Global Journal of Computer Science and Technology
Volume 11 Issue 17 Version 1.0 October 2011
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Implementation of K-Means Clustering Algorithm Using Java

By Prof. S.China Venkateswarlu, Prof. M.Arya Bhanu, Prof.Yudhaveer Katta, V.Badari

Department of CSE ,JNTUH,HITS COE-HYDERABAD

Abstract - Emergence of modern techniques for scientific data collection has resulted in large scale accumulation of data pertaining to diverse fields. Conventional database querying methods are inadequate to extract useful information from huge data analysis. Cluster analysis is one of the major data analysis methods and k-means clustering algorithm Emergence of modern techniques for scientific data collection has resulted in large scale accumulation of data per-tainting diverse felids. Conventional Data base methods are inadequate to extract useful information from huge data banks. Cluster analysis is one of the major data analysis methods and the k-means clustering algorithm is widely used for many practical applications. But the original k-means algorithm is computationally expensive and the quality of the resulting clusters heavily depends on the selection of initial cancroids. Several methods have been proposed in the literature for improving the performance of the k-means clustering algorithm. The k-means algorithm is computationally expensive and requires time proportional to the product of the number of data items, number of clusters and the number of iterations.This papert proposes a method for making the algorithm more effective and efficient.

Keywords : *About four key words or phrases in alphabetical order, separated by commas.*

GJCST-C Classification : *I.5.3*

IMPLEMENTATION OF K-MEANS CLUSTERING ALGORITHM USING JAVA

Strictly as per the compliance and regulations of:

Implementation of K-Means Clustering Algorithm Using Java

Prof. S.China Venkateswarlu[α], Prof. M.Arya Bhanu[Ω], Prof.Yudhaveer Katta[β], V.Badari[Ψ]

Abstract - Emergence of modern techniques for scientific data collection has resulted in large scale accumulation of data pertaining to diverse fields. Conventional database querying methods are inadequate to extract useful information from huge data analysis. Cluster analysis is one of the major data analysis methods and k-means clustering algorithm Emergence of modern techniques for scientific data collection has resulted in large scale accumulation of data per- tainting diverse felids. Conventional Data base methods are inadequate to extract useful information from huge data banks. Cluster analysis is one of the major data analysis methods and the k-means clustering algorithm is widely used for many practical applications. But the original k-means algorithm is computationally expensive and the quality of the resulting clusters heavily depends on the selection of initial cancroids. Several methods have been proposed in the literature for improving the performance of the k-means clustering algorithm. The k-means algorithm is computationally expensive and requires time proportional to the product of the number of data items, number of clusters and the number of iterations.This papert proposes a method for making the algorithm more effective and efficient.

Keywords : About four key words or phrases in alphabetical order, separated by commas.

I. Introduction

a) Module design and organization

i. Fixed Transmission

In this module the fixed transmission is computed by retrieving each link which is limited to maximum signal ratio. Then the fixed routes are computed which give the maximum transmitted power per node and is limited to hardware constraint. Then the distance between the nodes is compute to calculate the end-to-end reliability. The end-to end reliability is a decreasing function, which can be treated as the cost metric for route selection.

ii. End to End Reliability

This module focuses on the problem of optimizing transmission power levels and route selection on an end-to end basis. This module minimizes the end-to end power for a fixed route. The end-to-end route reliability under the optimal power allocation scheme is represented as any fixed route, different power allocation schemes result in different end-to-end reliability and power consumption. The next step is to retrieve the total bandwidth then the distance between nodes are minimized and reliability routes are computed.

iii. Outage Diversity

In the module the case of a point-to-point link, is considered and the trade-off between route outage and consumed power in a network setting. This type of analysis gives insight to how fast the end-to-end outage decreases as more power is spent on the transmission. First, we look at the case that the maximum transmitted power at each link is fixed. It is observed that the route selection does not have any effect on the form of this tradeoff. By selecting the optimal route, we minimized the end to end outage probability by minimizing.

