Neurocomputing 267 (2017) 664–681

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A review of clustering techniques and developments

Amit Saxena[a], Mukesh Prasad[b,*], Akshansh Gupta[c], Neha Bharill[d], Om Prakash Patel[d], Aruna Tiwari[d], Meng Joo Er[e], Weiping Ding[f], Chin-Teng Lin[b]

[a] Department of Computer Science & IT, Guru Ghasidas Vishwavidyalaya, Bilaspur, India
[b] Centre for Artificial Intelligence, University of Technology Sydney, Sydney, Australia
[c] School of Computational and Integrative Sciences, Jawaharlal Nehru University, New Delhi, India
[d] Department of Computer Science and Engineering, Indian Institute of Technology Indore, India
[e] School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
[f] School of Computer and Technology, Nantong University, Nantong, China

ARTICLE INFO

Article history:
Received 2 March 2016
Revised 9 April 2017
Accepted 24 June 2017
Available online 4 July 2017

Communicated by Deng Cai

Keywords:
Unsupervised learning
Clustering
Data mining
Pattern recognition
Similarity measures

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.

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 labelling 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

* Corresponding author.
E-mail address: mukeshnctu.cs99g@nctu.edu.tw (M. Prasad).

http://dx.doi.org/10.1016/j.neucom.2017.06.053
0925-2312/

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, ..., S_k, such that

$$S_1 \cap S_2 \cap S_3, \ldots, \cap S_k = \phi \quad (1)$$

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 et al. [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 and Yang [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].

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.

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 follows

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

2.1.2. Complete-linkage clustering

In complete-linkage clustering also called the diameter, the maximum method or the furthest neighbour method; the distance

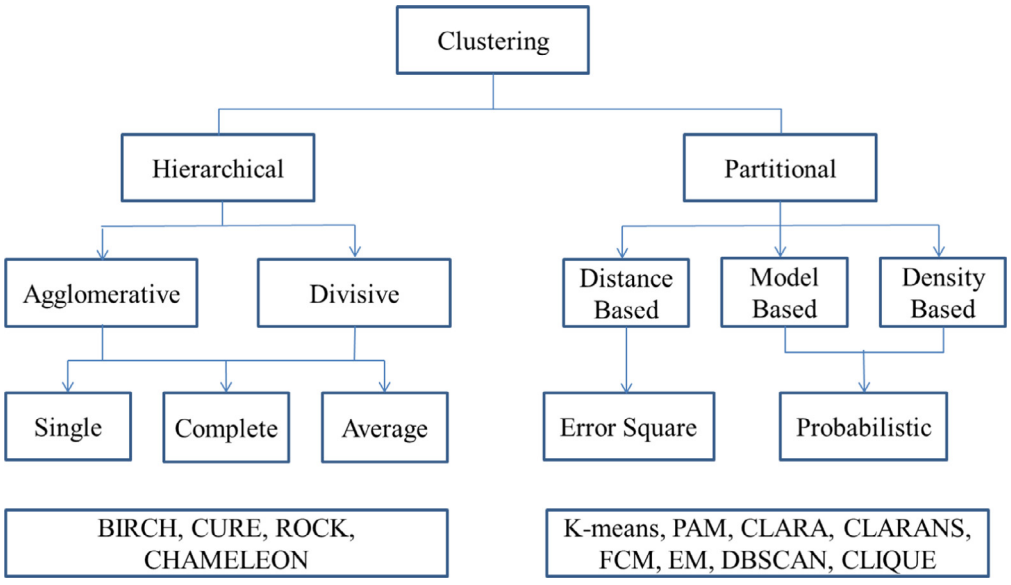


Fig. 1. Taxonomy of clustering approaches [27].

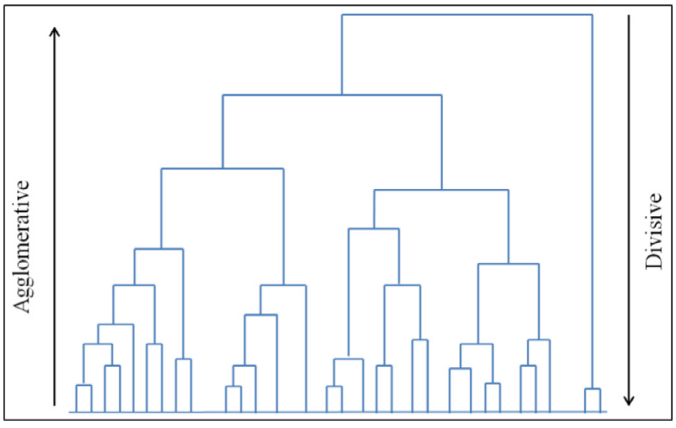


Fig. 2. Hierarchical clustering dendrogram.

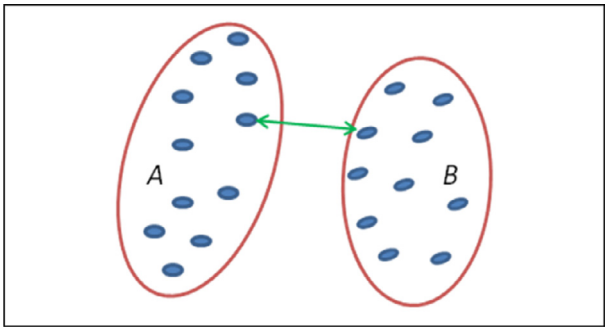


Fig. 3. Mapping of single linkage clustering.

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 follows

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

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

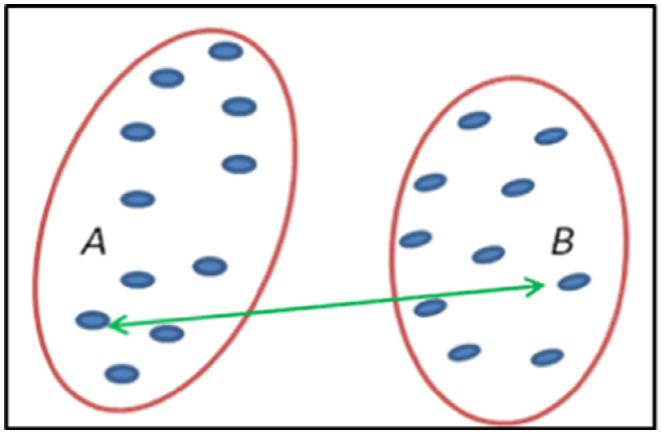


Fig. 4. Mapping of complete linkage clustering.

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

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

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

(i) 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.

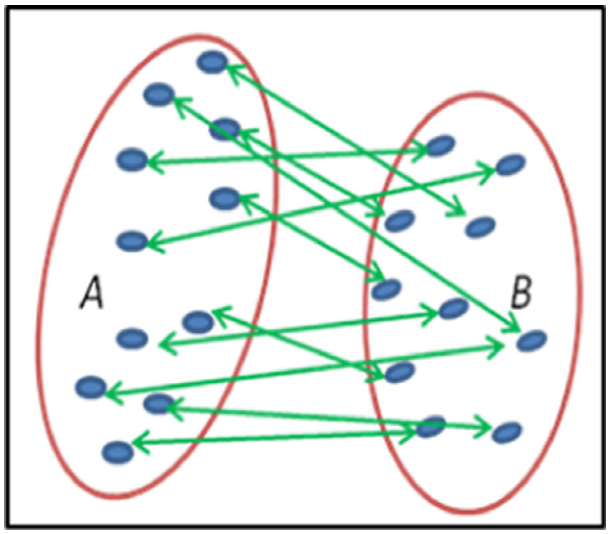


Fig. 5. Mapping of average linkage clustering.

