ISSN No. 0976-5697

Volume 3, No. 3, May-June 2012

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Survey of Different Partition Clustering Algorithms and their Comparative Studies

Anshul Parashar,
Associate Profesor
Department of Computer Sc. & Engineering
H.C.T.M, Kaithal, Haryana (India)
parashar_anshul@yahoo.com

Yogita Gulati*
M.Tech Student
Department of Computer Sc. & Engineering
H.C.T.M, Kaithal, Haryana (India)
yogitagulati1111@yahoo.com

Abstract: Clustering is one of the most important research areas in the field of data mining. Clustering means creating groups of objects based on their features in such a way that the objects belonging to the same groups are similar and those belonging in different groups are dissimilar. Clustering is an unsupervised learning technique. Data clustering is the subject of active research in several fields such as statistics, pattern recognition and machine learning. From a practical perspective clustering plays an outstanding role in data mining applications in many domains. The main advantage of clustering is that interesting patterns and structures can be found directly from very large data sets with little or none of the background knowledge. Data mining adds to clustering the complications of very large datasets with very many attributes of different types. This imposes unique computational requirements on relevant clustering algorithms. A variety of algorithms have recently emerged that meet these requirements and were successfully applied to real-life data mining problems. They are subject of this survey. Also, this survey explores the behavior of some of the partition based clustering algorithms.

Keywords: Clustering algorithms, K-mean clustering, K-medoids clustering, fuzzy C- mean clustering, k-attractor clustering.

I. INTRODUCTION

The goal of this survey is to provide a comprehensive review of different partition based clustering algorithms in data mining. Clustering is a division of data into groups of similar objects. Each group, called cluster, consists of objects that are similar between themselves and dissimilar to objects of other groups. Representing data by fewer clusters necessarily loses certain fine details (akin to lossy data compression), but achieves simplification. It represents many data objects by few clusters and hence, it models data by its clusters. Clustering is a method of unsupervised learning and a common technique for statistical data analysis used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics. Besides the term clustering, there are a number of terms with similar meanings, including automatic classification, numerical taxonomy, botryology and typological analysis. Therefore, clustering is unsupervised learning of a hidden data concept [1],[2],[3],[4].

A. *Components of a Clustering Task*

Typical pattern clustering activity involves the following steps [13]:

1) Pattern representation (optionally including feature extraction and/or selection),
2) Definition of a pattern proximity measure appropriate to the data domain,
3) Clustering or Grouping,
4) Data Abstraction (if needed), and
5) Assessment of output (if needed).

Figure 1 depicts a typical sequencing of the first three of these steps, including a feedback path where the grouping process output could affect subsequent feature extraction and similarity computations.

Pattern representation refers to the number of classes, the number of available patterns, and the number, type, and scale of the features available to the clustering algorithm. Some of this information may not be controllable by the practitioner. *Feature selection* is the process of identifying the most effective subset of the original features to use in clustering. *Feature extraction* is the use of one or more transformations of the input features to produce new salient features. Either or both of these techniques can be used to inorder to obtain an appropriate set of features to use in clustering.

Pattern proximity is usually measured by a distance function defined on pairs of patterns. A variety of distance measures are in use in the various communities [5], [6],[7]. A simple distance measure like Euclidean distance can often be used to reflect dissimilarity between two patterns, whereas other similarity measures can be used to characterize the conceptual similarity between patterns [8]. The grouping step can be performed in a number of ways. The output clustering (or clusterings) can be hard (a partition of the data into groups) or fuzzy (where each pattern has a variable degree of membership in each of the output clusters).

How is the output of a clustering algorithm evaluated? What characterizes a 'good' clustering result and a 'poor' one? All clustering algorithms will, when presented with data, produce clusters — regardless of whether the data contain clusters or not. If the data does contain clusters, some clustering algorithms may obtain 'better' clusters than others. The assessment of a clustering procedure's output, then, has several facets. One is actually an assessment of the data domain rather than the clustering algorithm itself— data which do not contain clusters should not be processed by a clustering algorithm. The study of *cluster tendency*, wherein the input data are examined to see if there is any merit to a

cluster analysis prior to one being performed, is a relatively inactive research area, and will not be considered further in this survey.

