Comparative Analysis of Clustering Methods

Abhineet Saxena
Computer Science & Engineering Department,
G.B. Pant Govt. Engineering College,
Guru Gobind Singh Indraprastha University,
New Delhi, India

Mamta Mittal
Computer Science & Engineering Department,
G.B. Pant Govt. Engineering College,
Guru Gobind Singh Indraprastha University,
New Delhi, India

Lalit Mohan Goyal
Computer Science & Engineering Department,
Noida Institute of Engineering & Technology,
Uttar Pradesh Technical University (UPTU),
Uttar Pradesh, India

ABSTRACT
The innumerable clustering methods which exist today form the basis of Data Mining and Cluster Analysis. This paper details the distinct classifications of clustering methods, describes prominent examples for each such classification and aims to bring about the comparison between the primary clustering techniques which form the basis of all the others, i.e. the Hierarchical and Partitional algorithms.

General Terms
Cluster Analysis, Data Mining, Classification

Keywords
Clustering, Data Mining, Partitional, Hierarchical

1. INTRODUCTION: WHAT IS CLUSTERING?
Clustering refers to the mathematical and algorithmic process of grouping 'n' elements into 'k' distinct groups such that the groups so produced have more similarity among its members than among members belonging to different, distinct groups. Here, similarity implies a similarity measure, such as Euclidian distance.

It is unsupervised, i.e. groupings of objects are produced not according to some pre-specified labels but naturally, depending upon the similarity measure chosen. As an example, clustering a population distribution depending upon risk factor for viral epidemic.

1.1 How does clustering figure as part of data mining?
Data Mining refers to the automated and scalable analysis of massive data sets wherein extraction of useful information and patterns is the aim. The extraction process is referred to as Knowledge Discovery from Data (KDD), as explained in Fig.1.

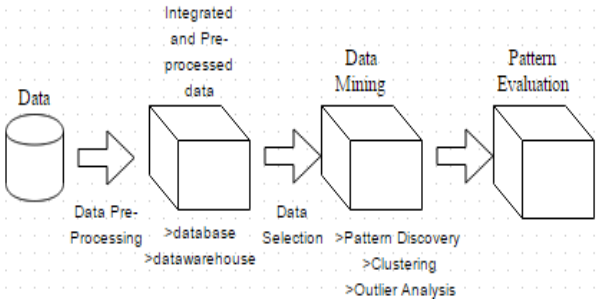

Fig 1. The Knowledge Discovery (KDD) Process explained. The need of clustering during the Data Mining phase arises in the grouping and classification of similar patterns.

Clustering helps in identifying groups of patterns which share similar characteristics but this regularity in the database may not be apparent prior to the clustering operation. Hence clustering is an integral part of extracting useful information from transformed data as part of the Data Mining discipline.

But applications of clustering are not simply restricted to the domain of data mining.

1.1.1 CLUSTERING FROM AN ALGORITHMIC POINT OF VIEW
Cluster Analysis has led to the contribution of distinct clustering algorithms which vary widely on the basis of their domain of application and algorithmic performance.

Clustering algorithms find use in multiple disciplines such as Genetics (gene grouping and analysis), Image Processing and Machine Learning to name a few.

The analysis of performance and characteristics of clustering algorithms includes choosing the appropriate data-set, selecting the similarity measure, applying the algorithm and analyzing the results obtained by comparison to expected groupings or using validation techniques [13].

1.1.2 TYPES OF CLUSTERING ALGORITHMS
The multiple clustering algorithms that exist are classified into the following types based primarily on two criteria-how the clusters are constituted and the type of distribution supplied as input (discreet points or continuous).

1.1.2.1 Partitional Algorithms
Under partitional algorithms, the underlying philosophy remains to constitute k distinct clusters out of n distinct data points such that each data point is identified with at-least one of the clusters and that each cluster,

$$C_i \in C \text{ where } C \text{ is the set of all clusters}$$

contains at-least one data point(non-empty) and does not overlap with another cluster.