This shows that as long as we limit our approach to a single transmitter and a single receiver per link, even under optimal power allocation and route selection, the trade-off maintains the same form as in the single link case. Phase 1 of the heuristic algorithm requires a time complexity of O(nkp) for finding the initial centroids, as the maximum time required here is for computing the distances between each data point and all other data-points in the set D. In the original k-means algorithm, before the algorithm converges, the centroids are calculated many times and the data points are assigned to their nearest centroids. Since complete redistribution of the data points takes place according to the new centroids, this takes O(nkl), where n is the number of data-points, k is the number of clusters and l is the number of iterations. To obtain the initial clusters, algorithm 4 requires O(nk). Here, some data points remain in its cluster while the others move to other clusters depending on their relative distance from the new centroid and the old centroid. This requires O(1) if a data-point stays in its cluster, and O(k) otherwise. As the algorithm converges, the number of data points moving away from their cluster decreases with each iteration. Assuming that half the data points move from their clusters, this requires O(nk/2). Hence the total cost of this phase2 of the heuristic algorithm is O(nk), not O(nkl). Thus the overall time complexity of the heuristic algorithm becomes O(nkp).

In an Enhanced k-means approach[4] we are not calculating the distance of elements from each

Author[α] : Professor & HOD in Department of CSE ,JNTUH,HITS COE-HYDERABAD , 08CS5028 : +91 9030076793;
E-mail : cvenkateswarlus@gmail.com
Author[Ω] : Professor & HOD in Department of IT, VJIT,
Author[β] : Prof. Yudhaveer Katta, working as a Professor & HOD in Department of IT, VJIT,
Author[Ψ] : Assistant Professor in Dept. of CSE-HITS.

Global Journal of Computer Science and Technology Volume XI Issue XVII Version I October 2011

centroid. The algorithm proceeds in the following manner.Once the initial centroids are thus determined, the distance between each data point and centroids of all the clusters are determined and the data points are included in the nearest cluster. Cluster means are then recalculated to find the new centroids. As a result of this, some of the data points may become more closer to a different cluster and such points are redistributed accordingly. The entire process is repeated until no more data points cross cluster boundaries. The enhancedmethod is described as Algorithm 4. This method involves keeping track of the distance between each data point and the centroid of its nearest cluster. During the subsequent iteration, instead of computing the distance of the data point from all cluster centroids, its distance from the previous nearest cluster alone is determined. If that distance is less than or equal to the previous nearest distance, the data point.

b) Algorithm: finding the initial centroids

i. *Finding Intial Centroids*

Input:
D={d1,d2,…..dn} // set of n data items
K // number of desired cluster Output: A set of k initial centroids.
Steps:
1.Set m=1:
2. Compute the distance between each data point and all other data-points in the set D:
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;
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:
5. Repeat step 4 until the number of data points in Am reaches 0.75* (n/k);
6. If m< k, then m=m+1, find another pair of data points from D between which the distance is the shortest, from another data –point set Am and delete them from D, Go to step 4:
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.

Algorithm 3 describes the method for finding initial centroids of the clusters [12]. 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. 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 discussed in [4]. 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 enhanced method are described below as Algorithm 3 and Algorithm 4.

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 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 presentnearest 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. This 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.

c) Algorithm: assigning data-points to clusters [1]

Assininig DataPoints to Centroids
Input:

Global Journal of Computer Science and Technology Volume XI Issue XVII Version I October 2011

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

Steps:

1. Compute the distance of each data point di (1<=i<=) to all the centroids cf (1<=j<=k) as d(di, cj);
2. For each data-point di, find the closest centroid cf and assign di to cluster j.
3. Set CllusterId[i]=j; // j: Id of the closest cluster
4. Set Nearest _ Dist[i] = d(di, cj) ;
5. For each cluster j (1<= j<= k) troids:
6. Repeat
7. For each data-point di.
8. Compute its distance is less than or equal to the present nearest cluster:
9. If this distance is < or = to the present nearest distance , the data- point satays in the cluster:
   Else
10. For every centroid cj (1 <=j<= k) Compute the distance d(di, cj); Endfor;
11. Asign the data-point di to the cluster with the nearest centroid cj
12. Set Cluster ID[i];
13. Set Nearest_Dist[i] = d(di, cf) ;
14. Endfor;
15. For each cluster j (1<= j<=), recalculate the centroids:
16. Until the convergence criterin is met.

