A Review of Clustering Techniques and Developments

Amit Saxena[1], Mukesh Prasad[2], Akshansh Gupta[3], Neha Bharill[4], Om Prakash Patel[4], Aruna Tiwari[4], Meng Joo Er[5], Weiping Ding[6], Chin-Teng Lin[2]

[1]Department of Computer Science & IT, Guru Ghasidas Vishwavidyalaya, Bilaspur, India

[2]Centre for Artificial Intelligence, University of Technology Sydney, Sydney, Australia

[3]School of Computational and Integrative Sciences, Jawaharlal Nehru University, New Delhi, India

[4]Department of Computer Science and Engineering, Indian Institute of Technology Indore, India

[5]School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore

[6]School of Computer and Technology, Nantong University, Nantong, China

Abstract

This paper presents a comprehensive study on clustering: exiting methods and developments made at various times. Clustering is defined as an unsupervised learning where the objects are grouped on the basis of some similarity inherent among them. There are different methods for clustering the objects such as hierarchical, partitional, grid, density based and model based. The approaches used in these methods are discussed with their respective states of art and applicability. The measures of similarity as well as the evaluation criteria, which are the central components of clustering are also presented in the paper. The applications of clustering in some fields like image segmentation, object and character recognition and data mining are highlighted.

Keywords: Unsupervised learning, Clustering, Data mining, Pattern recognition, Similarity measures

1 Introduction

Grouping of objects is required for various purposes in different areas of engineering, science and technology, humanities, medical science and our daily life. Take for an instance, people suffering from a particular disease have some symptoms in common and are placed in a group tagged with some label usually the name of the disease. Evidently, the people not possessing those symptoms (and hence the disease) will not be placed in that group. The patients grouped for that disease will be treated accordingly while patients not belonging to that group should be handled differently. It is therefore so essential for a medical expert to diagnose the symptoms of a patient correctly such that he/she is not placed in a wrong group. Whenever we find a labeled object, we will place it into the group with same label. It is rather a trivial task as the labels are given in advance. However, on many occasions, no such labeling information is provided in advance and we group objects on the basis of some similarity. Both of these instances represent a wide range of problems occurring in analysis of data. In generic terms, these cases are dealt under the scope of classification [1]. Precisely, the first case when the class (label) of an object is given in advance is termed as supervised classification whereas the other case when the class label is not tagged to an object in advance is termed as unsupervised classification. There has been a tremendous amount of work in supervised classification and evidently has been reported in the literature widely [2-9]. The main purpose behind the study of classification is to develop a tool or an algorithm, which can be used to predict the class of an unknown object, which is not labeled. This tool or algorithm is called a classifier. The objects in the classification process are more commonly represented by instances or patterns. A pattern consists of a number of features (also called attributes). The classification accuracy of a classifier is judged by the fact as how many testing patterns it has classified correctly. There has been a rich amount of work in supervised classification, some of the pioneer supervised classification algorithms can be found in neural networks [10, 11], fuzzy sets [12, 13], PSO [14, 15], rough sets [16-18] , decision tree [19], Bayes classifiers [20] etc.

Contrary to supervised classification, where we are given labeled patterns; the unsupervised classification differs in the manner that there is no label assigned to any pattern. The unsupervised classification is commonly known as clustering. As learning operation is central to the process of

classification (supervised or unsupervised), it is used in this paper interchangeably with the same spirit. Clustering is a very essential component of various data analysis or machine learning based applications like, regression, prediction, data mining [21] etc. According to Rokach [22] clustering divides data patterns into subsets in such a way that similar patterns are clustered together. The patterns are thereby managed into a well-formed evaluation that designates the population being sampled. Formally and conventionally, the clustering structure can be represented as a set S of subsets $S_1, S_2, \ldots, S_k$, such that:

$$S_1 \bigcap S_2 \bigcap S_3 ..... \bigcap S_k = \phi \tag{1}$$

This means obviously that any instance in S ($S_1 ... S_k$) belongs to exactly one subset and does not belong to any other subset. Clustering of objects is also applicable for charactering the key features of people in recognizing them on the basis of some similarity. In general, we may divide people in different clusters on the basis of gender, height, weight, color, vocal and some other physical appearances. Hence, clustering embraces several interdisciplinary areas such as: from mathematics and statistics to biology and genetics, where all of these use various terminology to explain the topologies formed using this clustering analysis technique. For example, from biological "taxonomies", to medical "syndromes" and genetic "genotypes" to manufacturing" group technology", each of these topics has same identical problem: create groups of instances and assign each instance to the appropriate groups.

Clustering is considered to be more difficult than supervised classification as there is no label attached to the patterns in clustering. The given label in the case of supervised classification becomes a clue to grouping data objects as a whole. Whereas in the case of clustering, it becomes difficult to decide, to which group a pattern will belong to, in the absence of a label. There can be several parameters or features which could be considered fit for clustering. The curse of dimensionality can add to the crisis. High dimensionality not only leads to high computational cost but also affects the consistency of algorithms. There are although feature selection methods reported as a solution [23]. The sizes of the databases (e.g. small, large or very large) can also guide the clustering criteria.

Jain [24] illustrated that the main aim of data clustering is to search the real grouping(s) of a set of instances, points, or objects. Webster (Merriam-Webster Online Dictionary) [25] explains clustering as ‘‘a statistical classification method for finding whether each of patterns comes into various groups by making quantitative comparisons of different features". It is evident from the above discussion that similarity is the central factor to a cluster and hence clustering process. The natural grouping of data based on some inherent similarity is to be discovered in clustering. In most of the cases, the number of clusters to be formed is specified by the user. As there is only numeric type data available to represent features of the patterns in a group, the only way to extract any information pertaining to the relationship among patterns is to make use of numeric arithmetic. The features of the objects are represented by numeric values. The most common approach to define similarity is taken as a measure of distance among the patterns, lower the distance (e.g. Euclidean distance) between the two objects, higher the similarity and vice versa.

The overall paper is organized as follows. Various clustering techniques will be discussed in Section 2. Section 3 presents measures of similarity for differentiating the patterns. In Section 4, the variants of clustering methods have been presented. The evaluation criteria of the clustering techniques applied for different problems are provided in Section 5. Section 6 highlights some emerging applications of clustering. Section 7 describes which clustering method to select under different applications followed by conclusions in Section 8. Due to a wide range of topics in the subject, the omission or the unbalancing of certain topics presented in the paper cannot be denied. The objective of the paper is however to present a comprehensive timeline study of clustering with its concepts, comparisons, existing techniques and few important applications.

2 Clustering Techniques

In this section, we will discuss various clustering approaches with inherent techniques. The reason for having different clustering approaches towards various techniques is due to the fact that there is no such precise definition to the notion of "cluster" [22, 26]. That is why, different clustering approaches have been proposed, each of which uses a different inclusion principle. Fraley and Raftery [27] suggested dividing the clustering approaches into two different groups: hierarchical and partitioning techniques. Han and Kamber [21] suggested the following three additional categories for applying clustering techniques: density-based methods, model-based methods and grid-based methods. An alternative categorization based on the induction principle of different clustering approaches is presented in Castro et al [26]. However, the number of clusters into which available dataset to be divided, is decided by the users judiciously by using some of the approaches including heuristic, trial and error or evolutionary. If the user decides suitable number, the accuracy judged by intra-cluster distance will be high otherwise the accuracy can become low. Fig. 1 shows the taxonomy of clustering approaches [27].

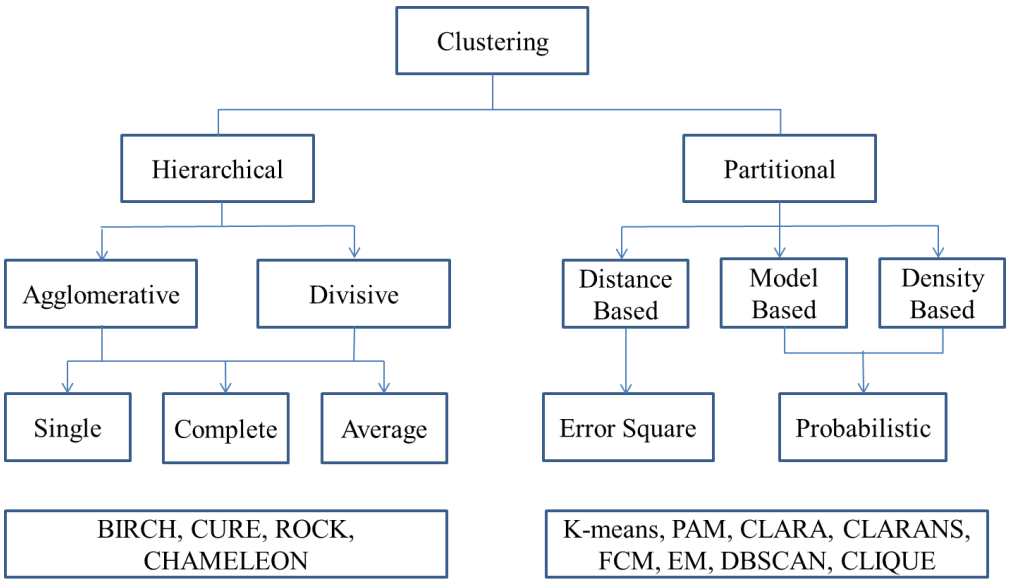


Fig. 1 Taxonomy of clustering approaches [27]

2.1 Hierarchical Clustering (HC) Methods

In hierarchical clustering methods, clusters are formed by iteratively dividing the patterns using top-down or bottom up approach. There are two forms of hierarchical method namely agglomerative and divisive hierarchical clustering [32]. The agglomerative follows the bottom-up approach, which builds up clusters starting with single object and then merging these atomic clusters into larger and larger clusters, until all of the objects are finally lying in a single cluster or otherwise until certain termination conditions are satisfied. The divisive hierarchical clustering follows the top-down approach, which breaks up cluster containing all objects into smaller clusters, until each object forms a cluster on its own or until it satisfies certain termination conditions. The hierarchical methods usually lead to formation of dendrograms as shown in Fig. 2 below.

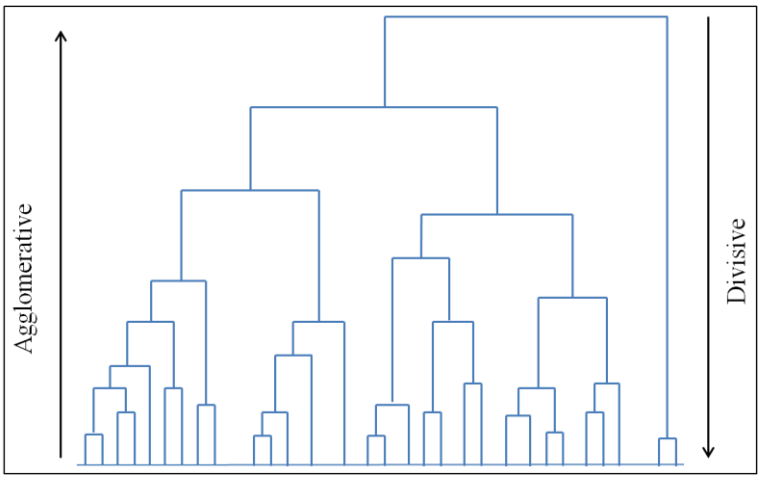


Fig. 2 Hierarchical clustering dendrogram

The hierarchical clustering methods could be further grouped in three categories based on similarity measures or linkages [28] as summarized in following sections.

