International Journal of
Electronic Commerce Studies
Vol.1, No.2 , pp.61-76, 2010

APPLICATION OF C-MEANS AND MC-MEANS CLUSTERING ALGORITHMS TO SOYBEAN DATASET

Faraj A. El-Mouadib,
University of Garyounis,
Faculty of Information Technology,
Dept. of Computer Science,
Benghazi, Libya
elmouadib@yahoo.com

Halima S. Talhi,
Great Man Made River Project
Benghazi, Libya
h.talhi@yahoo.com

ABSTRACT

At the present time, massive amounts of data are being collected. The availability of such data gives rise to the urgent need to transform the data into knowledge; this is the function of the field of Knowledge Discovery in Database (KDD). The most essential step in KDD is the Data Mining (DM) step which is the search engine to find the knowledge embedded in the data. The tasks of DM can be classified into two types, namely: predictive or descriptive, according to the sought functionality.

One of the older and well-studied functionalities in data mining is cluster analysis (Clustering). Clustering methods can be either hierarchal or partitioning. One of the very well known clustering algorithms is the C-means.

In this paper, we turn our focus on cluster analysis in general and on the C-means partitioning method in particular. We direct our attention to the modification of the C-means algorithm in the way it calculates the means of the clusters. We consider the mean of a cluster to be one of the objects instead of being an imaginary point in the cluster. Our modified C-means (MC-means) algorithm is implemented in a system developed in the visual basic.net programming language. The well-known Soybean dataset is used in an experiment to evaluate our modification to the

C-means algorithm. This paper is concluded with an analysis and discussion of the experiments' result on the bases of several criteria.

Keywords: C-means, Cluster analysis, Data Mining (DM), Knowledge Discovery in Database (KDD), MC-means.

1. INTRODUCTION

Recent advances in data acquisition tools and techniques have resulted in the generation of terabytes or more of data. These massive amounts of data are being stored in some kind of repository system (i.e. such as Databases or Data warehouses). These huge volumes of data exceed the capabilities of traditional data analysis tools to transform the data into useful knowledge.

Such huge volumes of data, call for the invention of some intelligent new data analysis tools and techniques to discover the embedded knowledge. These tools and techniques are being implemented in a newly emerging field known as Knowledge Discovery in Databases (KDD). KDD is also known by other names such as: knowledge mining from databases, data mining, knowledge extraction, data/pattern analysis, data archaeology, and data dredging "Han and Kamber [4]". Many people treat the term Data Mining (DM) as a synonym to KDD while others view DM as one step in the KDD process.

DM is the extraction of useful interesting knowledge (in the form of patterns or regularities) from large data sets, databases or other repository systems "Hand, Mannila and Smyth [5]".

According to "Han and Kamber [4]", the KDD process consists of:

1. Selection of the relevant data for the current mining task.
2. Pre-processing in order to clean the data from noise or outliers or irrelevant data.
3. The application of intelligent methods in order to extract patterns or regularities from the data.
4. A pattern evaluation step which is to identify the truly interesting patterns/ regularities by the use of some interestingness measures.
5. Knowledge presentation of the discovered/ newly found knowledge to the user.

KDD is defined as the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data "Fayyad, Piatetsky-Shapiro and Smyth [2]" and "Mitra and Acharya [10]". DM is a confluence of many disciplines such as: statistics, machine learning, database systems, data warehousing, and others. Recently, KDD and DM have become one of the most active research areas that have attracted the attention of many researchers. Many successful applications have been reported from different sectors such as security marketing, medicine, manufacturing, multimedia, education, finance, banking, telecommunication, etc…"Mitra and Acharya [10]".

2. DATA MINING TASKS

DM mining tasks are classified as predictive or descriptive. Predictive mining tasks are processes used to produce a model that describes the current data set and then to use the model to make prediction for new data objects. Classification and Prediction are some examples of predictive mining tasks. Descriptive tasks are processes that characterize the general properties of the current data set. Characterization, Discrimination, Cluster Analysis and Association Analysis are examples of descriptive mining tasks.