Data mining deals with large databases that impose on cluster analysis**.** Some of the challenges led to the emergence of powerful broadly applicable data mining clustering methods surveyed below. A variety of clustering algorithms have been used for research in the field of data mining [1],[2],[3],[9],[10],[11],[12]. The scope of this survey is modest: to provide an introduction to cluster analysis in the field of data mining, where, it is to define data mining to be the discovery of useful, but non-obvious, information or patterns in large collections of data. The survey is based on number of partitions based clustering techniques that have recently been developed.

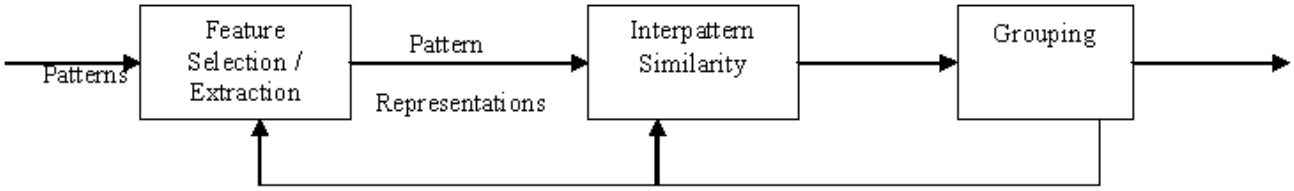

Figure1.Stages in Clustering

II. CLUSTERING OVERVIEW

Clustering algorithm can be divided into the following categories:

1. Hierarchical clustering algorithm
2. Partition clustering algorithm
3. Spectral clustering algorithm
4. Grid based clustering algorithm
5. Density based clustering algorithm

A. Hierarchical Clustering Algorithm

Hierarchical clustering algorithm groups data objects to form a tree shaped structure. It can be broadly classified into agglomerative hierarchical clustering and divisive hierarchical clustering. In agglomerative approach which is also called as bottom up approach, each data points are considered to be a separate cluster and on each iteration clusters are merged based on a criteria. The merging can be done by using single link, complete link, centroid or wards method. In divisive approach all data points are considered as a single cluster and they are splited into number of clusters based on certain criteria, and this is called as top down approach.

B. Spectral Clustering Algorithm

Spectral clustering refers to a class of techniques which relies on the Eigen structure of a similarity matrix. Clusters are formed by partition data points using the similarity matrix.

Any spectral clustering algorithm will have three main stages [13]. They are:

1) Preprocessing: Deals with the construction of similarity matrix.
2) Spectral Mapping: Deals with the construction of eigen vectors for the similarity matrix.
3) Post Processing: Deals with the grouping data points.

The following are advantages of Spectral clustering algorithm:

1) Strong assumptions on cluster shape are not made.
2) Simple to implement.
3) Objective does not consider local optima.
4) Statistically consistent.
5) Works faster.

The major drawback of this approach is that it exhibits high computational complexity. For the larger dataset it requires $O(n^3)$ where n is the number of data points [14].

C. Grid based Clustering Algorithm

Grid based algorithm quantize the object space into a finite number of cells that forms a grid structure .Operations are done on these grids. The advantage of this method is lower processing time. Clustering complexity is based on the number of populated grid cells and does not depend number of objects in the dataset. The major features of this algorithm Clusters

Feedback Loop are:

1. No distance computations.
2. Clustering is performed on summarized data points.
3. Shapes are limited to union of grid-cells.
4. The complexity of the algorithm is usually Big – Oh (Number of populated grid-cells)

D. Density based Clustering Algorithm

Density based algorithm continue to grow the given cluster as long as the density in the neighborhood exceeds certain threshold. This algorithm is suitable for handling noise in the dataset. The following points are enumerated as the features of this algorithm.

1) Handles clusters of arbitrary shape
2) Handle noise
3) Needs only one scan of the input dataset.
4) Needs density parameters to be initialized.

III. PARTITION CLUSTERING ALGORITHM

The term cluster does not have a precise definition so far. However, several working definitions of a cluster are present. A cluster is a set of points such that any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster. Sometimes a threshold is used to specify that all the points in a cluster must be sufficiently close (or similar) to one another. However, in many sets of data, a point on the edge of a cluster may be closer (or more similar) to some objects in another cluster than to objects in its own cluster. Consequently, many clustering algorithms use the center-based cluster criterion. The center of a cluster is often a

centroid, the average of all the points in the cluster, or a medoid, the most representative point of a cluster.

