Performance Analysis of Clustering Techniques to Normal and Uniform Distribution of Data Points

D. Prabhu[#], Dr.K.Vivekanandan[*], R. Vijayanandh[#]

[#]Department of Computer Applications, Imayam College of Information Technology
Tamil Nadu, India
[*]Department of School of Management, Bharathiar University
Tamil Nadu, India

Abstract— Clustering technique is one of the most important research areas in the field of data mining. This paper proposes an improved K-Means clustering algorithm form partition based clustering algorithms. It determines the initial centroid of the cluster and gives more efficient performance and reduces the time complexity of the large data sets. The data set used here is banking data. Fuzzy C-Means clustering algorithm also implemented since it produces the soft clusters. Finally the proposed algorithm is compared and analyzed for the Normal and Uniform distribution of data point with K-Means, Fuzzy C-Means and K-Medoids. The computational complexity and performance is showed much better with improved K-Means algorithm.

***Keywords*— Clustering, K-Means, Fuzzy C-Means, K-Mediods, Normal and Uniform distribution**

I. INTRODUCTION

Data-mining tools have great potential for counterterrorism, but to realize that potential fully, max and more research is needed. The government should support this research. A government policy for this research should take into account the context in which these tools may eventually be deployed. This means research on privacy-protecting technology and even some analysis of privacy policy issues should be included [1].

Data mining is a part of a process called KDD-knowledge discovery in databases this process consists basically of steps that are performed before carrying out data mining, such as data selection, data cleaning, pre-processing, and data transformation. Data mining, or knowledge discovery, is the computer-assisted process of digging through and analyzing enormous sets of data and then extracting the meaning of the data. Data mining tools predict behaviors and future trends, allowing businesses to make proactive, knowledge-driven decisions. Data mining tools can answer business questions that traditionally were too time consuming to resolve. They scour databases for hidden patterns, finding predictive information that experts may miss because it lies outside their expectations [2].

Most data-based modeling studies are performed for a particular application domain. Hence, domain-specific knowledge and experience are usually necessary in order to come up with a meaningful problem statement. Unfortunately, many application studies are likely to be focused on the data mining technique at the cost of a clear problem statement. In this step, a modeler usually specifies a set of variables for the unknown dependency and, if possible, a general form of this dependency as an initial hypothesis [3].

All raw data sets which are initially prepared for data mining are often large; many are related to humans and have the potential for being messy [4]. Real-world databases are subject to noise, missing, and inconsistent data due to their typically huge size, often several gigabytes or more. Data reduction; can reduce the data size by aggregating, eliminating redundant features. The data processing techniques, when applied prior to mining, can significantly improve the overall data mining results [5].

The proposed paper is organized is as follows: Section 2 describes the literature review. Section 3 and 4 explains the data mining task and discussed the proposed improved K-Means clustering algorithm respectively. Section 5 shown experimental results with comparative analysis and finally the conclusion and future work is discussed.

II. LITERATURE REVIEW

This section provides background information related to this paper. It starts with explaining the principles of clustering algorithms. Many organizations have recognized the importance of the knowledge hidden in their large databases and, therefore, have built data warehouses. When speaking of a data warehousing environment, we work on two characteristics namely, analysis and multiple updates. These characteristics or requirements tend to new approach called Data Mining [6]. Data Mining is a process of discovering various models, summaries, and derived values from a given collection of data. In other words, the uniform effect has been dominated by the large distance between two clusters. The uniform effect of K-means clustering on “true” clusters with different sizes does exist [7].

Data mining has been defined as the application of data analysis and discovery algorithms that under acceptable computational efficiency limitations produce a particular enumeration of patterns over the data. Several data mining tasks have been identified, e.g., clustering, classification and summarization. Our area of concentration is clustering. In data warehouse, data is not updated immediately when insertions and deletions on the operational databases occur. Updates are collected and applied to the data warehouse periodically in a batch mode, e.g., each night. Due to the very large size of the databases, it is unfeasible to cluster entire data for every updates. Hence, it is highly desirable to perform these updates incrementally [8].

Data Mining is the notion of all methods and techniques, which allow analyzing very large data sets to extract and discover previously unknown structures and relations out of such huge heaps of details. This information is filtered, prepared and classified so that it will be a valuable aid for decisions and strategies [9].

Clustering is a process in which a group of unlabeled patterns are partitioned into a number of sets so that similar patterns are assigned to the same cluster, and dissimilar patterns are assigned to different clusters. There are two goals for a clustering algorithm, determining good clusters and doing so efficiently. Clustering has become a widely studied problem in a variety of application domains, such as in data mining and knowledge discovery [10].

III. DATA MINING TASK

Data mining is the task of discovering interesting patterns from large amounts of data where the data to be stored in databases, data warehouse or other information repositories. It is a young interdisciplinary field, drawing from areas such as database system, data warehousing, statistics, machine learning, data visualization, information retrieval and high performance computing.

The primary task of data mining is prediction and description. Prediction involves using variables or fields in database to guess unknown or future values of other variables of interest. Description focuses on finding human-interpretable patterns describing the data. The descriptions are achieved by using the following primary data mining tasks [3].

