DE GRUYTER OPEN

JAISCR, 2013, Vol. 3, No. 1, pp. 51 – 71
DOI 10.2478/jaiscr-2014-0005

DSMK-MEANS "DENSITY-BASED SPLIT-AND-MERGE K-MEANS CLUSTERING ALGORITHM"

Raed T. Aldahdooh and Wesam Ashour

Computer Engineering Dept., Islamic University of Gaza (IUG), Gaza, Palestine
Raed.Ald@gmail.com, Washour@iugaza.edu.ps

Abstract

Clustering is widely used to explore and understand large collections of data. K-means clustering method is one of the most popular approaches due to its ease of use and simplicity to implement. This paper introduces Density-based Split- and -Merge K-means clustering Algorithm (DSMK-means), which is developed to address stability problems of standard K-means clustering algorithm, and to improve the performance of clustering when dealing with datasets that contain clusters with different complex shapes and noise or outliers. Based on a set of many experiments, this paper concluded that developed algorithms "DSMK-means" are more capable of finding high accuracy results compared with other algorithms especially as they can process datasets containing clusters with different shapes, densities, or those with outliers and noise.

1 Introduction

Clustering is a discipline aimed at revealing groups, or clusters, of similar entities in data. The existence of clustering activities can be traced a hundred years back, in different disciplines in different countries. In 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]. Since then, cluster analysis is considered to be the most popular tool in statistical data analysis which is widely applied in a variety of scienti?c areas such as data mining, pattern recognition, geographic information systems, information retrieval, microbiology , psychology and other social sciences , in order to identify natural groups in large amounts of data [2,3]. To satisfy the requirements of clustering; different clustering methods have been developed, each of which uses a different induction principles, and gives different grouping of a dataset. Deciding which the most suitable method depends on the type of the output desired, the known performance of a certain method with particular types of data, the hardware and software facilities available, and the size of the dataset. In general; clustering methods have different categorization, Farley and Raftery (1998) suggest dividing the clustering methods into two main groups: hierarchical and partitioning methods. Han and Kamber (2001) suggest categorizing the methods into additional three main categories: density-based clustering, model-based clustering and grid-based clustering. An alternative categorization method based on the induction principles of the various clustering methods is presented in (Estivill-Castro, 2000) [4]. Several studies examine a lot of clustering techniques, of which the researcher found most efficient categorization techniques are those organized into the following categories: partitioning, hierarchical, grid-based, density-based, model-based, methods for high-dimensional data, and constraint-based clustering techniques.

Partition-based clustering attempts to directly decompose the dataset into a set of disjoint clusters. The criterion function that the clustering algorithm tries to minimize may emphasize the local structure of the data, as by assigning clusters to peaks in the probability density function, or the global structure. Typically, the global criteria involve minimizing some measure of dissimilarity in the samples within each cluster, while maximizing the dissimilarity of different clusters. Cluster similarity is measured in regard to the mean value of the objects in a cluster, center of gravity, (K-means [5]) or each cluster is represented by one of the cluster objects located near its center (K-Medoid [6]). The most popular and the simplest partitional algorithm is K-means. Since partitional algorithms are preferred in pattern recognition due to the nature of available data, our coverage here is focused on these algorithms. K-means has a rich and diverse history as it was independently discovered in different scientific fields. Even though K-means was first proposed over 50 years ago, it is still one of the most widely used algorithms for clustering. Ease of implementation, simplicity, efficiency, and empirical success are the main reasons for its popularity [7].

The K-means algorithm is a simple and fast clustering technique that exhibits the problem of merging some clusters which are close together. In addition to that, the algorithm generally suffers from unsatisfactory accuracy when the dataset contains clusters with different complex shapes, sizes, noise and outliers. In this work, researcher addresses these problems by combining split and merge strategy and density clustering techniques. The proposed density-based split and merge K-means algorithm comprise of two parts, the first one depends on density to decide if the cluster to be split or not, and distance to decide if the clusters to be merged or not.

If the first part was not applicable, then the algorithms applies the second part which tackles noisy data and depends on density to identify noisy objects or points in a dataset. The next section explains this procedure in more details. Using density with split and merge techniques in this algorithm makes the proposed algorithm capable of detecting clusters with different complex shapes. Furthermore, density technique helps in discovering noise or outlier. This gives the proposed algorithm higher accurate results than the standard K-means algorithm when applied on datasets containing large numbers of objects, clusters with different shapes and/or clusters containing noise objects.

1.1 K-means algorithm

K-means algorithm divides a dataset X into k disjoint clusters based on the dissimilarities between data objects and cluster centroids. Let $\overline{\mu_i}$ be the centroid of cluster Ci and the distances between Xj that belong to Ci and $\overline{\mu_i}$ is equal to d(Xj, $\overline{\mu_i}$). Then, the objective function minimized by K-means is given by:

$$\min_{\overline{\mu_1},..,\overline{\mu_k}} E = \sum_{i=1}^{k} \sum_{x_j \in C_i} d(x_j, \overline{\mu_i}) \quad (1)$$

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 ($\overline{X\ Y}$). 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:

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

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.

$$d(X,Y) = \sum_{i=1}^{n} |x_i - y_i| \quad (3)$$

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 the selection process between the two methods of calculating the distance is left to the user based on the nature of the data. Figure 7 shows the difference between using Euclidean and Manhattan distance to calculating the distance between two points in two-dimensional space.

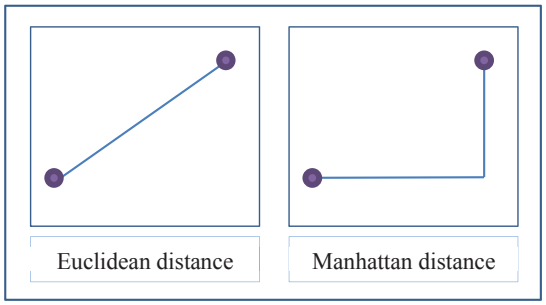

Figure 1. Euclidean and Manhattan distance between two point in tow-dimensional space.

K-means algorithm working process summarized as follows:

1. Determine the number of clusters (k parameters in k-means).
2. K-means selects randomly k cluster centroids.
3. Assign object to clusters based on distance function.
4. When all objects have been assigned, Recompute new cluster centroids by averaging the observations assigned to a cluster.
5. Repeat (3-4) until convergence criterion is satisfied.

Pseudo code for K-means algorithm:

1. Require: $k \geq 2 and t \geq 1$
   { k : number of cluster,
   t : max number of iteration.
2. Select initial cluster centroids $\overline{\mu_1}, \overline{\mu_2}, .., \overline{\mu_k}$.
3. Repeat
4. For each point xj in a dataset do
5. For all $\overline{\mu_i}$ do
6. Compute the dissimilarity $d(x_j, \overline{\mu_i})$;
7. End for.
8. assign point xj to closest cluster Ci;
9. End for.
10. For all $\overline{\mu_i}$ do
11. Update $\overline{\mu_i}$ as the centroid of cluster Ci;
12. End for.
13. 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.

2 Related works

This section is mainly concerned with presenting the algorithms that enhance and improve the performance of K-means. We will review methods that decrease the sensitivity of algorithm towards outlier or noise, and other related methods.

2.1 BNAK-Divide-and-Merge Clustering Algorithm [8]:

Divide-and-Merge is a methodology for clustering a set of objects that combines a top-down "divide" method with a bottom-up "merge" method. This algorithm proposes a normalized cut with automatically determining K clustering algorithm (BNAK-Divide-and-Merge) based on the Divide-and-Merge. Like the Divide-and-Merge, there are also two phases in this approach.

1) Divide phase:

Which is the first phase of Divide-and-Merge Algorithm, applies the spectral clustering algorithm to form a tree T whose leaves are the objects? A new threshold is proposed and called minDividedSize in Step 1 to control the number of tree nodes produced by the divide phase, which can greatly improve the efficiency of the divide phase. In Step 2, D is the diagonal matrix of the row sums of similarity matrix AAT.

Pseudo code for dividing phase:

Input: An m n× matrix A and a threshold minDividedSize

Output: A tree whose leaves are subsets of the objects

1. If the size of A is not less than minDividedSize, then go to step 2, else stop.
2. Compute the Laplacian matrix L=D-AAT.
3. Compute the two smallest eigenvectors V1 and V1 of D-1 L, let V={y1,y2,..,yn}T where V={v1,v2}
4. Partition the samples y1,y2,..,yn by K-means which k=2.
5. Let As,AT be the submatrices of A. Recurse (Step 1-4) on As and AT.

