World Applied Sciences Journal 32 (7): 1181-1191, 2014
ISSN 1818-4952

DOI: 10.5829/idosi.wasj.2014.32.07.929

Hybrid Approach of Hierarchical Clustering

[1]*Archana Singh and* [2]*Avantika Yadav*

[1]Amity School of Engineering and Technology, Amity University Uttar Pradesh, Noida, India
[2]Krishna Engineering College, Ghaziabad, U.P, India

Abstract: The clustering is process of grouping objects based on some similarity measure. In hierarchical clustering, the objects can be clustered on the basis of single linkage, average linkage or complete linkage. In this paper we have proposed a hybrid approach of clustering based on AGNES and DIANA clustering algorithms, an extension to the standard hierarchical clustering algorithm. In the proposed algorithm, we have used single linkage as a similarity measure. The proposed clustering algorithm provides more consistent clustered results from various sets of cluster centroids with tremendous efficiency.

Key words: Clustering · Hierarchical clustering · Linkages · Similarity matrix

INTRODUCTION

The Clustering is a process of forming group of objects of similar type based on some similarity measure. To achieve better clustering results the inter-cluster distance should be more and intra-cluster distance should be less. In the proposed algorithm, we have used single linkage mechanism to calculate the distance matrix at each step [1]. Hierarchical clustering is a technique of cluster analysis which is used to build a hierarchy of clusters [2]. Hierarchical cluster analysis (or hierarchical clustering) is a popular approach to cluster analysis, in which the group of objects is formed from together objects or records that are "near/similar" to one another [3]. A key component of analysis is repeated calculation of distance measures among objects and among clusters, once objects begin to be grouped into clusters. The result is represented graphically as a dendrogram [4]. Dendrogram is a graphical representation of the results of hierarchical cluster analysis).The initial data for the hierarchical cluster analysis of N objects is a set of $N\times(N-1)/2$ object-to-object distances and a linkage function [5] for computation of the cluster-to-cluster distances. A linkage function is an important characteristic for hierarchical cluster analysis. Its value is a measure of the "distance" between two groups of objects (i.e. between two clusters). The two main categories of methods for hierarchical cluster analysis are *divisive methods* and *agglomerative methods* [6]. Generally the agglomerative methods are used. On each step, the pair of clusters with smallest cluster-to-cluster distance is fused into a single cluster and finally all the objects are grouped into a single cluster. In divisive methods, on each step, the pair of clusters is divided into smaller clusters and at the final step all the clusters contain the single object.

In our present work, we have proposed an algorithm which is a hybrid approach [7] using the concept of AGNES (agglomerative approach) and DIANA (divisive approach) algorithm [8]. The algorithm is better combination of both algorithms AGNES and DIANA providing better functionality in reduced response time.

The organization of the entire paper is as follows, section-2 illustrates the related work regarding clustering. In section-3, the brief introduction of AGNES and DIANA algorithm with examples are discussed. In section-4 illustrates the concept of proposed algorithm using the same example. The section-5 demonstrates the experiments and results found from the proposed algorithm. In section-6 comparative study of three algorithms i.e. AGNES, DIANA and proposed AGGLO-DIVISIVE algorithm is presented. Section-7 gives explored the business applications of the proposed algorithm. Section-8 winds up with the conclusion and future scope.

Corresponding Author: Archana Singh, Amity School of Engineering and Technology, Amity University Uttar Pradesh, Noida, India.

Related Work (Year Wise): The researchers proposed various approaches of clustering. The distinguished authors described simple step-wise procedure for clustering. There are two alternative criteria for the merger of groups at each pass as follows:- (a) maximization of the pairwise correlation between the centroids of two groups and (b) minimization of wilks' statistic to test the hypothesis of independence between two groups. Douglass. R. Cutting (1992), *et al.* [5] discussed about the problem in document clustering and he attempted to improve conventional search techniques. The author also proposed a cluster-based approach to browsing large document collections. The authors P.K. Agarwal And C.M. Procopinc, 1998, [18] discussed about, How exact approximation for clustering algorithms is done? The complete process of approximation is also hashed out by the generators. In 1998, Daniel Boley [2] worked in the principal direction divisive partitioning. The method proposed by Daniel, is capable of partitioning a set of documents or other samples based on embedding in a dimensional Euclidean scope using divisive approach. In the algorithm documents are assembled in a matrix which is very sparse and this sparse improves provides efficiency. The authors discussed more about cluster analysis in volume 33 of social science research council reviews of current research. They also summed up the idea of dendrogram, linkages and similarity matrices. The clustering techniques for trees is explained by the authors V. Bagels, A. Mrvar and M. Zaversnik in 2000, in the work optimizing cluster sizes and number of sub-trees [10]. Eui-Hong Han and George Karypis, 2000, proposed centroid-based document classification analysis and experimental results [4]. This document explains the document classification process using the centroids and distance metrics. The various aspects of graph drawing are discussed by authors A. Quigly and P. Eades. Brian S Everitt, reviewed clusters and research done in the field of cluster analysis [14]. A fast multi-shell method for drawing large graphs is proposed by authors D. Harel and Y. Koren, 2001 [15]. They also explained the procedure for drawing large graphs in a very efficient manner. The authors G. Hamerly and C. Elkan discussed about the application of k-means in the neural network field. The authors as well explained how k-clusters of k-means can be trained for neural nets. The authors J. May-Six and I.G. Tollis, 2001, explored about a tool for visualizing graphs in her Ph.D. dissertation[20]. In 2005, survey of various clustering algorithms is performed by the authors RuiXu and Donald wunsh[9]. Their work includes working of algorithms, differences etc. The authors P.A. Vijaya, M. Narsimha Murty and D.K. Subramanian (2005) proposed a very efficient hybrid hierarchical agglomerative clustering [3]. Key differences among k-means clustering and validation measures on the basis of data distribution perspective, is explored by the authors Xiong Hui, Wu Jungie and Chen Jian (2009) [11]. Two level k-means clustering algorithm for k relationship establishment and linear time classification is explored by Radha Chitta and M. Narsimha Murty (2010) [21].

