Agglo-Divisive Approach of Clustering

Archana Singh, Avantika Yadav

Amity Institute of Information Technology, Amity University , Noida, India
Amity School of Engineering and Technology, Amity University, Noida, India

Summary
Clustering is the process of grouping the 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.

Keywords
clustering, hierarchical clustering, linkages, similarity matrix

1. Introduction

The Clustering is process of forming group of data-items of similar type based on some similarity measure. For 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 [3]. Hierarchical clustering is a technique of cluster analysis which is used to build a hierarchy of clusters [8]. 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 [13].

A key component of the 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 dendogram [3, 8] (the dendogram 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 [8] 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 [3, 7, 8, 13]. In general, the agglomerative methods are mostly 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 this paper, we have proposed an algorithm which is a hybrid approach using the concept of AGNES (agglomerative approach) and DIANA (divisive approach) algorithm. The algorithm provides all the results obtained from AGNES and DIANA at each steps. Proposed algorithm combines the benefit of both algorithms.

2. Literature survey

In the research paper, Step-wise clustering procedures written by B.King, the author has described simple step-wise procedure for clustering is discussed. According to him, 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[3]. In the paper, cluster based approach to browsing large document collections, Douglass R. Cutting with David. R. Karger, Jan. O. Pedersen and John W. Tukey has discussed that problem with document clustering only when clustering is used in an attempt to improve conventional search techniques [4]. In this paper document browsing techniques has been presented. In the paper, Principal direction divisive partitioning, Daniel Boley has proposed a new algorithm that is capable of partitioning a set of documents or other samples based on an embedding in a high dimensional Euclidean scope using divisive approach [8]. In the proposed approach, the documents are assembled into a matrix which is very sparse and in this algorithm sparsity provides efficiency. Brian S Everitt, Sabine Landau, and Morven Leese discussed more about Cluster Analysis in volume 33 of Social Science Research Council Reviews of Current Research. Arnold, 2001 including dendogram, linkages and similarity matrices[2]. Eui-Hong Han and George Karypis proposed Centroid-based document classification: Analysis and experimental results, 2000 in his work[6]. This document explains the document classication process using the Centroids and distance metrics.

Manuscript received December 5, 2014
Manuscript revised December 20, 2014

3. Organization of paper

The organization of the entire paper is as follows- section-I gives the introduction about what is clustering, hierarchical clustering and small introduction about AGNES, DIANA and linkages and similarity matrix. In the section-II literature survey is discussed. Section-IV gives the brief introduction of AGNES algorithm with example, section-V gives brief introduction about DIANA and section-VI summarizes the proposed algorithm. Section-VII gives the comparative study of three algorithms i.e. AGNES, DIANA and AGGLO-DIVISIVE. Section-VIII concludes all the applications of proposed algorithm and future scope. Section-IX is the conclusion of overall work.

4. Agglomerative clustering: agnes

A. Algorithm

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

4.1 Example (AGNES)

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

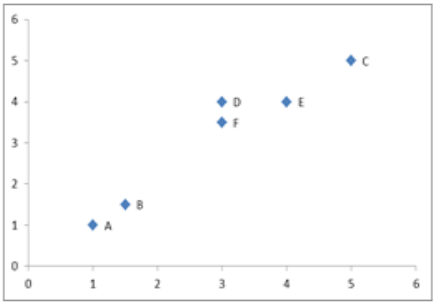

Fig: 1. Scatter Plot

Distance matrix-D0

Min Distance (Single Linkage)

Table: 2. Matrix D0

Dist	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

Min Distance (Single Linkage)

Table: 3. Matrix D1

	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.

Table: 4. Matrix D1

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

The cluster A and cluster B is grouped into a single cluster name (A, B).

Min Distance (Single Linkage)

Table: 5. Matrix D2

Dist	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)-->A=min(d(DA, dFA)=min(3.61,3.20)=3.20

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

d(D,F)→B=min(d(DB, dFB)=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)

Table: 6. Matrix D3

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

***Distance matrix*-D4**
Min Distance (Single Linkage)

Table: 7. Matrix D4

Dist	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 $d_{((D,F),E)} \rightarrow_{(A,B)} = \min_{(dDA,dDB,dFA,dFB)}$
$= \min_{(3.61,2.92,3.20,2.5,4.24,3.54)} = 2.50$

Distance matrix- D5
Min Distance (Single Linkage)