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 computational complexity of *O(N)*.

(i) *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 $O(N^2 \log N)$.

(i) *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.

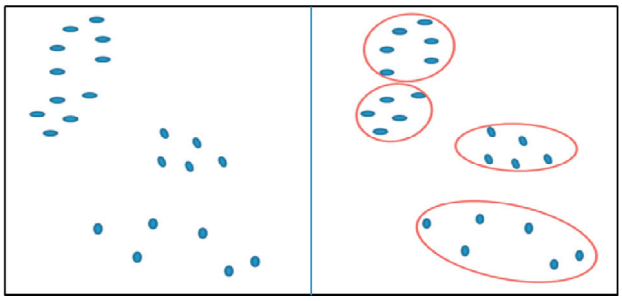


Fig. 6. Partitional clustering approaches.

(i) *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.

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.

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

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

where $\|x_i^{(j)} - c_j\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centrec_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

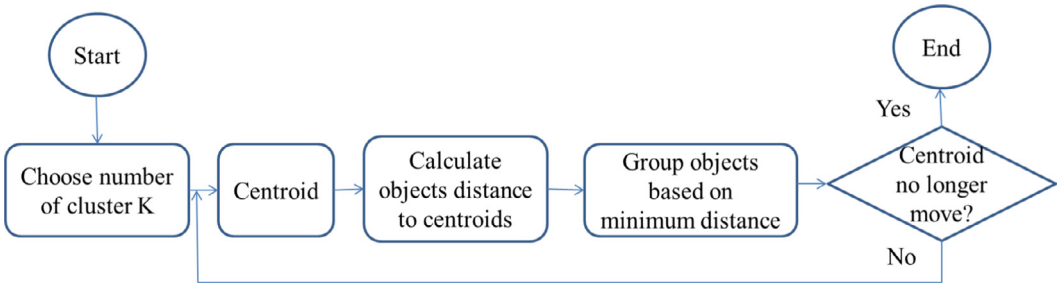


Fig. 7. Flow diagram of *k*-means algorithm.

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].

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

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

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 ith pattern of D-dimension data, v_j is jth cluster centre of the D-dimension and $\|*\|$ is any norm expressing the similarity between any measured data and the centre.

Procedure for FCM

1. *Set up a value of c (number of cluster).*
2. *Select initial cluster prototype* $V_1, V_2, \ldots, V_c$ *from* X_i ,$i = 1, 2, \ldots, N$.
3. *Compute the distance* $\|X_i - V_j\|$ *between objects and prototypes.*
4. *Computer the elements of the fuzzy partition matrix*
$(i = 1, 2, \ldots, N; j = 1, 2, \ldots, c)$
$u_{ij} = [\sum_{l=1}^{c} (\frac{\|x_i - v_j\|}{\|x_i - v_l\|})]^{-1}$.
5. *Compute the cluster prototypes* $(j = 1, 2, \ldots, c)$
$V_j = \frac{\sum_{i=1}^{N} u_{ij}^2 x_i}{\sum_{i=1}^{N} u_{ij}^2}$.
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 centres 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 centres. 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, Krishnapuram and 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

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

$$\exists_j,\ u_{ij} > 0,\ \forall i \quad (8)$$

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

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

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

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

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 kth variable for entity i, x_{jk} is the value of the kth 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 and Wang [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

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

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

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

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

3.4. Extended Jaccard measure

Strehl et al. [107] represented the extended Jaccard measure as follows

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

3.5. Dice coefficient measure

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

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

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 and Karp [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.

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,

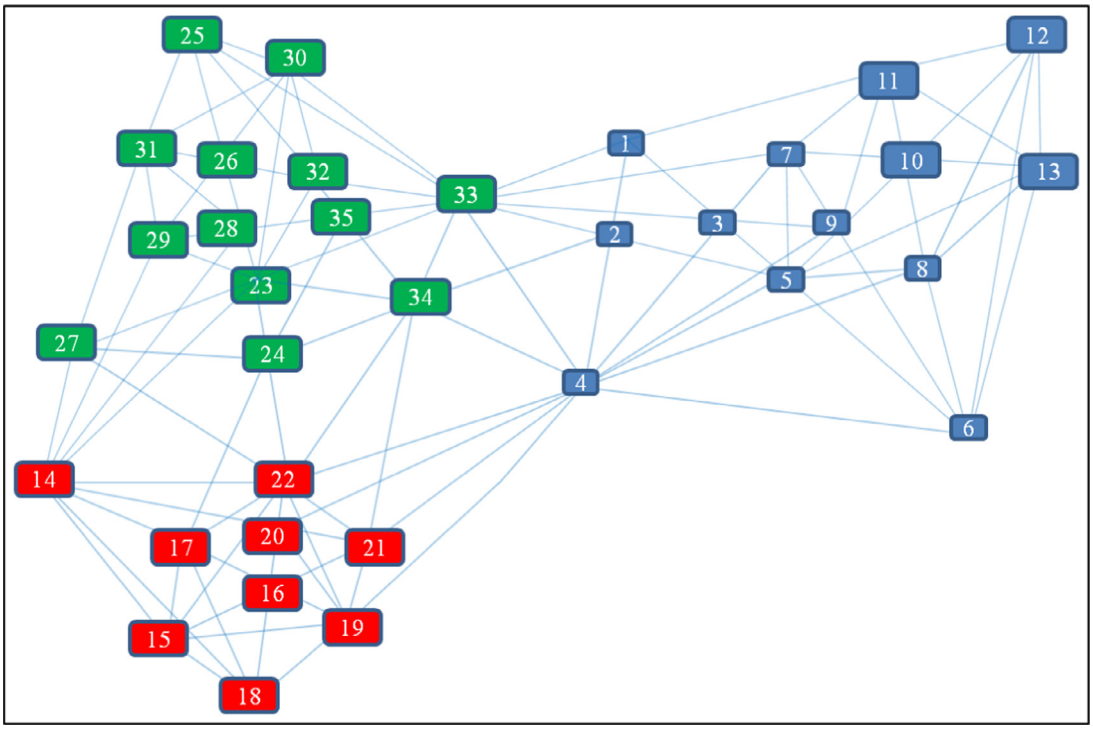


Fig. 8. Sub-network clustering of graph.

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 neighbourhood 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 $S=(s_{ij})I, j=1, \ldots, n$.

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* $i=1,\ldots, n$, *let* $y_i \in R^k$ *be the vector corresponding to the* ith *row of U.*
7. *Cluster the points* $(y_i)i=1,\ldots,n$ *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 [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* $u1,\ldots, uk$ *of the generalized eigen problem* $Lu=\lambda Du$.
5. *Let* $U \in R\ n\times k$ *be the matrix containing the vectors* $u1,\ldots, uk$ *as columns.*
6. *For* $i=1,\ldots, n$, *let* $yi \in R\ k$ *be the vector corresponding to the* ith *row of U.*
7. *Cluster the points* $(yi)i=1,\ldots,n$ *in* $R\ k$ *with the k-means algorithm into clusters* $C1,\ldots, Ck$.
8. *Output: Clusters* $A1,\ldots, Ak$ *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 et al. [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 et al. [99], LABYRINTH by Thompson and Langley [100], SUBDUE by Jonyer et al. [101], UNIMEM by Lebowitz [102] and WITT by Hanson and Bauer [103].