K-Means Clustering and Partitioning-Around-Medoids (PAM) figure as popular partitional algorithms.

K-Means Algorithm

K-Means is a highly customizable and versatile partitional clustering method which is iterative in nature. The standard implementation, as suggested by (MacQueen, 1967) [11] works by randomly selecting k initial centers, associating each data point to the nearest center thus constituting clusters, and finally updating the centers to new values by taking mean of all the cluster point dimension variables and using the updated center values in the next iteration, until a convergence criteria is met. The iterative process has been depicted in Fig. 2.

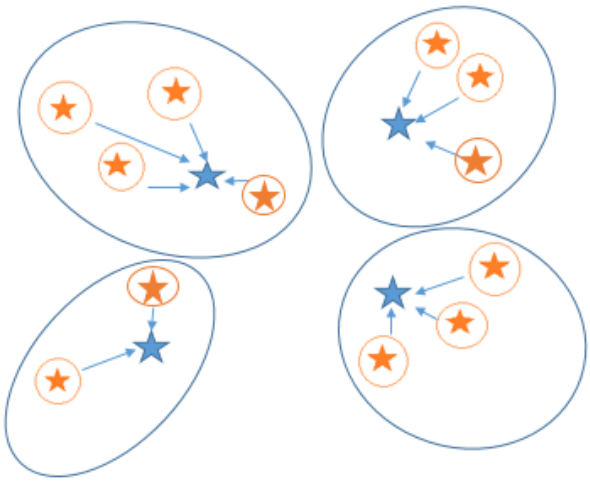

Figure 2. Visualization of how data points are being associated to the nearest cluster center.

Partitioning Around Medoids (PAM)

This algorithm as developed by (Kaufman and Rousseeuw, 1990) [10] makes use of a dissimilarity matrix. It works with Medoids, i.e. values representative of the whole cluster which it assumes initially. Its aim remains to minimize the overall dissimilarity between the representatives of each cluster and its members.

It's more robust than K-Means as it minimizes a sum of dissimilarities instead of a sum of squared Euclidian distances.

1.1.2.2. Hierarchical Algorithms

Hierarchical Clustering Algorithms operate by assigning each data point to a cluster of its own. Then by making use of the similarity measure, the two most similar clusters are identified. These are merged together, and this process is repeated until the required number of clusters have been generated or all the clusters become part of one cluster. Hence a tree hierarchy exists among all the clustered elements.

Hierarchical algorithms are either agglomerative or divisive. They are agglomerative in nature, if they employ a bottom-up approach to forming clusters. Beginning with distinct single element clusters, larger clusters are constituted by merging two clusters at each step. Fig. 3 depicts the process taking into consideration 5 sample points.

They are divisive if initially, a single, large, all-element-encompassing cluster exists which is decomposed at each step to produce the requisite clusters.

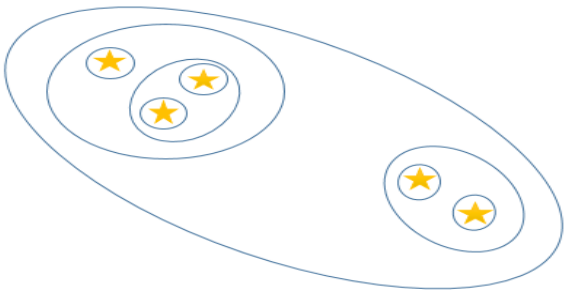

Figure 3. Visualization of agglomerative hierarchical clustering of 5 points. The data-points have been clustered hierarchically with Euclidian distance metric and centroid link similarity measure.

1.1.3 Probabilistic and Generation Model based Algorithms

Such algorithms work by assuming the given data set to be a sample of an underlying data distribution. This distribution can be continuous or may consist of discrete points.

The algorithm then attempts to construct a probabilistic distribution model for the dataset provided. If it's a parameterized distribution, then it tunes the parameters so that the closest fit to the given data distribution can be obtained.

