



## Survey paper

## From clustering to clustering ensemble selection: A review

Keyvan Golalipour<sup>a</sup>, Ebrahim Akbari<sup>a</sup>, Seyed Saeed Hamidi<sup>b</sup>, Malrey Lee<sup>c,\*</sup>, Rasul Enayatifar<sup>d</sup><sup>a</sup> Department of Computer Engineering, Sari Branch, Islamic Azad University, Sari, Iran<sup>b</sup> Department of Computer Engineering, Qaemshahr Branch, Islamic Azad University, Qaemshahr, Iran<sup>c</sup> Center for Advanced Image and Information Technology, School of Electronics & Information Engineering, Chonbuk National University, Jeonju, ChonBuk, South Korea<sup>d</sup> Department of Computer Engineering, Firoozkooh Branch, Islamic Azad University, Firoozkooh, Iran

## ARTICLE INFO

## Keywords:

Data clustering  
Cluster analysis  
Clustering ensemble  
Consensus clustering  
Clustering ensemble selection

## ABSTRACT

Clustering, as an unsupervised learning, is aimed at discovering the natural groupings of a set of patterns, points, or objects. In clustering algorithms, a significant problem is the absence of a deterministic approach based on which users can decide which clustering method best matches a given set of input data. This is due to using certain criteria for optimization. Clustering ensemble as a knowledge reuse offers a solution to solve the challenges inherent in clustering. It seeks to explore results of high stability and robustness by composing computed solutions achieved by base clustering algorithms without getting access to the features. Combining base clusterings together degrades the quality of the final solution when low-quality ensemble members are used. Several researchers in this field have suggested the concept of clustering ensemble selection for the aim of selecting a subset of base clustering based on quality and diversity. While clustering ensemble makes a combination of all ensemble members, clustering ensemble selection chooses a subset of ensemble members and forms a smaller cluster ensemble that performs better than the clustering ensemble. This survey includes the historical development of data clustering that makes an overview on basic clustering techniques, discusses clustering ensemble algorithms including ensemble generation mechanisms and consensus function, and point out clustering ensemble selection techniques with considering quality and diversity.

## 1. Introduction

In general, clustering analyzes and categorizes the available data based on certain criteria in which the same samples in one category will find maximum differences with the samples in other categories. Clustering has been employed in different sciences, including pattern recognition, statistics, bio-informatics, data mining, and machine learning (Eisen et al., 1998; Shi and Malik, 2000; Arbelaez et al., 2011; Sirbu et al., 2012; Zarikas et al., 2020).

Clustering is normally considered an unsupervised method since the connectivity and the succeeding grouping are measured with no knowledge regarding the class labels patterns. Clustering techniques optimize the global objective function based on the certain criteria such as similarity measure between patterns (Jain, 2010; Sharma and Seal, 2020). To obtain the final clusters, the clustering problem is partitioned accordingly through separating the data points into clusters and the results will be evaluated using the objective function. As data clustering has an unsupervised learning nature, it is classified as an essential, ill-posed problem (Barthélémy and Brucker, 2001; Drineas et al., 2004). Hence, optimizing the objective function for obtaining the global results is infeasible. Although there are several approaches (including heuristic

methods) to finding proper solutions that are often acceptable, but not necessarily the best, most of clustering methods can only converge to local optimum. However, in some cases they can converge to global optimum when datasets have spatial shapes. For instance, k-means has the capacity of converging to global optimum in cases where the clusters are separated appropriately with the same sizes (Jain, 2010).

Choosing a suitable clustering algorithm that can apply to all datasets is difficult. Therefore, different clustering algorithms have been proposed (Kleinberg, 2002; Han et al., 2011; Sinaga and Yang, 2020; Li et al., 2020). To solve this problem, the researchers have introduced the concept of consensus clustering(cluster ensembles) (Strehl and Ghosh, 2003). Consensus clustering or clustering ensemble(CE) makes a combination of several clustering results into the final clusters without gaining access to the algorithms or features (Fred and Jain, 2002; Faceli et al., 2007; Yu et al., 2013; Rafiee et al., 2013; Li et al., 2017; Wu et al., 2018; Hamidi et al., 2019). The consensus function is used for combination of the clusterings to obtain the final solution. Traditionally, all of the base clusterings(clustering members) are combined together with the use of a consensus function in a way to achieve the final solution. Remember that clusterings acquired this

\* Corresponding author.

E-mail address: [mrlee@jbnu.ac.kr](mailto:mrlee@jbnu.ac.kr) (M. Lee).

way are not necessarily capable of providing true benefits to the final solution of clustering ensembles (Hong et al., 2009; Azimi and Fern, 2009a).

Recently, authors suggested selecting a subset of clustering solutions to achieve better clustering results (Fern and Lin, 2008; Lu et al., 2013; Alizadeh et al., 2014b; Yousefnezhad et al., 2016; Shi et al., 2018; Abbasi et al., 2019). This approach is called clustering ensemble selection. The selection strategy aims to select better clustering members from among base clusterings. The main objective of the cluster ensemble selection (CES) is the selection of an appropriate subset of base clusterings(BC) and forms a smaller cluster ensemble that performs better than the set of all of the base clusterings (Fern and Lin, 2008; Azimi and Fern, 2009a). Fig. 1 shows the historical development of data clustering.

Clustering ensemble has a lot of beneficial applications in a large number of fields, including engineering. Some of the engineering application of the clustering ensemble are as transportation (Xiao et al., 2016), image processing (Liu et al., 2018; Wu et al., 2017), remote sensing (Yao et al., 2017; Sarkar et al., 2019), malware detection (Chakraborty et al., 2017; Zhang et al., 2017), time series analysis (Ramasso et al., 2015; Stolz et al., 2020), and business process management (Zhao et al., 2016). For example, Xiao et al. (2016) proposed a clustering ensemble method and applied it in fault diagnosis of high-speed train running gear. A clustering ensemble algorithm for image segmentation, which is the initial step in image processing, was presented in Liu et al. (2018) and Wu et al. (2017). A spectral clustering ensemble method was proposed in Yao et al. (2017) and Sarkar et al. (2019) for land cover identification using remote sensing images. Chakraborty et al. (2017) and Zhang et al. (2017) introduced an Intelligent malware categorization framework using the clustering ensemble method to detecting and combating the malware. Ramasso et al. (2015) introduced a clustering ensemble approach in acoustic emission time-series to estimate damage in composites. Stolz et al. (2020) proposed a monitoring network using the clustering ensemble method to identify similar and redundant air quality stations. A clustering ensemble approach to solving the problem of resource allocation in business process management was proposed by Zhao et al. (2016).

The rest of the article is organized as follows: Section 2 discusses the basic clustering techniques already introduced in the literature. Section 3 provides the evaluation criteria. Section 4 introduces cluster ensembles, including different ensemble generation mechanisms and different consensus functions. The focus of this section is on different consensus functions as the most basic step of all clustering ensemble methods. Then, the cluster ensemble selection techniques are described in Section 5. The survey is finally concluded in Section 6 with discussion about future research directions.

## 2. Basic clustering techniques

Clustering methods, from a general point of view, can be categorized into: partition methods and hierarchical methods. Partitional methods return a single clustering as final clusters. While, the hierarchical methods return a hierarchy of clusterings with producing the nested clusters of datasets which include agglomerative algorithms and divisive algorithms. Agglomerative algorithms consider each point (pattern) as a cluster, and iteratively, the pair of closest clusters in the current clustering is identified and merged into one in order to give rise to the next clustering. In contrast to the agglomerative algorithms, divisive algorithms generate a sequence of clusters at each step and subsequently the appropriately chosen cluster is split into two smaller clusters.

Some of clustering algorithms are known as basic clustering techniques, which are used in many advance clustering methods such as clustering ensemble. One of the famous partitional algorithms is k-means algorithm which is relatively simple and so is used more frequently by researchers. The popular basic algorithm in hierarchical

algorithms are Single-link (Sibson, 1973), Complete-link (King, 1967), Average-link (Olson, 1995).

Given a set of  $n$  objects in  $d$ -dimensional space,  $X = \{x_1, x_2, \dots, x_n\}$ , where  $x_i = (x_{i1}, \dots, x_{id})$ ,  $2 < d, i = 1, \dots, n$ . Clustering algorithm finds a partition of the data into  $k$  clusters that achieves a required objective, defined in terms of a given similarity (distance) measure  $d(x_i, x_j)$ . By notation, a clustering  $\pi$  of  $X$  is  $k$  groups of objects as  $\pi = \{c_1, c_2, \dots, c_k\}$ , where  $c_i$  is a cluster of clustering  $\pi$ ,  $i = 1, 2, \dots, k$ ; and  $\bigcap_{i=1}^k c_i = \emptyset$ ,  $\bigcup_{i=1}^k c_i = X$ . In the presence of outlier,  $\bigcup_{i=1}^k c_i \subset X$ .

### 2.1. Partitional clustering

The task defined for the partitioning clustering algorithms is partitioning the dataset into several clusters so that the points that exist within one cluster have higher similarity with each other compared to their similarity with the points within other clusters. Essentially, in the partitioning clustering, a dataset is divided into  $k$  clusters. The hard partitioning needs to meet two conditions: (1) every cluster needs to be consisted of at least one point, and (2) each point needs to belong to only a single cluster. To search throughout a dataset for all probable partitions, which is required to be done for a global optimization, can be prohibitively extensive. The most commonly used local optimization method is k-means (MacQueen et al., 1967).

**k-means clustering algorithm** is considered as an partitional clustering, in which only the numerical dataset is applied (MacQueen et al., 1967). The k-means algorithm is based on the idea that  $k$  clusters are developed in the way that points in its own cluster to centroid are closer than centroids of other clusters. The process of the algorithm starts performing with selecting the  $k$  points as centroid. Having chosen the  $k$  points, all points will be assigned to the closest centroid, by which  $k$  clusters are created. Afterwards, average of the points of each cluster will be measured as a centroid. These centroids are known as the mean vector where each field of the vector equals to each centroid of the cluster. Finally this process leads to the new clusters being created by the new centroid. In case the centroids do not change, the algorithm will be accomplished. Fig. 2 illustrates an example of k-means clustering with different  $k$ . The steps of the k-means algorithm is shown in algorithm 1 (Jain et al., 1988):

---

#### Algorithm 1 k-means algorithm applied to exploring $k$ clusters

---

1. Choose  $k$  points as the beginning centroids.
  2. Assign all points to the nearest centroid.
  3. Re estimates the centroid of each cluster.
  4. Make sure the centroids do not change by repeating the steps 2 and 3.
- 

Considering the fact that the vectors are only stored, it is necessary that storage space appears as  $O(n * d)$ , where  $n$  denotes the number of points, and  $d$  stands for the number of attributes (dimension). Besides the time indicators appear as  $O(I * k * n * d)$ , where  $I$  indicates the number of iterations which is needed for the convergence process. The k-means has linear complexity based on the number of points,  $n$ , in which the number of clusters and dimension of points are usually less than  $n$ . In case the data set appears in numerical form and convex shape, the k-means will not be complicated which leads to performing efficiently.

Taking advantage of the additional heuristics, scientists have tried to extend the basic k-means algorithm, in which the size of the clusters is optimized under the merged or split clusters. **k-modes** (Huang, 1997b) and **k-prototypes** (Huang, 1997a) are considered as two variants of k-means which have been mentioned in pattern recognition

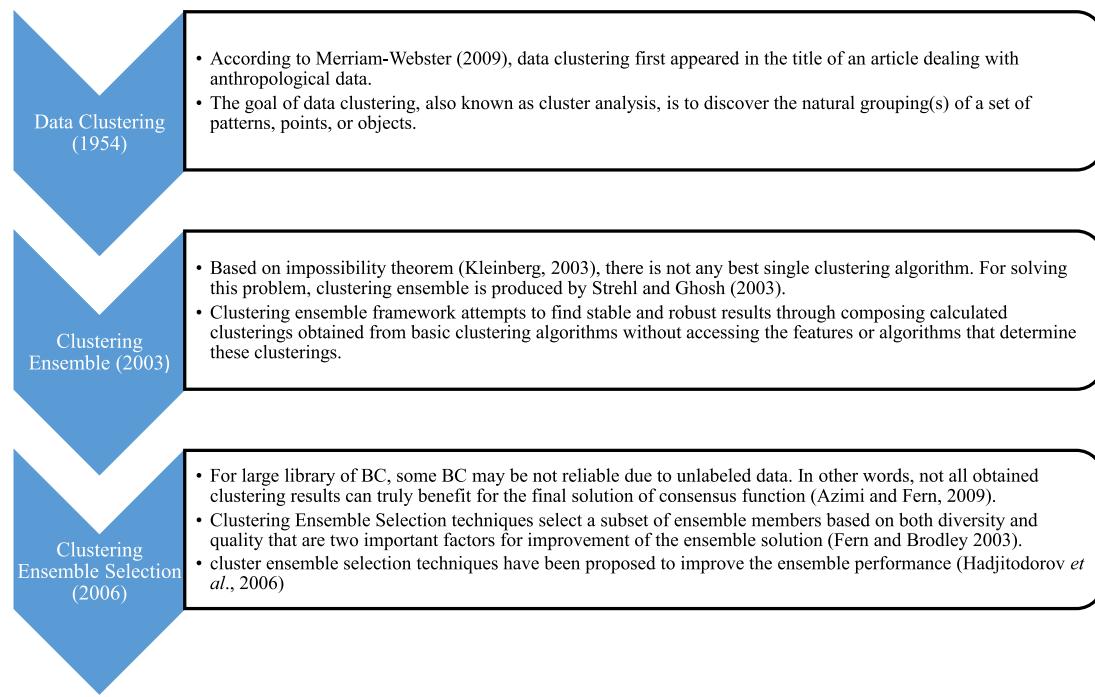
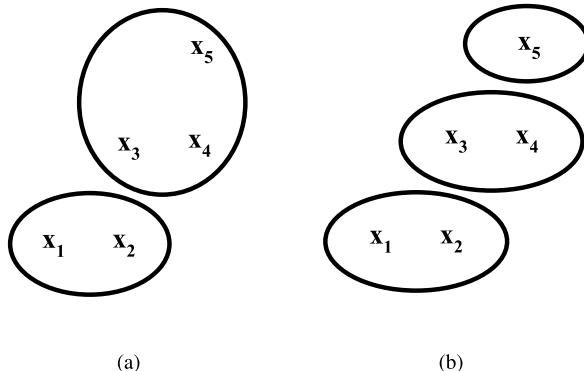


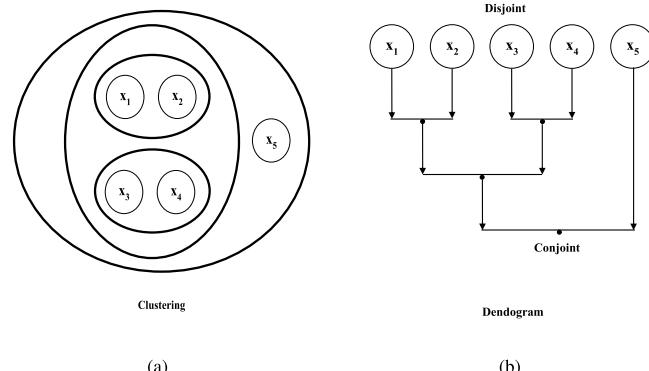
Fig. 1. Historical developments of data clustering.

