Abstract

One of new challenges of our daily activity, is how to deal with uprising amount of unclassified and erratic data, to search through high volume of data, in order to point to specific category of data. Traditional statistical approach for analyzing, organizing and identifying categories among large volume of data has computational barriers and is a time-consuming task. Modern technology and solutions are applied on spatial data with the purpose of extracting knowledge autonomously. Advanced data mining techniques are potential candidates to reduce risk of being involved with dizzy, gloaming unclassified information which are proliferated increasingly. One of the unsupervised techniques to classification problem is clustering mechanism leveraging many aspect of big-data analysis. this paper presents a new approach of clustering technique with definition of new hybrid model outperform classic methods of clustering such as partitioned or hierarchical algorithms. It uses the systematic cooperation of two popular clustering algorithms: the AGlomerative NEStive (AGNES), as a hierarchical clustering method and κ-means, as a partitional clustering method. It inherits the aspects of two approaches: *low time complexity and closest relative neighbors’* mechanism in k-means and AGNES algorithms, respectively. The proposed method expects to be faster and more accurate than two classic methods. The paper evaluate the result using several popular evaluation criteria indicated in the evaluation section. The result reveals that the proposed algorithm performs faster with a higher quality of clusters regarding to inherited merits from parents (k-means, AGNES).

1. **Introduction**

With subtle growth in tendency among researchers to carry out data mining techniques on different type of data (Medicines, Geographical or Weather-related data), knowledge discovery is considered to be needed more frequently ever before. KDD[[1]](#footnote-1) is the higher level process of obtaining facts through data mining and distilling this information into knowledge or ideas and beliefs about the mini-world described by the data. This generally requires a human-level intelligence to guide the process and interpret the results based on pre-existing knowledge[1]. Exiting methods for exploratory spatial analysis and spatial datamining span across three main categories: computational, statistical, and visual approaches[2]. The paper subjects mainly on first category of spatial analysis. Computational approaches which resort to computer algorithms to search large volume of data in order to find specific type of patterns such as spatial clusters[3], spatial association rules[4] and spatial outliers[5].

In general, computational methods are able to search for structures in large datasets with great efficiency but lack the ability to interpret and attach meaning to patterns[2].Statistical methods are rigorous and verifiable but often assume a priorimodel which has been roughly predetermined by the analyzer[2]. Visualization techniques in spatial analysis is………. (Source should indicates the topic)

This paper organized around topics to propose a solution in computational aspect of spatial analysis. Computational methods are able to search for structures in large datasets with great efficiency but lack the ability to interpret and attach meaning to patterns[2]. Presenting an effective method in order to cluster spatial data which are gathered from diverse sources is a challenging task. Employing clustering methods in order to discover related data in spatial analysis is a main purpose of our study. The proposed method utilizes a systematic hybrid approach by combining AGNES as a hierarchical and K-means as a partitional clustering algorithms. The paper will cover both the new proposed algorithm, and its function in crime incidents’ location data as spatial analysis. The case study assess clustering approach in different type of dataset…. (Crimes location data, weather data, or diabetes data) … Eventually, the method has been tested and evaluated through using different types of data (spatial data). Different types of data are selected in order to assess the accuracy and quality of the new proposed clustering method.

* 1. Clustering From (Survey of Clustering Data mining Techniques)

Cluster analysis or clustering is a task of assigning a set of objects into groups (called clusters) so that the object in the same cluster are more similar to each other base on selected features than to those on other clusters [6]. Comparison between different clustering methods, have been performed in order to present the quality, accuracy and efficiency of one algorithm comparing with other methods.

There is a close relationship between clustering techniques and many other disciplines. Clustering has always been used in statistics [Arabie & Hubert 1996] and science [Massart & Kaufman 1983]. The classic introduction into pattern recognition framework is given in [Duda & Hart 1973]. Typical applications include speech and character recognition. Machine learning clustering algorithms were applied to image segmentation and computer vision [Jain & Flynn 1996]. For statistical approaches to pattern recognition see [Dempster et al. 1977] and [Fukunaga 1990]. Clustering can be viewed as a density estimation problem. This is the subject of traditional multivariate statistical estimation [Scott 1992]. Clustering is also widely used for data compression in image processing, which is also known as vector quantization [Gersho & Gray 1992]. Data 3 fitting in numerical analysis provides still another venue in data modeling [Daniel & Wood 1980].

Categorization of clustering algorithms is neither straightforward, nor canonical. In reality, groups below overlap. For readers convenience we provide a classification closely followed by this survey. Corresponding terms are explained below.

1. Hierarchical Methods

A1. Agglomerative Algorithms

A2. Divisive Algorithms

1. Partitioning Methods

B1. Relocation Algorithms

B2. Probabilistic Clustering

B3. K-medoids Methods

B4. K-means Methods

1. Density-Based Algorithms

C1. Density-Based Connectivity Clustering

C2. Density Functions Clustering

1. Grid-Based Methods
2. Methods Based on Co-Occurrence of Categorical Data
3. Constraint-Based Clustering
4. Clustering Algorithms Used in Machine Learning
5. Gradient Descent and Artificial Neural Networks

H1. Evolutionary Methods

1. Scalable Clustering Algorithms

I1. Algorithms for High Dimensional Data

I2. Subspace Clustering

I3. Projection Techniques

I4. Co-Clustering Techniques

* 1. **K-means**

According to [7], K-mean is an algorithm of type Partitioning Relocation Clustering. The k-means algorithm [Hartigan 1975; Hartigan & Wong 1979] is by far the most popular clustering tool used in scientific and industrial applications. The name comes from representing each of k clusters C by the mean (or weighted average) c of its points, the so-called centroid. While this obviously does not work well with categorical attributes, it has the good geometric and statistical sense for numerical attributes. The sum of discrepancies between a point and its centroid expressed through appropriate distance is used as the objective function. For example, the -norm based objective function, the sum of the squares of errors between the points and the corresponding centroids, is equal to the total intra-cluster variance.

Two versions of k-means iterative optimization are known. The first version is similar to EM algorithm and consists of two-step major iterations that (1) reassign all the points to their nearest centroids, and (2) recomputed centroids of newly assembled groups. Iterations continue until a stopping criterion is achieved (for example, no reassignments Happen). This version is known as Forgy’ s algorithm [Forgy 1965] and has many advantages:

1. It allows straightforward parallelization [Dhillon & Modha 1999]
2. It is insensitive with respect to data ordering.

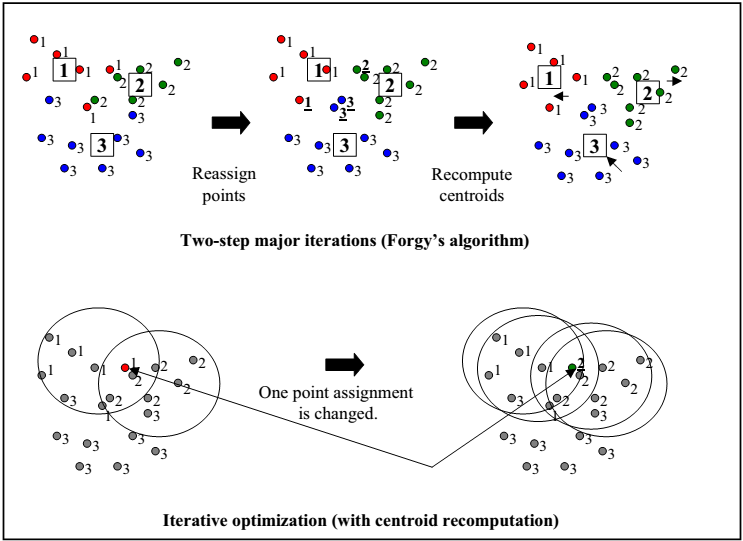
The second (classic in iterative optimization) version of k-means iterative optimization reassigns points based on more detailed analysis of effects on the objective function caused by moving a point from its current cluster to a potentially new one. If a move has a positive effect, the point is relocated and the two centroids are recomputed. It is not clear that this version is computationally feasible, because the outlined analysis requires an inner loop over all member points of involved clusters affected by centroids shifts. However, in case it is known [Duda & Hart 1973; Berkhin & Becher 2002] that all computations can be algebraically reduced to simply computing a single distance. Therefore, in this case both versions have the same computational complexity.

There is experimental evidence that compared with Forgy’s algorithm, the second (classic) version frequently yields better results [Larsen & Aone 1999; Steinbach et al.2000]

The wide popularity of k-means algorithm is well deserved. It is simple, straightforward, and is based on the firm foundation of analysis of variances. The k-means algorithm also suffers from all the usual suspects:

The result strongly depends on the initial guess of centroids (or assignments)

1. Computed local optimum is known to be a far cry from the global one
2. It is not obvious what is a good k to use
3. The process is sensitive with respect to outliers
4. The algorithm lacks scalability
5. Only numerical attributes are covered
6. Resulting clusters can be unbalanced (in Forgy’s version, even empty)



* 1. **Hierarchical Clustering.**

Hierarchical clustering builds a cluster hierarchy or, in other words, a tree of clusters, also known as a dendrogram. Every cluster node contains child clusters; sibling clusters partition the points covered by their common parent. Such an approach allows exploring data on different levels of granularity. Hierarchical clustering methods are categorized into agglomerative (bottom-up) and divisive (top-down) [Jain & Dubes 1988; Kaufman & Rousseeuw 1990]. An agglomerative clustering starts with one-point (singleton) clusters and recursively merges two or more most appropriate clusters. A divisive clustering starts with one cluster of all data points and recursively splits the most appropriate cluster. The process continues until a stopping criterion (frequently, the requested number k of clusters) is achieved. Advantages of hierarchical clustering include:

1. Embedded flexibility regarding the level of granularity
2. Ease of handling of any forms of similarity or distance
3. Consequently, applicability to any attribute types

Disadvantages of hierarchical clustering are related to:

1. Vagueness of termination criteria
2. The fact that most hierarchical algorithms do not revisit once constructed (intermediate) clusters with the purpose of their improvement

In hierarchical clustering our regular point-by-attribute data representation is sometimes of secondary importance. Instead, hierarchical clustering frequently deals with the matrix of distances (dissimilarities) or similarities between training points. It is sometimes called connectivity matrix. Linkage metrics are constructed (see below) from elements of this matrix. The requirement of keeping such a large matrix in memory is unrealistic. To relax this limitation different devices are used to introduce into the connectivity matrix some sparsely. This can be done by omitting entries smaller than a certain threshold, by using only a certain subset of data representatives, or by keeping with each point only a certain number of its nearest neighbors. For example, nearest neighbor chains have decisive impact on memory consumption [Olson 1995]. A sparse matrix can be further used to represent intuitive concepts of closeness and connectivity.

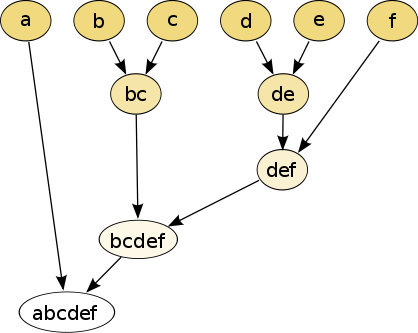
Notice that the way we process original (dis)similarity matrix and construct a linkage metric reflects our a priori ideas about the data model.

**Linkage Metrics**

Hierarchical clustering initializes a cluster system as a set of singleton clusters (agglomerative case) or a single cluster of all points (divisive case) and proceeds iteratively with merging or splitting of the most appropriate cluster(s) until the stopping criterion is achieved. The appropriateness of a cluster(s) for merging/splitting depends on the (dis)similarity of cluster(s) elements. This reflects general presumption that clusters consist of similar points. An important example of dissimilarity between two points is the distance between them.

To merge or split subsets of points rather than individual points, the distance between individual points has to be generalized to the distance between subsets. Such derived proximity measure is called a linkage metric. The type of the linkage metric used significantly affects hierarchical algorithms, since it reflects the particular concept of closeness and connectivity. Major inter-cluster linkage metrics [Murtagh 1985, Olson1995] include single link, average link, and complete link.