The above literature work imparted knowledge about various Cluster partitioning techniques. These techniques take more time to form clusters from the large data set and the complexity of individual approach is quite high. This paper is intended to propose a new hybrid clustering technique in order to reduce the complexity and execution time of clustering algorithm.

Introduction to Existing Hierarchical Clustering Algorithms

Agglomerative Clustering Algorithm: Agglomerative clustering algorithm is based on a bottom up approach. In this method initially all objects belong to distinct clusters and further clusters are formed by a footstep by step process on the basis of similarity measure and results obtained from the algorithm as represented in below algorithm [10].

Algorithm: AGNES
Input: 2-dimensional dataset.
Output: Clusters in 2-dimensional space.
Assumptions: Single linkage is used as a similarity measure and initially all objects are in distinct clusters.

Step 1: Begin with the disjoint clustering having level L(0) = 0 and sequence number m = 0.

Step 2: Find the least dissimilar pair of clusters in the current clusters, say pair (r, s), according to d[(r, s)] = min d[(i, j)], where the min is complete pairs of clusters in the current clustering.

Step 3: Increment the sequence number: m = m +1. Merge clusters (r) and (s) into a single cluster to form the next cluster. Make level of this clustering to L(m) = d[(r, s)]

Step 4: Update the similarity matrix, D, by deleting the rows and columns corresponding to clusters (r) and (s) and adding a row and column corresponding to the newly formed cluster. The similarity between the new cluster, denoted (r, s) and old cluster (k) is defined in this way: d[(k), (r,s)] = min d[(k, r)], d[(k, s)]. If all objects are in one cluster, stop. Else, repeat from step 2.

Working Example of AGNES: In the next lesson, we have chosen a small data set of six points A, B, C, D and E as represented in Table 1, to explain the working operation of existing algorithms of hierarchical clustering [11].

Table 1: Data Set

	X1	X2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5

The scatter plot of the points shown in Table 1 is represented in Fig 1. And distance matrix which is computed from the given points is calculated using Euclidian distance metric and represented in Table 2, denoted as matrix D0.

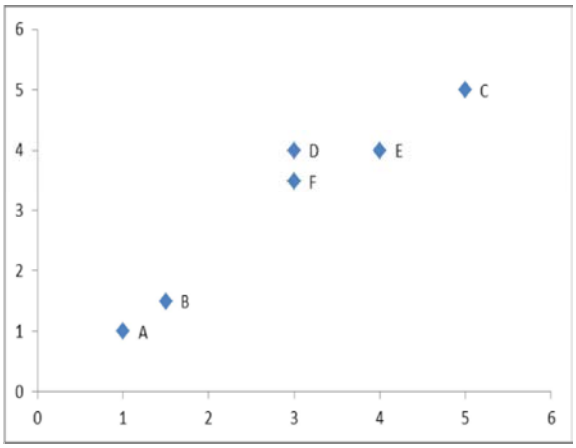

Fig. 1: Scatter Plot

Distance Matrix-D0: This is calculated using the Euclidian distance metric on the basis of given data-set. The minimum distance is among the points d and F, so these points are merged in the next step and further distances are calculated.

Distance (Single Linkage)

Distance	A	B	C	D	E	F
A	0	O.71	5.66	3.61	4.24	3.2
B	0.71	0	4.95	2.92	3.54	2.5
C	5.66	4.95	0	2.24	1.41	2.5
D	3.61	2.92	2.24	0	1	0.5
E	4.24	3.54	1.41	1	0	1.12
F	3.2	2.5	2.5	0.5	1.12	0

Distance Matrix-D1: It is observed from the above distance matrix that minimum distance is among the points d and f. So, further calculations will be performed to merge these points.

Min Distance (Single Linkage)

	A	B	C	D,F	E
A	0	O.71	5.66	?	4.24
B	0.71	0	4.95	?	3.54
C	5.66	4.95	0	?	1.41
D, F	?	?	?	0	?
E	4.24	3.54	1.41	?	0

Minimum distance between cluster B and cluster A is now 0.71.

Dist	A	B	C	D,F	E
A	0	O.71	5.66	3.2	4.24
B	0.71	0	4.95	2.5	3.54
C	5.66	4.95	0	2.24	1.41
D, F	3.2	2.5	2.24	0	1
E	4.24	3.54	1.41	1	0

Distance Matrix-D2: Form the previous matrix, the minimum distance is between the points A and B. So, the cluster A and cluster B is grouped into a single cluster name (A, B).