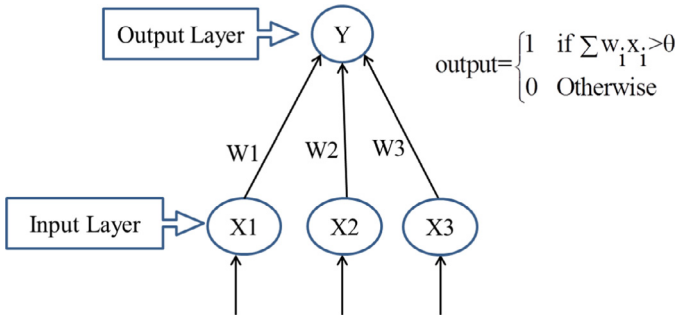


Fig. 9. Model of a single layered network. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

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].

(i) *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].

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

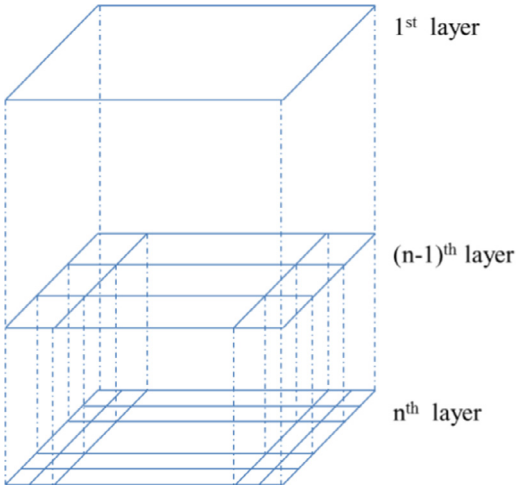


Fig. 10. Rectangular cells corresponding to different levels of resolution.

[52,56]. For EM algorithm, the log likelihood function to maximize is as follows

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

where X denotes the set of all observed data ($X = \{\vec{x_1}, \ldots, \vec{x_N}\}$), and Y denotes the set of all latent variables ($Y = \{\vec{y_1}, \ldots, \vec{y_N}\}$). The complete data set is formed as $(X, Y) = \{(\vec{x_i}, \vec{y_i})\}$ and the joint distribution $p(\vec{x}, \vec{y}|\Theta)$ 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* $p(Y|X, \Theta^{old})$
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 et al. [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.

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 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 ∀ structure ∈ 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 and Khan [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]

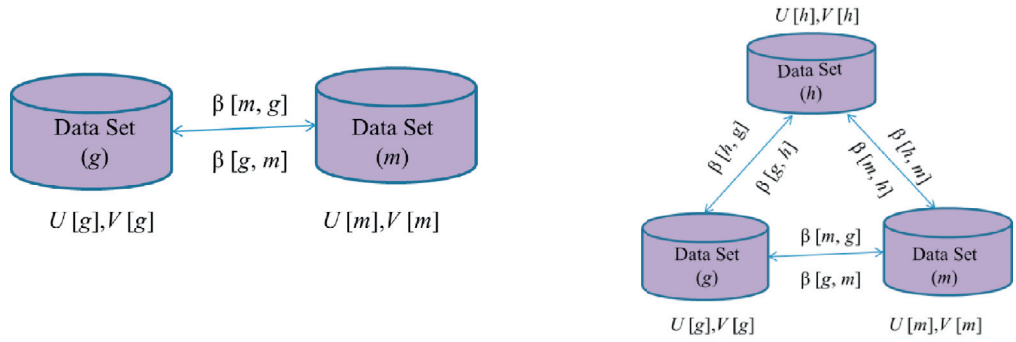
$$Q[l] = \sum_{i=1}^{N}\sum_{j=1}^{c} u_{ij}^2[l]d_{ij}^2[l] + \sum_{\substack{m=1 \\ m\neq l}}^{p} \beta[l,m] \sum_{i=1}^{N}\sum_{j=1}^{n} \{u_{ij}[l] - u_{ij}[m]\}^2 d_{ij}^2[l] \quad (18)$$

where β is a user defined parameter based on datasets ($\beta > 0$), $\beta[l, m]$ denotes the collaborative coefficient with collaborative effect on dataset l through m, c is a number of cluster. $l = 1, 2, \ldots, P$. 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).

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



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

Fig. 11. Collaborative clustering scheme.

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. Palla 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

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

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 kth 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

$$\sum_{C_i \in C} \sum_{\substack{x_j, x_k \in C_i \\ x_j \neq x_k}} (s(x_j, x_k) - \gamma) + \sum_{C_i \in C} \sum_{x_j \in C_i; x_k \notin C_i} (\gamma - s(x_j, x_k)) \tag{22}$$

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

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

where $p(c)$ is the prior probability of an entity belonging to the positive category c, $p(f_i|c)$is the conditional probability of an entity having feature f_i given that the entity belongs to category c, $p(f_i|c)$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 and Ghosh [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 $\mathrm{dom}(z)=\{c_1,\ldots,c_k\}$ is defined as follows

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

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

$$RAND = \frac{TP + TN}{TP + FP + FN + TN} \tag{25}$$

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).

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

Table 1
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.

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.

Table 3
Confusion matrix.

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

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 Connell and 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

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 $O(k(n-k)^2))$	Low	No	No	little
	CLARA	Middle $O(ks^2+k(n-k))$	High	Yes	No	Little
	CLARANS	High $O(n^2)$	Middle	Yes	No	Little
Hierarchy	BIRCH	Low $O(n)$	High	Yes	No	Little
	CURE	Low $O(s^2*logs)$	High	Yes	Yes	Little
	ROCK	High $O(n^2*logn)$	Middle	No	Yes	Little
	Chameleon	High $O(n^2)$	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 $O(n+k^2)$	High	No	Yes	Moderate

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 and Macready [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 and Raftery present 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].