2) Merge phase:

For a large class of natural objective functions proposed by the merge phase can be executed optimally when the expected number of clusters (i.e. K) is specified at first. Alternately, they use the most obvious turning point of K-TSS curve to automatically determine the value of K. Many inner measurements of the clusters effectiveness are based on the conception of cohesion and separation. Cluster cohesion (i.e. SSE) is the sum of the weight of all links within a cluster. Cluster separation (i.e. SSB) is the sum of the weights between nodes in the cluster and nodes outside the cluster. In some cases, there is a strong connection between the cohesion and the separation. Specifically, the sum of SSE and SSB is equal to total sum of squares TSS. TSS is defined as follows: TSS=SSE+SSB. They observing that most obvious turning point of the K-TSS curve can help us determine the expected number of clusters.

This concludes that K-Divide-and-Merge clustering algorithm (BNAK-Divide-and-Merge) based on the Divide-and-Merge, improves the efficiency and performance of the clustering.

2.2 A Modified K-means Algorithm for Noise Reduction in Optical Motion Capture Data [9]:

A modification to K-means algorithm has been used for removing noise in multicolor motion capture image sequences. The proposed algorithm takes into account the nature of the motion capture images in terms of the number of data pixels normally clustered together and the acceptable degree of compactness of a data cluster. The modified K-means algorithm is used to clean up the noise embedded in the color regions in each image by creating clusters of pixels based on their relative spatial positions in the image. Following the classical K-means algorithm, the Euclidean Distance measure is used to determine which cluster a pixel belongs to. Each pixel is put into a cluster, which yields the minimum Euclidean Distance between the pixel and the respective centroid. The centroid of each cluster is changed iteratively by calculating its new coordinate as the average of the sum of the coordinates of the pixels in the cluster until it converges to a stable coordinate with a stable set of member pixels in the cluster. In each iteration, the memberships of each cluster keep changing depending on the result of the Euclidean Distance calculation of each pixel against the new centroid coordinates.

Classical K-means algorithm is modified upon the form of constraints on cluster size and cluster compactness. The value for the cluster size constraint is set just above the number of data points usually found in a noise cluster for the type of data at hand. The value for the cluster compactness constraint is set just below the minimum compactness of valid data clusters.

2.3 Automatic Cluster Number Selection using a Split and Merge K-means Approach [10]:

This research address the problem of cluster number selection by using a K-means approach that exploits local changes of internal validity indices to split or merge clusters. There split and merge K-means issues criterion functions to select clusters to be split or merged and fitness assessments on cluster structure changes.

Assume a set of data samples X ={x1,. . . ,xN} is given, C={c1,. . . ,ck} being the cluster centroid, the optimization criterion in the research is given as $L = \sum_{i=1}^{N} x_i^T c_{yi}$ where $y_i = \text{argmax}_{1 \le k \le K} x_i^T c_k$ the hard assignment of samples to cluster is denoted as set y={y1,. . . ,yN}

2.3.1 Split and Merge K-Means

- Require: X ,K, s(C), m(C), v(C)
- Ensure: C , Y

1. C = K-means (Xt , K)
2. Repeat
3. cs = s(C), $X_s = \{x_n \mid y_n = s\}$
4. $\{c_i \mid c_j\}$ = K-means (Xs, K = 2)
5. if v(C) > v(C/cs U $\{c_i \mid c_j\}$) then C=C/cs U $\{c_i \mid c_j\}$
6. until |C|is not changing
7. repeat
8. ci, cj = m(C)
9. Yj = Yi , C = C /cj.
10. if v(C) > v(C /cj) then
11. C = C /cj
12. until |C|is not changing
13. C = K-means (Xt , C)

This split and merge K-means creates an initial partitioning through a first K-means step with a predefined number of clusters. Afterwards consecutive split and merge steps are invoked where the changes on the cluster result are assessed using some internal validity measure v(C) like the Bayesian Information Criterion (BIC). Those split and merge steps are repeated until changes no longer improve the fitness. At the end of the algorithm, an optional K-means step can further refine the results of the dynamic updates. Note that the input parameter K is optional and per default two, but the algorithm allows setting a preliminary expectation on the cluster number to reduce runtime. In order to reduce the number of splits and merges, algorithm also introduces a splitting criterion s(C) and a merging criterion m(C) for selecting the cluster to split or merge in a step. In this approach, s(C) selects the cluster with the lowest average data sample similarity. Similarly, m(C) selects the two most similar clusters as merging candidates. Researcher claims that split and merge K-means reaches the goal of providing a clustering structure that dynamically selects its cluster number with an acceptable runtime and a favorable precision. In addition, this approach can be highly effective to generate an initial clustering result with an automatically detected number of clusters as well as in incremental applications where the given cluster hierarchy should be updated dynamically as new documents are added or old documents are removed. As a final remark, this split and merge approach seems to reach the goal of providing a clustering structure that dynamically selects its cluster number with an acceptable runtime and a favorable precision.

3 Performance of K-means

This section discusses a set of experiments on K-means algorithm with different datasets. These experiments illustrate the ability of K-means algorithm to find the true cluster, as highlight the strengths and weaknesses of algorithm is the principle aim of these experiments.

To establish practical applicability of K-means algorithm, its performance was tested on a number of artificial and real world datasets. Those datasets contain clusters with different complex shapes, densities, sizes, noise and outliers. The main purpose is to show how K-means work with this type of datasets. It was experimented on two different types of datasets which are: Artificial (Ground Separation, document Sim, and Rnoisy) and real datasets (Web Log, Image Extraction). These datasets are described in depth in section 5.

The next paragraphs illustrate researcher observations on the results of standard K-means algorithm on all previous datasets.

3.1 Interpreting Results of K-means with Ground_Separation dataset:

In many clustering analysis problems, one would like to extract structure from cluttered background. This is the case in the Ground_Separation dataset. In such cases, it is easy to predict that K-means will not get accurate results, due to their requirement to partitioning all the input data. To illustrate this point, consider the Ground_Separation dataset shown in Figure 2, which contains a dense central cluster of random points surrounded by evenly, distributed clutter points (the "background") and there are four extra clusters around the ring cluster. As expected, on these data, K-means failed as it splits the central group into multi pieces.

This experiment was conducted many times on Ground_Separation dataset as shown in Figure 2. The main feature of this dataset is that it contains different structurally clusters, one is compact, the other with extended structure. Here, K-means produces inaccurate results, as shown in Figure 3.(A, B, C, and D). After running the algorithm several times with these datasets, the results were inconsistent every time. Researcher noticed that some parts of the ring-shaped cluster were classified with disparities between five or six different clusters, even though all the points forming the ring belong to one cluster. These results shown in Figure 3.(A,B,C and D).The most important general observation is the fact that centroids of clusters obtained from K-means results, which plotted as x on Figure 3, is always not in a dense area.

3.2 Interpreting results of K-means with Rnoisy dataset:

K-means was applied on Rnoisy dataset and the results were off inconsistent accuracy. This convinced the researcher that K-means has unstable results when applied on datasets similar to Rnoisy dataset, which contain many noisy points. K-means algorithm gives high accurate results when applied on Rnoisy dataset are shown in Figure 16. It is clear in Figure 4 that the data contain many noisy points which K-means algorithm is very sensitive to. The researcher observed during the tests that the shape of clusters in results takes different forms in each time. Figure 5.A summarizes the results and it is easy to observe visually inaccurate the obtained results were. Curves in Figure 5.A show the inaccurate results area where red-dotted line shows that one true cluster has been split into two clusters “blue and red”, while the black-dotted line shows that two true clusters merged into one cluster “yellow”. Many other inaccurate results occur repeatedly in Figure 5(.B.C.D). Finally, researcher observed that noisy points are always not in dense areas. This observation was the basis upon which the researchers depended to develop new ways of overcoming the weakness of K-means algorithm when working with noisy datasets.

4 Proposed Method

A discussion of the previous section experiments results shows the performance of K-means algorithm with different datasets with different behavior. Now researcher reviews the proposed ideas designed to overcome and solve major limitation and weaknesses of K-means algorithm. Generally, the algorithm suffers from unsatisfactory accuracy when the dataset contains clusters with different complex shapes, sizes, noise and/or outliers.

Based on the observation from the previous experiments where K-means merged true clusters, the resulting cluster centroid was -most of the time- not located in a density unit as it is locate between multiple true clusters. This observation was a result of the fact that K-means algorithm gets low accurate results when working with datasets contains clusters with different complex shapes. So, the researcher proposes to apply Split and Merge technique to overcome such limitation.