Fig. 2. An example of k-means clustering (a)  $k = 2$  (b)  $k = 3$ . Note that  $X = \{x_1, x_2, x_3, x_4, x_5\}$ , and  $k$  is the number of clusters.

literature to clustering mixed numeric and categorical data. As mentioned above, k-means performs by assuming that the data points assigned to a single cluster which is named the hard assignment.

The k-means can be turned and extended into a **fuzzy c-means** which is developed by Dunn (1973) and enhanced by Bezdek (1981) in case each data point performs as a member of multiple clusters which has membership value indicating as a soft assignment. Substituting group samples with their centroids before they are clustered, leads to data reduction to accelerate the k-means and fuzzy c-means (Bobrowski and Bezdek, 1991).

Although the k-means has less space and time complexity in comparison with other clustering methods and it benefits from equal sized globular clusters and separated clusters to achieve the global optimum results, it has some drawbacks as follows: the k-means algorithm is limited to data in Euclidean spaces because the means and medians do not make sense. Moreover, the k-means is sensitive to outlier and noise. In general, the process of partitioning shows sensitivity to the initial choice of centroids, and in cases where the data have high dimensionality, it shows a tendency to break down. The reason is that

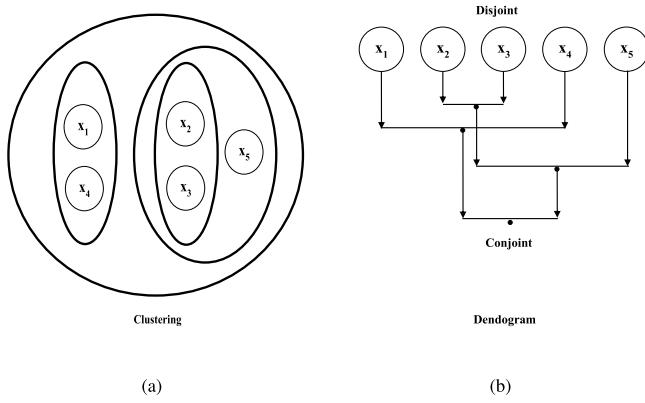
Fig. 3. An example of (a) hierarchical clustering and (b) the corresponding dendrogram. Note that  $X = \{x_1, x_2, x_3, x_4, x_5\}$ .

partitioning algorithms might generate increasingly poor results when dimensionality is increased (Jain, 2010).

## 2.2. Hierarchical clustering

A dendrogram is able to demonstrate the processes taking place in a hierarchical clustering algorithm. This tree-like diagram offers an understandable form of the hierarchical clustering. It consists of a number of layers of nodes each of which stands for a clustering. By cutting the dendrogram at the appropriate layer, a clustering is gained. Fig. 3 demonstrates an example of clustering and its dendrogram.

If hierarchical clustering algorithm built dendrogram as a bottom-up, it is called agglomerative hierarchical clustering and if this algorithm built dendrogram as a top-down then is called divisive hierarchical clustering (Guénoche *et al.*, 1991). Hierarchical clustering algorithms which will be studied in this section include: Single-link clustering (S-link), Complete-link clustering (C-link), Average-link clustering (A-link) algorithms which are famous hierarchical clustering techniques.



**Fig. 4.** Applying S-link algorithm on matrix  $D$  (Eq. (2)). (a) hierarchical clustering. (b) the corresponding dendrogram.

**S-link method:** In this method, also known as the nearest neighbor method, the distance between the two clusters is taken into account as the minimum distance. If  $c_i$  and  $c_j$  are two clusters, distance between them will be as Eq. (1).

$$D(c_i, c_j) = \min_{a \in c_i, b \in c_j} d(a, b) \quad (1)$$

where  $d(a, b)$  shows the distance between  $a$  and  $b$  points in clusters  $c_i$  and  $c_j$ .

Suppose  $D = [d(x_i, x_j)]$  is the  $n \times n$  proximity matrix, the sequence numbers of  $0, 1, \dots, (n-1)$  represents the clusterings, and  $L(k)$  is defined as the level of the  $k$ th clustering. Moreover,  $(m)$  stands for cluster with sequence number  $m$ , and  $d[(r), (s)]$  denotes the proximity between clusters  $(r)$  and  $(s)$ . The S-link algorithm is illustrated in Algorithm 2.

### Algorithm 2 S-link algorithm

1. Level  $L(0) = 0$  shows the disjoint clustering in which each cluster includes the primary points with the sequence number  $m = 0$ .
  2. Use the matrix  $D$  according to  $d[r, s] = \min d[i, j], i \neq j$  to calculate pair of clusters,  $(r), (s)$ , with highest similarity in the current clustering.
  3. Let  $m = m + 1$ , form a single cluster by merging clusters  $(r)$  and  $(s)$ , and set the level of this clustering to  $L(m) = d[(r), (s)]$ .
  4. Delete the corresponding rows and columns to clusters  $(r)$  and  $(s)$  and add a row and column corresponding to the newly formed cluster to update the proximity matrix,  $D$ .

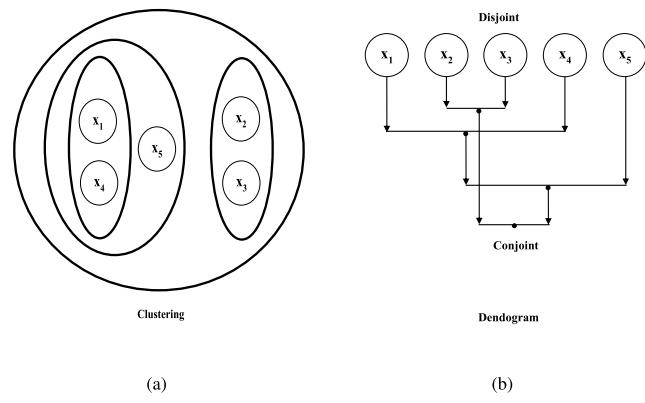
The proximity between the new cluster, denoted  $(r, s)$  and old cluster  $(k)$  is defined as  $d[(k), (r, s)] = \min d[(k), (r)], d[(k), (s)]$ .

  5. Stop the algorithm if all objects are in one cluster. Else, go to step 2.

For example, suppose that  $X = \{x_1, x_2, x_3, x_4, x_5\}$  and  $D$  is the proximity matrix, of size  $5 \times 5$ , which represents distances among data points (Eq. (2)).

$$D = x_3 \begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \\ 0 & 6 & 8 & 2 & 7 \\ 6 & 0 & 1 & 5 & 3 \\ 8 & 1 & 0 & 10 & 9 \\ 2 & 5 & 10 & 0 & 4 \\ 7 & 3 & 9 & 4 & 0 \end{pmatrix} \quad (2)$$

**Fig. 4** demonstrates the result of applying S-link algorithm on matrix  $D$  (Eq. (2)).



**Fig. 5.** Applying C-link algorithm on matrix  $D$  (Eq. (2)). (a) hierarchical clustering. (b) the corresponding dendrogram.

**C-link method:** In C-link method, which is called farthest neighbor method, the distance between the two clusters is taken into account as the maximum distance. If  $c_i$  and  $c_j$  are two clusters, the distance between them is calculated as follows:

$$D(c_i, c_j) = \max_{a \in c_i, b \in c_j} d(a, b) \quad (3)$$

If maximum distance between  $c_i$  and  $c_j$  be minimum, two clusters merge together in each level. The C-link algorithm is described in algorithm 3.

**Algorithm 3** C-link algorithm

1. Level  $L(0) = 0$  shows the disjoint clustering in which each cluster includes the primary points with the sequence number  $m = 0$ .
  2. Use the matrix  $D$  according to  $d[r, s] = \min d[i, j], i \neq j$  to calculate pair of clusters,  $(r), (s)$ , with highest similarity in the current clustering.
  3. Let  $m = m + 1$ , form a single cluster by merging clusters  $(r)$  and  $(s)$ , and set the level of this clustering to  $L(m) = d[(r), (s)]$ .
  4. Delete the corresponding rows and columns to clusters  $(r)$  and  $(s)$  and add a row and column corresponding to the newly formed cluster to update the proximity matrix,  $D$ .  
The proximity between the new cluster, denoted  $(r, s)$  and old cluster  $(k)$  is defined as  $d[(k), (r, s)] = \max d[(k), (r)], d[(k), (s)]$ .
  5. Stop the algorithm if all objects are in one cluster. Else, go to step 2.

Fig. 5 shows the result of applying C-link algorithm on matrix  $D$  (Eq. (2)).

**A-link method:** The distance that exists between the two clusters is considered as the average of all the distance. If  $c_i$  is a cluster with number of  $n_i$  members and  $c_j$  is a cluster with number of  $n_j$  members, distance between them in the A-link method will be as Eq. (4):

$$D(c_i, c_j) = \frac{1}{n_i * n_j} \sum_{a \in c_i, b \in c_j} d(a, b) \quad (4)$$

where  $d(a, b)$  stands for distance between  $a$  and  $b$  points in clusters  $c_i$  and  $c_j$ . The result of applying A-link algorithm on matrix  $D$  is the same with applying C-link algorithm (Fig. 5). It can be said that hierarchical clustering algorithms should be considered as the combinatorial optimization problem. Such approaches do not have difficulty in case of the local minima or identification of the primary points.

Additionally, it needs no information in regard to the number of clusters. However, the time and space complexity appears as  $O(n^2)$ ,

where  $n$  signifies the number of points, which makes this approach inappropriate for large dataset in many cases. Moreover, performance of hierarchical clustering algorithms is affected by noise and outlier (Jain et al., 1999). In summary, Most clustering techniques suffer from two problematic issues: data distribution and single clustering algorithms.

Data distribution leads to have data with different size, density, and shape. Additionally, data may be high dimensional and they may include outlier. With regard to the natural data such as web data, medical data, gene expression data and business data, which have distribution data identity, finding clusters using clustering algorithms is very difficult. In fact, clustering algorithms focus on specific parts of data characteristics. For example,  $k$ -means (MacQueen et al., 1967), which has a linear complexity, is suitable for large and separated convex-shaped data but it is not suitable for data with arbitrary shapes and different sizes. Single-link hierarchical clustering algorithm as another type of clustering (Sneath et al., 1973) is suitable for arbitrary shape, but is not suitable for large data, because it has quadratic complexity. It is also sensitive to the outlier that affects the clustering algorithm performance.

Different single clustering algorithms find different clusters in the natural data. This is because every clustering algorithm has a certain criteria for optimization. Moreover, a single algorithm with different parameters (e.g., the  $k$ -means algorithm with different values of  $k$ ) or initialization data order finds various clusters, because the data are unlabeled.

A challenging task in data clustering is finding the true number of clusters automatically. Internal criteria are applied to finding the true number of clusters, while the data is unlabeled. When the class labels are available, clustering algorithms partition the dataset into clusters without accessing labels, and then external criteria are employed to evaluate the clustering result.

### 3. Evaluation criteria

The evaluation of clustering is significant due to the lack of supervisor. The majority of the cluster validity measures that exist in the literature can be classified into two groups: internal criteria and external criteria.

#### 3.1. Internal quality measures

Internal criteria describes the basic structure of the data imposed by the clustering algorithm through assigning the best fit (based on a defined measure) between the data and the underlying structure (clustering) without resorting to a reference partition. In the following, a number of commonly-used internal measures are presented in brief.

**Dunn Index (DI):** The Dunn Index (Dunn, 1974) is defined as follows:

$$DI(k) = \min_{i=1,\dots,k} \left\{ \min_{j=i+1,\dots,k} \left( \frac{d_1(C_i, C_j)}{\max_{h=1,\dots,k} d_2(C_h)} \right) \right\} \quad (5)$$

where,

$$d_1(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y), \quad d_2(C_h) = \max_{x, y \in C_h} d(x, y) \quad (6)$$

where  $d(x, y)$  stands for the Euclidean distance between two data points  $x$  and  $y$ ,  $d_1(C_i, C_j)$  calculates the minimum distance between two clusters  $C_i$  and  $C_j$ , and  $d_2(C_h)$  defines the diameter of the cluster  $C_h$ . If there is a large distance between two small-diameter clusters, then we can say that  $DI(k)$  is large. The number of clusters, which is corresponding to the largest  $DI(k)$ , is determined as the estimated  $k$  value. DI identifies the properly-separated, compact clusters. Thus, the main objective is the maximization of the inter-cluster distance and, at the same time, minimization of the intra-cluster distance.

**Davies–Bouldin Index (DBI):** The Davies–Bouldin Index (Davies and Bouldin, 1979) is defined as follows:

$$DBI_k = \frac{1}{k} \sum_{h=1}^k F_{C_h} \quad (7)$$

where,

$$F_{C_h} = \max_{C_j \neq C_h} F_{C_h C_j}, \quad F_{C_h C_j} = \frac{f_1(C_h) + f_1(C_j)}{f_2(C_h, C_j)} \quad (8)$$

where  $C_h$  and  $C_j$  denote the  $h$ th cluster and the  $j$ th cluster,  $h, j \in \{1, \dots, k\}$ , respectively,  $f_1(C_h)$  stands for the average distance between the samples in the cluster  $C_h$  and the centroid of the cluster, and  $f_2(C_h, C_j)$  represents the distance between the centroids of the clusters  $C_h$  and  $C_j$ . The number of clusters, which is corresponding to the smallest  $DBI_k$ , is determined as the estimated  $k$  value. Similar to DI, DBI marks out the compact clusters that are situated far from each other.

