ISSN No. 0976-5697

Volume 5, No. 5, May-June 2014

International Journal of Advanced Research in Computer Science

REVIEW ARTILCE

Available Online at www.ijarcs.info

Comparative Study Between Primitive Operation Complexity Against Running Time Application On Clustering Algorithm

Tb. Ai Munandar
Universitas Serang Raya
Informatics Eng. Department
Serang – Banten – Indonesia

Aina Musdholifah
Universitas Gadjah Mada
Faculty of Math. & Natural Sciences
Yogyakarta - Indonesia

Abstract: Time complexity of an algorithm is a standard test to obtain the execution time-efficient when implemented in a programming language. Asymptotic analysis approach uses the concept of the Big-O is one of the techniques commonly used to test the time complexity of an algorithm. This study will conduct a comparison test between the three clustering algorithms using time complexity analysis of primitive operations with the running time of applications when the algorithm is used in a programming language or an application. K-means clustering algorithm, Fuzzy C-Means (FCM) and the Hierarchy Agglomerative Clustering (HAC) will be compared based on the analysis of primitive operations and their implementation using MATLAB applications. The results showed that, HAC algorithm has running time that is much more stable than the K-means, although based on the analysis of Big-O, both have the same time complexity. So also between HAC and FCM, HAC is much more stable than the FCM algorithm for all testing using different data sets.

Keywords: time complexity, clustering algorithm, K-means, Fuzzy C-means, Hierarchy Agglomerative Clustering

I. INTRODUCTION

Clustering is a technique of grouping data based on similarity of characteristics that form a particular data clusters correspond to the proximity of the data from each other [1]. Various algorithms developed by scientists for data classification requirements, some of which are partitioning and hierarchical clustering..

The presence of various clustering algorithm, attracted the attention of scientists to conduct testing of these algorithms. Call it, for example, the emergence of K-means algorithm that is believed to have the ability to cluster computing faster for the case of a lot of data, but also has the disadvantage of having to begin by determining the center of the cluster before clustering is done [2] giving rise to a new variant of the algorithm that combines the concept of logic fuzzy K-means, and known as Fuzzy c-means developed by James C. Bezdek in 1981 [3]. So is the hierarchy agglomerative clustering algorithm, which had been developed in line with the tests conducted on the algorithm. Call it like Enhance Hierarchy Agglomerative Clustering variant that has a better performance than the previous algorithms [4], a variant Bidirectional HAC [5] and many more, such as incorporating the concept of HAC with CUDA programming [6]. Some of the tests performed, more focused on how the algorithm handles the data, the performance in handling the data and generate a data group, versatility and popularity algorithms are used [7], [2], yet many are led to the testing of algorithmic complexity, both in the calculation of primitive operation and its implementation in the programming language, or the study of the correlation between the two.

The complexity here is intended to test and find out the time, memory and other resources needed an algorithm to solve computational problems [8] through several approaches, one of which is the asymptotic analysis using asymptotic notation or Big-O notation in order to analyze the time complexity of an algorithm. Scientists use complexity theory to examine the value of the cardinality of an algorithm to achieve the highest score, especially on evolutionary algorithm [9], [10], testing the computational bottleneck Nelder-Mead search algorithm with a single iteration [11], testing the performance of Estimation of Distribution Algorithm (EDA) based on its time complexity [12], optimization of the search order of the symbols of the determinant decision diagram (DDD) uses a binary decision diagrams [13], the data sorting algorithms using the technique of quick, heap, insertion and merge [14] and even for testing complexity of a system that has a different system in it [15].

This study discusses the complexity of testing three clustering algorithms using computation primitive operation then compared the results with the implementation of the algorithm into a program. Complexity is referred to in this discussion only to the complexity of running time for all four algorithms will be compared, namely K-means, Fuzzy c-means and hierarchical agglomerative clustering.

The discussion is divided into five parts, the first part is an introduction that describes the background of the problem, and review some of the research related to the research conducted. The second part describes a literature review of the four algorithms, namely K-means, Fuzzy c-means and Hierarchical Agglomerative Clustering. Part three is the methodology used in this study and the fourth section is a discussion of research and discussion, and the last section concludes with a conclusion.