Min Distance (Single Linkage)

Distance	A,B	C	D,F	E
A,B	0	?	?	?
C	?	0	2.24	1.41
D, F	?	2.24	0	1
E	?	1.41	1	0

Using single linkage, we specify minimum distance between original objects of the two clusters. Using the input distance matrix, distance between cluster (D, F) and cluster A is computed as

$d_{(D,F)\text{-->}A}=\min(d_{DA}, d_{FA})=\min(3.61,3.20)=3.20$

Distance between cluster (D, F) and cluster B is

$d_{(D,F)\_B}=\min(d_{(DB}, d_{FB})=\min(2.92,2.50)=2.50$

Distance Matrix-D3: We can see that the closest distance between clusters happens between cluster E and (D, F) at distance 1.00. Thus, we cluster them together into cluster ((D, F), E).

Min Distance (Single Linkage)

Distance	A,B	C	D,F	E
A,B	0	4.95	2.5	3.54
C	4.95	0	2.24	1.41
D, F	2.5	2.24	0	1
E	3.54	1.41	1	0

Distance Matrix-D4:

Min Distance (Single Linkage)

Distance	A,B	C	(D,F),E
A,B	0	4.95	4.95
C	4.95	0	1.41
(D, F),E	2.5	1.41	0

Distance between cluster ((D, F), E) and cluster (A, B) is calculated as

$d_{((D,F),E)-(A,B)}=\min(d_{DA},d_{DB},d_{FA},d_{FB})=\min(3.61,2.92,3.20,2.5,4.24,3.54)=2.50$

Distance Matrix- D5:

Min Distance (Single Linkage)

Distance	(A,B)	((D,F),E),C)
A,B	0	4.95
((D,F),E),C)	2.5	0

$d_{(((D,F),E),c)-(A,B)}=\min(d_{DA},d_{DB},d_{FA},d_{FB}, d_{EA}, d_{EB} ,d_{CA} ,d_{CB})$

$d_{(((D,F),E),c)-(A,B)}=\min(3.61,2.92,3.20,2.5,4.24,3.54,5.66,4.95)=2.50$

Results (AGNES)

Results Obtained from AGNES:

- In the beginning we have clusters as : A, B, C, D, E and F.
- Clusters D and F are merged into cluster (D, F) at distance 0.50
- Clusters A and cluster B are merged into (A, B) at distance 0.71
- Clusters E and (D, F) are merged into ((D, F), E) at distance 1.00
- Clusters ((D, F), E) and C are merged into (((D, F), E), C) at distance 1.41
- F), E), C), (A, B)) at distance 2.50
- In the last step, cluster contain all the objects, thus terminate the computation.
- The hierarchy is given as (((D, F), E), C), (A, B).

Below graph [14-15] is obtained after the implementation of AGNES in JAVA programming language.

Below figure represents the obtained clusters in XY space.

Limitations of AGNES Algorithm: Main limitations of AGNES clustering method are :

- They do not scale well: time complexity of at least $O(n^2)$, where n denotes the number of total objects;

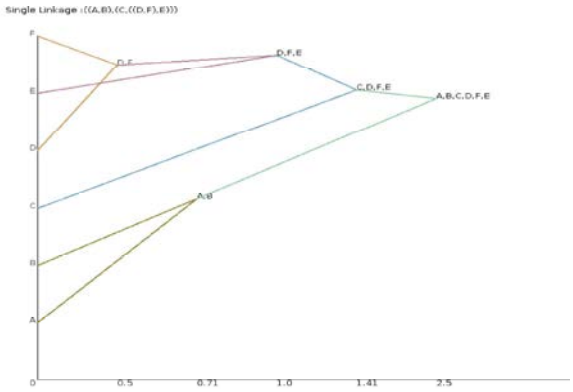

Fig. 2: Dendrogram AGNES

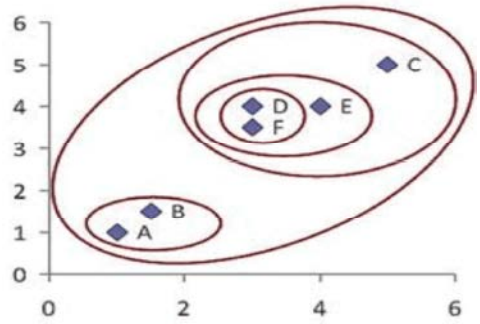

Fig. 3: Clusters in XY space

- The actions performed in previous steps can't be undone.

Divisive Algorithm: Divisive clustering algorithm is based on top-down approach. In this method initially all objects belong to single cluster and further clusters are formed by a step by step process on the basis of similarity measure and results obtained from the algorithm as represented in below algorithm.

Algorithm: DIANA
Input: 2-dimensional dataset.
Output: Clusters in 2-dimensional space.
Assumptions: Single linkage is used as a similarity measure and initially all objects are in single cluster.

Step 1: Find the object, which has the highest average dissimilarity to all other objects. This object initiates a new cluster- a sort of a splinter group.
Step 2: For each object i outside the splinter group compute
Step 3: Distancei = [average distance(i,j) j Ssplinter group] - [average distance(i,j) j Ssplinter group]

Step 4: Find an object x for which the difference Distancex is the largest. If Distancex is positive, then x is, on the average close to the splinter group.

Step 5: Repeat Steps 2 and 3 until all differences Distancex are negative. The data set is then split into two clusters.

Step 6: Select the cluster with the largest diameter. The diameter of a cluster is the largest dissimilarity between any two of its objects. Then divide this cluster, following steps 1-4.

Working Example of DIANA: Consider the same data set of 6 points. Calculate the distance matrix using the Euclidian distance matrix. Now, we will solve the same example using DIANA, which uses a divisive approach of clustering. The steps performed are as follows.

Distance Matrix-D0: In the first step, the algorithm has to split up the dataset into two clusters. This is not done by considering all possible divisions but rather by means of a kind of iterative procedure.