- Classification: The task is to learn to assign instances to predefined classes.
- Regression: Is learning a function which maps a data item to a real valued prediction Variable.
- Clustering: A collection of data objects that object is similar to one another and thus can be treated collectively as one group.
- Summarization: Involves methods for finding a compact description for a subset of data.
- Sequence Analysis: It models sequential patterns, like time series analysis, gene sequence etc., and the goal are to model the status of the process generating the sequence.

Association Rule: The aim of association rule mining is to find patterns in a transaction database. The database contains transactions which consist of a set of items and a transaction identifier. Association rules are implications of the form X -> Y where X and Y are two disjoint subsets of all available items. X is called the antecedent or LHS (left hand side) and Y is called the consequent or RHS (right hand side).

IV. IMPROVED K-MEANS CLUSTERING ALGORITHM

In the Improved clustering method discussed in this paper, both the phases of the original K-Means algorithm are personalized to improve the efficiency. A popular clustering method that minimizes the clustering error is the K-Means algorithm [11]. However, the K-Means algorithm is a local search procedure and it is well known that it suffers from the serious drawback that its performance heavily depends on the initial starting conditions.
The Improved method

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

Steps:

Phase 1: Determine the initial centroids of the clusters by using Algorithm 1.
Phase 2: Assign each data point to the appropriate clusters.

In the first phase, the initial centroids are determined systematically so as to produce clusters with better accuracy [12]. The second phase makes use of a variant of the clustering method. It starts by forming the initial clusters based on the relative distance of each data-point from the initial centroids. These clusters are subsequently fine-tuned by using a heuristic approach, thereby improving the efficiency. The two phases of the improved method are described below as Algorithm 1 and Algorithm 2. [5].

Algorithm : Finding the initial centroids

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

Output: A set of k initial centroids .

Step 1 : Set m = 1;

Step 2 : Compute the distance between each data point and all other data- points in the set D;
Step 3 : Find the closest pair of data points from the set D and form a data-point set Am (1<= m <= k) which contains these two data- points, Delete these two data points from the set D;

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

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

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

Step 7 : For each data-point set Am (1<=m<=k) find the arithmetic mean of the vectors of data points in Am, these means will be the initial centroids [13].

The above algorithm describes the method for finding initial centroids of the clusters [12], [14]. Initially, compute the distances between each data point and all other data points in the set of data points. Then find out the closest pair of data points and form a set A1 consisting of these two data points, and delete them from the data point set D. Then determine the data point which is closest to the set A1, add it to A1 and delete it from D. Repeat this procedure until the number of elements in the set A1 reaches a threshold.

At that point go back to the second step and form another data-point set A2. Repeat this till 'k' such sets of data points are obtained. Finally the initial centroids are obtained by averaging all the vectors in each data-point set. The Euclidean distance is used for determining the closeness of each data point to the cluster centroids. The distance between one vector X = (x1, x2, … , xn) and another vector Y = (y1, y2, … , yn) is obtained as

$$d(X,Y)=\sqrt{((X1-Y1)^2+(X2-Y2)^2+...+(Xn-Yn)^2)} \quad (1)$$

The distance between a data point X and a data-point set D is defined as , d(X, D) = min (d (X, Y), where Y∈ D).The initial centroids of the clusters are given as input to the second stage, for assigning data points to appropriate clusters.

The steps involved in this phase are outlined as Algorithm 3.2 [3].

Algorithm : Assigning data-points to clusters

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

Output: A set of k clusters

Steps:

```
Step 1 : Compute the distance of each data-point di (1<=i<=n) to all the centroids cj (1<=j<=k) as d(di , cj);
Step 2 : For each data-point di, find the closest centroid cj and assign di to cluster j.
Step 3 : Set ClusterId[i]=j; // j:Id of the closest cluster
Step 4 : Set Nearest_Dist[i]= d(di, cj);
Step 5 : For each cluster j (1<=j<=k), recalculate the centroids;
Step 6 : Repeat
Step 7 : For each data-point di,
        Step 7.1 : Compute its distance from the centroid of the present nearest cluster;
        Step 7.2 : If this distance is less than or equal to the present nearest distance, the data-point stays in the cluster;
                Else
        Step 7.2.1 For every centroid cj (1<=j<=k) Compute the distance d(di, cj);
Endfor;
        Step 7.2.2      Assign the data-point di to the cluster with the nearest centroid cj
        Step 7.2.3      Set ClusterId[i]=j;
        Step 7.2.4      Set Nearest_Dist[i]= d(di, cj);
Endfor;
Step 8 : For each cluster j (1<=j<=k), recalculate the centroids;
Until the convergence criteria is met.
```

The first step in Phase 2 is to determine the distance between each data-point and the initial centroids of all the clusters. The data-points are then assigned to the clusters having the closest centroids. This results in an initial grouping of the data-points. For each data-point, the cluster to which it is assigned (ClusterId) and its distance from the centroid of the nearest cluster (Nearest_Dist) are noted. Inclusion of data-points in various clusters may lead to a change in the values of the cluster centroids.

For each cluster, the centroids are recalculated by taking the mean of the values of its data-points. Up to this step, the procedure is almost similar to the original K-Means algorithm except that the initial centroids are computed systematically [13].

The next stage is an iterative process which makes use of a heuristic method to improve the efficiency. During the iteration, the data-points may get redistributed to different clusters. The method involves keeping track of the distance between each data-point and the centroid of its present nearest cluster. At the beginning of the iteration, the distance of each data point from the new centroid of its present nearest cluster is determined.