2.1.1 Single-linkage Clustering

This type of clustering is often called as the connectedness, the minimum method or the nearest neighbour method. In single-linkage clustering, the link between two clusters is made by a single element pair, namely those two elements (one in each cluster) that are closest to each other. In this clustering, the distance between two clusters is determined by nearest distance from any member of one cluster to any member of the other cluster, this also defines similarity. If the data is equipped with similarities, the similarity between a pair of clusters is considered to be equal to the greatest similarity from any member of one cluster to any member of the other cluster [29]. Fig. 3 shows the mapping of single linkage clustering. The criteria between two sets of clusters A and B is as follow:

$$\min\left\{d(a,b):a\in A,b\in B\right\} \tag{2}$$

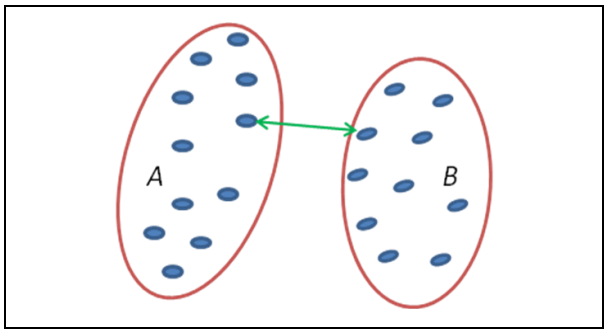


Fig. 3 Mapping of single linkage clustering

2.1.2 Complete-linkage Clustering

In complete-linkage clustering also called the diameter, the maximum method or the furthest neighbour method; the distance between two clusters is determined by longest distance from any member of one cluster to any member of the other cluster [30]. Fig. 4 shows the mapping of complete linkage clustering. The criteria between two sets of clusters A and B is as follow:

$$\max\left\{d(a,b):a\in A,b\in B\right\} \tag{3}$$

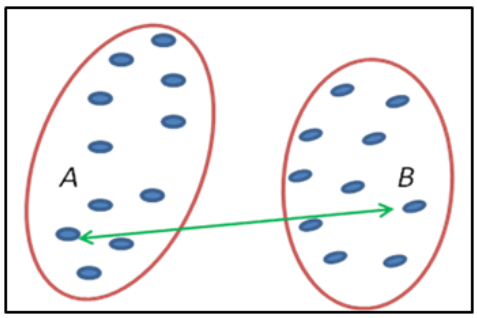


Fig. 4 Mapping of complete linkage clustering

2.1.3 Average-linkage Clustering

In average linkage clustering also known as minimum variance method; the distance between two clusters is determined by the average distance from any member of one cluster to any member of the other cluster [31]. Fig. 5 shows the mapping of average linkage clustering. The criteria between two sets of clusters A and B is as follow:

$$\frac{1}{|A||B|}\sum_{a\in A}\sum_{b\in B} d(a,b) \tag{4}$$

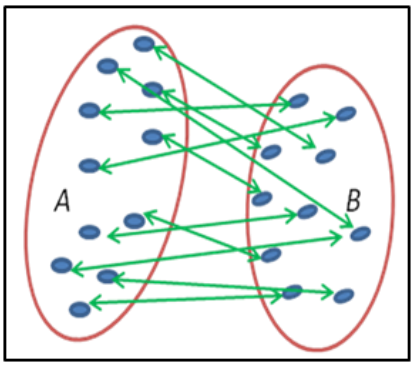


Fig. 5 Mapping of average linkage clustering

2.1.4 Steps of Agglomerative and Divisive Clustering

(i) Steps of agglomerative clustering

1. Make each point a separate cluster
2. Until the clustering is satisfactory
3. Merge the two clusters with the smallest inter-cluster distance
4. End

(ii) Steps of divisive clustering

1. Construct a single cluster containing all points
2. Until the clustering is satisfactory
3. Split the cluster that yields the two components with the largest inter-cluster distance
4. End

The common criticism for classical HC algorithms is that they lack robustness and are, hence, sensitive to noise and outliers. Once an object is assigned to a cluster, it will not be considered again, which means that HC algorithms are not capable of correcting possible previous misclassification. The computational complexity for most of HC algorithms is at least $O(N^2)$ and this high cost limits their application in large-scale data sets. Other disadvantages of HC include the tendency to form spherical shapes and reversal phenomenon, in which the normal hierarchical structure is, distorted [50]. With the requirement of large-scale datasets in recent years, the HC algorithms are also enriched with some new techniques as modifications to classical HC methods presented in following section.

2.1.5 Enhanced Hierarchical Clustering

The main deficiency of hierarchical clustering [33] is that after the two points of the clusters are linked to each other, they cannot move in other clusters in a hierarchy. Few algorithms, which use hierarchical clustering with some enhancements, are given below:

(i) Balanced Iterative Reducing and Clustering Using Hierarchies (BIRCH)

BIRCH [131] contains the idea of cluster features (CF). CF is the triple (n, LS, SS) where n is the number of data objects in the cluster, LS is the linear sum of the attribute values of the objects in the cluster and SS is the sum of squares of the attribute values of the objects in the cluster. These are stored in a CF-tree form, so no need to keep all tuples or all clusters in main memory, but only, their tuples [34]. The main motivations of BIRCH lie in two aspects, the ability to deal with large data sets and the robustness to outliers [131]. Also the BIRCH can achieve a comutational complexity of O(N).

(ii) Clustering Using Representatives (CURE)

CURE [35] is a clustering technique for dealing with large-scale databases, which is robust towards outliers and accepts clusters of various shapes and sizes. Its performance is good with 2-D data sets. BIRCH and CURE both handle outliers well but CURE clustering quality is better than that of BIRCH [35]. On the reverse, in terms of time complexity, BIRCH is better than CURE as it attains computational complexity of $O(N)$ compared to CURE $O(N^2 \log N)$.

(iii) ROCK

ROCK [130] is applied for categorical data sets which follows the agglomerative hierarchical clustering algorithm. It is based on the number of links between two records; links capture the number of other records, which are very similar to each other. This algorithm does not use any distance function. CURE [35] also proposed ROCK, which uses a random sample strategy to handle large datasets.

(iv) CHAMELEON

CHAMELEON [36] is a hierarchical clustering algorithm, where clusters are merged only if the interconnectivity and closeness (proximity) between two clusters are high relative to the internal interconnectivity of the clusters and closeness of items within the clusters. One limitation of CHAMELEON is that it is known for low dimensional spaces, and was not applied to high dimensions.

Table1 Features of hierarchical clustering-based enhanced methods

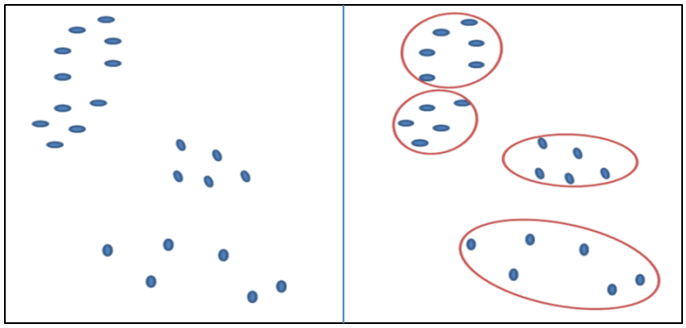
Name	Type of data	Complexity	Ability to handle high dimensional data
BIRCH	Numerical	$O(N)$	No
CURE	Numerical	$O(N^2 \log N)$	Yes
ROCK	Categorical	$O(N^2+Nm_m m_a+N^2 \log N)$*	No
CHEMELEON	Numerical/ Categorical	$O(Nm + N\log N + m^2 \log N)$**	No

*m_m is the maximum number of neighbours for a point m_a is the average number of neighbours for a point.

**m is the number of initial sub-clusters produced by the graph partitioning algorithm.

2.2 Partition Clustering Methods

Partitional clustering is opposite to hierarchical clustering; here data are assigned into K clusters without any hierarchical structure by optimizing some criterion function [37]. The most commonly used criterion is the Euclidean distance, which finds the minimum distance between points with each of the available clusters and assigning the point to the cluster. The algorithms [33] studied in this category include: k-means [38], PAM [173], CLARA [173], CLARANS [174], Fuzzy C-means, DBSCAN etc. Fig. 6 shows the partitional clustering approach.



Data points Partitional clusters

Fig. 6 Partitional clustering approaches

2.2.1 K-means Clustering

K-means algorithm is one of the best-known, bench marked and simplest clustering algorithms [37, 38], which is mostly applied to solve the clustering problems. In this procedure the given data set is classified through a user defined number of clusters, *k*. The main idea is to define *k* centroids, one for each cluster. The objective function *J* is given as follows:

$$\textit{Minimize} \quad J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 \tag{5}$$

where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j, Fig. 7 shows the flow diagram of K-means algorithm.

An algorithm similar to k-means, known as the Linde-Buzo-Gray (LBG) algorithm, was suggested for vector quantization (VQ) [39] for signal compression. In this context, prototype vectors are called code words, which constitute a code book. VQ aims to represent the data with a reduced number of elements while minimizing information loss. Although K- Means clustering is still one of the most popular clustering algorithms yet few limitation are associated with K Means clustering include: (a) There is no efficient and universal method for identifying the initial partitions and the number of clusters K and (b) K-means is sensitive to outliers and noise. Even if an object is quite far away from the cluster centroid, it is still forced into a cluster and, thus, distorts the cluster shapes [50].

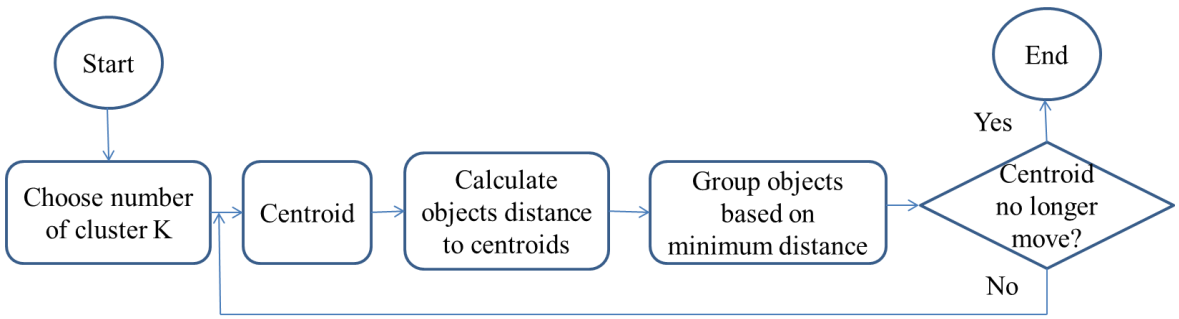


Fig. 7 Flow diagram of *K* -means algorithm

The procedure of K-means algorithm is composed of the following steps:

1. ***Initialization****: Suppose we decide to form K clusters of the given dataset. Now take K distinct points (patterns) randomly. These points represent initial group centroids. As these centroids will be changing after each iteration before clusters are fixed, there is no need to spend time in decision of choosing the centroids.*
2. *Assign each object to the group that has the closest centroid.*
3. *When all objects have been assigned, recalculate the positions of the K centroids.*
4. *Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.*

2.2.2 Fuzzy C-means Clustering

Fuzzy c-means (FCM) is a clustering method which allows one point to belong to two or more clusters unlike *K*-means where only one cluster is assigned to each point. This method was developed by Dunn in 1973 [40] and improved by Bezdek in 1981 [41]. The procedure of fuzzy c-means [50] is similar to that of K-means. It is based on minimization of the following objective function:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{c} u_{ij}^m \| x_i - v_j \|^2 ; 1 < m < \infty \tag{6}$$

where m is fuzzy partition matrix exponent for controlling the degree of fuzzy overlap, with $m > 1$. Fuzzy overlap refers to how fuzzy the boundaries between clusters are, that is the number of data points that have significant membership in more than one cluster, u_{ij} is the degree of membership of x_i in the cluster j, x_i is the i-th pattern of d-dimension data, v_j is j-th cluster center of the d-dimension and $\|*\|$ is any norm expressing the similarity between any measured data and the center.

Procedure for FCM

1. *Set up a value of c (number of cluster);*
2. *Select initial cluster prototype* $V_1, V_2, \ldots\ldots, V_c$ *from* X_i , $i = 1, 2, \ldots\ldots, N$;
3. *Compute the distance* $\|X_i - V_j\|$ *between objects and prototypes;*
4. *Computer the elements of the fuzzy partition matrix* ($i = 1, 2, \ldots\ldots, N$; $j = 1, 2, \ldots\ldots, c$)
$$u_{ij} = \left[\sum_{l=1}^{c} \left(\frac{\|x_i - v_j\|}{\|x_i - v_l\|} \right) \right]^{-1}$$
5. *Compute the cluster prototypes* ($j = 1, 2, \ldots\ldots, c$)
$$V_j = \frac{\sum_{i=1}^{N} u_{ij}^2 x_i}{\sum_{i=1}^{N} u_{ij}^2}$$
6. *Stop if the convergence is attained or the number of iterations exceeds a given limit. Otherwise, go to step 3.*

FCM suffers from initial partition dependence, as well as noise and outliers like k-means. Yager and Filev [42] proposed the mountain method to estimate the cluster centers as an initial partition. Gath and Geva [43] addressed the initialization problem by dynamically adding cluster prototypes, which are located in the space that is not represented well by the previously generated centers.

Changing the proximity distance can improve the performance of FCM in relation to outliers [44]. In another approach for reducing the effect of noise and outliers, Keller [45] interpreted memberships as "the compatibility of the points with the class prototype" rather than as the degree of membership. This relaxes $u_{ij} = 1$ to $u_{ij} > 0$ and results in a possibilistic K-means clustering algorithm.

The conditions for a possibilistic fuzzy partition matrix are:

$$u_{ij} \in [0,1], 1 \le i \le N,\ 1 \le j \le C \tag{7}$$

$$\exists_j,\ u_{ij} > 0,\ \forall i \tag{8}$$

$$0 < \sum_{i=1}^{N} u_{ij} < N,\ 1 \le j \le C \tag{9}$$

Table 2 Features of partition clustering based techniques

Name	Type of data	complexity	Ability to handle high dimensional data
K-Mean	Numerical	$O(N)$	No
PAM	Numerical	$O(K(N-K)^2)$*	No
CLARA	Numerical	$O(K(40+K)^2+K(N-K))$	No
CLARANS	Numerical	$O(KN^2)$	No
Fuzzy C-Means	Numerical	$O(N)$	No

*N is the number of points in the dataset and K is the number of clusters defined.

The k-means algorithms have problems like defining the number of clusters initially, susceptibility to local optima, and sensitivity to outliers, memory space and unknown number of iteration steps that are required to cluster. The fuzzy C means clustering are really suitable for handling the issues related to understand ability of patterns, incomplete/noisy data, mixed media information, human interaction and it can provide approximate solutions faster. They have been mainly used for discovering association rules and functional dependencies as well as image retrieval. However the time complexity of K means is much less than that of FCM thus K means works faster than FCM [191].

Some of the advantages of partition based algorithms includes that they are (i) relatively scalable and simple and (ii) suitable for datasets with compact spherical clusters that are well-separated. However, disadvantages with these algorithms include poor (i) cluster descriptors (ii) reliance on the user to specify the number of clusters in advance (iii) high sensitivity to initialization phase, noise and outliers and (iv) inability to deal with non-convex clusters of varying size and density [175].