Probability-based Clustering

In probability-based clustering, a data distribution is initially provided which is then subjected to clustering based on the premise that clusters contain objects which belong to the same range of values or distribution. Here, a probability distribution can be specified by either defining the characteristic function or the parameterized probability distribution function (density function for continuous distribution) such as one for Gaussian distribution is specified as follows:

$$f_X(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), x \in (-\infty, \infty)$$

$$\mu: mean, \sigma^2: variance$$

Under probability based clustering, Expectation Maximization (EM) algorithm is the most prominent one.

1.1.1.1 Expectation Maximization (EM) Algorithm

This algorithm was developed by (Arthur Dempster, 1977) [6]. In this iterative approach, two main steps are used, first being the Expectation step, wherein the logarithmic likelihood function is first constituted and second being the Maximization step, which computes the parameters maximizing the expected log-likelihood found in the first step.

1.2 Density-based and Grid-based Algorithms

Density-based Clustering

Density based algorithms work on the principle that clusters are so constituted, that they grow till the neighborhood density exceeds a certain threshold value, i.e. the number of elements lying within a maximum radius value becomes greater than a certain minimum value. It works on those distributions where density of data-points varies substantially in order for high and low density regions to be discernible.

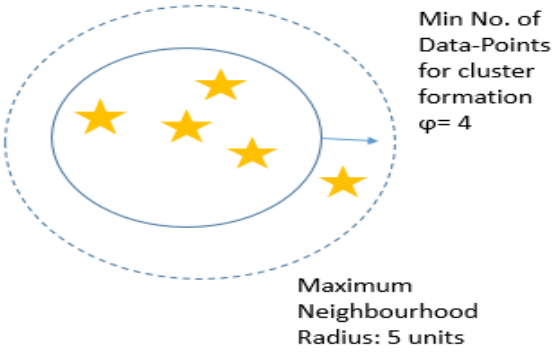

Figure 4. Visualization of Density based clustering with sample constraints for Min. points and maximum cluster radius being (4, 5 units) respectively.

Examples of such algorithms include DBSCAN (Density-based Spatial Clustering of Applications with Noise) and OPTICS (Ordering Points To Identify Clustering Structure) algorithms.

1.2.1.1 Density-based Spatial Clustering of Application with Noise (DBSCAN)

The algorithm as developed by (M.Ester et al., 1996) [12] has the ability to discover clusters of arbitrary shapes. It defines the concepts of Direct Density Reachability, Density Reachability, Density Connectedness, Core Points, Border Points and the Eps-Neighborhood of a point to introduce the notion of density in cluster analysis. It arbitrarily selects a point 'p', retrieves all the points density reachable from it and continues the process until all the points have been processed. It is immune to the potential distortions in observations produced due to outliers, as it effectively prunes out the outliers.

However, this method is sensitive to the variations of input parameters.

1.2.1.2 Ordering Points To Identify Clustering Structure (OPTICS)

The algorithm as developed by (Ankerst et al., 1999)[2] works on the premise that higher density clusters are completely contained within clusters of lower density and as such can be identified by defining further and extending upon the concepts specified for DBSCAN algorithm, namely Core Distance and Reachability Distance.

The ordering of points that it produces preserves the clustering structure information and can be used to construct reachability plots to identify density based cluster orderings corresponding to a broad range of parameter settings thus overcoming the drawback posed by the DBSCAN algorithm.

1.2.2 Grid-based Clustering

Grid-based methods partition data space into finite number of cells to form a grid structure. It helps in discretizing the data-space so that dense regions can be located though measuring the density of data points present within a cell.

It's an efficient and scalable method if the number of cells can be made very small in comparison to the number of data points being analyzed.

However, clustering high-dimensional data is particularly hard with the pre-defined cell sizes and density thresholds localizing its cluster identification capability.