A partitioning method first creates an initial set of k partitions, where, parameter k is the number of partitions to construct. It then uses an iterative relocation technique that attempts to improve the partitioning by moving objects from one group to another. These clustering techniques create a one-level partitioning of the data points. There are a number of such techniques, but this survey shall only describe three approaches namely K-means, K-medoids , fuzzy C-means and K-attractor. For K-medoid, the notion of a medoid is used, which is the most representative (central) point of a group of points. K-means is a simple algorithm that has been adapted to many problem domains. It can see that the K-means algorithm is a good candidate for extension to work with fuzzy feature vectors. Therefore the algorithm with fuzzy feature is called the Fuzzy C-Means (FCM) algorithm. We propose here k-Attractors, a partitional clustering algorithm tailored to numeric data analysis. As a pre-processing (initialization) step, it employs maximal frequent itemset discovery and partitioning to define the number of clusters *k* and the initial cluster "attractors". During its main phase the algorithm utilizes a distance measure, which is adapted with high precision to the way initial attractors are determined. Comparison favored k-Attractors in terms of convergence speed and cluster formation quality in most cases. On the downside, its initialization phase adds an overhead that can be deemed acceptable only when it contributes significantly to the algorithm's accuracy.

A. K-MEANS ALGORITHM

K-Means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori [9],[15]. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When, no point is pending, the first step is completed and an early group age is done. At this point it is necessary to re-calculate k new centroids as bar centers of the clusters resulting from the previous step. After obtaining these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop, one may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more. Finally, this algorithm aims at minimizing an objective function, in this case a squared error function. The objective function:

$$j = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^{m} \left\| x_i - c_j \right\|^2 \qquad (1)$$

where, $\|x_i^{(j)}-c_j\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j, is an indicator of the distance of the n data points from their respective cluster centers. The algorithm is composed of the following steps:

1) *Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.*
2) *Assign each object to the group that has the closest centroid.*
3) *When all objects have been assigned, recalculate the positions of the K centroids.*
4) *Repeat Steps 2 and 3 until the centroids no longer move.*

This produces a separation of the objects into groups from which the metric to be minimized can be calculated. Always the algorithm can be proved that the procedure will terminate, the K-means algorithm does not necessarily find the most optimal configuration, corresponding to the global objective function minimum. The algorithm is also significantly sensitive to the initial randomly selected cluster centers. The K-means algorithm can be run multiple times to reduce this effect [2],[3],[15]. K-means is a simple algorithm that has been adapted to many problem domains and it is a good candidate to work for a randomly generated data points.

B. K-MEDOIDS ALGORITHM

The K-means algorithm is sensitive to outliers since an object with an extremely large value may substantially distort the distribution of data. How might the algorithm be modified to diminish such sensitivity? Instead of taking the mean value of the objects in a cluster as a reference point, a medoid can be used, which is the most centrally located object in a cluster. Thus, the partitioning method can still be performed based on the principle of minimizing the sum of the dissimilarities between each object and its corresponding reference point. This forms the basis of the K-medoids method [3],[4],[16]. The basic strategy of K-medoids clustering algorithms is to find k clusters in n objects by first arbitrarily finding a representative object (the medoids) for each cluster. Each remaining object is clustered with the medoid to which it is the most similar. K-medoids method uses representative objects as reference points instead of taking the mean value of the objects in each cluster. The algorithm takes the input parameter k, the number of clusters to be partitioned among a set of n objects [2], [3], [11], [17], [18].

A typical K-Mediods algorithm for partitioning based on medoid or central objects is as follows:

***Input*:** '*k*', the number of clusters to be partitioned;'n', the number of objects.

***Output*:** A set of '*k*' clusters that minimizes the sum of the dissimilarities of all the objects to their nearest medoid

***Steps*:** *i) Arbitrarily choose 'k' objects as the initial medoids;*
ii) Repeat,
a) Assign each remaining object to the cwith the nearest medoid;
b) Randmly select a non - medoid object;
c) Compute the total cost of swapping old medoid object with newly selected non-medoid object.
d) If the total cost of swapping is less than zero, then perform that swap operation to form the new set of k-medoids.
iii) Until no change.

Like this algorithm, a Partitioning Around Medoids (PAM) was one of the first K-medoids algorithms introduced. It attempts to determine k partitions for n objects. After an initial random selection of k medoids, the algorithm repeatedly tries to make a better choice of

medoids. Therefore, the algorithm is often called as representative object based algorithm.