8. Conclusion

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, D.G. Stork, Pattern Classification, Wiley Publications, 2001.
[2] Y. Zhang, Y. Yin, D. Guo, X. Yu, L. Xiao, Cross-validation based weights and structure determination of Chebyshev-polynomial neural networks for pattern classification, Pattern Recognit. 47 (10) (2014) 3414–3428.
[3] H. Nakayama, N. Kagaku, Pattern classification by linear goal programming and its extensions, J. Global Optim. 12 (2) (1998) 111–126.
[4] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, Berlin. ISBN 978-0-387-31073-2.
[5] G.P. Zhang, Neural networks for classification: a survey, IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. 30 (4) (2002) 451–462.
[6] H. Zhang, J. Liu, D. Ma, Z. Wang, Data-core-based fuzzy min–max neural network for pattern classification, IEEE Trans. Neural Netw. 22 (12) (2011) 2339–2352.
[7] X. Jiang, A.H.K.S. Wah, Constructing and training feed-forward neural networks for pattern classification, Pattern Recognit. 36 (4) (2003) 853–867.
[8] G. Ou, Y.L. Murphey, Multi-class pattern classification using neural networks, Pattern Recognit. 40 (1) (2007) 4–18.
[9] J.D. Paola, R.A. Schowengerdt, A detailed comparison of back propagation neural network and maximum-likelihood classifiers for urban land use classification, IEEE Trans. Geosci. Remote Sens. 33 (4) (1995) 981–996.
[10] D.E. Rumelhart, 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 Trans. Geosci. Remote Sens. 37 (2) (1999) 771–779.
[12] G. Jaeger, U.C. Benz, Supervised fuzzy classification of SAR data using multiple sources, IEEE Int. Geosci. Remote Sens. Symp. (1999).
[13] F.S. Marzano, D. Scaranari, G. Vulpiani, Supervised fuzzy-logic classification of hydrometeors using C-band weather radars, IEEE Trans. Geosci. Remote Sens. 45 (11) (2007) 3784–3799.
[14] B. Xue, M. Zhang, W.N. Browne, Particle swarm optimization for feature selection in classification: a multi-objective approach, IEEE Trans. Cybern. 43 (6) (2013) 1656–1671.
[15] A. Saxena, M. Vora, Novel approach for the use of small world theory in particle swarm optimization, in: Proceedings of the Sixteenth International Conference on Advanced Computing and Communications, 2008.
[16] Z. Pawlak, Rough sets, Int. J. Comput. Inf. Sci. 11 (5) (1982) 341–356.
[17] Z. Pawlak, Rough Sets in Theoretical Aspects of Reasoning about Data, Kluwer, Netherlands, 1991.
[18] S. Dalai, B. Chatterjee, D. Dey, S. Chakravorti, K. Bhattacharya, Rough-set-based feature selection and classification for power quality sensing device employing correlation techniques, IEEE Sens. J. 13 (2) (2013) 563–573.
[19] J.R. Quinlan, Induction of decision trees, Mach. Learn. 1 (1) (1986) 81–106.
[20] D.M. Farida, L. Zhang, C.M. Rahman, M.A. Hossain, R. Strachan, Hybrid decision tree and naïve Bayes classifiers for multi-class classification tasks, Expert Syst. Appl. 41 (2) (2014) 1937–1946.
[21] J. Han, M. Kamber, J. Pei, Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, 2011.
[22] L. Rokach, Clustering methods, Data Mining and Knowledge Discovery Handbook, Springer, 2005, pp. 331–352.
[23] A. Saxena, N.R. Pal, M. Vora, Evolutionary methods for unsupervised feature selection using Sammon's stress function, Fuzzy Inf. Eng. 2 (3) (2010) 229–247.
[24] A.K. Jain, Data clustering: 50 years beyond k-means, Pattern Recognit. Lett. 31 (8) (2010) 651–666.
[25] Merriam-Webster Online Dictionary, 2008
[26] V.E. Castro, J. Yang, A fast and robust general purpose clustering algorithm, in: Proceedings of the International Conference on Artificial Intelligence, 2000.
[27] C. Fraley, A.E. Raftery, How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis, Department of Statistics University of Washington, 1998 Technical Report No. 329.
[28] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Comput. Surv. 31 (3) (1999) 264–323.
[29] P. Sneath, R. Sokal, Numerical Taxonomy, W.H. Freeman Co, San Francisco, CA, 1973.
[30] B. King, Step-wise clustering procedures, J. Am. Stat. Assoc. 69 (317) (1967) 86–101.
[31] J.H. Ward, Hierarchical grouping to optimize an objective function, J. Am. Stat. Assoc. 58 (301) (1963) 236–244.
[32] F. Murtagh, A survey of recent advances in hierarchical clustering algorithms which use cluster centers, Comput. J. 26 (4) (1984) 354–359.
[33] A. Nagpal, A. Jatain, D. Gaur, Review based on data clustering algorithms, in: Proceedings of the IEEE Conference on Information and Communication Technologies, 2013.
[34] A. Periklis, Data Clustering Techniques, University of Toronto, 2002.
[35] S. Guha, R. Rastogi, S. Kyuseok, CURE: An Efficient Clustering Algorithm For Large Databases, ACM, 1998.
[36] K. George, E.H. Han, V. Kumar, Chameleon: a hierarchical clustering algorithm using dynamic modeling, IEEE Comput. 32 (8) (1999) 68–75.
[37] D. Lam, D.C. Wunsch, Clustering, academic press library in signal processing, Signal Process. Theory Mach. Learn. 1 (2014) 1115–1149.
[38] J.B. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Symposium on Mathematical Statistics and Probability, vol. 1, Berkeley, University of California Press, 1967, pp. 281–297.
[39] A. Gersho, 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, J. Cybern. 3 (3) (1973) 32–57.
[41] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.