3. CLUSTER ANALYSIS

Cluster analysis is the process of finding groups or clusters in the data, so that objects contained within one group are as similar to each other as possible, and objects in different clusters are as dissimilar as possible from those in this particular group. Cluster analysis has been in use since the 1940s in many fields such as: Biology, Chemistry, Botany and others. In the past, clustering was carried in a subjective manner/style and researchers relied on their perception and judgment in interpreting the results.

Nowadays, old clustering techniques are unsuitable due to the vast increase of magnitude/amount of the data to be clustered. Automatic classification of data is a new scientific discipline being vigorously developed, "Kaufman and Rousseeum [6]". Since the mid 80's clustering has been established as an independent scientific discipline. Some scientific periodicals such as the Journal of Classification and the International Federation of Classification Societies are dedicated to this particular discipline. Computer scientists consider cluster analysis as a branch of pattern recognition and artificial intelligence.

3.1 Clustering methods

Generally, most of the clustering methods can be classified as partitioning methods or hierarchical methods.

3.1.1 The Partitioning clustering method

Partitioning clustering methods constructs K mutual-exclusive clusters or groups out of n data objects. The partitioning method must satisfy the following constraints:

1. Each cluster must contain at least one object.
2. Each object must belong to only one cluster.

The first constraint implies that there may be as many (but no more) clusters as there are objects: $K \leq n$. The second constraint states that no object can belong to two or more clusters at the same time.

The created clusters or groups of objects must be such that objects within a cluster have high similarity (Intra-cluster) and objects in different clusters have high dissimilarity (Inter-cluster) as depicted in figure 1.

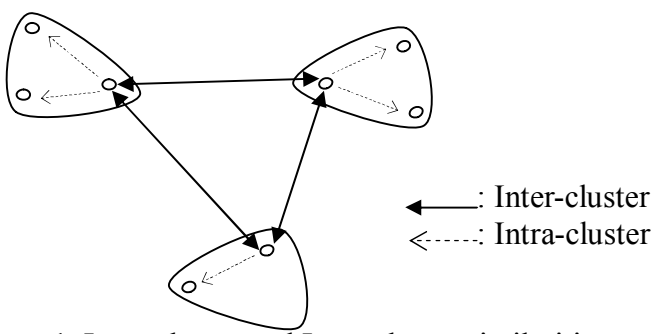

Figure 1. Intra-cluster and Inter-cluster similarities.

The partitioning algorithms start by picking up number of representative objects (depending on the value of *K*) as seeds of clusters and then assigns each data object to its nearest (fit) cluster.

3.1.2 Hierarchical clustering methods

Hierarchical methods are different from partitioning methods in the sense that all of the values of K (number of clusters) are present in the constructed dendrogram. Starting from the cluster of $K = 1$ (all objects are together in one cluster) to n (singleton clusters) clusters and in between there are all values of $K = 2, 3, ..., n - 1$. In hierarchical clustering, a tree-like hierarchy of data objects is constructed. There are two ways

(techniques) to construct the hierarchical tree, namely, agglomerative and divisive as depicted in figure 2. These two techniques construct the hierarchical representation in opposite directions.

3.1.2.1 The agglomerative technique

Starting with n clusters where each of them is singleton. At each step of the formation process two of the clusters (the closest) are combined into one cluster. This process of merging continues until there is only one cluster with n objects.

3.1.2.2 The divisive technique

The divisive clustering technique starts with a single cluster with n data objects, and in each subsequent step of the process one of the clusters will be split up into two clusters. This process continues until there are n singleton clusters.

