particle swarm optimization based K-means clustering approach for security assessment in power systems," *Expert Systems with Applications*, vol. 30, pp. 10839–10846, 2011.

[21] E. H., "Ruspini," *Numerical methods for fuzzy clustering. InformationScience*, vol. 2, pp. 319-350, 1970.

[22] University of Massachusetts Amherst. Funding support from the National Science Foundation. UC Irvine Machine Learning Repository. [Online]. http://archive.ics.uci.edu/ml/

[23] Oded Maimon and Lior Rokach, *Data Mining And Knowledge Discovery Handbook*, 1st ed., 978-0387244358, Ed.: amazon, 2005.

[24] H. Bozdogan, "Akaike’s Information Criterion and Recent Developments in Information Complexity," *Journal of Mathematical Psychology*, vol. 44, pp. 62–91, 2000.

[25] wikipedia. [Online]. http://en.wikipedia.org/wiki/Akaike_information_criterion

[26] G. Schwarz, "Estimating the dimension of a model," *Annals of Statistics*, vol. 6(2), pp. 461-464, 1978.

Authors’ Profiles

Raed T. ALdahdooh

Raed Aldahdooh obtained B.Sc. degree in Computer System Engineering from Alazhar University Gaza-Palestine. He is currently pursuing Master degree curriculum in Computer Engineering from Islamic university of Gaza. His area of interests includes data mining, artificial intelligence, and pattern recognition.

Wesam Ashour

Wesam Ashour is an assistant professor at Islamic University of Gaza. He is an active researcher at the Applied Computational Intelligence Research Unit in the University of the West of Scotland. He got his Master and Doctorate degrees from UK. His research interests include data mining, artificial intelligence, reinforcement learning and neural networks.