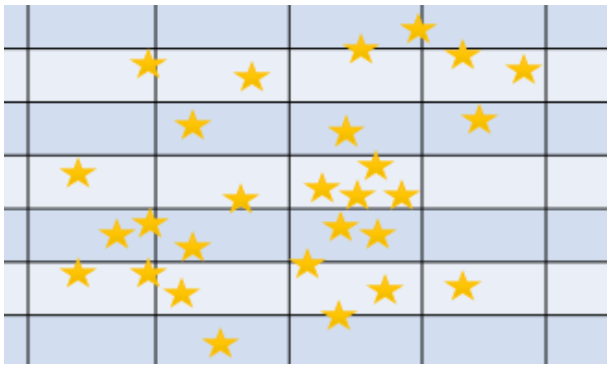

Figure 5. Visualization of a Grid-based partitioning over a 2D data space. The approach can be extended to higher dimensions with each cell storing greater details about its corresponding region.

STING and CLIQUE figure as important algorithms embodying this approach.

1.2.2.1 Statistical Information Grid Approach (STING)

This approach, as developed by (W. Wang et al., 1997) [17] divides a spatial area into rectangular cells at different levels of resolution. Conceptually, the multi-resolution data structure can be perceived to be multi-planar, with each plane storing details at a particular depth.

Here the cells have a hierarchical tree-structure with cells at a higher level storing information of smaller cells at the next lower level.

Each cell stores statistical information like mean, standard deviation, minimum value, maximum value and type of distribution.

It can be easily parallelized and the area-based queries that it caters to can be answered specifically. However, due to its probabilistic nature, a loss of accuracy can occur.

1.2.2.2 Clustering In Quest (CLIQUE)

This approach as developed by (R.Agrawal et al., 1998) [1] involves automatic identification of subspaces belonging to a high dimension data space. It generates minimal descriptions of all the dense regions in a subspace and makes use of Apriori principle to find potential candidates in the next higher-dimension space till all the dimensions have been covered in a level-wise maner.

Fuzzy Method Algorithms

The data points to be clustered can have the following membership association levels, i.e. exclusive or overlapped clustering.

Exclusive Membership-based Clustering

Exclusive or hard clustering is when data is grouped in an exclusive way, so that if a certain datum belongs to a definite cluster then it could not be included in another cluster. Hence distinct clusters are obtained. K-Means often produces clusters wherein the data-points have exclusive membership levels

Fig. 5 illustrates the concept as the members belonging to the two clusters produced, are completely distinct.

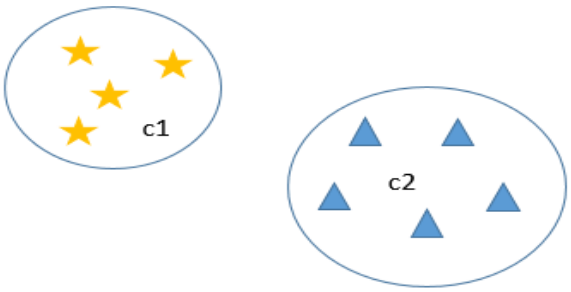

Figure 6. Visualization of Exclusive Membership-based Clustering.

Overlapped Membership-based Clustering

Overlapped clustering occurs when the membership criterion for each data-point is relaxed and it can be part of more than one cluster, as exemplified in Fig. 6 wherein two members are common to both the clusters.

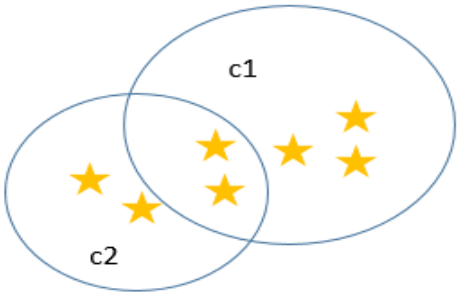

Figure 7. Visualization of Overlapped Membership-based Clustering.

Fuzzy-logic based or soft-clustering algorithms are based on the philosophy that each data element can belong to more than one clusters and hence associated with each data point, we have a set of membership levels. Here, a value indicates how strongly associated an element is with a particular cluster.