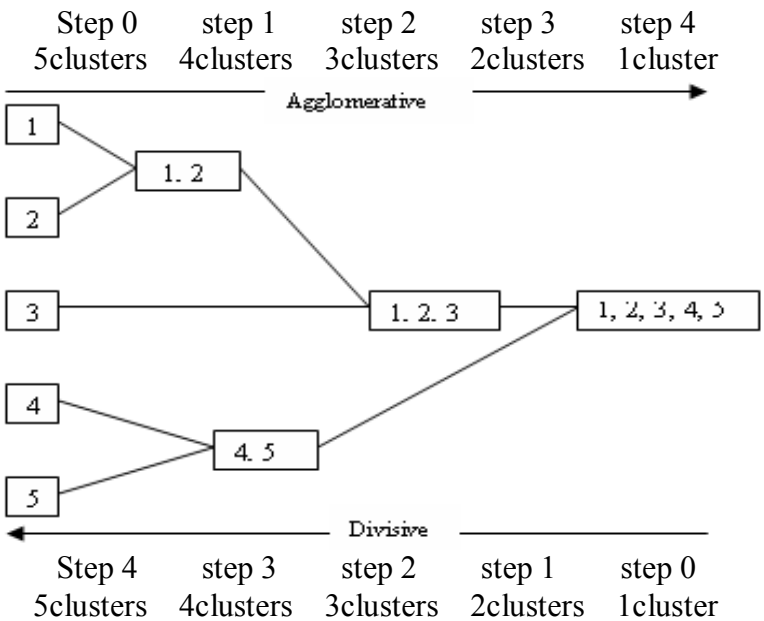

Figure 2. Agglomerative and divisive techniques.

4. K-MEANS METHODS

The K-means clustering methods are of the relocation partitioning type. These methods of clustering depend on the calculation of the means of the clusters in the process of assigning objects to clusters. The K-means

methods can be further divided into Crisp (C-mean) or hard or Fuzzy (called Fuzzy C-means) or soft.

The C-means clustering algorithm is widely used for finding clusters in data—where each object belongs to one and only one cluster. According to "Kogan [7], Larose [8] and Miyamoto, Ichihashi and Honda [11]", the standard C-means clustering algorithm was first proposed by MacQueen [15]. The steps of the standard C-means clustering algorithm (as depicted in figure 3) are as follows:

Step 1. The user must provide the K value that represents the number of clusters sought.

Step 2. K objects are chosen randomly as seeds (representative) of the K clusters.

Step 3. Each of the remaining objects in the data set is assigned to the nearest seed. This process is repeated until all of the objects are assigned to clusters. In this way we will end up with K clusters; C_1, C_2, …, C_K. This called the initial clustering.

Step 4. The mean of each cluster is calculated and the location of each of the objects is updated in such a way that each object must be located in the cluster where it occupies the minimum distance to the cluster mean.

Step 5. Repeat step 4 until some termination condition is met.

The concept of nearest is based on the Euclidean distance:

$$d(i,j)=\sqrt{(x_{i1}-x_{j1})^2+(x_{i2}-x_{j2})^2+\ldots+(x_{ip}-x_{jp})^2}$$

The standard C-means clustering algorithm steps are as the follows:

Input: A data set, K: number of clusters, n: number of data objects, $K \leq n$. Output: Clusters

```
While (true) do:      [Outer loop].
  if   y < 2 then   (y is number of iteration)      [Check].
    Randomize ( )      [Initialize].
    for I←1 to k      [Loop].
      Rnd ( )  * Val (n-1)    (choose seeds).
    End for
    Clustering ( )    (create initial clustering).
  Else
    Means ( )      [Calculate the means].
  End if
  Clustering ( )    (relocate objects)        [Create the clusters].
  if BCV/WCV (true) then      [Test clustering quality].
    Exit while    (find best clustering in this data set).
  Else
  [Repeat].
    Set y← y+1      [Iterations counter].
  End if
End while
```

Figure 3. Standard C-means algorithm.

5. MODIFIED C-MEANS ALGORITHM

After the creation of the initial clustering by the standard C-means algorithm, the mean of each cluster is calculated and depending on the outcome some of the objects may subsequently be reallocated to other clusters. It could happen that the mean of a given cluster can be an object or it is more likely to be an imaginary point in the cluster. Our modification to the C-means algorithm is based on the decision about the mean of the cluster. After the calculation of the means of the clusters as in the standard C-means algorithm, the mean of the cluster is the closest data object to the coordinate of the calculated mean (Medoid). So, the mean of a cluster will always be an actual object. In this case, we assume that there will be substantial reduction in the number of calculations and time required in subsequent iterations of the algorithm. This alteration or modification is done by adding one new step (designated by *) to the standard C-means algorithm clustering algorithm. The steps of the MC-means algorithm are depicted in figure 4.

```
While (true) do:      [Outer loop].
  if   y < 2 then   (y is number of iteration)      [Check].
    Randomize ( )      [Initialize].
    For I←1 to k      [Loop].
      Rnd ( ) * Val (n-1)    (choose seeds).
    End for
    Clustering ( )    (create initial clustering).
  Else
    Means ( )      [Calculate the means].
  End if
  Clustering ( ) (relocate objects)       [Create the clusters].
*  Medoids ( )   (the closest object to the calculated mean)      [Find the medoid].
  Clustering ( ) (considers an object to be the mean) [Create the clusters].
  if BCV/WCV (true) then      [Test clustering quality].
    Exit while    (find best clustering in this data set).
  Else      [Repeat].
    Set y← y+1      [Iterations counter].
  End if
End while
```

