
Journal of Intelligent Information Systems, 17:2/3, 107–145, 2001

On Clustering Validation Techniques

MARIA HALKIDI mhalk@aueb.gr
YANNIS BATISTAKIS yannis@aueb.gr
MICHALIS VAZIRGIANNIS mvazirg@aueb.gr
Department of Informatics, Athens University of Economics & Business, Patision 76, 10434, Athens, Greece (Hellas)

Abstract. Cluster analysis aims at identifying groups of similar objects and, therefore helps to discover distribution of patterns and interesting correlations in large data sets. It has been subject of wide research since it arises in many application domains in engineering, business and social sciences. Especially, in the last years the availability of huge transactional and experimental data sets and the arising requirements for data mining created needs for clustering algorithms that scale and can be applied in diverse domains.

This paper introduces the fundamental concepts of clustering while it surveys the widely known clustering algorithms in a comparative way. Moreover, it addresses an important issue of clustering process regarding the quality assessment of the clustering results. This is also related to the inherent features of the data set under concern. A review of clustering validity measures and approaches available in the literature is presented. Furthermore, the paper illustrates the issues that are under-addressed by the recent algorithms and gives the trends in clustering process.

Keywords: clustering algorithms, unsupervised learning, cluster validity, validity indices

1. Introduction

Clustering is one of the most useful tasks in data mining process for discovering groups and identifying interesting distributions and patterns in the underlying data. Clustering problem is about partitioning a given data set into groups (clusters) such that the data points in a cluster are more similar to each other than points in different clusters (Guha et al., 1998). For example, consider a retail database records containing items purchased by customers. A clustering procedure could group the customers in such a way that customers with similar buying patterns are in the same cluster. Thus, the main concern in the clustering process is to reveal the organization of patterns into "sensible" groups, which allow us to discover similarities and differences, as well as to derive useful conclusions about them. This idea is applicable in many fields, such as life sciences, medical sciences and engineering. Clustering may be found under different names in different contexts, such as unsupervised learning (in pattern recognition), numerical taxonomy (in biology, ecology), typology (in social sciences) and partition (in graph theory) (Theodoridis and Koutroubas, 1999).

In the clustering process, there are no predefined classes and no examples that would show what kind of desirable relations should be valid among the data that is why it is perceived as an unsupervised process (Berry and Linoff, 1996). On the other hand, classification is a procedure of assigning a data item to a predefined set of categories (Fayyad et al., 1996). Clustering produces initial categories in which values of a data set are classified during the classification process.

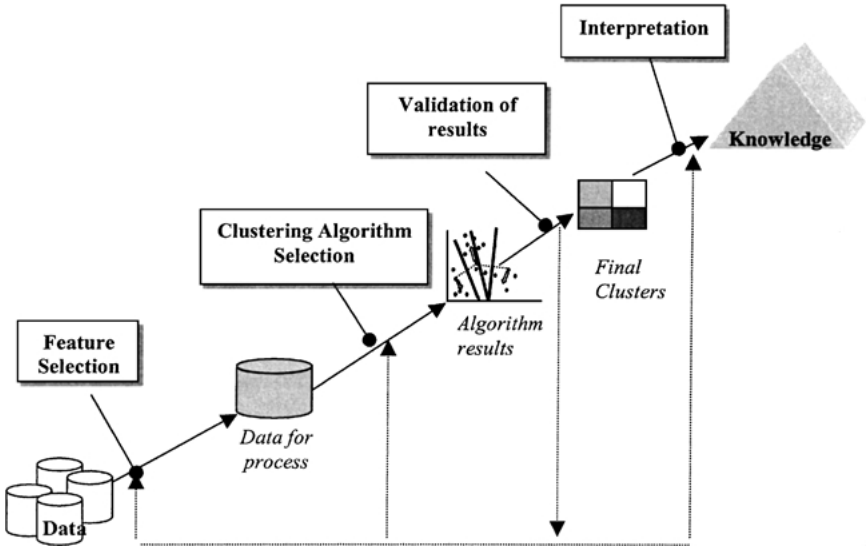

Figure 1. Steps of clustering process.

The clustering process may result in different partitioning of a data set, depending on the specific criterion used for clustering. Thus, there is a need of preprocessing before we assume a clustering task in a data set. The basic steps to develop clustering process are presented in figure 1 and can be summarized as follows (Fayyad et al., 1996):

- ***Feature selection.*** The goal is to select properly the features on which clustering is to be performed so as to encode as much information as possible concerning the task of our interest. Thus, preprocessing of data may be necessary prior to their utilization in clustering task.
- ***Clustering algorithm.*** This step refers to the choice of an algorithm that results in the definition of a good clustering scheme for a data set. A proximity measure and a clustering criterion mainly characterize a clustering algorithm as well as its efficiency to define a clustering scheme that fits the data set.

  i) *Proximity measure* is a measure that quantifies how "similar" two data points (i.e. feature vectors) are. In most of the cases we have to ensure that all selected features contribute equally to the computation of the proximity measure and there are no features that dominate others.

  ii) *Clustering criterion.* In this step, we have to define the clustering criterion, which can be expressed via a cost function or some other type of rules. We should stress that we have to take into account the type of clusters that are expected to occur in the data set. Thus, we may define a "good" clustering criterion, leading to a partitioning that fits well the data set.

- ***Validation of the results.*** The correctness of clustering algorithm results is verified using appropriate criteria and techniques. Since clustering algorithms define clusters that are not known a priori, irrespective of the clustering methods, the final partition of data requires some kind of evaluation in most applications (Rezaee et al., 1998).
- ***Interpretation of the results.*** In many cases, the experts in the application area have to integrate the clustering results with other experimental evidence and analysis in order to draw the right conclusion.

1.1. Clustering applications

Cluster analysis is a major tool in a number of applications in many fields of business and science. Hereby, we summarize the basic directions in which clustering is used (Theodoridis and Koutroubas, 1999):

- ***Data reduction***. Cluster analysis can contribute in compression of the information included in data. In several cases, the amount of available data is very large and its processing becomes very demanding. Clustering can be used to partition data set into a number of "interesting" clusters. Then, instead of processing the data set as an entity, we adopt the representatives of the defined clusters in our process. Thus, data compression is achieved.
- ***Hypothesis generation***. Cluster analysis is used here in order to infer some hypotheses concerning the data. For instance we may find in a retail database that there are two significant groups of customers based on their age and the time of purchases. Then, we may infer some hypotheses for the data, that it, "*young people go shopping in the evening*", "*old people go shopping in the morning*".
- ***Hypothesis testing***. In this case, the cluster analysis is used for the verification of the validity of a specific hypothesis. For example, we consider the following hypothesis: "*Young people go shopping in the evening*". One way to verify whether this is true is to apply cluster analysis to a representative set of stores. Suppose that each store is represented by its customer's details (age, job etc) and the time of transactions. If, after applying cluster analysis, a cluster that corresponds to "*young people buy in the evening*" is formed, then the hypothesis is supported by cluster analysis.
- ***Prediction based on groups***. Cluster analysis is applied to the data set and the resulting clusters are characterized by the features of the patterns that belong to these clusters. Then, unknown patterns can be classified into specified clusters based on their similarity to the clusters' features. Useful knowledge related to our data can be extracted. Assume, for example, that the cluster analysis is applied to a data set concerning patients infected by the same disease. The result is a number of clusters of patients, according to their reaction to specific drugs. Then for a new patient, we identify the cluster in which he/she can be classified and based on this decision his/her medication can be made.

More specifically, some typical applications of the clustering are in the following fields (Han and Kamber, 2001):

- ***Business***. In business, clustering may help marketers discover significant groups in their customers' database and characterize them based on purchasing patterns.
- ***Biology***. In biology, it can be used to define taxonomies, categorize genes with similar functionality and gain insights into structures inherent in populations.
- ***Spatial data analysis***. Due to the huge amounts of spatial data that may be obtained from satellite images, medical equipment, Geographical Information Systems (GIS), image database exploration etc., it is expensive and difficult for the users to examine spatial data in detail. Clustering may help to automate the process of analysing and understanding spatial data. It is used to identify and extract interesting characteristics and patterns that may exist in large spatial databases.
- ***Web mining***. In this case, clustering is used to discover significant groups of documents on the Web huge collection of semi-structured documents. This classification of Web documents assists in information discovery.

In general terms, clustering may serve as a pre-processing step for other algorithms, such as classification, which would then operate on the detected clusters.

1.2. Clustering algorithms categories

A multitude of clustering methods are proposed in the literature. Clustering algorithms can be classified according to:

- *The type of data input to the algorithm.*
- *The clustering criterion defining the similarity between data points.*
- *The theory and fundamental concepts on which clustering analysis techniques are based (e.g. fuzzy theory, statistics).*

Thus according to the method adopted to define clusters, the algorithms can be broadly classified into the following types (Jain et al., 1999):

- *Partitional clustering* attempts to directly decompose the data set into a set of disjoint clusters. More specifically, they attempt to determine an integer number of partitions that optimise a certain criterion function. The criterion function may emphasize the local or global structure of the data and its optimization is an iterative procedure.
- *Hierarchical clustering* proceeds successively by either merging smaller clusters into larger ones, or by splitting larger clusters. The result of the algorithm is a tree of clusters, called dendrogram, which shows how the clusters are related. By cutting the dendrogram at a desired level, a clustering of the data items into disjoint groups is obtained.
- *Density-based clustering.* The key idea of this type of clustering is to group neighbouring objects of a data set into clusters based on density conditions.
- *Grid-based clustering*. This type of algorithms is mainly proposed for spatial data mining. Their main characteristic is that they quantise the space into a finite number of cells and then they do all operations on the quantised space.

For each of above categories there is a wealth of subtypes and different algorithms for finding the clusters. Thus, according to the type of variables allowed in the data set can be categorized into (Guha et al., 1999; Huang et al., 1997; Rezaee et al., 1998):

- *Statistical*, which are based on statistical analysis concepts. They use similarity measures to partition objects and they are limited to numeric data.
- *Conceptual*, which are used to cluster categorical data. They cluster objects according to the concepts they carry.

  Another classification criterion is the way clustering handles uncertainty in terms of cluster overlapping.
- *Fuzzy clustering*, which uses fuzzy techniques to cluster data and they consider that an object can be classified to more than one clusters. This type of algorithms leads to clustering schemes that are compatible with everyday life experience as they handle the uncertainty of real data. The most important fuzzy clustering algorithm is *Fuzzy C-Means* (Bezdeck et al., 1984).
- *Crisp clustering*, considers non-overlapping partitions meaning that a data point either belongs to a class or not. Most of the clustering algorithms result in crisp clusters, and thus can be categorized in crisp clustering.
- *Kohonen net clustering*, which is based on the concepts of neural networks. The Kohonen network has input and output nodes. The input layer (input nodes) has a node for each attribute of the record, each one connected to every output node (output layer). Each connection is associated with a weight, which determines the position of the corresponding output node. Thus, according to an algorithm, which changes the weights properly, output nodes move to form clusters.

In general terms, the clustering algorithms are based on a criterion for assessing the quality of a given partitioning. More specifically, they take as input some parameters (e.g. number of clusters, density of clusters) and attempt to define the best partitioning of a data set for the given parameters. Thus, they define a partitioning of a data set based on certain assumptions and *not* necessarily the "best" one that fits the data set.

Since clustering algorithms discover clusters, which are not known a priori, the final partitions of a data set requires some sort of evaluation in most applications (Rezaee et al., 1998). For instance questions like "how many clusters are there in the data set?", "does the resulting clustering scheme fits our data set?", "is there a better partitioning for our data set?" call for clustering results validation and are the subjects of methods discussed in the literature. They aim at the quantitative evaluation of the results of the clustering algorithms and are known under the general term *cluster validity* methods.

The remainder of the paper is organized as follows. In the next section we present the main categories of clustering algorithms that are available in literature. Then, in Section 3 we discuss the main characteristics of these algorithms in a comparative way. In Section 4 we present the main concepts of clustering validity indices and the techniques proposed in literature for evaluating the clustering results. Moreover, an experimental study based on some of these validity indices is presented in Section 5, using synthetic and real data sets. We conclude in Section 6 by summarizing and providing the trends in clustering.