3 Measures of Similarities

Similarity of objects within a cluster plays the most important role in clustering process. A good cluster finds maximum similarity among its objects. The measure of similarity in cluster is mainly decided by the distance among its members. In a conventional cluster (non-fuzzy), a member either belongs to a cluster wholly or not at all. Many clustering methods use distance measures to determine the similarity or dissimilarity between any pair of objects [22]. It is useful to denote the distance between two instances x_i and x_j as: $d(x_i, x_j)$. A valid distance measure should be symmetric i.e $d(x_i, x_j) = d(x_j, x_i)$ and obtain its minimum value (ideally zero) in case of identical vectors. The distance measure is called a metric distance measure if it also satisfies the following properties:

Triangle inequality $$d(x_i, x_k) \le d(x_i, x_j) + d(x_j, x_k)\ \forall x_i, x_j, x_k \in S \tag{10}$$

$$d(x_i, x_j) = 0 \Rightarrow x_i = x_j\ \forall x_i, x_j \in S \tag{11}$$

3.1 Minkowski: Distance Measures for Numeric Attributes

A measurement of distance is a fundamental operation in the unsupervised learning process [91]. Smaller is the distance between any two objects; closer these objects are assumed on the basis of similarity. A family of distance measures is the Minkowski metrics [29], where the distance is measured by following equation

$$\|ij\|_r = \left\{ \sum_{k=1}^{d} \left| x_{ik} - x_{jk} \right|^r \right\}^{1/r} \tag{12}$$

where x_{ik} is the value of the k-th variable for entity i, x_{jk} is the value of the k-th variable for entity j. The most popular and common distance measure is the Euclidean or L_2 norm (r =2). More details on unsupervised classification for various non-Euclidean distances can be seen in Saxena et al. [160].

3.2 Cosine Measure

Cosine Measure [153] is a popular similarity score in text mining and information retrieval [152]. The normalized inner product for Cosine measure is defined as:

$$d(x_i, x_j) = \frac{x_i \cdot x_j}{\|x_i\| \cdot \|x_j\|} \tag{13}$$

3.3 Pearson Correlation Measure

Correlation coefficient is first discovered by Bravais [154] and later shown by Person [155]. The normalized Pearson correlation for two vectors x_i and x_j is defined as:

$$d(x_i, x_j) = \frac{(x_i - \bar{x}_i) \cdot (x_j - \bar{x}_j)}{\|x_i - \bar{x}_i\| \cdot \|x_j - \bar{x}_j\|} \tag{14}$$

where $\bar{x}_i$ denotes the average feature value of x over all dimensions.

3.4 Extended Jaccard Measure

Strehl and Ghosh [107] represented the extended Jaccard measure as follows:

$$d(x_i, x_j) = \frac{x_i^T x_j}{\|x_i\|^2 + \|x_j\|^2 - x_i^T \cdot x_j} \tag{15}$$

3.5 Dice Coefficient Measure

It was independently developed by the Thorvald Sørensen[156] and Raymond Dice [157] The dice coefficient measure is similar to the extended Jaccard measure and it is defined as:

$$d(x_i, x_j) = \frac{2 x_i^T x_j}{\|x_i\|^2 + \|x_j\|^2} \tag{16}$$

3.6 Choice of Suitable Similarity Measure

The measures of similarities have been applied on millions of applications in clustering. In fact every clustering problem applies one of the similarity measures. The Euclidean distance is mostly applied to find similarity between two objects, which are expressed numerically. Euclidean distance is highly sensitive to noise and usually not applied to data with hundreds of attributes also features with high values tend to dominate others [50] so it may be applied when translations of non-numeric objects to numeric values are almost nil or minimum. Jaccard similarity coefficient is suitable sufficiently to be employed in the documents or word similarity measurement. In efficiency measurement, the program performance can deal appropriately with high stability when failure and mistake spelling occurred. Nevertheless, this method is not able to detect the over-type words in the data sets [192]. Pearson correlation is usually unable to detect the difference between two variables [50]. Cosine similarity is

also a good choice for document clustering, it is invariant to rotation but not to linear transformations [50].

4 Variants of Clustering Methods

4.1 Graph (Theoretic) Clustering

The graph theoretic clustering is a method that represents clusters via graphs. The edges of the graph connect the instances represented as nodes. A well-known graph-theoretic algorithm is based on the minimal spanning tree (MST) [46]. Inconsistent edges are edges whose weight (in the case of clustering length) is significantly larger than the average of nearby edge lengths. Another graph theoretic approach constructs graphs based on limited neighbourhood sets [47]. The graph theoretic clustering is convenient to represent clusters via graphs but is weak in handling outliers especially in MST as well as detecting overlapping of clusters [176].

The graph clustering [177] involves the task of dividing nodes into clusters, so that the edge density is higher within clusters as opposed to across clusters. A natural, classic and popular statistical setting for evaluating solutions to this problem is the stochastic block model, also referred to as the planted partition model. The general graph l-partition problem is to partition the nodes of an undirected graph into l equal-sized groups so as to minimize the total number of edges that cross between groups. Condon [178] presented a simple, linear-time algorithm for the graph l-partition problem and analyzed it on a random "planted l-partition" model. In this model, the n nodes of a graph are partitioned into l groups, each of size n/l; two nodes in the same group are connected by an edge with some probability p, and two nodes in different groups are connected by an edge with some probability $r<p$. They showed that if $p-r \geq n-1/2+\epsilon$ for some constant ϵ, then the algorithm finds the optimal partition with probability $1-\exp(-n\Theta(\varepsilon))$. Graph clustering decomposes a network into sub networks based on some topological properties. In general we look for dense sub networks as shown in Fig. 8.

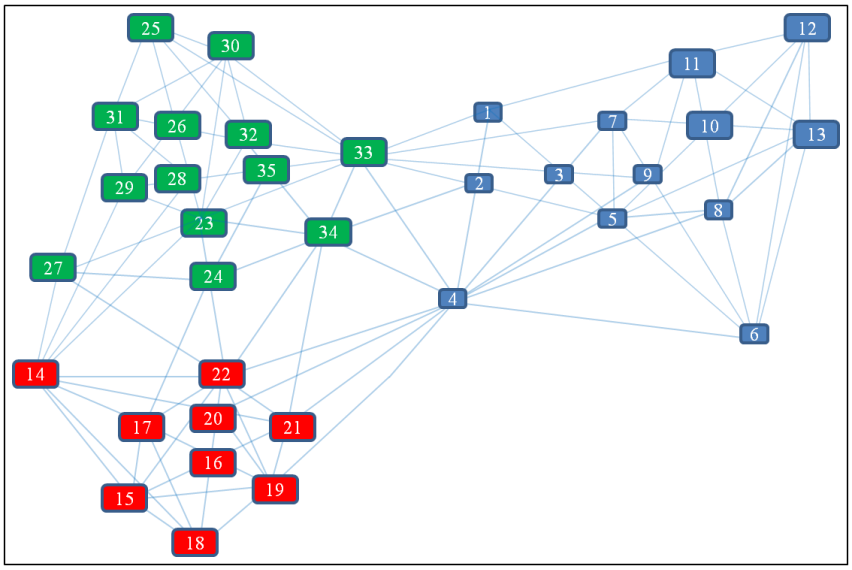


Fig. 8 Sub-network clustering of graph

Spectral Clustering, proposed by Donath and Hoffman [179], is an emerging technique under graph clustering which consists of algorithms cluster points using eigenvectors of matrices derived from the data. In the machine learning community, spectral clustering has been made popular by the works of Shi and Malik [180]. A useful tutorial is available on spectral clustering by Luxburg [181]. The success of spectral clustering is mainly based on the fact that it does not make strong assumptions on the form of the clusters. As opposed to k-means, where the resulting clusters form convex sets (or, to be precise, lie in disjoint convex sets of the underlying space), spectral clustering can solve very

general problems like intertwined spirals. Moreover, spectral clustering can be implemented efficiently even for large data sets, as long as we make sure that the similarity graph is sparse. Once the similarity graph is chosen, we just have to solve a linear problem, and there are no issues of getting stuck in local minima or restarting the algorithm for several times with different initializations. However, we have already mentioned that choosing a good similarity graph is not trivial, and spectral clustering can be quite unstable under different choices of the parameters for the neighborhood graphs. So spectral clustering cannot serve as a "black box algorithm" which automatically detects the correct clusters in any given data set. But it can be considered as a powerful tool which can produce good results if applied with care [181]. More literature (partially) on graph and spectral clustering can be seen in [182-190].

4.2 Spectral Clustering Algorithms [181]

Now we would like to state the most common spectral clustering algorithms. We assume that our data consists of *n* "points" $x_1, \ldots, x_n$, which can be arbitrary objects. We measure their pair wise similarities $s_{ij} = s(x_i, x_j)$ by some similarity function which is symmetric and non-negative, and we denote the corresponding similarity matrix by $S = (s_{ij})I, j=1, ..., n$.

4.2.1 Un-normalized Spectral Clustering

1. ***Input**: Similarity matrix $S \in R^{n \times n}$, number k of clusters to construct.*
2. *Construct a similarity graph by one of the ways described in Section* 2 [181]. *Let W be its weighted adjacency matrix.*
3. *Compute the un-normalized Laplacian L.*
4. *Compute the first k eigenvectors $u_1, \ldots, u_k$ of L.*
5. *Let $U \in R^{n \times k}$ be the matrix containing the vectors $u_1, \ldots, u_k$ as columns.*
6. *For i = 1, . . . , n, let $y_i \in R^k$ be the vector corresponding to the i-th row of U.*
7. *Cluster the points $(y_i)i=1,...,n$ in R^k with the k-means algorithm into clusters $C_1, \ldots, C_k$.*
8. ***Output**: Clusters $A_1, \ldots, A_k$ with $A_i = \{j | y_j \in C_i\}$.*

4.2.2 Normalized Spectral Clustering According to Shi and Malik (2000)[180]

1. ***Input**: Similarity matrix $S \in R\ n \times n$, number k of clusters to construct.*
2. *Construct a similarity graph by one of the ways described in Section* 2 [181]. *Let W be its weighted adjacency matrix.*
3. *Compute the unnormalized Laplacian L.*
4. *Compute the first k generalized eigenvectors u1, . . . , uk of the generalized eigen problem $Lu = \lambda Du$.*
5. *Let $U \in R\ n \times k$ be the matrix containing the vectors u1, . . . , uk as columns.*
6. *For i = 1, . . . , n, let $yi \in R\ k$ be the vector corresponding to the i-th row of U.*
7. *Cluster the points (yi)i=1,...,n in R k with the k-means algorithm into clusters C1, . . . , Ck.*
8. ***Output**: Clusters A1, . . . , Ak with $Ai = \{j | yj \in Ci\}$.*

4.3 Model Based Clustering Methods

Model based clustering methods optimize as well as find the suitability of given data with some mathematical models. Similar to conventional clustering; model-based clustering methods also detect feature details for each cluster, where each cluster represents a concept or class. Decision trees and neural networks are two most frequently used induction methods.

(i) Decision Trees

The representation of data in decision tree [19] is modelled by a hierarchical tree, in which each leaf denotes a concept and implies a probabilistic description of that concept. There are many algorithms, which produce classification trees for defining the unlabelled data. Number of algorithms that have been proposed for conceptual clustering are follows: CLUSTER/2 by Michalski and Stepp [93], COBWEB by Fisher [48], CYRUS by Kolodner [95], GALOIS by Carpineto and Romano [96], GCF by Talavera and Béjar [97], INC by Hadzikadic and Yun [98], ITERATE by Biswas, Weinberg and Fisher [99], LABYRINTH by Thompson and Langley [100], SUBDUE by Jonyer, Cook and Holder [101], UNIMEM by Lebowitz [102] and WITT by Hanson and Bauer [103]. COBWEB is one of the best known algorithms, where each concept defines a set of objects and each object defined as a binary values property list. Its aim is to achieve high predictability of nominal variable values, given a cluster. This algorithm is not suitable for clustering large database data [48].

(ii) Neural Networks

Neural networks [49] represent each cluster by a neuron, whereas input data is also represented by neurons, which are connected to the prototype neurons. Each connection is attributed by some weight, which is initialized randomly before learning of these weights adaptively. A very popular neural algorithm for clustering is the self-organizing map (SOM) [104, 105]. SOM is commonly used for vector quantization, feature extraction and data visualization along with clustering analysis. This algorithm constructs a single-layered network as shown in Fig. 9. The learning process takes place in a "winner-takes-all" fashion: The prototype neurons compete for the current instance. The winner is the neuron whose weight vector is closest to the instance currently presented. The winner and its neighbours learn by having their weights adjusted. While SOFMs has the merits of input space density approximation and independence of the order of input patterns, a number of user dependent parameters cause problems when applied in real practice. Like the K-means algorithm,SOFM need to predefine the size of the lattice, i.e., the number of clusters, which is unknown for most circumstances. Additionally, trained SOFM may be suffering from input space density mis representation [49], where areas of low pattern density may be over represented and areas of high density under represented [50].