II. LITERATURE REVIEW

A. K-Means Clustering

K-means clustering algorithm is a clustering technique that represents groups of data based on the distance between the object with its cluster mean. The process runs for function grouping criteria have not been found and continue to shape the new mean values for each cluster, a cluster is formed until the desired end. Some implementations include K-mean clustering is widely used for the needs of the crops [16], the identification of the structure of the data [17] and image segmentation in the health sector [18]. Here are the steps for the completion of the K-means algorithm [19] (see pseudo code in Figure 1):

1. Determine the number of clusters K and number of input datasets n follow the pattern of $X = \{x_1, \ldots , x_n\}$. Select initial centers cluster k, $V^{(0)}$ of a random dataset.
2. Determine the shortest distance between the clusters closest to the dataset. For each data x_i, count membership of $m(C_j \mid x_i)$ for each cluster of C_j. Membership function of $m(C_j \mid x_i)$ shows that x_i part of cluster C_j. K-means algorithm using hard membership function where $m(C_j \mid x_i) \in \{0,1\}$. If dataset x_i close to cluster C_j (i.e, the distance of x_i to cluster C_j is the minimal distance), then $m(C_j \mid x_i)=1$, otherwise $m(C_j \mid x_i)=0$.
3. Recalculate the cluster cancroids point k to search for a new cluster center v_j using (1) and calculate the value of square error E using (2) as follows :

$$v_j = \frac{\sum_{i=1}^{n} m(C_j|x_i)x_i}{\sum_{i=1}^{n} m(C_j|x_i)} \quad untuk\, j = 1 \ldots k \qquad (1)$$

$$E = \sum_{j=1}^{k} \sum_{x_i \in C_j} ||x_i - v_j||^2 untuk\, i = 1 \ldots n; j = 1 \ldots k \qquad (2)$$

4. Repeat steps 2 and 3 until the clustering found. Iterations stop if the dataset is no longer form a new cluster, change the value of square error E below its threshold value or a predetermined number of iterations has been reached.

```
K-means Algorithm:
    Input : X = {x_1, ... , x_N} ∈ R^D (N x D input data set)
    Output : C = {c_1, ... , c_J} ∈ R^D (J cluster centers)

Select a random subset C of X as the initial set of Cluster center;
While termination criterion is not met do
    For (i=1;i≤N; i=i+1) do
        Assign x_i to the nearest cluster;
        m[i] = argmin ||x_i – c_j ||^2 (where j ∈ {1,.. ,J};
    End
    Recalculate the clusteer centers;
    For (k=1; k ≤J; k = k+1) do
        Cluste S_k contains the set of points x_i that are nearest to the center c_k;
        S_k = {x_i | m[i] = k };
        Calculate the new center ck as the mean of the points that belong to S_k;
        c_k = 1/|S_k| * (Σx_i | x_i ∈ S_k)
    End
End
```

Figure 1. K-means pseudocode

B. *Fuzzy C-Means*

Fuzzy C-means is an unsupervised clustering algorithm and has been widely used for a variety of needs analysis in various fields, such as agriculture, astronomy, chemistry, geology, health diagnostics and so forth [20], image data analysis [21]. Here are the steps to completion of FCM [22] (see pseudo code FCM in Figure 2) :

1. Input data to be clustered X, a matrix of size $n \times m$ (n = number of sample data, m = attribute of each data), where X_{ij} are i^{-th} dataset (i=1,2,3,..,n), and j^{-th} attribute (j=1,2,..,m).
2. ext, specify the initial values of such calculation, the number of clusters, rank, maximum iterations, the smallest expected error (ξ), initial objective function and iteration.
3. Generate random values in the matrix elements form the initial partition U (μ_{ik}, i=1,2,..,n; k=1,2,..,c).
4. Calculate the k^{-th} cluster center; V_{kj}, where k=1,2,...,c; and j=1,2,...,m, using (3) as follows :

$$V_{kj} = \frac{\sum_{i=1}^{n}((\mu_{ik})^w * X_{ij})}{\sum_{i=1}^{n}(\mu_{ik})^w} \qquad (3)$$