Distance	A	B	C	D	E	F
A	0	O.71	5.66	3.61	4.24	3.2
B	0.71	0	4.95	2.92	3.54	2.5
C	5.66	4.95	0	2.24	1.41	2.5
D	3.61	2.92	2.24	0	1	0.5
E	4.24	3.54	1.41	1	0	1.12
F	3.2	2.5	2.5	0.5	1.12	0

Object	Average dissimilarity to other objects
A	(0.71+5.66+3.61+4.24+3.20)/5=3.484
B	(0.71+4.95+2.92+3.54+2.50)/5=2.924
C	(5.66+4.95+2.24+1.41+2.50)/5=3.352
D	(3.61+2.92+2.24+1.00+0.50)/5=2.054
E	(4.24+3.54+1.41+1.00+1.12)/5=2.262
F	(3.20+2.50+2.50+0.50+1.12)/5=1.964

So object A is chosen to initiate the so-called splinter-group. At this stage we have the groups {a} and {b, c, d, e, f}. For each object of the larger group we compute the remaining objects and compare it to the average dissimilarity with the objects of splinter group.

Object	Average distance	Average distance to splinter group	Difference
B	(4.95+2.92+3.54+2.50)/4=3.4775	0.71	2.7675
C	(4.95+2.24+1.41+2.50)/4=2.775	5.66	-2.885
D	(2.92+2.24+1.00+0.50)/4=1.665	3.61	-1.945
E	(3.54+1.41+1.00+1.12)/4=1.7675	4.24	-2.4725
F	(2.50+2.50+0.50+1.12)/4=1.655	3.20	-1.545

So object B is chosen to add in the splinter-group. At this stage we have the groups {a, b} and {c, d, e, f}. For each object of the larger group we compute the remaining objects and compare it to the average dissimilarity with the objects of splinter group [13].

Object	Average distance	Average distance to splinter group	Difference
C	(2.24+1.41+2.50)/3=2.05	(0.71+5.66)/2=3.185	-1.135
D	(2.24+1.00+0.50)/3=1.2466	(3.61+2.92)/2=3.265	-2.01884
E	(1.41+1.00+1.12)/3=1.1766	(4.24+3.54)/2=3.89	-2.7134
F	(2.50+0.50+1.12)/3=1.8733	(3.20+2.50)/2=2.85	-1.4767

All negative. Therefore, to find the rebel to start the splinter-group with, we have

Object	Average distance
C	(2.24+1.41+2.50)/3=2.05
D	(2.24+1.00+0.50)/3=1.2466
E	(1.41+1.00+1.12)/3=1.1766
F	(2.50+0.50+1.12)/3=1.3733

So the object C is chosen to initiate another splinter-group. At this stage, we have groups {A, B},{C} and {D, E, F}. For each object of the larger group we compute the remaining objects and compare it to the average dissimilarity with the objects of splinter group.

Object	Avg. Dis	Avg. dis to SG1	Avg. dis to SG2	Diff 1	Diff 2
D	(1.00+0.50)/2=0.75	(3.61+2.92)/2=3.265	2.24	-2.515	-1.49
E	(1.00+1.12)/2=1.06	(4.24+3.54)/2=3.89	1.41	-2.83	-2.83
F	(0.50+1.12)/2=0.81	(3.20+2.50)/2=0.81	2.5	-2.04	-1.69

All negative. Therefore, to find the rebel to start the splinter-group with, we have

Object	Average distance
D	(1.00+0.50)/2=0.75
E	(1.00+1.12)/2=1.06
F	(0.50+1.12)/2=0.81

So object E is chosen to add in the splinter-group. At this stage we have the groups {A, B}, {C},{E} and{D, F}. For each object of the larger group we compute the remaining objects and compare it to the average dissimilarity with the objects of splinter group.

Diameter {A, B}=0.71 and diameter{D, F}=0.50, therefore, group with larger diameter will be splitted first.

- In the beginning we have single cluster as: (A, B, C, D, E, F).
- Cluster (A, B, C, D, E, F) is split into clusters (A) and (B, C, D, E, F).
- Cluster (B, C, D, E, F) is split into (C, D, E, F) and (A, B).
- Cluster (C, D, E, F) is split into (cluster C) and (D, E, F).
- Cluster (D, E, F) is split into cluster E and (D, F).
- Cluster (A, B) is split into cluster A, B.
- Cluster (D, F) is split into D and F.
- In the end we have single-single object in all clusters: (A), (B), (C), (D), (E), (F).

- The last clusters contain single object, thus terminate.

Graph [17] represented in below figure illustrates the clusters obtained from the implementation of DIANA in JAVA programming language.