If this distance is less than or equal to the previous nearest distance, that is an indication that the data point stays in that cluster itself and there is no need to compute its distance from other centroids [7]. These results in the saving of time required to compute the distances to k-1 cluster centroids. On the other hand, if the new centroid of the present nearest cluster is more distant from the data-point than its previous centroid, there is a chance for the data point getting included in another nearer cluster. In that case, it is required to determine the distance of the data point from all the cluster centroids. Identify the new nearest cluster and record the new value of the nearest distance. The loop is repeated until no more data-points cross cluster boundaries, which indicates the convergence criterion. The heuristic method described above results in significant reduction in the number of computations and thus improves the efficiency [13].

The Improved k-means clustering algorithm is implemented in the normal and uniform distributed data points.

V. EXPERIMENTAL RESULTS

The banking data set is used as the input of the system. Residential areas, location of bank are the classes taken in this dataset and the eight attributes are residential area1, location of bank2, location of bank 3, population size for area2, population size for area3, made up temperature controlling bank choice, maximum possible length of queues, rejection rate.

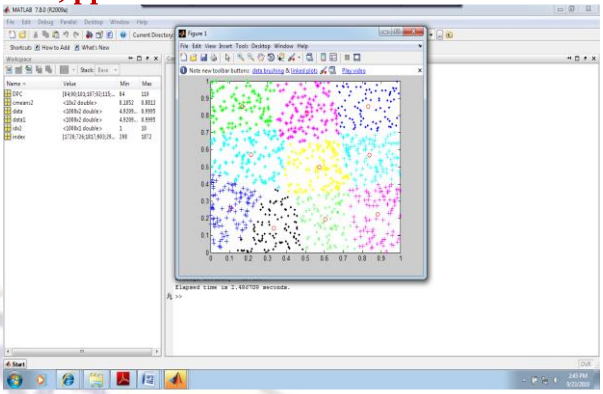
These data points are converted into normal and uniform distribution data point's. These normal and uniform distribution data points can be used to very flexible for users. Then the time complexity could be reduced by the normal and uniform distribution data points using clustering algorithms. This implementation should taken 1000 data points and 2 attributes and the number of clusters given by the user is 10 (k = 10) for normal and uniform distribution.

The number of clusters and data points is given by the user during execution of the program. The algorithm is repeated 1000 times to get efficient output. The cluster centers (centroids) are calculated for each cluster by its mean value and clusters are formed depending upon the distance between data points.

The experiment is carried out in normal and uniform distribution of data points with K-Means, Fuzzy C-Means, K-Mediods and improved K-Means clustering algorithm. Then performance analysis is made.

A. *Experiment A*

K-Means clustering algorithm for normal distribution data points is explained with executing results and followed by K-Means algorithm for uniform distribution data points. The experimental results are discussed for the K-Means algorithm. The resulting clusters of the normal distribution of K-Means algorithm is presented in different runs. For this work the system is used with specifications of core i3 processor and 3GHz, disk space on windows 7 system. The results may be varying by the system specification. The goal is to divide the objects into K clusters such that the Homogeneity score, which is calculated relatively to the centroids of the clusters, is minimized.



Name	Value	Min	Max
DPC	[84;90;101;107;92;115;...	84	119
cmeans2	<10x2 double>	0.1852	0.8813
data	<1000x2 double>	4.9209...	0.9995
data1	<1000x2 double>	4.9209...	0.9995
idx2	<1000x1 double>	1	10
index	[1726;726;1817;603;29...	398	1872

```
Elapsed time is 2.496709 seconds.
>>
```

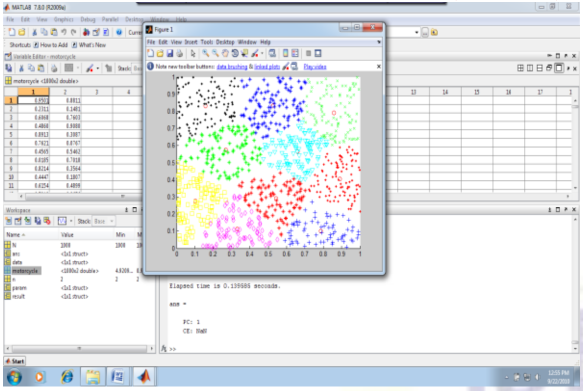
MATLAB 7.8.0 (R2009a)
File Edit Debug Parallel Desktop Window Help
Current Directory
Shortcuts How to Add What's New
Workspace
Stack: Base
Figure 1
File Edit View Insert Tools Desktop Window Help
Note new toolbar buttons: data brushing & linked plots Play video
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Start
OVR

Figure 1. K-Means Clustering Algorithm for Normal Distribution Data Points in Run1

- In run 1 cluster size = {102,95,81,75,86,117,76,102,135,131} and the time could be taken in elapsed time is 2496.6ms.
- In run 2 cluster size={118,101,93,119,99,90,94,61,99,126} and the time could be taken in elapsed time is 2141.4ms.
- In run 3 cluster size={136,99,102,132,96,90,88,84,86,90} and the time could be taken in elapsed time is 2141.1ms.
- In run 4 cluster size={78,96,110,120,146,50,100,80,94,126} and the time could be taken in elapsed time is 2140.2ms.
- In run 5 cluster size={102,96,99,100,97,115,65,126,105,95} and the time could be taken in elapsed time is 2140.3ms. (In run2 to run5 the screen shots have been added the appendix.).