Another observation is the low accuracy of K-means algorithm when working with noisy datasets where noise or outliers always spread between datasets objects not in density unit. A proposed solution to overcome such limitation is by temporarily ignoring noisy objects which are not located in dense units, then rerunning standard K-means which is expected to give better results without the neglected noise. After that, re-include the neglected noisy objects to the nearest clusters.

The proposed algorithm includes solutions for cluster with complex shapes and datasets with noisy objects. The solution for the first problem is split and merge while the solution for the other problem is called anti-noise. This algorithm is applied on the results of standard K-means starting with checking if all the clusters’ centroids are located in density units, anti-noise solution is applied, but if one or more centroids are located in non-density unit, then split and merge solution is applied.

The following subsections explain in details how each case solution is implemented:

4.1 Split and Merge Method:

When applying standard K-means on datasets containing clusters with different complex shapes, some of the resulting clusters are either merged into

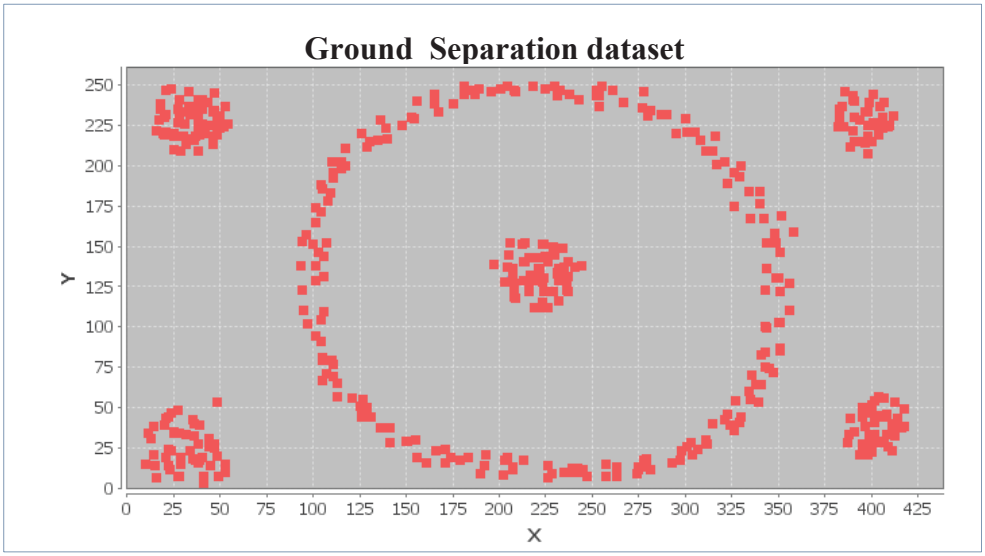

Figure 2. plot points belong to Ground_Separation dataset.

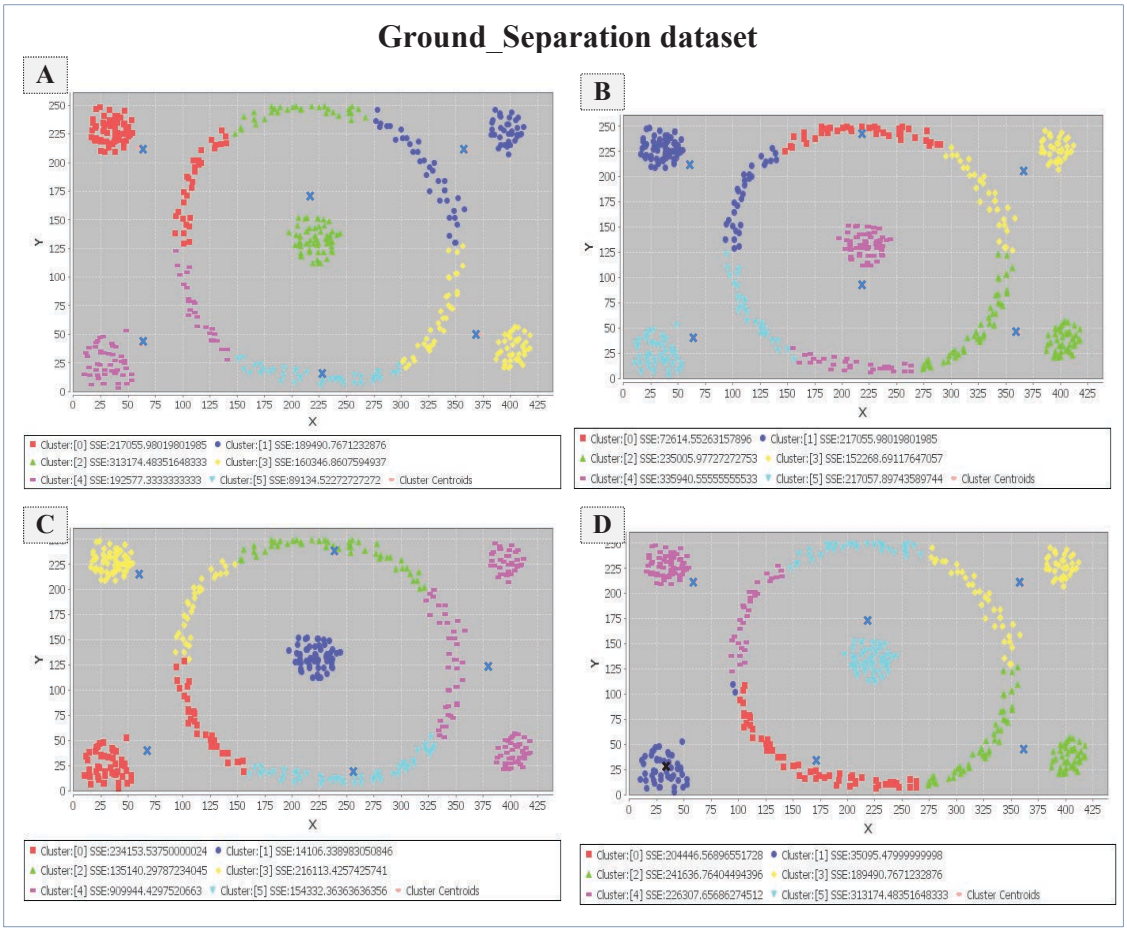

Figure 3. Low accurate results obtained with standard K-means algorithm with (Ground_Separation dataset).

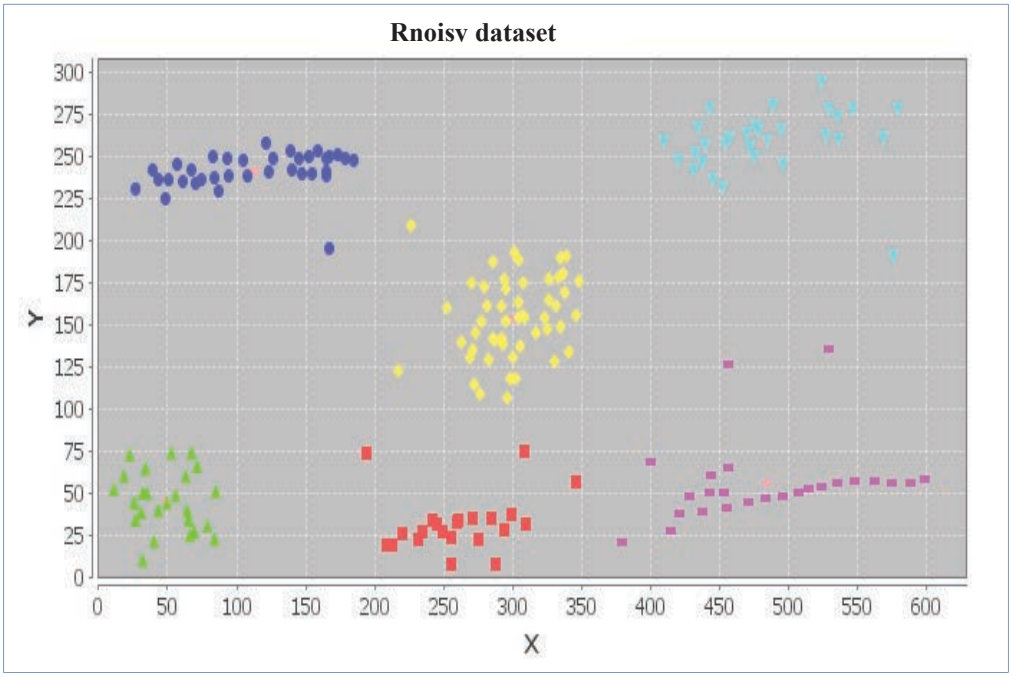

Figure 4. High accurate result obtained with standard K-means algorithm with (Rnoisy).

larger clusters or split to smaller ones. **First**, in order to determine which method to apply we need to identify if the clusters have centroids in non-density units. So, Sum Square Error "SSE" is computed for each cluster in standard K-means results where the cluster with smallest SSE value is selected, then we compute Epsilon "ε" which is the radius that delimitate the neighborhood area of a point by calculating the distance between the centroid of the selected cluster and the nearest point multiplied by 2. Then calculating "MinPts" which represent the minimum number of points that must exist in the ε. MinPts is equal to 0.75 of that number of points within ε radius (approximated to an integer number).