Phase 1 of *2* the enhanced algorithm requires a time com- plexity of *O (n)* for finding the initial centroids, as the max-imum time required here is for computing the distances be- tween each data point and all other data-points in the set D. Inthe original k-means algorithm, before the algorithm converges the centroids are calculated many times and thedata points are assigned to their nearest centroids. Since complete redistribution of the data points takes place ac-cording to the new centroids, this takes *O (nkl)*, where *n* is the number of data-points, *k* is the number of clusters and *l* is the number of iterations. To obtain the initial clusters, Algorithm4 requires *O(nk)*. Here, some data points remain in its cluster while the others move to other clusters depending on theirrelative distance from the new centroid and the old centroid. This requires *O(1)* if a data-point stays in its cluster, and *O(k)* otherwise. As the algorithm converges, the number of data points moving away from their cluster decreases with eachiteration. Assuming that half the data points move from their clusters, this requires *O (nk/2)*. Hence the total cost of thisphase of the algorithm is *O(nk)*, not *O(nkl)*. Thus the overall time complex*2*ity of the enhanced algorithm (Algorithm 2) becomes *O (n)*, since *k* is much less than *n*.

II. Conclusion

In this section we have shown how testing is performed and different test cases are designed to test the system for its performance as well as debugging process. The validation of the test cases is also shown. The k-means algorithm is widely used for clustering large sets of data. But the standard algorithm do not always guarantee good results as the accuracy of the final clusters depend on the selection of initial centroids. Moreover, the computational complexity of the standard algorithm is objectionably high owing to the need to reassign the data points a number of times, during every iteration of the loop. This Project presents an enhanced k-means algorithm which combines a systematic method for finding initial centroids and efficient way for assigning data points to clusters.

III. Implementation and Results

a) Introduction

In the module the case of a point-to-point link, is considered and the trade-off between route outage and consumed power in a network setting. This type of analysis gives insight to how fast the end-to-end outage decreases as more power is spent on the transmission. First, we look at the case that the maximum transmitted power at each link is fixed. It is observed that the route selection does not have any effect on the form of this tradeoff. By selecting the optimal route, we minimized the end to end outage probability by minimizing

IV. Implementation and Results

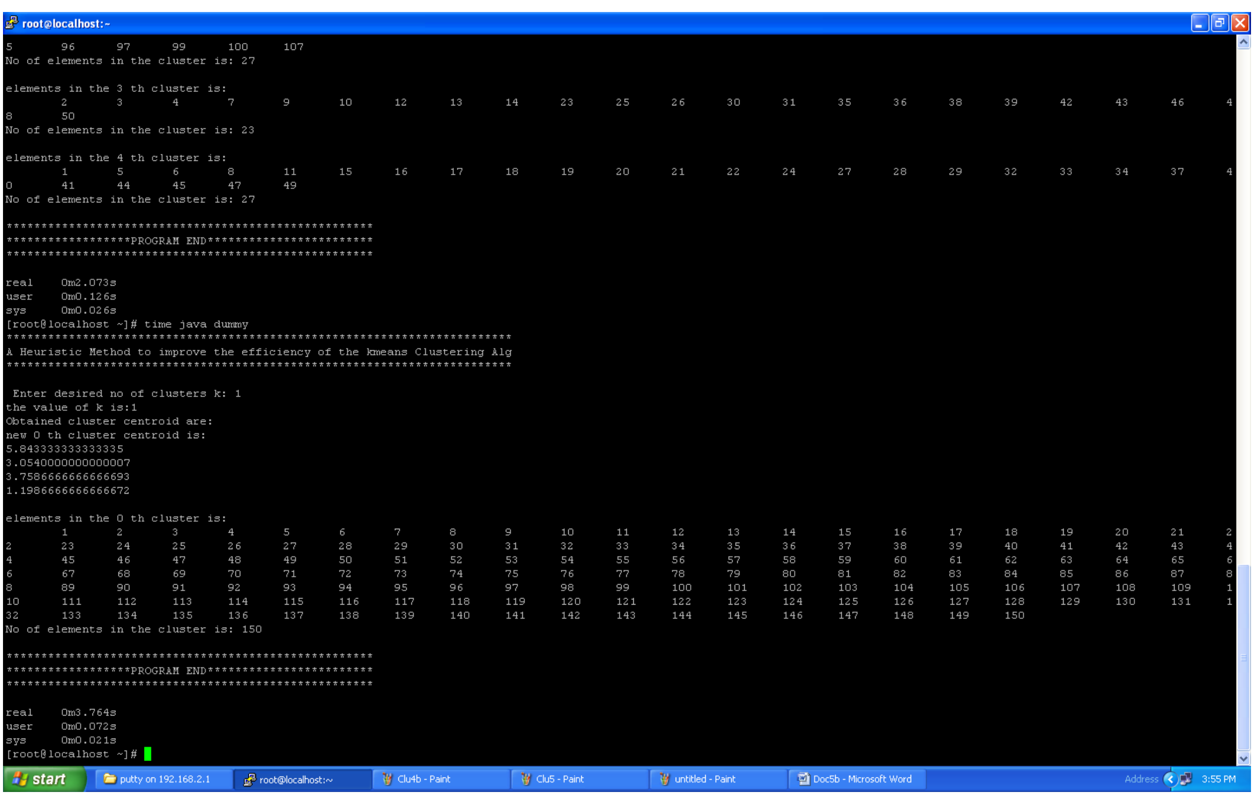
Algorithm 3 describes the method for finding initial centroids of the clusters [12]. Initially, compute the distances between each data point and all other data points in the set of data points. Then find out the closest pair of data points and form a set A1 consisting of these two data points, and delete 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. 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 discussed in [4]. 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 enhanced method are described below as Algorithm 3 and Algorithm 4.