2. Clustering algorithms

In recent years, a number of clustering algorithms has been proposed and is available in the literature. Some representative algorithms of the above categories follow.

2.1. Partitional algorithms

In this category, *K-Means* is a commonly used algorithm (MacQueen, 1967). The aim of *K-Means* clustering is the optimisation of an objective function that is described by the equation

$$E = \sum_{i=1}^{c} \sum_{x \in C_i} d(x, m_i) \tag{1}$$

In the above equation, m_i is the center of cluster C_i, while $d(x, m_i)$ is the Euclidean distance between a point x and m_i. Thus, the criterion function Eattempts to minimize the distance of each point from the center of the cluster to which the point belongs. More specifically, the algorithm begins by initialising a set of c cluster centers. Then, it assigns each object of the dataset to the cluster whose center is the nearest, and re-computes the centers. The process continues until the centers of the clusters stop changing.

Another algorithm of this category is *PAM* (*Partitioning Around Medoids*). The objective of PAM is to determine a representative object (*medoid*) for each cluster, that is, to find the most centrally located objects within the clusters. The algorithm begins by selecting an object as medoid for each of c clusters. Then, each of the non-selected objects is grouped with the medoid to which it is the most similar. PAM swaps medoids with other non-selected objects until all objects qualify as medoid. It is clear that PAM is an expensive algorithm as regards finding the medoids, as it compares an object with entire dataset (Ng and Han, 1994).

CLARA (*Clustering Large Applications*), is an implementation of PAM in a subset of the dataset. It draws multiple samples of the dataset, applies PAM on samples, and then outputs the best clustering out of these samples (Ng and Han, 1994).

CLARANS (*Clustering Large Applications based on Randomized Search*), combines the sampling techniques with PAM. The clustering process can be presented as searching a graph where every node is a potential solution, that is, a set of k medoids. The clustering obtained after replacing a medoid is called the *neighbour* of the current clustering. CLARANS selects a node and compares it to a user-defined number of their neighbours searching for a local minimum. If a better neighbour is found (i.e., having lower-square error), CLARANS moves to the neighbour's node and the process start again; otherwise the current clustering is a local optimum. If the local optimum is found, CLARANS starts with a new randomly selected node in search for a new local optimum.

Finally K-prototypes, *K-mode* (Huang, 1997) are based on K-Means algorithm, but they aim at clustering categorical data.

2.2. Hierarchical algorithms

Hierarchical clustering algorithms according to the method that produce clusters can further be divided into (Theodoridis and Koutroubas, 1999):

- *Agglomerative algorithms*. They produce a sequence of clustering schemes of decreasing number of clusters at east step. The clustering scheme produced at each step results from the previous one by merging the two closest clusters into one.
- *Divisive algorithms*. These algorithms produce a sequence of clustering schemes of increasing number of clusters at each step. Contrary to the agglomerative algorithms the clustering produced at each step results from the previous one by splitting a cluster into two.

In sequel, we describe some representative *hierarchical clustering* algorithms.

BIRCH (Zhang et al., 1996) uses a hierarchical data structure called CF-tree for partitioning the incoming data points in an incremental and dynamic way. CF-tree is a height-balanced tree, which stores the clustering features and it is based on two parameters: *branching factor B* and *threshold T*, which referred to the diameter of a cluster (the diameter (or radius) of each cluster must be less than T). *BIRCH* can typically find a good clustering with a single scan of the data and improve the quality further with a few additional scans. It is also the first clustering algorithm to handle noise effectively (Zhang et al., 1996). However, it does not always correspond to a natural cluster, since each node in CF-tree can hold a limited number of entries due to its size. Moreover, it is order-sensitive as it may generate different clusters for different orders of the same input data.

CURE (Guha et al., 1998) represents each cluster by a certain number of points that are generated by selecting well-scattered points and then shrinking them toward the cluster centroid by a specified fraction. It uses a combination of random sampling and partition clustering to handle large databases.

ROCK (Guha et al., 1999), is a robust clustering algorithm for Boolean and categorical data. It introduces two new concepts, that is a point's neighbours and links, and it is based on them in order to measure the similarity/proximity between a pair of data points.

2.3. Density-based algorithms

Density based algorithms typically regard clusters as dense regions of objects in the data space that are separated by regions of low density.

A widely known algorithm of this category is *DBSCAN* (Ester et al., 1996). The key idea in *DBSCAN* is that for each point in a cluster, the neighbourhood of a given radius has to contain at least a minimum number of points. DBSCAN can handle noise (outliers) and discover clusters of arbitrary shape. Moreover, *DBSCAN* is used as the basis for an incremental clustering algorithm proposed in Ester et al. (1998). Due to its density-based nature, the insertion or deletion of an object affects the current clustering only in the neighbourhood of this object and thus efficient algorithms based on *DBSCAN* can be given for incremental insertions and deletions to an existing clustering (Ester et al., 1998).

In Hinneburg and Keim (1998) another density-based clustering algorithm, *DENCLUE*, is proposed. This algorithm introduces a new approach to cluster large multimedia databases.

The basic idea of this approach is to model the overall point density analytically as the sum of influence functions of the data points. The influence function can be seen as a function, which describes the impact of a data point within its neighbourhood. Then clusters can be identified by determining density attractors. Density attractors are local maximum of the overall density function. In addition, clusters of arbitrary shape can be easily described by a simple equation based on overall density function. The main advantages of DENCLUE are that it has good clustering properties in data sets with large amounts of noise and it allows a compact mathematically description of arbitrary shaped clusters in high-dimensional data sets. However, DENCLUE clustering is based on two parameters and as in most other approaches the quality of the resulting clustering depends on the choice of them. These parameters are (Hinneburg and Keim, 1998): i) parameter N which determines the influence of a data point in its neighbourhood and ii) $<$ describes whether a density-attractor is significant, allowing a reduction of the number of density-attractors and helping to improve the performance.

2.4. Grid-based algorithms

Recently a number of clustering algorithms have been presented for spatial data, known as grid-based algorithms. These algorithms quantise the space into a finite number of cells and then do all operations on the quantised space.

STING (Statistical Information Grid-based method) is representative of this category. It divides the spatial area into rectangular cells using a hierarchical structure. *STING* (Wang et al., 1997) goes through the data set and computes the statistical parameters (such as mean, variance, minimum, maximum and type of distribution) of each numerical feature of the objects within cells. Then it generates a hierarchical structure of the grid cells so as to represent the clustering information at different levels. Based on this structure *STING* enables the usage of clustering information to search for queries or the efficient assignment of a new object to the clusters.

WaveCluster (Sheikholeslami et al., 1998) is the latest grid-based algorithm proposed in literature. It is based on signal processing techniques (wavelet transformation) to convert the spatial data into frequency domain. More specifically, it first summarizes the data by imposing a multidimensional grid structure onto the data space (Han and Kamber, 2001). Each grid cell summarizes the information of a group of points that map into the cell. Then it uses a wavelet transformation to transform the original feature space. In wavelet transform, convolution with an appropriate function results in a transformed space where the natural clusters in the data become distinguishable. Thus, we can identify the clusters by finding the dense regions in the transformed domain. A-priori knowledge about the exact number of clusters is not required in *WaveCluster*.

2.5. Fuzzy clustering

The algorithms described above result in crisp clusters, meaning that a data point either belongs to a cluster or not. The clusters are non-overlapping and this kind of partitioning is further called *crisp clustering*. The issue of uncertainty support in clustering task leads to

the introduction of algorithms that use fuzzy logic concepts in their procedure. A common fuzzy clustering algorithm is the *Fuzzy C-Means* (*FCM*), an extension of classical *C-Means* algorithm for fuzzy applications (Bezdeck et al., 1984). *FCM* attempts to find the most characteristic point in each cluster, which can be considered as the "center" of the cluster and, then, the grade of membership for each object in the clusters.

Another approach proposed in literature to solve the problems of crisp clustering is based on probabilistic models. The basis of this type of clustering algorithms is the EM algorithm, which provides a quite general approach to learning in presence of unobservable variables (Mitchell, 1997). A common algorithm is the probabilistic variant of K-Means, which is based on the mixture of Gaussian distributions. This approach of K-Means uses probability density rather than distance to associate records with clusters (Berry and Linoff, 1996). More specifically, it regards the centers of clusters as means of Gaussian distributions. Then, it estimates the probability that a data point is generated by jth Gaussian (i.e., belongs to jth cluster). This approach is based on Gaussian model to extract clusters and assigns the data points to clusters assuming that they are generated by normal distribution. Also, this approach is implemented only in the case of algorithms, which are based on EM (Expectation Maximization) algorithm.

3. Comparison of clustering algorithms

Clustering is broadly recognized as a useful tool in many applications. Researchers of many disciplines have addressed the clustering problem. However, it is a difficult problem, which combines concepts of diverse scientific fields (such as databases, machine learning, pattern recognition, statistics). Thus, the differences in assumptions and context among different research communities caused a number of clustering methodologies and algorithms to be defined.

This section offers an overview of the main characteristics of the clustering algorithms presented in a comparative way. We consider the algorithms categorized in four groups based on their clustering method: *partitional, hierarchical, density-based* and *grid-based* algorithms. Tables 1–4 summarize the main concepts and the characteristics of the most representative algorithms of these clustering categories. More specifically our study is based on the following features of the algorithms: i) the type of the data that an algorithm supports (numerical, categorical), ii) the shape of clusters, iii) ability to handle noise and outliers, iv) the clustering criterion and, v) complexity. Moreover, we present the input parameters of the algorithms while we study the influence of these parameters to the clustering results. Finally we describe the type of algorithms results, i.e., the information that an algorithm gives so as to represent the discovered clusters in a data set.

As Table 1 depicts, *partitional algorithms* are applicable mainly to numerical data sets. However, there are some variants of K-Means such as K-mode, which handle categorical data. K-Mode is based on K-means method to discover clusters while it adopts new concepts in order to handle categorical data. Thus, the cluster centers are replaced with "modes", a new dissimilarity measure used to deal with categorical objects. Another characteristic of partitional algorithms is that they are unable to handle noise and outliers and they are not suitable to discover clusters with non-convex shapes. Moreover, they are based on certain

Table 1. The main characteristics of the partitional clustering algorithms.

Category	Partitional						
Name	Type of data	Complexity[a]	Geometry	Outliers, noise	Input parameters	Results	Clustering criterion
K-Mean	Numerical	$O(n)$	Non-convex shapes	No	Number of clusters	Center of clusters	$\min_{v1,v2,\ldots,vk}(E_k)$ $E_k = \sum_{i=1}^{k}\sum_{k=1}^{n} d^2(x_k, v_i)$
K-mode	Categorical	$O(n)$	Non-convex shapes	No	Number of clusters	Modes of clusters	$\min_{Q1,Q2,\ldots,Qk}(E_k)$ $E = \sum_{i=1}^{k}\sum_{l=1}^{n} d(X_l, Q_i)$ $D(X_i,\mathrm{Q}_l)$ = distance between categorical objects X_l, and modes Q_i
PAM	Numerical	$O(k(n-k)^2)$	Non-convex shapes	No	Number of clusters	Medoids of clusters	min (TC_{ih}) $TC_{ih} = \Sigma_j\ C_{jih}$
CLARA	Numerical	$O(k(40+k)^2 + k(n-\mathrm{k}))$	Non-convex shapes	No	Number of clusters	Medoids of clusters	min (TC_{ih}) $TC_{ih} = \Sigma_j\ C_{jih}$ (C_{jih} = the cost of replacing center i with h as far as Oj is concerned)
CLARANS	Numerical	$O(kn^2)$	Non-convex shapes	No	Number of clusters, maximum number of neighbors examined	Medoids of clusters	min (TC_{ih}) $TC_{ih} = \Sigma_j\ C_{jih}$
FCM	Numerical	$O(n)$	Non-convex shapes	No	Number of clusters	Center of cluster, beliefs	$\min_{U,v1,v2,\ldots,vk}(Jm(U,V))$
Fuzzy C-Means							$J_m(U, V) = \sum_{i=1}^{k}\sum_{j=1}^{n} U_{ik}^m d^2(x_j, v_i)$

[a] n is the number of points in the dataset and k the number of clusters defined.