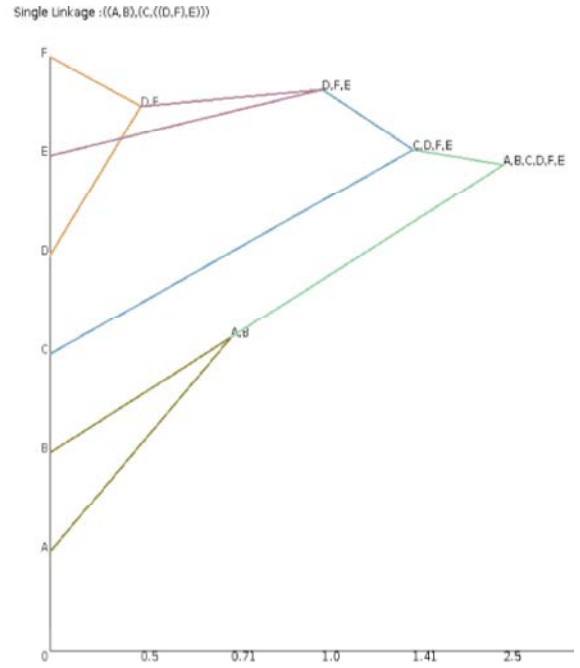

Fig. 4: Dendogram of DIANA

Proposed Algorithm:

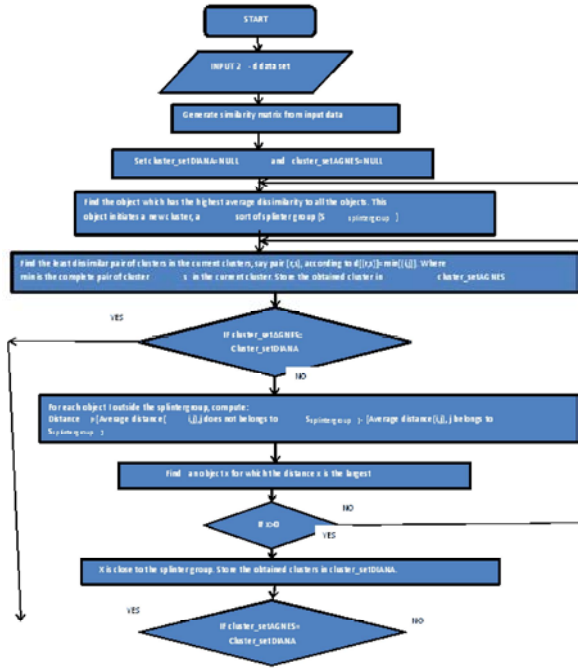

Fig. 5: Flow diagram of proposed algorithm

Working example of Proposed algorithm

Step 1: Distance matrix generated from the data-set of 6 points taken in previous examples and average dissimilarity to other objects will be as follows. In the first step, the algorithm has to split up the dataset into two clusters. This is not done by considering all possible divisions but rather by means of a kind of iterative procedure.

Distance	A	B	C	D	E	F
A	0	O.71	5.66	3.61	4.24	3.2
B	0.71	0	4.95	2.92	3.54	2.5
C	5.66	4.95	0	2.24	1.41	2.5
D	3.61	2.92	2.24	0	1	0.5
E	4.24	3.54	1.41	1	0	1.12
F	3.2	2.5	2.5	0.5	1.12	0

Step 2: After execution of step-1, further calculations will be performed as follows:

Object	Average dissimilarity to other objects
A	(0.71+5.66+3.61+4.24+3.20)/5=3.484
B	(0.71+4.95+2.92+3.54+2.50)/5=2.924
C	(5.66+4.95+2.24+1.41+2.50)/5=3.352
D	(3.61+2.92+2.24+1.00+0.50)/5=2.054
E	(4.24+3.54+1.41+1.00+1.12)/5=2.262
F	(3.20+2.50+2.50+0.50+1.12)/5=1.964

So object A is chosen to initiate the so-called splinter-group. At this stage we have the groups {A} and {B, C, D, E, F}. For each object of the larger group we compute the remaining objects and compare it to the average dissimilarity with the objects of splinter group.

Distance	A	B	C	D	E	F
A	0	O.71	5.66	3.61	4.24	3.2
B	0.71	0	4.95	2.92	3.54	2.5
C	5.66	4.95	0	2.24	1.41	2.5
D	3.61	2.92	2.24	0	1	0.5
E	4.24	3.54	1.41	1	0	1.12
F	3.2	2.5	2.5	0.5	1.12	0

From the above matrix it is observed that minimum distance is among the points D and F. So, object d and f will be combined to make clusters as {A, B, C, E} and {D, F} and distance matrix will take the form as follows-

Distance	A	B	C	D,F	E
A	0	O.71	5.66	3.2	4.24
B	0.71	0	4.95	2.5	3.54
C	5.66	4.95	0	2.24	1.41
D, F	3.2	2.5	2.24	0	1
E	4.24	3.54	1.41	1	0

Step 3: After execution of step-2 cluster_setAGNES will contain clusters as {A}, {B}, {C}, {E} and {D,F} and cluster_setDIANA will contain cluster {A}, {B, C, D, E, F}and further distances will be calculated as follows.

Distance	A	B	C	D,F	E
A	0	O.71	5.66	3.2	4.24
B	0.71	0	4.95	2.5	3.54
C	5.66	4.95	0	2.24	1.41
D, F	3.2	2.5	2.24	0	1
E	4.24	3.54	1.41	1	0

Thus objects {a, b} will be combined in next step and cluster_setAGNES will contain clusters {a, b} {c},{e},{d, f} and cluster_setDIANA will contain {a, b} and {c, d, e, f}.

Object	Average distance	Average distance to splinter group	Difference
B	(4.95+2.92+3.54+2.50)/4=3.4775	0.71	2.7675
C	(4.95+2.24+1.41+2.50)/4=2.775	5.66	-2.885
D	(2.92+2.24+1.00+0.50)/4=1.665	3.61	-1.945
E	(3.54+1.41+1.00+1.12)/4=1.7675	4.24	-2.4725
F	(2.50+2.50+0.50+1.12)/4=1.655	3.20	-1.545

Step 4: After execution of step-3 objects {a, b} will be combined in next step and cluster_setAGNES will contain clusters {a, b} {c},{e},{d, f} and cluster_setDIANA will contain {a, b} and{c, d, e, f} and further distances will be calculated as follows.