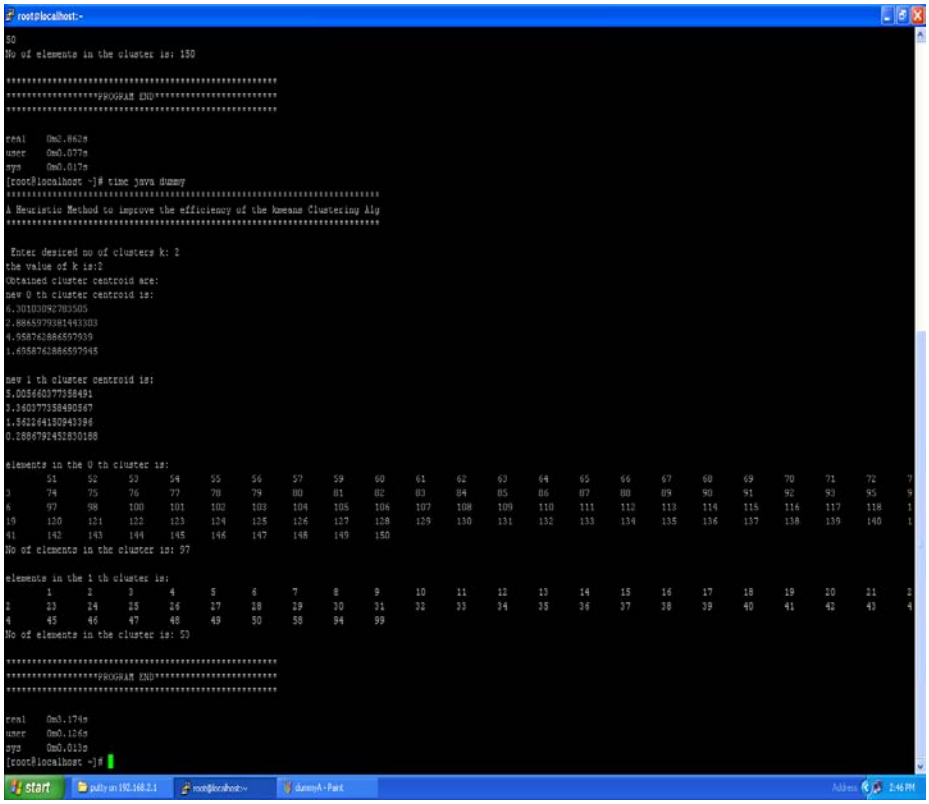
The first step in Phase 2 is to determine the distance between each data-point and the initial centroids of all the clusters. The data-points are then assigned to the clusters having the closest centroids. This results in an initial grouping of the data-points. For each data-point, the cluster to which 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 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 presentnearest 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. This 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.