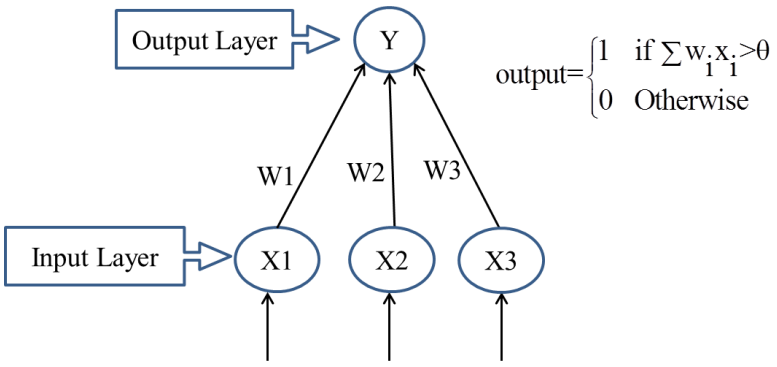


Fig. 9 Model of a single layered network

4.4 Mixture Density-Based Clustering

Xu and Wunsch [50, 51] described clustering in the perspective of probability that data objects are drawn from a specific probability distribution and the overall distribution of the data is assumed to be a mixture of several distributions [53]. Data points [117] can be derived from different types of density

functions (e.g., multivariate Gaussian or *t*-distribution), or from the same families but with different parameters. The aim of these methods is to identify the clusters and their distribution. Cheeseman and Stutz introduced an algorithm named AUTOCLASS [55], which is widely used and covers a broad variety of distributions, including Gaussian, Bernoulli, Poisson, and log-normal distributions. Ester et al. [54] demonstrated an algorithm called DBSCAN (density-based spatial clustering of applications with noise), which discovers clusters of arbitrary shapes and is efficient for large spatial databases.

Other well-known density-based techniques are: SNOB proposed by Wallace and Dowe in 1994 [56] and MCLUST introduced by Fraley and Raftery in 1998 [27]. Among these methods, the expectation-maximization (EM) algorithm is the most popular [52, 56]. For EM algorithm, the log likelihood function to maximize is as follows:

$$\ln p(X \mid \Theta) = \ln \sum_{Y} p(X, Y \mid \Theta) \tag{17}$$

where X denotes the set of all observed data $\left(X = \left\{ \overrightarrow{x_1}, ..., \overrightarrow{x_N} \right\} \right)$, and Y denotes the set of all latent variables $\left(Y = \left\{ \overrightarrow{y_1}, ..., \overrightarrow{y_N} \right\} \right)$. The complete data set is formed as $(X, Y) = \left\{ \left(\overrightarrow{x_i}, \overrightarrow{y_i} \right) \right\}$ and the joint distribution $p\left(\vec{x}, \vec{y} \mid \Theta \right)$ is ruled by a set of parameters. The major disadvantages for EM algorithm are the sensitivity to the selection of initial parameters, the effect of a singular co-variance matrix, the possibility of convergence to a local optimum, and the slow convergence rate [50] [52].

Procedure of EM algorithm

1. *Initialize the parameters* Θ^{old}
2. *E step: evaluate* $p(Y \mid X, \Theta^{old})$
3. *M step: re-estimate the parameters* $\Theta^{new} = \arg\max_{\Theta} L(\Theta)$
4. *Check for convergence. If the convergence criterion is not satisfied, let* $\Theta^{old} \leftarrow \Theta^{new}$ *and return to step 2.*

4.5 Grid-Based Clustering Methods

These methods partition the space into a finite number of cells that form a grid structure on which all of the operations for clustering are performed. The main advantage of the approach is its fast processing time [122], no need of distance computations and easy to determine which clusters are neighbouring.

The basic steps of Grid based algorithm

1. *Define a set of grid cells*
2. *Assign objects to the appropriate grid cell and compute the density of each cell*
3. *Eliminate cells, whose density is below a certain threshold*
4. *Form clusters from contiguous groups of dense cells*

There are many others interesting grid based techniques including: STING (statistical information grid approach) by Wang, Yang and Muntz [57] in 1997, one of the highly scalable algorithm and has the ability to decompose the data set into various levels of detail. STING retrieves spatial data and divides into rectangular cells corresponding to different levels of resolution as shown in Fig. 10.

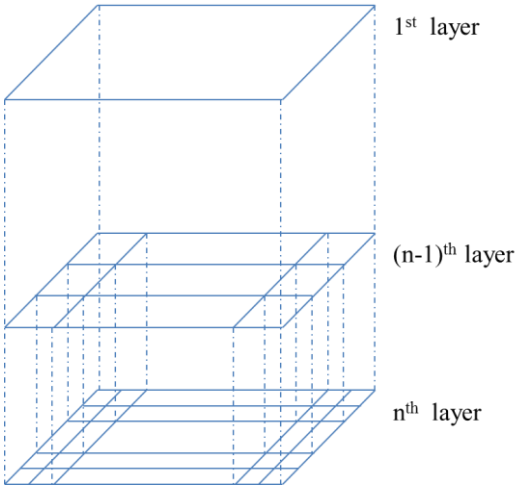


Fig. 10 Rectangular cells corresponding to different levels of resolution

Each cell at a higher level is partitioned into a number of smaller cells in the next lower level. Then mean, variance, minimum, maximum of each cell is computed by using the normal and uniform distribution. Statistical information of each cell is calculated and stored in advance and it uses a top down approach to answer spatial data queries. Wave Cluster [58] introduced by Sheikholeslami et al. [58] uses multi-resolution approach like STING and allows natural clustering to become more distinguishable. It uses a signal processing technique that decomposes a signal into different frequency sub-band and data are transformed to preserve relative distance between objects at different levels of resolution. It is highly scalable and can handle outliers well. It is not suitable for high dimensional data set. It can be considered as both grid-based and density-based. CLIQUE is developed by Agrawal et al. [59] in 1998, which can be considered as both density-based and grid based clustering methods. It automatically finds subspaces of high dimensional data space that allow better clustering than original space. The accuracy of the clustering result may be degraded at the expense of simplicity of the method CLIQUE.

4.6 Evolutionary Approaches Based Clustering Methods

The famous evolutionary approaches [60] include evolution strategies (ES) [61], evolutionary programming (EP) [62], genetic algorithm (GA) [63, 64], particle swarm optimization (PSO) [65-66], ant colony optimization (ACO) [67] etc.

The common approach of evolutionary techniques to data clustering is as follows:

1. *Choose a random population of solutions. Each solution here corresponds to valid k partitions of the data.*
2. *Associate a fitness value with each solution. Typically fitness is inversely proportional to the squared error value. Higher the error, smaller the fitness and vice versa.*
3. *A solution with a small squared error will have a larger fitness value.*
4. *Use the evolutionary operators' viz. selection, recombination and mutation to generate the next population of solutions.*
5. *Evaluate the fitness values of these solutions.*
6. *Repeat step until some termination condition is satisfied.*

Out of these approaches, GA has been most frequently used in clustering, where solutions are in the form of binary strings. In GAs, a selection operator propagates solutions from the current generation to the next generation based on their fitness Selection employs a probabilistic scheme so

that solutions with higher fitness have a higher probability of getting reproduced. A major problem with GAs is their sensitivity to the selection of various parameters such as population size crossover and mutation probabilities etc. Grefenstette [123] has studied this problem and suggested guidelines for selecting these control parameters.

The general steps of GA for clustering are:

Input: S (instance set), K (number of clusters), n (population size)

Output: clusters

1. *Randomly create a population of n structures; each corresponds to valid K-clusters of the data.*
2. *repeat*
    a. *Associate a fitness value $\forall$ structure $\in$ population.*
    b. *Regenerate a new generation of structures.*
3. *until some termination condition is satisfied*

4.7 Search Based Clustering Approaches

Search techniques are basically used to obtain the optimum value (minimum or maximum) of the criterion function (e.g. distance) called objective function also. The search based approaches are categorized into stochastic and deterministic search techniques. The stochastic search techniques can evolve an approximate optimal solution (based on fitness value). Most of the stochastic techniques are evolutionary approaches based. The rest of the search techniques come under deterministic search techniques which guarantee an optimal solution by performing exhaustive enumeration. The deterministic approaches are typically greedy descent approaches. The stochastic search techniques are either sequential or parallel such as simulated annealing (SA) [172] while evolutionary approaches are inherently parallel. Simulated annealing procedures are designed to avoid or recover from solutions which correspond to local optima of the objective functions. This is accomplished by accepting with some probability a new solution for the next iteration of lower quality as measured by the criterion function. The probability of acceptance is governed by a critical parameter called the temperature by analogy with annealing in metals which is typically specified in terms of a starting first iteration and final temperature value. Al Sultan et al [92] studied the effects of control parameters on the performance of the algorithm and used SA to obtain near optimal partition of the data SA is statistically guaranteed to find the global optimal solution.

The SA algorithm can be slow in reaching the optimal solution because optimal results require the temperature to be decreased very slowly from iteration to iteration. Tabu search [68, 69] like SA is a method designed to cross boundaries of feasibility or local optimality and to systematically impose and release constraints to permit exploration of otherwise forbidden regions. Tabu search was used to solve the clustering problem in [3].

4.8 Collaborative Fuzzy Clustering

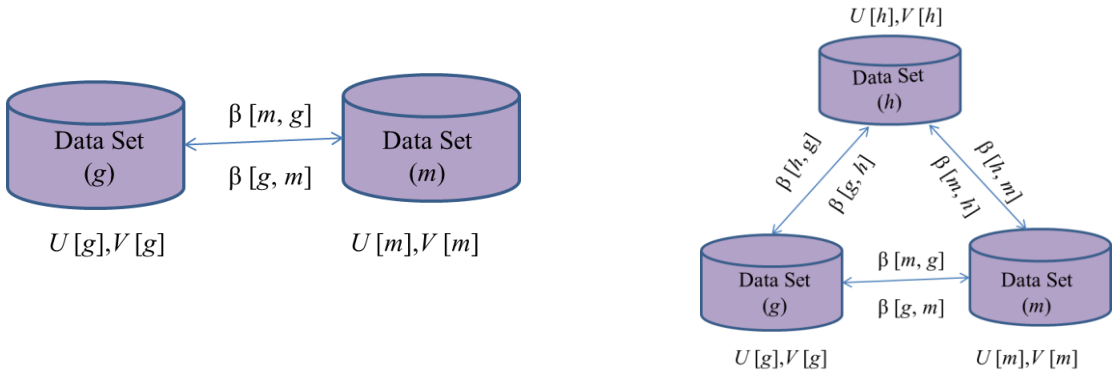
This is relatively a recent type of clustering which has various applications. The database is distributed on several sites. The collaborative clustering proposed by Pedrycz [70-73] concerns a process of revealing a structure being common or similar to a number of subsets. There are mainly two forms of collaborative clustering; horizontal and vertical collaborative clustering [74]. In horizontal collaborative clustering, same database is split into different subsets of features, each subset having all patterns in the database. The horizontal collaborative clustering has been applied for Mamdani type fuzzy inference system [124] in order to decide some association between datasets. In vertical collaborative clustering, database is divided into subsets of patterns such that each pattern of any subset has all features.

The objective function for horizontal collaboration technique is explained in Eq. (13). For vertical collaboration technique, please refer [73]:

$$Q[l] = \sum_{i=1}^{N}\sum_{j=1}^{c} u_{ij}^{2}[l]\, d_{ij}^{2}[l] + \sum_{\substack{m=1 \\ m\neq l}}^{p} \beta[l,m] \sum_{i=1}^{N}\sum_{j=1}^{n} \{u_{ij}[l] - u_{ij}[m]\}^{2} d_{ij}^{2}[l] \tag{18}$$

where β is a user defined parameter based on datasets (β >0), $\beta[l,m]$ denotes the collaborative coefficient with collaborative effect on dataset *l* through *m*, *c* is a number of cluster. $l = 1,2,\ldots\ldots,P$. P is a number of datasets, N is the number of patterns in the dataset, *u* represents the partition matrix, n is a number of features, and d is an Euclidean distance between patterns and prototypes.

The general scheme of collaborative clustering is shown in Fig. 11, which demonstrates the connections of matrices in order to accomplish the collaboration between the subsets of the dataset. First, we solve the problem for each dataset separately and allow the results to interact globally by forming a collaborative process between the datasets. Collaborative fuzzy partitioning is carried out through an iterative optimization of the objective function as shown in Eq. (13). The optimization of *Q*[*l*] involves the determination of the partition matrix *U* and the prototypes *V* of different data sets as shown in Fig. 11(a) and (b).



(a) Collaborative clustering scheme for two datasets (b) Collaborative clustering scheme for three datasets

Fig. 11 Collaborative clustering scheme

4.9 Multi Objective Clustering

In case of multi-objective clustering, many clustering approaches are optimized simultaneously. In multi-objective clustering with automatic k-determination (MOCK) [78, 79], compactness of clusters is maximized as the first objective while the connectivity of the clusters is maximized as the second objective. The Pareto [80] approach is used to optimize the aforesaid two objectives simultaneously. The multi objective clustering ensemble (MOCE) proposed by Faceili et.al [81] uses MOCK along with a special crossover operator which utilizes ensemble clustering. In Law et. al [82], different clustering methods with different objectives are used. Some more surveys can be seen in [50].