The resulting clusters of the uniform distribution of K-Means algorithm is presented in different runs. Compute modified cluster centers and the cumulative distance measure between each centroid and its assigned elements. The elapsed time can be taken in milli seconds. In the performance for each run is differing. In the K-Means algorithm performed 500 random initiations of cluster centroids for each number of required clusters.

- In run 1 cluster size={87,75,107,99,145,90,94,115,98,90} and the time could be taken in elapsed time is 1543 ms.



```
Elapsed time is 0.139685 seconds.

ans =

    PC: 1
    CE: NaN
```

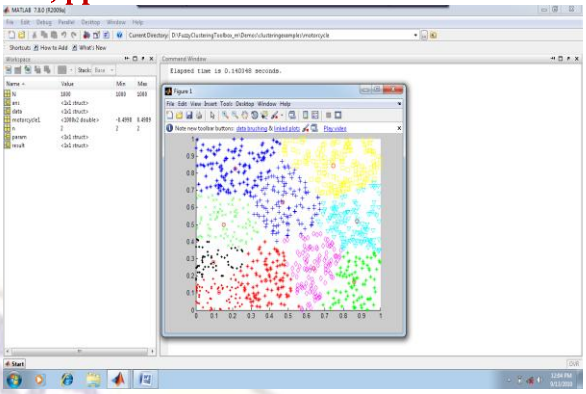
Figure 2. K-Means Clustering Algorithm for Uniform Distribution Data Points in Run1

- In run 2 cluster size={ 92,99,83,116,116,93,110,91,109,91} and the time could be taken in elapsed time is 1396 ms.
- In run 3 cluster size={ 108,88,130,74,95,115,94,98,100,98} and the time could be taken in elapsed time is 1392 ms.
- In run 4 cluster size={ 121,111,107,115,67,55,97,99,125,103} and the time could be taken in elapsed time is 1258 ms.
- In run 5 cluster size={ 86,100,149,90,139,86,77,98,94,81} and the time could be taken in elapsed time is 1402 ms. (In run2 to run5 the screen shots have been added the appendix.)

B. *Experiment B*

The K-Medoids algorithm for normal distribution data points is explained with executing results and followed by K-Medoids algorithm for uniform distribution data points. The experimental results are discussed for the K-Medoids algorithm. The normal distribution is often used to describe, at least approximately, any variable that tends to cluster around the mean. By the central limit theorem, under certain conditions the sum of a number of random variables with finite means and variances approaches a normal distribution as the number of variables increases. For this reason, the normal distribution is commonly encountered in practice, and is used throughout statistics, natural science, and social science[2] as a simple model for complex phenomena.

- In run 1 cluster size={ 96,121,95,106,103,89,110,118,78,84} and the time could be taken in elapsed time is 1781.6 ms.



```
Elapsed time is 0.140348 seconds.
```

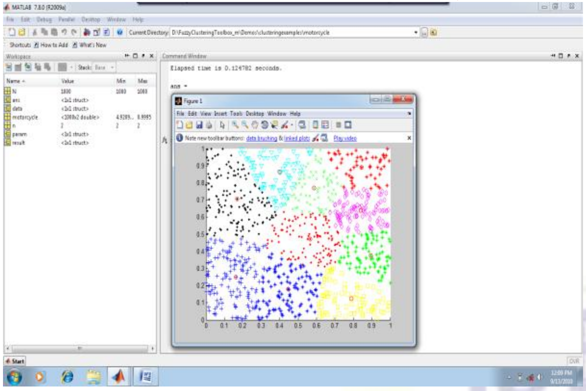
Figure 3. K-Medoid Clustering Algorithm for Normal Distribution Data Points in Run1

- In run 2 cluster size={ 120,78,97,110,146,50,100,80,93,126} and the time could be taken in elapsed time is 1291.14ms.
- In run 3 cluster size={ 96,102,99,100,97,115,65,126,105,95} and the time could be taken in elapsed time is 1151.41ms.
- In run 4 cluster size={ 88,112,95,82,120,86,119,106,102,90} and the time could be taken in elapsed time is 1154.41ms.
- In run 5 cluster size={ 121,132,98,74,82,149,118,67,81,78} and the time could be taken in elapsed time is 1134.94ms. (In run2 to run5 the screen shots have been added in the appendix.)

K-Medoids Clustering Algorithm for Uniform Distribution Data Points.

In this study, the K-Medoids algorithm for uniform distribution data points is explained with executing results .The experimental results are discussed for the K-Medoids algorithm. In uniform distribution data points could be used the probability density function. The domain value is a≤ x ≤ b. a,b is a boundary values a < b.