[42] R. Yager, D. Filev, Approximate clustering via the mountain method, IEEE Trans. Syst. Man Cybern. Part B Cybern. 24 (8) (1994) 1279–1284.
[43] I. Gath, A. Geva, Unsupervised optimal fuzzy clustering, IEEE Trans. Pattern Anal. Mach. Intell. 11 (7) (1989) 773–781.
[44] R. Hathaway, J. Bezdek, Y. Hu, Generalized fuzzy *c*-means clustering strategies using Lp norm distances, IEEE Trans. Fuzzy Syst. 8 (5) (2000) 576–582.
[45] R. Krishnapuram, J. Keller, A possibilistic approach to clustering, IEEE Trans. Fuzzy Syst. 1 (2) (1993) 98–110.
[46] C.T. Zahn, Graph-theoretical methods for detecting and describing gestalt clusters, IEEE Trans. Comput. C-20 (1) (1971) 68–86.
[47] R. Urquhart, Graph-theoretical clustering based on limited neighborhood sets, Pattern Recognit. 15 (3) (1982) 173–187.
[48] D.H. Fisher, Knowledge acquisition via incremental conceptual clustering, Mach. Learn. 2 (1987) 139–172.
[49] S. Haykin, Neural Networks: A Comprehensive Foundation, second ed., Prentice Hall, 1999.
[50] R. Xu, D. Wunsch, Survey of clustering algorithms, IEEE Trans. Neural Netw. 16 (3) (2005) 645–678.
[51] R. Xu., D.C. Wunsch, Clustering algorithms in biomedical research: a review, IEEE Rev. Biomed. Eng. 3 (2010) 120–154.
[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] S M. Ester, H.P. Kriegel, S. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of the Second 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, American Association for Artificial Intelligence Menlo Park, CA, USA, 1996, pp. 153–180.
[56] C.S. Wallace, D.L. Dowe, Intrinsic classification by MML-the snob program, in: Proceedings of the Seventh Australian Joint Conference on Artificial Intelligence, 1994, pp. 37–44.
[57] W. Wang, J. Yang, R.R. Muntz, STING: a statistical information grid approach to spatial data mining, in: Proceedings of the Twenty-third International Conference on Very Large Data Bases, 1997, pp. 86–195.
[58] G. Sheikholeslami, S. Chatterjee, A. Zhang, WaveCluster: a wavelet-based clustering approach for spatial data in very large databases, Int. J. Very Large Data Bases 8 (3–4) (2000) 289–304.
[59] R. Agrawal, G. Johannes, G. Dimitrios, P. Raghavan, Automatic subspace clustering of high dimensional data for data mining applications, in: Proceedings of the ACM SIGMOD International Conference on Management of Data Conference, 1998, pp. 94–105.
[60] A.K. Jain, M. Flynn, Data clustering: a review, ACM Comput. Surv. 31 (3) (1999) 264–323.
[61] H.P. Schwefel, Numerical Optimization of Computer Models, John Wiley, New York, 1981.
[62] L.J. Fogel, A.J. Owens, MJ 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, R.C. Eberhart, Swarm Intelligence (2001).
[66] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of the Fourth IEEE International Conference on Neural Networks, 1995, pp. 1942–1948.
[67] M. Dorigoand, T. Stützle, Ant Colony Optimization, MIT Press, 2004.
[68] F. Glover, Future paths for integer programming and links to artificial intelligence, Comput. Oper. Res. 5 (5) (1986) 533–549.
[69] K.S. Al. Sultan, A tabu search approach to clustering problem, Pattern Recognit. 28 (9) (1995) 1443–1451.
[70] W. Pedrycz, Collaborative fuzzy clustering, Pattern Recognit. Lett. 23 (14) (2002) 1675–1686.
[71] L.F.S. Coletta, L. Vendramin, E.R. Hruschka, R.J.G.B. Campello, W. Pedrycz, Collaborative fuzzy clustering algorithms: some refinements and design guidelines, IEEE Trans. Fuzzy Syst. 20 (3) (2012) 444–462.
[72] W. Pedrycz, P. Rai, Collaborative clustering with the use of fuzzy *c*-means and its quantification, Fuzzy Sets Syst. 159 (18) (2008) 2399–2427.
[73] W. Pedrycz, Knowledge Based Clustering: From Data to Information Granules, Wiley Publications, 2005.
[74] M. Prasad, C.T. Lin, C.T. Yang, A. Saxena, Vertical collaborative fuzzy *c*-means for multiple EEG data sets, in: Proceedings of the Sixth International Conference on Intelligent Robotics and Applications, 8102, Springer, 2013, pp. 246–257.
[75] C. Pizzuti, Overlapping community detection in complex networks, in: Proceedings of the Eleventh Annual Conference on Genetic and Evolutionary Computation, 2009, pp. 859–866.
[76] S. Gregory, A fast algorithm to find overlapping communities in networks, in: Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (PKDD), 2008, pp. 408–423.
[77] Y.Y. Ahn, J.P. Bagrow, S. Lehmann, Link communities reveal multi-scale complexity in networks, Nature 466 (2010) 761–764.
[78] G Forestier, P Gancarski, C Wemmert, Collaborative clustering with back ground knowledge, Data Knowl. Eng. 69 (2) (2010) 211–228.
[79] J. Handl, J. Knowles, An evolutionary approach to multiobjective clustering, IEEE Trans. Evolut. Comput. 11 (1) (2007) 56–76.
[80] A. Konak, D. Coit, A. Smith, Multiobjective optimization using genetic algorithms: a tutorial, Reliab. Eng. Syst. Saf. 91 (9) (2006) 992–1007.
[81] K. Faceili, A.D. Carvalho, D. Souto, Multiobjective clustering ensemble, in: Proceedings of the International Conference on Hybrid Intelligent Systems, 2006.
[82] M.K. Law, A. Topchy, A.K. Jain, Multiobjective data clustering, IEEE Conf. Comp. Vis. Pattern Recognit. 2 (2004) 424–430.
[83] D. Forsyth, 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 409 (2001) 860–921.
[85] C. Dorai, A.K. Jain, Shape spectra based view grouping for free form object, in: Proceedings of the International Conference on Image Processing, 3, 1995, pp. 240–243.
[86] S. Connell, A.K. Jain, Learning prototypes for on-line handwritten digits, in: Proceedings of the Fourteenth International Conference on Pattern Recognition, 1, 1998, pp. 182–184.
[87] E. Rasmussen, Clustering Algorithms, Information Retrieval: Data Structures and Algorithms, Prentice Hall, Englewood Cliffs, 1992, pp. 419–442.
[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 Intell. Syst. Appl. 11 (5) (1996) 4–7.
[90] J. Cohen, Communications of the ACM, Data Mining Association for Computing Machinery, 1996.
[91] A. Saxena, J. Wang, Dimensionality reduction with unsupervised feature selection and applying non-Euclidean norms for classification accuracy, Int. J. Data Wareh. Min. 6 (2) (2010) 22–40.
[92] K.S.Al. Sultan, M.M. Khan, Computational experience on four algorithms for the hard clustering problem, Pattern Recognit. Lett. 17 (3) (1996) 295–308.
[93] R. Michalski, R.E. Stepp, E. Diday, Automated construction of classifications: conceptual clustering versus numerical taxonomy, IEEE Trans. Pattern Anal. Mach. Intell. 5 (4) (1983) 396–409.
[94] J.C. Venter, The sequence of the human genome, Science 291 (2001) 1304–1351.
[95] J.L. Kolodner, Reconstructive memory: a computer model, Cogn. Sci. 7 (4) (1983) 281–328.
[96] C. Carpineto, G. Romano, An order-theoretic approach to conceptual clustering, in: Proceedings of the Tenth International Conference on Machine Learning, 1993, pp. 33–40.
[97] L. Talavera, J. Bejar, Generality-based conceptual clustering with probabilistic concepts, IEEE Trans. Pattern Anal. Mach. Intell. 23 (2) (2001) 196–206.
[98] M. Hadzikadic, D. Yun, Concept formation by incremental conceptual clustering, in: Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, 1989, pp. 831–836.
[99] G. Biswas, J.B. Weinberg, D.H. Fisher, Iterate: a conceptual clustering algorithm for data mining, IEEE Trans. Syst. Man Cybern. Part C 28 (2) (1998) 219–230.
[100] K. Thompson, P. Langley, Concept formation in structured domains, Concept Formation: Knowledge and Experience in Unsupervised Learning, Morgan Kaufmann, 1991.
[101] I. Jonyer, D. Cook, L. Holder, Graph-based hierarchical conceptual clustering, J. Mach. Learn. Res. 2 (2001) 19–43.
[102] M. Lebowitz, Experiments with incremental concept formation: UNIMEM, Mach. Learn. 2 (2) (1987) 103–138.
[103] S. Hanson, M. Bauer, Conceptual clustering, categorization and polymorphy, Mach. Learn. J. 3 (4) (1989) 343–372.
[104] T. Kohonen, The self-organizing map, Neurocomputing 21 (1–3) (1998) 1–6 Pages.
[105] J. Vesanto, E. Alhoniemi, Clustering of the self-organizing map, IEEE Trans. Neural Netw. 11 (3) (2000) 586–600.
[106] J.G. Upton, B. Fingelton, Spatial data analysis by example, Point Pattern and Quantitative Data, 1, John Wiley & Sons, New York, 1985.
[107] A. Strehl, J. Ghosh, R. Mooney, Impact of similarity measures on web-page clustering, in: Proceedings of the Workshop on Artificial Intelligence for Web Search, 2000, pp. 58–64.
[108] J.J. Fortier, H. Solomon, Clustering procedures, The Multivariate Analysis, SAS Institute Inc., Cary, NC, USA, 1996, pp. 493–506.
[109] M.A. Gluck, J.E. Corter, Information, uncertainty, and the utility of categories, in: Proceedings of the Seventh Annual Conference of the Cognitive Science Society, 1985, pp. 283–287.
[110] M.J.A.N. Condorcet, Essai sur l'Application de l'Analyse 'a la Probabilite´des Decisions Rendues a la Pluralite´des Voix, L'Imprimerie Royale, Paris, 1785.
[111] J.F. Marcotorchino, P. Michaud, Optimisation En Analyse Ordinale Des Donnees, Masson, Paris, 1979.
[112] J.E. Corter, M.A. Gluck, Explaining basic categories: feature predictability and information, Psychol. Bull. 111 (2) (1992) 291–303.
[113] A. Strehl, J. Ghosh, Clustering guidance and quality evaluation using relationship-based visualization, Intelligent Engineering Systems Through Artificial Neural Networks, St. Louis, Missouri, USA, 2000, pp. 483–488.
[114] S.V. Stehman, Selecting and interpreting measures of thematic classification accuracy, Remote Sens. Environ. 62 (1) (1997) 77–89.