C. FUZZY C-MEANS ALGORITHM

Traditional clustering approaches generate partitions; in a partition, each pattern belongs to one and only one cluster. Hence, the clusters in a hard clustering are disjoint. Fuzzy clustering extends this notion to associate each pattern with every cluster using a membership function . The output of such algorithms is a clustering, but not a partition. Fuzzy clustering is a widely applied method for obtaining fuzzy models from data. It has been applied successfully in various fields including geographical surveying, finance or marketing. This method is frequently used in pattern recognition. It is based on minimization of the following objective function:

$$j = \sum_{i=1}^{n}\sum_{j=1}^{c} u_{ij}^{m} \left\| x_i - c_j \right\|^2 , \qquad (2)$$

1≤m<∞

where, m is any real number greater than 1, u_{ij} is the degree of membership of x_i in the cluster j, x_i is the ith of d-dimensional measured data, c_j is the d-dimension center of the cluster and ||*|| is any norm expressing the similarity between any measured data and the center [19],[20]. Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above, with the update of membership u_{ij} and the cluster centers c_j by:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{\left\| x_i - c_j \right\|}{\left\| x_i - c_k \right\|}\right)^{\frac{2}{m-1}}}, \quad c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} . x_i}{\sum_{i=1}^{N} u_{ij}^{m}} \qquad (3)$$

This iteration will stop when $\max_{ij} \{|u_{ij}^{(k+1)} - u_{ij}^{(k)}|\} < \varepsilon$, where, ε is a termination criterion between 0 and 1, whereas k are the iteration steps. This procedure converges to a local minimum or a saddle point of J_m. The algorithm is composed of the following steps:

1) *Initialize U = [uij] matrix, U(0).*
2) *At k-step: calculate the centers vectors C(k) = [cj] with U(k).*
3) *Update U(k), U(k+1).*
4) *If || U(k+1)-U(k)||<ε then STOP; otherwise return to step 2.*

In this algorithm, data are bound to each cluster by means of a membership function**,** which represents the fuzzy behavior of the algorithm. To do that, the algorithm has to build an appropriate matrix named U whose factors are numbers between 0 and 1 and represent the degree of membership between data and centers of clusters. FCM clustering techniques are based on fuzzy behavior and provide a natural technique for producing a clustering where membership weights have a natural (but not probabilistic) interpretation. This algorithm is similar in structure to the K-means algorithm and also behaves in a similar way.

D. K-ATTRACTOR ALGORITHM

K-Means is a classic partitional clustering algorithm [21]. It represents each cluster with the mean value of its objects. As a result, inter-cluster similarity is measured based on the distance between the object and the mean value of the input data in a cluster. It is an iterative algorithm in which objects are moved among clusters until a desired set is reached. Its main problems are that the users have to define the number of clusters *k*, and that it is sensitive to the initial partitioning. That is, different initial partitions, indicated by user input, lead to different results [22].

This paper presents a further elaborated version of k-Attractors, a partitional clustering algorithm introduced in, which has the following characteristics:

- It defines the desired number of clusters (i.e. the number of k), without user
- intervention.
- It locates the initial attractors of cluster centers with great precision.
- It measures similarity based on a composite metric that combines the Hamming distance and the inner product of transactions and clusters' attractors.

The k-Attractors algorithm employs the maximal frequent itemset discovery and partitioning in order to find the number of desired clusters and the initial attractors of the centers of these clusters. The intuition is that a frequent itemset in the case of software metrics is a set of measurements that occur together in a minimum part of a software system's classes. Classes with similar measurements are expected to be on the same cluster. The term attractor is used instead of centroid, as it is not determined randomly, but by its frequency in the whole population of a software system's classes. The main characteristic of k-Attractors is that it proposes a similarity measure which is adapted to the way initial attractors are determined by the preprocessing method. Hence, it is primarily based on the comparison of frequent itemsets. More specifically, a composite metric based on the Hamming distance and the dot (inner) product between each transaction and the attractors of each cluster is utilized. The two basic steps of the k-Attractors algorithm are:

- **Initialization Phase**

- The first step of this phase is to generate frequent itemsets using the APriori algorithm. The derived frequent itemsets are used to construct the itemset graph, and a graph partitioning algorithm is used to find the number of the desired clusters and assign each frequent itemset into the appropriate cluster. As soon as the number of the desired clusters (k) is determined, we select the maximal frequent itemsets of every cluster, forming a set of k frequent itemsets as the initial attractors.