**PBM Index:** The PBM Index (Pakhira et al., 2004) is defined as follows:

$$PBM_k = \left( \frac{1}{k} \times \frac{E_1}{E_k} \times D_k \right)^2 \quad (9)$$

where  $k$  signifies the number of clusters;  $E_1$  denotes the sum of distances between the objects of each cluster and the centroid of the cluster;  $E_k$  represents the sum of distances between each cluster centroid and the objects of the cluster, and  $D_k$  is the maximum distance between two clusters. The number of clusters, which is corresponding to the largest  $PBM_k$ , is determined as the estimated  $k$  value.

**Silhouette Index (SI):** The Silhouette Index (Kaufman and Rousseeuw, 2009) is defined as follows:

$$SI(k) = \frac{1}{k} \sum_{h=1}^k SI_h \quad (10)$$

where,

$$SI_h = \frac{1}{|C_h|} \sum_{i=1}^{|C_h|} \left[ \frac{b_i^h - a_i^h}{\max\{a_i^h, b_i^h\}} \right] \quad (11)$$

$$a_i^h = \frac{1}{|C_h| - 1} \sum_{l=1, l \neq i}^{|C_h|} d(x_i^h, x_l^h), \quad b_i^h = \min_{j \in \{1, \dots, k\}, j \neq h} \left\{ \frac{1}{|C_j|} \sum_{l=1}^{|C_j|} d(x_i^h, x_l^j) \right\} \quad (12)$$

where  $x_i^h$  and  $x_l^h$  are the  $i$ th and  $l$ th data points in the  $h$ th cluster,  $d(\cdot)$  stands for the Euclidean distance,  $a_i^h$  signifies the average distance from the  $i$ th data point to the other data points in the  $h$ th cluster, and  $b_i^h$  denotes the average distance from the  $i$ th point to the points in another cluster. The Silhouette value ranges between  $-1$  and  $1$ . Then, the number of clusters, which is corresponding to the largest  $SI(k)$ , is determined as the optimal  $k$  value. For each point, SI calculates a width concerning its membership in any cluster. Then, this silhouette width is taken into account as an average over all observations.

All of the previously-discussed internal measures are applicable to estimating the number of clusters existing within a single dataset, which typically includes the computation of clustering results for a range of various numbers of clusters. Subsequently, the internal measure plots the performance of the clustering as a function of the number of clusters. If both the internal measure and the employed clustering algorithm are sufficient for the considered dataset, by using the resulted performance or a knee point of the curve, the optimum number of clusters can often be determined.

In addition, all the previous works related to determine the number of clusters suffer from one main drawback, i.e., the implementation of internal criteria in the process of evaluating a cluster. This is due to the fact that the best quality value of the consensus result based on an internal measure does not necessarily provide the most helpful information (Manning et al., 2008). Since the internal criteria characterize the underlying structure of the data imposed by the clustering algorithm without accessing to the references partition, this evaluation is not similar for the algorithms employing an identical cluster model. For instance,  $k$ -means clustering works based on the object distances optimization, while the internal criteria used in  $k$ -means will likely overrule the resulting clustering.

### 3.2. External quality measures

The external criteria are capable of measuring the discrepancy between the structure defined by a clustering and another structure defined by the class labels. The external criteria evaluate the clustering results. Several common external criteria are introduced below.

**Normalized Mutual Information (NMI):** The Normalized Mutual Information introduced by [Strehl and Ghosh \(2003\)](#) can be defined as follows:

$$NMI(\pi_a, \pi_b) = \frac{-2 \sum_{i=1}^{k_a} \sum_{j=1}^{k_b} n_{ij} \log\left(\frac{n_{ij}}{n_{ia} n_{bj}}\right)}{\sum_{i=1}^{k_a} n_{ia} \log\left(\frac{n_{ia}}{n}\right) + \sum_{j=1}^{k_b} n_{bj} \log\left(\frac{n_{bj}}{n}\right)} \quad (13)$$

where  $\pi_a$  and  $\pi_b$  with  $k_a$  and  $k_b$  clusters, respectively are two clusterings on dataset  $D$  with  $n$  samples;  $n_{ij}$  stands for the number of common objects in cluster  $c_i$  in clustering  $\pi_a$  and in cluster  $c_j$  in clustering  $\pi_b$ ;  $n_{ia}$  denotes the number of objects in cluster  $c_i$  in clustering  $\pi_a$ ; and  $n_{bj}$  stands for the number of objects in cluster  $c_j$  in clustering  $\pi_b$ . The NMI value ranges from 0 to 1. In this range, 1 shows that two clusterings are the same.

**Adjusted Rand Index (ARI):** The Adjusted Rand Index ([Hubert and Arabie, 1985](#)) is defined as follows:

$$ARI(\pi_a, \pi_b) = \frac{\sum_{i=1}^{k_a} \sum_{j=1}^{k_b} \binom{n_{ij}}{2} - t_3}{\frac{1}{2}(t_1 + t_2) - t_3} \quad (14)$$

where,

$$t_1 = \sum_{i=1}^{k_a} \binom{n_{ia}}{2}, \quad t_2 = \sum_{j=1}^{k_b} \binom{n_{bj}}{2}, \quad t_3 = \frac{2t_1 t_2}{n(n-1)} \quad (15)$$

where  $\pi_a$  and  $\pi_b$  with  $k_a$  and  $k_b$  clusters, respectively are two clusterings on dataset  $D$  with  $n$  samples;  $n_{ij}$  represents the number of common objects in cluster  $c_i$  in clustering  $\pi_a$  and in cluster  $c_j$  in clustering  $\pi_b$ ;  $n_{ia}$  denotes the number of objects in cluster  $c_i$  in clustering  $\pi_a$ ; and  $n_{bj}$  stands for the number of objects in cluster  $c_j$  in clustering  $\pi_b$ . The ARI value lies between 0 and 1. When the two clusterings agree perfectly, the value is 1.

**Disagreement and Agreement Index (DAI):** The Disagreement and Agreement Index was proposed by [Yu and Wong \(2009\)](#) as an external measure. Let  $\pi^*$  is the consensus solution with  $k^*$  clusters, which is achieved by a consensus function from the set of base clusterings  $\{\pi_1, \pi_2, \dots, \pi_L\}$ . As a function of  $k$  value, DAI is defined as:

$$DAI(k) = \frac{1}{L} \sum_{i=1}^L \tau_k(\pi^*, \pi_i) \quad (16)$$

where,

$$\tau_k(\pi^*, \pi_l) = \frac{\sum_{i < j} 1\{m_{ij}^* \neq m_{ij}^l\}}{\sum_{i < j} 1\{m_{ij}^* = m_{ij}^l\}}, i = \{1, \dots, k^*\}, j = \{1, \dots, k_l\} \quad (17)$$

$$m_{ij}^l = \begin{cases} 1, & x_i \text{ and } x_j \text{ are in clustering } \pi_l; \\ 0, & \text{else.} \end{cases} \quad i, j = \{1, \dots, k_l\} \quad (18)$$

where  $m_{ij}^*$  and  $m_{ij}^l$  stand for the entries of the adjacency matrixes  $M^*$  and  $M^l$  corresponding to the clusterings  $\pi^*$  and  $\pi_l$ , respectively.

**F-measure (FM):** The F-measure (F-score) ([Larsen and Aone, 1999](#)) is defined as follows:

$$FM(\pi_a, \pi_b) = \max \sum_{i=1}^{k_a} \frac{2 \times n_{ia} \times \left( \frac{n_{ij}}{n_{ia}} + \frac{n_{ij}}{n_{jb}} \right)}{n \times \left( \frac{n_{ij}}{n_{ia}} + \frac{n_{ij}}{n_{jb}} \right)} \quad (19)$$

where  $\pi_a$  and  $\pi_b$  are two clusterings with  $k_a$  and  $k_b$  clusters, respectively;  $n$  is the total number of objects;  $n_{ij}$  denotes the number of common objects in cluster  $c_i \in \pi_a$  and in cluster  $c_j \in \pi_b$ ;  $n_{ia}$  stands for the number of objects in cluster  $c_i$ ; and,  $n_{jb}$  represents the number

of objects in cluster  $c_j$ . The F-measure best value is 1 and its worst score is 0.

These performance measures especially NMI are well established and widely used in the literature ([Fern and Brodley, 2004](#); [Fred and Jain, 2005](#); [Fern and Lin, 2008](#); [Ayad and Kamel, 2008, 2010](#); [Jia et al., 2011](#); [Franek and Jiang, 2014](#); [Yu et al., 2014](#); [Alizadeh et al., 2014a](#); [Abbasi et al., 2019](#)).

When the data is unlabeled, Internal criteria are employed for the aim of finding the true number of clusters. Generally, a single clustering algorithm is run with various numbers of clusters; afterward, the proper number of clusters is selected based on internal criteria. Many of researchers found the number of true clusters,  $k$ , approximately for real life datasets such as microarray dataset ([Monti et al., 2003](#); [Yu et al., 2007](#); [Ayad and Kamel, 2008](#); [Yu and Wong, 2009](#); [Mimaroglu and Erdil, 2011, 2013a](#); [Hamidi et al., 2019](#)). The general framework of these techniques is shown in Fig. 6.

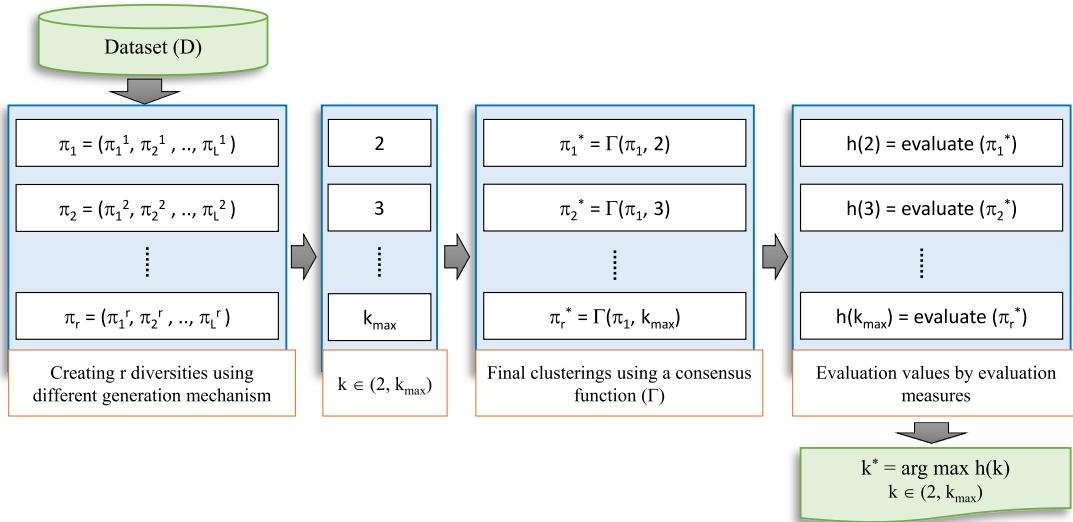
The application of a specific algorithm to clustering a set of input data generally leads to remarkable errors. On the other hand, the absence of unique criteria in the clustering process results in significant difficulties when selecting a clustering method from amongst a number of available ones. Indeed, a key challenge in clustering algorithms is the lack of a method to determine clearly which clustering method is the best one for a given set of input data. The impossibility theorem ([Kleinberg, 2002](#)), as well as its extended version ([Zadeh and Ben-David, 2009](#); [Correa-Morris, 2013](#)), maintains that any single clustering algorithm cannot be marked out as the best case to apply. Combining different clustering results by clustering ensemble is considered as an approach for solving the problems inherent in the single clustering algorithms.

### 4. Clustering ensemble

Clustering ensemble (CE) has been extensively applied to clustering research in recent years for the purpose of enhancing the robustness and quality of clustering results ([Fred and Jain, 2005](#); [Berikov, 2014](#); [Li et al., 2019a](#); [Zhou et al., 2020](#)). In this approach, multiple clustering results (clusterings) are integrated into final clusters with no access to features or algorithms. Note that CE only requires the access to base clusterings (BC) rather than the data itself; for that reason, it gives a convenience when dealing with privacy-related issues and knowledge reuse ([Strehl and Ghosh, 2003](#)). In numerous applications, different clusterings may exist for the objects under consideration. In such situation, the clusterings could be combined with each other to make a single solution. The accessibility of original features of raw data and the algorithms obtaining the clusterings in knowledge reuse is limited ([Strehl and Ghosh, 2003](#)).

Unlike the knowledge reuse, greater gains can be achieved with applying an ensemble to the improvement of clustering quality ([Topchy et al., 2005](#); [Mimaroglu and Erdil, 2013a](#); [Berikov, 2014](#)). The CE problem is more challenging compared to the classifier ensemble problem; this is because cluster labels are symbolic; thus in this process, a correspondence problem must be also solved. Through composing BC, the CE approach has the capacity of achieving a number of characteristics, e.g., robustness, stability, novelty, and scalability ([Gionis et al., 2007](#); [Ghosh and Acharya, 2011](#); [Hamidi et al., 2019](#); [Tan et al., 2020](#)). These characteristics are described below:

- **Robustness:** better average performance compared to single clustering algorithms.
- **Novelty:** finding a new combined solution unattainable by any single clustering algorithm.
- **Stability:** clustering solutions with lower sensitivity to noise and outlier.
- **Distributed computing and scalability:** In some conditions, data is distributed inherently and one cannot gather the entire data first at a certain center, which is because of the issues related to ownership and privacy or the bandwidth, computational, and storage costs.



**Fig. 6.** General framework for finding the appropriate number of final clusters using internal measure.

Two stages of clustering ensemble include:

1. **Diversity:** generation of base clusterings.
2. **Consensus function:** combination of the base clusterings to Obtain the final solution.

**Fig. 7** shows clustering ensembles architecture.

In the first stage, some basic clustering algorithms or one basic clustering algorithm with a variety of initializations are applied to the data set, and the obtained results are saved as labeling, clusterings, or base clusterings (BC). The next stage, consensus function will be used for combination of the clusterings to obtain the final solution. In conventional approaches, to improve the clustering quality, a set of large library of clusterings is constricted; after that, a Consensus Function (CF) is used to obtain the consensus solution, based on all base clusterings. Generally, both BC and CF are two problems which affect the clustering ensemble performance.