Linkage metrics-based hierarchical clustering suffers from time complexity. Under reasonable assumptions, such as reducibility condition (graph methods satisfy this condition), linkage metrics methods have complexity O([Olson 1995]. Despite the unfavorable time complexity, these algorithms are widely used. When the connectivity matrix is sparsified, graph methods directly dealing with the connectivity graph G can be used. In particular, hierarchical divisive MST (MinimumSpanning Tree) algorithm is based on graph partitioning [Jain & Dubes 1988].



* 1. **Clustering Performance and Test Data**

In this section we briefly discusses the evaluation methods and dataset which are used in related researches to evaluate and compare different clustering algorithms.

***Test Data***

According to [8] dataset of (kdd cup) is obtained from [www.kdd.ics.uci.edu](http://www.kdd.ics.uci.edu). This is good dataset to test time series clustering algorithms because Euclidean distance will not be able to achieve perfect accuracy. Another good dataset in order to run on evaluation phase is IRIS and DIABETS dataset which are popular and also available at UCI repository of machine learning[9]. There is also a dataset containing earthquake phenomena which have occurred in Iran in the year of 2008, was selected from collection of datasets of geophysics Institute of Tehran University[10]. This dataset is elected as a type of geo-spatial category of data in order to analyze different algorithms. However, the data set contains the accurate coordinates of Iran’s earthquake events which have been collected by seismographs established across the country.

***Evaluation Metrics***

In general view, clustering evaluation criteria are categorized in three major classes. External Index, Internal Index and Relative Index. External Index is used to measure the extent to which cluster labels match externally supplied class labels. An example of this category is entropy metric. Internal Index is used to measure the goodness of clustering structure without external provided information. The last category is used to compare different clustering or clusters. External and Internal indexes can be used on this type of metrics.

Two of the most popular evaluation criteria on clustering algorithms are*: Inter-cluster distance and Intra-cluster distance*[11]. They are also known as Cohesion and Separation criteria, respectively. Two rules must be applied on above criteria that are maximization of intra cluster distance and minimization of inter cluster distance. There are also other metrics for cluster evaluation such as *Fisher’s separability criterion*, *Minimum Total Distance, Silhouette coefficient, Davies–Bouldin index, Daunn Index. The implementation of each of criterion are ignored and curious researchers might refer to references provided to each criteria.* Fisher criterion provides the ratio of inter-cluster variance to intra-cluster variance. By combining the two mentioned criteria, a more generic criterion is created, which is the simplified form of Fisher criteria [12].

**A) Intra-cluster variance**: Basically, variance measures the distribution of the data objects within a data set around the mean value of that data set and it can be calculated by equation (8).

(8)

In the above equation, *N* represents the number of objects in a data set and is the mean of the objects. This criterion is usually used for measuring the distribution of data objects within a cluster. Thus the average of the variance of the data objects within each cluster is considered as the algorithm’s intra-cluster variance. Henceforward, the intra-cluster variance measure will be referenced as *Var*. So if the result of running clustering method *C,* includes *n* clusters, the value of the intra-cluster variancewill be calculated from equation (9).

*=* (9)

**B) Inter-cluster variance:** For computing the inter-cluster variance of a specific clustering method’s result, the following algorithm was used;

The distance between cluster and is defined as the average distance among all of the data objects within cluster and the centroid of cluster. It can be calculated by equation (10).

(10)

In this equation, *N* represents the number of objects within *ith* cluster. is the centroid of *Jth* cluster which is obviously obtained by: ; *M* is the number of data objects in *jth* cluster.

**C) The ratio of inter-cluster variance to intra-cluster variance:** By combining the two mentioned criteria, a more generic criterion is created, which is the simplified form of the Fisher’s criterion. Suppose that the result of the clustering method *C*, contains *k* clusters *(C1,C2,…,Ck)*, Then the mentioned generic criterion can be calculated by equation (12).

(12)

In equation (12), is the intra-cluster variance of *ith* cluster and is the inter-cluster variance of cluster *i,* which are obtained from the equation (9) and (11). According to this criterion, decreasing the intra-cluster variance will result in decreasing the value of *Vari* and consequently, increasing the value of *f(c)*.

**D) Minimum Total Distance**

In this criterion, we minimize the total of the sum of distances of objects to their cluster centroids and the sum of the distances of the cluster centroids from the global centroid *[36]*. Let a clustering assignment discrete the data set into *m* clusters and *Cj* be one of the clusters. The value for Minimum Total Distance is computed as follows:

Where TD is the Minimum Total Distance for a specific clustering assignment, *Ri*is an object in cluster *Cj*, is the centroid of *Jth* cluster, and GC is the global centroid of the data set. Finally is the distance between and . It is noteable that unlike the Fisher’s criterion, the better clustering answers expect to have a lower number of TD.

**F)** [**Davies–Bouldin index**](https://en.wikipedia.org/wiki/Davies%E2%80%93Bouldin_index)

The [Davies–Bouldin index](https://en.wikipedia.org/wiki/Davies%E2%80%93Bouldin_index) can be calculated by the following formula:


DB = \frac {1} {n} \sum_{i=1}^{n} \max_{j\neq i}\left(\frac{\sigma_i + \sigma_j} {d(c_i,c_j)}\right)


where n is the number of clusters, c_xis the [centroid](https://en.wikipedia.org/wiki/Centroid) of cluster x, \sigma_xis the average distance of all elements in cluster xto centroid c_x, and d(c_i,c_j)is the distance between centroids c_iand c_j. Since algorithms that produce clusters with low intra-cluster distances (high intra-cluster similarity) and high inter-cluster distances (low inter-cluster similarity) will have a low Davies–Bouldin index, the clustering algorithm that produces a collection of clusters with the smallest [Davies–Bouldin index](https://en.wikipedia.org/wiki/Davies%E2%80%93Bouldin_index) is considered the best algorithm based on this criterion.

**G)** [**Dunn index**](https://en.wikipedia.org/wiki/Dunn_index)

The Dunn index aims to identify dense and well-separated clusters. It is defined as the ratio between the minimal inter-cluster distance to maximal intra-cluster distance. For each cluster partition, the Dunn index can be calculated by the following formula: [[32]](https://en.wikipedia.org/wiki/Cluster_analysis#cite_note-32)


D = \frac{\min_{1 \leq i < j \leq n} d(i,j)}{\max_{1 \leq k \leq n} d^{\prime}(k)} \,,


where d(i,j) represents the distance between clusters i and j, and d '(k) measures the intra-cluster distance of cluster k. The inter-cluster distance d(i,j) between two clusters may be any number of distance measures, such as the distance between the [centroids](https://en.wikipedia.org/wiki/Centroids) of the clusters. Similarly, the intra-cluster distance d '(k) may be measured in a variety ways, such as the maximal distance between any pair of elements in cluster k. Since internal criterion seek clusters with high intra-cluster similarity and low inter-cluster similarity, algorithms that produce clusters with high Dunn index are more desirable.

**I)** [**Silhouette coefficient**](https://en.wikipedia.org/wiki/Silhouette_%28clustering%29)

The silhouette coefficient contrasts the average distance to elements in the same cluster with the average distance to elements in other clusters. Objects with a high silhouette value are considered well clustered, objects with a low value may be outliers. This index works well with k-means clustering, and is also used to determine the optimal number of clusters.

Assume the data have been clustered via any technique, such as [k-means](https://en.wikipedia.org/wiki/K-means), into kclusters. For each [datum](https://en.wikipedia.org/wiki/Data) i, let a(i) be the average dissimilarity of iwith all other data within the same cluster. We can interpret a(i) as how well iis assigned to its cluster (the smaller the value, the better the assignment). We then define the average dissimilarity of point ito a cluster cas the average of the distance from ito all points in c.

Let b(i)be the lowest average dissimilarity of ito any other cluster, of which iis not a member. The cluster with this lowest average dissimilarity is said to be the " neighboring cluster" of ibecause it is the next best fit cluster for point i. We now define a silhouette:

s(i) = \frac{b(i) - a(i)}{\max\{a(i),b(i)\}}


* 1. **Outlier Detection**

Outlier detection is to find outliers in a spatial data. One of the approaches in outlier detection is multi-dimensional metric space-based methods and graph-based methods. In the first category, the definition of spatial neighborhood is based on Euclidean distance, while in graph-based spatial outlier detections the definition is based on graph connectivity.

Distribution-based methods consider just the statistical distribution of attribute values, ignoring the spatial relationships among items, density-based approach consider both attribute values and spatial relationship[13].

***1.5.1) Local Outlier Factor:***

One of the convenient algorithm in density based clustering is Local Outlier Factor (LOF), which detects local outliers based on the local density of an object’s neighborhood. We refer to LOF as a method from multi-dimensional metric space-based category of density-based approach. The key difference between LOF approach and existing notions of outliers is that being outlier is not a binary property[14].

According to [14], *k-distance(xi)* provides a measure on the sparsity or density around the object *xi*. When the *k-distance* of *xi* is small, it means that the area around *xi* is dense and vice versa.

***Definition***: (k-distance neighborhood of an object )

The k-distance of xi, the k-distance neighborhood of xi contains every object whose distance from is not greater than the k-distance, i. e.



These objects are called the k-nearest neighbors of.

***Definition***: (reachability distance of an object . object )

*The reachability distance of object xi with respect to object as defined as*

*reach-*(*,* ) = max { *k-distance*(), d(*,* ) }.

If object is far away from , then the reachability distance between the two is simply their actual distance. However, if they are close, the actual distance is replaced by the *k-distance* of .

***Definition***: (local reachability density of )

*The local reachability density of an object is the inverse of the average reachability distance from the k-nearest neighbors of* :

***Definition***: (local outlier factor of )

LOF is the average of the ratios of the local reachability density of *xi* and those of *xi*’s knearest-neighbors. Intuitively, *xi*’s local outlier factor will be very high if its local reachability density is much lower than those of its neighbors[15]. The extension of LOF method is presented in[15], where one method for finding the top-n local outliers in large databases is considered. The strength of that method is it avoids computation of LOF for most objects. And provide users to find only *n* most outstanding local outliers.

The paper is consisted of several major sections. Section 2 is dedicated to discuss on background theories, related works and also recent researches around this subject. Next section (3) explains about the proposed hybrid clustering methods named HAK, its overall procedure and the parameters which effects the algorithm to output better results. Section 4 discusses about clustering methods applied on spatial data (to classify them without supervise), in this section other types of dataset are utilized in order to justify the accuracy, validity and commensurate computational cost of our proposed algorithm. In section 6, some of the most popular evaluation criteria (Fisher’s separability criterion and minimum total distance), Davis Bolden, Silhouette coefficient, and other criteria are introduced, after which the proposed hybrid technique is evaluated on the basis of those criteria. The last section concludes our work and explains about possibilities and ideas to be done later.

1. **Literature Review**

(Reference should be added) K-means clustering requires a specified number of clusters in advance. It needs initial centroids to be selected, randomly. K-means is also sensitive to outliers. Randomly selecting initial start point might affects the quality of output clusters. Therefore, much iteration must be performed of the entire clustering process in order to identify best fitted-clusters. On the other hand, hierarchical clustering cannot well-cluster data with similar pattern. When the size of clusters becomes larger, the cluster actual expression patterns become less relevant. Hierarchical clustering uses dendrogram that provides an easy understanding of the data but it decrease the quality of clusters as more quantity of data is increased.

One of the rigorous advantage of k-mean or other partitioned clustering methods is its lowest time complexity. Therefore, algorithms of this category are fast and are comfortable for high dimensional and large scale datasets due to low time cost allocated to clustering. On the other hand, hierarchical clustering, are more precise and clustered generated from this category are well-clustered. However, due to computation cost of calculating distances between members in different hierarchies. In this paper, we focus on two naïve algorithm of k-mean and accumulative hierarchical clustering, which merits main characters inherited from both approaches.