Based on experiments, the researcher found that multiplying the distance between the centroid and its nearest point by 2 is the most convenient and yields the best results most of the time, as well as determining MinPts by multiplying the number of points falling within ε radius by 0.75.

Second, each cluster centroids in the standard K-means results is tested to make sure it has a number of point equal to or greater than MinPts. If there is at least one centroid that has a number of neighbor points within ε" radius that is less than MinPts; then it is not in a density area, and we start the Split and Merge method, otherwise we use Anti-noise method as in case 2.

4.1.1 Density-based cluster split:

The splitting process is applied on clusters with centroids located in non-density units, each of those clusters is split into two new clusters. The resulting cluster centroids are tested to assure them all located in density units. The process of splitting is repeated until all the resulting cluster centroids are located in density units using the same ε and MinPnts calculated at the first run. This process is applied for three levels at most as any further splitting will not be useful based on the researcher experiments.

Splitting clusters into only two new sub-clusters instead of three or more is based on the fact that the possibility of having new cluster centroids in density unit in the least number of possible sub-clusters is higher than having such results in more than two sub-clusters.

A counter in increased by one each time a cluster is split, in order to keep record of how many split process were done to be used in the merge process.

4.1.2 Single linkage based cluster merge:

When the split process is finished, all clusters' centroids are in density units and the number of clusters is more that the number of clusters obtained from the standard K-means applied in the first step.

The merge process starts by creating "distance

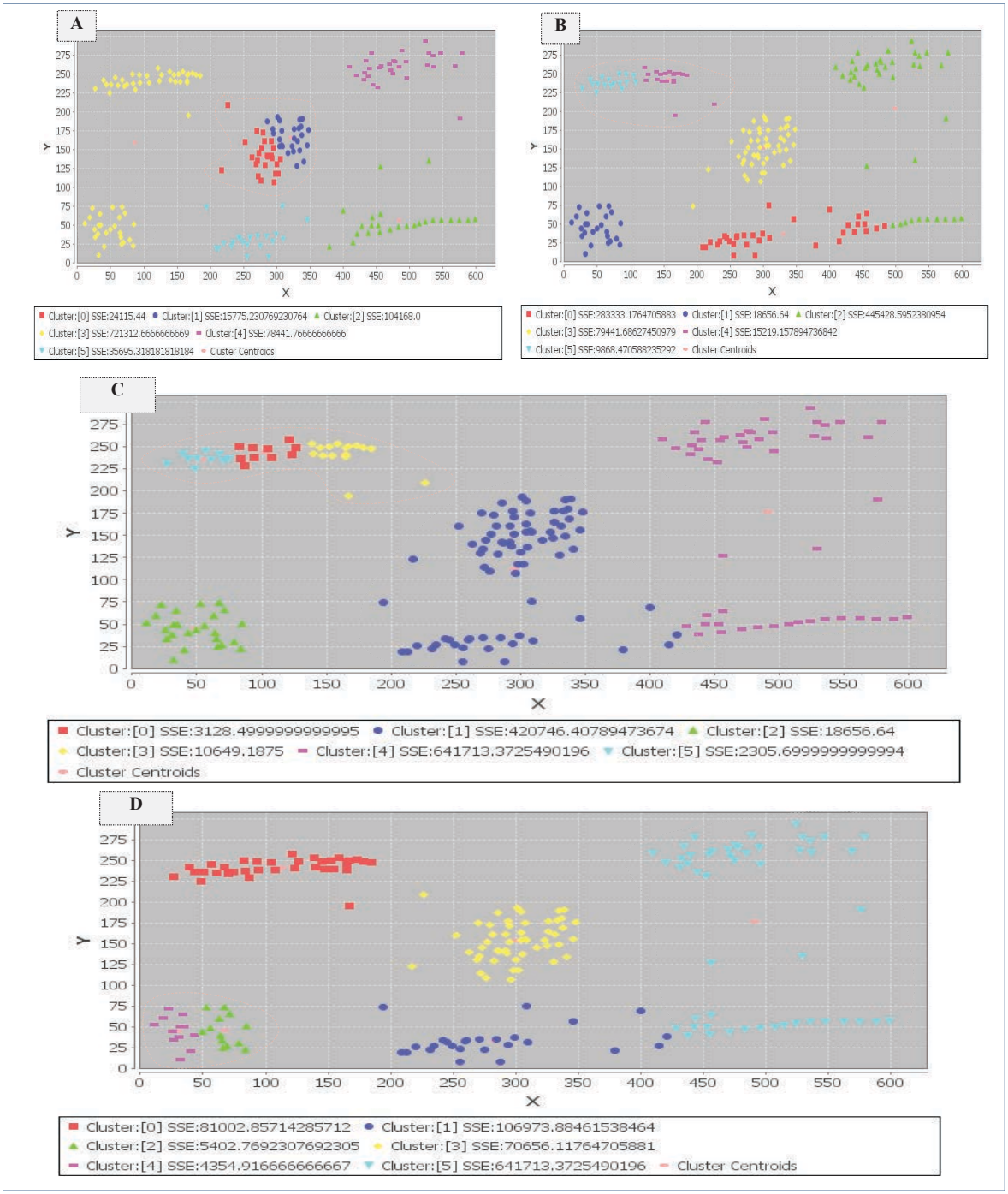

Figure 5. Low accurate results obtained with standard K-means algorithm with (Rnoisy dataset)

matrix" between each pair of clusters' centroids including all clusters within the dataset. The resulting distance matrix is n*n matrix where n is the number of all the clusters in the dataset including the clusters resulted after the split process. This matrix is used to identify the two most close cluster centroids with the dataset in order to check if they both belong to one true cluster. Single linkage "nearest neighbor or shortest distance" concept is applied for this purpose, where it calculates and finds the shortest distance between a pair of objects each of them is located in one of the selected closest clusters. Then, the merging will take place if at least one of the following conditions is true:

1: The distance between the two nearest points that belong to the clusters with closest centroids is less than or equals to ε.

2: The point in the middle between the selected pair of objects is checked if it is in a density unit and has a number of points equal or larger than MinPts within ε radius that belong to the closest clusters.

If one of the above conditions is fulfilled, then the two closest clusters are merged, then the distance matrix is calculated and the process is repeated as many times as the split process. Otherwise, the second shortest distance from the distance matrix is selected and the process is repeated.

At the end of this process, the number of resulting clusters is the same as the number of clusters resulted from the standard K-means which is k parameter.

4.2 Anti-Noise proposed Method:

In standard K-means clustering, when applied on datasets containing noise objects , the results are -most of the time- of low accuracy. As the standard K-means includes all noise objects in the calculations, the end result will lack accuracy, in addition, standard K-means will either merge some true clusters into larger clusters or -in some cases- identify groups of noise points as clusters.

The researcher has developed a way to decrease the effect of noise objects on the end results through observations during lots of experiments applied on different datasets some of which were explained in the previous section. The researcher concluded that -most of the time- the noise points were in non-density unit as well as most of the points far from the centroids even when the K;2-means results are highly accurate. Based on that conclusion, the researcher build the Anti-noise method which is mainly about neglecting points far from the centroids in order to acquire high accuracy results.

Anti-noise method starts with calculating distances between each point in a cluster and its centroid where the distance are listed in an ascending order. Starting with the farthest points -which has the largest distance-, Anti-noise checks if that point is located in density unit or not. If it was located in non-density unit, then it is temporarily neglected and the next farthest point is check. This process goes on until a point that is located in a density unit is found or all the points are checked. In the case of finding no points in density unit, then the whole cluster is neglected, and the next cluster is checked in the same manner.

After checking all clusters within the dataset, standard K-means is applied again on the clusters without the neglected points. The results of such run will have higher accuracy than those when including the neglected points and the resulting centroids will be very close to the true centroids. Afterwards, each of the neglected points is assigned to the cluster with nearest centroid.

4.3 DSMK-means Algorithm Pseudo-Code:

Suppose that we are going to partition X={x1,x2,...,xn} which is a dataset with n number of objects, and k is an input parameter equal to number of clusters required.