Table: 8. Matrix D5

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

$d_{(((D,F),E),c)} \rightarrow_{(A,B)} = \min(d_{DA}, d_{DB}, d_{FA}, d_{FB}, d_{EA}, d_{EB}, d_{CA}, d_{CB})$
$d_{(((D,F),E),c)} \rightarrow_{(A,B)} = \min(3.61,2.92,3.20,2.5,4.24,3.54,5.66,4.95) = 2.50$

4.2 RESULTS (AGNES)

1. In the beginning we have clusters as : A, B, C, D, E and F.
2. Clusters D and F are merged into cluster (D, F) at distance 0.50
3. Clusters A and cluster B are merged into (A, B) at distance 0.71
4. Clusters E and (D, F) are merged into ((D, F), E) at distance 1.00
5. Clusters ((D, F), E) and C are merged into (((D, F), E), C) at distance 1.41
6. Clusters (((D, F), E), C) and (A, B) are merged into ((((D, F), E), C), (A, B)) at distance 2.50
7. In the last step, cluster contain all the objects, thus terminate the computation.

The hierarchy is given as (((D, F), E),C), (A,B).

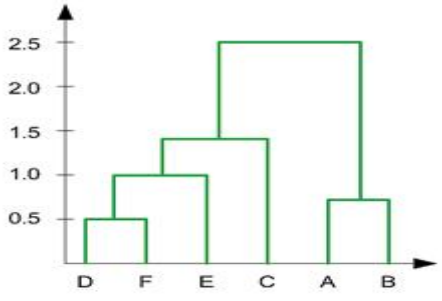

Figure: 2. Dendogram AGNES

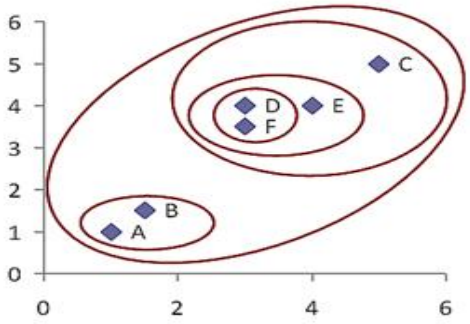

Figure: 3. Clusters in XY space

4.3 Limitations

- Main limitations of agglomerative clustering methods are[13]:
- They do not scale well: time complexity of at least O(n2), where n denotes the number of total objects;
- The actions performed in previous steps can't be undone.

5. Divisive clustering: diana

Algorithm

1. Begin with the single cluster having level L(0) = n and sequence number m = 0.
2. Find the most 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 cluster.
3. Increment the sequence number: m = m +1. Split the cluster into clusters (r) and (s) to form the next cluster. Make the level of this clustering to
   L(m1) = d[(r)] and L(m2)=d[(s)]
4. Update the similarity matrix, D, by adding the rows and columns corresponding to clusters (r) and (s) and deleting a row and column corresponding to the newly formed cluster. The similarity between the new cluster, denoted r and s and old cluster (k) is defined in this way:
   d[(k), (r,s)] = max d[(k),(r)], d[(k),(s)]
   If all objects are in distinct clusters, stop. Else, go to step 2.

5.1. Example (DIANA)

Distance matrix-D0
Min Distance (Single Linkage)

Table: 9. Matrix D0

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

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

Distance matrix-D1
Min Distance (Single Linkage)
$d_{(((D,F),E),c)\rightarrow(A,B)}=\min(3.61,2.92,3.20,2.5,4.24,3.54,5.66,4.95)=2.50$

Table: 10. Matrix D1

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

Distance matrix-D2
Min Distance (Single Linkage)

Table: 11. Matrix D2

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

Distance matrix-D3
Min Distance (Single Linkage)

Table: 12. Matrix D3

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-D4
Min Distance (Single Linkage)

Table: 13. Matrix D4

Dist	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-D5
Min Distance (Single Linkage)

Table: 14. Matrix D5

Dist	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

5.2. Results(DIANA)

1. In the beginning we have single cluster as:(((D, F), E), C), (A, B)) .
2. Cluster :(((D, F), E), C), (A, B)) is split into clusters (((D, F), E), C) and (A, B) .
3. Cluster (((D, F), E),C) is split into ((D, F), E) and (cluster C) at distance 1.41
4. Cluster ((D, F), E) is split into (D, F) and (cluster E) at distance 1.00
5. Cluster (A, B) is split into cluster A and cluster B into at distance 0.71
6. Cluster (D, F) is split into D and F at distance 0.50
7. In the end we have single-single object in all clusters: (A, B, C, D, E, F).
8. The last clusters contain single object, thus terminate.