* 1. **Comparison Between Clustering Techniques**

In this section, strength and weakness of K-means and Hierarchical Clustering are discussed. Comparison K-means and Hierarchical Clustering are also discussed.

* Strengths of K-Mean

1. Simple: - Easy to understand and to implement.
2. Efficient: Time complexity: O(tkn), where n is the number of data points, k is the number of clusters, and t is the number of iterations.
3. Since both k and t are small. K-Means is considered a linear algorithm

* Weaknesses of k-means

1. The algorithm is only applicable if the mean is defined.
2. For categorical data, k-mode - the centroid is represented by most frequent values.
3. The user needs to specify k.
4. the algorithm is sensitive to outliers: Outliers are data points that are very far away from other data points.

* Weaknesses of k-means: To deal with outliers

1. One method is to remove some data points in the clustering process that are much further away from the centroids than other data points. To be safe, we may want to monitor these possible outliers over a few iterations and then decide to remove them.
2. Another method is to perform random sampling. Since in sampling we only choose a small subset of the data points, the chance of selecting an outlier is very small. Assign the rest of the data points to the clusters by distance or similarity comparison, or classification
3. The k-means algorithm is not suitable for discovering clusters that are not hyper-ellipsoids (or hyper-spheres).

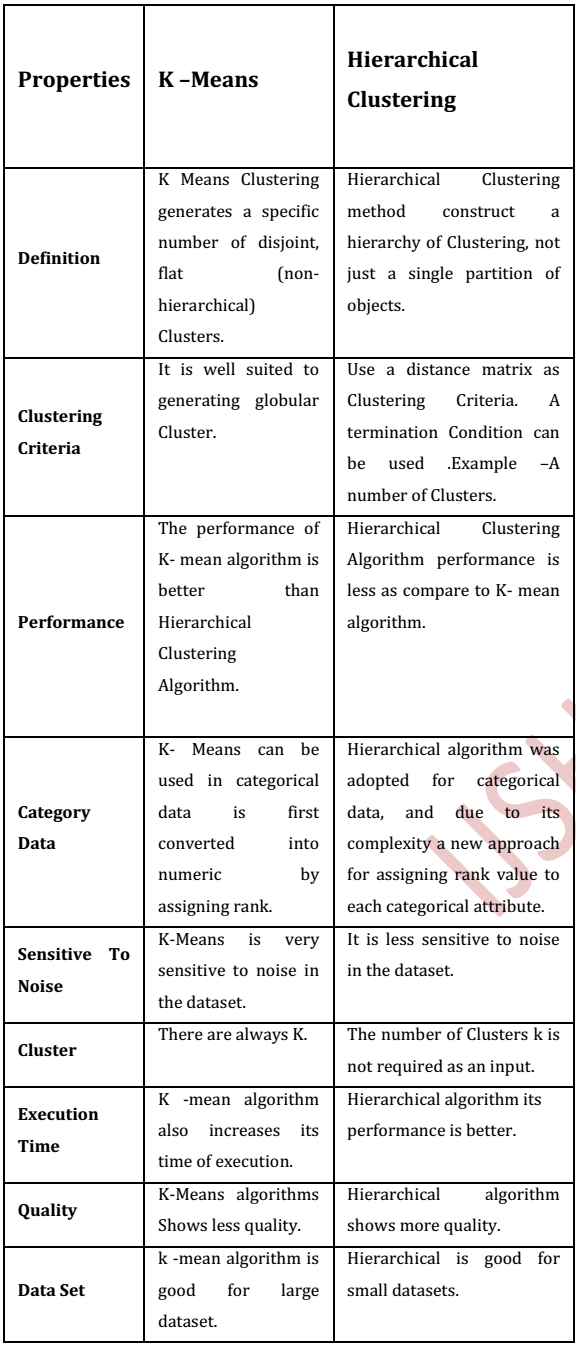
* Strengths of Hierarchical Clustering

1. Conceptually Simple.
2. Theoretical properties are well understood.
3. When Clusters are merged /split, the decision is permanent => the number of different alternatives that need to be examined is reduced.

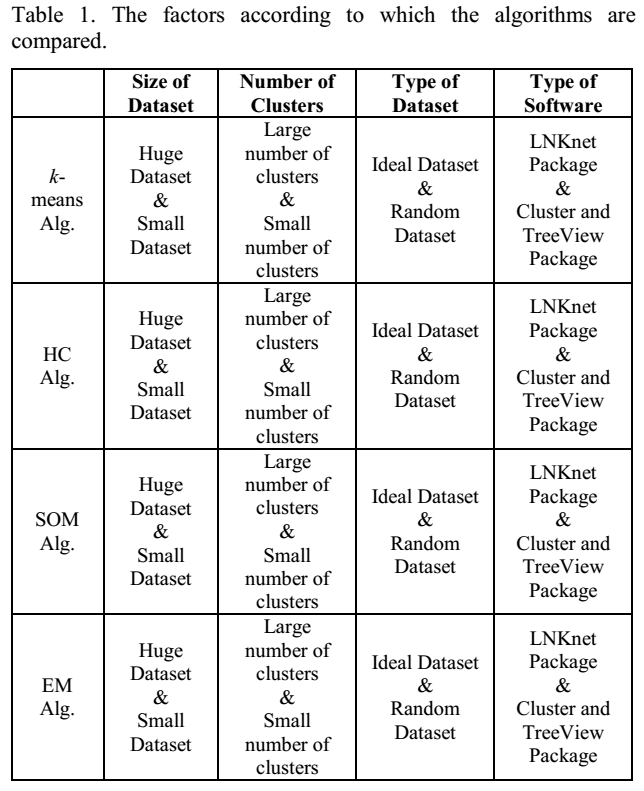
* Weakness of hierarchical Clustering

1. Merging /splitting of clusters is permanent => Erroneous decisions are impossible to correct later.
2. Divisive methods can be computational hard.
3. Methods are not (necessarily) scalable for large datasets.
4. It does not require the number of clusters k in advance.
5. It needs a termination/readout condition. The final mode in both Agglomerative and Divisive is of no use.

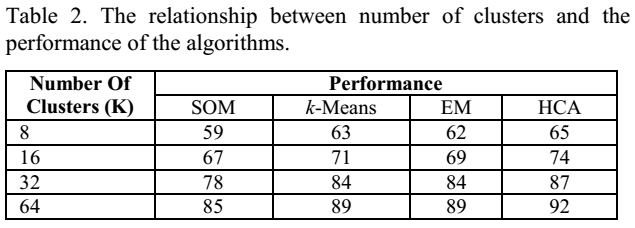
* Comparisons on K-Means and Hierarchical Clustering



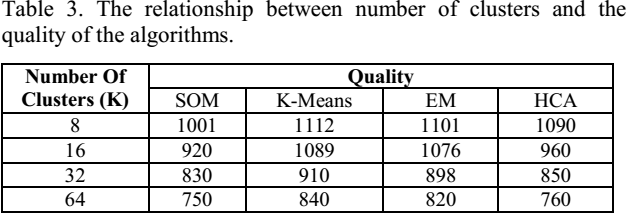
* According to the research [8], different factors were used to compare different clustering algorithms



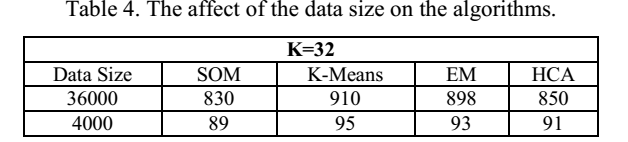
The result shows that when number of cluster increases, the performance of SOM algorithm becomes lower. The performance of k-means and E-M becomes better then hierarchical clustering algorithm.



The result of SOM based Clustering algorithm are more accurate than other algorithm in advance. However, when the number of clusters becomes greater, hierarchical clustering the accuracy becomes better than others.



The quality of K-means and EM becomes very good working with large datasets. On the other hand to other algorithms shows more quality when size of dataset is small.



* 1. **Integration of K-Means and Hierarchical Clustering:**

Cheng-Ru Lin and Ming-Syan Chen [16] have presented a similarity measure, recognized as cohesion, to determine the inter-cluster distances. By employing the cohesion measure, they had developed a two-phase clustering algorithm, known as cohesion based self-merging (CSM), which functions in time linear to the size of input data set. Joining the features of partitioned and hierarchical clustering methods, algorithm CSM divided the input data set into a number of small sub-clusters in the first phase and continuously combined the sub-clusters on the basis of cohesion in a hierarchical manner in the second phase.

Ickjai Lee and Jianhua Yang [17] studied a topology on the basis of merge technique for merging partitioning and hierarchical clustering. The presented merge technique improved the efficiency of partitioning clustering while maintaining the efficiency of hierarchical clustering.

Chen et al. [18] introduced a hybrid approach that appeared diverse from the available method. Initially it executed hierarchical clustering to choose a location and number of clusters in the first round and run the K-means clustering in next round. They cluster around half of the data by hierarchical clustering and accomplish it by K-means for the remaining half in one single round. This hybrid approach produce much better quality on Eisen's yeast microarray data.

[19] presented efficient bottom-up hybrid hierarchical clustering (BHHC) techniques for the use of prototype selection for protein sequence classification. The Order Leader No Update (OLNU) clustering algorithm is used in the first stage to determine number of sub-clusters. In the second stage, either a hierarchical agglomerative clustering (HAC) scheme or a partitioned clustering Algorithm—‘K-medians’ was employed on these sub-cluster representatives to acquire a requisite number of clusters.

The paper uses the early proposed modified k\_Mode algorithm as in[] for the first step of partitioned algorithm followed by implementation of hierarchical algorithm for efficient working of hybrid clustering algorithm. Several clustering algorithms [20] have been proposed to combine the features of these two types of clustering algorithms. In general, these algorithms first partition the input data set into sub-clusters using partitioned clustering algorithm. Then, they construct a hierarchical tree called dendrogram based on these sub-clusters. Based on this hybrid technique, we develop a novel IPHC algorithm that uses the top- q points of sub-clusters as a representative to construct the dendrogram.

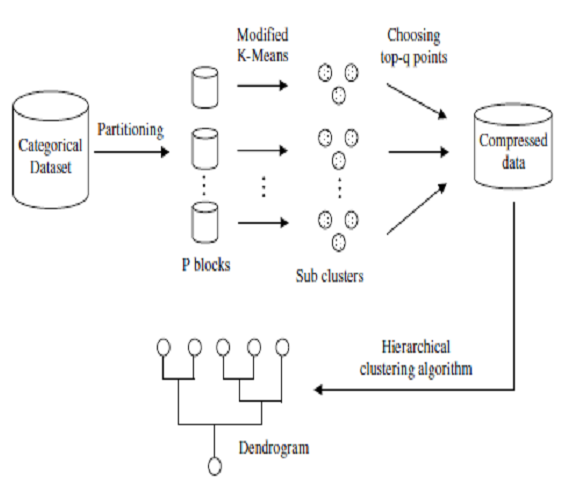


Figure 1) Block diagram of the proposed IPHC Algorithm

The proposed IPHC algorithm has been evaluated with the aid of Nursery Database available in the UCI machine learning repository. Its dataset contains hierarchical decision model originally developed to rank application in nursery school. The dataset consists of 12960 categorical objects with 8 categorical attributes. It partition the 12960 categorical objects into 4 blocks and each block we apply the modified k-means algorithm with k=5. Hence, the proposed IPHC algorithm is validated with the real categorical dataset available in the UCI Machine Learning Repository and the results indicates that the proposed IPHC algorithm is efficient.

In 1999, a new method of hierarchical clustering using dynamic modeling have been proposed called Chameleon [21]. Chameleon is a new agglomerative hierarchical clustering algorithm. Chameleon uses an approach to model the degree of interconnectivity and closeness between each pair of clusters. Chameleon finds the clusters in the data set by using a two-phase algorithm. During the first phase, Chameleon uses a graph-partitioning algorithm to cluster the data items into several relatively small sub-clusters. During the second phase, it uses an algorithm to find the genuine clusters by repeatedly combining these sub-clusters.