1 RUN standard K-means algorithm

2 COMPUTE sum square error "SSE" for each cluster.

3 COMPUTE ε and MinPts value for cluster with minimum "SSE" value. Eps or ε , the radius that delimitate the neighbourhood area of a point (Eps-neighbourhood) MinPts, the minimum number of points that must exist in the Eps-neighbourhood.

4 FOREACH cluster

5 Create list of clusters with centroids Ci in non-density units.

6 IF number of point's within Eps-neighborhood contains $<$ MinPts (centroid in density unit).

7 THEN add cluster to list Ci

8 ENDFOR EACH

9 CASE METHOD OF

10 **CASE-ONE "If one or more centroids Ci is not in density unit":** (Spit and Merge started)

11 Declare count=0 represent number of splitting operation SPLIT PROCESS

12 For all Clusters Ci List

13 IF centroid Ci is in non-density unit

14 Split Ci cluster into two clusters with standard K-means algorithm (K=2)

15 DELETE Ci cluster and ADD split clusters to List

16 Increase count by 1

17 ENDIF

18 ENDFOR
MERGE PROCESS

19 WHILE count != 0

20 Calculate centroids distance matrix

21 FOR each item in distance matrix

22 Find the tow nearest clusters centroids from all dataset clusters using distance matrix

23 Find the two closest points from the two closest clusters using single linkage.

24 IF (distance between two nearest points is less than or equals to ε) THEN Merge those two clusters.

25 ELSE, Find middle point

26 IF (middle point between two nearest points from two closest clusters "Single Linkage" is in density unit) THEN

27 Merge those two clusters.

28 Decrease count by 1

29 ELSE,

30 Go To step 22

31 ENDIF

32 ENDFOR

33 IF no clusters are merged THEN

34 Merge tow nearest clusters' centroids

35 Decrease count by 1

36 ENDIF

37 ENDWHILE

38 **CASE-TWO " If all centroids are in density units":**

39 FOR each cluster Ci

40 FOR each point Pn

41 Compute distance between the centroid and Pn where n is the number of points in a cluster Ci

42 ENDFOR

43 Sort the distances in ascending order in each Ci

44 ENDFOR

45 FOR all centroids

46 WHILE the farthest point from centroid Ci is not in density unit

47 Neglect the point and considered as noise

48 ENDFOR

49 RUN standard K-means algorithm without the neglected points "noise"

50 Depending on the K-means cluster results; the neglected points are assigned to the closest cluster.

51 ENDCASE

52 End ALGORITHM

4.4 Advantages and limitations of DSMK -means algorithm:

Advantages:

1. The algorithm can handle large numbers of datasets as it solves two different problems in standard K-means (sensitivity to noise, complex shapes).
2. The algorithm has combined the characteristics of partition clustering and density clustering concepts.
3. The algorithm is not difficult to implement.
4. The algorithm does not require any additional parameters more than the standard K-means algorithm.
5. The algorithm is less sensitive to noise and outlier.
6. Algorithm got better accuracy when datasets containing clusters with complex shapes and sizes.

Limitations:

1. Algorithm did not reduce the number of parameters needed.
2. Algorithm increases the computational complexity.
3. In some rare cases, algorithm had bad results as the standard K-means.

The next Figures exhibits the flow chart of the DSMK-means algorithm:

Figure **6**: Flowchart of DSMK-means algorithm.

5 Experimental Results

Description of the datasets used in experiments and the measurement techniques in addition to measuring the accuracy of the proposed algorithms' results to ensure their ability in delivering better results than other algorithms.

5.1 Datasets Description

This subsection describes and identifies the specifications of datasets used in the experiment on the proposed algorithm. The datasets varied between real-world and artificial datasets.

5.1.1 Artificial datasets

The Artificial datasets used in the experiments are:

– **Rnoisy dataset:** Artificially polluted datasets with noise generated by the researcher with two dimensions, this dataset designed in a way to contain a lot of noise and outliers. This dataset consists of 188 points distributed in six true clusters. Values of the generated artificial dataset are used to assess the level of K-means algorithm accuracy and ability to identify true clusters.

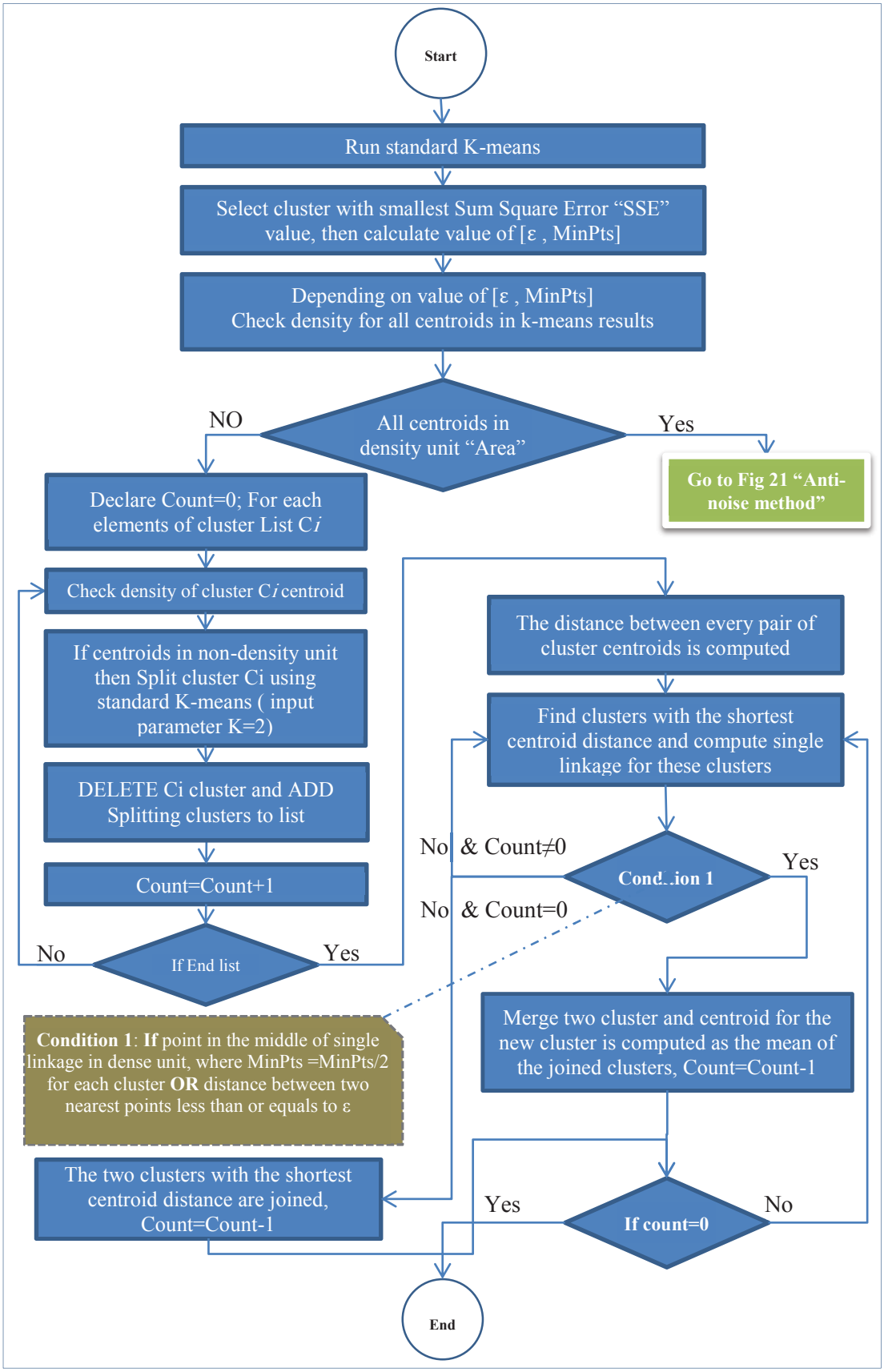

Figure 6. Flowchart of DSMK-means algorithm.