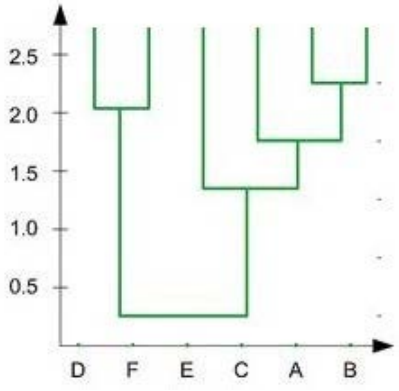

Figure: 4. Dendogram DIANA

6. Hybrid clustering: agglo-divisive

ALGORITHM: AGGLO-DIVISIVE

1. Begin with the single cluster having level L1(0) = n and sequence number m1 = 0 and Begin with the single cluster having level L2(0) = n and sequence number m2 = 0.
2. Find the most dissimilar pair of clusters in the current clusters, say pair (r), (s), according to d2[(r),(s)] = min d2[(i),(j)]
   where the min is complete pairs of clusters in the current clusters, and
   Find the least dissimilar pair of clusters in the current clustering, say pair (r), (s), according to
   d1[(r),(s)] = min d1[(i),(j)]
   where the min is complete pairs of clusters in the current clustering.
3. Increment the sequence number: m = m +1. Split the cluster into clusters (r) and (s) to form the next cluster. Make level of this clustering to L2(m11) = d2[(r)] and L2(m12)=d2[(s)] and if L1!=L2 then
   L1(m1) = d1[(r)] and L1(m2)=d1[(s)]
4. Update the similarity matrix, D, by adding the rows and columns corresponding to clusters (r) and (s) and deleting a row and column corresponding to the newly formed cluster. The similarity between the new cluster, denoted r and s and old cluster (k) is defined in this way:
   d2[(k), (r,s)] = min d2[(k),(r)], d2[(k),(s)]
   And, Update the similarity matrix, D, by adding the rows and columns corresponding to clusters (r) and (s) and deleting a row and column corresponding to the newly formed cluster. The similarity between the new cluster, denoted r and s and old cluster (k) is defined in this way:
   d1[(k), (r,s)] = max d1[(k),(r)], d[(k),(s)]
5. If L1=L2 merge L1 and L2 to generate L, otherwise repeat from step 2.

6.1. Example (AGGLO-DIVISIVE)

Step-1

Table 15: Step1 of Agglo-divisive

Dist	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

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

Step-2

Table: 16 Step2 of agglo-divisive

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

Dist	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

7. Comparison among agnes, diana and proposed algorithm

S.No.	AGNES	DIANA	AGGLO-DIVISIVE
1.	Follows bottom-up approach.	Follows top-down approach.	Follows hybrid approach
2.	Converges slowly.	Convergence time is same as AGNES.	Converges fast.
2.	Time complexity is O(n2)	Time complexity is O(n2)	Time complexity is O(n2)
3.	In the beginning, all the objects are in different clusters.	In the beginning, all the objects belong to single cluster.	Two sets are maintained, one for cluster from top-down approach and second for clusters from bottom-up approach.
4.	Then we merge these atomic clusters into bigger and bigger clusters.	We then subdivide the cluster into reduced and reduced clusters.	Both the steps are performed.

8. Applications of proposed algorithm : agglo-divisive

8.1 Greedy matching application.

Suppose that each member of a set of n applicants rank a subset of m posts in strict order of priority. A matching is set of (post, applicant) pair such that each applicant and each post appears in at most one pair. A greedy matching is the matching in which the maximum possible number of applicants are matched to their first choice post, and subject to that condition, then the maximum possible number are matched to their second choice post and so on. This is an important concept in any practical matching situation where the priorities are only at one side of the market. A greedy matching can be performed by a transformation to the classical problem of maximum weight bi-partite matching. However, an exponentially decreasing sequence of weights must be assigned 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 great results.