Chameleon’s sparse-graph representation of the items is based on the commonly used k-nearest-neighbor graph approach. Each vertex of the k-nearest-neighbor graph represents a data item. An edge exists between two vertices v and u if u is among the k most similar points of v, or v is among the k most similar points of u.



Figure ) Chameleon Algorithm

1. **HAK Algorithm: A Novel Clustering Algorithm Integrated Hierarchical and Partitioned Clustering**

This section describes the proposed HAK algorithm for efficient clustering. The proposed hybrid algorithm combines two popular and well known categories of clustering algorithms (Partitional and Hierarchical algorithm). It provides better clustering results since the advantages of the hierarchical algorithms are the disadvantages of the partitional algorithms, and vice versa. In general, we combines two methods of k-means (partitional category) and AGNES algorithms.

Problems of using K-Means and Hierarchical Clustering are interrelated. K-mean has issue in selecting the initial points and also the number of clusters. However, Hierarchical Clustering suffers from iterating through different levels of dendogram constructing cluster nodes in each hierarchy without knowledge of stop criteria. These intrinsic problems of two approaches motivate us to integrate two algorithm whereas each algorithm compensate the weakness of the other one. Hierarchical clustering is used to estimate the clusters approximately. Then, we introduce stop criteria for Hierarchical clustering. The result of hierarchical clustering is used in order to identify initial point and number of clusters as K-means inputs. However, we use k-means which outperforms the AGNES (Hierarchical clustering Algorithms) as it is mentioned in previous sections. In Overall, K-Means is faster and more accurate in large scale data when distribution of data is flat and symmetric. One of the problems of K-means is its sensitivity to noise and outliers. In order to adjust this problem, we use outlier detection mechanism to identify noise and regulate our cluster centroid for initial points of K-means.

The rough idea for combining both algorithms can be described as follows: first, m iterations of the AGNES algorithm are executed, so some clusters will be found, the execution of the AGNES will be interrupted base on some criteria. As the next step, the result of the AGNES algorithm will be filtered using outlier detection mechanism (LOF introduced in Section 1) to identify and eliminate noise. The filtered result will be passed to κ-means as its initializing inputs (seeds). Then κ-means algorithm will do the rest of the clustering job.

How many AGNES iterations is enough to be run? The answer will solve a significant sub-problem in the issue of combining two mentioned algorithms. It should be noted that executing too many iterations of the AGNES Algorithm will enforce the hybrid algorithm to behave like a pure hierarchical algorithm and as a result, it has its own mentioned disadvantages. On the other hand, if a too few number of the AGNES iterations is executed, clustering results won’t have desirable quality because of non-proper candidate centroids.

Identifying number of iteration in AGNES is another issue. It is also possible to tune m parameter indirectly by manipulating the distance threshold of the AGNES algorithm. The AGNES distance threshold is the maximum inter-cluster distance which is considered as a stop value for the most hierarchical algorithms[22]. Anyway, using this hybrid method, there is no need to specify the initializing parameter(s) of the classic K-means algorithm directly. Actually, the proposed method can be manipulated by means of three parameters which are introduced subsequently. Although initializing these parameters is optional but if they are set wisely, the performance will be improved significantly.

Actually, the proposed method can be manipulated by means of three parameters which are introduced subsequently. Although initializing these parameters is optional but if they are set wisely, the performance will be improved significantly.

1. **Parameter *m*:** Specifies the number of iterations of the AGNES algorithm.
2. **Parameter *T*:** Specifies the AGNES algorithm’s threshold which was defined above.
3. **Parameter *λ*:** Specifies the minimum number of data objects that a cluster should contain to be involved in the κ-means algorithm. In other words, valid clusters must have at least *λ* objects within them.

Actually, the first two parameters will tune the AGNES algorithms and the last one will adjust the κ-means algorithm. Usually, initializing the input parameter of the naive AGNES clustering algorithm requires setting the number of output clusters. The value of this parameter will be equivalent to the difference between the number of entities in dataset and the mentioned parameter *m.* The reason is that the AGNES algorithm will certainly merge two clusters of the dataset in each iteration of execution[23]. Some notable guidelines for specifying the parameter *m* are declared in the following sections.

### Identifying the Upper Bound of Parameter *m*

As already discussed, combining the above-mentioned clustering methods, requires finding an upper bound for parameter m to limit its domain. If the value for m is chosen to be more than a specific threshold, certainly, the proposed method will have more time-space complexity than the classic AGNES algorithm. Identifying an upper bound value for m is considered as an essential requirement for obtaining a rational performance justification for the hybrid approach. So it is recommended that the value of m do not exceeds a calculable threshold. As a rough estimation, let n be the number of data objects in the target clustering data set. In the case of using the naive AGNES clustering method, with centroid inter-cluster distance strategy, running the first iteration of merging the nearest data objects, requires n(n-1)/2 comparisons. Thus in the second iteration (n-1)(n-2)/2 comparisons are needed to select the two nearest data objects. As the worst case scenario for the proposed method, suppose a situation in which an entire κ-means algorithm process is executed immediately after finishing each iteration of the AGNES process. Consequently, [(n)(n-1) /2]+ n comparisons is required in the first iteration of the proposed method. So the following equations can be used as a rough estimation:

Required number of comparisons in the naive AGNES algorithm:

; (3)

Required number of comparisons in hybrid approach (worst case scenario):

1/2

overhead generated by κ-means *n-p+1* iterations of the naive AGNES algorithm

(4)

Equations (3) and (4) are in the form of summation of the products. In equation (3), each product term represents the number of comparisons required in corresponding iteration of the AGNES algorithm. Similarly, in equation (4), each product term represents the number of comparisons needed in the corresponding iteration of proposed hybrid approach. In order to have the computational overhead of the hybrid method be less than the classic AGNES algorithm, a specific number of terms in equation (4) should be computed rather than computing all of the terms. This specific number of terms will be equal to n-p+1.

Let the maximum number of AGNES’ iterations be . As it is obvious in the euation (4),the maximum number of included terms, which is actually equal to the maximum number of iterations(), will be reached, when the value of P is minimized. Let this minimum value for P be . This way, the value for will be obtained by equation (5).

(5)

Including terms of the equation (4), the overhead which is generated by κ-means will be (n-p+1) n. Consequently, the upper bound of parameter m is calculated from inequality (6).

(6)

By expanding the inequality (6), we will obtain inequality (7):

*=>*

*=>*

*=>*

(7)

Now, we can determine the minimum value of p which satisfies the above inequality . By substituting n with a proper integer, is obtained and subsequently,ed by equation (5). It is notable that because of the integer nature of m, there is no need to solve the mentioned third-degree inequality. It means that, it will be solved by means of a simple try-and-error approach. As an example, consider a situation in which there are 648 objects in the target data set (n=648). By substituting n in the inequality (7) the following will be obtained:

6×648× (648-p+1) ≤

By trying P=128, we will obtain: 2025648≤ 1999869, which does not satisfy the inequality. Similarly, by trying P=129 we will obtain: 2021760 ≤ 2047104, which satisfies the inequality. So is the minimum value for the P which satisfies the inequality (7). Subsequently the value of can be calculated by the equation (5) as follows: = n-p+1= 648-129+1=520.Actually, this means that in order to have a rational computational complexity, the number of the AGNES iterations in proposed method, must be less than or equal to .

In the other words, if the number of the AGNES algorithm’s iterations is chosen to be lower than 520 (equivalent to 129 clusters), the proposed method’s computational complexity will be also expected to be lower than the AGNES algorithm’s complexity. Although the proposed algorithm will not force the user to select values which are lower than mmax, but it is notable that disobeying this rule will cause the algorithm to behave as like as its hierarchical parent AGNES. For example, if m=647 is selected, then the algorithm will be transformed into the pure AGNES, so it will lose the benefits declared in section 5.1.

### Identifying the Lower Bound of Parameter *m*

It was previously mentioned that the hybrid algorithm is able to interact with the user. This means that a quality evaluation sub-algorithm will be run to determine the clustering result’s quality according to some criteria which will be declared in section 6. If the user is not satisfied with the clustering result, she/he will increase or decrease the value of parameter m. It is likely that manipulating the value of parameter m leads to a higher quality clustering. Therefore, it is recommended that in the situations when the user has no knowledge about distribution of data, the algorithm be initialized by the starting value of m=2. The value will be increased gradually according to a method introduced in the following sub-section. The lower bound of parameter m varies for different clustering problems, because it directly depends on the distribution of the data objects. So calculating the lower bound for each different problem seems to be a complicated task, nevertheless, finding an accurate lower bound for parameter m is useful to decrease the time complexity of hybrid algorithm. This problem can be the issue for further studies.

### Identifying Outliers in AGNES result

After detecting number of clusters and initial points by AGNES algorithm using parameters defined above, AGNES will detects clusters and its centroid to be passed to K-Means. However, there is also possible that number of outlier clustered have been identified by AGNES algorithm which are also noise to our result and must be eliminated before passing to K-means algorithm, otherwise, it might contaminate the result of K-means. Therefore, it is needed to cleanse the output of AGNES algorithm in one step and then pass the purified out clusters to K-means as initial nodes. Also number of clusters might change during the process of outlier detection.

Explanation of outlier detection

# **6. Evaluating the Algorithm**

This section is mainly devoted to the comparative performance evaluation of the proposed hybrid method, classic AGNES and κ-means algorithms. Actually, comparing two clustering algorithms is a laborious and complicated task and there are various criteria to accomplish this goal. Some of these criteria have single-purpose usages and some others are widely applicable in different domains. Unfortunately there is not any all-purpose clustering algorithm which satisfies all of the existing criteria. Thus the algorithms which perform well from the point of view of a specific criterion often do not perform well from the point of view of another criterion. As discussed later in section 1, there are some criteria elected to analyze our proposed method from different perspectives. Fisher’s separability criterion, Intera-Cluster Distance, Inter Cluster Distance and other evaluation criteria are introduced. Fisher criterion is considered as a widely applicable criterion [34]. At the end of this section, the parent algorithms (AGNES and κ-means) and the proposed hybrid method will be evaluated.

## **6.1. Preparing the Evaluation Prerequisites**

There are two main Prerequisites for evaluating the algorithms: 1) understanding the data set origins and characteristics and 2) a proper clustering evaluation criterion. These two prerequisites are discussed in the following two sub-sections.

Explanation of our dataset:

## **6.2. Evaluating the Parent Algorithms**

The performance issues of the classic AGNES and κ-means algorithms are discussed in this section. The previously introduced criteria have been applied to accomplish this goal. As already mentioned about test data set, this set contains 648 earthquake incident’s coordinates. Each algorithm has been evaluated by f(c) and TD (c) measures. The former represents Fisher’s criterion value and the later is the Minimum Total Distance value for the corresponding algorithm.

### 6.1.1. Evaluating the Naive AGNES Algorithm

Table 1, demonstrates the value of Fisher’s criterion (f(c)) for the various cluster’s quantities in the AGNES algorithm. The average-link strategy has been used as an inter-cluster distance measuring strategy. As the table shows, the maximum value for f(c) and the minimum value for TD(c) has been occurred in the relatively low numbers of clusters and moving toward the higher cluster’s quantities has been resulted in reduction of the value for f(c) and increase of the value for TD(c). In the other words, the more number of clusters we choose, the worse clustering answer will be gained. It is noteworthy that the outliers are merged in the latest iterations of the AGNES algorithm. Consequently, the existence of the outliers among the objects of target data set, may cause deceptive results due to the increasing of f(c) value.

According to the table 1, it can be realized that there are several clustering results which own a relatively high quality and some of them may be preferred based on the domain expert idea. If there are 648 data objects in the data set, then the number of iterations of the naive AGNES algorithm must be lower than 520 (equivalent to 129 clusters) to have a rational computational complexity (see section 5). The related cell for this value is colored in gray and also underlined in the table 2.