Figure 4. MC-means algorithm.

6. DESIGN AND IMPLEMENTATION

Here, we demonstrate the design and implementation of our clustering system named, Clustering Software System (CSS), which consists of two sub systems—one for standard C-means and the other for Modified C-means (MC-means). The execution of any of the sub systems is independent. UML "Fowler and Scott [3]", "Loton, McNeish, Schoellmann, Slater and Wu [9]", Pender [12]" and "Weilkiens [13]", is used in the analysis and design phases, and Visual Basic.NET 2005 is used for the programming phase.

Here, we demonstrate only the activity diagram, class diagram and the sequence diagram, of our system due to their importance and their ability to give a clear view of the system. Figure 5 depicts the activity diagram for the MC-means subsystem, which is the implementation of the MC-means algorithm. The activity diagram shows the different processes of the system, define the sequence of tasks, the needed conditions, and demonstrates the concurrency of the system.

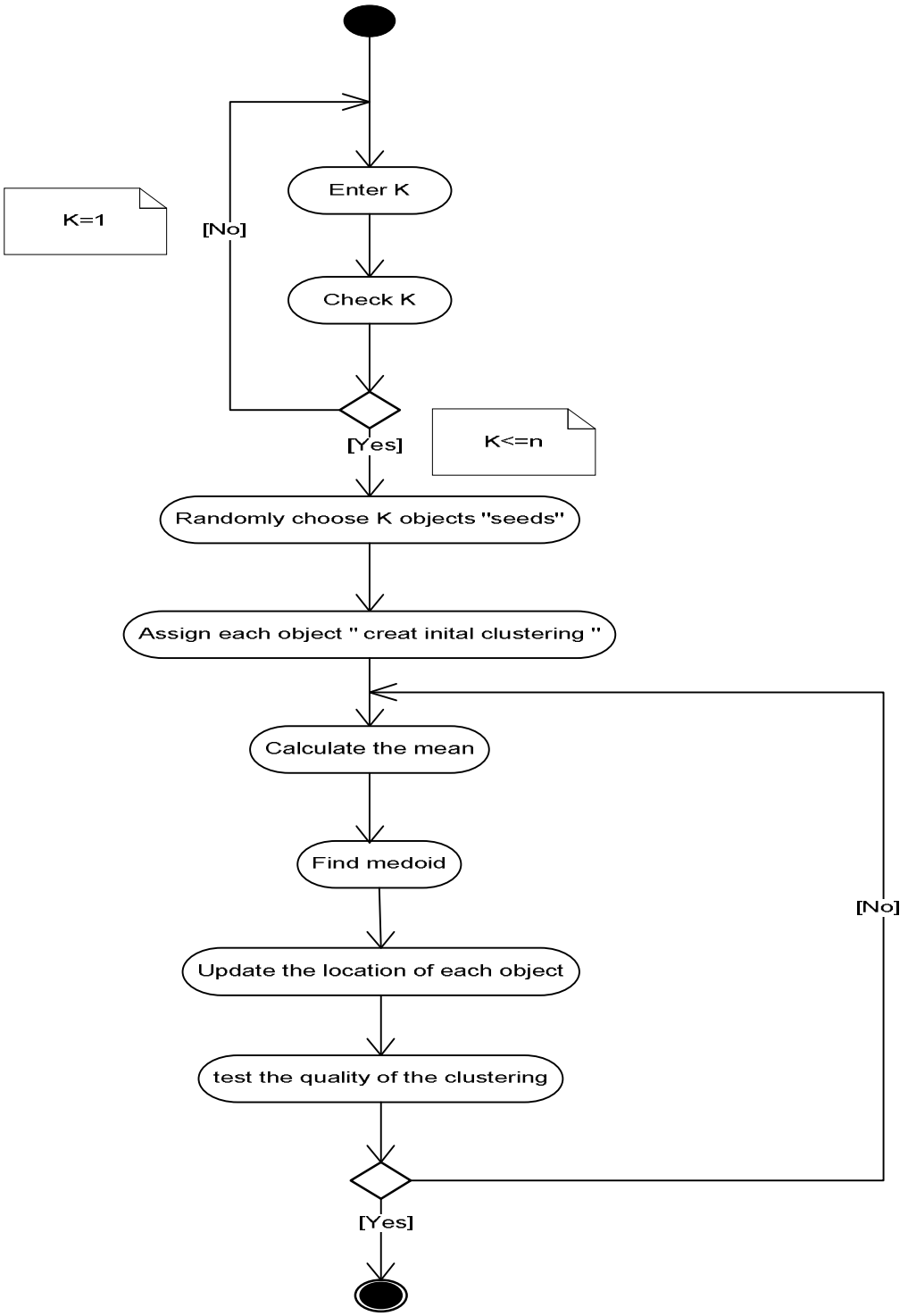

Figure 5. Activity diagram for the MC-means sub-system.

The purpose of the class diagram is to describe the systems' attributes and operations of each of the classes, and also of the different types of static relationships among those classes. Figure 6 depicts the class diagram of our system. The New (constructor) method used to initialize the objects' parameters when it is created is common to all classes. The CCS Class is the starting point of the system. The purpose of the Main window class is to start the execution of the desired sub system. The class Data is made use of to establish database connection, read the data, display it and execute SQL commands. Both of the C-means and the MC-means Classes employ the same methods but they perform different tasks depending on the sub-system chosen to be explored.