- In run 1 cluster size={ 164,72,91,103,101,107,77,108,91,86} and the time could be taken in elapsed time is 1249.6ms.



```
Elapsed time is 0.124782 seconds.

ans =
```

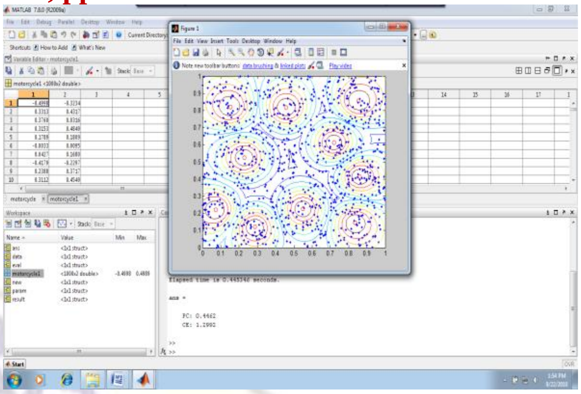
Figure 4. K-Medoid Clustering Algorithm for Uniform Distribution Data Points in Run1

- In run 2 cluster size={133,98,103,116,56,96,108,101,91,98} and the time could be taken in elapsed time is 1244.74ms.
- In run 3 cluster size={ 81,98,90,103,80,119,115,97,104,113} and the time could be taken in elapsed time is 1247.31ms.
- In run 4 cluster size={ 114,117,96,97,102,98,81,95,92,108} and the time could be taken in elapsed time is 1400.39ms.
- In run 5 cluster size={ 100,74,118,113,109,88,95,104,100,99} and the time could be taken in elapsed time is 1248.94ms. (In run2 to run5 the screen shots have been added in the appendix.)

C. *Experiment C*

Fuzzy C-Means allows to searching for soft clusters. In this study, the Fuzzy C-Means algorithm for normal distribution data points is explained with executing results and followed by Fuzzy C-Means algorithm for uniform distribution data points. The experimental results are discussed for the Fuzzy C-Means. While many algorithms have been proposed for large and very large data sets for the crisp case, not as much work has been done for the fuzzy case. As pointed out in [10], the crisp case may not be easily generalized for fuzzy clustering.

- In run 1 cluster size={ 91,107,105,78,106,123,81,115,105, 89} and the time could be taken in elapsed time is 434.480ms.



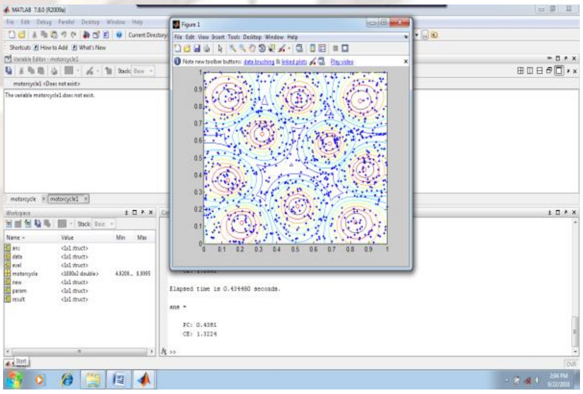
```
Elapsed time is 0.445346 seconds.

ans =

    PC: 0.4462
    CE: 1.2992

>>
>>
```

Figure 5. Fuzzy C-Means Clustering Algorithm for Normal Distribution Data Points in Run1

- In run 2 cluster size={ 93,101,126,97,84,91,92,97,95,124} and the time could be taken in elapsed time is 445.644ms.
- In run 3 cluster size={ 96,92, 123,82,125,93,96,102,94,97} and the time could be taken in elapsed time is 448.084ms.
- In run 4 cluster size={ 95,126,92,103,97,91,125,93,81,97} and the time could be taken in elapsed time is 461.180 ms.
- In run 5 cluster size={ 125,81,95,98,93,103,122,93,92,98 } and the time could be taken in elapsed time is 445.199 ms. (In run2 to run5 the screen shots have been added in the appendix.).

The Fuzzy C-Means algorithm for uniform distribution data points is explained with executing results .The experimental results are discussed for the Fuzzy C-Means algorithm.

- In run 1 cluster size={ 90,101,,87,89,103,106,127,135,98,64} and the time could be taken in elapsed time is 434.480ms.



```
Elapsed time is 0.434480 seconds.

ans =

    PC: 0.4381
    CE: 1.3224

>>
```

Figure 6. Fuzzy C-Means Clustering Algorithm for Uniform Distribution Data Points in Run1

In run 2 cluster size={ 110,86,79,102,123,108,125,84,83,100} and the time could be taken in elapsed time is 228.034ms.

In run 3 cluster size={ 65,85,103,108,89,135,127,103,88,97} and the time could be taken in elapsed time is 241.502ms.

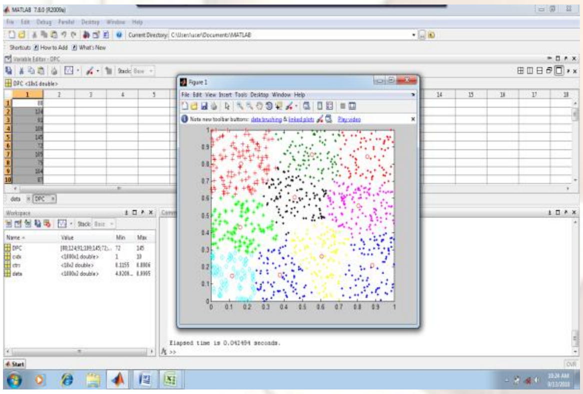
In run 4 cluster size={ 86,108,109,125,88,66,105,99,129,85} and the time could be taken in elapsed time is 245.073ms.

In run 5 cluster size={ 102,111,65,90,125,87,96,133,102,89} and the time could be taken in elapsed time is 245.073ms. (In run2 to run5 the screen shots have been added in the appendix.)