If the base clusterings have a better quality and more diversity, the ultimate outcome of the resulted clustering ensemble also has a higher quality (Kuncheva and Hadjitodorov, 2004; Fern and Lin, 2008; Li et al., 2019b). Many studies in this field has been based in a way that more diversity of base clusterings are created for combination of the clusterings (Hadjitodorov et al., 2006; Jia et al., 2011; Naldi et al., 2013; Alizadeh et al., 2014b). Although the previous experimental results have shown that the diversity in the base clusterings usually improves results of clustering, Azimi and Fern (2009b) has shown that having only more diversity is not well established for some of datasets. In addition, previous studies have also indicated that both diversity and quality have a great potential to impact the ensemble performance (Fern and Lin, 2008). However, obtaining the BC with appropriate diversity and quality requires a compulsory post analysis (Bae and Bailey, 2006). This is because, clustering solutions have a great dependency upon the similarity function applied by the particular algorithm used (Law et al., 2004).

On the other hand, combining multiple clusterings using a consensus function is a hard challenging in clustering ensemble, because there is no predefined labeling. Moreover, the labelings in BC are virtual or not real. Several independent studies have pioneered clustering ensembles as an innovative alternative to the traditional methods of clustering algorithms taxonomy (Strehl and Ghosh, 2003; Fern and Brodley, 2003a; Topchy et al., 2004a; Fred and Jain, 2005). Overall, the combination of multiple clusterings to find the final clusters is considered as (Non-Polynomial) NP-complement approach (Topchy et al., 2005).

In **Fig. 8**, clustering ensemble approach has been categorized by different solving for diversity generative mechanism and consensus function.

#### 4.1. Generative mechanism

Generative mechanism is composed of a number of approaches that can produce diversity of the individual clusterings of a given dataset. Regardless of obtaining the base clusterings with the appropriate diversity and quality, primary diversity is found by different ways. This is because there are no constraints about how the base clusterings must be acquired, different single clustering algorithms must be applied or one algorithm with different parameter settings. Generally, generative mechanism of diversity can be divided as follows:

##### 4.1.1. Different clustering algorithms

To create different and sporadic results of a data set, the easiest way is the use of various clustering algorithms. Every clustering algorithm partitions dataset from a special view (based on certain criteria), and different algorithms create different results on common dataset. Thus, errors can be different in various methods and may cause distribution in basic clustering algorithms. The basic algorithms, which were discussed in Section 2, are useful for generation diversity in clustering ensemble (Topchy et al., 2003, 2005; Mimaroglu and Erdil, 2013a; Berikov, 2014; Yu et al., 2014).

##### 4.1.2. Different parameters and built-in initialization

Another way to increase the diversity is changing of primary parameters of the clustering algorithms. This method can only be used from a single algorithm. If the algorithm is sensitive to the initialization then different results produce with changing initial value such as using randomness or various parameters of some algorithms, e.g., initializations and various values of  $k$  in k-means algorithm. Since k-means algorithm selects the initial centroids randomly, different solutions can be obtained by several runs of k-means. Although several clustering algorithms have been proposed by researches, k-means is used in the most of clustering ensemble methods typically for generation of diversity. Due to its simplicity and capability in clustering, k-means algorithm is the first choice in clustering ensemble (Topchy et al., 2003; Fred and Jain, 2005; Caruana et al., 2006; Ayad and Kamel, 2010; Mimaroglu and Erdil, 2013a; Yu et al., 2014; Bagherinia et al., 2020). Moreover, this algorithm is suitable for large dataset because of its low complexity.

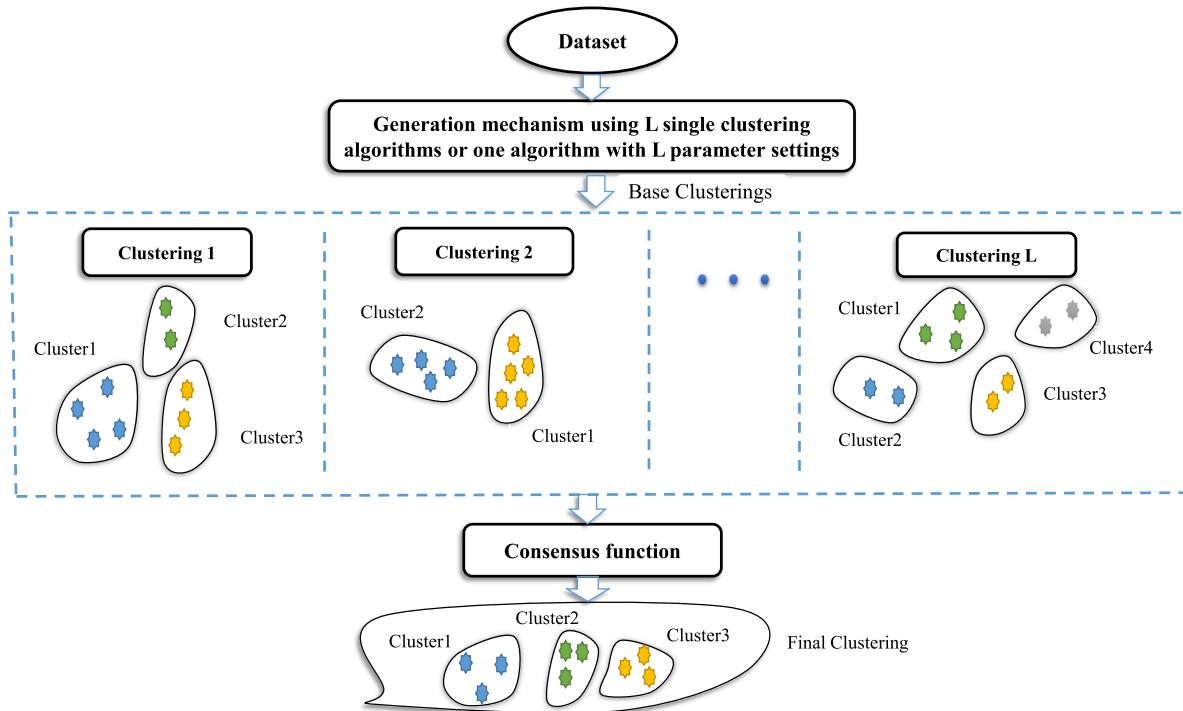


Fig. 7. Process of clustering ensemble approaches.

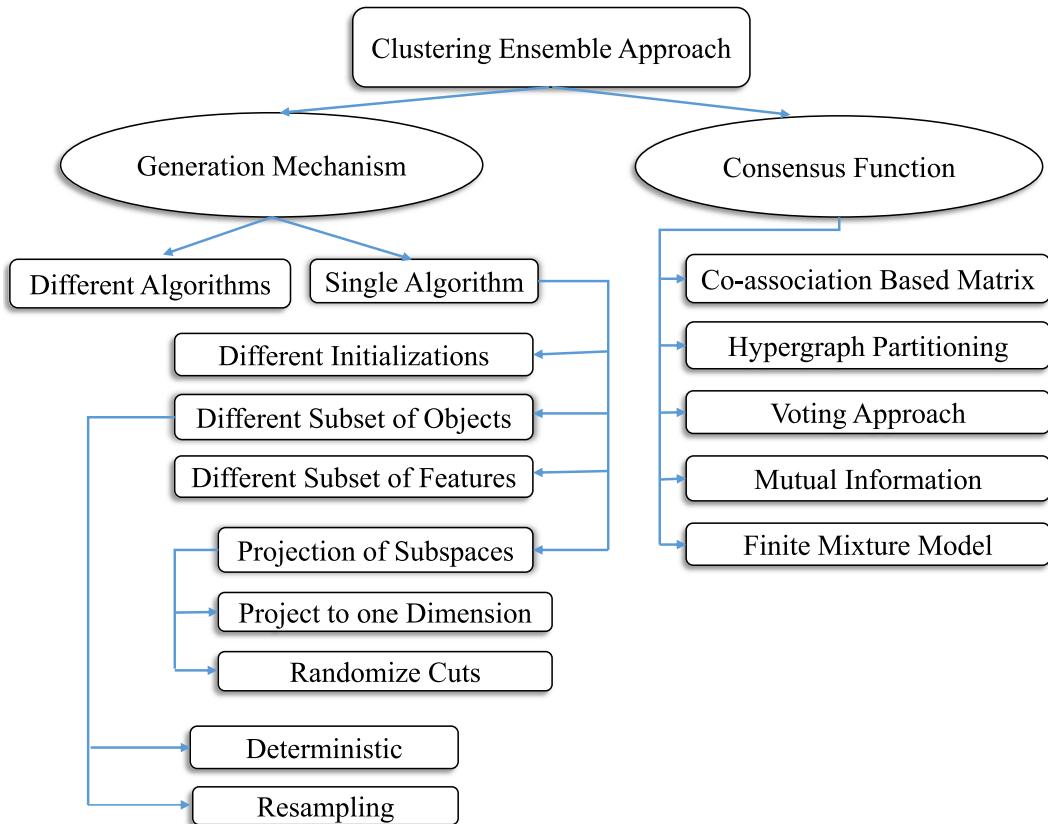


Fig. 8. Categorization of clustering ensemble approach for generative mechanism and consensus function (Minaei-Bidgoli et al., 2014).

#### 4.1.3. Different subset of features

In this method, different subsets of features are produced from initial samples, and then basic clustering algorithm is run on these samples. Due to differences in the features used in each of basic clustering algorithms, various combinations of clustering results will be achieved. In this regard, this method is useful for high dimensional data samples such as microarray dataset (Monti et al., 2003; Topchy et al., 2004b; Fred and Jain, 2005; Yu et al., 2007; Hong et al., 2008; Ye et al., 2016).

Good results in the clustering ensemble could be achieved by employing bagging (subset of features with replacement) method (Strehl and Ghosh, 2003; Hong et al., 2008). Theoretically, it is assumed that the loss of information is probable by using a subset of features. However, many researches showed that due to problem of “curse of dimensionality” in high dimensional data using a subset of features enhances the clustering results on high dimensional data (Strehl and Ghosh, 2003; Domeniconi et al., 2007). The problems related to clustering high-dimensional data include: poor discrimination of distances, redundant features, and irrelevant features.

**Poor discrimination of distances:** By increasing a data dimensionality, some definitions like distance, proximity, or neighborhood become less meaningful (Hinneburg et al., 2000; Aggarwal, 2001; Houle et al., 2010). The results of the previous works proved that by increasing dimensionality,  $d$ , the relative distance of the farthest point and the nearest point converges to 0. It is formulated by Eq. (20).

$$\lim_{d \rightarrow \infty} \frac{dist_{max} - dist_{min}}{dist_{min}} \rightarrow 0 \quad (20)$$

**Redundant features:** Usually clustering algorithms cluster datasets based on a distance or density similarity measure. By increasing the dimensionality the density is low and distance is high which causes the clustering algorithms to obtain final clusters with low quality (Aggarwal, 2001).

**Irrelevant features:** As the dimensionality increases the noise increases respectively. Since many of clustering algorithms are affected by noise such as Single-link hierarchical algorithm, they are not compatible with high dimensional data. Moreover, time and space complexity will increase by increasing the dimensionality; therefore, several algorithms such as hierarchical algorithms are not well suited for high-dimensional data (Elghazel and Aussem, 2013).

#### 4.1.4. Different subsets of data points

Creating the distribution (diversity) is one of the fundamental factors which improves the quality of the final clusters of the clustering ensemble. Using several subsets of all of data is one of the effective ways in obtaining this distribution which has two advantages: (1) Reduction of calculations (2) Increase of dispersion. Resampling methods are useful in establishing distribution in clustering especially for large datasets. In this regard various methods have been introduced including: sampling with replacement (bagging) (Levine and Domany, 2001; Minaei-Bidgoli et al., 2004), sampling without replacement (sub-sampling) (Agrawal and Goldfarb, 2006).

#### 4.1.5. Projecting data onto different subspaces

A random projection from  $d$  dimensions to  $d'$  dimensions is a linear transformation represented by a  $d \times d'$  matrix  $D$ , which reduces space dimension from  $d$  to  $d'$ . On the other hand, random projection suffers from a considerable instability, and various random projections might obtain different clustering results (Fern and Brodley, 2003b; Topchy et al., 2003, 2005).

The diversity (a set of BC) generated by the above methods is considered as input of consensus function.

#### 4.2. Consensus function

The consensus functions are the major hardship in clustering ensembles for combination of clusterings to produce final clusters (Topchy et al., 2004a). Making a combination of results from several supervised classifications has been a focal point of research in this field of study and it has provided the main incentive for combining the clusterings. Remember that there is not any labeled training data in clustering; as a result, it is more difficult to combine multiple clusterings compared to combining multiple classifications. As mentioned before, diversity and consensus function are two main problems in clustering ensembles. These problems can be explained as follows:

- Different consensus functions on the same diversity are obtained different results with different quality values. Fig. 9 shows effect of different consensus algorithms on the same diversity. Since consensus function as an optimization problem is ill-posed and NP problem (Strehl and Ghosh, 2003), different heuristic consensus algorithms have been proposed by researchers in which each consensus algorithm finds a consensus solution which may different with another consensus solution.
- One consensus function obtains different solutions based on different diversities of base clusterings. This scheme is shown in Fig. 10. Since diversity and quality are extremely important to consensus algorithm performance, one consensus algorithm can find a good result based on BC with having appropriate diversity and quality simultaneously.

Based on these problems, impact of different consensus algorithms on the same diversity will be discussed. Moreover, strengths and weaknesses of these algorithms will be explained.

Suppose  $\Pi = \{\pi_1, \pi_2, \dots, \pi_L\}$  be a set of  $L$  clusterings, where each clustering is signified by  $\pi_j = \{c_1^j, c_2^j, \dots, c_{k_j}^j\}$ ;  $k_j$  stands for the number of clusters in clustering  $\pi_j$ ,  $j = 1, 2, \dots, L$ . In the present study, there is a set of hard clusterings in which  $c_i^j \cap c_r^j = \emptyset, i \neq r = 1, 2, \dots, k_j; j = 1, 2, \dots, L$ . In the consensus clustering problem, a number of clusterings are formed with the use of single clustering algorithm with different initializations or different clustering algorithms. The goal of consensus function is the exploration of a clustering  $\pi^*$  sharing the highest volume of information regarding the produced clusterings  $(\pi_1, \pi_2, \dots, \pi_L)$ . On the basis of an average normalized similarity measure between the set of  $L$  clusterings ( $\Pi$ ) and a consensus clustering  $\pi$ , the consensus function can be described as an optimization problem attempting to explore the consensus solution  $\pi^*$  as follows:

$$\pi^* = \arg \max_{\pi \in \Pi_X} \left\{ \frac{1}{L} \sum_{j=1}^{j=L} \Phi(\pi_j, \pi) \right\} \quad (21)$$

where,  $\Pi_X$  stands for the set of all possible partitions with the set of objects  $X$ , and  $\Phi$  denotes a similarity measure. In the next sections, an analysis of each kind of consensus algorithms will be described. In this analysis, most popular clustering ensemble algorithms will be discussed. Also, their advantages and drawbacks will be explained.