5. Next, calculate the value of the objective function at iteration t (P_t), using (4) as follows :

$$P_t = \sum_{i=1}^{n}\sum_{k=1}^{c}\left(\left[\sum_{j=1}^{m}(X_{ij} * V_{kj})^2\right](\mu_{ik})^w\right) \qquad (4)$$

6. Calculate the change in the partition matrix U, using (5) as follows :

$$\mu_{ik} = \frac{\left[\sum_{j=1}^{m}(X_{ij} - V_{kj})^2\right]^{\frac{-1}{w-1}}}{\sum_{k=1}^{c}\left[\sum_{j=1}^{m}(X_{ij} - V_{kj})^2\right]^{\frac{-1}{w-1}}} \qquad (5)$$

Where i=1,2,...,n; and k=1,2,...,c;

7. The final step is to check the condition of stops, with the following conditions:
   a. If $(|P_t - P_{t-1}| < \xi$ or (t > max iter) then stop
   b. Otherwise, then $t=t+1$ and repeat step 4.

```
FCM Algorithm :
Input : θ (k,l), N
Output U*_FCM, V*_FCM
Initialize Partition U^(0) randomly
for i = 1 to n
    for k = 1 to c
        Repeat for j = 1, 2, 3, ...
            Update centroid V^(0) with U^(j-1) Using (3)
            Compute Distance D^(j) with V^(j)
            Update Partition Matrix U^(j) with D^(j) using (5)
        Until ||U^(j) – U^(j-1)|| < ∈
    end
end
Return U*_FCM ← U^(j) and V*_FCM ← V^(j)
```

Figure 2. Fuzzy C-means pseudocode

C. *Hierarchy Aglomerative Clustering*

Hierarchy Agglomerative Clustering (HAC) is a technique of grouping the data into the category hierarchy, in which the process of the formation of groups of data are performed sequentially forming a nested hierarchy from the bottom up. For n samples, the algorithm begins by forming n clusters where each cluster contains a single sample or a point. Then the two clusters will be joined so that the similarity between the two is getting close to the number of clusters that are formed into a 1 or according to a predetermined. The following are the steps to resolve the clustering using HAC [23] (see HAC pseudo code in Figure 3):

1. Start with n clusters and a single sample that indicates a cluster..
2. Find clusters Ci and Cj that have closest similarity, then combine the two into a cluster.
3. Repeat step 2 until the number of clusters into a single cluster, or as desired.

There are several ways that can be used to find the distance between each pair making it possible for the merged cluster, including the use of methods of Single Linkage, Complete Linkage and Average Linkage. Here is pseudocode for the algorithm of HAC [24] :

```
Hierarchy Agglomerative Clustering:

1. $t = 0$
2. choose $R_0 = [C_i = x_i, i = 1, ... N]$ as initial clustering
3. repeat
4.     $t = t + 1$
5.     Find the closest cluster $C_i,C_j$ in the existing cluster
       $R_{t-1}$ such that
6.         $g(C_i,C_j) = max_{r,s}(C_r,C_s)$ if g is similarity function
7.         $g(C_i,C_j) = minr,s(Cr,Cs)$ if g is dissimilarity function
8.     Define $C_q = C_i \cup C_j$ and produce the new
       clustering $R_t = [R_{t-1} – C_i – C_j] \cup C_q$
9. Until only one cluster is left
```

Figure 3. HAC pseudocode

III. RESEARCH METHODOLOGY

This study begins with the collection of literature related to the algorithm that will be compared and primitive operation analysis using Big-O for all three algorithms, namely K-means, FCM and HAC. The next step is to analyze the time complexity using the approach of the Big-O to see the time owned the complexity of each algorithm. After that, every agoritma then tested on MATLAB applications using a number of test data sets to see the running time required of each algorithm. Tests performed five times by dividing the data set into several parts, and each test performed three times for a total measurement tests performed 15 times. Results of testing running time, then analyzed to obtain information relationships between the Big-O analysis with running time of each algorithm.