D. Experiment D

The Improved K-Means algorithm for normal distribution data points is explained with executing results and followed by Improved K-Means algorithm for uniform distribution data points. The experimental results are discussed for the Improved K-Means algorithm.

- In run 1 cluster size={ 42,77,65,148,125,104,67,95,111,166} and the time could be taken in elapsed time is 62.2ms.



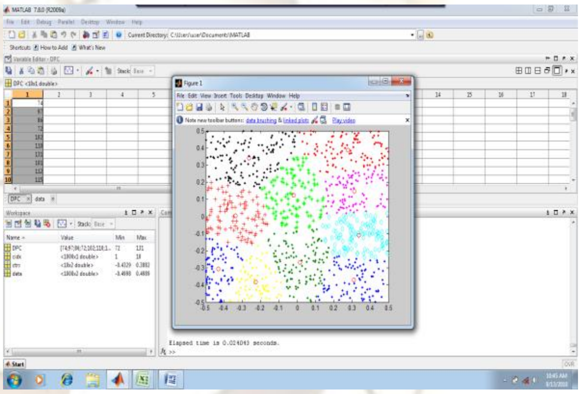
```
Elapsed time is 0.042494 seconds.
>>
```

Figure 7. Improved K-Means Clustering Algorithm for Normal Distribution Data Points in Run1

- In run 2 cluster size={ 102,112,108,74,103,121,102,97,87,94} and the time could be taken in elapsed time is 45.97ms.
- In run 3 cluster size={ 120,91,106,101,122,97,88,87,111,77 } and the time could be taken in elapsed time is 44.76ms.
- In run 4 cluster size={ 92,133,71,95,95,101,74,101,121,117} and the time could be taken in elapsed time is 34.45.
- In run 5 cluster size = {123,86,93,64,96,104,133,84,121,96 } and the time could be taken in elapsed time is 28.59 ms. (In run2 to run5 the screen shots have been added in the appendix.)

The Improved K-Means algorithm for uniform distribution data points is explained. The experimental results are discussed for the Improved K-Means algorithm. The proposed idea comes from the fact that the K-Means algorithm discovers spherical shaped cluster, whose center is the gravity center of points in that cluster, this center moves as new points are added to or removed from it. This motion makes the center closer to some points and far apart from the other points, the points that become closer to the center will stay in that cluster, so there is no need to find its distances to other cluster centers. The points far apart from the center may change the cluster, so only for these points their distances to other cluster centers will be calculated, and assigned to the nearest center.

- In run 1 cluster size={ 88,124,91,109,145,72,105,75,104,87} and the time could be taken in elapsed time is 31.27ms.



```
Elapsed time is 0.024043 seconds.
>>
```

Figure 8. Improved K-Means Clustering Algorithm for Uniform Distribution Data Points in Run1

- In run 2 cluster size={ 98,78,92,91,122,83,93,111,122,110} and the time could be taken in elapsed time is 48.97ms.
- In run 3 cluster size={104,71,115,100,89,102,104,121,110,84} and the time could be taken in elapsed time is 54.33ms.
- In run 4 cluster size={ 111,87,123,114,118,62,90,95,116,84} and the time could be taken in elapsed time is 28.69 ms.
- In run 5 cluster size={ 85,91,83,99,96,98,82,108,113,145} and the time could be taken in elapsed time is 24.59 ms. (In run2 to run5 the screen shots have been added the appendix.)

In this experimental results are taken in each algorithm for normal and uniform distribution data points in different run.

And different number of iterations gives the better results for Improved K-Means clustering algorithm.

The K-Means Clustering algorithm similarity measures are calculated by Euclidean distance. The Cluster size (in even size) and elapsed time is taken as shown in Table 1.

TABLE 1: CLUSTER RESULTS USING K-MEANS FOR NORMAL DISTRIBUTION DATA POINTS

No.of clusters	2	4	6	8	10	Time in ms
Run1	95	75	117	102	131	2124.6
Run2	101	119	90	61	126	2141.4
Run3	99	132	88	86	87	2141.1
Run4	96	120	50	80	126	2140.2
Run5	96	100	115	126	95	2140.3

The Fuzzy-C-Means Clustering algorithm similarity measures are calculated by Euclidean distance. The Fuzzy-C-Means cluster size (in even size) and elapsed time taken as shown in Table 2.

TABLE 2: CLUSTER RESULTS USING FUZZY C-MEANS FOR NORMAL DISTRIBUTION DATA POINTS

No.of clusters	2	4	6	8	10	Time in ms
Run1	107	78	123	115	89	434.480
Run2	101	97	91	97	124	445.644
Run3	92	82	93	102	97	448.084
Run4	126	103	91	93	97	461.180
Run5	81	98	103	93	98	445.199

The Improved K-Means Clustering algorithm similarity measures are calculated by Euclidean distance. The Improved K-Means Clustering algorithm use two phases, first phase is to determine the initial centroids of the clusters. Second phase is to assign each data point to the appropriate clusters.

The Improved K-Means cluster size (in even size) and elapsed time taken as shown in Table 3.