4.10 Overlapping Clustering or Overlapping Community Detection

The partition clustering usually indicates exclusive and overlapping clustering algorithms (like k-means discussed above) such that each member or the object belongs to just one cluster. When an object belongs to more than one cluster, it becomes overlapping clustering method or algorithm, e.g. fuzzy c-means clustering. Nowadays, community detection, as an effective way to reveal the relationship between structure and function of networks, has drawn lots of attention and been well developed [195]. Networks are modeled as graphs, where nodes represent objects and edges represent interactions among them. Community detection divides a network into groups of nodes, where nodes are densely connected inside but sparsely connected outside. However, in real world, objects often have diverse roles and belong to multiple communities. For example, a professor collaborates with researchers in different fields and a person has his family group as well as friend group at the same

time. In community detection, these objects should be divided into multiple groups, which are known as overlapping nodes [196]. The aim of overlapping community detection is to discover such overlapping nodes and communities. Until now, lots of overlapping community detection approaches have been proposed, which can be roughly divided into two categories: node-based and link-based algorithms. The node-based overlapping community detection algorithms [75, 76] directly divide nodes of the network into different communities. Based on an intuition that a link in networks usually represents the unique relation, the link-based algorithms firstly cluster on edges of network, and then map the link communities to node communities by gathering nodes incident to all edges within each link community [77]. The newly proposed link-based algorithms have shown its superiority on detecting complex multi-scale communities. However, they have the high computational complexities and bias on the discovered communities. Shi et. al. [196] proposed a genetic algorithm, GaoCD, for overlapping community detection based on the link clustering framework. Different from those node-based overlapping community detection algorithms, GaoCD utilized the property of the unique role of links and applies a novel genetic algorithm to cluster on edges. Experiments on artificial and real networks showed that GaoCD can effectively reveal overlapping structure.

5 Evaluation Criteria

The formation of clusters is an important process. However, it is also meaningful to test the validity and accuracy of the clusters so formed by any method. It should be tested whether the clusters formed by a certain method show maximum similarity among the objects in the same cluster and minimum similarity among those in other clusters. Recently, many evaluation criteria have been developed. These criteria are divided mainly into two categories: Internal and External.

5.1 Internal Quality Criteria Measures

Internal Criteria generally measure the compactness of the clusters by applying similarity measure techniques. In general, it measures the inter-cluster separability and intra-cluster homogeneity, or a combination of these two.

5.1.1 Sum of Squared Error

Sum of Square Error (SSE) [158, 159] is the most frequently used criterion measure for clustering. It is defined as:

$$SSE = \sum_{k=1}^{K} \sum_{\forall x_i \in C_k} \left\| x_i - \mu_k \right\|^2 \tag{19}$$

where C_k is the set of instances in cluster k; μ_k is the vector mean of cluster k.

5.1.2 Scatter Criteria

The scatter criteria matrix [1, 22] is defined as follows for the k-th cluster:

$$S_k = \sum_{x \in C_k} (x - \mu_k)(x - \mu_k)^T \tag{20}$$

5.1.3 Condorcet's Criterion.

The Condorcet's criterion [110] is another approach to apply for the ranking problem [111]. The criterion is defined as follows:

$$\sum_{C_i \in C} \sum_{\substack{x_j, x_k \in C_i \\ x_j \neq x_k}} s(x_j, x_k) + \sum_{C_i \in C} \sum_{x_j \in C_i; x_k \notin C_i} d(x_j, x_k) \tag{21}$$

where $s(x_j , x_k)$ and $d(x_j , x_k)$ measure the similarity and distance of the vectors x_j and x_k.

5.1.4 The C-criterion

Fortier and Solomon [108] defined the C-criterion, which is an extension of Condorcet's criterion and it is defined as: $$\sum_{C_i \in C} \sum_{\substack{x_j, x_k \in C_i \\ x_j \neq x_k}} (s(x_j, x_k) - \gamma) + \sum_{C_i \in C} \sum_{x_j \in C_i; x_k \notin C_i} (\gamma - s(x_j, x_k)) \quad (22)$$

where γ is a threshold value.

5.1.5 Category Utility Metric

The category utility defined in [109, 112] which measures the goodness of category. A set of entities with size n binary feature set $F = \{f_i\}$, $i=1, \ldots, n$ and a binary category $C = \{c, \bar{c}\}$ is calculated as follows:

$$CU(C,F) = \left[p(c) \sum_{i=1}^{n} p(f_i \mid c) \log p(f_i \mid c) + p(\bar{c}) \sum_{i=1}^{n} p(f_i \mid \bar{c}) \log p(f_i \mid \bar{c}) \right] - \sum_{i=1}^{n} p(f_i) \log p(f_i) \quad (23)$$

where $p(c)$ is the prior probability of an entity belonging to the positive category c, $p(f_i \mid c)$ is the conditional probability of an entity having feature f_i given that the entity belongs to category c, $p(f_i \mid c)$ is likewise the conditional probability of an entity having feature f_i given that the entity belongs to category $\bar{c}$, and $p(f_i)$ is the prior probability of an entity processing feature f_i.

5.1.6 Edge Cut Metrics

An edge cut minimization problem [125, 126] is very useful in some cases for dealing with clustering problems. In this case, the cluster quality is measured as the ratio of the remaining edge weights to the total precut edge weights. Finding the optimal value is easy with edge cut minimization problem, where there is no restriction on the size of the clusters.

5.2 External Quality Criteria Measures

In order to match the structure of cluster to a predefined classification of the instances, the external quality criteria measure can be useful.

5.2.1 Mutual Information Based Measure

Strehl *et al* [113] proposed mutual information based measure, which can be used as an external measure for clustering. The criteria measure for m instances clustered using $C = \{C_1, \ldots, Cg\}$ and referring to the target attribute z whose domain is $\text{dom}(z) = \{c_1, \ldots, c_k\}$ is defined as follows:

$$C = \frac{2}{m} \sum_{l=1}^{g} \sum_{h=1}^{k} m_{l,h} \log_{g.k} \left(\frac{m_{l,h} \cdot m}{m_{.,l} \cdot m_{l,.}} \right) \quad (24)$$

where $m_{l,h}$ indicates the number of instances that are in cluster C_l and also in class c_h. $m_{.,h}$ denotes the total number of instances in the class c_h. Similarly, $m_{l,.}$ Indicates the number of instances in cluster C_l.

5.2.2 Rand Index

The Rand index [115] is a simple criterion used to compute how similar the clusters are to the benchmark classifications. The Rand index is defined as;

$$RAND = \frac{TP + TN}{TP + FP + FN + TN} \quad (25)$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives and FN is the number of false negatives. The Rand index lies between 0 and 1. When the two partitions agree perfectly, the Rand index is 1.

5.2.3 F-measure

In Rand index, the false positives and false negatives are equally weighted and this may cause for an undesirable features for some clustering applications. The F-measure [116] addresses this concern and used to balance of false negatives by weighting recall parameter $\eta \geq 0$. The F-measure is defined as follows:

$$F = \frac{(\eta^2 + 1) \cdot P \cdot R}{\eta^2 \cdot P + R} \tag{26}$$

where *P* is the precision rate and *R* is the recall rate. Recall has no impact when $\eta = 0$ and increasing η allocates an increasing amount of weight to recall in the final F-measure. Precision and Recall [119, 120] is defined as follows:

$$P = \frac{TP}{TP + FP} \tag{27}$$

$$R = \frac{TP}{TP + FN} \tag{28}$$

5.2.4 Jaccard Index

The Jaccard index [121] is considered to identify the equivalency between two datasets. The Jaccard index is defined as follows:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{TP}{TP + FP + FN} \tag{29}$$

If *A* and *B* are both empty, then $J(A,B) = 1$, i.e $0 \leq J(A,B) \leq 1$. This is simply the number of unique elements common to both sets divided by the total number of unique elements in both sets.

5.2.5 Fowlkes–Mallows Index

The Fowlkes-Mallows index [118] determines the similarity between the clusters obtained after the clustering algorithm. The higher value of the Fowlkes-Mallows index indicates a more similarity between the clusters. It can be determined as follows:

$$FM = \sqrt{\frac{TP}{TP + FP} \cdot \frac{TP}{TP + FN}} \tag{30}$$

5.2.6 Confusion Matrix

A confusion matrix is also known as a contingency table or an error matrix [114]. It can be used to quickly visualize the results of a clustering. If a classification system has been trained to distinguish between apples, oranges and tomatoes, a confusion matrix will summarize the results of testing the algorithm for further inspection. Assuming a sample of 27 fruits; 8 apples, 6 oranges, and 13 tomatoes, the result of confusion matrix look like the table below:

Table 3 Confusion Matrix

	Predicted class		
Actual class	Apple	Orange	Tomato
Apple	5	3	0
Orange	2	3	1
Tomato	0	2	11

External indices are based on some pre-specified structure, which is the reflection of prior information on the data, and used as a standard to validate the clustering solutions [50]. Internal tests are not dependent on external information (prior knowledge). On the contrary, they examine the clustering structure directly from the original data. For more on evaluation, refer to [193,194].

6 Applications

Clustering is useful in several applications. Out of endless useful applications, a few applications are given below in diverse fields.

6.1 Image Segmentation

Image segmentation is an essential component of image processing. Image segmentation can be achieved using hierarchical clustering [37, 83]. K-means can also be applied for segmentation. Magnetic resonance imaging (MRI) provides a visualization of the internal structures of objects and living organisms. MRI images have better contrast than computerized tomography; therefore, most medical image segmentation research uses MRI images. Segmenting an MRI image is a key task in many medical applications, such as surgical planning and abnormality detection. MRI segmentation aims to partition an input image into significant anatomical areas, each of which is uniform according to certain image properties. MRI segmentation can be formulated as a clustering problem in which a set of feature vectors obtained through transformation image measurements and pixel positions is grouped into a number of structures [28].

6.2 Bioinformatics—Gene Expression Data

Recently, advances in genome sequencing projects and DNA microarray technologies have been achieved [50]. The first draft of the human genome sequence project was completed in 2001, several years earlier than expected [84, 94]. The applications of clustering algorithms in bioinformatics can be seen from two aspects. The first aspect is based on the analysis of gene expression data generated from DNA microarray technologies. The second aspect describes clustering processes that directly work on linear DNA or protein sequences. The assumption is that functionally similar genes or proteins usually share similar patterns or primary sequence structures [50].

6.3 Object Recognition

The use of clustering to group views of 3D objects for the purposes of object recognition in range data was described in [85]. The system under consideration employed a view point dependent (or view cantered) approach to the object recognition problem; each object to be recognized was represented in terms of a library of range images of that object.

6.4 Character Recognition

Clustering was employed in Jain [86] to identify lexemes in handwritten text for the purposes of writer independent hand writing recognition. The success of a handwriting recognition system is vitally dependent on its acceptance by potential users. Writer dependent systems can give a higher level of recognition accuracy than that given by writer independent systems but the former require a large amount of training data. A writer independent system on the other hand must be able to recognize a wide variety of writing styles in order to satisfy an individual user.

6.5 Information Retrieval

Information retrieval (IR) is concerned with automatic storage and retrieval of documents [87]. Many university libraries use IR systems to provide access to books, journals and other documents. Libraries use the library of congress classification (LCC) scheme for efficient storage and retrieval of books. The LCC scheme consists of classes labelled A to Z [88] which are used to characterize books belonging to different subjects. For example, label Q corresponds to books in the area of science and the subclass QA is assigned to mathematics. Labels QA76 to QA76.8 are used for classifying books related to computers and other areas of computer science.

6.6 Data Mining

Data mining [21] is the extraction of knowledge from large databases. It can be applied to relational, transaction and spatial databases as well as large stores of unstructured data such as the World Wide Web. There are many data mining systems in use today and applications include the U.S. Treasury detecting money laundering. National basketball association coaches detecting trends and patterns of play for individual players and teams and categorizing patterns of children in the foster care system [89]. Several articles have had recent published in special issues on data mining [90].

6.7 Spatial Data Analysis

Clustering is useful to extract interesting features and identify the patterns, which exist in huge amounts of spatial databases [106, 127-129]. It is expensive and very hard for user to deal with large spatial datasets like satellite images, medical equipment, geographical information systems (GIS), image database exploration etc. Clustering process helps to understand spatial data by analyzing process automatically.

6.8 Business

The role of clustering is quite interesting in business areas [135-139]. It helps marketer researchers to do some analysis and prediction about customers in order to provide services based on their requirements and it also helps for market segmentation, new product development and product positioning. Clustering may be used to set all available shopping items on web into a group of unique products.

6.9 Data Reduction

Data reduction or compression is one of the necessary tasks for handling very large data [132-134] and its processing becomes very demanding. Clustering can be applied to help in compressing data information by clustering them in different set of interesting clusters. After different set of clusters we can choose the information or set of data which is useful for us. This process will save data processing time along with doing data reduction.

6.10 Big Data Mining

Big data [161-168] is also an emerging issue. The volume of data which is beyond the capacity of conventional data base management tools is processed under big data mining. Due to use of various social sites, travel, e-governance etc practices, mammoth amount of data is being heaped every moment. Clustering of information (data) can help in aggregating similar information collected in unformatted databases (mainly text). Hadoop is one such big data processing tool [169-171]. It is expected that big data processing will play an important role in detection of cyber crime, clustering groups of people with similar behaviour on social network such as face book, WhatsApp etc. or predicting market behaviour based on various polls over these social sites.

6.11 Other Applications

Sequence analysis [140], human genetic clustering [141], social network analysis [142], search result grouping [143], software evolution [144, 145], recommender systems [146], educational data mining [147-149], Climatology [150], Field Robotics [151] etc.