V. OUTPUT

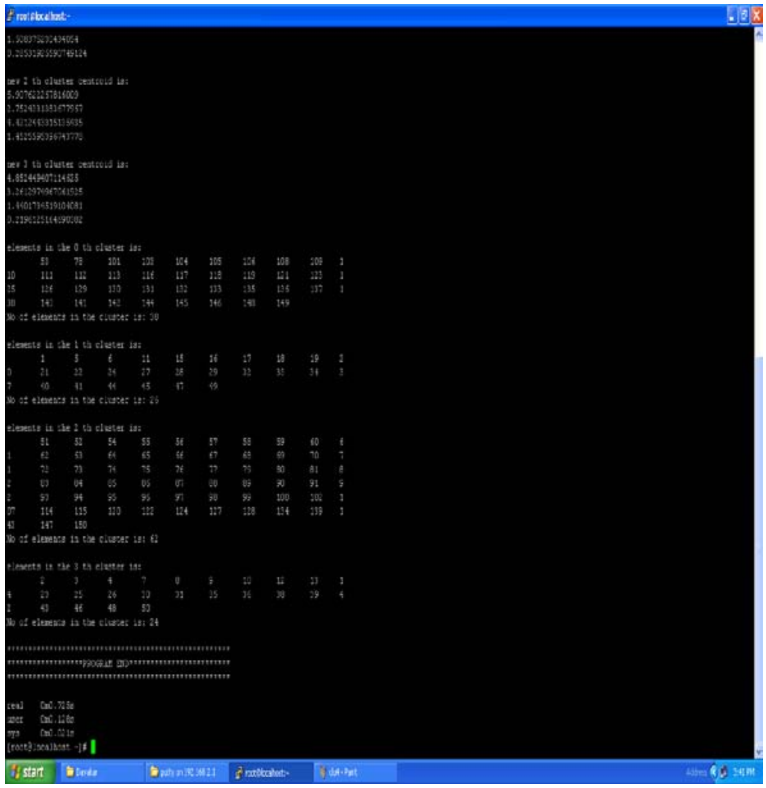
The modified algorithm is applied to multidimensional gene expression data taken from the UCI(university of california irvine) repository[7]. The input dataare the iris data[10], the breast cancer data[11], the e coli data[9], the echo cardiogram data[12], the yeast data[13] and the height-weight data obtained from the web site of disabled-world[8]. The results are compared with that of the original k-means algorithm as well as Enhanced k-means algorithm. Tables 6.1 to 6.6 show the performance comparison of the three algorithms. Figures 6.1 to 6.6 illustrate that the modified algorithm provide better accuracy and efficiency compared to the k-means and enhanced k-means methods.