TABLE 3: CLUSTER RESULTS USING IMPROVED K-MEANS FOR NORMAL DISTRIBUTION DATA POINTS

No.of clusters	2	4	6	8	10	Time in ms
Run1	77	148	104	95	166	62.2
Run2	112	74	121	97	94	45.97
Run3	91	101	97	87	77	44.76
Run4	133	95	101	101	117	34.45
Run5	86	64	104	84	96	28.59

The K-Means Clustering algorithm similarity measures are calculated by Euclidean distance. The K-Means cluster size (in even size) and elapsed time taken as shown in Table 4.

TABLE 4: CLUSTER RESULTS USING K-MEANS FOR UNIFORM DISTRIBUTION DATA POINTS

No.of clusters	2	4	6	8	10	Time in ms
Run1	75	99	90	115	90	1543
Run2	99	116	93	91	91	1396
Run3	88	74	115	98	98	1392
Run4	111	115	55	99	103	1258
Run5	100	90	86	98	81	1402

The Fuzzy-C-Means Clustering algorithm similarity measures are calculated by Euclidean distance. The cluster size (in even size) and elapsed time taken as shown in Table 5

TABLE 5: CLUSTER RESULTS USING FUZZY C-MEANS FOR UNIFORM DISTRIBUTION DATA POINTS

No.of clusters	2	4	6	8	10	Time in ms
Run1	101	89	106	135	64	434.480
Run2	86	102	108	84	100	228.034
Run3	85	108	135	103	97	241.502
Run4	108	125	66	99	85	245.073
Run5	111	90	87	133	89	245.073

The Improved K-Means Clustering algorithm similarity measures are calculated by Euclidean distance. The Improved K-Means Clustering algorithm use two phases, first phase is to determine the initial centroids of the clusters. Second phase is to assign each data point to the appropriate clusters. The cluster size (in even size) and elapsed time taken as shown in Table 6.

TABLE 6: CLUSTER RESULTS USING IMPROVED K-MEANS FOR UNIFORM DISTRIBUTION DATA POINTS

No.of clusters	2	4	6	8	10	Time in ms
Run1	124	109	72	75	87	31.27
Run2	78	91	83	111	110	48.97
Run3	71	100	102	121	84	54.33
Run4	87	114	62	95	84	28.69
Run5	91	99	98	108	145	24.59

E. Experiment E

The elapsed time of clustering for normal distribution data points by the Improved K-Means algorithm is less than K-Medoids, Fuzzy-C-Means, K-Means algorithms. The Normal Distribution data point's elapsed time accuracy is shown in the following chart (Fig.8).

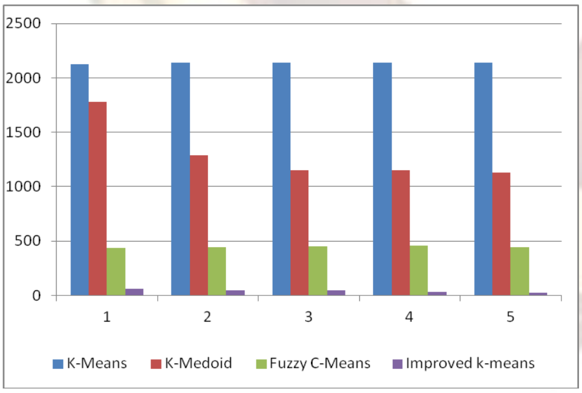


Figure 8. Time complexity for Normal Distribution Data Points

The elapsed time of clustering for Uniform distribution data points by the Improved K-Means algorithm is less than K-Medoids, Fuzzy-C-Means, K-Means algorithms. The Uniform distribution data point's elapsed time accuracy is shown in the following chart (Fig.9).

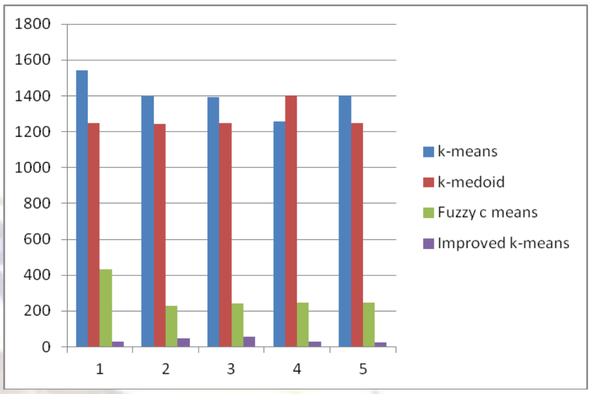


Figure 9. Time complexity for Uniform Distribution Data Points

The time complexity of uniform distribution data points is less than normal distribution data points. Fig 10 shows the time complexity for normal and uniform distribution data points.

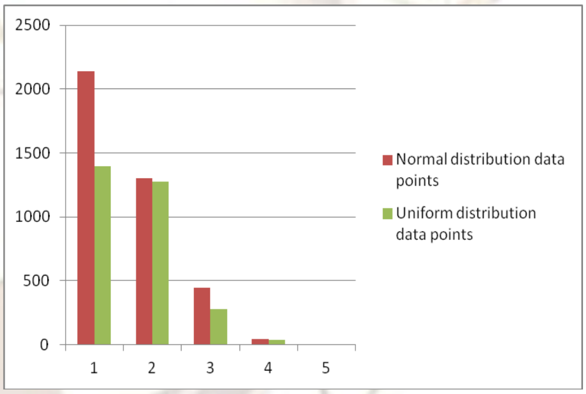


Figure 10. Time complexity for Normal and Uniform Distribution Data Points

The performance of the four algorithms for normal distribution data points is compared. The accuracy of performance could be increased in Improved K-Means algorithm. The normal distribution data point's performance accuracy is shown in the chart. Fig. 11 shows the performance measures for normal distribution data points of different clustering algorithms.