##### 4.2.1. Co-association matrix

This consensus function is based on correlation matrix. The similarity between points (value correlation) shows the number of times the points are positioned within the same clusters of the clusters formed.

**Combining Multiple Clusterings Using EAC:** Fred and Jain (2005) introduced a method for combining various base clusterings, namely, evidence accumulation (EAC). They mapped the clustering ensemble into a new similarity criterion. The similarity between samples of  $x$  and  $y$  can be defined as:

$$sim(x, y) = \frac{1}{L} \sum_{i=1}^L \delta(\pi_i(x), \pi_i(y)) \quad (22)$$

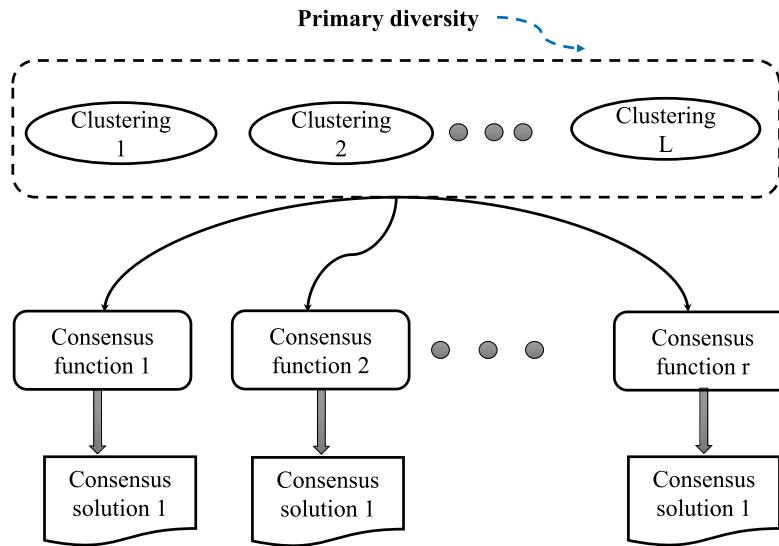


Fig. 9. Applying different consensus algorithms on the same diversity.

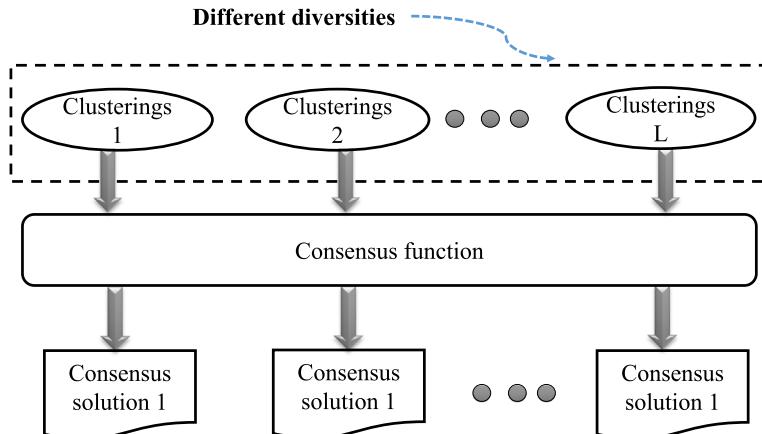


Fig. 10. Applying one consensus algorithm on different diversities.

where  $L$  is the number of base clusterings,  $\pi_i$  is a clustering and,

$$\delta(\pi_i(x), \pi_i(y)) = \begin{cases} 1, & \text{both } x \text{ and } y \text{ are in } \pi_i; \\ 0, & \text{else.} \end{cases} \quad (23)$$

In general, hierarchical algorithms like S-link, A-link, and C-link are employed to combine the results obtained by correlation matrix.

**CE Based on Normalized Edges:** A hierarchical clustering algorithm was proposed by Li et al. (2007) for the aim of enhancing the quality of the consensus solution that can be implemented in the co-association matrix. This method applies normalized edges to the measurement of the similarity between clusters. Between two clusters,  $C_i$  and  $C_j$ , the number of normalized edges (NE) can be computed by the following equation:

$$NE(C_i, C_j) = \frac{\text{edges}(C_i, C_j)}{(n_i + n_j)^{1+f(\theta)} - n_i^{1+f(\theta)} - n_j^{1+f(\theta)}} \quad (24)$$

where  $\text{edges}(C_i, C_j)$  represents the number of edges between two clusters  $C_i$  and  $C_j$ ,  $n_i$  and  $n_j$  stand for the number of objects in clusters  $C_i$  and  $C_j$ , respectively, and,  $n_i^{1+f(\theta)}$  and  $n_j^{1+f(\theta)}$  denote the estimated expected number of edges in the clusters  $C_i$  and  $C_j$ , respectively. Two clusters with the largest  $NE$  values are merged together. The worst time complexity of the algorithm is  $O(n^2 \log n)$ .

**Pairwise Similarity Matrix for CE:** Iam-On et al. (2008) suggested two link-based similarity matrices, i.e., the Connected-Triple Based Similarity (CTS) and SimRank Based Similarity (SRS). The algorithms

working based on link can detect more hidden relationships amongst objects. In the CTS matrix, each entry is calculated as follows:

$$CTS(x_i, x_j) = \frac{1}{M} \sum_{m=1}^M S_m(x_i, x_j) \quad (25)$$

where  $S_m(x_i, x_j)$  is the similarity of data points  $x_i$  and  $x_j$  in the  $m$ th ensemble member, which is calculated as follows:

$$S_m(x_i, x_j) = \begin{cases} 1, & \text{if } C(x_i) = C(x_j) \\ SWT(C(x_i), C(x_j)) \times DC, & \text{else.} \end{cases} \quad (26)$$

where  $SWT(C(x_i), C(x_j))$  signifies the similarity between clusters  $C_i$  and  $C_j$ ; DC is a constant decay factor and  $DC \in [0, 1]$ . Additionally, each entry in the SRS matrix is computed using the following equation:

$$SRS(a, b) = \frac{DC}{|N_a||N_b|} \sum_{a' \in N_a} \sum_{b' \in N_b} SRS(a', b') \quad (27)$$

where  $N_a$  and  $N_b$  denote the set of neighbors of object  $a$  and  $b$  respectively. Finally, to obtain consensus solution, hierarchical approaches and CSPA were employed.

**Probability Accumulation Algorithm:** Wang et al. (2009) introduced an approach for clustering aggregation, namely, the Probability accumulation matrix. In this method, the cluster size of the original clusterings considered. Each entry in the Probability accumulation (PA)

matrix is computed as follows:

$$PA(x_i, x_j) = \begin{cases} 1, & i = j \\ 0, & i \neq j \text{ and } C(x_i) \neq C(x_j) \\ \frac{1}{1 + \sqrt[m]{|X_k|}} & i \neq j \text{ and } C(x_i) = C(x_j) = k. \end{cases} \quad (28)$$

where  $C(x_i)$  means that  $x_i$  is partitioned into the  $C(x_i)$ th cluster;  $X_k$  denotes samples which are partitioned into the  $k$ th cluster; and,  $m$  signifies the number of attributes. After producing the PA matrix for each partition, the p-association matrix is calculated through computing the mean of all these matrices. The Minimum Spanning Tree (MST) is applied to p-association matrix with the highest lifetime criterion (Fred and Jain, 2005) to obtain the final solution. Moreover, they presented an Improved scheme with pre/post-processing strategy.

**Ensemble Clustering by Matrix Completion (ECMC):** Yi et al. (2012) introduced an EC approach with the implementation of the matrix completion technique. In this approach, first, a partially-observed similarity matrix is constructed. The matrix completion algorithm is used for the purpose of completing the partially-observed similarity matrix. Finally, a spectral clustering algorithm is applied to the matrix aiming to achieve the clustering result.

**Density-based similarity matrix construction:** Beauchemin (2015) presented a method to build density-based similarity matrix based on K-means with sub-bagging procedure. The k-means algorithm used for density estimation method and the sub-bagging method introduced to improve the density estimate accuracy. They showed that the construction of the density-based similarity matrix is the same with the ensemble generation technique of the evidence accumulation clustering approach (EAC), though normalization is not applied for EAC.

**Weighted co-association matrices:** Weighted co-association matrices were used by Berikov and Pestunov (2017) in order to propose an effective approach to EC. An evaluation function was used to computes the weights of the objects and produces the co-association matrix. At last, Hierarchical agglomerative algorithm employed to construct final partition.

The methods based on the Co-association matrix are straightforward and easy to use. The main drawbacks of methods based on correlation matrix are:

- They generally suffer from a quadratic computational complexity in the number of patterns and features, and cannot be applied to large datasets.
- There is no guidance for choosing the proper clustering algorithm for combination of clusters.
- A combination with a small number of clusterings might fail to offer an estimation of the correlation amount with a high reliability.

#### 4.2.2. Hypergraph partitioning

In hypergraph partitioning, hyperedges are introduced as clusters and vertices are equivalent of samples or points. Hypergraph algorithms are efficient for semi balanced clusters. Although partition of the hypergraph is NP-hard, there are several heuristic algorithms for partitioning of the hypergraph (Strehl and Ghosh, 2003).

A clustering,  $\pi_i$  can be shown as a label vector  $\lambda^i$ . Consensus function creates label vector  $\lambda$  with combination of  $r$  label vectors  $\lambda^1, \lambda^2, \dots, \lambda^r$  in which  $r$  is the number of clusterings. Objective function is defined by function  $\Gamma : N^{n*r} \rightarrow N^n$  mapping a set of clusterings to an integrated clustering  $\Gamma : \{\lambda^i | i \in \{1, 2, \dots, r\}\} \rightarrow \lambda$ .

The label vector of  $\lambda^i$  can be illustrated by binary matrix  $H^i$  in which each cluster is defined with a column. In case the row corresponding to an object with known label, all entries of the row in the binary membership indicator matrix  $H^i$  are considered equal to 1. While, the rows for unknown-label objects are equal to zero. Matrix  $H = (H^1 H^2 \dots H^r)$  as a hypergraph adjacency matrix is defined with  $n$  vertices and  $m = \sum_{j=1}^r k^j$  hyperedges. For example Table 1

**Table 1**

Binary shape clusters with  $r = 2$  clusterings and  $m = 5$  hyperedges ( $m = 3 + 2$ )  $K^1 = 3$ ,  $K^2 = 2$ , original label vectors (left) and equivalent hypergraph representation.

	$\lambda^1$	$\lambda^2$	$H^1$			$H^2$		
			$h^1$	$h^2$	$h^3$	$h^4$	$h^5$	
$x_1$	1	1	$v_1$	1	0	0	1	0
$x_2$	1	1	$v_2$	1	0	0	1	0
$x_3$	2	2	$v_3$	0	1	0	0	1
$x_4$	2	2	$v_4$	0	1	0	0	1
$x_5$	3	1	$v_5$	0	0	1	1	0

shows label vectors and corresponding binary matrix. Each hyperedge  $h^j, j = 1, 2, \dots, m$  which is illustrated as a column of the matrix  $H$  corresponds to the related cluster.

There are three algorithms in the hypergraph methods: Cluster-based Similarity Partitioning Algorithm (CSPA), HyperGraph Partitioning Algorithm (HGPA) and Meta-Clustering Algorithm (MCLA) (Strehl and Ghosh, 2003).

**Cluster-based Similarity Partitioning Algorithm (CSPA):** A clustering shows the relationships among the objects that exist within the same cluster; as a result, it can be applied to establishing a measure of pair-wise similarity. The entries of  $S$  signify the fraction of the clusterings in which two objects exist within the same cluster, and can be calculated in one sparse matrix multiplication  $\frac{1}{r} HH^T$ . The similarity matrix is employed for the aim of re-clustering the objects with the use of any reasonable similarity-based clustering algorithm. METIS (Karypis and Kumar, 1998) is utilized to partition the induced similarity graph.

**HyperGraph Partitioning Algorithm (HGPA):** This algorithm is used for partitioning the hypergraph through cutting a minimal number of hyperedges. All hyperedges are measured to have a similar weight. Moreover, all vertices are equally weighted. The partitions are obtained by the minimal cut algorithm into  $K$  unconnected components of approximately the same size. Hypergraph partitioning package, HMETIS, Han et al. (1997) is used for these partitions. Contrary to CSPA that considers the local piecewise similarity, HGPA only takes into consideration the relatively-global relationships among the objects across various partitions. Furthermore, HMETIS has a tendency to obtain a final partition in such a way that all clusters are roughly equal in their size.

**Meta-Clustering Algorithm (MCLA):** Integration in MCLA deals with cluster correspondence. MCLA identifies and consolidates groups of clusters and turn them into meta-clusters. This method finds the final clusters of objects in three main parts: Constructing the meta-graph, Computing meta-clusters and Computing clusters of the objects. The vertices of the meta-graph are the hyperedges  $h^j, j = 1, 2, \dots, m$  and the edge weights are proportional to the similarity between vertices. Through partitioning the meta-graph into  $k$  balanced meta-clusters, matching labels can be marked out. Each vertex is weighted proportionally to the size of its corresponding cluster. Balancing indeed makes sure that the sum of vertex-weights is roughly equal within each meta-cluster. At this step, graph partitioning package, METIS, is implemented to achieve the clustering of the  $h$  vectors. In the meta-graph, each vertex signifies a definite cluster label; as a result, a meta-cluster denotes a group of corresponding labels. For each of the  $k$  meta-clusters, the hyperedges are collapsed into a single meta-hyperedge. The meta-cluster that has the highest entry in the association vector is assigned with an object. Through this procedure, ties are broken in a random way.

Time complexity in CSPA, HGPA, and MCLA is shown as  $O(kn^2B)$ ,  $O(knB)$ , and  $O(k^2nB^2)$ , respectively (Strehl and Ghosh, 2003). There is one major problem with regard to the Hypergraph algorithms, these algorithms act properly just for balanced clusters (Topchy et al., 2004a). In case of imbalanced data clusters, hypergraph method does not function properly.