Object	Average distance	Average distance to splinter group	Difference
C	(2.24+1.41+2.50)/3=2.05	(0.71+5.66)/2=3.185	-1.135
D	(2.24+1.00+0.50)/3=1.2466	(3.61+2.92)/2=3.265	-2.01884
E	(1.41+1.00+1.12)/3=1.1766	(4.24+3.54)/2=3.89	-2.7134
F	(2.50+0.50+1.12)/3=1.8733	(3.20+2.50)/2=2.85	-1.4767

All negative. Therefore, to find the rebel to start the splinter-group with, we have

Object	Average distance
C	(2.24+1.41+2.50)/3=2.05
D	(2.24+1.00+0.50)/3=1.2466
E	(1.41+1.00+1.12)/3=1.1766
F	(2.50+0.50+1.12)/3=1.3733

So the object C is chosen to initiate another splinter-group. At this stage, we have groups {a,b},{c} and {d,e,f}. For each object of the larger group we compute the remaining objects and compare it to the average dissimilarity with the objects of splinter group.

We can see that the closest distance between clusters happens between cluster E and (D, F) at distance 1.00. Thus, we cluster them together into cluster ((D, F), E).

Dist	A,B	C	D,F	E
A,B	0	4.95	2.5	3.54
C	4.95	0	2.24	1.41
D, F	2.5	2.24	0	1
E	3.54	1.41	1	0

After execution of this step, cluster_setAGNES is equal to cluster_setDIANA, so algorithm will be terminated. In the below figure red line indicates the convergence point of proposed algorithm [19]. In the Figure 6, it shows the results obtained by proposed algorithm.

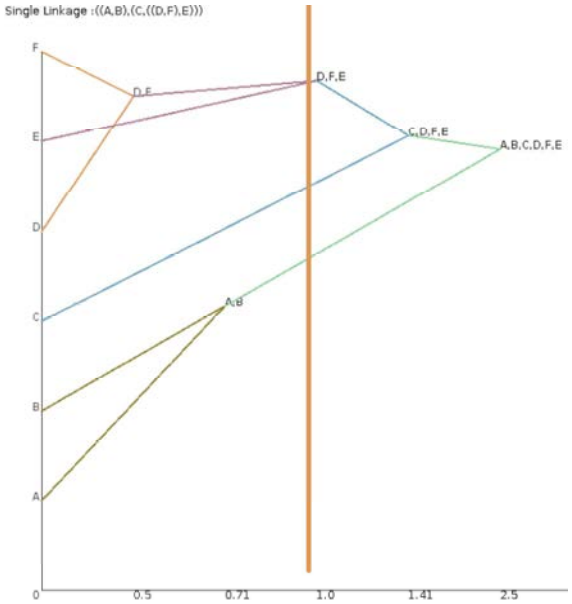

Fig. 6: Proposed Algorithm result

RESULTS

The proposed algorithm is experimented on the data about the star group denoted by CYG OBI, which consists of 47 stars in the direction of Cygnus. This data set was taken from [12]. The variable x denotes the logarithm of star's surface temperature (as measured from its spectrum) and y is the logarithm of its light intensity. In astronomy it is common practice to draw the so-called Hertzprung-Russel diagram, which is simply a scatter plot of y versus x (but note that x is plotted from left to right). The regions of the Hertzprung-Russel diagram are well known [13, 14]. The 43 stars belong to the so-called main sequence, whereas the four remaining ones are giant stars. The execution of DIANA gives the result same as Figure 8, in which the first step distinguishes the main sequence from the giants. In the subsequent steps, the algorithm then splits up both clusters until all the points are isolated. When the data is standardized, DIANA finds the right clusters in the first step.

S.No.	X	Y
1	4.37	5.23
2	4.56	5.74
3	4.26	4.93
4	4.56	5.74
5	4.3	5.19
6	4.46	5.46
7	3.84	4.65
8	4.57	5.27
9	4.26	5.57
10	4.37	5.12
11	3.49	5.73
12	4.43	5.45
13	4.48	5.42
14	4.01	4.05
15	4.29	4.26
16	4.42	4.58
17	4.23	3.94
18	4.42	4.18
19	4.23	4.18
20	3.49	5.89
21	4.29	4.38
22	4.29	4.22
23	4.42	4.42
24	4.49	4.85
25	4.38	5.02
26	4.42	4.66
27	4.29	4.66
28	4.38	4.9
29	4.22	4.39
30	3.48	6.05
31	4.38	4.42
32	4.56	5.1
33	4.45	5.22
34	3.49	6.29
35	4.23	4.34
36	4.62	5.62
37	4.53	5.1
38	4.45	5.22
39	4.53	5.18
40	4.43	5.57
41	4.398	4.62
42	4.45	5.06
43	4.5	5.34
44	4.45	5.34
45	4.55	5.54
46	4.45	4.98
47	4.42	4.5

The scatter plot of the above dataset of stars can be plotted as represented as represented in below graph.