- **Main Phase :**

-As soon as the attractors have been found, we assign

each transaction to the cluster that has the minimum Score (Ci← tj) against its attractor.

When all transactions have been assigned to clusters we recalculate the attractors for each cluster in the same way as during the initialization phase.

The k-Attractors algorithm utilizes a hybrid similarity metric based on vector representation of both the data items and the cluster's attractors. The similarity of these vectors is measured employing the following composite metric:

$$\mathbf{Score\ (Ci \leftarrow tj) = h * H(ai, tj) + i*(a_1 * t_1 + \dots a_n * t_n)} \quad (4)$$

Table.1 k-Attractors Input Parameters

Parameter	Description
Support s	It defines the required support for discovery of initial attractors.
Hamming Distance power h	It defines the similarity metric's sensitivity to hamming distance.
Inner Product power i	It defines the similarity metric's sensitivity to Inner product.
Number of initial attractors k	It defines the similarity metric's sensitivity to clusters.

K-attractor algorithm:

/*Input Parameters*/
Support: s
Hamming Distance power : h
Inner Product power : i
Number of initial attractors: k
Given a set of m data items $t_1, t_2, \dots t_m$.

/*Initialization Phase*/
1) Generate frequent itemsets using Apriori Algorithm;
2) Construct the *itemset graph* and partition it using the confidence similarity related to the support of these itemsets;
3) Use the no. of partitions as final k ;
4) Select the maximal frequent itemset of every cluster in order to form a set of k initial attractors;

/*Main Phase*/
Repeat
6) Assign each dta item to the cluster that has the minimum *Score* ($C_i \rightarrow t_j$);
7) When all data items have been assigned ,recalculate new attractors; Untill ti don't move.
8) Search all clusters to find outliers and group them in a new cluster.

In this formula, the first term is the Hamming distance between the attractor and the data item. It is given by the number of positions that pair of strings is different and is defined as follows:

$$H(ai, tj) = n - \#(ai \cap tj) \quad (5)$$

As the algorithm is primarily based on itemsets' similarity, we want to measure the number of substitutions required to change one into the other. The second term is the dot (inner) product between this data item and the attractor. It is used in order to compensate for the position of both vectors in the Euclidean space. Because of the semantics of software measurement data, the usually utilized internal metrics (such as lines of code, coupling between objects, number of comments etc) have large positive integer values. Thus in order for the inner product distance to be more accurate, we firstly normalize all the values in the interval [-1, 1] and then apply the k-Attractors algorithm. The multipliers in equation define the metric's sensitivity to Hamming distance and inner product respectively. For example, the case indicates the composite metric is insensitive to the inner product between the data item and the cluster's centroid. Thus, k-Attractors provides theflexibility of changing the sensitivity of the composite distance metric to both Hamming distance and inner product, in correspondence with the each clustering scenario's semantics.

IV. CONCLUSION AND FUTURE SCOPE

Usually the time complexity varies from one processor to another processor, which depends on the speed and the type of the system. The partition based algorithms work well for finding spherical-shaped clusters in small to medium-sized data points. The advantage of the K-means algorithm is its favorable execution time. Its drawback is that the user has to know in advance how many clusters are searched for. It is observed that K-means algorithm is efficient for smaller data sets and K-medoids algorithm seems to perform better for large data sets. The performance of FCM is intermediary between them. FCM produces close results to K-means clustering, yet it requires more computation time than K-means because of the fuzzy measures calculations involved in the algorithm. The survey shows that k-Attractors, tailored to numeric data analysis, overcome the weaknesses of other partitional algorithms. The initialization phase of the proposed algorithm involves a preprocessing step which calculates the initial partitions for k-Attractors. During this phase, the exact number of k-Attractors clusters was calculated in addition with the initial attractors of each cluster. Thus the problems of defining the number of clusters and initializing the centroids of each cluster are resolved. In addition, the constructed initial attractors approximate the real clusters' attractors, improving that way the convergence speed of the proposed algorithm.