[115] W.M. Rand, Objective criteria for the evaluation of clustering methods, J. Am. Stat. Assoc. 66 (336) (1971) 846–850.
[116] V. Rijsbergen, Information Retrieval, Butterworths, London, 1979.
[117] J.F. Brendan, D. Dueck, Clustering by passing messages between data points, Science 315 (2007) 972–976.
[118] E.B. Fowlkes, C.L. Mallows, A method for comparing two hierarchical clusterings, J. Am. Stat. Assoc. 78 (383) (1983) 553–569 2010.
[119] D.L. Olson, D. Delen, Advanced Data Mining Techniques, first ed., Springer, 2008.
[120] D.M.W. Powers, Evaluation: from precision, recall and F-factor to ROC, informedness, markedness & correlation, J. Mach. Learn. Technol. 2 (1) (2007) 37–63.
[121] P. Jaccard, Distribution de la flore alpine dans le bassin des dranses et dans quelques régions voisines, Bull. Soc. Vaud. Sci. Nat. 37 (1901) 241–272.
[122] J. Han, M. Kamber, 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 Trans. Syst. Man Cybern. 16 (1) (1986) 122–128.
[124] C.T. Lin, M. Prasad, J.Y. Chang, Designing Mamdani type fuzzy rule using a collaborative FCM scheme, in: Proceedings of the 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, 2001, pp. 117–120.
[126] C.H. Papadimitriou, K. Steiglitz, Chapter 6.1: the max-flow, min-cut theorem, Combinatorial Optimization: Algorithms and Complexity, Dover, 1998, pp. 120–128.
[127] A.S. Fotheringham, M.E. Charlton, C. Brunsdon, Geographically weighted regression: a natural evolution of the expansion method for spatial data analysis, Environ. Plann. 30 (11) (1998) 1905–1927.
[128] M. Honarkhah, J. Caers, Stochastic simulation of patterns using distance-based pattern modeling, Math. Geosci. 42 (5) (2010) 487–517.
[129] P. Tahmasebi, A. Hezarkhani, M Sahimi, Multiple-point geostatistical modeling based on the cross-correlation functions, Comput. Geosci. 16 (3) (2012) 779–797.
[130] S. Guha, R. Rastogi, K. Shim, Rock: a robust clustering algorithm for categorical attributes, in: Proceedings of the IEEE Conference on Data Engineering, 1999.
[131] T. Zhang, R. Ramakrishnan, M. Linvy, BIRCH: an efficient method for very large databases, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, ACM, 1996.
[132] D. Jiang, G. Chen, B.C. Ooi, K.L. Tan, S. W, epiC: an Extensible and Scalable System for Processing Big Data, in: 40th VLDB Conference, 2014, pp. 541–552.
[133] Z. Huang, A Fast Clustering Algorithm to Cluster very Large Categorical Data Sets in Data Mining, Data Mining and Knowledge Discovery, 1997 DMKD.
[134] A. Hinneburg, D. Keim, An efficient approach to clustering in large multimedia databases with noise, in: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, KDD, 1998.
[135] M.J.A. Berry, 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, Y. Edwards, The effectiveness of demographics and psychographic variables for explaining brand and product category use, Quant. Market. Econ. 1 (2) (2003) 223–224.
[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, Comput. Stat. Data Anal. 51 (12) (2007) 5940–5948.
[138] S. Dolnicar, Using cluster analysis for market segmentation–typical misconceptions, established methodological weaknesses and some recommendations for improvement, J. Market. Res. 11 (2) (2003) 5–12.
[139] R. Wagner, S.W. Scholz, R. Decker, The number of clusters in market segmentation, Data Analysis and Decision Support, Springer, Heidelberg, 2005, pp. 157–176.
[140] R.M. Durbin, S.R. Eddy, A. Krogh, G. Mitchison, Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids, Cambridge University Press, Cambridge, 1998.
[141] J.M. Kaplan, R.G. Winther, Prisoners of abstraction? The theory and measure of genetic variation, and the very concept of “Race”, Biol. Theory 7 (2012) 401–412.
[142] P.J. Carrington, J. Scott, Social network analysis: an introduction, The Sage Handbook of Social Network Analysis, 1, Sage, London, 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, in: Proceedings of the Fifth Joint Conference on Knowledge Based Software Engineering, 2002.
[145] M.G.B. Dias, N. Anquetil, K.M.D. Oliveira, Organizing the knowledge used in software maintenance, J. Univ. Comput. Sci. 9 (7) (2003) 641–658.
[146] R. Francesco, L. Rokach, 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, 7, third ed., Elsevier, Oxford, UK, 2010, pp. 112–118.
[149] G. Siemens, R.S.J.D. Baker, Learning analytics and educational data mining: towards communication and collaboration, in: Proceedings of the Second International Conference on Learning Analytics and Knowledge, 2012, pp. 252–254.
[150] R. Huth, C. Beck, A. Philipp, M. Demuzere, Z. Ustrnul, M. Cahynova, J. Kysely, O.E. Tveito, Classifications of atmospheric circulation patterns: recent advances and applications, Ann. N.Y. Acad. Sci. 1146 (1) (2008) 105–152.
[151] A. Bewley, R. Shekhar, S. Leonard, B. Upcroft, P. Lever, Real-time volume estimation of a dragline payload, in: Proceedings of the IEEE International Conference on Robotics and Automation, 2011, pp. 1571–1576.
[152] C.D. Manning, P. Raghavan, H. Schütze, An Introduction to Information Retrieval, Cambridge University Press, 2009.
[153] D.T. Nguyen, L. Chen, C.K. Chan, Clustering with multi-viewpoint-based similarity measure, IEEE Trans. Knowl. Data Eng. 24 (6) (2012) 988–1001.
[154] A. Bravais, Analyse mathamatique sur les probabilities des erreus de situation d'un point. Memoires presentes par divers savants. France: l Academie Royale des Sciences de l Institute de France, 9, 1846, pp. 255–332.
[155] K. Pearson, Mathematical contributions to the theory of evolution, III, regression, heredity, and panmixia, Philos. Trans. R. Soc. Lond. Ser. A 187 (1896) 253–318.
[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, K. Dan. Vidensk. Selsk. 5 (4) (1948) 1–34.
[157] L.R. Dice, Measures of the amount of ecologic association between species, Ecology 26 (3) (1945) 297–302.
[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, J. Wang, Dimensionality reduction with unsupervised feature selection and applying non-Euclidean norms for classification accuracy, Int. J. Data Warehous. Min. 6 (2) (2010) 22–40.
[161] S. Arora, I. Chana, A survey of clustering techniques for big data analysis, in: Proceedings of the Fifth International Conference on The Next Generation Information Technology Summit (Confluence), 2014.
[162] A.S. Shirkhorshidi, S. Aghabozorgi, T.Y. Wah, T. Herawan, Big Data Clustering: A Review, 8583, Springer, 2014, pp. 707–720. Lecture Notes Computer Science.
[163] H. Wang, W. Wang, J. Yang, P.S. Yu, Clustering by pattern similarity in large data sets, in: Proceedings of the International Conference on Management of Data, ACM, 2002.
[164] N. Bharill, A. Tiwari, A. Malviya, Fuzzy Based Scalable Clustering Algorithms for Handling Big Data Using Apache Spark, IEEE Trans. Big Data 2 (4) (2016) 339–352.
[165] X. Wu, X. Zhu, G.Q. Wu, W. Ding, Data mining with big data, IEEE Trans. Knowl. Data Eng. 26 (1) (2014) 97–107.
[166] P. Russom, Big Data Analytics, TDWI best practices report, The Data Warehousing Institute (TDWI) Research, 2011.
[167] C. Xiao, F. Nie, H. Huang, Multi-view *k*-means clustering on big data, in: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, AAAI, 2013.
[168] W. Fan, B. Albert, Mining big data: current status and forecast to the future, ACM SIGKDD Explor. Newsl. 14 (2) (2013) 1–5.
[169] K. Shvachko, H. Kuang, S. Radia, R. Chansler, The hadoop distributed file system, in: Proceedings of the IEEE Twenty-sixth Symposium on Mass Storage Systems and Technologies (MSST), 2010.
[170] D. Jeffrey, S. Ghemawat, MapReduce: a flexible data processing tool, Commun. ACM 53 (1) (2010) 72–77.
[171] J. Dean, S. Ghemawat, Map Reduce: a flexible data processing tool, Communications of the ACM 53 (1) (2010) 72–77.
[172] G. Celeux, G. Govaert, A classification EM algorithm for clustering and two stochastic versions, Comput. Stat. Data Anal. 14 (3) (1992) 315–332.
[173] L. Kaufman, P. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, Wiley, 1990.
[174] R. Ngand, J. Han, CLARANS: a method for clustering objects for spatial data mining, IEEE Trans. Knowl. Data Eng. 14 (5) (2002) 1003–1016.
[175] D. Sisodia, S. Sisodia, K. Saxena, Clustering techniques: a brief survey of different clustering algorithms, Int. J. Latest Trends Eng. Technol. 1 (3) (2012) 82–87.
[176] C. Zhong, D. Miao, R. Wang, A graph-theoretical clustering method based on two rounds of minimum spanning trees, Pattern Recognit. 43 (2010) 752–766.
[177] Y. Chen, S. Sanghavi, H. Xu, Improved graph clustering, IEEE Trans. Inf. Theory 60 (10) (2014) 6440–6455.
[178] A. Condon, R. Karp, Algorithms for graph partitioning on the planted partition model, Random Struct. Algorithms 18 (2) (2001) 116–140.
[179] W.E. Donath, A.J. Hoffman, Lower bounds for the partitioning of graphs, IBM J. Res. Dev. 17 (1973) 420–425.
[180] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 888–905.
[181] U. Luxburg, A tutorial on spectral clustering, Stat. Comput. 17 (4) (2007) 395–416.
[182] K. Rohe, S. Chatterjee, B. Yu, Spectral clustering and the high-dimensional stochastic block model, Ann. Stat. 39 (4) (2011) 1878–1915.
[183] S. Gunnemann, I. Farber, B. Boden, T. Seidl, Subspace clustering meets dense sub-graph mining: a synthesis of two paradigms, in: Proceedings of the IEEE International Conference on Data Mining, ICDM, 2010.
[184] K. Macropol, A. Singh, Scalable discovery of best clusters on large graphs, in: Proceedings of the VLDB Endowment, 3, 2010, pp. 693–702.
[185] J.J. Whang, X. Sui, I.S. Dhillon, Scalable and memory-efficient clustering of large-scale social networks, in: Proceedings of the Twelfth IEEE International Conference on Data Mining, ICDM, 2012.