IV. RESULT AND DISCUSSING

A. Primitive Operation Analysis

This section discusses how each algorithm's time complexity is calculated using the asymptotic approach to analysis using Big-O notation. In this study, we tried to analyze the classical K-means clustering algorithm, Fuzzy C-Means and Agglomerative Clustering Hierarchy (HAC) based pseudo code structure obtained from the literature according to the programming logic when the algorithm will be made into a programming language. Pseudo code that appear in the image 1-3, is seen as closer to the real implementation of the program will be made in the form of any programming language. It is also the foundation of our to recalculate the time complexity of the algorithm K-means, Fuzzy C-Means and Agglomerative Clustering to further clarify owned complexity of each algorithm according to pseudo code approach. The following is a discussion of each algorithm.

1) K-means Algorithm

Referring to figure 1 above, the line (1) shows the process of selecting initial cluster centers of the data set is set X. At least on this line requires a random search n times, so the time complexity is owned line (1) is $O(n)$.

In line (2) to (14), each time checking if the cluster has met the search criteria or not, at least do a search as many (n-1) times the loop body in it. Each time line (2) is executed, it will at least run two looping in it, ie on the line to (3) and line (8). The same, applies to each iteration that are in line (3) and (8), each iteration in it will run n times at each iteration body. So for looping the line (3) has a time complexity of $O(n)$, as well as looping the line (8) has a time complexity of $O(n)$.

Thus, the total time complexity is owned by K-means clustering algorithm based on the above pseudocode is $O(n) + (n\text{-}1).(O(n) + O(n)) = O(n) + (n\text{-}1).(max(O(n),O(n)) = O(n) + (n\text{-}1).O(n) = O(n) + O(n^2) – O(n) = O(n^2)$.

2) Fuzzy C-means Algorithm

In figure 2 for FCM algorithm, it can be seen that this algorithm runs in iteration with three instructions. Line to 5 to 8, indicate that this line has a time complexity of $(n\text{-}1)$. $O(1)$ or in other words of $O(n\text{-}1)$. While the looping line 2, each time it is run, will spend as much time on the body n times iteration, and n times the time spent on the body of the loop line to 3 each time the loop is executed on the first line. In other words, the total time for the use of kompleksita FCM algorithm is $n.O(n).O(n\text{-}1) = O(n^2).O(n\text{-}1) = O(n^3)\text{-}O(n) = O(n^3)$.

3) Hierarchy Agglomerative Clustering Algorithm.

Pseudocode in Figure 3, shows that in line 1 has a time complexity of $O(1)$, required for $O(n)$ to choose R_0 as initial cluster of a number of existing data sets. In the third row, required by $(n\text{-}1)$ times the amount of complexity that exist in the body of the loop of lines 5-8. At line (5), required at least $O(n)$ time to search the nearest cluster of $R_{t\text{-}1}$. So is the line (8) need for $O(n)$ time to form a new cluster R_t. Thus the total time complexity hierarchy which is owned by the agglomerative clustering algorithm is $O(1) + O(n) + (n\text{-}1).(O(n) + O(n)) = O(1) + O(n) + (n\text{-}1).(max(O(n),O(n)) = O(1) + O(n) + (n\text{-}1).O(n) = O(1) + O(n) + O(n^2)\text{-}O(n) = O(n^2)$.

B. Running Time Analysis

This section discusses the implementation of each algorithm in MATLAB to test the application running time of every algorithm. Tests carried out using sample data taken from John Rasp's Website Statistics, Stetson University - Florida (Stetson). The data sample consists of 16 parameters clustering and forming as many as 252 dataset [25]. This data is a collection of a percentage of body size measurements are based on 16 parameters. The test is divided into five stages, and divide the data set into multiples of two, starting with the 50 data sets, 100, 150, 200 and 252. Each test is divided into three times running time measurements to ensure the accuracy of the running time, which is owned by each algorithm in handling a number of different data. Here is a table of test results (see Table 1-3).