Recently, researchers have more focused on graph-based consensus function to improve the consensus solution on the real life datasets (Yu

et al., 2007; Iam-on et al., 2010; Mimaroglu and Erdil, 2011; Mimaroglu and Yagci, 2012; Mimaroglu and Erdil, 2013b). A key benefit of the graph-based consensus clustering approach is the existence of different ways to represent the vertices and edges. For example each vertex can be represented by object or cluster or both and each edge can be represented by similarity between two objects or two clusters. In the graph-based approach, user is capable of formulating a consensus function as a solution to  $k$ -way min-cut hypergraph partitioning problem (Strehl and Ghosh, 2003).

The principle objective of graph partitioning is the  $k$ -way min-cut partitioning in such a way that  $k$  sub-graphs with an approximately equal number of vertices could be achieved. Based on the above goal, different optimization criteria are defined by different graph partitioning algorithms. Two popular graph partitioning algorithms are the ratio cut criterion (Hagen and Kahng, 1992) and the normalized cut criterion (Shi and Malik, 2000).

The literature consists of various graph partitioning methods, including instance-based, cluster-based, and hybrid methods. As far as an instance-based method such as CSPA is concerned, the objects are taken into consideration as vertices, the edges weights are determined with the implementation of a similarity measure between the objects that exist within the clusters. This method works similar to co-association matrix which has usually high computational complexity. However, in the cluster-based method, a meta-graph is created in which clusters act as vertices, and the edges weights are computed using the similarity measure between the clusters. One of the most popular cluster-based methods is MCLA. In this method, the binary Jaccard measure is implemented as a similarity measure between two corresponding clusters (Strehl and Ghosh, 2003). On the other hand, in the hybrid method, both clusters and objects are taken into account as vertices, and the computation of the similarity measures is done simultaneously on the basis of the clusters and objects positioned between two vertices (Fern and Brodley, 2004).

**Hybrid Bipartite Graph Formulation (HBGF):** This method is an example for hybrid method. HBGF constructs a bipartite graph  $G = (V, E)$  where  $V = V^C \cup V^I$ ,  $V^C$  includes vertices that represent the clusters of clusterings,  $V^I$  includes vertices that represent the objects of the dataset  $X$  and the edge weight  $(i, j) \in E$  is defined as follows: If  $(i, j) \in V^C$  and  $(i, j) \in V^I$ ,  $W(i, j) = 0$ ; otherwise if  $i \in V^I$ ,  $j \in V^C$ , and the object  $i$  belongs to cluster  $j$ ,  $W(i, j) = W(j, i) = 1$  else if the object  $i$  is not belongs to cluster  $j$ ,  $W(i, j) = W(j, i) = 0$ .

**Link-based Cluster Ensemble (LCE):** This method is a developed method of HBGF which is a graph-based hybrid method (Iam-on et al., 2010). In LCE, the categorization of the bipartite graph is performed using a spectral clustering. However, LCE constructs a dense graph with implied similarity between every cluster and every object, which needs too many computations. These graph-based algorithms require number of clusters  $k$  as prior knowledge, while the estimation of the true number of clusters on real life datasets is considered a key challenge in most clustering algorithms and consensus function (Jain, 2010).

**Combining multiple clusterings using similarity graph (COMUSA):** This instance-based approach makes use of the evidence gathered within input clusterings, where the number of the clusters within the final clustering is determined in an automatic way (Mimaroglu and Erdil, 2011). Similar to CSPA, COMUSA uses the pair-wise similarity measure as the edge weight. Thus, the problematic issue of COMUSA is its execution time.

**COMUSA at cluster level (COMUSACL):** This cluster-based algorithm is able to combine multiple clusterings (Mimaroglu and Erdil, 2013a). COMUSACL is based on COMUSA, but unlike COMUSA, it operates upon a similarity graph at cluster level. Therefore, COMUSACL is executed faster than the COMUSA. The edge weight between a pair of vertices in the similarity graph is calculated using Eq. (29):

$$\text{weight}(c_i^k, c_j^l) = ECS(c_i^k, c_j^l) = \frac{1}{|c_i^k||c_j^l|} \sum_{d_m \in c_i^k, d_n \in c_j^l} sim(d_m, d_n) \quad (29)$$

where  $sim(d_m, d_n)$  stands for the number of times objects  $d_m$  and  $d_n$  are allocated to the same clusters within the collection of input clusterings.

Similar to COMUSACL, COMUSACL-DEW operates on a similarity graph that is built at the cluster level. On the other hand, unlike COMUSACL, in COMUSACL-DEW, the vertices are fused and the values of edge weight are updated. The most important difference between COMUSACL-DEW and COMUSACL lies in the fact that the former updates the edge weights in a dynamic way. This characteristic causes the algorithm to be capable of adjusting to diverse groups of datasets.

**Locally Weighted Meta-Clustering (LWMC):** LWMC is a cluster-based algorithm that makes use of the Jaccard coefficient for the purpose of calculating the edge weights between clusters (Huang et al., 2017). Then, the normalized cut algorithm (Shi and Malik, 2000) is employed to partition the graph into meta-clusters each of which contains a set of clusters. Final clusters are obtained by applying a locally weighted voting strategy on the basis of the ECI measure. The locally-weighted voting score of object  $o_i$  is computed as follows:

$$Score(o_i, MC_j) = \frac{1}{|MC_j|} \sum_{C_k \in MC_j} ECI(C_k) \cdot 1(o_i \in C_k), \quad (30)$$

with

$$1(\text{statement}) = \begin{cases} 1, & \text{if statement is true,} \\ 0, & \text{else.} \end{cases} \quad (31)$$

where  $MC_j$  denotes a meta-cluster, and  $|MC_j|$  stands for the number of clusters in the meta cluster  $MC_j$ . Object  $o_i$  assigns to the meta-cluster which gives it the highest score.

**Locally Weighted Graph Partitioning (LWGP):** This hybrid method works on the basis of bipartite graph formulating and partitioning (Huang et al., 2018). In other words, in this method, bipartite graph constructed using clusters and objects as graph nodes. The link weight between objects and clusters is determined considering the ECI value. The link weight between two nodes  $v_i$  and  $v_j$  is calculated using Eq. (32):

$$l_{ij} = \begin{cases} ECI(v_j), & \text{if } v_i \in O, v_j \in C, \text{ and } v_i \in v_j, \\ ECI(v_i), & \text{if } v_j \in O, v_i \in C, \text{ and } v_j \in v_i, \\ 0, & \text{else.} \end{cases} \quad (32)$$

where  $O$  is set of all objects, and  $C$  signifies set of all clusters. Then the transfer cut algorithm (Li et al., 2012) is used for the aim of partitioning the graph into a certain number of disjoint node sets.

**Consensus Clustering By Partitioning Similarity Graph:** Hamidi et al. (2019) suggested a cluster-based algorithm in order to prune the similarity graph. The number of clusters in this algorithm is obtained automatically with the use of the proposed graph cut. First, through pruning the similarity graph, outlier clusters are acquired automatically. Then, graph is split and sub-graphs are merged in a way to obtain meta-clusters. Finally, the consensus solution is obtained using majority voting. Each vertex is a cluster whose weight is the average of similarity values of the edges constructed by this vertex to other vertices, which is computed using similarity measures such as Jaccard.

**Ensemble of Locally Reliable Cluster Solutions:** Niu et al. (2020) suggested a clustering ensemble algorithm working on the basis of the kmedoids clustering algorithm. This method defines the concept of valid local clusters which are data points around a cluster center. At first to generate diverse clustering, kmedoids clustering algorithm is used. Then a weighted undirected graph (WUG) formed to show the relationship between clusters. Each vertex of WUG is a cluster and the similarity between a pair of clusters is applied as the weight of the edge between them. The similarity between two clusters  $C_i^{k_1}$  and  $C_j^{k_2}$  is computed using Eq. (33):

$$sim(C_i^{k_1}, C_j^{k_2})$$

$$\begin{cases}
 \frac{|(C_i^{k_1}, C_j^{k_2})|}{|U(C_i^{k_1}, C_j^{k_2})|} + \frac{|U_{q=1}^9 T_q(C_i^{k_1}, C_j^{k_2}) - U(C_i^{k_1}, C_j^{k_2})|}{\sqrt{\sum_{w=1}^m |M_w^{C_i^{k_1}} - M_w^{C_j^{k_2}}|^2}} & \text{if } \sqrt{\sum_{w=1}^m |M_w^{C_i^{k_1}} - M_w^{C_j^{k_2}}|^2} \leq 4\gamma, \\
 0, & \text{else.}
 \end{cases} \quad (33)$$

where  $C_i^{k_1}$  and  $C_j^{k_2}$  denote the  $i$ th cluster of  $k_1$ th clustering result and  $j$ th cluster of  $k_2$ th clustering result, respectively.  $m$  denotes the number of features in the dataset;  $T_q(C_i^{k_1}, C_j^{k_2})$  is an assumptive cluster;  $\gamma$  signifies the neighboring radius parameter of the valid cluster in the proposed algorithm; and  $M_w^{C_i^{k_1}}$  and  $M_w^{C_j^{k_2}}$  denote the  $w$ th feature of the center of clusters  $C_i^{k_1}$  and  $C_j^{k_2}$ , respectively. After obtaining the WUG, a normalized spectral clustering algorithm (Shi and Malik, 2000) is applied to obtain the final partition. They showed that the proposed algorithm complexity is linear with the data size.

#### 4.2.3. Voting or relabeling

Another term used for the voting approach in the literature is relabeling approach. This approach solves the correspondence problem between the labels of known and achieved clusters, unlike other approaches that do not necessitate the solution of the correspondence problem. The cluster labels are needed to be set in such a way that the best agreement could be achieved between the labels of two partitions. Re-labeling can be best performed between two clusterings with the use of the Hungarian algorithm (Kuhn, 1955). When an optimum re-labeling is executed perfectly, objects can be assigned to clusters by means of a simple voting; this helps to identify the final consensus partitions.

A general formulation for the voting problem as a multi-response regression problem was proposed by Ayad and Kamel (2010). The label correspondence makes it difficult to perform the unsupervised combination. Remember that in cases where all partitions have equal clusters, this correspondence problem can be often solved accurately.

**Plurality Voting (PV):** This method, which is based on bagging technique, was proposed by Dudoit and Fridlyand (2003) for the aim of enhancing the quality of clustering processes. In the bagging procedure or bootstrap aggregating, perturbed sets are produced by random replacement of the original dataset. The clustering algorithm is implemented in a repeated way on each bootstrap object, and the ultimate solution is achieved through plurality voting. They demonstrated that cluster votes are generally good indicators of the quality of a cluster assignment.

**Voting Active Clusters (VACs):** This method proposed by Turner and Agogino (2008) for obtaining ensemble clusterings. They also presented a VAC reward function named Difference Normalized Mutual Information (DNMI). The reward values were used to update the votes of clusterings in order to maximize an overall quality measure.

**Cumulative Voting (CV):** Ayad and Kamel (2008) introduced a consensus method named cumulative voting method. First, a reference partition is selected, Then an optimal consensus partition is produced. The reference partition was updated incrementally through averaging the partitions that have been combined up to that certain time. A weighting scheme was used to weight the clusters and a weight matrix was computed. Finally, the consensus solution was obtained by applying an agglomerative algorithm.

**Cumulative Voting-based Aggregation Algorithm (CVAA):** Saeed et al. (2012) proposed the cumulative voting-based aggregation algorithm consisting of two steps: obtaining the optimal relabeling for all clusterings and the voting-based aggregation algorithm to achieve the consensus solution. In this method, an initial reference clustering is used for all the ensemble members throughout the aggregation process. the reference clustering generated by Ward's clustering algorithm (Brown and Martin, 1996) which is a clustering method for Chemoinformatics applications.

**Table 2**  
Representation of new features.

	$\pi_1$	...	$\pi_L$
$x_1$	$\pi_1(x_1)$	...	$\pi_L(x_1)$
$x_2$	$\pi_1(x_2)$	...	$\pi_L(x_2)$
...	...	...	...
$x_n$	$\pi_1(x_n)$	...	$\pi_L(x_n)$

**Iterative Combining Clustering Method (ICCM):** Khedairia and Khadir (2019) proposed ICCM which processes the whole dataset iteratively. After generating base clusterings, a voting process employed among samples to extract a set of sub-clusters. Each sample votes for the sub-cluster in which it belongs. Samples with majority voting are allocated to their correspondent sub-clusters, and the samples that fail to achieve a majority are re-clustered in the subsequent iterations. In the end, a clustering algorithm is applied to classifying the already-acquired sub-clusters and extracting the final solution.

Compared to the other consensus clustering methods, voting approaches need comparatively more clusterings to achieve results of high reliability (Vega-Pons and Ruiz-Shulcloper, 2011). The Hungarian algorithm has a solution to the label correspondence problem with  $O(k^3)$ , where  $k$  stands for the number of clusters; as a result, the voting approaches could have high computational costs in the consensus partition. Typically, these types of algorithms are straightforward and simple to use.

#### 4.2.4. Mutual information

Through the mutual information, which is known as an information theory approach, the CE problem is transformed to the clustering of categorical data. A cluster label is delivered by each base clustering as a new feature that describes each data point. In other words, each clustering algorithm delivers an output that is taken into consideration as a categorical feature. In this process, the  $L$  features are able to act as an “intermediate feature space” upon which other clustering algorithm are able to work. The new features based on base clusterings for each sample  $x_i, i = 1, \dots, n$  is shown in Table 2.

For a CE, the objective function can be computed as the Mutual Information (MI) that is positioned between the partition and the labels within the ensemble. Eq. (34) represents the MI entropy.

$$H(P) = - \sum_{i=1}^n p_i \log_2 p_i \quad (34)$$

where  $P = (p_1, \dots, p_n)$  is a discrete probability distribution.

Based on the rule of independence of partitions, the function of MI can be articulated as the sum of pair-wise MIs between the target and given partitions. Based on the way MI has been conventionally defined, it is possible to calculate the value for a candidate partition solution and the related ensemble. The problem is that such a definition is incapable of introducing an algorithm with which the consensus can be maximized.