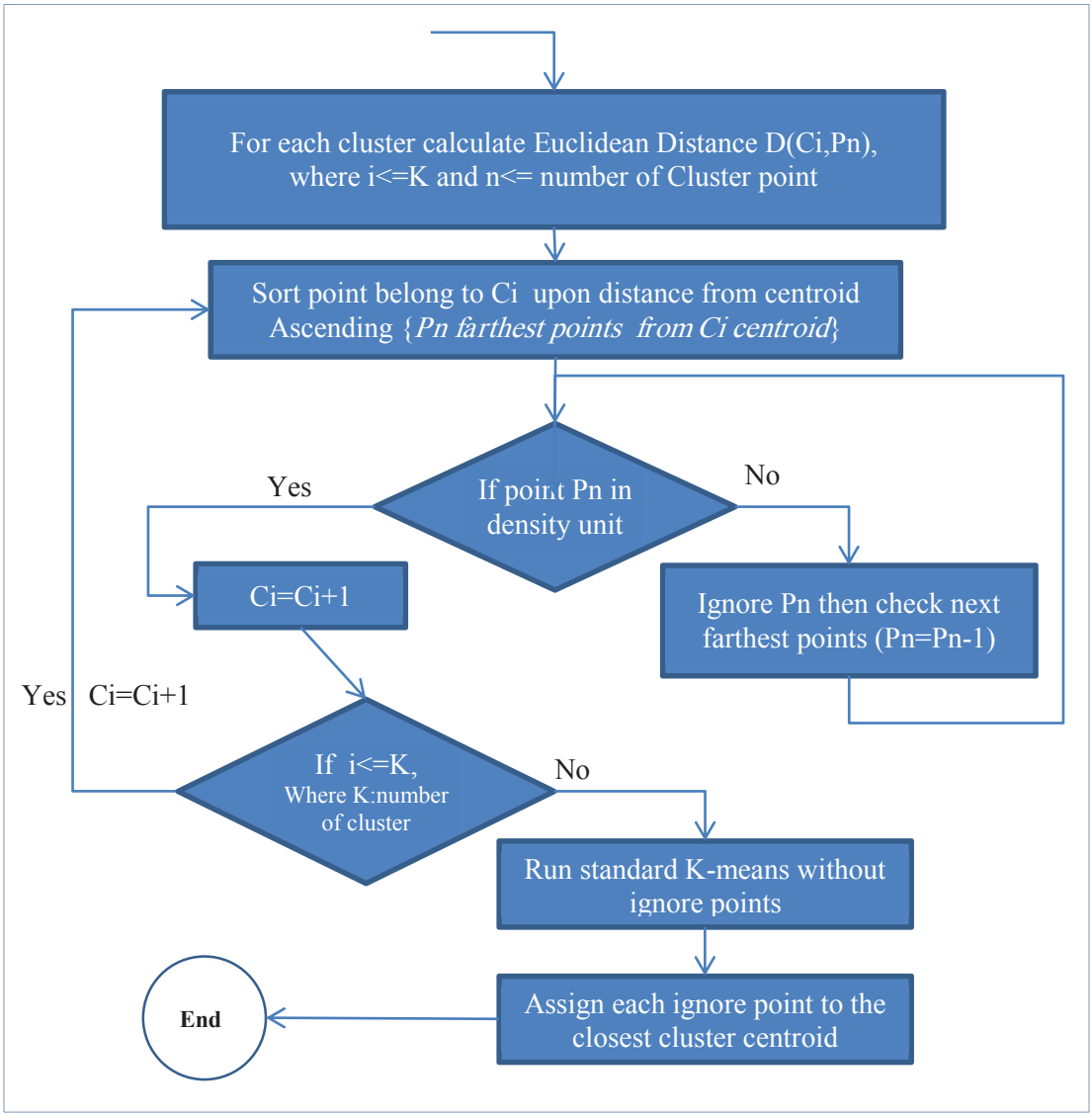

Figure 7. Flowchart of DSMK-means algorithm.

– **Ground_Separation dataset:** Dataset contains six different complex shapes and sizes generated by the researcher with two dimensions. The dataset consists of 479 points distributed into six true clusters. This dataset was designed to be "hard" because of different clusters' shapes. It is designed to measure K-means ability to identify clusters with complex shapes.

– **Separation_2Circle dataset:** Dataset generated with two different complex shapes and sizes with two dimensions. The dataset consists of 337 points in two true clusters. This dataset is designed to be "hard" because of different clusters' shapes. It is designed to measure K-means ability to identify clusters with complex shapes.

– **Document_Sim dataset:** Dataset generated so that many noises are scattered. The dataset consists of 200 points in five true clusters. The dataset is designed to be "hard". i.e. there is a large number of outliers and noise are scattered between five true clusters. It is designed to measure K-means ability to identify true clusters in noisy datasets.

– **Aggregation dataset:** Dataset consists of the seven perceptually distinct clusters with different shapes. The dataset consists of 788 points distributed in seven true clusters. The dataset is designed to be "hard" in order to measure K-means ability to identify clusters with complex shapes.

5.1.2 Real datasets

All datasets used in the following experiments and more can be found in UCI Machine Learning Repository [11] 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.

– **Iris Dataset:** This is perhaps the best known database to be found in the pattern recognition and clustering literature. The Iris flower dataset or Fisher's Iris dataset is a multivariate dataset introduced by Sir Ronald Fisher (1936) as an example of discriminant analysis. It is sometimes called Anderson's Iris dataset because Edgar Anderson collected the data to quantify the morphologic variation of Iris flowers of three related species. 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 two; the latter are not linearly separable from each other.

– **Wine recognition Dataset:** This dataset is the result of a chemical analysis of wines grown in Italy but derived from three different cultivars as exhibited in Figure 27. The analysis determined the quantities of 13 constituents found in each of the three types of wines. Table 5.2 illustrates the dataset specifications.

– **Weblogs dataset:** This dataset is real with two dimensions; it is suitable to describe the performance of K-means when dealing with datasets containing clusters with different complex shapes and sizes. The datasets contain two metadata of weblog entries: number of visits and purchase. The datasets had been gathered by crawling from the WWW.

– **Image_Extraction dataset:** This dataset is simplified extraction of local image features. It takes image data as input and returns a dataset with feature vectors computed from image blocks on a regular grid. The dataset consists of samples from each of two species of image.

Table 1 presents a summary of artificial datasets used in this thesis while Table 2 presents a summary of real datasets. The details of each dataset are described.

5.2 Cluster validity measures

Evaluation of clustering results sometimes is referred to as cluster validation. There have been several suggestions for a measure of quality of clustering algorithms. Such a measure can be used to compare how well different clustering algorithms perform on a set of data. These measures are usually tied to the type of criterion being considered in assessing the quality of a clustering algorithm [12].

Measuring clustering validity

Table 1. Summary of all artificial datasets information

Datasets Name	clusters	type	dimension
Rnoisy	6	Integer	2
Ground_Seperation	6	Integer	2
Separation_2Circle	2	Integer	2
Document_Sim	5	Integer	2
Aggregation	7	Real	2

Table 2. summary of all Real datasets information

Datasets Name	clusters	type	dimension
IRIS	3	Real	3
Wine recognition	3	Real, Integer	12
Web Log	3	Real	3
Image_Extraction	2	Real	2

1. **External validity:** In external validity, clustering results are evaluated based on already clustered data such as known class labels and external benchmarks. Such benchmarks consist of a set of pre-classified items, and these sets are often created by human (experts). Thus, the benchmark sets can be thought of as a gold standard for evaluation. These types of evaluation methods measure how close the clustering is to the predetermined benchmark classes. In summary, external evaluation measures similarity of clustering against known class labels.

2. **Internal validity:** When a clustering result is evaluated based on the data that was clustered itself, this is called internal validity. These methods usually assign the best score to the algorithm that produces clusters with high similarity within a cluster and low similarity between clusters. One drawback of using internal criteria in cluster evaluation is that high scores on an internal measure do not necessarily result in effective information retrieval applications. In summary, internal validity measure the goodness of a clustering without any external information just like Sum of Squared Error (SSE) [13], Akaike Information Content score (AIC) [14] [15], The Bayesian Information Criterion (BIC) [16], and Sum Of Average Pairwise Similarities (SAPS) [17].

5.3 Performance Evaluation of DSMK-means algorithm

To test the performance of "DSMK-means" algorithm, the researcher here introduces the datasets used in the test and reviews the results of the experiments, comparing the results with standard K-means algorithm and "BNAK-Divide-and-Merge Clustering Algorithm (BNAKDAM) [8]".

5.3.1 Datasets selection

The performance evaluation of DSMK-means algorithm is applied on nine different artificial and real-world datasets (Ground_Separation, Separation_2Circle,Rnoisy,Aggregation,Document_Sim, Weblogs, Image_Extraction, Wine recognition, and Iris). Furthermore, the performance of DSMK-means algorithm is evaluated using popular internal clustering validity indices, which employed to evaluate the clustering results, such indices include: Sum of Square Errors (SSE), Akaike Information Content (AIC), The Bayesian Information Criterion (BIC), and Sum of Average Pairwise Similarities (SAPS); which were described in the previous section. The results of such evaluation are compared with standard K-means and BNAKDAM algorithms in order to identify the differences.