Table 2. The main characteristics of the hierarchical clustering algorithms.

Category	Hierarchical						
Name	Type of data	Complexity[a]	Geometry	Outliers	Input parameters	Results	Clustering criterion
BIRCH	Numerical	$O(n)$	Non-convex shapes	Yes	Radius of clusters, branching factor	CF = (number of points in the cluster N, linear sum of the points in the cluster LS, the square sum of N data SS) points	A point is assigned to closest node (cluster) according to a chosen distance metric. Also, the clusters definition is based on the requirement that the number of points in each cluster must satisfy a threshold.
CURE	Numerical	$O(n^2 \log n)$	Arbitrary shapes	Yes	Number of clusters, number of clusters representatives	Assignment of data values to clusters	The clusters with the closest pair of representatives (well scattered points) are merged at each step.
ROCK	Categorical	$O(n^2 + nm_m m_a + n^2 \log n)$, $O(n^2, nm_m m_a)$ where $m_m$is the maximum number of neighbors for a point and $m_a$is the average number of neighbors for a point	Arbitrary shapes	Yes	Number of clusters	Assignment of data values to clusters	$\max(E_l)$ $E_l = \sum_{i=1}^{k} n_i \times \sum_{p_q, p_r \in V_i} \frac{link(p_q, p_r)}{n_i^{1+2f(\theta)}}$ – v_i *center of cluster I* – *link* (p_q, p_r) = *the* number of *common neighbors between* p_i *and* p_r.

[a] n is the number of points in the dataset under consideration.

Table 3. The main characteristics of the density-based clustering algorithms.

Category	Density-based						
Name	Type of data	Complexity[a]	Geometry	Outliers, noise	Input parameters	Results	Clustering criterion
DBSCAN	Numerical	$O(n\log n)$	Arbitrary shapes	Yes	Cluster radius, minimum number of objects	Assignment of data values to clusters	Merge points that are density reachable into one cluster.
DENCLUE	Numerical	$O(n\log n)$	Arbitrary shapes	Yes	Cluster radius σ, Minimum number of objects ξ	Assignment of data values to clusters	$f^D_{Gauss}(x^*) = \sum_{x_1 \in near(x^*)} e^{\frac{d(x^*,x_1)^2}{2\sigma^2}}$ *x^* density attractor for a point x if $F_{Gauss} > \xi$ then x attached to the cluster belonging to x^*.*

[a] n is the number of points in the dataset under consideration.

Table 4. The main characteristics of the grid-based clustering algorithms.

Category	Grid-based						
Name	Type of data	Complexity[a]	Geometry	Outliers	Input parameters	Output	Clustering criterion
Wave-Cluster	Special data	$O(n)$	Arbitrary shapes	Yes	Wavelets, the number of grid cells for each dimension, the number of application of wavelet transform	Clustered objects	Decompose feature space applying wavelet transformation *Average sub-band* $\rightarrow$clusters Detail sub-bands $\rightarrow$ clusters boundaries
STING	Special data	$O(K)$ K is the number of grid cells at the lowest level	Arbitrary shapes	Yes	Number of objects in a cell	Clustered objects	Divide the spatial area into rectangle cells and employ a hierarchical structure. Each cell at a high level is partitioned into a number of smaller cells in the next lower level.

[a] n is the number of points in the dataset under consideration.

assumption to partition a data set. Thus, they need to specify the number of clusters in advance except for CLARANS, which needs as input the maximum number of neighbours of a node as well as the number of local minima that will be found in order to define a partitioning of a dataset. The result of clustering process is the set of representative points of the discovered clusters. These points may be the centers or the medoids (most centrally located object within a cluster) of the clusters depending on the algorithm. As regards the *clustering criteria*, the objective of algorithms is to minimize the distance of the objects within a cluster from the representative point of this cluster. Thus, the criterion of *K*-Means aims at the minimization of the distance of objects belonging to a cluster from the cluster center, while PAM from its medoid. CLARA and CLARANS, as mentioned above, are based on the clustering criterion of PAM. However, they consider samples of the data set on which clustering is applied and as a consequence they may deal with larger data sets than PAM. More specifically, CLARA draws multiple samples of the data set and it applies PAM on each sample. Then it gives the best clustering as the output. The problem of this approach is that its efficiency depends on the sample size. Also, the clustering results are produced based only on samples of a data set. Thus, it is clear that if a sample is biased, a good clustering based on samples will not necessarily represent a good clustering of the whole data set. On the other hand, CLARANS is a mixture of PAM and CLARA. A key difference between CLARANS and PAM is that the former searches a subset of dataset in order to define clusters (Ng and Han, 1994). The subsets are drawn with some randomness in each step of the search, in contrast to CLARA that has a fixed sample at every stage. This has the benefit of not confining a search to a localized area. In general terms, CLARANS is more efficient and scalable than both CLARA and PAM. The algorithms described above are crisp clustering algorithms, that is, they consider that a data point (object) may belong to one and only one cluster. However, the boundaries of a cluster can hardly be defined in a crisp way if we consider real-life cases. FCM is a representative algorithm of fuzzy clustering which is based on K-means concepts in order to partition a data set into clusters. However, it introduces the concept of uncertainty and it assigns the objects to the clusters with an attached degree of belief. Thus, an object may belong to more than one cluster with different degree of belief.

A summarized view of the characteristics of *hierarchical clustering* methods is presented in Table 2. The algorithms of this category create a hierarchical decomposition of the database represented as dendrogram. They are more efficient in handling noise and outliers than partitional algorithms. However, they break down due to their non-linear time complexity (typically, complexity $O(n^2)$, where n is the number of points in the dataset) and huge I/O cost when the number of input data points is large. BIRCH tackles this problem using a hierarchical data structure called CF-tree for multiphase clustering. In BIRCH, a single scan of the dataset yields a good clustering and one or more additional scans can be used to improve the quality further. However, it handles only numerical data and it is order-sensitive (i.e., it may generate different clusters for different orders of the same input data). Also, BIRCH does not perform well when the clusters do not have uniform size and shape since it uses only the centroid of a cluster when redistributing the data points in the final phase. On the other hand, CURE employs a combination of random sampling and partitioning to handle large databases. It identifies clusters having non-spherical shapes and

wide variances in size by representing each cluster by multiple points. The representative points of a cluster are generated by selecting well-scattered points from the cluster and shrinking them toward the centre of the cluster by a specified fraction. However, CURE is sensitive to some parameters such as the number of representative points, the shrink factor used for handling outliers, number of partitions. Thus, the quality of clustering results depends on the selection of these parameters. ROCK is a representative hierarchical clustering algorithm for categorical data. It introduces a novel concept called "link" in order to measure the similarity/proximity between a pair of data points. Thus, the ROCK clustering method extends to non-metric similarity measures that are relevant to categorical data sets. It also exhibits good scalability properties in comparison with the traditional algorithms employing techniques of random sampling. Moreover, it seems to handle successfully data sets with significant differences in the sizes of clusters.

The third category of our study is the *density-based* clustering algorithms (Table 3). They suitably handle arbitrary shaped collections of points (e.g. ellipsoidal, spiral, cylindrical) as well as clusters of different sizes. Moreover, they can efficiently separate noise (outliers). Two widely known algorithms of this category, as mentioned above, are: DBSCAN and DENCLUE. DBSCAN requires the user to specify the radius of the neighbourhood of a point, *Eps*, and the minimum number of points in the neighbourhood, *MinPts*. Then, it is obvious that DBSCAN is very sensitive to the parameters *Eps* and *MinPts*, which are difficult to determine. Similarly, DENCLUE requires careful selection of its input parameters' value (i.e., σ and ξ), since such parameters may influence the quality of clustering results. However, the major advantage of DENCLUE in comparison with other clustering algorithms are (Han and Kamber, 2001): i) it has a solid mathematical foundation and generalized other clustering methods, such as partitional, hierarchical, ii) it has good clustering properties for data sets with large amount of noise, iii) it allows a compact mathematical description of arbitrary shaped clusters in high-dimensional data sets, iv) it uses grid cells and only keeps information about the cells that actually contain points. It manages these cells in a tree-based access structure and thus it is significant faster than some influential algorithms such as DBSCAN. In general terms the complexity of density based algorithms is *O(nlogn)*. They do not perform any sort of sampling, and thus they could incur substantial I/O costs. Finally, density-based algorithms may fail to use random sampling to reduce the input size, unless sample's size is large. This is because there may be substantial difference between the density in the sample's cluster and the clusters in the whole data set.

The last category of our study (see Table 4) refers to *grid-based algorithms*. The basic concept of these algorithms is that they define a grid for the data space and then do all the operations on the quantised space. In general terms these approaches are very efficient for large databases and are capable of finding arbitrary shape clusters and handling outliers. STING is one of the well-known grid-based algorithms. It divides the spatial area into rectangular cells while it stores the statistical parameters of the numerical features of the objects within cells. The grid structure facilitates parallel processing and incremental updating. Since STING goes through the database once to compute the statistical parameters of the cells, it is generally an efficient method for generating clusters. Its time complexity is $O(n)$. However, STING uses a multiresolution approach to perform cluster analysis and thus the quality of its clustering results depends on the granularity of the lowest level of grid.

Moreover, STING does not consider the spatial relationship between the children and their neighbouring cells to construct the parent cell. The result is that all cluster boundaries are either horizontal or vertical and thus the quality of clusters is questionable (Sheikholeslami et al., 1998). On the other hand, WaveCluster efficiently achieves to detect arbitrary shape clusters at different scales exploiting well-known signal processing techniques. It does not require the specification of input parameters (e.g. the number of clusters or a neighbourhood radius), though a-priori estimation of the expected number of clusters helps in selecting the correct resolution of clusters. In experimental studies, WaveCluster was found to outperform BIRCH, CLARANS and DBSCAN in terms of efficiency and clustering quality. Also, the study shows that it is not efficient in high dimensional space (Han and Kamber, 2001).

4. Cluster validity assessment

One of the most important issues in cluster analysis is the evaluation of clustering results to find the partitioning that best fits the underlying data. This is the main subject of cluster validity. In the sequel we discuss the fundamental concepts of this area while we present the various cluster validity approaches proposed in literature.

4.1. Problem specification

The objective of the clustering methods is to discover significant groups present in a data set. In general, they should search for clusters whose members are close to each other (in other words have a high degree of similarity) and well separated. A problem we face in clustering is to decide the optimal number of clusters that fits a data set.

In most algorithms' experimental evaluations 2D-data sets are used in order that the reader is able to visually verify the validity of the results (i.e., how well the clustering algorithm discovered the clusters of the data set). It is clear that visualization of the data set is a crucial verification of the clustering results. In the case of large multidimensional data sets (e.g. more than three dimensions) effective visualization of the data set would be difficult. Moreover the perception of clusters using available visualization tools is a difficult task for humans that are not accustomed to higher dimensional spaces.