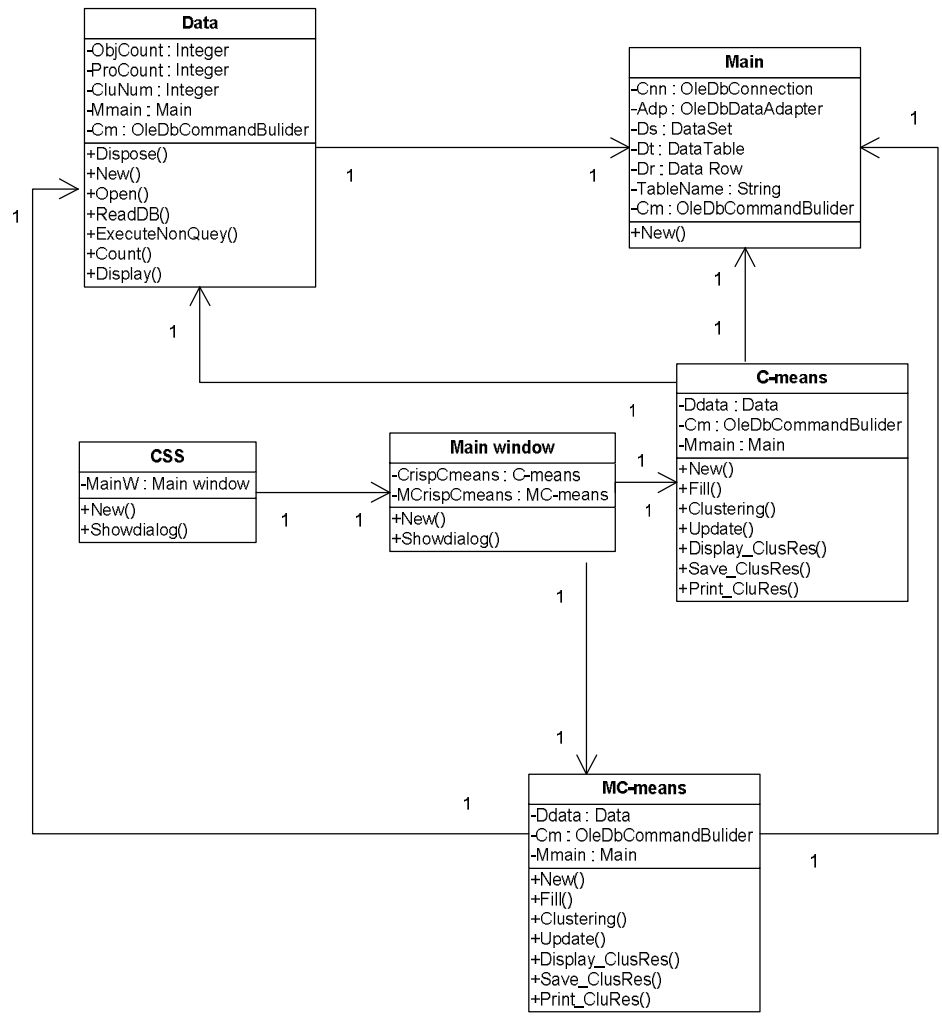

Figure 6. The class diagram of the MC-means subsystem.

The purpose of the sequence diagram is to describe the behaviour of the system and displays the messages passed between the objects during the period of the system execution. Figure 7 depicts the sequence diagram for the MCCS subsystem.

7. EXPERIMENTS AND RESULTS

All the work is performed on a Laptop computer with 1.73GHz Pentium IV processor, 1GB of RAM memory, 100MB Hard disk and Microsoft Windows XP Home Edition operating system.

Here, we demonstrate the results obtained from applying our system to a very well-known data set, the small Soybean database "Asuncion and Newman [1]". The data set consists of 47 instances, 35 categorical input attributes and four possible output classes that represent diseases in the soybean plant. This data set was donated/made available by "Michalski and Chilausky [14]" of the University of Illinois, Urbana, IL.

The experiment was run fifteen times to get the best possible results. The results of the fifteen runs for the C-means sub system are depicted in table-1. In addition, the results of the fifteen runs for the MC-means sub system are depicted in table-2.

The results obtained, as depicted graphically in Figure 8 and Figure 9, show an increase in the performance of the MC-means algorithm over the original C-means algorithm.

As far as the run time is concerned, on the average the MC-means algorithm was faster than the C-means algorithm by 6.69%.

On the average the MC-means algorithm was better than the C-means algorithm since it required 37.68% fewer passes.

Moreover in the 15 runs of each sub system and on the average, the MC-means algorithm had 14.61% of misclassified cases as compared to the 27.94% of the C-means algorithm. The MC-Means in fact saved considerable time in clustering huge data sets.