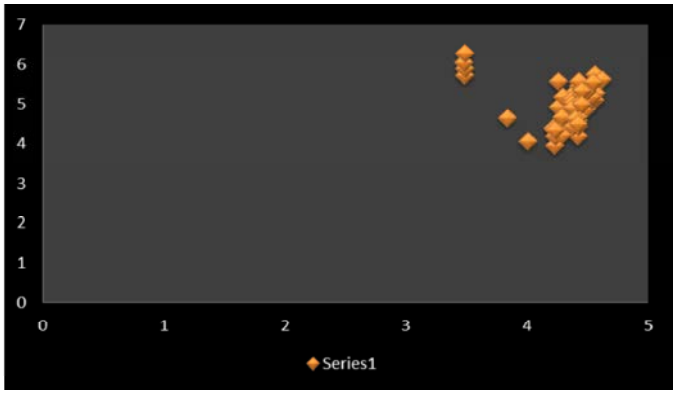

Fig. 7: Scatter plot of the data set

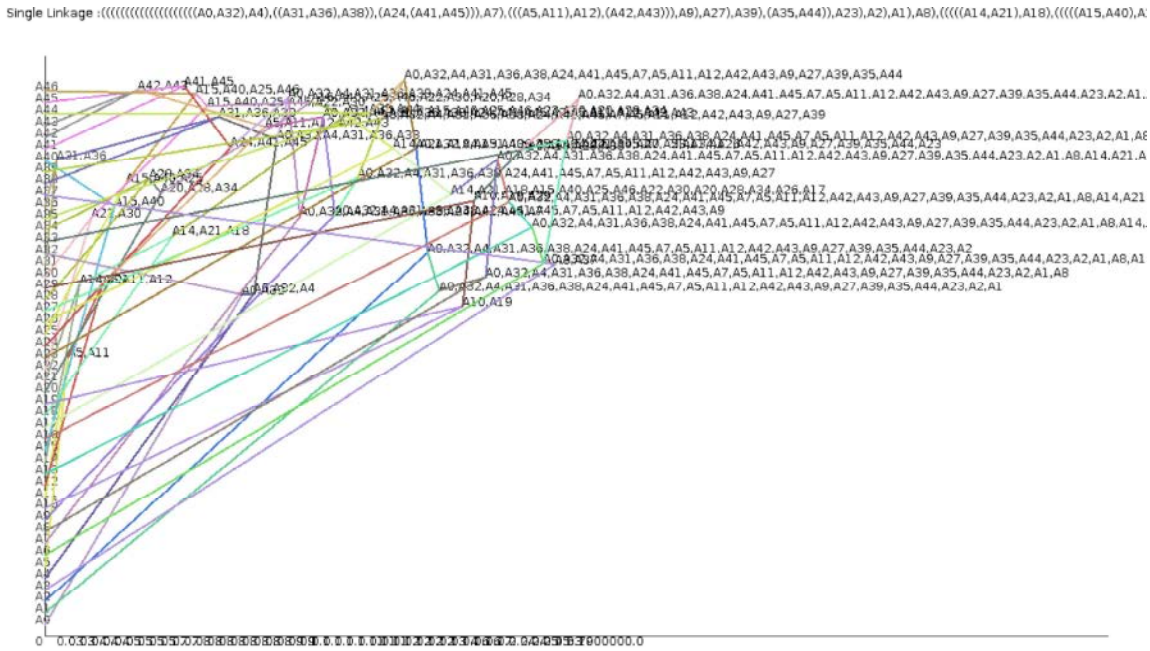

Fig. 8: Clusters obtained after implementation of proposed algorithm

Proposed algorithm is implemented in JAVA language and results are also computed using the same.

Comparison among Agnes, Diana and Proposed Algorithm

S.No.	Parameters	AGNES	DIANA	AGGLO-DIVISIVE
1	Approach used	Follows bottom-up approach. In the subsequent steps, the atomic clusters are merged into bigger and bigger clusters	Follows top-down approach. Subsequently, subdivide the cluster into reduced and more reduced clusters.	Follows hybrid approach, Two sets are maintained, one for cluster from top-down approach and second for clusters from bottom-up approach. Subsequently, both approaches are used
2	Convergence time	Converges slowly.	Convergence time is same as AGNES.	Converges fast by almost double time
3	Combinations to split/merge	n(n-1)/2 combinations for merging.	$2^{(n-1)}$-1 combinations are available for splitting, which is much larger than AGNES.	All the combinations of AGNES and DIANA are available i.e. n(n-1)/2 for merging and $2^{(n-1)}$-1 for splitting.
4	Complexity	$O(n^2.logn)$	$O(n^2.logn)$	$O((n^2.logn)/2)$
5	Memory and computations	Requires more memory and less computation.	Requires less memory but more computation.	Less memory as compared to AGNES and less computations as compared to DIANA.

Business A pplications of Proposed Algorithm: Agglo-divisive

Greedy Matching Application: Suppose that each member of a set of n applicants ranks a subset of my posts in strict order of priority [15,16]. A matching set of (post, applicant) pair such that each applicant and each post appear in at most one pair. A greedy matching is the matching in which the maximum possible number of applicants is matched to their first choice post and subject to that circumstance, then the maximum possible number is opposed to their second choice post and so along. This is a significant concept in any practical matching situation where the priorities are entirely at one side of the marketplace. A greedy matching can be performed by a transmutation of the classical problem of maximum weight bi-patient matching. Nevertheless, an exponentially decreasing sequence of weights must be set apart to the entries in each priority list and this adversely affects the complexity of the algorithm.

The proposed algorithm can also be used in greedy matching applications like above with big effects.