**Combining Multiple Weak Clusterings:** Topchy et al. (2005) introduced a consensus function on the basis of the Quadratic Mutual Information (QMI) that is achieved as a similarity measure between clusterings. Base on QMI, the similarity measure between two clusterings is defined as follows:

$$U(\pi_i, \pi_j) = \sum_{r=1}^{k_i} \rho(c_r^i) \sum_{h=1}^{k_j} \rho(c_h^j | c_r^i)^2 - \sum_{h=1}^{k_j} \rho(c_h^j)^2 \quad (35)$$

where,  $\rho(c_r^i) = \frac{|c_r^i|}{n}$ ;  $\rho(c_h^j) = \frac{|c_h^j|}{n}$ ; and,  $\rho(c_h^j | c_r^i) = \frac{|c_h^j \cap c_r^i|}{|c_r^i|}$ .

**Boosting based hierarchical cluster ensemble (Bob-Hic):** Rashedi and Mirzaei (2013) introduced a boosting based hierarchical cluster ensemble method which includes several boosting iterations. First, sample set is provided from the original dataset by applying random sampling. Then hierarchical clustering algorithm is employed.

Next, each sample's weight is updated based on the previous hierarchical clusterings' efficiency and the next clustering is generated on the basis of the new weights. The final partition is obtained by combining all hierarchies in the ensemble. This final integration is done through an information theoretic approach.

The objective function is equal to the total intra-cluster variance of the partition within the transformed space of labels. As a result, through applying the k-means algorithm to this space, the corresponding consensus solution can be provided. This method has an extremely low level of computational complexity,  $O(kLn)$ ; however, it needs to be restarted a number of times for the aim of preventing it from converging to low-quality local minima.

#### 4.2.5. Finite mixture model

The principle basis of this model is the idea that the labels are considered as the random variables in which they are derived from a distribution of possibility and shown as a sum of multi-nominal densities of the base clusterings.

**A mixture model for CE:** Topchy et al. (2004a) suggested a unified representation for multiple clusterings. They proposed a consensus function on the basis of the Expectation Maximization algorithm (EM). The consensus clustering can be shown as the maximum possibility measurement problem as shown in Eq. (36). The function of probability should be maximized with regard to the unknown parameters  $\Theta$  as shown in Eq. (37). In this case, the target is the identification of the best sum of density matching certain data.

$$\log L(\Theta|y) = \log \prod_{i=1}^N p(y_i|\Theta) = \sum_{i=1}^N \log \sum_{m=1}^M a_m p_m(y_i|\Theta_m) \quad (36)$$

where,

$$\Theta_* = \arg_{\Theta} \text{Max } \log L(\Theta|y) \quad (37)$$

In order to solve the aforementioned maximum probability function, the Expectation Maximization algorithm (EM) (Bailey et al., 1994) can be applied. With regard to changes provoked by data perturbations, the objective function is used to examine the multiple partitions. Hence, the achieved clusterings which are less sensitive to those perturbations are more likely to happen. By developing a correlation matrix and genetic algorithms, the problem related to the primary parameters adjustment is solved.

There are three main problems which limit finite mixture model including: (1) random variables are used to model the data (2) independency of the variables (3) identical distribution of the variables. Moreover, the number of clusters is needed as a prior knowledge in the mixture model. However, the low computational complexity,  $o(knm)$ , is the advantage of this method which is comparable with the k-means algorithm (Vega-Pons and Ruiz-Shulcloper, 2011).

Regardless of advantages and disadvantages of the aforementioned approaches, all these approaches have tried to improve the clustering ensemble performance aspects of combining the primary clusterings. Table 3 shows the comparison of the aforementioned approaches. In clustering ensemble approaches, all of the available base clusterings are combined together by a consensus function to produce the final solution. Some base clusterings may be unreliable since both noise and redundancy can be also found in these members. As a result, cluster ensemble selection methods have been proposed to enhance the clustering quality through the selection of a subset of base clusterings.

## 5. Clustering ensemble selection

Two main applications of clustering ensemble approach are knowledge reuse and improving the single clustering results (Strehl and Ghosh, 2003). In knowledge reuse issue, there are some clustering labelings in which the features of data are not available. The consensus function obtains a consensus solution such that the solution constructs new labeling of base clusterings. In this case, the number of base

clusterings may be small. While, based on the enhancement of the single clustering results quality, a set of wide library of clusterings is produced; after that, the consensus solution is achieved using a consensus function on the basis of all base clusterings (Fern and Lin, 2008; Azimi and Fern, 2009b).

In both small and large BC, diversity and quality importantly impact the ensemble performance. In other words, if the produced ensemble members (BC) differ from each other (diversity) and have a satisfactory quality, an ensemble solution of higher effectiveness can be obtained (Lu et al., 2013; Yang et al., 2014). It is due to the fact that combining several clusterings that are identical in terms of quality does not imply that combined clusterings outperform the individual clusterings (Haditidorov et al., 2006). Especially, when a small value of BC is available, the identical clusterings lead to inaccurate consensus solution. Then, the size of BC is important in diversity. Contrary to classification problems in which the data items' labels are known beforehand, in unsupervised clustering problems, the data items have no label. This causes some clustering results to be unreliable in large library of clusterings. As a result, all of the clusterings acquired cannot be necessarily beneficial to the final solution of CE (Hong et al., 2009; Azimi and Fern, 2009b; Li et al., 2019a).

The techniques of cluster ensemble selection (CES) have been introduced in the literature for the purpose of enhancing the ensemble performance (Haditidorov et al., 2006; Fern and Lin, 2008; Jia et al., 2011; Naldi et al., 2013; Wang et al., 2013). The main objective of these techniques is the selection of a subset from among a large library of clustering solutions based on diversity and quality. Based on these factors, CES forms a smaller CE performing as effectively as or better than the set of all available clustering solutions (Kuncheva and Haditidorov, 2004; Fern and Lin, 2008; Alizadeh et al., 2014b; Minaei-Bidgoli et al., 2014). Fig. 11 demonstrates the CES strategy.

### 5.1. Diversity and quality measures

Diversity measure is an important factor for selecting appropriate set of clusterings from the BC. Although various diversity measures are introduced by researchers, the exploration of a sensible quantitative diversity measure in CE has remained a difficult task (Kuncheva, 2005). The existing literature is consisted of several methods for measuring the diversity of ensemble members (Haditidorov et al., 2006; Fern and Lin, 2008; Azimi and Fern, 2009b; Naldi et al., 2013; Alizadeh et al., 2014b; Jackowski, 2018; Sesmero et al., 2018; Lim and Durrant, 2020). Most of these methods work on the basis of label matching between two partitions. Generally, in cases where the labels of a partition are not matched completely with another partition's labels, we say that the two partitions are diverse. The Adjusted Rand Index (ARI) (Hubert and Arabie, 1985) and Normalized Mutual Information (NMI) (Strehl and Ghosh, 2003) are widely-used methods applied to the measurement of partitions' diversity and quality.

Diversity measures, from a general viewpoint, are classified into external and internal diversity. Let  $H = \{h_1, h_2, \dots, h_L\}$  where, the set  $H$  has  $L$  base clusterings. In case there are class labels, the external diversity measure is defined on the basis of quality measures like NMI or ARI, as follows:

$$\text{diversity}(\bar{h}, h_i) = 1 - \text{quality}(\bar{h}, h_i) \quad (38)$$

where  $\bar{h}$  stands for known class label and  $h_i, i = 1, 2, \dots, L$  are clusterings.

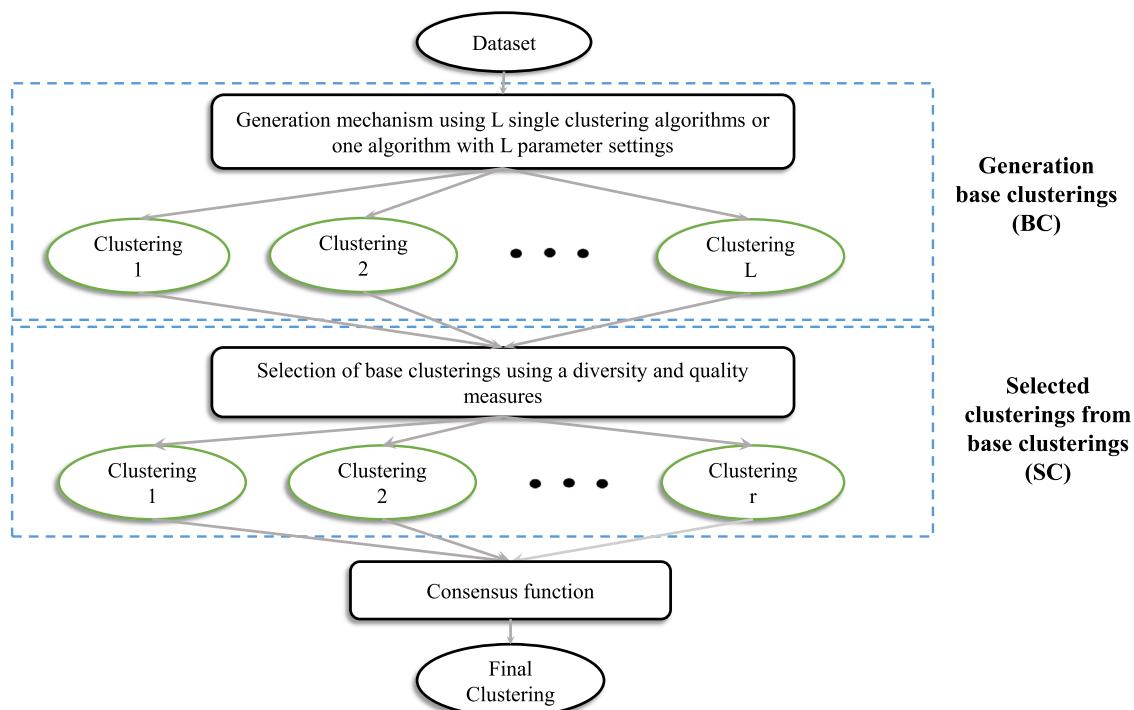
On the other hand, the internal diversity is generally categorized into pair-wise and non pair-wise diversity. In the former, one clustering is selected as class label implicitly, and other clusterings are measured by the selected class label. The pair-wise diversity is computed using Eq. (39):

$$\text{diversity}(h_i, h_j) = 1 - \text{quality}(h_i, h_j) \quad (39)$$

**Table 3**

Comparison of CE approaches based on consensus function.

CE Approach	Advantages	Disadvantages	Description
Co-association matrix	<ul style="list-style-type: none"> <li>No need the specification of the number of clusters in the consensus partition.</li> </ul>	<ul style="list-style-type: none"> <li>A quadratic computational complexity.</li> <li>There is no guidance for chosen the clustering algorithm for final clusters of combination of clusters.</li> <li>A combination with a small number of partitioning may be does not provide a reliable estimate of the amount of correlation.</li> </ul>	<ul style="list-style-type: none"> <li>Usually, hierarchical algorithms, such as S-link, A-link and C-link, are used for combining results by correlation matrix.</li> </ul>
Graph and HyperGraph partitioning	<ul style="list-style-type: none"> <li>Computational complexity for cluster-based hypergraph partitioning is low.</li> <li>There are different ways to representation of vertices and edges such as each vertex can be represented by object or cluster or both and each edge can be represented by similarity between two objects or two clusters.</li> <li>ability to handle missing data</li> </ul>	<ul style="list-style-type: none"> <li>These algorithms act properly for balanced clusters.</li> <li>They need the specification of the number of clusters in the consensus partition.</li> </ul>	<ul style="list-style-type: none"> <li>Three CSPA, HGPA, and MCLA are the famous methods in this approach, which are compared with many of others methods</li> </ul>
Relabeling and voting	<ul style="list-style-type: none"> <li>The approach is usually easy to understand and implement.</li> </ul>	<ul style="list-style-type: none"> <li>It requires a relatively larger number of clusterings to obtain reliable result.</li> <li>Correspondence problem can be solved in most cases with certain accuracy if all partitions have the same number of clusters.</li> <li>They have high computational cost.</li> </ul>	<ul style="list-style-type: none"> <li>Contrary to other approaches, the approach solves the correspondence problem.</li> </ul>
Mutual information	<ul style="list-style-type: none"> <li>Computational complexity of the algorithm is low.</li> <li>Complete avoidance of solving the label correspondence problem</li> <li>ability to handle missing data</li> </ul>	<ul style="list-style-type: none"> <li>this method requires to be restarted several times to avoid the convergence to low quality local minima.</li> </ul>	<ul style="list-style-type: none"> <li>This approach will be considered each output of clustering algorithm as a categorical feature. In this process, the L features can be considered as an intermediate feature space on which other clustering algorithm can work.</li> </ul>
Finite mixture model	<ul style="list-style-type: none"> <li>Computational complexity of the algorithm is low.</li> <li>Complete avoidance of solving the label correspondence problem</li> <li>ability to handle missing data</li> </ul>	<ul style="list-style-type: none"> <li>Adjusting the primary parameters is hard.</li> <li>It can be seen that those clusterings that are less sensitive to those perturbations are more likely to happen.</li> </ul>	<ul style="list-style-type: none"> <li>The main assumption is that the labels (label assigned to the object) are modeled as random variables drawn from a probability distribution described as a mixture of multivariate component Densities.</li> </ul>

**Fig. 11.** Cluster ensemble selection strategy.

where  $i \neq j = 1, 2, \dots, L$ . On the other hand, the non pair-wise diversity measure is computed by means of Eq. (40):

$$\text{diversity}(h^*, h_i) = 1 - \text{quality}(h^*, h_i) \quad (40)$$

where  $i = 1, 2, \dots, L$ ; and,  $h^*$  signifies a result achieved by a consensus function. Diversity and quality measures are very important to select an appropriate subset of base clusterings.

## 5.2. Selection of clusterings

In recent years, only a few researchers have focused heuristically on the way a subset of ensemble members must be chosen considering quality and diversity (Hong et al., 2009; Azimi and Fern, 2009b; Naldi et al., 2013; Yu et al., 2014).

**Moderate diversity CES:** This method was investigated by Hadjijitodorov et al. (2006) using the ARI measure for both diversity and quality of the ensemble and also for the selection of ensemble members. They used four diversity measures on the basis of ARI in order to select diversity. These diversity measures are  $D_{np-1}$ ,  $D_{np-2}$ ,  $D_{np-3}$ , and  $D_{np-4}$  that all are non pair-wise diversity. These measures are formulated as follow, respectively:

$$D_{np-1} = \frac{1}{L} \sum_{i=1}^L (1 - \text{ARI}(h^*, h_i)) \quad (41)$$

$$D_{np-2} = \sqrt{\frac{1}{(L-1)} \sum_{i=1}^L (1 - \text{ARI}(h^*, h_i) - D_{np-1})^2} \quad (42)$$

$$D_{np-3} = \frac{1}{2}(1 - D_{np-1} + D_{np-2}) \quad (43)$$

$$D_{np-4} = \frac{D_{np-2}}{D_{np-1}} \quad (44)$$

They showed that the ensembles selected with median diversity through  $D_{np-1}$  or  $D_{np-3}$  in this way are usually significantly better than a randomly chosen ensemble or ensemble selected through other diversity measures.