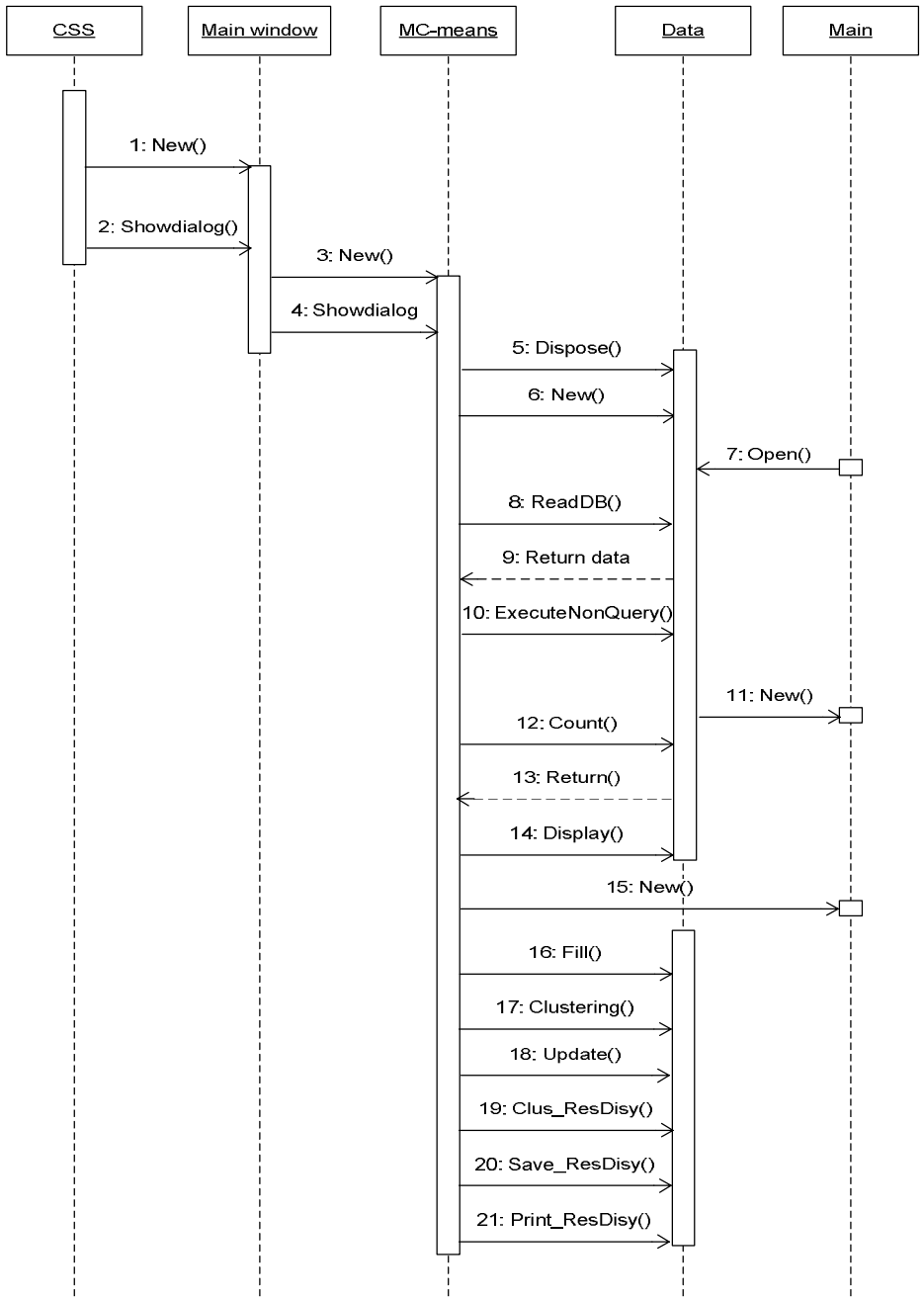

Figure 7. The sequence diagram for the MC-means sub-system.

Table 1. Fifteen runs for the C-means sub system.

Test run	Number of passes	Time (ns)	Number of misclassified objects
1	5	13193	15
2	6	15532	15
3	6	15124	13
4	5	12563	13
5	3	10326	14
6	6	15181	15
7	5	14901	13
8	3	12675	15
9	3	9084	11
10	3	9916	5
11	5	15819	11
12	3	9111	12
13	3	7549	16
14	4	10820	15