### 6.2.2. Evaluating the Naive κ-means Algorithm

As already mentioned, the classic κ-means algorithm, requires at least the number of primary seeds to be initialized. Because the seeds are often selected randomly, running the algorithm for two times and even with the same number of seeds will result in two different answers. In order to gain the more realistic results for comparative evaluation of this method, each different state was executed for 20 times. The f(c) criterion was computed for each state and then the average of the answers (Avg [f(c)]) was considered as the final answer for the method. Table 2, shows the value of Avg[f(c)] for each cluster number. As the table shows, the maximum value for Avg[f(c)] is occurred when the number of primary seeds is chosen to be 5. As a rough realization, it can be mentioned that by increasing the number of clusters Avg[f(c)] has been decreased and Avg[TD(c)] has been increased. So choosing high number of clusters will result in lower overall cluster quality.

### 6.3. Comparative Evaluation

In this section, time and space complexity of proposed hybrid approach are compared to its previously mentioned parents. Finally the results of evaluation are visualized as comparative diagrams. According to the rough estimations which were mentioned in the section 5, if assuming the worst case in which the hybrid algorithm is initialized by m=2 and also it is allowed to execute mmax iterations (mmax is obtained by inequality (6) and equation (7)), the algorithm will have the computational complexity equal to the AGNES complexity. In the other situations in which the value of m is less than mmax, it is expected that the hybrid method’s time complexity is also less than the AGNES complexity. The HAK algorithm has been executed by λ=2 (λ is defined in section 5-2 as a non-essential input parameter of HAK).

### 6.3.1. Comparing HAK with AGNES

### 6.3.2. Comparing HAK with κ-means

# **7. Conclusion and Future Works**

In this paper, the most important considerations and bottlenecks of using hierarchical and partitional clustering techniques discussed. A hybrid approach, which is named HAK, was proposed by combining the naive AGNES and κ-means clustering methods. The proposed hybrid algorithm represents a better quality of clustering rather than κ-means algorithm. Because the proposed method has a lower time complexity than AGNES algorithm, it is expected to be useful in rela-time clustering processes. Totally, the method improves the κ-means algorithm by using the AGNES clustering method for identifying the primary centroids. It is noteworthy that using Silhouette coefficients is another way for improving the κ-means clustering. Comparing HAK with silhouette coefficients approach is planned to be accomplished by the authors as one of the main issues which can improve the research.

The most important motivation for presenting the introduced hybrid approach was generating a moderate method which, unlike the κ-means, does not depend highly on the human user’s knowledge and also has a lower computational complexity than the naive AGNES algorithm. Consequently, the research results reveals that by combining hierarchical and partitional methods, it will be possible to achieve moderate approaches which are more efficient and also do not suffer from their parents’ deficiencies. Obviously, the hybrid approach should also have a relatively desirable clustering quality. According to the results of evaluation, the considerable sensitivity of the proposed hybrid algorithm to the outliers has still remained as an open issue to be dealt with. It seems possible to apply the hybrid method for different types of data (non-spatial data with more dimensions) to test the performance of the method in dealing with discrete variables and also non-numerical data objects.

[1] H. J. Miller, “Geographic data mining and knowledge discovery,” *Handb. Geogr. Inf. Sci.*, pp. 352–366, 2008.

[2] D. Guo, “12 Multivariate Spatial Clustering and Geovisualization,” *Geogr. Data Min. Knowl. Discov.*, p. 325, 2009.

[3] H. Miller and J. Han, “Spatial clustering methods in data mining: a survey,” *Geogr. data Min. Knowl. Discov. Taylor Fr.*, 2001.

[4] J. Han, K. Koperski, and N. Stefanovic, “GeoMiner: a system prototype for spatial data mining,” in *ACM SIGMOD Record*, 1997, vol. 26, no. 2, pp. 553–556.

[5] S. Shekhar, C.-T. Lu, and P. Zhang, “A unified approach to detecting spatial outliers,” *Geoinformatica*, vol. 7, no. 2, pp. 139–166, 2003.

[6] M. Kaushik and B. Mathur, “Comparative Study of K-Means and Hierarchical Clustering Techniques,” pp. 93–98, 2014.

[7] P. Berkhin, “Survey of clustering data mining techniques,” *Group. Multidimens. Data Recent Adv. Clust.*, vol. 10, pp. 25–71, 2006.

[8] O. A. Abbas, “Comparisons Between Data Clustering Algorithms,” *inernational Arab J. Inf. Technol.*, vol. 5, no. 3, pp. 320 – 325, 2008.

[9] M. C. and M. P, “No Title.” UCI Repository of machine learning Databases.

[10] “earthquake dataset.” .

[11] C. Raskutti, B. and Leckie, “An Evaluation of Criteria for Measuring the Quality of Clusters,” *Morgan Kaufmann Publ. Inc*, pp. 905–910, 1999.

[12] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, “On clustering validation techniques,” *J. Intell. Inf. Syst.*, vol. 17, no. 2–3, pp. 107–145, 2001.

[13] S. Shekhar, C. Lu, and P. Zhang, “Detecting Graph-Based Spatial Outliers: Algorithms and Applications(A Summary of Results),” *Proc. seventh ACM SIGKDD*, pp. 371–376, 2001.

[14] S. Cherednichenko, “Outlier detection in clustering,” *Univ. Joensuu*, p. 57, 2005.

[15] W. J. A. K. H. T. J. Han, “Mining top-n local outliers in large databases,” *Proc. seventh ACM SIGKDD Int. Conf. Knowl. Discov. data Min. - KDD ’01*, pp. 293–298, 2001.

[16] C. R. Lin and M. S. Chen, “Combining partitional and hierarchical algorithms for robust and efficient data clustering with cohesion self-merging,” *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 2, pp. 145–159, 2005.

[17] I. Lee and J. Yang, “Voronoi-based topological information for combining partitioning and hierarchical clustering,” *Int. Conf. Comput. Intell. Model. Control Autom. Jointly with Int. Conf. Intell. Agents, Web Technol. Internet Commer. Vol 2, Proc.*, pp. 484–489, 2006.

[18] B. Chen, P. C. Tai, R. Harrison, and Y. Pan, “Novel hybrid hierarchical-K-means clustering method (H-K-means) for microarray analysis,” *2005 IEEE Comput. Syst. Bioinforma. Conf. Work. Poster Abstr.*, pp. 105–108, 2005.

[19] P. A. Vijaya, M. Narasimha Murty, and D. K. Subramanian, “Efficient bottom-up hybrid hierarchical clustering techniques for protein sequence classification,” *Pattern Recognit.*, vol. 39, no. 12, pp. 2344–2355, 2006.

[20] R. Syal, “A Novel Hybrid Clustering Algorithm : Integrated Partitional and Hierarchical Clustering Algorithm for Categorical Data,” no. 5, pp. 138–146, 2012.

[21] G. Karypis, E.-H. Han, and V. Kumar, “Chameleon: hierarchical clustering using dynamic modeling,” *Computer (Long. Beach. Calif).*, vol. 32, no. 8, pp. 68–75, 1999.

[22] H. Jiawei and M. Kamber, *Data mining: concepts and techniques*. 2006.

[23] G. K. Gupta, *Introduction to Data Mining with Case Studies*. Prentice-Hall Of India Pvt. Limited, 2006.

------------------------------------------------------------------------

------------------------------iris------------------------------------------

------------------------------------------------------------------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*KMeans\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

centroids count:3

size of cluster 0:61

size of cluster 1:50

size of cluster 2:39

Incorrectly clustered instances : 17.0 11.3333 %

Average of within cluster variance():0.6494239504511713

Davies–Bouldin(smaller):0.6684333560944665

Dunn(greater):2.4320717428845486

silhouette(greater):0.5493531211013164

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*HAK\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

link type: CENTROID

max iteration: 15

outlier factor: 2

outlier min dense count: 2

outliers count: 88

136:1

141:1

4:1

140:1

5:1

6:1

129:1

131:1

130:1

133:1

13:1

135:1

14:1

134:1

15:1

16:1

18:1

21:1

20:1

23:1

22:1

144:1

145:1

146:1

147:1

26:1

148:1

149:1

31:1

32:1

33:1

36:1

42:1

43:1

41:1

44:1

45:1

51:1

50:1

55:1

54:1

53:1

52:1

59:1

58:1

56:1

62:1

61:1

60:1

68:1

70:1

71:1

64:1

67:1

76:1

77:1

78:1

79:1

72:1

73:1

85:1

87:1

86:1

83:1

89:1

90:1

102:1

103:1

100:1

98:1

110:1

111:1

108:1

109:1

106:1

107:1

104:1

105:1

119:1

118:1

117:1

115:1

114:1

113:1

125:1

124:1

122:1

121:1

middle HAK Centroids-1:10

number of clusters with size of 2:7

number of clusters with size of 3:2

number of clusters with size of 5:1

middle HAK Centroids-2:3

number of clusters with size of 1:1

number of clusters with size of 3:1

number of clusters with size of 6:1

centroids count:3

size of cluster 0:28

size of cluster 1:26

size of cluster 2:8

Incorrectly clustered instances : 20.0 13.3333 %

Average of within cluster variance():0.4499822810158614

Davies–Bouldin(smaller):0.5011557852562679

Dunn(greater):2.5505601761609507

silhouette(greater):0.639770515603812

-------current comparison--------

Cohision(Smaller):HAK(0.4499822810158614) - KMeans(0.6494239504511713) -

daviesBouldin(smaller):HAK(0.5011557852562679) - KMeans(0.6684333560944665) -

dunn(greater):HAK(2.5505601761609507) - KMeans(2.4320717428845486) -

silhouette(greater):HAK(0.639770515603812) - KMeans(0.5493531211013164) -

------------------------------------------------------------------------

------------------------------diabetes------------------------------------------

------------------------------------------------------------------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*KMeans\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

centroids count:2

size of cluster 0:515

size of cluster 1:253

Incorrectly clustered instances : 255.0 33.2031 %

Average of within cluster variance():96.8601208573361

Davies–Bouldin(smaller):5.9621229890106155

Dunn(greater):0.32035921624097713

silhouette(greater):0.05645222485047844

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*HAK\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

link type: CENTROID

max iteration: 15

outlier factor: 5

outlier min dense count: 3

outliers count: 31

684:1

4:1

8:1

9:1

349:2

622:2

193:1

13:1

763:2

459:1

159:1

212:2

453:1

579:1

519:2

39:2

445:1

371:1

370:1

584:2

502:2

228:1

45:2

254:1

357:1

294:2

661:2

58:2

247:2

177:1

537:1

middle HAK Centroids-1:15

number of clusters with size of 2:15

middle HAK Centroids-2:2

number of clusters with size of 1:1

number of clusters with size of 14:1

centroids count:2

size of cluster 0:478

size of cluster 1:259

Incorrectly clustered instances : 268.0 34.8958 %

Average of within cluster variance():88.15449730184497

Davies–Bouldin(smaller):3.980056314418062

Dunn(greater):0.4945644752977074

silhouette(greater):0.057068083818500315

-------current comparison--------

Cohision(Smaller):HAK(88.15449730184497) - KMeans(96.8601208573361) -

daviesBouldin(smaller):HAK(3.980056314418062) - KMeans(5.9621229890106155) -

dunn(greater):HAK(0.4945644752977074) - KMeans(0.32035921624097713) -

silhouette(greater):HAK(0.057068083818500315) - KMeans(0.05645222485047844) -

------------------------------------------------------------------------

------------------------------glass------------------------------------------

------------------------------------------------------------------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*KMeans\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

centroids count:7

size of cluster 0:38

size of cluster 1:24

size of cluster 2:2

size of cluster 3:22

size of cluster 4:94

size of cluster 5:16

size of cluster 6:18

Incorrectly clustered instances : 118.0 55.1402 %

Average of within cluster variance():1.1384264489299334

Davies–Bouldin(smaller):1.6553116281407252

Dunn(greater):0.198117694328386

silhouette(greater):0.16272964412517069

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*HAK\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

link type: CENTROID

max iteration: 40

outlier factor: 10

outlier min dense count: 2

outliers count: 14

102:1

68:1

117:1

32:1

3:1

21:1

4:1

173:1

126:1

24:1

124:1

56:1

46:1

150:1

middle HAK Centroids-1:19

number of clusters with size of 2:10

number of clusters with size of 3:6

number of clusters with size of 4:1

number of clusters with size of 5:1

number of clusters with size of 12:1

middle HAK Centroids-2:7

number of clusters with size of 1:5

number of clusters with size of 2:1

number of clusters with size of 12:1

centroids count:7

size of cluster 0:70

size of cluster 1:19

size of cluster 2:40

size of cluster 3:29

size of cluster 4:18

size of cluster 5:22

size of cluster 6:2

Incorrectly clustered instances : 121.0 56.5421 %

Average of within cluster variance():0.8690685420106934

Davies–Bouldin(smaller):1.5865591217519543

Dunn(greater):0.2266925601574207

silhouette(greater):0.20820991586304827

-------current comparison--------

Cohision(Smaller):HAK(0.8690685420106934) - KMeans(1.1384264489299334) -

daviesBouldin(smaller):HAK(1.5865591217519543) - KMeans(1.6553116281407252) -

dunn(greater):HAK(0.2266925601574207) - KMeans(0.198117694328386) -

silhouette(greater):HAK(0.20820991586304827) - KMeans(0.16272964412517069) -

------------------------------------------------------------------------

------------------------------ionosphere------------------------------------------