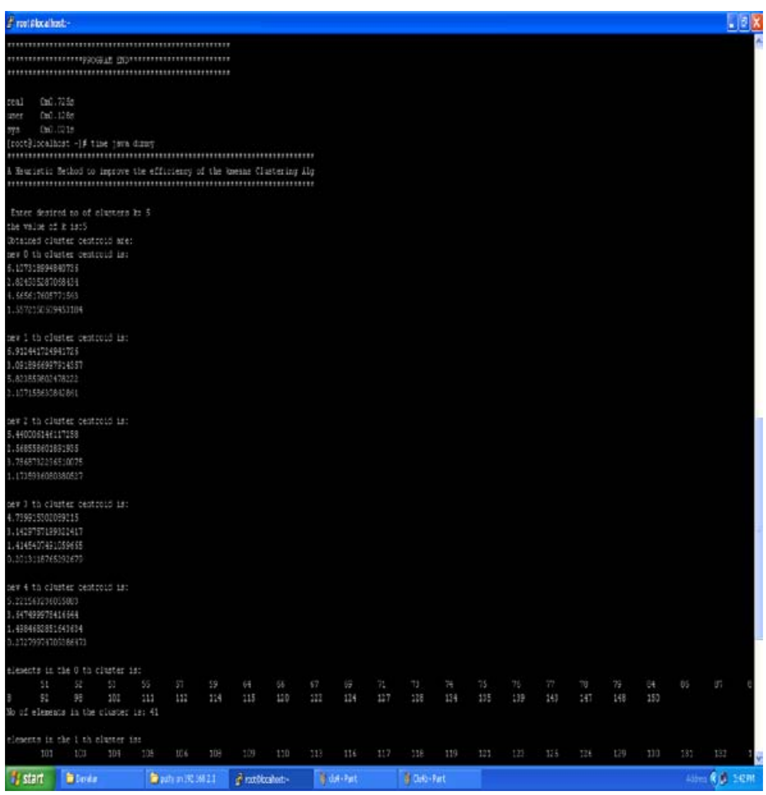


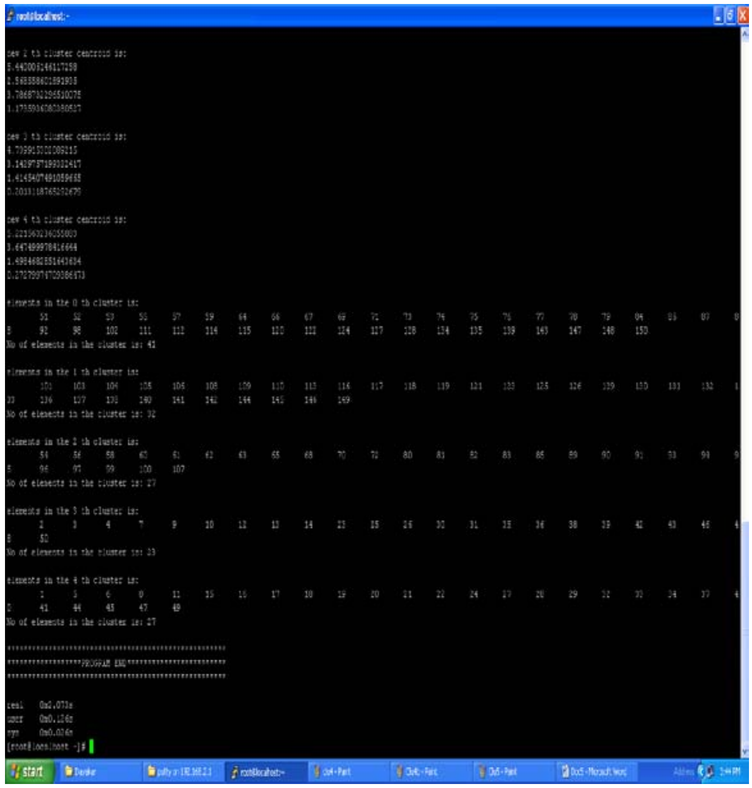


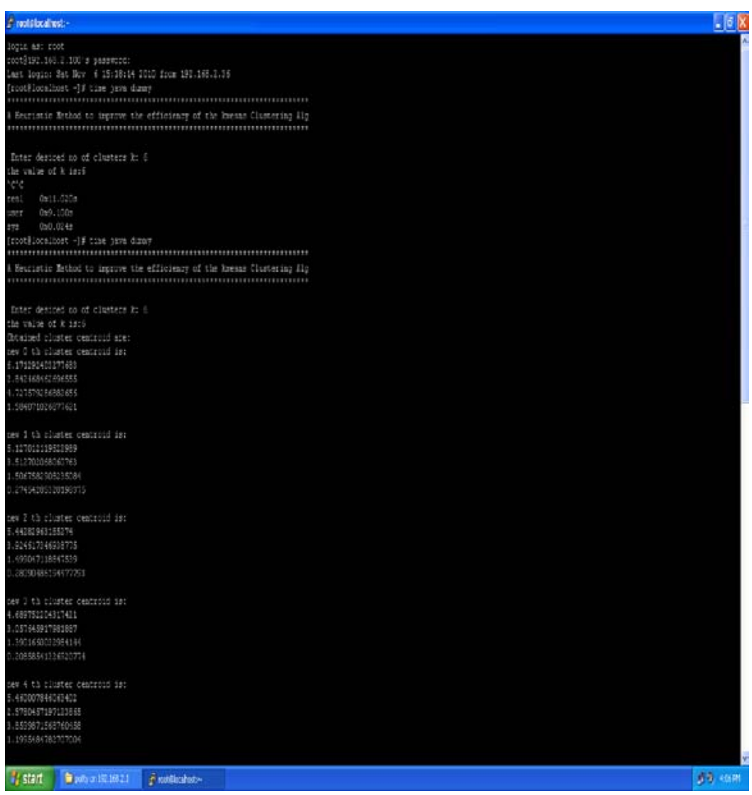
root@localhost:~
50
No of elements in the cluster is: 150
*****************PROGRAM END***********************
real 0m2.862s
user 0m0.077s
sys 0m0.017s
[root@localhost ~]# time java dummy
A Heuristic Method to improve the efficiency of the kmeans Clustering Alg
Enter desired no of clusters k: 2
the value of k is:2
Obtained cluster centroid are:
new 0 th cluster centroid is:
6.30103092783505
2.88659793814433O3
4.958762886597939
1.6958762886597945
new 1 th cluster centroid is:
5.005660377358491
3.360377358490567
1.562264150943396
0.2886792452830188
elements in the 0 th cluster is:
51 52 53 54 55 56 57 59 60 61 62 63 64 65 66 67 68 69 70 71 72
74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 95
97 98 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118
120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140
142 143 144 145 146 147 148 149 150
No of elements in the cluster is: 97
elements in the 1 th cluster is:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
45 46 47 48 49 50 58 94 99
No of elements in the cluster is: 53
*****************PROGRAM END***********************
real 0m3.174s
user 0m0.126s
sys 0m0.013s
[root@localhost ~]#
start
putty on 192.168.2.1
root@localhost:~
dummyA - Paint
2:46 PM