The various clustering algorithms behave in a different way depending on:

i) the *features of the data set* (geometry and density distribution of clusters),
ii) the *input parameters values*

For instance, assume the data set in figure 2a. It is obvious that we can discover three clusters in the given data set. However, if we consider a clustering algorithm (e.g. K-Means) with certain parameter values (in the case of K-means the number of clusters) so as to partition the data set in four clusters, the result of clustering process would be the clustering scheme presented in figure 2b. In our example the clustering algorithm (K-Means) found the best four clusters in which our data set could be partitioned. However, this is not the optimal partitioning for the considered data set. We define, here, the term "optimal" clustering scheme as the outcome of running a clustering algorithm (i.e., a partitioning) that

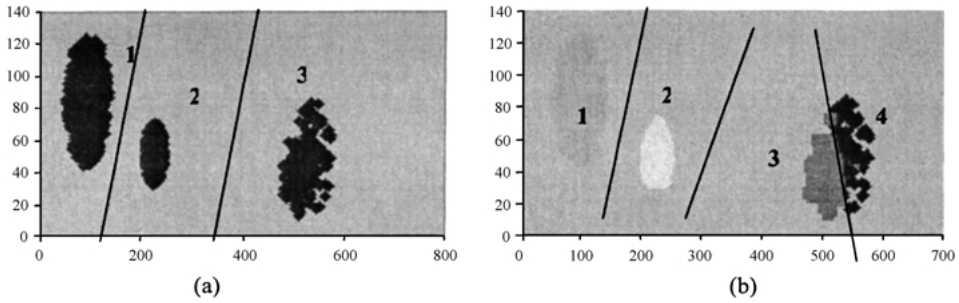

Figure 2. (a) A data set that consists of 3 clusters, (b) The results from the application of K-means when we ask four clusters.

best fits the inherent partitions of the data set. It is obvious from figure 2b that the depicted scheme is not the best for our data set i.e., the clustering scheme presented in figure 2b does not fit well the data set. The optimal clustering for our data set will be a scheme with three clusters.

As a consequence, if the clustering algorithm parameters are assigned an improper value, the clustering method may result in a partitioning scheme that is not optimal for the specific data set leading to wrong decisions. The problems of deciding the number of clusters better fitting a data set as well as the evaluation of the clustering results has been subject of several research efforts (Dave, 1996; Gath and Geva, 1989; Rezaee et al., 1998; Smyth, 1996; Theodoridis and Koutroubas, 1999; Xie and Beni, 1991).

In the sequel, we discuss the fundamental concepts of clustering validity and we present the most important criteria in the context of clustering validity assessment.

4.2. Fundamental concepts of cluster validity

The procedure of evaluating the results of a clustering algorithm is known under the term *cluster validity*. In general terms, there are three approaches to investigate cluster validity (Theodoridis and Koutroubas, 1999). The first is based on *external criteria*. This implies that we evaluate the results of a clustering algorithm based on a pre-specified structure, which is imposed on a data set and reflects our intuition about the clustering structure of the data set. The second approach is based on *internal criteria*. We may evaluate the results of a clustering algorithm in terms of quantities that involve the vectors of the data set themselves (e.g. proximity matrix). The third approach of clustering validity is based on *relative criteria*. Here the basic idea is the evaluation of a clustering structure by comparing it to other clustering schemes, resulting by the same algorithm but with different parameter values. There are two criteria proposed for clustering evaluation and selection of an optimal clustering scheme (Berry and Linoff, 1996):

1. *Compactness*, the members of each cluster should be as close to each other as possible. A common measure of compactness is the variance, which should be minimized.

2. *Separation,* the clusters themselves should be widely spaced. There are three common approaches measuring the distance between two different clusters:

   - *Single linkage*: It measures the distance between the closest members of the clusters.
   - *Complete linkage*: It measures the distance between the most distant members.
   - *Comparison of centroids*: It measures the distance between the centers of the clusters.

The two first approaches are based on statistical tests and their major drawback is their high computational cost. Moreover, the indices related to these approaches aim at measuring the degree to which a data set confirms an a-priori specified scheme. On the other hand, the third approach aims at finding the best clustering scheme that a clustering algorithm can be defined under certain assumptions and parameters.

A number of validity indices have been defined and proposed in literature for each of above approaches (Halkidi et al., 2000; Rezaee et al., 1998; Sharma, 1996; Theodoridis and Koutroubas, 1999; Xie and Beni, 1991).

4.3. Validity indices

In this section, we discuss methods suitable for quantitative evaluation of the clustering results, known as cluster validity methods. However, we have to mention that these methods give an indication of the quality of the resulting partitioning and thus they can only be considered as a tool at the disposal of the experts in order to evaluate the clustering results. In the sequel, we describe the fundamental criteria for each of the above described cluster validity approaches as well as their representative indices.

4.3.1. External criteria. In this approach the basic idea is to test whether the points of the data set are randomly structured or not. This analysis is based on the *Null Hypothesis,* H_0, expressed as a statement of random structure of a dataset, let X. To test this hypothesis we are based on statistical tests, which lead to a computationally complex procedure. In the sequel Monde Carlo techniques are used as a solution to high computational problems (Theodoridis and Koutroubas, 1999).

4.3.1.1. How Monde Carlo is used in cluster validity. The goal of using Monde Carlo techniques is the computation of the probability density function of the defined statistic indices. First, we generate a large amount of synthetic data sets. For each one of these synthetic data sets, called X_i, we compute the value of the defined index, denoted q_i. Then based on the respective values of q_i for each of the data sets X_i, we create a scatter-plot. This scatter-plot is an approximation of the probability density function of the index. In figure 3 we see the three possible cases of probability density function's shape of an index q. There are three different possible shapes depending on the critical interval $\overline{D_\rho}$, corresponding to *significant level* ρ (statistic constant). As we can see the probability density function of a statistic index q, under H_0, has a single maximum and the $\overline{D_\rho}$ region is either a half line, or a union of two half lines.

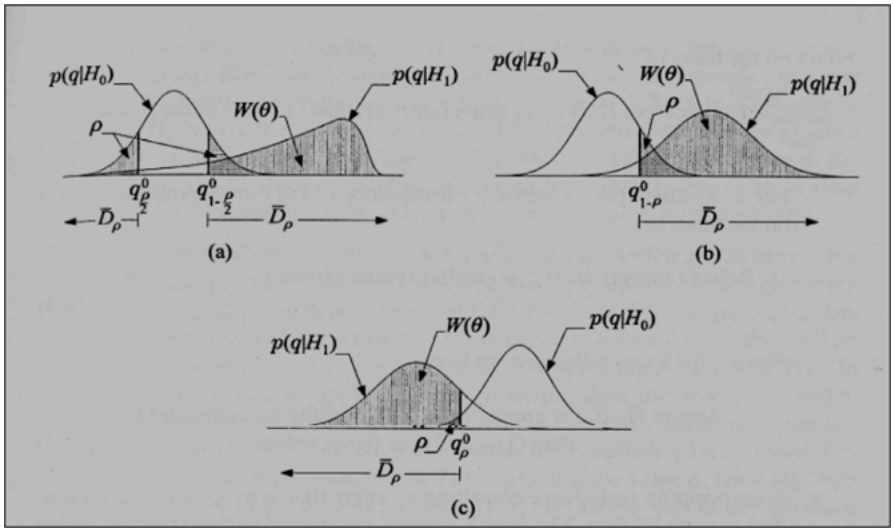

Figure 3. Confidence interval for (a) two-tailed index, (b) right-tailed index, (c) left-tailed index, where q_p^0 is the ρ proportion of q under hypothesis H_0. (Theodoridis and Koutroubas, 1999).

Assuming that this shape is right-tailed (figure 3b) and that we have generated the scatterplot using r values of the index q, called q_i, in order to accept or reject the *Null Hypothesis* H_0 we examine the following conditions (Theodoridis and Koutroubas, 1999):

We reject (accept) H_0 *If q's value for our data set, is greater (smaller) than (1->)·r of* q_i *values, of the respective synthetic data sets* X_i*.*
Assuming that the shape is left-tailed (figure 3c), *we reject (accept)* H_0 *if q's value for our data set, is smaller (greater) than* $> \cdot r$ *of* q_i *values.*
Assuming that the shape is two-tailed (figure 3a) *we accept* H_0 *if q is greater than (>/2)·r number of* q_i *values and smaller than (1- >/2)·r of* q_i *values.*

Based on the external criteria we can work in two different ways. Firstly, we can evaluate the resulting clustering structure **C**, by comparing it to an independent partition of the data $\boldsymbol{P}$ built according to our intuition about the clustering structure of the data set. Secondly, we can compare the proximity matrix P to the partition $\boldsymbol{P}$.

4.3.1.2. Comparison of C with partition P (not for hierarchy of clustering). Consider $\boldsymbol{C} = \{C_1 \cdots C_m\}$ is a clustering structure of a data set X and $\boldsymbol{P} = \{P_1 \cdots P_s\}$ is a defined partition of the data. We refer to a pair of points $(\mathbf{x_v}, \mathbf{x_u})$ from the data set using the following terms:

- **SS**: if both points belong to the same cluster of the clustering structure **C** and to the same group of partition ***P***.
- **SD**: if points belong to the same cluster of **C** and to different groups of ***P***.
- **DS**: if points belong to different clusters of **C** and to the same group of ***P***.
- **DD**: if both points belong to different clusters of **C** and to different groups of ***P***.

Assuming now that **a**, **b**, **c** and **d** are the number of SS, SD, DS and DD pairs respectively, then $a+b+c+d=\mathbf{M}$ which is the maximum number of all pairs in the data set (meaning, $M = N(N-1)/2$ where N is the total number of points in the data set).

Now we can define the following indices to measure the degree of similarity between ***C*** and ***P***:

- *Rand Statistic*: $R = (a+d)/M$,
- *Jaccard Coefficient*: $J = a/(a+b+c)$,

The above two indices take values between 0 and 1, and are maximized when $m = s$. Another index is the:

- *Folkes and Mallows index*:

$$\mathrm{FM} = a/\sqrt{m_1 m_2} = \sqrt{\frac{a}{a+b}\cdot\frac{a}{a+c}} \tag{2}$$

where $m_1 = (a+b)$, $m_2 = (a+c)$.

For the previous three indices it has been proven that high values of indices indicate great similarity between ***C*** and ***P***. The higher the values of these indices are the more similar ***C*** and ***P*** are. Other indices are:

- *Huberts* Γ *statistic*:

$$\Gamma = (1/M)\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} X(i,j)Y(i,j) \tag{3}$$

High values of this index indicate a strong similarity between X and Y.

- *Normalized* Γ *statistic*:

$$\bar{\Gamma} = \left[(1/M)\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}(X(i,j)-\mu_x)(Y(i,j)-\mu_Y)\right] \Big/ \sigma_X\sigma_Y \tag{4}$$

where $X(i,j)$ and $Y(i,j)$ are the (i,j) element of the matrices X,Y respectively that we have to compare. Also μ_x, μ_y, σ_x, σ_y are the respective means and variances of X,Y matrices. This index takes values between -1 and 1.

All these statistics have right-tailed probability density functions, under the random hypothesis. In order to use these indices in statistical tests we must know their respective

probability density function under the Null Hypothesis H_0, which is the hypothesis of random structure of our data set. This means that using statistical tests, if we accept the Null Hypothesis then our data are randomly distributed. However, the computation of the probability density function of these indices is difficult. A solution to this problem is to use Monde Carlo techniques. The procedure is as follows:

1. For $i = 1$ to r
   - ***Generate*** a data set X_i with N vectors (points) in the area of X, which means that the generated vectors have the same dimension with those of the data set X.
   - ***Assign*** each vector $y_{j,i}$ of X_i to the group that $x_j \in X$ belongs, according to the partition $\boldsymbol{P}$.
   - ***Run*** the same clustering algorithm used to produce structure C, for each X_i, and let C_i the resulting clustering structure.
   - ***Compute*** $q(C_i)$ value of the defined index q for P and C_i.

   End For
2. ***Create*** scatter-plot of the r validity index values, $q(C_i)$ (that computed into the for loop).

After having plotted the approximation of the probability density function of the defined statistic index, we compare its value, let q, to the $q(C_i)$ values, let q_i. The indices R, J, FM, Γ defined previously are used as the q index mentioned in the above procedure.