Table 3 and Table 4 show the comparison of the three clustering algorithms: Standard K-means, BNAKDAM, and proposed DSMK-means algorithm.

Table 3. Clustering algorithms mean results of artificial datasets over 50 runs (K is an input parameter obtained from user, which represent clusters number).

Dataset	algorithm	K	SSE	AIC	BIC	SAPS
Ground_Separation	K-means	6	**2855774.813**	**21565.0806**	**21565.74618**	**447.3896701**
	BNAKDAM		**4845704.853**	**21528.5757**	**21529.24129**	**439.9237214**
	DSMK-means		**6159061.753**	**21494.59273**	**21495.25831**	**435.4722115**
Separation_2Circle	K-means	2	**1646936.594**	**14267.35041**	**14267.87804**	**333.0847394**
	BNAKDAM		**2154097.931**	**14254.68015**	**14255.20778**	**328.7270521**
	DSMK-means		**2577578.554**	**14253.17946**	**14253.70709**	**325.1817196**
Document_Sim	K-means	5	**1306929.15**	**14744.83774**	**14745.38305**	**342.5543868**
	BNAKDAM		**858283.6622**	**14586.10662**	**14586.65193**	**341.2153644**
	DSMK-means		**563265.0873**	**14480.64838**	**14481.19369**	**339.6457059**
Rnoisy	K-means	6	**1486294.884**	**21177.29713**	**21177.57129**	**183.0293608**
	BNAKDAM		**865616.0755**	**19627.6786**	**19727.92254**	**172.2823015**
	DSMK-means		**601464.5215**	**19143.84816**	**19144.07983**	**167.9597529**
Aggregation	K-means	7	**23875.26004**	**49524.31143**	**49525.20795**	**771.7122547**
	BNAKDAM		**24395.60533**	**49454.00687**	**49454.90339**	**772.9397117**
	DSMK-means		**25387.49156**	**49358.04323**	**49358.93976**	**769.6763346**

Table 4. Clustering algorithms mean results of real datasets over 30 runs (K is an input parameter obtained from user, which represent the number of clusters)

Dataset	algorithm	K	SSE	AIC	BIC	SAPS
Weblogs	K-means	3	**1001992.83**	**8670.09057**	**8670.369323**	**187.3771721**
	BNAKDAM		**1326821.065**	**8747.432511**	**8747.711265**	**187.8597177**
	DSMK-means		**1552251.178**	**8631.13863**	**8600.417384**	**186.1648071**
Image_Extraction	K-means	2	**1799087.147**	**9185.98039**	**9186.28142**	**191.6552485**
	BNAKDAM		**2991817.961**	**9305.285303**	**9305.586333**	**184.5466748**
	DSMK-means		**3634144.897**	**9069.956687**	**9070.257717**	**180.7335421**
Wine recognition	K-means	3	**996130.8991**	**9059.918373**	**9060.219403**	**196.2587883**
	BNAKDAM		**798077.4038**	**8892.310846**	**8892.606815**	**194.5382956**
	DSMK-means		**590473.4668**	**8624.408488**	**8624.694335**	**190.6957176**
IRIS	K-means	3	**223.37357471**	**9926.572673**	**9926.748764**	**149.5602358**
	BNAKDAM		**162.35621565**	**9912.365894**	**9950.464253**	**149.5591551**
	DSMK-means		**158.28685748**	**9899.236799**	**9899.4128911**	**149.5536856**

Table 3 shows the comparison applied on the artificial datasets described in previous subsection, while Table 4 shows the comparison applied on the real datasets described before in 5.1.2 subsection.

It is observed from the experiment results on artificial datasets described in Table 3, that DSMK-means has the best results among the other two algorithms in AIC, BIC, and SAPS indices, while it did not have the best results with SSE index. The reason of the SSE high score in DSMK-means algorithm depends on that the shape of cluster, as SSE sums the square differences between each attribute value and the corresponding one in the cluster centroid. In another example, SSE results for the Separation_2Circle dataset using standard K-means algorithm as shown in Figure 8.A were lower than the results using DSMK-means as in Figure 8.F,E. It is known that the lower SSE score is the better, but in this case it is visually clear that Figure 8.A -which obtained lower SSE value- is very bad clustering result compared to the resulting cluster of DSMK-means. This observation indicates the SSE score can not be used to judge the clustering accuracy in cases of complex shapes. While the other indices give more accurate indication for the best clustering results.*Table*

It is observed that DSMK-means algorithm scores smaller values for each type of clustering validity indices (SSE, SAPS, AIC, and BIC) where the sometimes, DSMK-means algorithm scores big values for clustering validity index (SSE). The value of measurement algorithm depends on the nature of algorithm formula and datasets clusters shapes, however DSMK-means could identify clusters with different complex shapes that may increase the result of SSE index while decrease the rest of indices results. Obviously the clustering results of the DSMK-means clustering algorithm perform best compared to k-means and BNAKDAM clustering algorithms.

To prove the efficiency of DSMK-means algorithm, the graph of datasets is shown to make a comparison between the results of the standard K-means and DSMK-means algorithm. The BNAKDAM results were not shown here as they were similar to the graphs in Figure 8.A,B,C.

The Separation_2Circle is composed of two different clusters with different shapes. In Figure 36, the results show that the DSMK-means can detect both clusters with different shapes and sizes while the standard K-means cannot deal with this kind of dataset.

Each cluster identified by a different plotting character and color. It is observed that the standard K-means get inefficient results as it split the inner circle true cluster into two different groups as well as the outer circle; and merged each part of the inner circle with another part from the outer circle. It's observed that standard K-means always get the same results with this dataset which are very bad results (which are plotted as red square and blue circle in Figure 8.A). While the proposed DSMK-means algorithm gives, more efficient and accurate results in identifying each cluster very close to true ones. It is worth mentioning that the results in Figure 8.E and F are the most common case in the results of the algorithm.

Figure 9 shows the results of running the K-means and DSMK-means algorithms with Ground_Separation dataset, which consists of 6 clusters. The shape of this dataset is one of the most complicated shapes to be tested on standard K-means, which is can not provide accurate clustering results that are close to the true clusters.

It is observed that the standard K-means get inefficient results as it split the ring-shaped cluster (which is plotted in Figure 9.A) into two different groups, one of them was identified as single cluster, while the other was merged with one of the circle-shaped clusters in the left-bottom corner.

On the other hand, the proposed DSMK-means algorithm gives more efficient and accurate results in identifying each cluster very close to the true ones (which are plotted in Figure 9.B and C).

Finally, DSMK-means algorithm is in general an improved clustering algorithm based on standard K-means. It consists of two main stages: split and merge stage, and anti-noise stage; these stages enable the algorithm to detect different clusters with different shapes, sizes and densities. Moreover, DSMK-means is robust to noises. Experiments demonstrate that DSMK-means clustering algorithm outperforms the traditional K-means and BNAKDAM clustering algorithms. However, DSMK-means has higher computation complexity compared to standard K-means.