Table I. The first measurement

Alg.	Sample Data set				
	50	100	150	200	252
K-means	0,2620	0,2840	0,3546	0,3737	0,3653
FCM	0,1289	0,1388	0,2170	0,2212	0,4984
HAC	0,1016	0,0938	0,1051	0,1398	0,1347

Table 1-3 shows the measurement results for all tests on data sets that have been shared. The reading of the results is done from left to right, where, K-means clustering algorithm on the first measurement time running time has increased each time tested by the number of different data sets. The more data sets, the running time required by the K-means clustering algorithm increases. So also with the results of the second and third measurements. (see Figure 1 for a comparison chart running time of each algorithm on the first measurement).

Table II. The second measurement

Alg.	Sample Data set				
	50	100	150	200	252
K-means	0,2533	0,2863	0,3411	0,3493	0,3669
FCM	0,1294	0,1481	0,1972	0,5436	0,5028
HAC	0,0808	0,0939	0,1021	0,1172	0,1327

Table III. The third measurement

Alg.	Sample Data set				
	50	100	150	200	252
K-means	0,2657	0,2763	0,3414	0,3467	0,3667
FCM	0,1292	0,4468	0,4401	0,2112	0,7141
HAC	0,0804	0,0927	0,1035	0,1160	0,1329

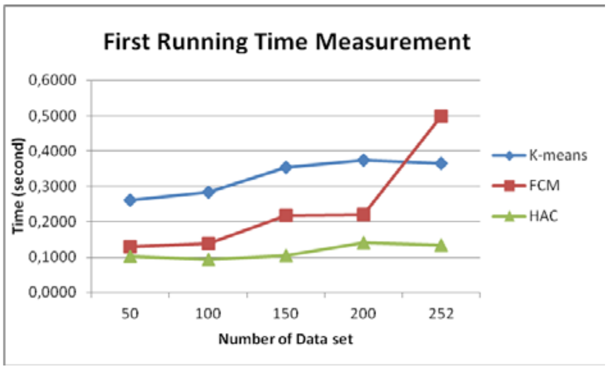

Figure 4. First running time measurement chart

As for the FCM algorithm, both on the first measurement, second and third (see table 1-3), the running time required fluctuated, however, globally, an increase in running time requirements for large data sets resulted in the use of a large running time anyway . (See Figure 2 for a comparison chart running time on the second measurement).

Unlike the HAC algorithm, although seen increased use of running time according to the number of data sets used, HAC still has the smallest running time comparison between the two algorithms.

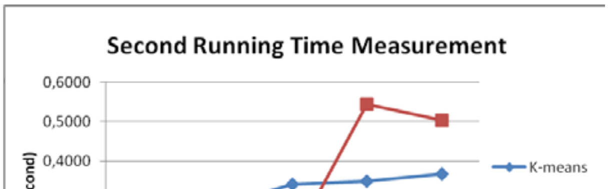

Figure 5. Second running time measurement chart

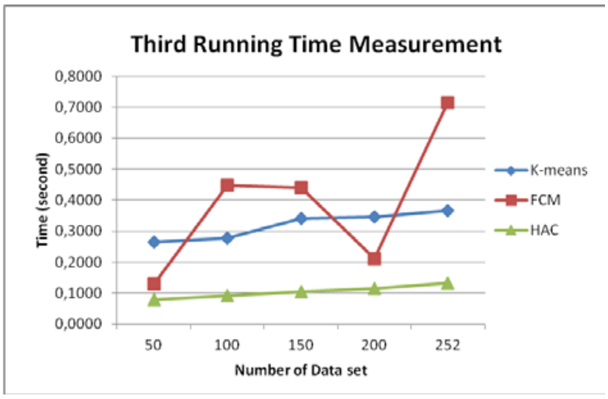

Figure 6. Third running time measurement chart

Table IV. The average of running time

Algorithm	1st Meas	2nd Meas	3rd Meas
K-means	0,32791	0,31939	0,31936
FCM	0,24086	0,30423	0,38829
HAC	0,11500	0,10535	0,10512