The main phase of k-Attractors forms clusters employing a composite distance metric which utilises the Hamming distance and the inner product of data item vector representations. Thus, the employed metric is adapted to the way the initial attractors are determined by the preprocessing step. The last step deals with outliers and is based on the distance between a data item and its cluster's attractor. The discovered outliers are grouped into a separate cluster. The results from the conducted experiments are promising, as k-Attractors' main phase outperformed in

performance and accuracy the other algorithms in most of the cases. This is attributed to its initialization phase which however adds an overhead which is deemed acceptable when it contributes significantly to algorithm's accuracy.

For this reason we plan on improving the way the initial attractors are derived in order to minimize the cost of the initialization phase. We could also attempt to customise the proposed distance metric in order to adapt it to categorical semantics thus making it applicable to categorical datasets.

V. REFERERENCES

[1] Berkhin, P., 2002. Survey of Clustering Data Mining Techniques. Accure Software Inc., San Jose, CA., USA.

[2] Dunham, M., 2003. Data Mining: Introductory and Advanced Topics. Prentice Hall, USA.

[3] Han, J. and M. Kamber, 2006. Data Mining: Concepts and Techniques. 2nd Edn., Morgan Kaufmann Publisher, San Fransisco, USA., ISBN:1-55860-901-6.

[4] Jain, A.K., M.N. Murty and P.J. Flynn, 1999. Data clustering: A review. ACM Comput. Surveys, 31: 264-323.

[5] ANDERBERG, M. R. 1973. Cluster Analysis for Applications. Academic Press, Inc., NewYork, NY.

[6]] DIDAY,E.AND SIMON, J. C. 1976. Clustering analysis. In Digital Pattern Recognition,K. S. Fu, Ed. Springer-Verlag, Secaucus, NJ, 47–94

[7] Jain, A.K. and R.C. Dubes, 1988. Algorithms for Clustering Data. Prentice Hall Inc., Englewood Cliffs, New Jerssy, ISBN: 0-13-022278-X,pp:320.

[8] MICHALSKI,R., STEPP,R.E.,AND DIDAY,E. 1983. Automated construction of classifications : conceptual clustering versus numerical taxonomy. IEEE Trans. Pattern Anal Mach. Intell. PAMI-5, 5 (Sept.), 396–409.

[9] AAlexander, R. and A. Caponnetto, 2007. Stability of K-Means Clustering, Advances in Neural Information Processing Systems 12. MIT Press, Cambridge, MA, pp: 216-222.

[10] Khan, S.S. and A. Ahmad, 2004. Cluster center initialization algorithm for K-Means clustering. Pattern Recognition Lett., 25: 1293-1302.

[11] Park, H.S., J.S. Lee and C.H. Jun, 2009. A K-Means-Like Algorithm for K-Medoids Clustering and Its Performance. Department of Industrial and Management Engineering, POSTECH, South Korea.

[12] Xiong, H., J. Wu and J. Chen, 2006. K-means clustering versus validation measures: A data distribution perspective.

[13] M. Meila, D Verma,2001.Comparison of spectral clustering algorithm. University of Washington, Technical report.

[14] Cai X. Y. et al, 2008. Survey on Spectral Clustering Algorithms. Computer Science:14-18

[15] Borah, S. and M.K. Ghose, 2009. Performance analysis of AIM-K-Means and K-Means in quality cluster generation. J. Comput., 1: 175-178.

[16] Kaufman, L. and P.J. Rousseeuw, 1990. Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley and Sons, New York, ISBN: 0471878766, pp: 342.

[17] Sheng, W. and X. Liu, 2006. A genetic k-medoids clustering algorithm. J. Heuristics, 12: 447-466

[18] Zeidat, N. and C.F. Eick, 2004. K-medoid-style clustering algorithms for supervised summary generation. Proceedings of the International Conference on Machine Learning.

[19]A1-Zoubi, M.B., A. Hudaib and B. Al-Shboul, 2007. A fast fuzzy clustering algorithm. Proceedings of the 6th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases, February 2007, Corfu Island, Greece, pp: 28-32.

[20] Yong, Y., Z. Chongxun and L. Pan, 2004. A novel fuzzy c-means clustering algorithm for image thresholding. Measurement Sci. Rev., 4: 9-9.

[21] Hartigan 1975) Hartigan J. A. 1975, Clustering Algorithms. John Wiley & Sons, New York, NY.

[22] (Han, Kamber, 2001) Han J. and Kamber M., 2001, Data Mining: Concepts and Techniques, Academic Press.