The most popular fuzzy algorithm is the Fuzzy C-means algorithm, refined by (James C. Bezdek et al., 1984) [4]. It involves iterative optimization of a given objective function, with the update of membership and cluster centers at each iteration.

2. COMPARISON BETWEEN HIERARCHICAL AND PARTITIONAL METHODS

Hierarchical and Partitioning based algorithms have subtle differences in their modus-operandi.

Hierarchical methods produce a nested series of partitions, while partition-based algorithms produce non-nested, distinct clusters.

Hierarchical clustering is often portrayed as the better quality clustering approach, but is limited in use because of its quadratic or cubic time complexity. In contrast, partitional methods, like k-Means and its variants have a time complexity that is linear in the number of data elements, but are thought to produce inferior clusters.

To establish the comparison between the two approaches, we perform quality and time-based analysis for standard implementations of both.

In order to point out the differences in the quality of clustering outputs produced, we consider here the Aggregation dataset, retrieved from (Gionis et al., 2007) [8].

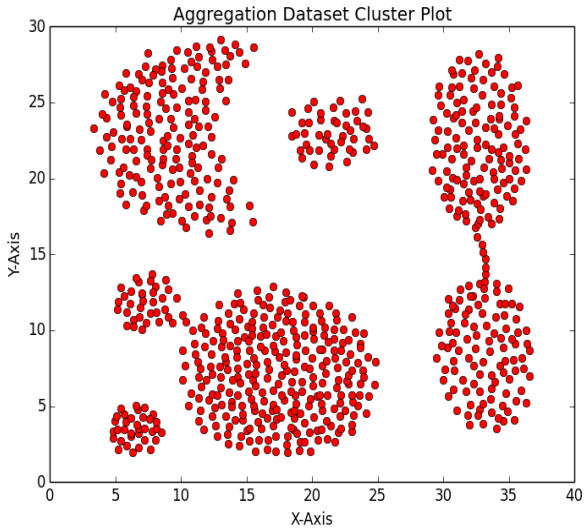

Figure 8. Cluster plot of Aggregation Dataset. It's a 2-Dimensional dataset with 788 data points and 7 main clusters.

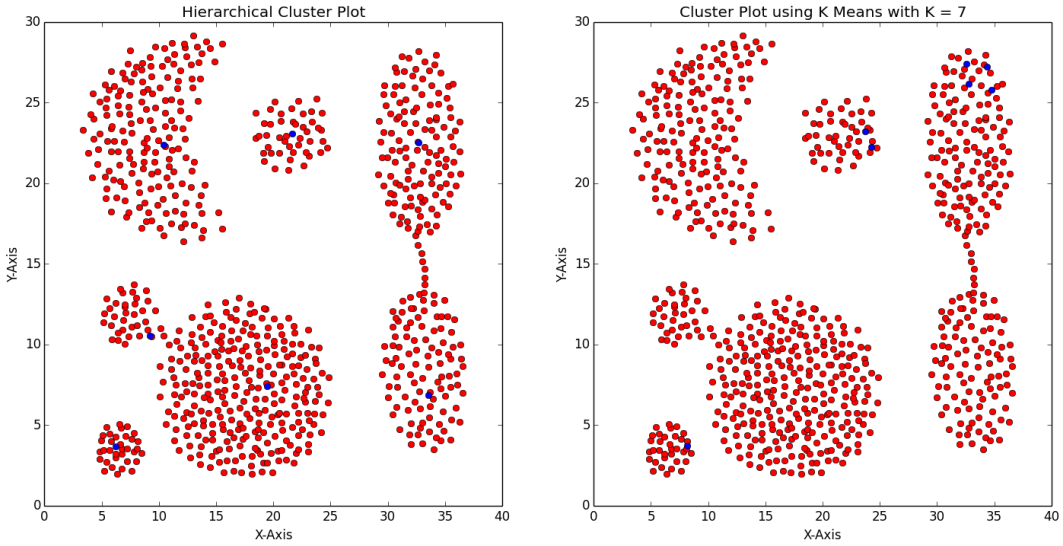

Figure 9. The Cluster plot above depicts the clustering produced by Agglomerative Hierarchical Clustering and K-Means with random initial center selection with the value of K set to 7. The blue dots denote the final centroids arrived at by the algorithm.