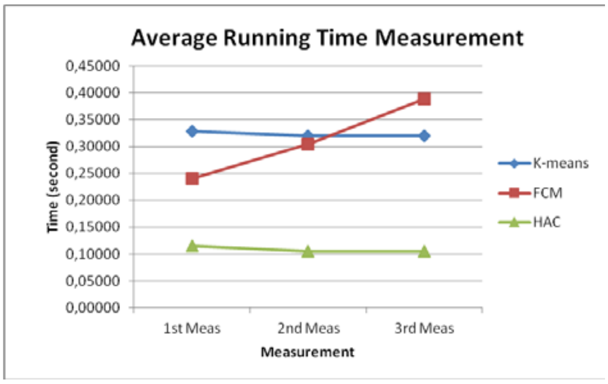

Figure 7. The average of running time chart

Judging from the average running time first to third measurement results (see Table 4) in this case, it was clear that, FCM algorithm has a running time rate continues to climb along with the increasing data sets. Unlike the K-means clustering algorithm which decreased running time although not significantly, as does the HAC algorithm. However, among the three, HAC algorithm has a level smaller running time (see figure 7).

C. Comparative analysis of Primitive Operation and Running Time

The results of the analysis indicate that primitive operations, the level of complexity of the FCM algorithm is bigger than the other two algorithms, in other words, the Big-O analysis results indicate that the *HAC < FCM > K-Means*. Three relationships were then tested using MATLAB application to see the correlation between the time complexity analysis of the actual running time. As a result, the HAC algorithm has running time rate is much smaller than the K-means, although according to the Big-O analysis, both have the same time complexity. But in this case, the HAC has a running time that is much smaller than the FCM and K-means, even the value of its running time so far when compared with FCM algorithm (see figure 7).

V. CONLUSION

Based on the results of tests performed, either using an approach based on the concept of a primitive operation Big-O, as well as the analysis of the running time, it can be seen that, in general, running time analysis proves the existence of a correlation between the Big-O analysis is based on the average value of the running time of the three algorithms. However, in this case, both K-means and HAC, although both have the same time complexity, in fact has a much different running time, which, HAC can be said to be more stable in terms of the use of running time because the changes are not too significant compared with K-means clustering for each data set tested. In addition, the analysis shows that the running time, K-means has a running time of three times that of HAC.

VI. REFERENCES

[1] Witten, Ian H., and Eibe Frank. 2005. Data Mining : Practical Machine Learning Tools and Techniques, Second Edition. Morgan Kaufmann Publishers. San Fransisco.

[2] Alfina, Tahta., Budi Santosa and Ali Ridho Barakhah. 2012., "Analisa Perbandingan Metode Hierarchical Clustering, K-means dan Gabungan Keduanya Dalam Cluster Data (Study Kasus : Problem Kerja Prakterk Jurusan Teknik Industri ITS)", Jurnal TEKNIK ITS, Vol. 1, pp. 521 – 525. Available in Bahasa.

[3] Ross, Timothy J., 2010. Fuzzy Logic With Engineering Applications 3rd Edition. John Wiley and Sons Ltd. Publication. Mexico. United State of Americca.

[4] Krishnaiah, V.V Jaya Rama., D.V Chandra Sekar and K. Ramchand H Rao. 2012. "Data Analysis of Bio-Medical Data Mining Using Enhanced Hierarchical Agromerative Clustering". International Journal of Engineering and Innovative Technology (IJEIT), Vol. 2, Issue. 3, pp. 43 – 49

[5] Abu-Dalbouh, Hussain and Norita Md Norwawi. 2011. "Bidirectional Agglomerative Hierarchical Clustering Using AVL Tree Algorithm", International Journal of Computer Science Issues. Vol. 8, Issue : 5, pp. 95 – 102

[6] Shalom, Arul, S.A., and Manoranjan Dash. 2013. "Efficient Partitioning Based Hierarchical Agglomerative Clustering Using Graphics Accelerators With CUDA". International Journal of Artificial Intelligence & Applications, pp. 13 - 33

[7] Abu Abbas, Osama., 2008. "Comparisons Between Data Clustering Algorithms". The International Arab Journal of Information Technology, Vol. 5, No. 3, pp. 320 – 325

[8] Sipser, Michael, 2006. Introduction to the Theory of Computation – Second Edition. Thomson Course Technology. Massachusetts.