Example: Assume a given data set, X, containing 100 three-dimensional vectors (points). The points of X form four clusters of 25 points each. Each cluster is generated by a normal distribution. The covariance matrices of these distributions are all equal to $0.2I$, where I is the 3×3 identity matrix. The mean vectors for the four distributions are $[0.2, 0.2, 0.2]^{\mathrm{T}}$, $[0.5, 0.2, 0.8]^{\mathrm{T}}$, $[0.5, 0.8, 0.2]^{\mathrm{T}}$, and $[0.8, 0.8, 0.8]^{\mathrm{T}}$. We independently group data set X in four groups according to the partition $\boldsymbol{P}$ for which the first 25 vectors (points) belong to the first group P_1, the next 25 belong to the second group P_2, the next 25 belong to the third group P_3 and the last 25 vectors belong to the fourth group P_4. We run k-means clustering algorithm for $k = 4$ clusters and we assume that C is the resulting clustering structure. We compute the values of the indices for the clustering C and the partition $\boldsymbol{P}$, and we get $R = 0.91$, $J = 0.68$, $\mathrm{FM} = 0.81$ and $\Gamma = 0.75$. Then we follow the steps described above in order to define the probability density function of these four statistics. We generate 100 data sets X_i, $i = 1, \ldots, 100$, and each one of them consists of 100 random vectors (in 3 dimensions) using the uniform distribution. According to the partition $\boldsymbol{P}$ defined earlier for each X_i we assign the first 25 of its vectors to P_1 and the second, third and forth groups of 25 vectors to P_2, P_3 and P_4 respectively. Then we run k-means i-times, one time for each X_i, so as to define the respective clustering structures of datasets, denoted C_i. For each of them we compute the values of the indices R_i, J_i, FM_i, Γ_i, $i = 1, \ldots, 100$. We set the significance level $\rho = 0.05$ and we compare these values to the R, J, FM and Γ values corresponding to X. We accept or reject the null hypothesis whether $(1 - \rho) \cdot r = (1 - 0.05)100 = 95$ values of R_i, J_i, FM_i, Γ_i are greater or smaller than the corresponding values of R, J, FM, Γ. In our case the R_i, J_i, FM_i, Γ_i values are all smaller than the corresponding values of

R, J, FM, and Γ, which lead us to the conclusion that the null hypothesis H_0 is rejected. Something that we were expecting because of the predefined clustering structure of data set X.

4.3.1.3. Comparison of P (proximity matrix) with partition P. Partition $\boldsymbol{P}$ can be considered as a mapping

$$g : X \longrightarrow \{1 \cdots nc\}.$$

Assuming matrix Y: $Y(i, j) = \{1, \text{if } g(x_i) \neq g(x_j) \text{ and } 0, \text{otherwise}\}$, $i, j = 1 \cdots N$, we can compute Γ (or normalized Γ) statistic using the proximity matrix P and the matrix Y. Based on the index value, we may have an indication of the two matrices' similarity.

To proceed with the evaluation procedure we use the Monde Carlo techniques as mentioned above. In the "*Generate*" step of the procedure we generate the corresponding mappings g_i for every generated X_i data set. So in the *"Compute*" step we compute the matrix Y_i, for each X_i in order to find the Γ_i corresponding statistic index.

4.3.2. Internal criteria. Using this approach of cluster validity our goal is to evaluate the clustering result of an algorithm using only quantities and features inherent to the dataset. There are two cases in which we apply internal criteria of cluster validity depending on the clustering structure: a) hierarchy of clustering schemes, and b) single clustering scheme.

4.3.2.1. Validating hierarchy of clustering schemes. A matrix called cophenetic matrix, P_c, can represent the hierarchy diagram that produced by a hierarchical algorithm. The $P_c(i, j)$ element of cophenetic matrix represents the proximity level at which the two vectors x_i and x_j are found in the same cluster for the first time. We may define a statistical index to measure the degree of similarity between P_c and P (proximity matrix) matrices. This index is called *Cophenetic Correlation Coefficient* and defined as:

$$\text{CPCC} = \frac{(1/M)\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} d_{ij}\, c_{ij} - \mu_P \mu_c}{\sqrt{\left[(1/M)\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} d_{ij}^2 - \mu_P^2\right]\left[(1/M)\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} c_{ij}^2 - \mu_C^2\right]}},$$
$$-1 \leq \text{CPCC} \leq 1 \tag{5}$$

where $M = N \cdot (N-1)/2$ and N is the number of points in a dataset. Also, μ_p*and* μ_c are the means of matrices P and P_c respectively, and are given by Eq. (6):

$$\mu_P = (1/M)\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} P(i, j), \quad \mu_C = (1/M)\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} P_c(i, j) \tag{6}$$

Moreover, d_{ij}, c_{ij} are the (i, j) elements of P and P_c matrices respectively. A value of the index close to 0 is an indication of a significant similarity between the two matrices. The procedure of the Monde Carlo techniques described above is also used in this case of validation.

4.3.2.2. Validating a single clustering scheme. The goal here is to find the degree of agreement between a given clustering scheme C, consisting of nc clusters, and the proximity matrix P. The defined index for this approach is Hubert's Γ statistic (or normalized Γ statistic). An additional matrix for the computation of the index is used, that is $Y(i, j) =$ {1, if x_i and x_j belong to different clusters, and 0 , otherwise}, $i, j = 1, \ldots, N$.

The application of Monde Carlo techniques is also here the way to test the random hypothesis in a given data set.

4.3.3. Relative criteria. The basis of the above described validation methods is statistical testing. Thus, the major drawback of techniques based on internal or external criteria is their high computational demands. A different validation approach is discussed in this section. It is based on relative criteria and does not involve statistical tests. The fundamental idea of this approach is to choose the best clustering scheme of a set of defined schemes according to a pre-specified criterion. More specifically, the problem can be stated as follows:

"*Let P_{alg} the set of parameters associated with a specific clustering algorithm (e.g. the number of clusters nc). Among the clustering schemes C_i, $i = 1, \ldots, nc$, defined by a specific algorithm, for different values of the parameters in P_{alg}, choose the one that best fits the data set.*"

Then, we can consider the following cases of the problem:

I) $\boldsymbol{P_{\text{alg}}}$ **does not contain the number of clusters, *nc*, as a parameter**. In this case, the choice of the optimal parameter values are described as follows: We run the algorithm for a wide range of its parameters' values and we choose the largest range for which nc remains constant (usually nc $<<$ N (number of tuples)). Then we choose as appropriate values of the P_{alg} parameters the values that correspond to the middle of this range. Also, this procedure identifies the number of clusters that underlie our data set.

II) $\boldsymbol{P_{\text{alg}}}$ **contains *nc* as a parameter**. The procedure of identifying the best clustering scheme is based on a validity index. Selecting a suitable performance index, q, we proceed with the following steps:

- We run the clustering algorithm for all values of nc between a minimum n_{cmin} and a maximum n_{cmax}. The minimum and maximum values have been defined a-priori by user.
- For each of the values of nc, we run the algorithm r times, using different set of values for the other parameters of the algorithm (e.g. different initial conditions).
- We plot the best values of the index q obtained by each nc as the function of nc.

Based on this plot we may identify the best clustering scheme. We have to stress that there are two approaches for defining the best clustering depending on the behaviour of q with respect to nc. Thus, if the validity index does not exhibit an increasing or decreasing trend as nc increases we seek the maximum (minimum) of the plot. On the other hand, for indices that increase (or decrease) as the number of clusters increase we search for the values of nc at which a significant local change in value of the index occurs. This change appears as a "knee" in the plot and it is an indication of the number of clusters underlying

the dataset. Moreover, the absence of a knee may be an indication that the data set possesses no clustering structure.

In the sequel, some representative validity indices for crisp and fuzzy clustering are presented.

4.3.3.1. Crisp clustering. This section discusses validity indices suitable for crisp clustering.

The modified Hubert Γ *statistic*. The definition of the modified Hubert Γ statistic is given by the equation

$$\Gamma = (1/M)\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} P(i, j) \cdot Q(i, j) \tag{7}$$

where $M = N(N-1)/2$, P is the proximity matrix of the data set and Q is an $N \times N$ matrix whose (i, j) element is equal to the distance between the representative points (v_{ci}, v_{cj}) of the clusters where the objects x_i and x_j belong.

Similarly, we can define the normalized Hubert Γ statistic (given by Eq. (4)). If the $d(v_{ci}, v_{cj})$ is close to $d(x_i, x_j)$ for $i, j = 1, 2, \ldots, N$, P and Q will be in close agreement and the values of Γ and $\hat{\Gamma}$ (normalized Γ) will be high. Conversely, a high value of Γ ($\hat{\Gamma}$) indicates the existence of compact clusters. Thus, in the plot of normalized Γ versus nc, we seek a significant knee that corresponds to a significant increase of normalized Γ. The number of clusters at which the knee occurs is an indication of the number of clusters that underlie the data. We note, that for $nc = 1$ and $nc = N$ the index is not defined.

Dunn and Dunn-like indices. A cluster validity index for crisp clustering proposed in Dunn (1974), attempts to identify "compact and well separated clusters". The index is defined by Eq. (8) for a specific number of clusters

$$D_{nc} = \min_{i=1,\ldots,nc}\left\{\min_{j=i+1,\ldots,nc}\left(\frac{d(c_i, c_j)}{\max_{k=1,\ldots,nc} diam\,(c_k)}\right)\right\} \tag{8}$$

where $d(c_i, c_j)$ is the dissimilarity function between two clusters c_i and c_j defined as

$$d(c_i, c_j) = \min_{x \in c_i, y \in c_j} d(\mathbf{x}, \mathbf{y})\,, \tag{9}$$

and diam(c) is the diameter of a cluster, which may be considered as a measure of dispersion of the clusters. The diameter of a cluster C can be defined as follows:

$$diam(C) = \max_{x, y \in C} d(x, y) \tag{10}$$

It is clear that if the dataset contains compact and well-separated clusters, the distance between the clusters is expected to be large and the diameter of the clusters is expected to be small. Thus, based on the Dunn's index definition, we may conclude that large values of the index indicate the presence of compact and well-separated clusters.

The index D_{nc} does not exhibit any trend with respect to number of clusters. Thus, the maximum in the plot of D_{nc} versus the number of clusters can be an indication of the number of clusters that fits the data. The implications of the Dunn index are: i) the considerable amount of time required for its computation, ii) the sensitive to the presence of noise in datasets, since these are likely to increase the values of diam(c) (i.e., dominator of Eq. (8))

Three indices, are proposed in Pal and Biswas (1997) that are more robust to the presence of noise. They are widely known as Dunn-like indices since they are based on Dunn index. Moreover, the three indices use for their definition the concepts of the minimum spanning tree (MST), the relative neighbourhood graph (RNG) and the Gabriel graph respectively (Theodoridis and Koutroubas, 1999).

Consider the index based on MST. Let a cluster c_i and the complete graph G_i whose vertices correspond to the vectors of c_i. The weight, w_e, of an edge, e, of this graph equals the distance between its two end points, x, y. Let E_i^{MST} be the set of edges of the MST of the graph G_i and e_i^{MST} the edge in E_i^{MST} with the maximum weight. Then the diameter of C_i is defined as the weight of e_i^{MST}. Then the Dunn-like index based on the concept of the MST is given by equation

$$D_{nc} = \min_{i=1,\ldots,nc} \left\{ \min_{j=i+1,\ldots,nc} \left(\frac{d(c_i, c_j)}{\max_{k=1,\ldots,nc} diam_k^{\text{MST}}} \right) \right\} \tag{11}$$

The number of clusters at which $D_{\text{m}}^{\text{MST}}$ takes its maximum value indicates the number of clusters in the underlying data. Based on similar arguments we may define the Dunn-like indices for GG nad RGN graphs.

The Davies-Bouldin (DB) index. A similarity measure R_{ij} between the clusters C_i and $C_j$is defined based on a measure of dispersion of a cluster C_i and a dissimilarity measure between two clusters d_{ij}. The R_{ij} index is defined to satisfy the following conditions (Davies and Bouldin, 1979):

1. $R_{ij} \geq 0$
2. $R_{ij} = R_{ji}$
3. if $s_i = 0$ and $s_j = 0$ then $R_{ij} = 0$
4. if $s_j > s_k$ and $d_{ij} = d_{ik}$ then $R_{ij} > R_{ik}$
5. if $s_j = s_k$ and $d_{ij} < d_{ik}$ then $R_{ij} < R_{ik}$.