------------------------------------------------------------------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*KMeans\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

centroids count:2

size of cluster 0:190

size of cluster 1:161

Incorrectly clustered instances : 101.0 28.7749 %

Average of within cluster variance():2.3265048113368674

Davies–Bouldin(smaller):1.5134759507430136

Dunn(greater):1.0177164042660343

silhouette(greater):0.2959701497777326

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*HAK\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

link type: CENTROID

max iteration: 15

outlier factor: 5

outlier min dense count: 3

outliers count: 196

0:1

1:1

2:1

3:1

4:1

6:1

7:1

9:1

11:1

13:1

14:2

15:1

17:1

19:1

18:1

21:1

23:1

25:1

27:1

26:1

29:1

28:1

31:1

30:1

34:1

35:1

33:1

39:1

37:1

42:1

43:1

41:1

47:1

45:1

51:1

50:1

49:1

55:1

53:1

52:1

59:1

57:1

56:1

63:1

61:1

60:2

68:1

69:1

70:1

71:1

64:1

65:1

67:1

77:1

78:1

79:1

73:1

75:1

85:1

87:1

81:1

80:1

83:1

93:1

92:1

95:1

94:2

89:1

91:1

90:1

102:2

100:1

98:1

110:1

108:1

109:1

106:1

104:1

105:1

118:1

116:1

114:1

112:1

126:1

125:1

124:1

123:2

122:1

121:1

120:1

137:2

136:1

139:1

140:1

142:1

128:1

130:1

133:1

132:1

134:1

157:1

159:2

146:1

148:1

150:1

170:1

169:1

168:1

175:1

174:1

173:1

172:1

162:1

167:1

166:1

164:1

186:1

187:1

184:1

190:1

191:1

188:1

189:1

176:1

182:1

183:1

180:1

181:1

204:1

206:1

200:1

202:1

197:1

196:1

198:1

193:1

192:1

195:1

194:1

220:1

221:1

222:1

216:1

218:1

212:1

213:1

214:1

208:1

209:1

210:1

211:2

236:1

235:1

234:1

232:1

230:1

229:1

228:1

226:1

224:1

254:2

252:1

253:1

250:1

248:2

249:1

246:1

244:1

242:1

240:1

272:1

286:2

285:1

257:1

266:1

267:1

271:1

268:1

305:1

307:1

306:1

311:2

310:2

313:1

317:1

290:1

291:2

294:2

298:1

300:2

340:1

338:2

345:1

327:1

323:2

328:1

middle HAK Centroids-1:5

number of clusters with size of 2:3

number of clusters with size of 5:1

number of clusters with size of 9:1

middle HAK Centroids-2:2

number of clusters with size of 2:1

number of clusters with size of 3:1

centroids count:2

size of cluster 0:55

size of cluster 1:100

Incorrectly clustered instances : 115.0 32.7635 %

Average of within cluster variance():1.6330318482056727

Davies–Bouldin(smaller):0.9174586530136186

Dunn(greater):1.5845760446437338

silhouette(greater):0.5174633524981576

-------current comparison--------

Cohision(Smaller):HAK(1.6330318482056727) - KMeans(2.3265048113368674) -

daviesBouldin(smaller):HAK(0.9174586530136186) - KMeans(1.5134759507430136) -

dunn(greater):HAK(1.5845760446437338) - KMeans(1.0177164042660343) -

silhouette(greater):HAK(0.5174633524981576) - KMeans(0.2959701497777326) -

------------------------------------------------------------------------

------------------------------segment-challenge------------------------------------------

------------------------------------------------------------------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*KMeans\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

centroids count:7

size of cluster 0:210

size of cluster 1:207

size of cluster 2:220

size of cluster 3:235

size of cluster 4:248

size of cluster 5:197

size of cluster 6:183

Incorrectly clustered instances : 501.0 33.4 %

Average of within cluster variance():70.97030607216573

Davies–Bouldin(smaller):1.638285946592566

Dunn(greater):0.47934351873811615

silhouette(greater):0.2088774256479497

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*HAK\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