The above-mentioned dataset when clustered using the standard K-Means and standard Agglomerative Hierarchical Clustering algorithm with centroid link and Euclidian distance metric for both, yields the two outputs as depicted in figure 9.

The figure demonstrates the problem faced by the K-Means algorithm when initial centroid selection is concerned. If the selection is not smart, the final clustering produced may not be acceptable.

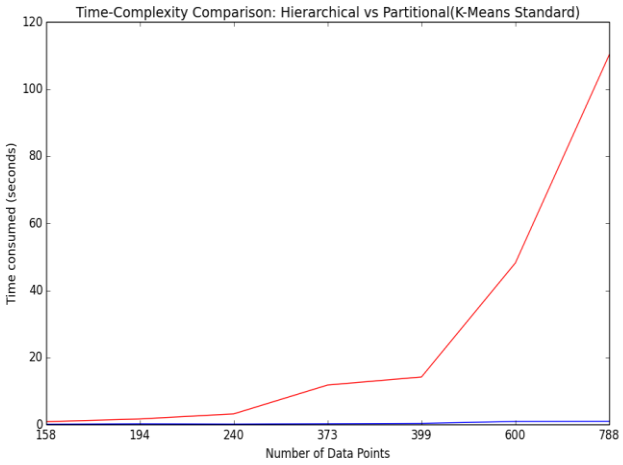

Figure 10. The time-complexity comparison chart for the time-consumed by the two methods when clustering the distinct datasets chosen indicated by the number of data points present within each.

For time-complexity analysis, we made use of the Aggregation dataset and the Compound, Flame, R15, Jain and Pathbased datasets acquired from the University of Eastern Finland's website [18]. If we disregard the cluster quality produced using the standard implementations of either algorithms, Figure 10 is indicative of the time complexity difference between the two. Table 1 further clarifies the difference.

Data-Set Name	No. of Points	Hierarchical (seconds)	K-Means (seconds)
Artificial dataset	158	0.8864	0.1271
Artificial dataset	194	1.7020	0.2501
Flame	240	3.2095	0.1601
Jain	373	11.8070	0.2727
Compound	399	14.2125	0.3880
R15	600	48.1943	0.9611
Aggregation	788	110.2562	0.8555

Table 1: The time-consumption table for the two algorithms.

Among Partitional algorithms, K-Means holds the central position as the most prominent and widely used clustering method. It has a run-time complexity of its standard implementation as $O(knl)$ where 'n' is number of data-points, 'k' is the number of clusters and 'l' is the number of iterations. It however requires 'k' and 'l' to be known beforehand, although much enhancements have been performed for the algorithm to detect the number of clusters required in an automated sense [14]. One of its major forte is its clustering speed which becomes effectively linear when $k, l \ll n$.

Despite having the linear time complexity advantage over Hierarchical clustering, K-Means suffers from the following well known drawbacks:

1. The outlier sensitivity, i.e. the clustering is sensitive to points which lie at a farther distance than the other points in the cluster.
2. It's sensitive to the initial cluster centers chosen and the clustering outcome can largely vary due to poor initial center choices.
3. Variability in the outputs produced as the clustering produced may not be unique.
4. Convergence onto a locally optimal clustering solution and not taking into account the globally optimal solution, if available.

3. CONCLUSION

Taking due consideration of the relative advantages and disadvantages offered by the usage of Hierarchical and Partitional methods, the application of either techniques onto a problem domain varies according to the size of the dataset involved and the cluster quality desired. The practical implementation may even incorporate both the techniques to get the best of both worlds [7].

4. FUTURE SCOPE

The K-Means algorithm, although ineffective in its standard implementation with random initial center selection, has been subjected to continual improvements over an extended period of time right from its inception.