These conditions state that R_{ij} is nonnegative and symmetric.

A simple choice for R_{ij} that satisfies the above conditions is Davies and Bouldin (1979):

$$R_{ij} = (s_i + s_j)/d_{ij}. \tag{12}$$

Then the DB index is defined as

$$DB_{nc} = \frac{1}{nc}\sum_{i=1}^{nc} R_i$$
$$R_i = \max_{i=1,\ldots,nc, i\neq j} R_{ij}, \quad \text{i} = 1, \ldots, \text{nc} \tag{13}$$

It is clear for the above definition that DB_{nc} is the average similarity between each cluster $c_i, i = 1, \ldots,$ nc and its most similar one. It is desirable for the clusters to have the minimum possible similarity to each other; therefore we seek clusterings that minimize DB. The DB_{nc} index exhibits no trends with respect to the number of clusters and thus we seek the minimum value of DB_{nc} in its plot versus the number of clusters.

Some alternative definitions of the dissimilarity between two clusters as well as the dispersion of a cluster, c_i, is defined in Davies and Bouldin (1979).

Moreover, in Pal and Biswas (1997) three variants of the DB_{nc} index are proposed. They are based on MST, RNG and GG concepts similarly to the cases of the Dunn-like indices.

Other validity indices for crisp clustering have been proposed in Dave (1996) and Milligan et al. (1983). The implementation of most of these indices is very computationally expensive, especially when the number of clusters and number of objects in the data set grows very large (Xie and Beni, 1991). In Milligan and Cooper (1985), an evaluation study of thirty validity indices proposed in literature is presented. It is based on small data sets (about 50 points each) with well-separated clusters. The results of this study (Milligan and Cooper, 1985) place Caliski and Harabasz (1974), Je(2)/Je(1) (1984), C-index (1976), Gamma and Beale among the six best indices. However, it is noted that although the results concerning these methods are encouraging they are likely to be data dependent. Thus, the behaviour of indices may change if different data structures were used (Milligan and Cooper, 1985). Also, some indices based on a sample of clustering results. A representative example is Je(2)/Je(1) which is computed based only on the information provided by the items involved in the last cluster merge.

RMSSDT, SPR, RS, CD. In this point we will give the definitions of four validity indices, which have to be used simultaneously to determine the number of clusters existing in the data set. These four indices can be applied to each step of a *hierarchical* clustering algorithm and they are known as (Sharma, 1996):

- *Root-mean-square standard deviation (RMSSTD) of the new cluster*
- *Semi-partial R-squared (SPR)*
- *R-squared (RS)*
- *Distance between two clusters.*

Getting into a more detailed description of them we can say that:

RMSSTD of a new clustering scheme defined in a level of clustering hierarchy is the square root of the pooled sample variance of all the variables (attributes used in the clustering process). This index measures the homogeneity of the formed clusters at each step of the hierarchical algorithm. Since the objective of cluster analysis is to form homogeneous

groups the RMSSTD of a cluster should be as small as possible. In case that the values of RMSSTD are higher at this step than the ones of the previous step, we have an indication that the new clustering scheme is not homogenous.

In the following definitions we shall use the symbolism SS, which means *Sum of Squares* and refers to the equation: $SS = \sum_{i=1}^{n} (X_i - \bar{X})^2$. Along with this we shall use some additional symbolism like:

i) SS_w referring to the within group sum of squares,
ii) SS_b referring to the between groups sum of squares.
iii) SS_t referring to the total sum of squares, of the whole data set.

SPR of the new cluster is the difference between the pooled SS_w of the new cluster and the sum of the pooled SS_w's values of clusters joined to obtain the new cluster (*loss of homogeneity*), divided by the pooled SS_t for the whole data set. This index measures the loss of homogeneity after merging the two clusters of a single algorithm step. If the index value is zero then the new cluster is obtained by merging two perfectly homogeneous clusters. If its value is high then the new cluster is obtained by merging two heterogeneous clusters.

RS of the new cluster is the ratio of SS_b to SS_t. As we can understand SS_b is a measure of difference between groups. Since $SS_t = SS_b + SS_w$ the greater the SS_b the smaller the SS_w and vise versa. As a result, the greater the differences between groups are the more homogenous each group is and vise versa. Thus, RS may be considered as a measure of the degree of difference between clusters. Furthermore, it measures the degree of homogeneity between groups. The values of RS range between 0 and 1. In case that the value of RS is zero (0) indicates that no difference exists among groups. On the other hand, when RS equals 1 there is an indication of significant difference among groups.

The *CD* index measures the distance between the two clusters that are merged in a given step. This distance is measured each time depending on the selected representatives for the hierarchical clustering we perform. For instance, in case of *Centroid hierarchical clustering* the representatives of the formed clusters are the centers of each cluster, so CD is the distance between the centers of the clusters. In case that we use *single linkage* CD measures the minimum Euclidean distance between all possible pairs of points. In case of *complete linkage* CD is the maximum Euclidean distance between all pairs of data points, and so on.

Using these four indices we determine the number of clusters that exist into our data set, plotting a graph of all these indices values for a number of different stages of the clustering algorithm. In this graph we search for the steepest knee, or in other words, the greater jump of these indices' values from higher to smaller number of clusters.

Example: Assume the data set presented in Table 5. After running hierarchical clustering with *Centroid* method we evaluate our clustering structure using the above-defined indices. The *Agglomerative Schedule* presented in Table 6 gives us the way that the algorithm worked. Thus the indices computed as follows:
'At stage (step) 4 for instance (see Table 6), the clusters 3 and 5 merged (meaning tuples {S3, S4} and {S5, S6}). Merging these subjects the resulting cluster is called 3 (S3, S4).

Table 5. Data set used in the example.

Subject Id	Income ($ thous.)	Education (years)
S1	5	5
S2	6	6
S3	15	14
S4	16	15
S5	25	20
S6	30	19

Table 6. Hierarchical algorithm results (centroid method).

	Agglomeration schedule					
	Cluster combined			Stage cluster first appears		
Stage	Cluster 1	Cluster 2	Coefficients	Cluster 1	Cluster 2	Next Stage
1	3	4	2.000	0	0	4
2	1	2	2.000	0	0	5
3	5	6	26.000	0	0	4
4	3	5	169.000	1	3	5
5	1	3	388.250	2	4	0

The new cluster is denoted using the smaller label number of the clusters' labels that are merged. At stage (step) 4:

RMSSTD: Sample variances and sample means analytic equation are:

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2, \quad \bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i,$$

for variable *income* $S^2 = 157/3 = 52.333$, and
for variable *education* $S^2 = 26/3 = 8.667$. Then

$$RMSSTD = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \bar{X})^2_{income} + \sum_{e=1}^{m}(X_e - \bar{X})^2_{education}}{(n-1)_{income} + (m-1)_{education}}}$$

where n, m (here $m = n$) are the respective number of values that variables *income, education* have. From the previous equation we compute that RMSSTD = 5.523.

RS: Lets now compute RS index of the two merging clusters 3 ({S3, S4}) and 5 ({S5, S6}). $SS_{w(income)3} = 157$ for variable *income* of cluster 3, and $SS_{w(income)5} = 0.5$ of cluster 5, giving the total $SS_{w(income)} = 157.5$ for variable *income*. Similarly for variable *education* we have $SS_{w(education)3} = 26$, $SS_{w(income)5} = 0.5$ giving $SS_{w(income)} = 26.5$. So the pooled sum

Table 7. Indices values.

Stage (step)	RMSSTD	SPR	RS	CD
1	0.707107	0.001425	0.998575	1.4142
2	0.707107	0.001425	0.997150	1.4142
3	2.549510	0.018527	0.978622	5.0990
4	5.522681	0.240855	0.737767	13
5	8.376555	0.737767	0.000000	19.7041

of squares within clusters among all the variables is $SS_w = 157{,}5 + 26.5 = 184$. SS_t pooled from all the variables of the data set is 701.166, then $SS_b = SS_t - SS_w = 701.166 - 184 = 517.166$. Using these we can compute $RS = 517.166/701.166 = 0.738$.

SPR: at this stage 4, the Loh*(loss of homogeneity)* $= SS_{w(\text{of new cluster 3})} - [SS_{w(cl3)} + SS_{w(cl5)}]$, so $SPR = \text{Loh}/SS_t = [0*(183) - (1+13)]/701.166 = 0.241$.

CD: This index is shown at Table 6, in *coefficients* column.

The same procedure is followed to find the values of each index for the rest of the algorithm's stages. Table 7 summarizes all these values. Based on these values we plot the graphs shown in figure 4. In these graphs we search for a point at which a significant change in values of each of the consider indices occur.

In the case of nonhierarchical clustering (e.g. K-Means) we may also use some of these indices in order to evaluate the resulting clustering. The indices that are more meaningful to use in this case are RMSSTD and RS. The idea, here, is to run the algorithm a number of times for different number of clusters each time. Then we plot the respective graphs of the validity indices for these clusterings and as the previous example shows, we search for the significant "knee" in these graphs. The number of clusters at which the "knee" is observed indicates the optimal clustering for our data set. In this case the validity indices described

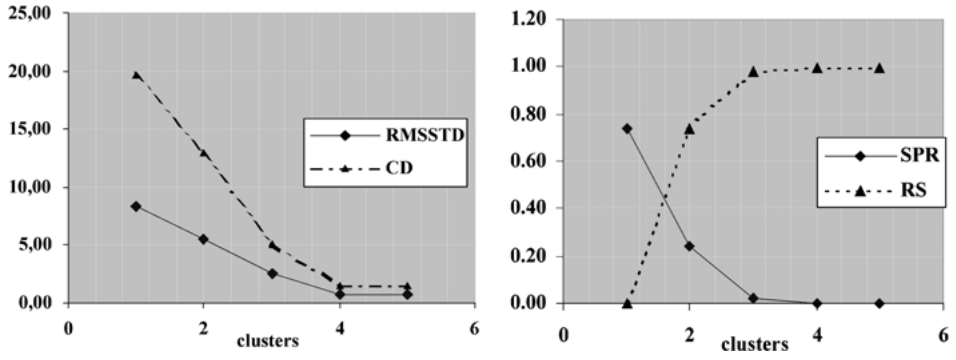

Figure 4. Validity graphs.

before take the following form:

$$RMSSTD = \left[\frac{\sum_{\substack{i=1\ldots nc \\ j=1\ldots d}} \sum_{k=1}^{n_{ij}} (x_k - \bar{x}_j)^2}{\sum_{\substack{i=1\ldots nc \\ j=1\ldots d}} (n_{ij} - 1)} \right]^{\frac{1}{2}} \tag{14}$$

$$RS = \frac{\left\{ \sum_{j=1\cdots d} \left[\sum_{k=1}^{n_j} (x_k - \bar{x}_j)^2 \right] \right\} - \left\{ \sum_{\substack{i=1\cdots nc \\ j=1\cdots d}} \left[\sum_{k=1}^{n_{ij}} (x_k - \bar{x}_j)^2 \right] \right\}}{\sum_{j=1\cdots d} \left[\sum_{k=1}^{n_j} (x_k - \bar{x}_j)^2 \right]} \tag{15}$$

where nc is the number of clusters, d the number of variables(data dimension), n_j is the number of data values of j dimension while n_{ij} corresponds to the number of data values of j dimension that belong to cluster i. Also $\overline{x_j}$ is the mean of data values of j dimension.

The SD validity index. A most recent clustering validity approach is proposed in Halkidi, et al. (2000). The SD validity index is defined based on the concepts of the a*verage scattering for clusters* and *total separation between clusters*. In the sequel, we give the fundamental definition for this index.