15	3	9849	14

Table 2. Fifteen runs for the MC-means sub system.

Test run	Number of passes	Time (ns)	Number of misclassified objects
1	2	6910	8
2	2	8080	7
3	3	9479	9
4	2	7634	5
5	2	7498	6
6	2	8903	5
7	3	10988	5
8	2	9918	8
9	3	10316	5
10	3	10316	5
11	2	8025	8
12	3	12647	7
13	2	8136	9
14	3	11815	7
15	2	9941	9

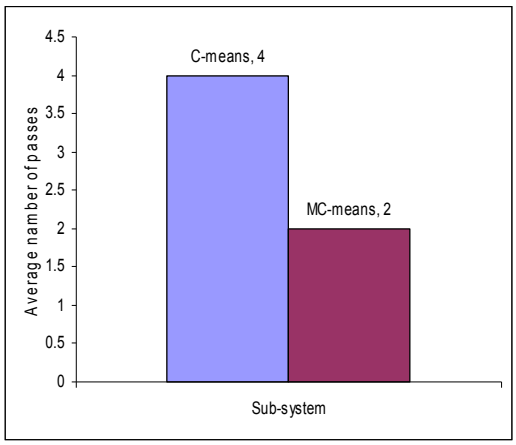

Figure 8. Average number of passes of sub-systems.