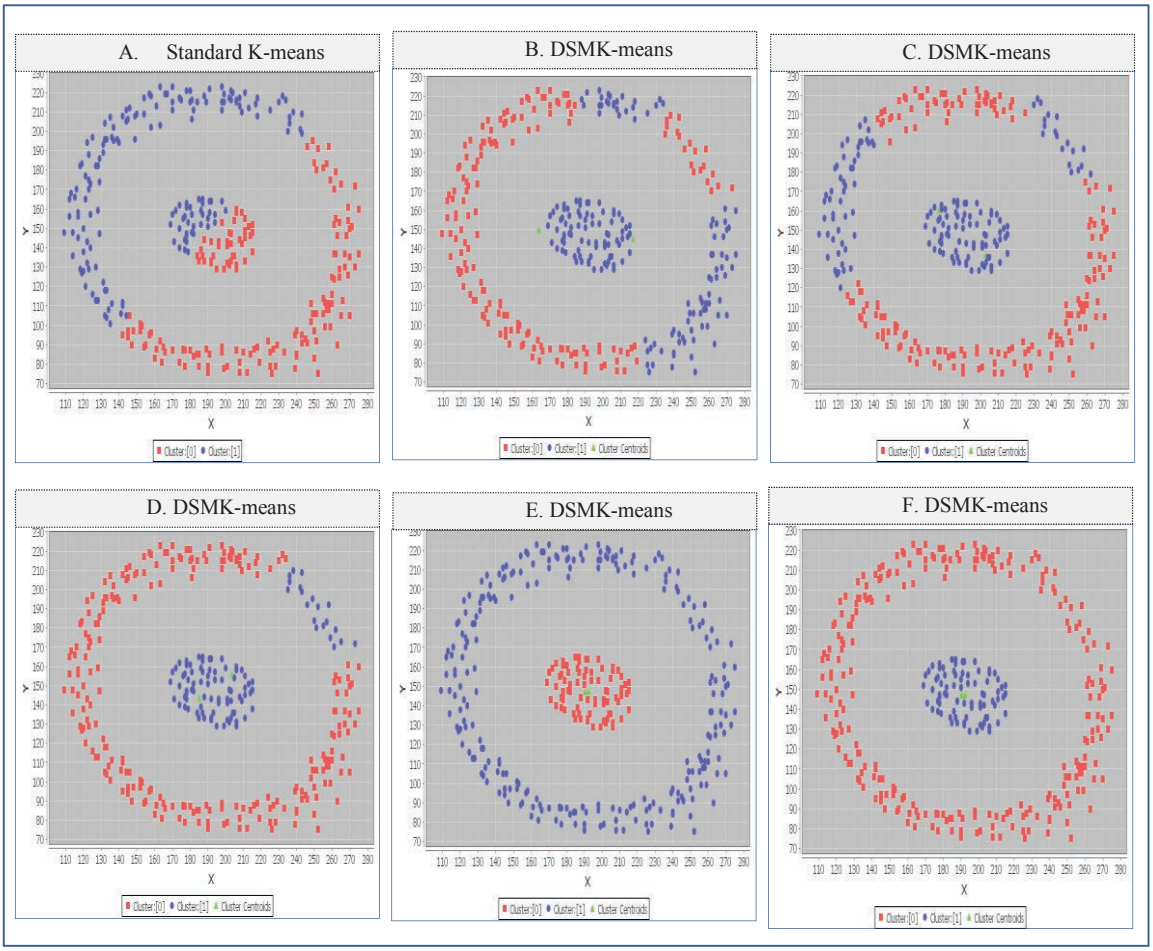

Figure 8. Results of running K-means and DSMK-means with k=2, on Separation_2Circle dataset

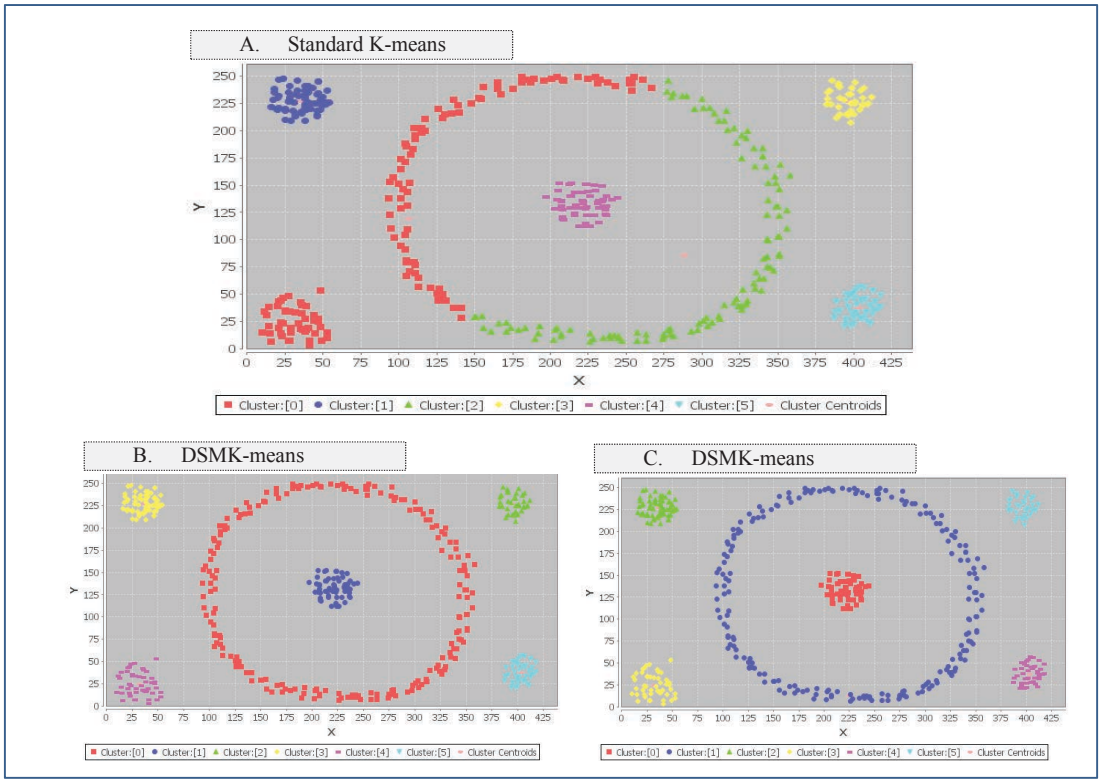

Figure 9. Results of running K-means and DSMK-means with k=6, with Ground_Separation dataset.

6 Conclusion

In this paper, researchers have introduced new clustering algorithm DSMK-means "Density-based Split-and-Merge K-means clustering Algorithm. This algorithm was developed from k-means, which suffers from unsatisfactory accuracy when the dataset contains clusters with different complex shapes, sizes, noise and/or outliers. DSMK-means included Split and Merge technique, which are proposed to overcome standard K-means merging, or splitting true clusters when working with datasets that contain clusters with different complex shapes. In addition, DSMK-means included Anti-noise technique, which was proposed to overcome the sensitivity of standard K-means algorithm to noise. DSMK-means algorithm includes solutions for cluster with complex shapes and datasets with noisy objects. Experimental results demonstrate that the algorithm gives efficient performance when dealing with several virtual and real-world datasets. In addition, it is observed that the proposed method is able to define clusters with different shapes that K-means can not.

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.W540W4226

[3] D. Birant and A. Kut, "ST-DBSCAN: an algorithm for clustering spatial–temporal data," Data & Knowledge Engineering, vol. 60, pp. 208-221, 2007.W540W4226

[4] Oded Maimon (Editor) and Lior Rokach (Editor),.: Springer; 1 edition, September 1, 2005.W540W4226

[5] 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.W540W4226

[6] H. Vinod, "Integer programming and the theory of grouping," Journal of the American Statistical Association, vol. 64, pp. 506–519, 1969.W540W4226

[7] Anil K. Jain, "Data Clustering: 50 Years Beyond K-Means," Pattern Recognition Letters, 2009.W540W4226

[8] Zhiwu Huang, DongZhan Zhang, and JiangJiao Duan, "BNAK-Divide-and-Merge Clustering Algorithm," in ICISE '09 Proceedings of the 2009 First IEEE International Conference on Information Science and Engineering , 2009 , pp. 810-813.W540W4226

[9] Jan Carlo Barca and Grace Rumantir, "A Modified K-means Algorithm for Noise Reduction in Optical Motion Capture Data ," in Computer and Information Science, 2007. ICIS 2007. 6th IEEE/ACIS International Conference on, 11-13 July 2007.W540W4226

[10] M. Muhr and M. Granitzer, "Automatic Cluster Number Selection Using a Split and Merge K-Means Approach ," IEEE Conference Publications, 20th International Workshop on Database and Expert Systems Application, pp. 363 - 367, 2009.W540W4226

[11] University of Massachusetts Amherst. Funding support from the National Science Foundation. UC Irvine Machine Learning Repository. [Online]. http://archive.ics.uci.edu/ml/W540W4226

[12] Clustering analysis. wikipedia. [Online]. http://en.wikipedia.org/wiki/Cluster_analysis Evaluation_of_clustering_resultsW540W4226

[13] Oded Maimon and Lior Rokach, Data Mining And Knowledge Discovery Handbook, 1st ed., 978-0387244358, Ed.: amazon, 2005.W540W4226

[14] H. Bozdogan, "Akaike's Information Criterion and Recent Developments in Information Complexity," Journal of Mathematical Psychology, vol. 44, pp. 62–91, 2000.W540W4226

[15] wikipedia. [Online]. http://en.wikipedia.org/wiki/Akaike_information_criterionW540W4226

[16] G. Schwarz, "Estimating the dimension of a model," Annals of Statistics, vol. 6(**??**), pp. 461-464, 1978.W540W4226

[17] Y. Zhao and G. Karypis, "Criterion functions for document clustering," Technical report, Department of Computer Science, University of Minnesota / Army HPC Research Center, 2002.W540W4226 x