[186] G. Karypis, V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs, SIAM J. Sci. Comput. 20 (1) (1998) 359–392.
[187] G. Karypis, V. Kumar, Multilevel *k*-way partitioning scheme for irregular graphs, J. Parallel Distrib. Comput. 48 (1998) 96–129.
[188] D. Yan, L. Huang, M.I. Jordan, Fast approximate spectral clustering, in: Proceedings of the Fifteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009, pp. 907–916.
[189] J. Liu, C. Wang, M. Danilevsky, J. Han, Large-scale spectral clustering on graphs, in: Proceedings of the Twenty-third International Joint Conference on Artificial Intelligence, IJCAI, 2013.
[190] W. Yang, H. Xu, A divide and conquer framework for distributed graph clustering, in: Proceedings of the Thirty-second International Conference on Machine Learning (ICML-15), 2015.
[191] S. Ghosh, S.K. Dubey, Comparative analysis of *k*-means and fuzzy *c*-means algorithms, Int. J. Adv. Comput. Sci. Appl. 4 (4) (2013) 35–39.
[192] S. Niwattanakul, J. Singthongchai, E. Naenudorn, S. Wanapu, Using of Jaccard coefficient for keywords similarity, in: Proceedings of the International Multi-Conference of Engineers and Computer Scientists (IMECS), I, Hong Kong, 2013, pp. 1–5.
[193] C. Chen, L. Pau, and P. Wang, Cluster analysis and related issue, R. Dubes Eds. Handbook of Pattern Recognition and Computer Vision, World Scientific, Singapore, pp. 3–32.
[194] A. Jain, R. Dubes, Algorithms for Clustering Data, Prentice-Hall, Englewood, Cliffs, NJ, 1988.
[195] C. Shi, Y. Cai, D. Fu, Y. Dong, B. Wu, A link clustering based overlapping community detection algorithm, Data Knowl. Eng. 87 (2013) 394–404.
[196] G. Palla, I. Derenyi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, Nature 435 (2005) 814–818.
[197] D.H. Wolpert, W.G. Macready, No free lunch theorem for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82.
[198] H. Bensmail, G. Celeux, A.E. Raftery, C.P. Robert, Inference in model-based cluster analysis, Stat. Comput 7 (1997) 1–10.
[199] D. Xu, Y. Tian, A comprehensive survey of clustering algorithms, Ann. Data Sci 2 (2015) 165–193.

Amit Saxena received his B.Sc. and M.Sc. degree from Bundelkhand University, India in 1984 and 1986, respectively, MCA degree from Jiwaji University, India in 1990, and Ph.D. degree from Guru Ghasidas University, India in 1998. Currently, Prof. Saxena is head of Dept. of Computer Science & Information Technology (CSIT), at Guru Ghasidas Central University, India. He is a member of IEEE (USA), Computer Society of India. Prof. Amit Saxena has authored a book on C Programming, published papers in National and International Journals, conference proceedings. His area of interests includes computational intelligence, data mining, and soft computing. He has been a reviewer of various conferences and papers. He has visited different countries on teaching and other academic assignments.

Mukesh Prasad received his Ph.D. degree in computer science from National Chiao Tung University, Hsinchu, Taiwan in 2015 and master degree in computer application from Jawaharlal Nehru University, New Delhi, India, in 2009. He is currently a Lecturer at School of Software, University of Technology Sydney, Australia.

Dr. Prasad current research interests include machine learning, pattern recognition, fuzzy systems, neural networks, artificial intelligence and brain computer interface. He has published papers in international journal and conferences including *IEEE Transactions, ACM, Elsevier* and *Springer*.

Akshansh Gupta is currently a postdoctoral research fellow in the school of Computational and Integrative Sciences at Jawaharlal Nehru University (JNU), New Delhi, India. He received his master and Ph.D. degree from the school of Computer and Systems Sciences, JNU, in the year 2010 and 2015 respectively. His research interests include signal processing, brain-computer interface, cognitive science, and healthcare.