link type: CENTROID

max iteration: 100

outlier factor: 2

outlier min dense count: 2

outliers count: 1311

0:1

1:1

2:1

4:1

5:1

6:1

7:1

8:1

9:1

10:1

11:1

12:1

13:1

14:1

15:1

17:1

16:1

18:1

21:1

20:1

23:1

22:1

24:1

27:1

29:1

28:1

31:1

30:1

34:1

35:1

32:1

33:1

38:1

39:1

36:1

37:1

42:1

43:1

40:1

47:1

44:1

45:1

51:1

50:1

48:1

55:1

54:1

53:1

52:1

59:1

58:1

57:1

56:1

62:1

61:1

60:1

68:1

69:1

70:1

71:1

64:1

65:1

66:1

67:1

76:1

78:1

79:1

73:1

74:1

75:1

85:1

84:1

87:1

81:1

80:1

83:1

82:1

93:1

92:1

95:1

94:1

89:1

88:1

91:1

90:1

102:1

103:1

100:1

101:1

98:1

99:1

96:1

97:1

110:1

111:1

108:1

109:1

106:1

107:1

104:1

105:1

119:1

118:1

117:1

116:1

115:1

114:1

113:1

112:1

127:1

126:1

125:1

124:1

123:1

122:1

120:1

137:1

136:1

139:1

138:1

141:1

140:1

143:1

142:1

128:1

131:1

130:1

133:1

132:1

134:1

153:1

154:1

157:1

158:1

159:1

145:1

148:1

149:1

150:1

151:1

171:1

170:1

169:1

168:1

175:1

174:1

173:1

172:1

163:1

162:1

161:1

160:1

167:1

165:1

164:1

186:1

187:1

184:1

185:1

190:1

191:1

188:1

189:1

178:1

179:1

176:1

177:1

183:1

180:1

181:1

205:1

204:1

207:1

206:1

201:1

200:1

203:1

202:1

197:1

196:1

199:1

198:1

193:1

192:1

195:1

194:1

220:1

222:1

223:1

216:1

217:1

218:1

212:1

213:1

214:1

210:1

211:1

239:1

238:1

237:1

236:1

235:1

234:1

233:1

232:1

231:1

230:1

229:1

228:1

227:1

226:1

225:1

224:1

254:1

255:1

252:1

253:1

250:1

251:1

248:1

249:1

247:1

244:1

245:1

242:1

243:1

240:1

275:1

274:1

273:1

272:1

279:1

278:1

277:1

276:1

283:1

282:1

281:1

280:1

287:1

286:1

285:1

284:1

258:1

259:1

257:1

262:1

263:1

260:1

261:1

266:1

267:1

265:1

270:1

271:1

268:1

269:1

305:1

307:1

306:1

309:1

308:1

311:1

310:1

313:1

312:1

314:1

317:1

316:1

319:1

318:1

288:1

289:1

290:1

291:1

292:1

293:1

295:1

296:1

297:1

298:1

299:1

300:1

301:1

302:1

303:1

343:1

342:1

341:1

340:1

339:1

338:1

337:1

351:1

350:1

349:1

348:1

347:1

346:1

345:1

344:1

326:1

327:1

325:1

322:1

320:1

334:1

335:1

332:1

330:1

331:1

328:1

329:1

373:1

372:1

375:1

374:1

369:1

368:1

370:1

381:1

380:1

383:1

382:1

377:1

376:1

378:1

356:1

357:1

358:1

359:1

352:1

353:1

354:1

355:1

364:1

365:1

366:1

367:1

361:1

362:1

363:1

410:1

411:1

409:1

414:1

415:1

412:1

402:1

403:1

400:1

407:1

404:1

405:1

395:1

394:1

393:1

392:1

399:1

398:1

397:1

396:1

387:1

386:1

385:1

384:1

391:1

390:1

389:1

440:1

441:1

442:1

443:1

444:1

445:1

446:1

447:1

432:1

433:1

434:1

435:1

436:1

437:1

438:1

439:1

425:1

427:1

426:1

429:1

428:1

431:1

430:1

417:1

416:1

419:1

418:1

421:1

420:1

423:1

422:1

478:1

479:1

476:1

477:1

474:1

475:1

472:1

473:1

470:1

471:1

468:1

464:1

465:1

463:1

462:1

461:1

460:1

459:1

458:1

456:1

455:1

454:1

453:1

452:1

451:1

450:1

449:1

448:1

508:1

509:1

510:1

511:1

505:1

506:1

507:1

500:1

501:1

502:1

503:1

496:1

497:1

493:1

492:1

495:1

494:1

489:1

488:1

491:1

490:1

485:1

484:1

487:1

486:1

480:1

483:1

482:1

550:1

551:1

548:1

549:1

546:1

547:1

544:1

545:1

558:1

559:1

556:1

557:1

554:1

555:1

552:1

553:1

567:1

566:1

565:1

564:1

563:1

562:1

561:1

560:1

575:1

574:1

573:1

572:1

571:1

570:1

568:1

517:1

518:1

519:1

512:1

514:1

515:1

524:1

525:1

526:1

527:1

520:1

522:1

523:1

533:1

535:1

534:1

529:1

528:1

531:1

530:1

541:1

540:1

543:1

542:1

537:1

536:1

539:1

538:1

610:1

608:1

609:1

615:1

612:1

613:1

618:1

619:1

617:1

622:1

623:1

620:1

621:1

627:1

625:1

624:1

631:1

630:1

629:1

628:1

635:1

634:1

633:1

632:1

639:1

638:1

636:1

576:1

578:1

579:1

580:1

581:1

582:1

583:1

584:1

585:1

586:1

589:1

590:1

591:1

593:1

592:1

595:1

594:1

597:1

596:1

599:1

598:1

600:1

603:1

602:1

604:1

607:1

606:1

687:1

686:1

684:1

683:1

682:1

681:1

680:1

677:1

676:1

675:1

674:1

673:1

672:1

702:1

700:1

701:1

699:1

696:1

697:1

694:1

695:1

692:1

693:1

690:1

691:1

688:1

653:1

652:1

654:1

649:1

648:1

651:1

650:1

645:1

644:1

647:1

646:1

640:1

643:1

642:1

668:1

669:1

670:1

671:1

665:1

666:1

667:1

660:1

661:1

662:1

656:1

657:1

658:1

747:1

746:1

745:1

744:1

751:1

750:1

749:1

739:1

738:1

737:1

736:1

743:1

742:1

741:1

740:1

763:1

760:1

761:1

766:1

767:1

764:1

765:1

754:1

752:1

753:1

758:1

759:1

756:1

757:1

713:1

712:1

714:1

717:1

716:1

719:1

705:1

704:1

706:1

709:1

708:1

711:1

710:1

728:1

729:1

730:1

731:1

733:1

734:1

735:1

721:1

722:1

723:1

724:1

725:1

726:1

727:1

821:1

820:1

823:1

822:1

817:1

819:1

829:1

828:1

831:1

830:1

825:1

824:1

827:1

826:1

804:1

805:1

807:1

800:1

801:1

802:1

812:1

813:1

814:1

815:1

808:1

809:1

810:1

811:1

791:1

790:1

789:1

788:1

787:1

786:1

784:1

799:1

798:1

797:1

796:1

795:1

793:1

792:1

774:1

775:1

772:1

773:1

770:1

768:1

769:1

783:1

780:1

781:1

778:1

779:1

776:1

881:1

880:1

882:1

885:1

884:1

887:1

886:1

889:1

888:1

891:1

893:1

892:1

895:1

894:1

864:1

865:1

866:1

867:1

868:1

869:1

870:1

871:1

872:1

873:1

874:1

875:1

877:1

878:1

879:1

851:1

850:1

849:1

848:1

855:1

854:1

853:1

859:1

858:1

857:1

856:1

863:1

861:1

860:1

834:1

832:1

833:1

839:1

836:1

837:1

842:1

843:1

840:1

841:1

846:1

847:1

844:1

956:1

958:1

959:1

952:1

953:1

954:1

955:1

948:1

949:1

951:1

944:1

945:1

946:1

947:1

941:1

940:1

943:1

942:1

937:1

936:1

939:1

938:1

933:1

932:1

935:1

934:1

929:1

928:1

931:1

930:1

926:1

927:1

924:1

925:1

922:1

923:1

921:1

919:1

916:1

917:1

915:1

912:1

913:1

911:1

910:1

909:1

908:1

907:1

906:1

905:1

904:1

903:1

902:1

900:1

899:1

898:1

897:1

896:1

1016:1

1017:1

1018:1

1019:1

1020:1

1021:1

1022:1

1023:1

1008:1

1009:1

1010:1

1011:1

1012:1

1013:1

1014:1

1015:1

1000:1

1003:1

1002:1

1005:1

1007:1

1006:1

992:1

995:1

994:1

997:1

996:1

999:1

998:1

986:1

987:1

984:1

990:1

991:1

988:1

989:1

978:1

979:1

976:1

982:1

983:1

980:1

981:1

971:1

970:1

969:1

968:1

975:1

974:1

973:1

972:1

962:1

961:1

967:1

965:1

964:1

1101:1

1102:1

1103:1

1096:1

1097:1

1098:1

1099:1

1092:1

1093:1

1094:1

1095:1

1088:1

1090:1

1117:1

1119:1

1118:1

1113:1

1112:1

1115:1

1109:1

1108:1

1111:1

1110:1

1105:1

1104:1

1107:1

1106:1

1134:1

1135:1

1132:1

1133:1

1130:1

1128:1

1129:1

1126:1

1127:1

1124:1

1122:1

1123:1

1120:1

1121:1

1151:1

1150:1

1149:1

1148:1

1147:1

1146:1

1145:1

1143:1

1142:1

1141:1

1140:1

1139:1

1138:1

1137:1

1136:1

1032:1

1033:1

1034:1

1035:1

1036:1

1037:1

1038:1

1039:1

1024:1

1025:1

1026:1

1027:1

1029:1

1030:1

1031:1

1049:1

1048:1

1051:1

1050:1

1053:1

1052:1

1055:1

1054:1

1040:1

1043:1

1042:1

1045:1

1044:1

1047:1

1046:1

1066:1

1064:1

1071:1

1068:1

1069:1

1058:1

1059:1

1056:1

1057:1

1062:1

1063:1

1060:1

1083:1

1082:1

1081:1

1080:1

1087:1

1086:1

1085:1

1084:1

1075:1

1074:1

1073:1

1072:1

1079:1

1078:1

1077:1

1076:1

1221:1

1220:1

1223:1

1216:1

1219:1

1218:1

1229:1

1231:1

1230:1

1225:1

1224:1

1227:1

1226:1

1237:1

1238:1

1239:1

1232:1

1233:1

1234:1

1235:1

1244:1

1245:1

1246:1

1247:1

1240:1

1241:1

1242:1

1243:1

1255:1

1253:1

1252:1

1251:1

1249:1

1248:1

1262:1

1261:1

1260:1

1259:1

1258:1

1257:1

1256:1

1271:1

1268:1

1269:1

1266:1

1264:1

1265:1

1279:1

1276:1

1277:1

1275:1

1273:1

1153:1

1152:1

1154:1

1156:1

1160:1

1162:1

1165:1

1164:1

1167:1

1166:1

1168:1

1170:1

1171:1

1172:1

1173:1

1174:1

1176:1

1177:1

1178:1

1179:1

1180:1

1181:1

1182:1

1183:1

1187:1

1186:1

1185:1

1184:1

1191:1

1190:1

1189:1

1188:1

1195:1

1194:1

1193:1

1199:1

1198:1

1197:1

1202:1

1203:1

1200:1

1206:1

1207:1

1204:1

1210:1

1208:1

1209:1

1214:1

1215:1

1212:1

1213:1

1373:1

1372:1

1371:1

1369:1

1367:1

1366:1

1365:1

1364:1

1363:1

1362:1

1361:1

1360:1

1358:1

1359:1

1357:1

1354:1

1355:1

1352:1

1353:1

1350:1

1351:1

1348:1

1349:1

1346:1

1347:1

1345:1

1405:1

1404:1

1407:1

1406:1

1401:1

1400:1

1403:1

1402:1

1397:1

1396:1

1398:1

1393:1

1392:1

1395:1

1394:1

1388:1

1389:1

1390:1

1391:1

1384:1

1386:1

1387:1

1380:1

1381:1

1382:1

1376:1

1377:1

1378:1

1379:1

1307:1

1306:1

1305:1

1304:1

1311:1

1309:1

1308:1

1299:1

1298:1

1296:1

1303:1

1302:1

1301:1

1300:1

1291:1

1288:1

1289:1

1294:1

1295:1

1292:1

1293:1

1283:1

1280:1

1281:1

1286:1

1287:1

1284:1

1285:1

1337:1

1336:1

1339:1

1338:1

1341:1

1340:1

1343:1

1342:1

1329:1

1328:1

1330:1

1333:1

1332:1

1335:1

1334:1

1320:1

1321:1

1323:1

1324:1

1325:1

1326:1

1327:1

1312:1

1313:1

1315:1

1316:1

1317:1

1318:1

1494:1

1495:1

1492:1

1493:1

1490:1

1491:1

1488:1

1498:1

1499:1

1496:1

1497:1

1479:1

1478:1

1477:1

1476:1

1475:1

1473:1

1472:1

1487:1

1486:1

1482:1

1481:1

1480:1

1426:1

1427:1

1424:1

1425:1

1430:1

1431:1

1428:1

1434:1

1435:1

1432:1

1433:1

1438:1

1439:1

1436:1

1437:1

1410:1

1409:1

1408:1

1414:1

1413:1

1419:1

1418:1

1417:1

1416:1

1423:1

1422:1

1421:1

1420:1

1456:1

1457:1

1458:1

1459:1

1460:1

1461:1

1462:1

1463:1

1464:1

1465:1

1466:1

1467:1

1468:1

1469:1

1470:1

1471:1

1441:1

1440:1

1443:1

1442:1

1445:1

1444:1

1447:1

1446:1

1448:1

1451:1

1450:1

1452:1

1455:1

middle HAK Centroids-1:98

number of clusters with size of 2:96

number of clusters with size of 3:2

middle HAK Centroids-2:7

number of clusters with size of 1:2

number of clusters with size of 2:1

number of clusters with size of 4:1

number of clusters with size of 5:1

number of clusters with size of 10:1

number of clusters with size of 75:1

centroids count:7

size of cluster 0:96

size of cluster 1:20

size of cluster 2:6

size of cluster 3:10

size of cluster 4:8

size of cluster 5:2

size of cluster 6:47

Incorrectly clustered instances : 809.0 53.9333 %

Average of within cluster variance():59.7876068325631

Davies–Bouldin(smaller):1.492324263894213

Dunn(greater):0.781048075970198

silhouette(greater):0.09853979868730835

-------current comparison--------

Cohision(Smaller):HAK(59.7876068325631) - KMeans(70.97030607216573) -

daviesBouldin(smaller):HAK(1.492324263894213) - KMeans(1.638285946592566) -

dunn(greater):HAK(0.781048075970198) - KMeans(0.47934351873811615) -

silhouette(greater):KMeans(0.2088774256479497) - HAK(0.09853979868730835) -

------------------------------------------------------------------------

------------------------------segment-test------------------------------------------

------------------------------------------------------------------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*KMeans\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

centroids count:7

size of cluster 0:125

size of cluster 1:157

size of cluster 2:110

size of cluster 3:130

size of cluster 4:120

size of cluster 5:132

size of cluster 6:36

Incorrectly clustered instances : 260.0 32.0988 %

Average of within cluster variance():72.0569362790567

Davies–Bouldin(smaller):1.7777039917247086

Dunn(greater):0.3859709402950007

silhouette(greater):0.21428451819485916

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*HAK\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

link type: CENTROID

max iteration: 100

outlier factor: 10

outlier min dense count: 3

outliers count: 135

551:1

272:1

279:1

544:1

545:1

276:1

281:1

14:1

285:1

567:2

565:2

257:1

803:1

265:1

570:2

569:1

34:1

35:2

304:1

789:2

309:1

36:2

798:1

795:1

794:1

533:1

51:2

289:2

535:1

49:1

773:2

770:1

531:1

294:2

63:1

60:2

343:2

71:2

65:1

351:2

619:2

349:2

347:1

346:2

85:1

326:1

82:1

638:2

88:2

331:2

373:1

580:1

369:2

99:1

368:1

97:1

381:2

380:2

377:2

107:2

376:1

104:1

119:1

595:1

354:1

598:2

127:1

366:2

124:2

361:1

607:1

606:1

137:1

415:2

143:2

680:1

142:1

673:2

135:1

152:1

699:1

157:1

695:1

147:2

171:1

170:2

441:1

169:1

174:2

173:1

172:2

447:1

165:1

425:1

426:1

429:1

428:2

430:1

418:1

421:1

183:1

473:1

470:1

471:2

469:1

192:2

195:1

221:2

766:2

767:1

755:1

213:2

214:2

215:1

209:2

450:1

449:1

448:2

508:1

715:2

714:2

236:2

505:1

707:2

228:1

496:2

710:2

728:2

493:2

495:1

731:2

253:2

489:2

723:2

242:2

middle HAK Centroids-1:87

number of clusters with size of 2:76

number of clusters with size of 3:9

number of clusters with size of 4:2

middle HAK Centroids-2:7

number of clusters with size of 1:4

number of clusters with size of 66:1

number of clusters with size of 7:1

number of clusters with size of 10:1

centroids count:5

size of cluster 0:322

size of cluster 1:106

size of cluster 2:101

size of cluster 3:16

size of cluster 4:130

Incorrectly clustered instances : 345.0 42.5926 %

Average of within cluster variance():75.75298685029567

Davies–Bouldin(smaller):1.6607918735241198

Dunn(greater):0.8596630433689769

silhouette(greater):0.2732213193856892

-------current comparison--------

Cohision(Smaller):KMeans(72.0569362790567) - HAK(75.75298685029567) -

daviesBouldin(smaller):HAK(1.6607918735241198) - KMeans(1.7777039917247086) -

dunn(greater):HAK(0.8596630433689769) - KMeans(0.3859709402950007) -

silhouette(greater):HAK(0.2732213193856892) - KMeans(0.21428451819485916) -

------------------------------------------------------------------------

------------------------------unbalanced------------------------------------------