7 Choice of Appropriate Clustering Methods

As depicted in Fig.1, and from the wide amount of literature available with some referred in the paper, it becomes an obvious question: which method is uniformly good? It is to remember that according to *No Free Lunch* concept given by Wolpert [197], no algorithm can be uniformly good under all circumstances. In fact, each algorithm has its merit (strength) under some specific nature of data but fails on other type of data. The selection of an appropriate clustering method may sometimes also involve decision on certain parameters. Whether one wants only a proper alignment (or unsupervised grouping) of objects into a number of clusters (say user define *k*), then only choosing the value of *k* matters. This choice can be made on the ‘how fine tuning among the intra-cluster objects (or patterns) by virtue of distance is expected’. Selecting *k* can be heuristic or stochastic and evolutionary

computing like genetic algorithms (GA) can be applied to find *k*. On the other hand, in case of data mining or data processing applications with dimensionality reduction, mostly it is required to reduce the number of attributes or features in the existing dataset in order to extract rules with better prediction capability. In many of these occasions, it is expected that while reducing the dimensionality of the dataset, whether the structure or the internal topology of the dataset is not disturbed in the reduced data space. Saxena et. al [23] proposed four unsupervised methods for feature selection using genetic algorithms.

In [27], Fraley presents a comprehensive discussion on how to decide a clustering method and described a clustering methodology based on multivariate normal mixture models and shown that it can give much better performance than existing methods. This approach has some limitations, however. The first limitation is that computational methods for hierarchical clustering have storage and time requirements that grow at a faster than linear rate relative to the size of the initial partition, so that they cannot be directly applied to large data sets. Secondly, although experience to date suggests that models based on multivariate normal distribution are sufficiently flexible to accommodate many practical situations, the underlying assumption is that groups are concentrated locally about linear subspaces, so that other models or methods may be more suitable in some instances. Bensmail et al. [198] showed that exact Bayesian inference via Gibbs sampling, with calculations of Bayes factors using the Laplace–Metropolis estimator, works well in several real and simulated examples [27].

Further, for large data sets, CURE method is advisable whereas BIRCH being also good but with less time complexity although quality of clustering is inferior to that obtained by CURE, refer to Table 1. Under partitioned clustering method, k-means clustering dominates and is still the most popular clustering method, refer to Table 2. How many clusters i.e. k depends on how close or fine tuning we want among clusters. We should also keep in mind, for what purpose we are applying k-means. In various clustering methods presented in the paper already, the strengths and weaknesses of each are mostly given therein. Apart from the discussion above on selection of appropriate method for clustering, it is worth noting looking to a huge amount of literature available with wide variety of application of clustering; it is not possible to settle to an agreeable recommendation. Specific task (objectives) calls for specific strategy and should be tested experimentally. Finally, a part of comprehensive and comparative table for various clustering algorithms presented before is given in Table 4, for details and meaning of symbols refer to [199].

Table 4. Comparative study of some clustering algorithms [199]

Category of Clustering	**Algorithm Name**	**Time complexity**	**Scalability**	**Suitable for large scale data**	**Suitable for high dimensional data**	**Sensitive of noise/ outlier**
Partition	k-means	Low O(knt)	Middle	Yes	No	High
	PAM	High O(k(n-k)ˆ2))	Low	No	No	little
	CLARA	Middle O(ksˆ 2+k(n-k))	High	Yes	No	Little
	CLARANS	High O(nˆ2)	Middle	Yes	No	Little
Hierarchy	BIRCH	Low O(n)	High	Yes	No	Little
	CURE	Low O(s ˆ2*logs)	High	Yes	Yes	Little
	ROCK	High O(nˆ2*logn)	Middle	No	Yes	Little
	Chameleon	High O(nˆ2)	High	No	No	Little
Fuzzy based	FCM	Low O(n)	Middle	No	No	High
Density based	DBSCAN	Middle O(n*logn)	Middle	Yes	No	Little
Graph theory	CLICK	Low O(k*f(v,e))	High	Yes	No	High
Grid based	CLIQUE	Low O(n+kˆ2)	High	No	Yes	Moderate

8 Conclusions

The classification of objects finds prime importance in several data processing applications including data mining, medical diagnostics, pattern recognition and social paradigms. The objects already labeled are placed in supervised classified groups while those not labeled are grouped in unsupervised classified groups. This paper presented various methods used for clusters with their states of arts and limitations. In the hierarchical type of clustering methods, clusters are formed by iteratively dividing the patterns (instances) into top-down or bottom up manner accordingly agglomerative and divisive or splitting hierarchical clustering methods are discussed. As opposed to hierarchical clustering, partitional clustering assigns data into *K* clusters without any hierarchical structure by optimizing some criterion function. The most common criterion is finding Euclidean distance between the points with each of the available clusters and assigning the point to the cluster with minimum distance. The benchmark k-means clustering methods with its variations like Fuzzy K-means are discussed. The graph theoretic methods produce clusters via graphs. In the mixture density based methods, data objects are assumed to be generated according to several probability distributions and can be derived from different types of density functions (e.g., multivariate Gaussian or *t*-distribution), or from the same families but with different parameters. The grid based clustering techniques include: STING (statistical information grid approach) a highly scalable algorithm and has the ability to decompose the data set into various levels of details. The evolutionary approaches for clustering start with a random population of candidate solutions with some fitness function, which would be optimized. Clustering based on simulated annealing, collaborative clustering, multi objective clustering with their states of art are also included. Various types of the similarity criteria for clustering have been given in the paper. After the clusters have been formed, the evaluation criteria are also summarised to see the performance and accuracy of clusters. The applications of clustering in image segmentation, object and character recognition, information retrieval and data mining are highlighted in the paper. Of course there is an abundant amount of literature available in clustering and its applications; it is not possible to cover that entirely, only basic and few important methods are included in this paper with their merits and demerits.

Acknowledgement

The authors would like to thank the anonymous reviewers for their valuable suggestions and comments to improve the quality of the paper. This work is partially supported by the Australian Research Council (ARC) under discovery grant DP150101645.

References

1. R. O. Duda, P. E. Hart, and D. G. Stork, “Pattern Classification,” Wiley Publications, 2001.
2. Y. Zhang, Y. Yin, D. Guo, X. Yu, and L. Xiao, “Cross-validation based weights and structure determination of Chebyshev-polynomial neural networks for pattern classification,” Pattern Recognition, vol. 47, no. 10, pp. 3414-3428, 2014.
3. H. Nakayama, N. Kagaku, “Pattern classification by linear goal programming and its extensions,” Journal of Global Optimization, vol. 12, no. 2, pp. 111–126, 1998.
4. C. M. Bishop, “Pattern recognition and machine learning,” Berlin: Springer, ISBN 978-0-387-31073-2.
5. G.P. Zhang, “Neural networks for classification: a survey,” IEEE Transaction on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 30, no. 4, pp. 451–462, 2002.
6. H. Zhang, J. Liu, D. Ma, and Z. Wang, “Data-core-based fuzzy min–max neural network for pattern classification,” IEEE Transaction on Neural Networks, vol. 22, no. 12, pp. 2339–2352, 2011.
7. X. Jiang and A. H. K. S. Wah, “Constructing and training feed-forward neural net- works for pattern classification,” Pattern Recognition, vol. 36, no. 4, pp. 853–867, 2003.
8. G. Ou and Y. L. Murphey, “Multi-class pattern classification using neural networks,” Pattern Recognition, vol. 40, no. 1, pp. 4–18. 2007.
9. J. D. Paola and R. A. Schowengerdt, “A detailed comparison of back propagation neural network and maximum-likelihood classifiers for urban land use classification,” IEEE Transaction on Geoscience and Remote Sensing, vol. 33, no. 4, pp. 981–996, 1995.
10. D. E. Rumelhart and J. L. McClelland, “Parallel Distributed Processing,” MIT Press, Cambridge, 1986.
11. W. Zhou, “Verification of the nonparametric characteristics of back-propagation neural networks for image classification,” IEEE Transaction on Geoscience and Remote Sensing, vol. 37, no. 2, pp. 771–779, 1999.
12. G. Jaeger, U. C. Benz, “Supervised fuzzy classification of SAR data using multiple sources,” IEEE International Geoscience and Remote Sensing Symposium, 1999.

13. F. S. Marzano, D. Scaranari, and G. Vulpiani, "Supervised Fuzzy-Logic Classification of Hydrometeors Using C-Band Weather Radars," IEEE Transaction on Geoscience and Remote Sensing, vol. 45 , no. 11, pp. 3784-3799, 2007.
14. B. Xue, M. Zhang, and W. N. Browne, "Particle Swarm Optimization for Feature Selection in Classification: A Multi-Objective Approach," IEEE Transaction on Cybernetics, vol. 43, no. 6, pp. 1656-1671, 2013.
15. A. Saxena and M. Vora, "Novel Approach for the use of Small World Theory in Particle Swarm Optimization," 16th International Conference on Advanced Computing and Communications, 2008.
16. Z. Pawlak, "Rough sets", International Journal of Computer and Information Science, vol. 11, no. 5, pp. 341-356. 1982.
17. Z. Pawlak, "Rough sets In Theoretical Aspects of Reasoning about Data," Kluwer, Netherlands, 1991.
18. S. Dalai, B. Chatterjee, D. Dey, S. Chakravorti, and K. Bhattacharya, "Rough-Set-Based Feature Selection and Classification for Power Quality Sensing Device Employing Correlation Techniques," IEEE Sensors Journal, vol. 13, no. 2, pp. 563–573, 2013
19. J. R. Quinlan, "Induction of decision trees," Machine Learning, vol. 1, no. 1, pp. 81-106, 1986.
20. D. M. Farida, L Zhang, C. M. Rahman, M. A. Hossain, and R. Strachan, "Hybrid decision tree and naïve Bayes classifiers for multi-class classification tasks," Expert Systems with Applications, vol. 41, no. 2, pp. 1937–1946, 2014.
21. J. Han, M. Kamber, and J. Pei, "Data Mining: Concepts and Techniques," Morgan Kaufmann Publishers, 2011.
22. L. Rokach, "Clustering Methods," Data Mining and Knowledge Discovery Handbook, pp 331-352, Springer 2005.
23. A. Saxena, N. R. Pal, and M. Vora, "Evolutionary methods for unsupervised feature selection using Sammon's stress function, Fuzzy Information and Engineering," vol. 2, no. 3, pp. 229-247, 2010.
24. A. K. Jain, "Data Clustering: 50 years beyond k-means," Pattern Recognition Letters, vol. 31, no. 8, pp. 651–666, 2010.
25. Merriam-Webster Online Dictionary, 2008
26. V. E. Castro and J. Yang, "A Fast and robust general purpose clustering algorithm," International Conference on Artificial Intelligence, 2000.
27. C. Fraley and A. E. Raftery, "How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis", Technical Report No. 329, Department of Statistics University of Washington, 1998.
28. A. K. Jain, M. N. Murty, and P. J. Flynn, "Data Clustering: A review. ACM Computing Surveys, vol. 31, no. 3, pp. 264-323, 1999.
29. P. Sneath and R. Sokal, "Numerical Taxonomy," W.H. Freeman Co, San Francisco, CA, 1973.
30. B. King, "Step-wise Clustering Procedures," Journal of American Statistical Association , vol. 69, no. 317, pp. 86-101, 1967.
31. J. H. Ward, "Hierarchical grouping to optimize an objective function," Journal of the American Statistical Association, vol. 58, no. 301, pp. 236-244, 1963.
32. F. Murtagh, "A survey of recent advances in hierarchical clustering algorithms which use cluster centers," Computer Journal, vol. 26, no. 4, pp. 354-359, 1984.
33. A. Nagpal, A. Jatain, and D. Gaur, "Review based on Data Clustering Algorithms," IEEE Conference on Information and Communication Technologies, 2013.
34. A. Periklis, "Data Clustering Techniques," University of Toronto, 2002.
35. S. Guha, R. Rastogi, and S. Kyuseok, "CURE: An efficient clustering algorithm for large databases," ACM, 1998.
36. K. George, E. H. Han, and V. Kumar, "CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling," IEEE Computer, vol. 32, no. 8, pp. 68-75, 1999.
37. D. Lam and D. C. Wunsch, "Clustering," Academic Press Library in Signal Processing," Signal Processing Theory and Machine Learning, vol. 1, 2014
38. J. B. MacQueen, "Some Methods for classification and Analysis of Multivariate Observations," 5th Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, vol. 1, pp. 281-297, 1967.
39. A. Gersho and R. Gray, "Vector Quantization and Signal Compression," Kluwer Academic Publishers, 1992.
40. J. C. Dunn, "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters," Journal of Cybernetics, vol. 3, no. 3, pp. 32-57, 1973.
41. J. C. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algorithms," Plenum Press, New York, 1981.
42. R. Yager and D. Filev, "Approximate clustering via the mountain method," IEEE Transaction on Systems, Man and Cybernetics, Part B: Cybernetics, vol. 24, no. 8, pp. 1279–1284. 1994
43. I. Gath and A. Geva, "Unsupervised optimal fuzzy clustering," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 11, no. 7, pp. 773–781. 1989.
44. R. Hathaway, J. Bezdek, and Y. Hu, "Generalized fuzzy c-Means clustering strategies using Lp norm distances," IEEE Transaction on Fuzzy Systems, vol. 8, no. 5, pp. 576–582. 2000.
45. R. Krishnapuram and J. Keller, "A possibilistic approach to clustering," IEEE Transaction on Fuzzy Systems, vol. 1, no. 2, pp. 98–110, 1993.
46. C. T. Zahn, "Graph-theoretical methods for detecting and describing gestalt clusters," IEEE Transaction on Computer, vol. C-20, no. 1, pp. 68-86, 1971.
47. R. Urquhart, "Graph-theoretical clustering based on limited neighborhood sets," Pattern Recognition, vol. 15, no. 3, pp. 173-187, 1982.