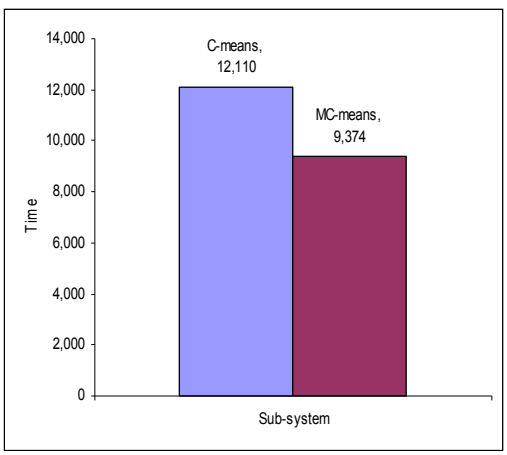

Figure 9. Time of sub-systems in nanoseconds.

8. CONCLUSION

The objectives of this research were met in implementing the C-means and a modified version of it MC-means algorithms. The implementation of the system was carried out in Visual basic.net programming language. The system was tested with the Soybean dataset on the criteria of run time, number of passes and misclassified objects.

The obtained results from this experiment show that the MC-means sub-system had outperformed the C-means sub-system on the comparison criteria.

From the experience gained while carrying out this work, the authors would like to make the following recommendations for further research:

1. Because of the experimental nature of cluster analysis, more experiments are needed with different data sizes.

2. To find more criteria on which to base the comparison in order to verify/confirm our findings.

9. REFERENCES

[1] Asuncion, A. and Newman, D. J.. UCI Machine Learning Repository [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, School of Information and Computer Science, 2007.

[2] Fayyad, U., Piatetsky-shapiro, G. and Smyth, P., , From Data Mining to Knowledge Discovery in data base, Association for Artificial Intelligence, American, p1 – 54, 1996.

[3] Fowler, M. and Scott, K., UML Distilled A Brief Guide to the Standard Object Modeling Language, Addison Wesley, USA, p1– 339, 2000.

[4] Han, J. and Kamber, M., Data Mining: Concepts and Techniques, Morgan Kaufmann publishers, Canada, p6 – 21, 2000.

[5] Hand, D., Mannila, H., and Smyth P., Principles of Data Mining, Massachusetts Institute of Technology Press, Cambridge, Massachusetts London England, p2 – 21, 2001.

[6] Kaufman, L. and Rousseeum, P., Finding Groups in Data, John Wiley & Sons, Inc, United States of America, p1 – 66, 1990.

[7] Kogan J., Introduction to Clustering Large and High-Dimensional Data, Cambridge University Press, United States of America, p9 – 37, 2007.

[8] Larose D., Discovering Knowledge in Data, John Wiley & Sons, Inc, New Jersey,p2 – 23, 2005.

[9] Loton, T., McNeish, K., Schoellmann, B., Slater, J. and Wu, Chaur., Professional UML with Visual Studio .NET Unmasking Visio for Enterprise Architects, Wrox Press Ltd, United Kingdom, p1– 343, 2002.

[10] Mitra, S. and Acharya, T., Data Mining Multimedia, John Wiley & Sons, Inc, New Jersey, p1 – 28, 2003.

[11] Miyamoto, S., Ichihashi, H. and Honda, K., Algorithms for Fuzzy Clustering Methods, Springer, Verlag Berlin Heidelberg, p16 – 39, 2008.

[12] Pender T., UML Weekend Crash Course, Wiley Publishing Inc, Indianapolis, Indiana, p1– 224, 2002.
[13] Weilkiens T., Systems engineering with SysML/UML: modeling, analysis, design, Morgan Kaufmann Publishers, United States of America, p1– 320, 2007.
[14] Michalski, R. S. and Chilausky, R. L., "Learning by Being Told and Learning from Examples: An Experimental Comparison of the Two Methods of Knowledge Acquisition in the Context of Developing an Expert System for Soybean Disease Diagnosis," *International Journal of Policy Analysis and Information Systems*, 4(2), p125-161, 1980.
[15] MacQueen, J, *Some methods for classification and analysis of multivariate observations*, Proceedings of Fifth Berkeley Symposium on Mathematical Statistics and Probability, p281- 297, 1967.