8.2 Travelling salesman heuristic application.

Travelling salesman problem is a popular optimization problem. Optimization solution to small instances can be found in reasonable time by linear programming. However, 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

Proposed algorithm provides the facility to have the benefits of AGNES and DIANA in the single algorithm. At the same time, the proposed algorithm minimizes the convergence time. At each step, we have all the clusters which are obtained from AGNES and DIANA. Proposed algorithm is applicable in all the scenarios where not only AGNES is applicable but also in all the scenarios where DIANA is applicable.
AGNES hierarchical clustering algorithm can be used in the situation where deductive approach is required and DIANA hierarchical clustering is applicable where inductive approach is required. Were as, proposed algorithm can be used for both scenarios. The proposed algorithm is better in the sense that it reduces the execution time and provides better results with greater flexibility.
In this area, there is scope of future work like these algorithms hierarchical algorithms) can also be implemented using average linkage and maximum linkage. After implementation, the comparative performance can be measured for all the algorithms.

References

[1] A. Quigley and P. Eades, FADE: Graph Drawing, Clustering, and Visual Abstraction, Proc. GD'2000, LNCS, pp. 197-210 (2001).

[2] Brian S Everitt, Sabine Landau, and Morven Leese. Cluster Analysis, volume 33 of Social Science Research Council Reviews of Current Research. Arnold, 2001.

[3] B King. Step-wise clustering procedures. Journal of the American Statistical Association, 69:86{101, 1967.

[4] Douglass R. Cutting, David R. Karger, Jan O. Pedersen, and John W. Tukey, Scatter/Gather: 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, June 1992. ACM Press.

[5] D. Harel and Y. Koren, A Fast Multi-scale Method for Drawing Large Graphs, Proc. GD'2000, LNCS, pp. 183-196 (2001).

[6] Eui-Hong Han and George Karypis. Centroid-based document classification: Analysis and experimental results, 2000.

[7] G Karypis, E H Han, and V Kumar. Chameleon: A hierarchical clustering algorithm using dynamic modeling. IEEE Computer, 32(8):68{75, 1999}.

[8] G. Daniel Boley. Principal direction divisive partitioning. Data Mining and Knowledge Discovery, 2(4):325{344, December 1998.

[9] G. Hamerly and C. Elkan, "Learning the k in k-Means," Proc. 17th Ann. Conf. Neural Information Processing Systems (NIPS '03), Dec 2003.

[10] I. Gath and A. Geve, "Unsupervised Optimal Fuzzy Clustering," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 11, no. 7, pp. 773-781, July 1989.

[11] J. May-Six and I. G. Tollis, Effective Graph Visualization Via Node Grouping, Proc. IEEE Symposium on information Visualization 2001, pp. 51-58 (2001).

[12] J. May-Six, Vistool: A Tool For Visualizing Graphs, PhD Thesis, The University of Texas at Dallas (2000).

[13] P.A. Vijaya, M. Narasimha Murty, and D.K. Subramanian. 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, 2005. Springer Berlin Heidelberg.

[14] P. K. Agarwal and C. M. Procopiuc, Exact and Approximation Algorithms for Clustering, Proc. 9th ACM-SIAM Symp., Discrete Algorithms (1998).

[15] Radha Chitta and M. Narasimha Murty. Two-level k-means clustering algorithm for k{relationship establishment and linear time classification. Pattern Recognition, 43(3):796{804, March 2010.

[16] Rui Xu and Donald Wunsch. Survey of clustering algorithms. IEEE Transactions on Neural Networks, 16(3):645{678, 2005.

[17] S. E. Hambrusch, C-M. Liu, and H-S. Lim, Clustering in Trees: Optimizing Cluster Sizes and Number of Subtrees, Journal of Graph Algorithms and Applications, Vol. 4, No. 4, pp.1-26 (2000).

[18] V. Batagelj, A. Mrvar, and M. Zaversnik, Partitioning Approaches to Clustering in Graphs, Proc. GD'1999, LNCS, pp. 90-97 (2000).

[19] V. Estivill-Castro and I. Lee, 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 (2000).

[20] Xiong Hui, Wu Junjie, and Chen Jian. K-Means clustering versus validation measures: A data-distribution perspective. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 39(2):318{331, April 2009.

Archana Singh in AIIT department at Amity University, Noida. She has received MCA and M.Tech degrees and currently pursuing Ph.D from Amity University. Her research area includes data mining, algorithms, big-data and Fuzzy and provides guidance to various research scholars.

Avantika Yadav is working as Asst. Professor at Krishna Engineering College, Ghaziabad. She has received MCA degree in 2006 with honors from UPTU, India and currently she is pursuing M.Tech-CSE from Amity University, Noida, India.
Her research area includes data mining algorithms, neural networks, artificial intelligence and database.