48. D. H. Fisher, “Knowledge acquisition via incremental conceptual clustering,” Machine Learning 2, pp. 139-172. 1987.
49. S. Haykin, “Neural Networks: A Comprehensive Foundation,” 2nd Edition, Prentice Hall, 1999.
50. R. Xu and D. Wunsch, “Survey of clustering algorithms,” IEEE Transaction on Neural Networks, vol. 16, no. 3, 645–678, 2005.
51. R. Xu. D.C. Wunsch, “Clustering algorithms in biomedical research: a review,” IEEE Reviews in Biomedical Engineering, vol. 3, pp. 120–154. 2010.
52. G. McLachlan, T. Krishnan, “The EM Algorithm and Extensions,” Wiley, New York, 1997.
53. J. D. Banfield and A. E. Raftery, “Model-based Gaussian and non-Gaussian clustering Biometrics,” vol. 49, no. 3, pp. 803-821, 1993.
54. M. Ester, H. P. Kriegel, S. Sander S, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” 2nd International Conference on Knowledge Discovery and Data Mining, 1996.
55. P. Cheeseman, J. Stutz, “Bayesian Classification (AutoClass): Theory and Results,” Advances in Knowledge Discovery and Data Mining, pp. 153-180, 1996.
56. C. S. Wallace and D. L. Dowe, “Intrinsic classification by mml–the snob program,” 7th Australian Joint Conference on Artificial Intelligence, pp. 37-44, 1994.
57. W. Wang, J. Yang, and R. R. Muntz, “STING: A Statistical Information Grid Approach to Spatial Data Mining,” 23rd VLDB Conference, pp. 86-195, 1997.
58. G. Sheikholeslami, S. Chatterjee and A. Zhang, “WaveCluster: a wavelet-based clustering approach for spatial data in very large databases,” The International Journal on Very Large Data Bases, vol. 8, no. 3-4, pp. 289-304, 2000.
59. R. Agrawal, G. Johannes, G. Dimitrios, and P. Raghavan, “Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications,” SIGMOD Conference, pp. 94-105, 1998.
60. A. K. Jain and M. Flynn, “Data clustering: a review,” ACM Computing Surveys (CSUR), vol. 31, no. 3, pp. 264-323, 1999.
61. H. P. Schwefel, “Numerical Optimization of Computer Models,” John Wiley, New York, 1981.
62. L. J. Fogel, A. J. Owens, and M J Walsh, “Artificial Intelligence Through Simulated Evolution,” John Wiley , New York, 1965.
63. J. H. Holland, “Adaption in Natural and Artificial Systems,” University of Michigan Press, 1975.
64. D. Goldberg, “Genetic Algorithms in Search Optimization and Machine Learning,” Addison Wesley Reading, 1989.
65. J. Kennedy and R. C. Eberhart, “Swarm Intelligence,” Morgan Kaufmann, 2001.
66. J. Kennedy and R. Eberhart, “Particle Swarm Optimization,” 4th IEEE International Conference on Neural Networks. pp. 1942–1948, 1995.
67. M. Dorigoand T. Stützle, “Ant Colony Optimization,” MIT Press, 2004.
68. F. Glover, “Future Paths for Integer Programming and Links to Artificial Intelligence,” Computers and Operations Research, vol. 5, no. 5, pp. 533-549, 1986.
69. K. S. Al. Sultan, “A Tabu Search Approach to Clustering Problem,” Pattern Recognition, vol. 28, no. 9, pp. 1443-1451, 1995.
70. W. Pedrycz, “Collaborative fuzzy clustering,” Pattern Recognition Letters, vol. 23, no. 14, pp. 1675–1686, 2002.
71. L. F. S. Coletta, L. Vendramin, E. R. Hruschka, R. J. G. B. Campello, and W. Pedrycz, “Collaborative Fuzzy Clustering Algorithms: Some Refinements and Design Guidelines,” IEEE Transactions on Fuzzy Systems, vol. 20, no. 3, pp. 444-462, 2012.
72. W. Pedrycz and P. Rai, “Collaborative clustering with the use of Fuzzy C-Means and its quantification,” Fuzzy Sets and Systems, vol. 159, no. 18, pp. 2399–2427, 2008.
73. W. Pedrycz, “Knowledge Based Clustering: From data to information granules,” Wiley Publications, 2005.
74. M. Prasad, C. T. Lin, C. T. Yang, and A. Saxena, “Vertical Collaborative Fuzzy C-Means for Multiple EEG Data Sets,” Springer Intelligent Robotics and Applications Lecture Notes in Computer Science, vol. 8102, pp 246-257, 2013.
75. C. Pizzuti, “Overlapping Community Detection in Complex Networks,” GECCO, pp. 859–866, 2009.
76. S. Gregory, “A Fast Algorithm to Find Overlapping Communities in Networks,” PKDD, pp. 408–423, 2008.
77. Y. Y. Ahn, J. P. Bagrow, and S. Lehmann, “Link communities reveal multi-scale complexity in networks,” Nature, vol. 466, pp. 761–764, 2010.
78. G Forestier, P Gancarski, and C Wemmert, “Collaborative Clustering with back ground knowledge,” Data and Knowledge Engineering, vol. 69, no. 2, pp. 211–228, 2010.
79. J. Handl and J. Knowles, “An evolutionary approach to Multiobjective clustering,” IEEE Transaction on Evolutionary Computation, vol.11, no. 1, pp. 56-76, 2007.
80. A. Konak, D. Coit, and A. Smith, “Multiobjective optimization using genetic algorithms: A tutorial,” Reliability Engineering and System Safety, vol. 91, no. 9, pp. 992-1007, 2006.
81. K. Faceili, A. D. Carvalho, and D. Souto, “Multiobjective Clustering ensemble,” International Conference, on Hybrid Intelligent Systems, 2006.
82. M. K. Law, A. Topchy, and A. K. Jain, “Multiobjective Data Clustering,” IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 424-430, 2004.
83. D. Forsyth and J. Ponce, “Computer vision: a modern approach,” Prentice Hall, 2002.

84. I. H. G. S. Consortium, “Initial sequencing and analysis of the human genome,” Nature, vol. 409, pp. 860–921, 2001.
85. C. Dorai and A. K. Jain, “Shape Spectra Based View Grouping for Free Form Object,” International Conference on Image Processing, vol. 3, pp. 240-243, 1995.
86. S. Connell and A. K. Jain, “Learning Prototypes for On-Line Handwritten Digits,” 14th International Conference on Pattern Recognition, vol. 1, pp. 182-184, 1998.
87. E. Rasmussen, “Clustering Algorithms,” Information Retrieval: Data Structures and Algorithms, Prentice Hall Englewood Cliffs, pp 419-442, 1992.
88. G. McKiernan, “LC Classification Outline,” Library of Congress Washington, D. C, 1990.
89. S. R. Hedberg, “Searching for the mother lode: Tales of the first data miners,” IEEE Expert: Intelligent Systems an Their Applications, vol. 11, no. 5, pp. 4-7, 1996.
90. J. Cohen, “Communications of the ACM: Data Mining Association for Computing Machinery,” Nov. 1996.
91. A. Saxena, J. Wang, “Dimensionality Reduction with Unsupervised Feature Selection and Applying Non-Euclidean Norms for Classification Accuracy,” International Journal of Data Warehousing and Mining, vol. 6, no. 2, pp 22-40, 2010.
92. K. S. Al. Sultan and M. M. Khan, “Computational experience on four algorithms for the hard clustering problem,” Pattern Recognition Letters, vol. 17, no. 3, pp. 295-308, 1996.
93. R. Michalski, R. E. Stepp, and E. Diday, “Automated construction of classifications: conceptual clustering versus numerical taxonomy,” IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 5, no. 4, pp. 396–409, 1983.
94. J. C. Venter et. al.,“The sequence of the human genome,”Science,vol. 291, pp. 1304–1351, 2001.
95. J. L. Kolodner, “Reconstructive memory: A computer model,” Cognitive Science, vol. 7, no. 4, pp. 281-328, 1983.
96. C. Carpineto and G. Romano, “An order-theoretic approach to conceptual clustering,” 10th International Conference on Machine Learning, pp. 33–40, 1993.
97. L. Talavera and J. Bejar. “Generality-Based Conceptual Clustering with Probabilistic Concepts,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 2, pp. 196-206, 2001.
98. M. Hadzikadic and D. Yun, “Concept formation by incremental conceptual clustering,” 11th International Joint Conference Artificial Intelligence, pp. 831-836, 1989.
99. G. Biswas, J. B. Weinberg, and D. H. Fisher, “Iterate: A conceptual clustering algorithm for data mining,” IEEE Transactions on Systems, Man, and Cybernetics, Part C, vol. 28, no. 2, pp. 219–230, 1998.
100. K. Thompson and P. Langley, “Concept formation in structured domains,” Concept Formation: Knowledge and Experience in Unsupervised Learning, Morgan Kaufmann, 1991.
101. I. Jonyer, D. Cook, and L. Holder, “Graph-based hierarchical conceptual clustering,” Journal of Machine Learning Research, vol. 2, pp. 19-43, 2001.
102. M. Lebowitz, “Experiments with Incremental Concept Formation: UNIMEM,” Machine Learning, vol. 2, no. 2, pp. 103-138, 1987.
103. S. Hanson and M. Bauer, “Conceptual clustering, categorization and polymorphy,” Machine Learning Journal, vol. 3, no. 4, pp. 343-372, 1989.
104. T. Kohonen, “The self-organizing map,” Neurocomputing, vol. 21, no. 1–3, Pages 1–6, 1998.
105. J. Vesanto and E. Alhoniemi, “Clustering of the Self-Organizing Map,” IEEE Transactions on Neural Networks, vol. 11, no. 3, 2000.
106. J. G. Upton and B. Fingelton, “Spatial Data Analysis by Example,” Point Pattern and Quantitative Data, John Wiley & Sons, New York, vol. 1, 1985.
107. A. Strehl, J. Ghosh, and R. Mooney, “Impact of similarity measures on web-page clustering,” Workshop on Artificial Intelligence for Web Search, pp 58–64, 2000.
108. J. J. Fortier, and H. Solomon, “Clustering procedures,” The Multivariate Analysis, pp. 493-506, 1996.
109. M. A. Gluck and J. E. Corter,(1985), “Information, uncertainty, and the utility of categories,” Program of the 7th Annual Conference of the Cognitive Science Society, pp. 283–287, 1985.
110. M. J. A. N. Condorcet, “Essai sur l’Application de l’Analyse `a la Probabilite´ des decisions rendues a la Pluralite´ des Voix,” paris: L’Imprimerie Royale, 1785.
111. J. F. Marcotorchino and P. Michaud, “Optimisation en Analyse Ordinale des Donnees Masson, Paris, 1979.
112. J. E. Corter and M. A. Gluck, “Explaining basic categories: Feature predictability and information,” Psychological Bulletin, vol. 111, no. 2, pp. 291–303, 1992.
113. A. Strehl and J. Ghosh, “Clustering Guidance and Quality Evaluation Using Relationship-based Visualization,” Intelligent Engineering Systems through Artificial Neural Networks, St. Louis, Missouri, USA, pp 483-488, 2000.
114. S. V. Stehman, “Selecting and interpreting measures of thematic classification accuracy” Remote Sensing of Environment, vol. 62, no. 1, pp. 77–89, 1997.
115. W. M. Rand, “Objective criteria for the evaluation of clustering methods,” Journal of the American Statistical Association, vol. 66, no. 336, pp. 846– 850, 1971.
116. V. Rijsbergen, “Information retrieval,” Butterworths, London, 1979.
117. J. F. Brendan and D. Dueck, “Clustering by passing messages between data points,”.Science, vol. 315, pp. 972–976, 2007.
118. E. B. Fowlkes and C. L. Mallows (1983), “A Method for Comparing Two Hierarchical Clusterings,” Journal of the American Statistical Association, vol. 78, no. 383, pp. 553–569, 2010.