Much enhancements have been made over the traditional algorithms [15] [16] from both the Hierarchical and Partitional domains.

Current challenges for Data-Mining and Cluster Analysis include higher dimensional clustering, stream data clustering and highly heterogeneous data clustering for which K-Means [9] has been adapted to perform well. Initial seeding [3] and centroid selection has been extensively improved upon as well. Hierarchical clustering algorithms have been subjected to an equally extensive extension and improvement with methods devised to reduce the complexity to log-linear [5] and to make them relevant for higher dimensional analysis. Thus both the algorithms are highly relevant to the multi-disciplinary applications of today's world.

With emerging domains of Artificial Vision, Machine Learning and Big Data, scalability and parallelization of both the techniques present a suitable field for active research and presents opportunities for further improvement.

5. REFERENCES

[1] Agrawal, R., Gehrke, J., Gunopulos, D., & Raghavan, P. (1998). Automatic subspace clustering of high dimensional data for data mining applications (Vol. 27, No. 2, pp. 94-105). ACM.

[2] Ankerst, M., Breunig, M. M., Kriegel, H. P., & Sander, J. (1999, June). OPTICS: ordering points to identify the clustering structure. In ACM Sigmod Record (Vol. 28, No. 2, pp. 49-60). ACM.

[3] Arthur, D., & Vassilvitskii, S. (2007, January). k-means++: The advantages of careful seeding. In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms (pp. 1027-1035). Society for Industrial and Applied Mathematics.

[4] Bezdek, J. C., Ehrlich, R., & Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. Computers & Geosciences, 10(2), 191-203.

[5] David Eppstein, Fast Hierarchical Clustering via Dynamic Closest Pairs, Dept. Information and Computer Science, Univ. of California, Irvine, http://www.ics.uci.edu/~eppstein/

[6] Dempster, A.P.; Laird, N.M.; Rubin, D.B. (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm". Journal of the Royal Statistical Society. Series B (Methodological).

[7] Ghwanmeh, Sameh H. "Applying Clustering of hierarchical K-means-like Algorithm on Arabic Language." International Journal of Information Technology3.3 (2005).

[8] Gionis, A., H. Mannila, and P. Tsaparas, Clustering aggregation. ACM Transactions on Knowledge Discovery from Data (TKDD), 2007. 1(1): p. 1-30.

[9] Jain, A. K. (2010). Data clustering: 50 years beyond K-means. Pattern recognition letters, 31(8), 651-666.

[10] Kaufman, L. and Rousseeuw, P.J. (1990). Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, New York

[11] MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. Berkeley, University of California Press, 1, 281-297.

[12] Martin Ester and Hans-peter Kriegel and Jörg S and Xiaowei Xu (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. KDD96 Proceedings, AAAI Press, 226-231.

[13] Mittal, Mamta. "Validation of k-means and Threshold based Clustering Method." International Journal of Advancements in Technology 5.2 (2014): 153-160.

[14] Mittal, M.; Singh, V.P.; Sharma, R.K., "Random automatic detection of clusters," Image Information Processing (ICIIP), 2011 International Conference on Intelligent Information Processing, vol., no., pp.1,6, 3-5 Nov. 2011, doi: 10.1109/ICIIP.2011.6108856.

[15] Park, Hae-Sang, and Chi-Hyuck Jun. "A simple and fast algorithm for K-medoids clustering." Expert Systems with Applications 36.2 (2009): 3336-3341.

[16] Reddy, Damodar, Prasanta K. Jana, and IEEE Senior Member. "Initialization for K-means clustering using Voronoi diagram." Procedia Technology 4 (2012): 395-400.

[17] Wang, W., Yang, J., & Muntz, R. (1997, August). STING: A statistical information grid approach to spatial data mining. In VLDB (Vol. 97, pp. 186-195).

[18] http://cs.joensuu.fi/sipu/datasets/ Speech and Image Processing Unit, School of Computing, University of Eastern Finland, Joensuu, Finland.