root@localhost:~
sys 0m0.029s
[root@localhost ~]# time java dummy
A Heuristic Method to improve the efficiency of the kmeans Clustering Alg
Enter desired no of clusters k: 3
the value of k is:3
Obtained cluster centroid are:
new 0 th cluster centroid is:
5.965145697477419
2.769178762110372
4.482486949119113
1.4723478133866412
new 1 th cluster centroid is:
5.003085412312783
3.388615290839429
1.5063857540926506
0.26185058559735275
new 2 th cluster centroid is:
6.821700928381963
3.0659037763506904
5.695787379589556
2.0505462096886777
elements in the 0 th cluster is:
51 52 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
74 75 76 77 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95
97 98 99 100 102 107 114 115 120 122 124 127 128 134 139 143 147 150
No of elements in the cluster is: 62
elements in the 1 th cluster is:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
45 46 47 48 49 50
No of elements in the cluster is: 50
elements in the 2 th cluster is:
53 78 101 103 104 105 106 108 109 110 111 112 113 116 117 118 119 121 123 125 126
130 131 132 133 135 136 137 138 140 141 142 144 145 146 148 149
No of elements in the cluster is: 38
*****************PROGRAM END***********************
real 0m0.819s
user 0m0.135s
sys 0m0.022s
[root@localhost ~]#
start
putty on 192.168.2.1
root@localhost:~
dummyA - Paint
dummyB - Paint
2:47 PM

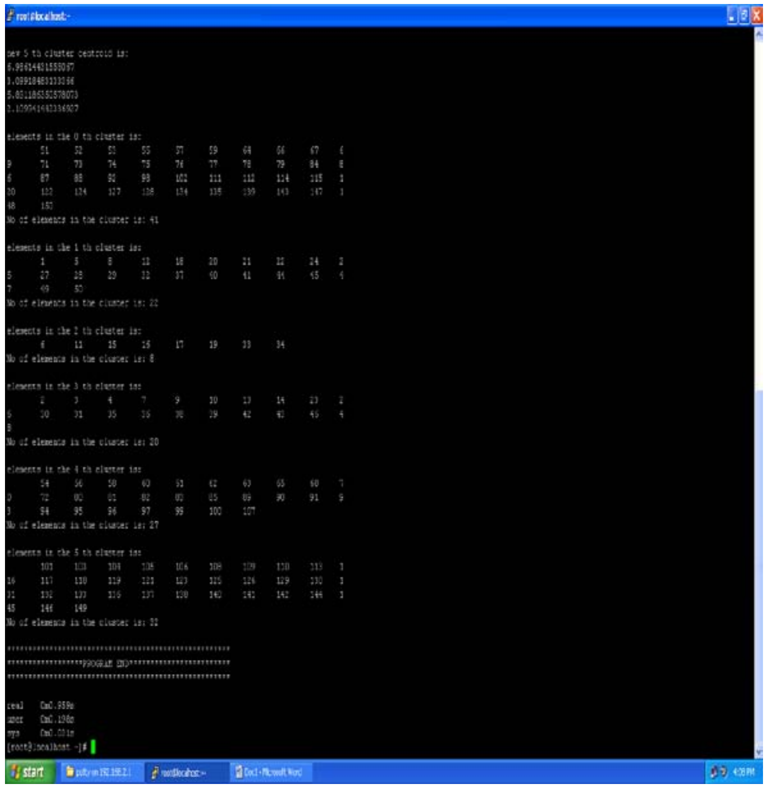








Global Journal of Computer Science and Technology Volume XI Issue XVII Version I October 2011



VI. Conclusion

In this section we have shown how testing is performed and different test cases are designed to test the system for its performance as well as debugging process. The validation of the test cases is also shown. The k-means algorithm is widely used for clustering large sets of data. But the standard algorithm do not always guarantee good results as the accuracy of the final clusters depend on the selection of initial centroids. Moreover, the computational complexity of the standard algorithm is objectionably high owing to the need to reassign the data points a number of times, during every iteration of the loop. This Project presents an enhanced k-means algorithm which combines a systematic method for finding initial centroids and efficient way for assigning data points to clusters.