[9] Jun He and Xin Yao. 2004. "Time Complexity of an Evo For Finding Nearly Maximum Cardin", Journal Comput. Sci & Technol., Vol. 19, No. 4, pp. 450 – 458

[10] Oliveto, Pietro S., Jun He and Xin Yao. 2007. "Time Complexity of Evolutionary Algorithms for Combinatorial Optimization : A Decade of Results". International Journal of Automation and Computing, Vol. 04(3), pp. 281 – 293

[11] Singer, Sanja and Sasa Singer. 1999. Complexity Analysis of Nelder-Mead Search Iteration. In Proceedings of the 1st Conference on Applied Mathematics and Computation, pp. 185- 196.

[12] Chen, Tianhsi., Ke Tang, Guoliang Chen and Xin Yao. 2007. On The Analysis of Average Time Complexity of Estimatiosn of Distribution of Algorithms, Proceeding of IEEE Congress on Evolutionary Computation 2007

[13] Shi, Guoyong, 2010. "Computational Complexity Analysis of Determinant Decision Diagram", IEEE Transactions on Circuits and Systems – II : Express Briefs, Vol. 57, No. 10, pp. 828 – 832

[14] Chhajed, Nidhi., Imran Uddin and Simarjeet Singh Bhatia. 2013. "A Comparison Based Analysis of Four Different Types of Shorting Algorithms in Data Structures with Their Performance", International Journal of Advance Research in Computer Science and Software Engineering, Vol. 3, Issue : 2, pp. 373 - 381

[15] Sharma, Kuldeep and Deepak Garg. 2009. "Complexity Analysis in Heterogeneous System", Journal of Computer and Information Science, Vol. 2, No. 1, pp. 48 – 52

[16] Sharma, Ritu., M. Afshar Alam and Anita Rani. 2012. "K-Means Clustering in Spatial Data Mining Using Weka Interface", In Proceeding of International Conference on Advances in Communication and Computing Technologies (ICACACT) 2012

[17] Suresh L., Jay B. Simha and Rajappa Velur. 2009., "Implementing K-means Algorithm Using Row Store and Column Store Database : A Case Study", International Journal of Recent Trends in Engineering, Vol. 2, No. 4, pp. 96 – 100

[18] Patel, Bhagwati Charan and G.R Sinha. 2010. "An Adaptive K-means Clustering Algorithm for Breast Image Segmentation", International Journal of Computer Applications, Vol. 10, No.4, pp. 35 – 38

[19] Hung, Ming-Chuan., Jungpin Wu and Jin-Hua Chang. 2005. "An Efficient K-means Clustering Algorithm Using Simple Partitioning", Journal of Information Science and Engineering, Vol. 21, pp. 1157 - 1177

[20] Lala, Archana., Jitendra Kumar Gupta and Mrinalini Shringirishi. 2013. "Implementation on K-means Clustering and Fuzzy C-means Algorithm For Brain Tumor Segmentation", International Journal of Computer Engineering & Science, Volume 3, Issue 1, pp. 27-33

[21] Shambharkar, Saroj and Shubhangi Tirpude. 2011. Fuzzy C-Means Clustering For Content Based Image Retrieval System, In Proceeding of 2011 International Conference on Advancements in Information Technology With workshop of ICBMG 2011

[22] Bezdek, James C., Robert Ehrlich and William Full. 1984. "FCM : The Fuzzy C-Means Clustering Algorithm", Journal of Computers & Geosciences, Vol. 10, No. 2 – 3, pp. 191 – 203.

[23] Rajalingam, N., and K. Ranjini. 2011. "Hierarchical Clustering Algorithm - A Comparative Study", International Journal of Computer Applications, Volume 19– No.3, April 2011, pp. 42 - 46

[24] http://cs.joensuu.fi/pages/oili/PR/?a=Some__Material&b= Hierarchical__Clustering

[25] John Rasp's Statistics Website - Data Sets for Classroom Use, available : http://www2.stetson.edu/~jrasp/data.htm