119. D. L. Olson and D. Delen, "Advanced Data Mining Techniques," Springer, 1st edition, 2008.
120. D. M. W. Powers, "Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation," Journal of Machine Learning Technologies, vol. 2, no. 1, pp. 37–63, 2007.
121. P. Jaccard, "Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines," Bulletin de la Société Vaudoise des Sciences Naturelles, vol. 37, pp. 241-272, 1901.
122. J. Han, M. Kamber, and J. Pei, "Data mining: Concepts and techniques," Morgan Kaufman, San Francisco, USA, 2011.
123. J. J. Grefenstette, "Optimization of Control Parameters for Genetic Algorithms," IEEE Transaction on Systems, Man and Cybernetics, vol. 16, no. 1, pp. 122–128, 1986.
124. C. T. Lin, M. Prasad, and J. Y. Chang, "Designing mamdani type fuzzy rule using a collaborative FCM scheme," International Conference on Fuzzy Theory and Its Applications, 2013.
125. L. Eugene, "Chapter 4.5. Combinatorial Implications of Max-Flow Min-Cut Theorem, Chapter 4.6. Linear Programming Interpretation of Max-Flow Min-Cut Theorem," Combinatorial Optimization: Networks and Matroids, Dover. pp. 117–120, 2001.
126. C. H. Papadimitriou and K. Steiglitz, "Chapter 6.1 The Max-Flow, Min-Cut Theorem," Combinatorial Optimization: Algorithms and Complexity. Dover. pp. 120– 128, 1998.
127. A. S. Fotheringham, M. E. Charlton, and C. Brunsdon, "Geographically weighted regression: a natural evolution of the expansion method for spatial data analysis," Environment and Planning, vol. 30, no. 11, pp. 1905-1927, 1998.
128. M. Honarkhah, and J. Caers, "Stochastic Simulation of Patterns Using Distance-Based Pattern Modeling," Mathematical Geosciences, vol. 42, no. 5, pp. 487–517, 2010.
129. P. Tahmasebi, A. Hezarkhani, and M Sahimi, "Multiple-point geostatistical modeling based on the cross-correlation functions," Computational Geosciences, vol.16, no. 3, pp. 779-797, 2012.
130. S. Guha, R. Rastogi, and K. Shim, "ROCK: A Robust Clustering Algorithm for Categorical Attributes," IEEE Conference on Data Engineering, 1999.
131. T. Zhang, R. Ramakrishnan, and M. Linvy, "BIRCH: An Efficient Method for Very Large Databases," ACM SIGMOD, 1996.
132. D. Jiang, G. Chen, B. C. Ooi, K. L. Tan, and S. W, "epiC: an Extensible and Scalable System for Processing Big Data," 40th VLDB Conference, pp. 541 - 552, 2014.
133. Z. Huang, "A Fast Clustering Algorithm to Cluster very Large Categorical Data Sets in Data Mining," DMKD, 1997.
134. A. Hinneburg and D. Keim, "An Efficient Approach to Clustering in Large Multimedia Databases with Noise," KDD Conference, 1998.
135. M. J. A. Berry and G. Linoff, "Data Mining Techniques For Marketing, Sales and Customer Support," John Wiley & Sons, Inc., USA, 1996.
136. G. Fennell, G. M. Allenby, S. Yang and Y. Edwards, "The Effectiveness of Demographics and Phychographic Variables for Explaining Brand and Product Category Use," Quantitative Marketing and Economics, vol. 1, no. 2, pp. 223-224, 2003.
137. M. Y. Kiang, D. M. Fisher, M. Y. Hu, "The effect of sample size on the extended self-organizing map network- A market segmentation application," Computational Statistics and Data Analysis, vol. 51, no. 12, pp. 5940-5948, 2007.
138. S. Dolnicar, "Using Cluster Analysis for Market Segmentation–Typical Misconceptions, Established Methodological Weaknesses and Some Recommendations for Improvement," Journal of Marketing Research, vol. 11, no. 2, pp. 5-12, 2003.
139. R. Wagner, S. W. Scholz, and R. Decker, "The number of clusters in market segmentation," Data Analysis and Decision Support, Heidelberg: Springer, pp. 157-176, 2005.
140. R. M. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, "Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids," Cambridge: Cambridge University Press, 1998.
141. J. M. Kaplan,el, R. G. Winther, "Prisoners of Abstraction? The Theory and Measure of Genetic Variation, and the Very Concept of "Race"," Biological Theory, vol. 7. 2012.
142. P. J. Carrington, and J. Scott, "Social Network Analysis: An Introduction," The Sage Handbook of Social Network Analysis, London, vol. 1, 2011.
143. "Yippy growing by leaps, bounds," The News-Press. 23 May 2010, Retrieved 24 May 2010.
144. D. Dirk, "A concept-oriented approach to support software maintenance and reuse activities" 5th Joint Conference on Knowledge Based Software Engineering, 2002.
145. M. G. B. Dias, N. Anquetil, and K. M. D. Oliveira, "Organizing the knowledge used in software maintenance," Journal of Universal Computer Science, vol. 9, no. 7, pp. 641–658, 2003.
146. R. Francesco, L Rokach and B. Shapira, "Introduction to Recommender Systems Handbook," Recommender Systems Handbook, Springer, 2011, pp. 1-35.
147. "www.educationaldatamining.org," 2013.
148. R. Baker, "Data Mining for Education," International Encyclopedia of Education (3rd edition), Oxford, UK, Elsevier, vol. 7, pp. 112-118, 2010.
149. G. Siemens, R. S. J. D. Baker, "Learning analytics and educational data mining: towards communication and collaboration," 2nd International Conference on Learning Analytics and Knowledge, pp. 252–254, 2012.

150. R. Huth, C. Beck, A. Philipp, M. Demuzere, Z. Ustrnul, M. Cahynova, J. Kysely, and O. E. Tveito, "Classifications of Atmospheric Circulation Patterns: Recent Advances and Applications" Annals of the New York Academy Science, vol. 1146, no. 1, pp. 105-152, 2008.
151. A. Bewley. R. Shekhar, S. Leonard, B. Upcroft, and P. Lever, "Real-time volume estimation of a dragline payload," IEEE International Conference on Robotics and Automation", pp. 1571-1576, 2011.
152. C. D. Manning, P. Raghavan, and H. Schu¨tze, "An Introduction to Information Retrieval," Cambridge University, Press, 2009.
153. D. T. Nguyen, L. Chen, and C. K. Chan, "Clustering with Multi-viewpoint-Based Similarity Measure," IEEE Transactions on Knowledge and Data Engineering, vol. 24, no. 6, pp. 988-1001, 2012.
154. Bravais, "Memoires par divers savants," T, IX, Paris, pp. 255–332, 1846.
155. K. Pearson, "Mathematical Contributions to the Theory of Evolution, III, Regression, Heredity, and Panmixia," Philosophical Transactions of the Royal Society of London, Series A, vol. 187, pp. 253–318, 1896.
156. T. Sørensen, "A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons," Kongelige Danske Videnskabernes Selskab, vol. 5, no. 4, pp. 1–34, 1948.
157. L. R. Dice, "Measures of the Amount of Ecologic Association Between Species," Ecology, vol. 26, no. 3, pp. 297–302, 1945.
158. J. D. Hamilton, "Time Series Analysis," Princeton University Press, 1994.
159. R. S. Tsay, "Analysis of Financial Time Series," John Wiley & SONS, 2005.
160. A Saxena and J. Wang, "Dimensionality Reduction with Unsupervised Feature Selection and Applying Non-Euclidean Norms for Classification Accuracy," International Journal of Data Warehousing and Mining (IJDWM), vol. 6, no. 2, pp. 22–40, 2010.
161. S. Arora, I. Chana, "A Survey of Clustering Techniques for Big Data Analysis," 5th International Conference on The Next Generation Information Technology Summit (Confluence), 2014.
162. A. S. Shirkhorshidi, S. Aghabozorgi, T. Y. Wah, and T. Herawan, "Big Data Clustering: A Review," Lecture Notes in Computer Science, vol. 8583, pp. 707-720, 2014.
163. H. Wang, W. Wang, J. Yang, and P. S. Yu, "Clustering by Pattern Similarity in Large Data Sets," International Conference on Management of Data, ACM, 2002.
164. Z. Huang, "A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining," DMKD. 1997.
165. X. Wu, X. Zhu, G. Q. Wu, and W. Ding, "Data mining with big data," IEEE Transaction on Knowledge and Data Engineering, vol. 26, no. 1, pp. 97-107, 2014.
166. P. Russom, "Big Data Analytics," TDWI Best Practices Report, Fourth Quarter, 2011.
167. C. Xiao, F. Nie, and H. Huang, "Multi-view k-means clustering on big data," The Twenty-Third International Joint Conference on Artificial Intelligence, AAAI, 2013.
168. W. Fan and B. Albert, "Mining Big Data: Current Status and Forecast to the Future," ACM SIGKDD Explorations Newsletter, vol. 14, no. 2, pp. 1-5, 2013.
169. K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The hadoop distributed file system," IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST), 2010.
170. D. Jeffrey and S. Ghemawat, "MapReduce: a flexible data processing tool," Communications of the ACM, vol. 53, no. 1, pp. 72-77, 2010.
171. https://hadoop.apache.org/
172. G. Celeux, and G. Govaert, "A classification EM algorithm for clustering and two stochastic versions," Computational statistics & Data analysis, vol. 14, no. 3, pp. 315-332, 1992.
173. L. Kaufman and P. Rousseeuw, "Finding Groups in Data: An Introduction to Cluster Analysis," Wiley, 1990.
174. R. Ngand and J. Han, "CLARANS: A method for clustering objects for spatial data mining," IEEE Trans. Knowledge Data Engineering, vol. 14, no. 5, pp. 1003–1016, 2002.
175. Sisodia, singh, sisodia, and saxena, "Clustering Techniques: A Brief Survey of Different Clustering Algorithms", International Journal of Latest Trends in Engineering and Technology (IJLTET), vol. 1, no. 3, pp. 82-87, 2012.
176. Zhong, Miao, and Wang, "A graph-theoretical clustering method based on two rounds of minimum spanning trees," Pattern Recognition, vol. 43, pp. 752 – 766, 2010.
177. Y. Chen, S. Sanghavi, and H. Xu, "Improved graph clustering," IEEE Transactions on Information Theory, vol. 60, no. 10, pp. 6440-6455, 2014.
178. A. Condon, and R. Karp, "Algorithms for graph partitioning on the planted partition model," Random Structures Algorithms, vol. 18, no. 2, pp. 116-140, 2001.
179. W. E. Donath and A. J. Hoffman, "Lower bounds for the partitioning of graphs," IBM J. Res. Develop., vol. 17, pp. 420 – 425, 1973.
180. J. Shi, J. and J. Malik, "Normalized cuts and image segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 888 – 905, 2000.
181. U. Luxburg, "A tutorial on spectral clustering," Statistics and Computing, vol. 17, no. 4, pp. 395-416, 2007.
182. K. Rohe, S. Chatterjee, and B. Yu, "Spectral clustering and the high-dimensional stochastic block model," The Annals of Statistics, vol. 39, no. 4, pp. 1878-1915, 2011.
183. S. Gunnemann, I. Farber, B. Boden, and T. Seidl, "Subspace clustering meets dense sub-graph mining," A synthesis of two paradigms, In ICDM, 2010.

184. K. Macropol and A. Singh, “Scalable discovery of best clusters on large graphs,” Proceedings of the VLDB Endowment, vol. 3, no. 1-2, pp. 693-702, 2010.
185. J. J. Whang, X. Sui, and I. S. Dhillon, “Scalable and memory-efficient clustering of large-scale social networks,” In ICDM, 2012.
186. G. Karypis and V. Kumar, “A fast and high quality multilevel scheme for partitioning irregular graphs,” SIAM Journal on Scientific Computing, vol. 20, no. 1, pp. 359-392, 1998.
187. G. Karypis and V. Kumar, “Multilevel k-way partitioning scheme for irregular graphs,” Journal of Parallel and Distributed Computing, vol. 48, pp. 96-129, 1998.
188. D. Yan, L. Huang, and M. I. Jordan, “Fast approximate spectral clustering,” In KDD, pp. 907-916, 2009.
189. J. Liu, C. Wang, M. Danilevsky, and J. Han, “Large-scale spectral clustering on graphs,” In IJCAI, 2013.
190. W. Yang and H. Xu, “A divide and conquer framework for distributed graph clustering,” In ICML, 2015.
191. Ghosh and Dubey, “Comparative Analysis of K-Means and Fuzzy C Means Algorithms,” International Journal of Advanced Computer Science and Applications, vol. 4, no.4, pp. 35-39, 2013.
192. S. Niwattanakul, J. Singthongchai, E. Naenudorn and S. Wanapu, “Using of Jaccard Coefficient for Keywords Similarity”, Proceedings of the International MultiConference of Engineers and Computer Scientists 2013 Vol I, IMECS 2013, March 13 - 15, 2013, Hong Kong, 1-5.
193. C. Chen, L. Pau, and P. Wang, “Hand book of Pattern Recognition and Computer Vision , Eds., World Scientific, Singapore, pp. 3 –32. R.Dubes, “Cluster analysis and related issue”.
194. A. Jain and R. Dubes, “Algorithms for Clustering Data,” Englewood, Cliffs, NJ: Prentice-Hall, 1988.
195. C. Shi, Y. Cai, D. Fu, Y. Dong, and B. Wu, “A link clustering based overlapping community detection algorithm,” Data & Knowledge Engineering, vol. 87, pp. 394–404, 2013.
196. G. Palla, I. Derenyi, I. Farkas, and T. Vicsek, “Uncovering the overlapping community structure of complex networks in nature and society,” Nature, vol. 435, pp. 814–818, 2005.
197. D. H. Wolpert and W. G. Macready, “No Free Lunch Theorem for Optimization,” IEEE Transactions on Evolutionary Computation, vol. 1, No. 1, pp. 67-82, 1997
198. Bensmail, H., Celeux, G., Raftery, A. E. and Robert, C. P. (1997) Inference in model-based cluster analysis. Stat.Comput., 7, 1–10.
199. Xu.D., Tian, Y., “A Comprehensive Survey o f Clustering Algorithms”, Ann. Data Sci. 2, 165-193,2015.