Neha Bharill received the B.E. degree from Department of Information Technology, Bansal Institute of Science and Technology, Bhopal, India, in 2008, M.E. degree from Department of Computer Science and Engineering, Shri Govindaram Saksseria Institute of Technology and Science, Indore, India, in 2011. She is currently working towards the PhD degree in the Department of Computer Science and Engineering, Indian Institute of Technology, Indore, India. Her current research interests include fuzzy sets and systems, big data, pattern recognition, data mining and machine learning. She is reviewer of the IEEE Transactions on Cybernetics, IEEE Transactions on Fuzzy Systems, Swarm and Evolutionary Computation of Elsevier and the Complex & Intelligent Systems of Springer. She is a member of the IEEE and the IEEE Computational Intelligence Society.

Om Prakash Patel received the B.E. degree in Department of Information Technology from UITRGPV, Bhopal, India, in 2009, M.E. degree in Department of Computer Science and Engineering from Shri Govindaram Saksseria Institute of Technology and Science, Indore, India, in 2011. He is currently pursuing the Ph.D. degree in Department of Computer Science and Engineering from Indian Institute of Technology, Indore, India since 2014. His current research interests include quantum based neural network learning algorithm, pattern recognition, data mining, fuzzy based soft computing algorithm. He is a reviewer of Neural Processing Letter, Springer. He is a student member of IEEE.

Aruna Tiwari received the B.E. degree (Computer Engineering) in 1994 and M.E. degree (Computer Engineering) in 2000 from Shri Govindaram Saksseria Institute of Technology and Science, Indore and Ph.D. degree from Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal, India. She joined the Indian Institute of Technology Indore, India, in 2012, where she is currently working as an Assistant Professor with the Department of Computer Science and Engineering. Her research interests include soft computing techniques with neural network learning algorithms, evolutionary approaches, fuzzy based approaches for handling Big Data and nonstationary data. She has many publications in peer reviewed journals, International conferences, and Book chapters. She is reviewer of many journals some of them are IEEE Transaction on KDE, Neurocomputing journal of Elsevier etc. Dr. Tiwari is a member of IEEE Computational Intelligence Society and life member of Computer Society of India

Er Meng Joo is currently a Full Professor in Electrical and Electronic Engineering, Nanyang Technological University, Singapore. He has authored 5 books, 16 book chapters and more than 400 refereed journal and conference papers in his research areas of interest. His areas of research interests are intelligent control theory and applications, computational intelligence, robotics and automation, sensor networks, biomedical engineering and cognitive science. In recognition of the significant and impactful contributions to Singapore's development by his research project entitled "Development of Intelligent Techniques for Modelling, Controlling and Optimizing Complex Manufacturing Systems," Professor Er won the Institution of Engineers, Singapore (IES) Prestigious Engineering Achievement Award 2011. He is also the only dual winner in Singapore IES Prestigious Publication Award in Application (1996) and IES Prestigious Publication Award in Theory (2001). He received the Teacher of the Year Award for the School of EEE in 1999, School of EEE Year 2 Teaching Excellence Award in 2008 and the Most Zealous Professor of the Year Award 2009. He also received the Best Session Presentation Award at the World Congress on Computational Intelligence in 2006 and the Best Presentation Award at the International Symposium on Extreme Learning Machine 2012. Under his leadership as Chairman of the IEEE CIS Singapore Chapter from 2009 to 2011, the Singapore Chapter won the CIS Outstanding Chapter Award 2012. In recognition of his outstanding contributions to professional bodies, he was bestowed the IEEE Outstanding Volunteer Award (Singapore Section) and the IES Silver Medal in 2011. On top of this, he has more than 50 awards at international and local competitions.

Currently, Prof. Er serves as the Editor-in-Chief of 2 international journals, namely the Transactions on Machine Learning and Artificial Intelligence and International Journal of Electrical and Electronic Engineering and Telecommunications, an Area Editor of International Journal of Intelligent Systems Science, an Associate Editor of thirteen refereed international journals including the IEEE Transaction on Fuzzy Systems and an editorial board member of the EE Times. Professor Er is a highly sought-after speaker and he has been invited to deliver more than 60 keynote speeches and invited talks overseas. Due to outstanding achievements in research and education, he is listed Who's Who in Engineering Singapore, Second Edition, 2013.

Wei-Ping Ding received the B.S. degree in Computer Science and Technology, Nantong University, Nantong, China, in 2002, the M.S. degree in Software Engineer from Soochow University, Suzhou, China, in 2005, and the Ph.D. degree in Computer Application, Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2013. His current research interests include co-evolutionary algorithms, granular computing, data mining, machine learning and their applications in medicine.

Dr. Ding was a visiting researcher at Department of Mathematics & Computer Science, University of Lethbridge, Alberta, Canada, with the financial support "Jiangsu Government Scholarship for Overseas Studies" in 2011. In 2014, he was a Postdoctoral Researcher at the Brain Research Center, National Chiao Tung University (NCTU) with Professor Chin-Teng Lin, Hsinchu, Taiwan. Now, he is an Associate Professor in School of Computer Science and Technology, Nantong University, Nantong, Jiangsu, China. He is a member of Association of Computing Machinery (ACM), IEEE Computer Society (IEEE-CS), and China Computer Federation (CCF). He has authored or co-authored more than 60 papers in journals and conference proceedings. He was a recipient of National Natural Science Young Foundation of China in 2013. He was awarded an excellent-young teacher of Jiangsu Province sponsored by Qing Lan Project, Jiangsu Province, China, in 2014.

Chin-Teng Lin received the B.S. degree from National Chiao-Tung University (NCTU), Taiwan in 1986, and the Master and Ph.D. degree in electrical engineering from Purdue University, USA in 1989 and 1992, respectively. He is currently the Chair Professor of Faculty of Engineering and Information Technology, University of Technology Sydney, Chair Professor of Electrical and Computer Engineering, NCTU, International Faculty of University of California at San-Diego (UCSD), and Honorary Professorship of University of Nottingham. He was elevated to be an IEEE Fellow for his contributions to biologically inspired information systems in 2005, and was elevated International Fuzzy Systems Association (IFSA) Fellow in 2012. He is elected as the Editor-in-chief of IEEE Transactions on Fuzzy Systems since 2011. He also served on the Board of Governors at IEEE Circuits and Systems (CAS) Society in 2005–2008, IEEE Systems, Man, Cybernetics (SMC) Society in 2003–2005, IEEE Computational Intelligence Society (CIS) in 2008–2010, and Chair of IEEE Taipei Section in 2009–2010. Dr. Lin is the Distinguished Lecturer of IEEE CAS Society from 2003 to 2005, and CIS Society from 2015–2017. He served as the Deputy Editor-in-Chief of IEEE Transactions on Circuits and Systems-II in 2006–2008.

Prof. Lin was the Program Chair of IEEE International Conference on Systems, Man, and Cybernetics in 2005 and General Chair of 2011 IEEE International Conference on Fuzzy Systems. He is the coauthor of Neural Fuzzy Systems (Prentice-Hall), and the author of Neural Fuzzy Control Systems with Structure and Parameter Learning (World Scientific). He has published more than 200 journal papers (Total Citation: 20,155, H-index: 53, i10-index: 373) in the areas of neural networks, fuzzy systems, multimedia hardware/software, and cognitive neuro-engineering, including approximately 101 IEEE journal papers.