Average scattering for clusters. The average scattering for clusters is defined as

$$Scat(nc) = \frac{1}{nc} \sum_{i=1}^{nc} \|\sigma(v_i)\| / \|\sigma(X)\| \tag{16}$$

Total separation between clusters. The definition of total scattering (separation) between clusters is given by Eq. (17)

$$Dis(nc) = \frac{D_{\max}}{D_{\min}} \sum_{k=1}^{nc} \left(\sum_{z=1}^{nc} \|v_k - v_z\| \right)^{-1} \tag{17}$$

where $D_{\max} = \max(\|v_i - v_j\|)\ \forall i, j \in \{1, 2, 3, \ldots, \text{nc}\}$ is the maximum distance between cluster centers. The $D_{\min} = \min(\|v_i - v_j\|)\ \forall i, j \in \{1, 2, \ldots, \text{nc}\}$ is the minimum distance between cluster centers.

Now, we can define a validity index based on Eqs. (16) and (17), as follows

$$SD(nc) = a \cdot Scat(nc) + Dis(nc) \tag{18}$$

where K is a weighting factor equal to $Dis(c_{\max})$ where $c_{\max}$ is the maximum number of input clusters.

The first term (i.e., $Scat(nc)$ is defined by Eq. (16) indicates the average compactness of clusters (i.e., intra-cluster distance). A small value for this term indicates compact clusters and as the scattering within clusters increases (i.e., they become less compact) the value of $Scat(nc)$ also increases. The second term $Dis(nc)$ indicates the total separation between the nc clusters (i.e., an indication of inter-cluster distance). Contrary to the first term the second one, $Dis(nc)$, is influenced by the geometry of the clusters centres and increase with

the number of clusters. It is obvious for previous discussion that the two terms of *SD* are of the different range, thus a weighting factor is needed in order to incorporate both terms in a balanced way. The number of clusters, nc, that minimizes the above index can be considered as an optimal value for the number of clusters present in the data set. Also, the influence of the maximum number of clusters $c_{\max}$, related to the weighting factor, in the selection of the optimal clustering scheme is discussed in Halkidi et al. (2000). It is proved that *SD* proposes an optimal number of clusters almost irrespectively of $c_{\max}$. However, the index cannot handle properly arbitrary shaped clusters. The same applies to all the aforementioned indices.

4.3.3.2. Fuzzy clustering. In this section, we present validity indices suitable for fuzzy clustering. The objective is to seek clustering schemes where most of the vectors of the dataset exhibit high degree of membership in one cluster. We note, here, that a fuzzy clustering is defined by a matrix $U = [u_{ij}]$, where u_{ij} denotes the degree of membership of the vector x_i in the j cluster. Also, a set of the cluster representatives has been defined. Similarly to the crisp clustering case we define validity index, q, and we search for the minimum or maximum in the plot of q versus m. Also, in case that q exhibits a trend with respect to the number of clusters, we seek a significant knee of decrease (or increase) in the plot of q.

In the sequel two categories of fuzzy validity indices are discussed. The first category uses only the memberships values, u_{ij}, of a fuzzy partition of data. On the other hand the latter one involves both the U matrix and the dataset itself.

Validity Indices involving only the membership values. Bezdek proposed in Bezdeck et al. (1984) the *partition coefficient*, which is defined as

$$PC = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{nc} u_{ij}^2 \tag{19}$$

The PC index values range in [1/nc, 1], where nc is the number of clusters. The closer to unity the index the "crisper" the clustering is. In case that all membership values to a fuzzy partition are equal, that is, u_{ij}=1/nc, the PC obtains its lower value. Thus, the closer the value of PC is to 1/nc, the fuzzier the clustering is. Furthermore, a value close to 1/nc indicates that there is no clustering tendency in the considered dataset or the clustering algorithm failed to reveal it.

The *partition entropy coefficient* is another index of this category. It is defined as

$$PE = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{nc} u_{ij}\cdot \log_a(u_{ij}) \tag{20}$$

where a is the base of the logarithm. The index is computed for values of *nc* greater than 1 and its values ranges in [0, $\log_a$nc]. The closer the value of PE to 0, the harder the clustering is. As in the previous case, the values of index close to the upper bound (i.e., $\log_a$nc), indicate absence of any clustering structure in the dataset or inability of the algorithm to extract it.

The drawbacks of these indices are:

i) their monotonous dependency on the number of clusters. Thus, we seek significant knees of increase (for PC) or decrease (for PE) in plot of the indices versus the number of clusters.,
ii) their sensitivity to the fuzzifier, m. More specifically, as $m \to 1$ the indices give the same values for all values of nc. On the other hand when $m \to \infty$, both PC and PE exhibit significant knee at $nc = 2$.
iii) the lack of direct connection to the geometry of the data (Dave, 1996), since they do not use the data itself.

Indices involving the membership values and the dataset. The *Xie-Beni index* (Xie and Beni, 1991), XB, also called the compactness and separation validity function, is a representative index of this category.

Consider a fuzzy partition of the data set $X = \{x_j;\ j = 1, \ldots, n\}$ with $v_i (\mathrm{i} = 1, \ldots, \mathrm{nc})$ the centers of each cluster and u_{ij} the membership of data point j belonging to cluster i.

The fuzzy deviation of x_j form cluster i, d_{ij}, is defined as the distance between x_j and the center of cluster weighted by the fuzzy membership of data point j belonging to cluster i.

$$d_{ij} = u_{ij} \| x_j - v_i \| \tag{21}$$

Also, for a cluster i, the summation of the squares of fuzzy deviation of the data point in X, denoted σ_i, is called variation of cluster i.

The summation of the variations of all clusters, σ, is called *total variation* of the data set.

The quantity $\pi = (\sigma_i / n_i)$, is called compactness of cluster i. Since n_i is the number of point in cluster belonging to cluster i, π, is the average variation in cluster i.

Also, the separation of the fuzzy partitions is defined as the minimum distance between cluster centres, that is

$$\mathrm{dmin} = \min \| v_i - v_j \|$$

Then the *XB index* is defined as

$$\mathrm{XB} = \pi / N \cdot \mathrm{dmin}$$

where N is the number of points in the data set.

It is clear that small values of XB are expected for compact and well-separated clusters. We note, however, that XB is monotonically decreasing when the number of clusters nc gets very large and close to n. One way to eliminate this decreasing tendency of the index is to determine a starting point, $c_{\max}$, of the monotonic behaviour and to search for the minimum value of XB in the range $[2, c_{\max}]$. Moreover, the values of the index XB depend on the fuzzifier values, so as if $m \to \infty$ then $\mathrm{XB} \to \infty$.

Another index of this category is the *Fukuyama-Sugeno index*, which is defined as

$$FS_m = \sum_{i=1}^{N} \sum_{j=1}^{nc} u_{ij}^m \left(\|\mathbf{x_i} - v_\mathbf{j}\|_A^2 - \|v_j - v\|_A^2 \right) \tag{22}$$

where v is the mean vector of X and A is an 1×1 positive definite, symmetric matrix. When $A = I$, the above distance become the squared Euclidean distance. It is clear that for compact and well-separated clusters we expect small values for $\mathrm{FS_m}$. The first term in the parenthesis measures the compactness of the clusters and the second one measures the distances of the clusters representatives.

Other fuzzy validity indices are proposed in Gath and Geva (1989), which are based on the concepts of hypervolume and density. Let Σj the fuzzy covariance matrix of the j-th cluster defined as

$$\Sigma_j = \frac{\sum_{i=1}^{N} u_{ij}^m (x_i - v_j)(x_i - v_j)^T}{\sum_{i=1}^{N} u_{ij}^m} \tag{23}$$

The *fuzzy hyper volume* of j-th cluster is given by equation

$$V_j = |\Sigma_j|^{1/2}$$

where $|\Sigma_j|$ is the determinant of Σ_j and is a measure of cluster compactness.

Then the *total fuzzy hyper volume* is given by the equation

$$FH = \sum_{j=1}^{nc} V_j \tag{24}$$

Small values of FH indicate the existence of compact clusters.

The *average partition density* is also an index of this category. It can be defined as

$$PA = \frac{1}{nc} \sum_{j=1}^{nc} \frac{S_j}{V_j} \tag{25}$$

Then $S_j = \sum_{x \in X_j} u_{ij}$, where X_j is the set of data points that are within a pre-specified region around v_j (i.e., the center of j cluster), is called the sum of the central members of the j cluster

A different measure is the *partition density index* that is defined as

$$\mathrm{PD} = \mathrm{S}/\mathrm{FH} \tag{26}$$

where

$$S = \sum_{j=1}^{nc} S_j.$$

A few other indices are proposed and discussed in Krishnapuram et al. (1993), Rezaee et al. (1998).

4.4. Other approaches for cluster validity

Another approach for finding the best number of cluster of a data set proposed in Smyth (1996). It introduces a practical clustering algorithm based on Monte Carlo cross-validation. More specifically, the algorithm consists of M cross-validation runs over M chosen train/test partitions of a data set, D. For each partition u, the EM algorithm is used to define nc clusters to the training data, while nc is varied from 1 to $c_{\max}$. Then, the log-likelihood $L_c^u(D)$ is calculated for each model with nc clusters. It is defined using the probability density function of the data as

$$Lk(D) = \sum_{i=1}^{N} \log f_k(x_i/\Phi_k) \tag{27}$$

where f_k is the probability density function for the data and Φ_k denotes parameters that have been estimated from data. This is repeated M times and the M cross-validated estimates are averaged for each nc. Based on these estimates we may define the posterior probabilities for each value of the number of clusters nc, $p(nc/D)$. If one of $p(nc/D)$ is near 1, there is strong evidence that the particular number of clusters is the best for our data set.

The evaluation approach proposed in Smyth (1996) is based on density functions considered for the data set. Thus, it is based on concepts related to probabilistic models in order to estimate the number of clusters, better fitting a data set, and it does not use concepts directly related to the data, (i.e., inter-cluster and intra-clusters distances).

5. An experimental study of validity indices

In this section we present a comparative experimental evaluation of the important validity measures, aiming at illustrating their advantages and disadvantages.

We consider the known relative validity indices proposed in the literature, such as RS-RMSSTD (Sharma, 1996), DB (Theodoridis and Koutroubas, 1999) and the recent one SD (Halkidi et al., 2000). The definitions of these validity indices can be found in Section 4.3.

RMSSTD and RS have to be taken into account simultaneously in order to find the correct number of clusters. The optimal values of the number of clusters are those for which a significant local change in values of RS and RMSSTD occurs. As regards DB, an indication of the optimal clustering scheme is the point at which it takes its minimum value.

For our study, we used four synthetic two-dimensional data sets further referred to as DataSet1, DataSet2, DataSet3 and DataSet4 (see figure 5a–d) and a real data set Real_Data1 (figure 5e), representing a part of Greek road network (Theodoridis, 1999).