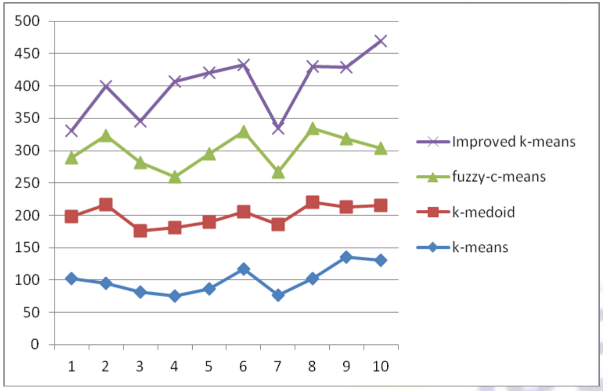


Figure 11. Performance for Normal Distribution Data Points

The performance for uniform distribution data points to compare the four algorithms. The accuracy of performance could be increased in Improved K-Means algorithm. The uniform distribution data point's performance accuracy is shown in below chart. Fig. 12 shows the performance measures for uniform distribution data points of different clustering algorithms.

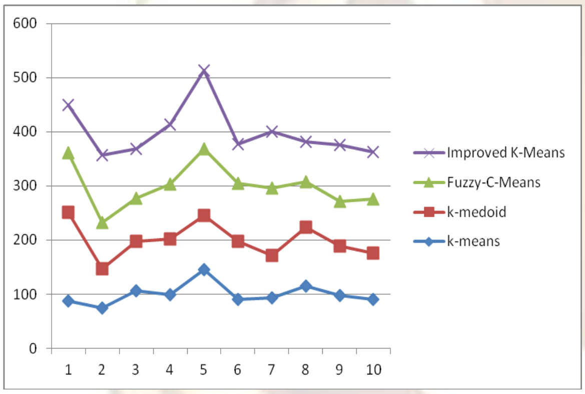


Figure 12. Performance for Uniform Distribution Data Points

VI. Conclusion

Analyzing the results of testing the clustering algorithms and running them were using normal and uniform distribution data points. As the number of clusters, K becomes greater. The performance of Improved K-Means is compared to other clustering algorithms K-Means, K-Medoids, and Fuzzy-C-Means. The K-Means algorithm is very effective for small dataset. The Accuracy of Improved K-Means is greater than others. The time complexity is reduced more than others. The Improved K-Means performance is also greater than comparing to the other clustering algorithms. In the future work, the hierarchical algorithms CURE, BIRCH will be implemented to improve the better efficiency and reduce time complexity of other partition algorithms.

References

[1] Mary De Rosa "Data Mining and Data Analysis for Counterterrorism" Conference report on H.R. 2658, Department of Defense Appropriations Act, 2004, H.R. Conf. Rep. No. 108-283 (9/24/2003), available at 108:FLD001:H08501.

[2] Hebah H. O. Nasereddin "Stream Data Mining" International Journal of Web Applications Volume 1 Number 4 December 2009.

[3] Herbert A. Edelstein "Introduction to Data Mining and knowledge Discovery", Two Crows Corporation .ISBN: 1-892095-02-5.

[4] McQueen J, "Some methods for classification and analysis of multivariate observations," Proc. 5th Berkeley Symp. Math. Statist.Prob., (1):281–297, 1967.

[5] Jesus Mena. "Investigative Data Mining for Security and Criminal Detection", First Edition, amazon.com/Investigative-Mining-Security-Criminal-Detection/dp.

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

[7] Xiong, H., J. Wu and J. Chen, 2009. "K-Means clustering versus validation measures: A data distribution perspective". IEEE Trans. Syst., Man, Cybernet. Part B, 39: 318-331.

[8] Pang-Ning Tan, Michael Steinback and Vipin Kumar, "Introduction to Data Mining", Pearson Education, 2007.

[9] David Hand, Heikki Mannila, Padhraic Smyth. "Principles of Data Mining", ISBN: 026208290 MIT Press, Cambridge,MA, 2001.

[10] Wei Li "Modified K-means clustering algorithm" Institute of Operational Research & Cybernetics Hangzhou Dianzi University 978-0-7695-3119-9/08 $25.00 © 2008 IEEE DOI 10.1109/CISP.2008.349.

[11] Mushfeq-Us-Saleheen Shameem, Raihana Ferdous "An efficient K-Means Algorithm integrated with Jaccard Distance Measure for Document Clustering" 978-1-4244-4570-7/09/$25.00 ©2009 IEEE.

[12] Jiawei Han, Micheline Kamber. "Data Mining: Concepts and Techniques", Morgan Kaufmann Publishers, Champaign:CS497JH, fall 2001.

[13] K. A. Abdul Nazeer and M. P. Sebastian, "Improving the accuracy and efficiency of the k-means clustering algorithm," in International Conference on Data Mining and Knowledge Engineering (ICDMKE),Proceedings of the World Congress on Engineering (WCE-2009),Vol 1, July 2009, London, UK.

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