------------------------------------------------------------------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*KMeans\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

centroids count:2

size of cluster 0:434

size of cluster 1:422

Incorrectly clustered instances : 428.0 50 %

Average of within cluster variance():68.69045404490518

Davies–Bouldin(smaller):3.8786606740801903

Dunn(greater):0.5076528917094687

silhouette(greater):0.03216109374522844

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*HAK\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

link type: CENTROID

max iteration: 30

outlier factor: 4

outlier min dense count: 3

outliers count: 791

0:1

1:1

2:1

3:1

4:2

5:1

6:1

7:1

8:1

9:2

10:1

11:1

12:1

13:1

14:1

15:1

17:1

16:1

19:1

18:1

21:2

20:1

23:1

22:1

25:1

24:1

27:2

26:2

29:1

28:1

31:1

30:1

35:1

33:1

38:1

39:1

36:1

37:1

42:1

43:1

40:2

41:2

46:1

47:1

44:1

45:1

51:1

50:1

49:1

48:1

55:1

54:1

53:1

52:1

59:1

58:1

57:1

56:1

63:1

62:1

61:1

60:1

68:2

69:1

70:1

71:2

64:1

65:1

66:1

67:1

76:1

78:2

72:1

73:1

74:1

75:1

84:2

87:1

86:1

81:1

83:2

82:1

95:2

89:1

91:1

90:1

102:1

103:1

100:1

101:1

98:1

99:1

97:2

110:2

111:1

108:2

109:1

106:1

107:1

104:1

105:2

119:1

118:1

117:2

116:2

115:1

114:2

127:1

126:1

125:1

124:1

123:1

122:1

121:2

120:1

137:2

136:1

139:1

138:2

141:2

140:1

143:1

142:1

129:1

128:1

131:1

130:1

133:1

132:1

135:1

134:1

152:1

153:1

154:2

155:1

156:1

157:1

158:1

159:1

144:1

145:1

146:1

147:2

148:1

149:1

151:1

171:2

170:1

169:2

168:1

175:1

173:2

172:1

163:1

162:2

161:1

160:1

167:1

166:2

165:1

164:1

186:1

187:1

185:2

189:1

178:1

179:2

176:1

177:1

182:1

183:1

180:1

181:1

205:1

204:2

207:1

206:1

201:1

200:2

203:1

202:1

196:2

199:1

198:2

220:1

221:1

222:1

223:1

216:1

217:1

218:1

219:1

212:2

213:1

214:1

215:1

208:2

209:2

210:2

211:1

239:1

238:1

237:1

236:1

235:1

234:1

233:1

232:1

231:1

230:1

229:1

228:1

227:1

226:1

254:1

255:1

252:1

250:1

251:2

248:1

249:1

246:1

247:1

244:1

245:1

242:1

243:1

240:1

241:1

275:1

274:1

273:1

272:1

279:1

278:1

277:1

276:2

283:1

282:2

281:2

280:1

287:1

286:1

284:1

258:1

259:1

256:1

262:1

263:2

260:1

261:1

266:2

265:1

270:1

271:1

269:1

305:1

307:1

306:1

309:2

308:1

311:1

310:2

313:1

312:2

315:1

314:1

317:2

316:1

319:2

318:1

288:2

289:1

290:2

292:1

293:1

294:1

295:2

296:1

297:1

298:2

299:1

300:1

301:2

302:1

303:2

343:2

342:1

341:1

340:1

339:1

338:1

337:1

336:2

351:1

350:1

349:1

348:1

347:1

346:1

345:1

344:2

326:1

327:1

324:1

325:1

322:1

323:1

320:1

321:1

335:1

332:2

333:1

328:1

329:1

373:1

372:1

375:2

374:2

369:1

368:1

381:1

380:1

383:1

382:1

377:2

376:1

379:1

378:1

356:1

357:1

358:2

359:1

352:1

353:1

354:1

355:1

364:1

365:1

366:1

367:1

360:1

361:1

362:2

363:1

410:1

411:2

408:2

409:1

415:2

413:2

402:1

403:1

400:2

401:1

406:1

407:2

404:2

405:1

395:1

394:1

393:1

392:1

399:2

398:1

397:1

396:1

387:2

386:1

385:1

391:1

390:1

389:2

388:1

440:1

442:1

443:2

444:1

445:1

446:1

447:1

432:1

433:1

434:2

435:2

436:1

437:1

438:1

439:1

425:1

424:1

427:1

426:1

429:1

428:1

431:2

430:1

417:1

416:1

419:1

418:1

421:1

420:1

423:1

422:1

478:1

479:2

476:1

477:1

474:1

475:1

472:1

473:2

470:2

471:1

468:1

469:2

466:1

467:1

464:1

465:1

463:1

462:1

461:1

460:1

459:1

458:1

457:1

456:1

455:1

454:1

453:1

452:1

451:1

450:1

449:2

448:1

508:1

509:1

510:1

511:1

505:1

506:2

507:2

500:1

501:1

502:1

503:1

496:1

497:1

498:1

499:1

493:1

492:2

495:1

494:1

489:1

488:1

491:2

490:1

485:2

484:1

487:1

486:2

481:2

480:1

483:1

482:2

550:1

551:2

548:2

549:1

546:2

547:1

544:1

545:1

558:2

559:1

556:1

557:1

554:1

555:1

552:1

553:1

567:1

566:1

565:1

564:1

563:2

562:1

561:1

560:2

575:1

574:2

573:1

572:1

571:2

570:1

569:1

568:1

516:1

517:1

518:1

519:1

513:1

514:2

515:2

524:2

525:1

526:2

527:1

520:1

521:1

523:1

533:1

532:1

534:1

529:1

528:2

530:1

541:2

540:2

543:2

542:1

537:2

539:1

538:2

610:1

611:1

608:1

609:1

614:1

615:1

612:1

613:1

618:1

619:1

616:1

617:1

622:1

623:1

620:1

621:2

626:2

625:1

624:1

631:1

630:1

629:1

628:1

635:1

634:1

633:1

632:1

639:1

638:2

637:1

636:1

577:2

578:1

579:1

580:1

581:1

582:2

583:2

584:2

585:1

586:1

587:1

589:1

590:1

592:1

595:2

594:2

597:1

599:2

598:1

601:2

600:2

603:1

602:1

605:1

604:1

607:1

606:1

687:1

686:1

685:2

684:1

683:1

682:1

681:1

680:1

679:1

678:1

677:1

676:1

675:1

674:1

673:2

672:2

702:1

703:1

700:1

701:1

698:1

699:1

696:1

697:1

694:1

695:1

692:2

693:1

690:1

691:1

688:1

689:1

653:1

652:1

655:1

654:1

649:1

648:1

651:1

650:1

645:1

644:1

647:1

646:1

641:1

640:2

643:1

642:1

668:1

669:1

670:1

671:1

664:1

665:1

666:1

667:1

660:1

661:1

662:1

663:1

656:1

657:1

658:1

659:1

747:1

746:1

745:2

751:1

750:1

749:1

748:1

737:1

736:1

743:2

742:2

741:2

740:2

762:1

763:1

760:1

761:1

766:1

767:1

764:1

765:1

754:1

755:2

752:1

753:1

758:1

759:1

757:1

713:1

712:1

715:1

714:1

717:1

716:1

719:2

718:2

705:1

704:1

709:1

708:1

711:1

710:1

728:2

730:2

731:1

732:1

733:1

734:1

735:1

720:1

721:1

722:1

723:2

724:2

725:1

726:1

727:1

821:2

820:1

823:1

822:1

817:1

816:1

819:1

818:1

829:1

828:1

831:1

830:1

825:1

824:1

827:2

826:1

805:1

806:1

807:1

800:2

801:1

802:1

803:1

812:1

813:1

814:1

815:1

Disconnected from the target VM, address: '127.0.0.1:4081', transport: 'socket'

808:2

809:1

810:1

811:1

791:1

790:1

789:2

788:1

787:1

786:2

785:1

784:1

799:2

798:1

797:1

796:1

795:1

794:1

793:2

792:2

774:1

775:1

772:1

773:1

770:1

771:1

768:2

769:1

782:1

783:1

780:1

781:1

778:1

779:1

851:1

850:1

849:1

848:1

855:1

854:1

853:1

852:1

834:1

835:1

832:2

833:2

838:2

839:2

836:1

837:1

842:1

843:1

840:1

841:1

846:1

847:1

844:1

845:1

middle HAK Centroids-1:26

number of clusters with size of 2:22

number of clusters with size of 3:4

middle HAK Centroids-2:2

number of clusters with size of 17:1

number of clusters with size of 9:1

centroids count:2

size of cluster 0:34

size of cluster 1:31

Incorrectly clustered instances : 247.0 28.8551 %

Average of within cluster variance():48.93172677973324

Davies–Bouldin(smaller):3.4509602655913767

Dunn(greater):0.4084434913231492

silhouette(greater):0.183309624472303

-------current comparison--------

Cohision(Smaller):HAK(48.93172677973324) - KMeans(68.69045404490518) -

daviesBouldin(smaller):HAK(3.4509602655913767) - KMeans(3.8786606740801903) -

dunn(greater):KMeans(0.5076528917094687) - HAK(0.4084434913231492) -

silhouette(greater):HAK(0.183309624472303) - KMeans(0.03216109374522844) -

-------Matrix--------

----Cohision(Smaller)--- HAK KMeans

iris :0.4499822810158614 , 0.6494239504511713 ,

diabetes :88.15449730184497 , 96.8601208573361 ,

glass :0.8690685420106934 , 1.1384264489299334 ,

ionosphere :1.6330318482056727 , 2.3265048113368674 ,

segment-challenge :59.7876068325631 , 70.97030607216573 ,

segment-test :75.75298685029567 , 72.0569362790567 ,

unbalanced :48.93172677973324 , 68.69045404490518 ,

----daviesBouldin(smaller)--- HAK KMeans

iris :0.5011557852562679 , 0.6684333560944665 ,

diabetes :3.980056314418062 , 5.9621229890106155 ,

glass :1.5865591217519543 , 1.6553116281407252 ,

ionosphere :0.9174586530136186 , 1.5134759507430136 ,

segment-challenge :1.492324263894213 , 1.638285946592566 ,

segment-test :1.6607918735241198 , 1.7777039917247086 ,

unbalanced :3.4509602655913767 , 3.8786606740801903 ,

----dunn(greater)--- HAK KMeans

iris :2.5505601761609507 , 2.4320717428845486 ,

diabetes :0.4945644752977074 , 0.32035921624097713 ,

glass :0.2266925601574207 , 0.198117694328386 ,

ionosphere :1.5845760446437338 , 1.0177164042660343 ,

segment-challenge :0.781048075970198 , 0.47934351873811615 ,

segment-test :0.8596630433689769 , 0.3859709402950007 ,

unbalanced :0.4084434913231492 , 0.5076528917094687 ,

----silhouette(greater)--- HAK KMeans

iris :0.639770515603812 , 0.5493531211013164 ,

diabetes :0.057068083818500315 , 0.05645222485047844 ,

glass :0.20820991586304827 , 0.16272964412517069 ,

ionosphere :0.5174633524981576 , 0.2959701497777326 ,

segment-challenge :0.09853979868730835 , 0.2088774256479497 ,

segment-test :0.2732213193856892 , 0.21428451819485916 ,

unbalanced :0.183309624472303 , 0.03216109374522844 ,

-------Count result--------

Cohision(Smaller):HAK(6) - KMeans(1) -

daviesBouldin(smaller):HAK(7) - KMeans(0) -

dunn(greater):HAK(6) - KMeans(1) -

silhouette(greater):HAK(6) - KMeans(1) -

-------average of all results--------

Cohision(Smaller):HAK(39.36841434795274) - KMeans(44.67031035202596) -

daviesBouldin(smaller):HAK(1.9413294682070876) - KMeans(2.4419992194837548) -

dunn(greater):HAK(0.9865068381317339) - KMeans(0.763033201208933) -

silhouette(greater):HAK(0.2825118014755455) - KMeans(0.21711831106324792) -

1. Knowledge Discovery of Data [↑](#footnote-ref-1)