Table 8 summarizes the results of the validity indices (RS, RMSSDT, DB, SD), for different clustering schemes of the above-mentioned data sets as resulting from a clustering algorithm. For our study, we use the results of the algorithms K-means and CURE with

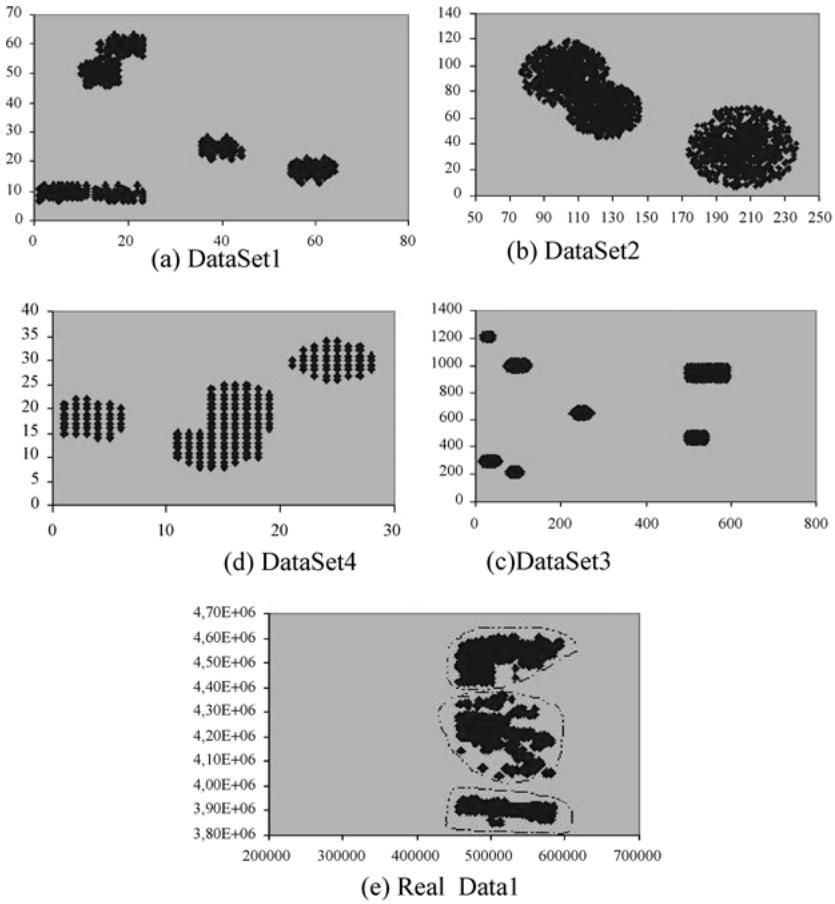

Figure 5. Datasets.

their input value (number of clusters), ranging between 2 and 8. Indices RS, RMSSTD propose the partitioning of DataSet1 into three clusters while DB selects six clusters as the best partitioning. On the other hand, SD selects four clusters as the best partitioning for DataSet1, which is the correct number of clusters fitting the data set. Moreover, the index DB selects the correct number of clusters (i.e., seven) as the optimal partitioning for DataSet3 while RS, RMSSTD and SD select the clustering scheme of five and six clusters respectively. Also, all indices propose three clusters as the best partitioning for Real_Data1. In the case of DataSet2, DB and SD select three clusters as the optimal scheme, while RS-RMSSDT selects two clusters (i.e., the correct number of clusters fitting the data set).

Moreover, SD finds the correct number of clusters (three) for DataSet4, on the contrary to RS–RMSSTD and DB indices, which propose four clusters as the best partitioning.

Table 8. Optimal number of clusters proposed by validity indices RS.

	DataSet1	DataSet2	DataSet3	DataSet4	Real_Data1
	Optimal number of clusters				
RS, RMSSTD	3	2	5	4	3
DB	6	3	7	4	3
SD	4	3	6	3	3

Here, we have to mention that the validity indices are not *clustering algorithms themselves* but a measure to evaluate the results of clustering algorithms and give an indication of a partitioning that best fits a data set. The essence of clustering is not a totally resolved issue and depending on the application domain we may consider different aspects as more significant. For instance, for a specific application it may be important to have well separated clusters while for another to consider more the compactness of the clusters. Having an indication of a good partitioning as proposed by the index, the domain experts may analyse further the validation procedure results. Thus, they could select some of the partitioning schemes proposed by indices, and select the one better fitting their demands for crisp or overlapping clusters. For instance DataSet2 can be considered as having three clusters with two of them slightly overlapping or having two well-separated clusters.

6. Conclusions and trends in clustering

Cluster analysis is one of the major tasks in various research areas. However, it may be found under different names in different contexts such as unsupervised learning in pattern recognition, taxonomy in biology, partition in graph theory. The clustering aims at identifying and extract significant groups in underlying data. Thus based on a certain clustering criterion the data are grouped so that data points in a cluster are more similar to each other than points in different clusters.

Since clustering is applied in many fields, a number of clustering techniques and algorithms have been proposed and are available in literature. In this paper we presented the main characteristics and applications of clustering algorithms. Moreover, we discussed the different categories in which algorithms can be classified (i.e., partitional, hierchical, density-based, grid-based, fuzzy clustering) and we presented representative algorithms of each category. We concluded the discussion on clustering algorithms by a comparative presentation and stressing the pros and cons of each category.

Another important issue that we discussed in this paper is the cluster validity. This is related to the inherent features of the data set under concern. The majority of algorithms are based on certain criteria in order to define the clusters in which a data set can be partitioned. Since clustering is an unsupervised method and there is no a-priori indication for the actual number of clusters presented in a data set, there is a need of some kind of clustering results validation. We presented a survey of the most known validity criteria available

in literature, classified in three categories: external, internal, and relative. Moreover, we discussed some representative validity indices of these criteria along with sample experimental evaluation.

6.1. Trends in clustering process

Though cluster analysis is subject of thorough research for many years and in a variety of disciplines, there are still several open of research issues. We summarize some of the most interesting trends in clustering as follows:

i) *Discovering and finding representatives of arbitrary shaped clusters.* One of the requirements in clustering is the handling of arbitrary shaped clusters and there are some efforts in this context. However, there is no well-established method to describe the structure of arbitrary shaped clusters as defined by an algorithm. Considering that clustering is a major tool for data reduction, it is important to find the appropriate representatives of the clusters describing their shape. Thus, we may effectively describe the underlying data based on clustering results while we achieve a significant compression of the huge amount of stored data (data reduction).

ii) *Non-point clustering.* The vast majority of algorithms have only considered point objects, though in many cases we have to handle sets of extended objects such as (hyper)-rectangles. Thus, a method that handles efficiently sets of non-point objects and discovers the inherent clusters presented in them is a subject of further research with applications in diverse domains (such as spatial databases, medicine, biology).

iii) *Handling uncertainty in the clustering process and visualization of results.* The majority of clustering techniques assumes that the limits of clusters are crisp. Thus each data point may be classified into at most one cluster. Moreover all points classified into a cluster, belong to it with the same degree of belief (i.e., all values are treated equally in the clustering process). The result is that, in some cases "interesting" data points fall out of the cluster limits so they are not classified at all. This is unlikely to everyday life experience where a value may be classified into more than one categories. Thus a further work direction is taking in account the uncertainty inherent in the data. Another interesting direction is the study of techniques that efficiently visualize multidimensional clusters taking also in account uncertainty features.

iv) *Incremental clustering.* The clusters in a data set may change as insertions/updates and deletions occur through out its life cycle. Then it is clear that there is a need of evaluating the clustering scheme defined for a data set so as to update it in a timely manner. However, it is important to exploit the information hidden in the earlier clustering schemes so as to update them in an incremental way.

v) *Constraint-based clustering.* Depending on the application domain we may consider different clustering aspects as more significant. It may be important to stress or ignore some aspects of data according to the requirements of the considered application. In recent years, there is a trend so that cluster analysis is based on less parameters but on more constraints. These constrains may exist in data space or in users' queries. Then a clustering process has to be defined so as to take in account these constrains and define the inherent clusters fitting a dataset.

Acknowledgments

This work was supported by the General Secretariat for Research and Technology through the PENED ("99EΔ 85") project. We thank C. Amanatidis for his suggestions and his help in the experimental study. Also, we are grateful to C. Rodopoulos for the implementation of CURE algorithm as well as to Dr Eui-Hong (Sam) Han for providing information and the source code for CURE algorithm.

References

Berry, M.J.A. and Linoff, G. (1996). *Data Mining Techniques For Marketing, Sales and Customer Support*. John Wiley & Sons, Inc., USA.

Bezdeck, J.C., Ehrlich, R., and Full, W. (1984). FCM: Fuzzy C-Means Algorithm. *Computers and Geoscience*, 10(2–3), 191–203.

Dave, R.N. (1996). Validating Fuzzy Partitions Obtained Through c-Shells Clustering. *Pattern Recognition Letters*, 17, 613–623.

Davies, D.L. and Bouldin, D.W. (1979). A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2), 224–227.

Dunn, J.C. (1974). Well Separated Clusters and Optimal Fuzzy Partitions. *J. Cybern.*, 4, 95–104.

Ester, M., Kriegel, H-P., Sander, J., and Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceeding of 2nd Int. Conf. On Knowledge Discovery and Data Mining*, Portland (pp. 226–23).

Ester, M., Kriegel, H.-P., Sander, J., Wimmer, M., and Xu, X. (1998). Incremental Clustering for Mining in a Data Warehousing Environment. In *Proceedings of 24th VLDB Conference*, New York, USA.

Fayyad, M.U., Piatesky-Shapiro, G., Smuth P., Uthurusamy, R. (1996). *Advances in Knowledge Discovery and Data Mining*. AAAI Press.

Gath I. and Geva A.B. (1989). Unsupervised Optimal Fuzzy Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7), 773–781.

Guha, S., Rastogi, R., and Shim K. (1998). CURE: An Efficient Clustering Algorithm for Large Databases. In *Proceedings of the ACM SIGMOD Conference*.

Guha, S, Rastogi, R., and Shim K. (1999). ROCK: A Robust Clustering Algorithm for Categorical Attributes. In *Proceedings of the IEEE Conference on Data Engineering*.

Halkidi, M., Vazirgiannis, M., and Batistakis, I. (2000). Quality Scheme Assessment in the Clustering Process. In *Proceedings of PKDD*, Lyon, France.

Han, J. and Kamber, M. (2001). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, USA.

Hinneburg, A. and Keim, D. (1998). An Efficient Approach to Clustering in Large Multimedia Databases with Noise. In *Proceedings of KDD Conference*.

Huang, Z. (1997). A Fast Clustering Algorithm to Cluster very Large Categorical Data Sets in Data Mining. *DMKD.*

Jain, A.K., Murty, M.N., and Flyn, P.J. (1999). Data Clustering: A Review. *ACM Computing Surveys*, 31(3), 264–323.

Krishnapuram, R., Frigui, H., and Nasraoui, O. (1993). Quadratic Shell Clustering Algorithms and the Detection of Second-Degree Curves. *Pattern Recognition Letters*, 14(7), 545–552.

MacQueen, J.B. (1967). Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of 5th Berkley Symposium on Mathematical Statistics and Probability*, Volume I: Statistics, pp. 281–297.

Milligan, G.W. and Cooper, M.C. (1985). An Examination of Procedures for Determining the Number of Clusters in a Data Set. *Psychometrika*, 50, 159–179.

Milligan, G.W., Soon, S.C., and Sokol, L.M. (1983). The Effect of Cluster Size, Dimensionality and the Number of Clusters on Recovery of True Cluster Structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5, 40–47.

Mitchell, T. (1997). *Machine Learning.* McGraw-Hill, USA.

Ng, R. and Han, J.(1994). Effecient and Effictive Clustering Methods for Spatial Data Mining. In *Proceeding's of the 20th VLDB Conference*, Santiago, Chile.

Pal, N.R. and Biswas, J. (1997). Cluster Validation Using Graph Theoretic Concepts. *Pattern Recognition*, 30(6), 847–857.

Rezaee, R., Lelieveldt, B.P.F., and Reiber, J.H.C. (1998). A New Cluster Validity Index for the Fuzzy c-Mean. *Pattern Recognition Letters*, 19, 237–246.

Sharma, S.C. (1996). *Applied Multivariate Techniques.* John Wiley and Sons.

Sheikholeslami, C., Chatterjee, S., and Zhang, A. (1998). WaveCluster: A-MultiResolution Clustering Approach for Very Large Spatial Database. In *Proceedings of 24th VLDB Conference*, New York, USA.

Smyth, P. (1996). Clustering using Monte Carlo Cross-Validation. In *Proceedings of KDD Conference*.

Theodoridis, S. and Koutroubas, K. (1999). *Pattern Recognition.* Academic Press.

Theodoridis, Y. (1999). Spatial Datasets: An "unofficial" collection. http://dias.cti.gr/∼ytheod/research/datasets/spatial.html

Wang, W., Yang, J., and Muntz, R. (1997). STING: A Ststistical Information Grid Approach to Spatial Data Mining. In *Proceedings of 23rd VLDB Conference*.

Xie, X.L. and Beni, G. (1991). A Validity Measure for Fuzzy Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4), 841–846.

Zhang, T., Ramakrishnman, R., and Linvy, M. (1996). BIRCH: An Efficient Method for Very Large Databases. *ACM SIGMOD*, Montreal, Canada.