Travelling Salesman Heuristic Application: The travelling salesman problem is a popular optimization problem [17, 18]. Optimization solution for small instances can be ground in reasonable time by linear programming. Nevertheless, since travelling salesman is NP-hard, it will be very time consuming to solve larger instances with guaranteed optimality. The proposed algorithm can be very efficiently used to solve larger instances of the problem in reduced time.

Conclusion and Future Work: The proposed algorithm facilitates to have the benefits of AGNES and DIANA in the single algorithm. At the same time, the proposed algorithm minimizes the overlap time and reduced the complexity of algorithm by 50%. At each step, we have obtained all the clusters from AGNES and DIANA. Since, the proposed algorithm is based on the hybrid approach, it is more applicable in approximately all the scenarios whereas, AGNES and DIANA are applicable individually in selected scenarios. AGNES hierarchical clustering algorithm can be applied in the place where the deductive approach is required and DIANA hierarchical clustering is applicable where the inductive approach is needed. Unlikely, the proposed algorithm can be applied in both scenarios. The proposed algorithm is better in the sense that it reduces the execution time and provides better results with greater flexibility and the proposed algorithms converges faster than other hierarchical algorithm. It uses time for n (n-1)/2 for merging and $2^{(n-1)}$-1 for splitting. The proposed algorithm also requires less computation and memory as compared to existing algorithms under the same category. It cuts down the pace count and computations to approximately 50%.

Further, these algorithms (hierarchical algorithms) can also be implemented using average linkage, median and maximum linkage. After execution, the comparative performance can be valued for all the algorithms.

REFERENCES

1. King, B., 1967. Step-wise clustering procedures. Journal of the American Statistical Association, 69: 86.
2. Daniel Boley, G., 1998. Principal direction divisive partitioning. Data Mining and Knowledge Discovery, 2(4): 325-344, December.
3. Vijaya, P.A., M. Narasimha Murty and D.K. Subramanian, 2005. An efficient hybrid hierarchical agglomerative clustering (HHAC) technique for partitioning large data sets. In PReMI, Lecture Notes in Computer Science, pages 583{588, Berlin, Heidelberg, Springer Berlin Heidelberg.
4. Eui-Hong Han and George Karypis. 2000. Centroid-based document classification: Analysis and experimental results.
5. Douglass R. Cutting, David R. Karger, Jan O. Pedersen and John W. Tukey, Scatter/Gather, 1992. A cluster-based approach to browsing large document collections. In Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '92, pages 318{329, New York, New York, USA, ACM Press.
6. Taher Niknam, Bahman Bahmani Firouzi and Majid Nayeripour, 2008. An efficient Hybrid Evolutionary algorithm for Cluster Analysis, World Applied Sciences Journal, 4(2): 300-307, ISSN 1818-4952.
7. Brian S. Everitt, Sabine Landau and Morven Leese, 2001. Cluster Analysis, volume 33 of Social Science Research Council Reviews of Current Research. Arnold.
8. Karypis, G., E.H. Han and V. Kumar, 1999. Chameleon: A hierarchical clustering algorithm using dynamic modeling. IEEE Computer, 32(8).
9. Rui Xu and Donald Wunsch, 2005. Survey of clustering algorithms. IEEE Transactions on Neural Networks, 16(3).
10. Estivill-Castro, V. and I. Lee, 2000. AUTOCLUST: Automatic Clustering via Boundary Extraction for Mining Massive Point-Data Sets, 5th Int'l Conf. on Geocomputation, Geo Computation CD-ROM: GC049, ISBN 0-9533477-2-9.
11. Xiong Hui, Wu Junjie and Chen Jian, 2009. K-Means clustering versus validation measures: A data-distribution perspective. IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics), 39(2).
12. Finding Groups in Data: An Introduction to cluster Analysis- Leonard Kaufman, Peter J. Rousseeuw.
13. Hambrusch, S.E., C.M. Liu and H.S. Lim, 2000. Clustering in Trees: Optimizing Cluster Sizes and Number of Subtrees, Journal of Graph Algorithms and Applications, 4(4): 1-26.
14. Quigley, A. and P. Eades, 2001. FADE: Graph Drawing, Clustering and Visual Abstraction, Proc. GD'2000, LNCS, pp: 197-210.
15. Harel, D. and Y. Koren, 2001. A Fast Multi-scale Method for Drawing Large Graphs, Proc. GD'2000, LNCS, pp: 183-196.
16. Hamerly, G. and C. Elkan, 2003. "Learning the k in k-Means," Proc. 17th Ann. Conf. Neural Information Processing Systems (NIPS '03).
17. Gath, I. and A. Geve, 1989. "Unsupervised Optimal Fuzzy Clustering," IEEE Trans. Pattern Analysis and Machine Intelligence, 11(7): 773-781.
18. Agarwal, P.K. and C.M. Procopiuc, 1998. Exact and Approximation Algorithms for Clustering, Proc. 9th ACM-SIAM Symp. Discrete Algorithms.
19. May-Six, J. and I.G. Tollis, 2001. Effective Graph Visualization Via Node Grouping, Proc. IEEE Symposium on information Visualization, pp: 51-58.

20. May-Six, J., 2000. Vistool: A Tool For Visualizing Graphs, PhD Thesis, The University of Texas at Dallas.
21. Radha Chitta and M. Narasimha Murty, 2010. Two-level k-means clustering algorithm for k{relationship establishment and linear time classification. Pattern Recognition, 43(3).
22. Batagelj, V., A. Mrvar and M. Zaversnik, 2000 Partitioning Approaches to Clustering in Graphs, Proc. GD'1999, LNCS, pp. 90-97.