**Cluster And Select (CAS):** Fern and Lin (2008) studied various heuristic methods proposed in the literature for subsets selection with considering both the quality and diversity of the ensemble members. Their results suggested the combination of partition quality and diversity for CES with the use of the Sum of the Normalized Mutual Information (SNMI) measure that can be computed as expressed in Eq. (45):

$$SNMI(h, H) = \sum_{i=1}^L NMI(h, h_i) \quad (45)$$

As suggested by them, a base partition quality is proportional to the SNMI of the partition with respect to the set of base partitions, and the selected set diversity has an inverse proportionality to SNMI between each of its members and the other members of that set. Lower values indicate a higher level of diversity. The most effective method in this regard was introduced by Fern and Lin (2008), called Cluster And Select (CAS). Empirical tests revealed that this method could obtain a performance of the highest robustness. In this method, all ensemble members are first clustered into  $k$  clusters; after that, one solution is chosen from each cluster in order to achieve the final solution. In this method,  $k$  does not denote a definite value.

**Adaptive CES:** This method was proposed by Azimi and Fern (2009b) in which base clusterings are categorized into **stable** and **non-stable** depending on the NMI values. The authors showed that selecting the clusterings with more diversity, in case of non-stable clusterings, can enhance the quality of the final solution. In this strategy, consensus solution,  $h^*$ , is determined through applying a consensus function on

base clusterings,  $H$ . The applied consensus function in this strategy was co-association matrix with A-link hierarchical clustering (HAC-AL) (Fischer and Buhmann, 2003; Fern and Brodley, 2003b). Then the stable and non-stable set of base clusterings are marked by Eq. (46).

$$NMI(h^*, H) = \frac{1}{L} \sum_{i=1}^L NMI(h^*, h_i) \quad (46)$$

If the value of  $NMI(h^*, H)$  is greater than 0.5, the set  $H$  is marked as stable category otherwise the set  $H$  is marked as non-stable category. In case of stable, the final solution is considered  $h^*$  as output. On the other hand, in case of non-stable, the most dissimilar subset of base clusterings,  $H$ , from the  $h^*$  is selected and the output is obtained by applying the consensus function on this subset.

**Resampling-based CES:** A method for selecting a subset of base clusterings using resampling technique was proposed by Hong et al. (2009). They proposed evaluating the qualities of all obtained clusterings using resampling technique and selecting part of favorable clusterings to build the final solution. In resampling technique firstly,  $T$  subsets of the full dataset are selected randomly. Then  $T$  clusterings  $\{p_1, \dots, p_T\}$  are obtained by a consensus function. At last, the quality (accuracy) of the clustering  $\pi_i$  is calculated by Eq. (47),

$$q_i^{(a)} = \frac{\sum_{k=1}^T \|\pi_i, p_k\|}{T} \quad (47)$$

where,  $q_i^{(a)}$  denotes the quality (accuracy) of the clustering  $\pi_i$ ;  $\|\pi_i, p_k\|$  is the agreement between clusterings  $\pi_i$  and  $p_k$ . The diversity of the clustering  $\pi_i$  is calculated as follows:

$$q_i^{(d)} = 1 - \frac{\sum_{j=1, j \neq i}^L \|M^i, M^j\|}{L-1}, i = 1, \dots, L; \quad (48)$$

where,  $q_i^{(d)}$  denotes the diversity of the clustering  $\pi_i$ ;  $M^i$  signifies the similarity matrix correspond to clustering  $\pi_i$ ; and,  $L$  stands for the number of base clusterings. Then the fitness value of the clustering  $\pi_i$  is calculated by Eq. (49) which is a combination of accuracy and diversity.

$$\text{fitness}(\pi_i) = (1 - \lambda) \cdot \frac{q_i^{(a)}}{Q^a} + \lambda \cdot \frac{q_i^{(d)}}{Q^d}, i = 1, \dots, L \quad (49)$$

where,  $Q^a = \max\{q_1^{(a)}, \dots, q_L^{(a)}\}$ ;  $Q^d = \max\{q_1^{(d)}, \dots, q_L^{(d)}\}$ ; and  $\lambda$  ( $0 < \lambda < 1$ ) denotes a control parameter that is used to balance between accuracy and diversity. Finally, a subset of base clusterings based on the fitness values selected and combined together to obtain final solution.

**Selective Spectral Clustering Ensemble (SELSCE):** Jia et al. (2011) made a generalization on the selective clustering ensemble algorithm introduced formerly by Azimi and Fern (2009b) in order to present an innovative clustering ensemble method, called SELSCE algorithm based on bagging technique. The production of the ensemble members was done using Spectral Clustering that was capable of engendering diverse committees. The random initialization and the random scaling parameter, Nystrom approximation, were employed to generate the ensemble system components. When the component clusterings were produced, the bagging technique was applied to ranking and evaluating the component clustering. On the basis of this ranking, the ensemble members were chosen to obtain the final solution. This method first calculated  $T$  rankings of all the ensemble members  $\{RC^1, RC^2, \dots, RC^T\}$  and then a consensus ranking,  $RC^{final}$ , is computed by Eq. (50). Finally, a subset of ensemble members selected according to  $RC^{final}$  values.

$$RC^{final} = \frac{\sum_{i=1}^T RC^i}{T} \quad (50)$$

**Selective clustering ensemble based on covariance (SCEBC):** A diversity measure was introduced by Lu et al. (2013) considering on covariance.  $L$  clustering results are deemed to an  $n$ -dimensional random variable  $X = (X_1, \dots, X_n)$ . Then the covariance matrix, which is a symmetric matrix, is defined by calculating the covariance of the

random variable of  $X$ . The values on the diagonal are the variances of variable of  $X$ . At last, only the difference between the base clustering results are considered, and the positive correlation and negative correlation are not considered. By selecting the maximum value of these differences, the subset of clusterings is chosen and then a consensus function (CSPA) is applied on this subset of clusterings.

**CES based on relative validity indexes:** Naldi et al. (2013) introduced a number of relative cluster validity indexes on the basis of both diversity and quality in a way to be applicable to clusterings selection. The authors also examined the effects of diversity upon partitions (clusterings) employed for the ensemble. To do this, they implemented various relative diversity measures. They used different internal validity indexes such as Eqs. (5), (7), and (10). They proposed different approaches for cluster ensemble selection, namely, Single Index Selection (SIS), Combination of relative indexes (CRI) and Best Validated Consensus Partition (BVCP). The optimal value of some validity index method such as DI index is corresponding to the maximum value, while the optimal value of some others validity index such as DBI index corresponds to the minimum value. In the SIS method the value of validity index of each clusterings is calculated, Then a subset of the base clusterings is selected, where the number of selected clusterings is chosen by expert, which is unknown in advance, while, The CRI method combine the value of validity indexes and uses three CRI methods. The BVCP method select the partition with the best validation value (individually or combined) according to the validity indexes. They consider a function *diversity* that can be computed using Eq. (51):

$$\text{diversity}(\pi_i, \Pi) = 1 - \sum_{\pi_j \in \Pi, j \neq i} \frac{s(\pi_i, \pi_j)}{|\Pi| - 1} \quad (51)$$

where  $\pi_i$  is a base partition;  $\Pi$  denotes the set of base clusterings;  $s(\pi_i, \pi_j)$  stands for the similarity measure between partitions  $\pi_i$  and  $\pi_j$ ; and  $|\cdot|$  is the set cardinality.

**Hybrid clustering solution selection strategy (HCSS):** It was proposed by Yu et al. (2014) on the basis of a weighting consensus function. HCSS makes a combination of different techniques of feature selection. The problem of selection of base clusterings in HCSS is converted to feature selection. They applied four feature selection strategies, i.e., (1) the spectrum-based feature selection approach (SPEC), (2) the feature selection approach based on double input symmetrical relevance (DISR), (3) the feature selection approach based on min-redundancy max-relevance (MRMR), and (4) the RELIEF based feature selection approach (RELIEF). By applying these four feature selection methods, four subsets of base clusterings are obtained. After that, a merged subset of these subsets is selected by assigning weights to various subsets and various clustering solutions in each subset based on a unified weighting function. Next, a consensus matrix is created on the basis of the selected clustering solutions. Afterwards, HCSS builds a graph considering both the dataset and consensus matrix. At last, the final solution is obtained by applying the normalized cut algorithm (Shi and Malik, 2000) to the consensus matrix constructed from the selected base clusterings. Additionally, a new criterion is designed on the basis of NMI for the aim of assessing the effectivity of the strategies of selecting the clustering solution, which is calculated using Eq. (52):

$$F = \frac{NMI(Y_1, Y)}{NMI(Y_2, Y)} \quad (52)$$

where  $Y$  represents the set of ground truth labels,  $Y_1$  stands for the set of selected clustering solution labels, and  $Y_2$  is the remaining clustering solutions label set.

**CES based on APMM criterion:** Alizadeh et al. (2014a) proposed a new criterion, called Alizadeh–Parvin–Moshki–Minaei (APMM), and a new method, i.e., Extended Evidence Accumulation Clustering (EEAC). They suggested selecting a subset of more effective clusters instead of selecting a subset of primary clusterings. The new cluster validity criterion, which is based on NMI, was used for the purpose of evaluating the association between a cluster and a set of partitionings, which can

be calculated using Eq. (53).

$$APMM(C_i^a, P^{b^*}) = \frac{-2 n_i^a \log \left( \frac{n}{n_i^a} \right)}{n_i^a \log \left( \frac{n_i^a}{n} \right) + \sum_{j=1}^{k_{b^*}} n_j^{b^*} \log \left( \frac{n_j^{b^*}}{n} \right)} \quad (53)$$

where  $C_i^a$  is a cluster of  $P^a$  partitioning;  $P^{b^*}$  is a partitioning;  $n$  denotes the number of objects in the dataset; and  $k_{b^*}$  is the number of clusters in  $P^{b^*}$ . Additionally, a new method is employed to construct the co-association matrix, which is on the basis of the Evidence Accumulation Clustering (EAC) algorithm. In this method, each entry of the co-association matrix is calculated as follows:

$$C(i, j) = \frac{n_{ij}}{\max(n_i, n_j)} \quad (54)$$

where  $n_i$  and  $n_j$  stand for the number of occurrences of the  $i$ th and  $j$ th samples in the selected or remaining clusters, and  $n_{ij}$  signifies the number of remaining clusters that contain  $i$ th and  $j$ th samples. Finally, the average-link method is utilized in order to obtain the final solution.

**Hierarchical cluster ensemble selection (HCES):** A hierarchical cluster ensemble selection method was introduced by Akbari et al. (2015) in which three techniques of average-linkage, single-linkage, and complete-linkage agglomerative are applied to choosing a subset of cluster members. HCES identifies the subset of cluster members with considering both diversity and quality. Additionally, a novel relative diversity measure was introduced in their paper as expressed in Eq. (55):

$$\text{diversity}(h_i, h_j) = |\text{quality}(h^*, h_i) - \text{quality}(h^*, h_j)| \quad (55)$$

where  $h_i$  and  $h_j$  are clusterings, and  $h^*$  is the reference consensus partition. The relative diversity measure computes the absolute distance between qualities of  $h_i$  and  $h_j$  w.r.t  $h^*$ . After the cluster ensemble selection phase, CSPA and HGPA were implemented in HCES aiming to obtain the final solution.

**CES with constraints:** Yang et al. (2017) proposed a greedy algorithm applicable to CES. They considered three factors of diversity, quality, and consistency in their selection strategy. Must-Link(ML) and Cannot-Link(CL) constraints were produced for each pair of objects to define the consistency. Finally, clusters were selected using a scoring function (Eq. (56)):

$$\begin{aligned} & \text{Score}(C^j) \\ &= \begin{cases} \text{consistency}(C^j) + \text{quality}(C^j) & |E| = 0, \\ \text{consistency}(C^j) + \text{quality}(C^j) + \frac{1}{|E|} \sum_{C^i \in E} \text{diversity}(C^j, C^i) & \text{else}. \end{cases} \end{aligned} \quad (56)$$

where  $C^j$  is a clustering, and  $E$  is the selected solution space. By applying a consensus function to the ensemble, the final partition was obtained.

**Transfer CES (TCES):** An algorithm was proposed by Shi et al. (2018) combining the idea of transfer learning and CES in a way to be used in the selection of clustering members. Three objective functions on the basis of the relationships between quality and diversity were suggested, and a multi-objective self-evolutionary process (MOSEP) was developed for the optimization of the clustering members selection problem. First, MOSEP applies two operations on base clusterings to generate two new subsets from original base clusterings. these two operations are two-segment local adjustment and two-segment global replacement. Afterwards, the original base clusterings and these two new subsets are combined. After that, MOSEP makes use of the fast non-dominated sorting algorithm (Deb et al., 2000) in order to choose the members with better performance. A consensus ensemble method based on a hypergraph is then employed in order to obtain the ensemble result.

**Multi-modal metrics selective clustering ensemble (MMSCE):** Wang and Liu (2018) suggested a novel selection strategy which select

**Table 4**

Comparison of CES methods.

CES method	Diversity approach			Diversity measure	Size of selected ensemble	Algorithm used to generate BC	Consensus function
	Pairwise	Non-pairwise	Hybrid				
Moderate diversity CES (Hadjitodorov et al., 2006)	✓	✓		ARI	Fixed	k-means, hierarchical clustering algorithm	k-means
CAS (Fern and Lin, 2008)	✓	✓		NMI	Fixed	k-means	CSPA
Adaptive CES (Azimi and Fern, 2009b)		✓		NMI	Automatic	k-means, Maximal similar features	HAC-AL
Resampling-based CES (Hong et al., 2009)	✓	✓		ARI	Fixed	k-means	CSPA, HGPA, MCLA
SELSCE (Jia et al., 2011)		✓		NMI, ARI	Fixed	Spectral clustering	CSPA, MCLA
SCEBC (Lu et al., 2013)	✓			Covariance	Fixed	k-means, AP, FCM	CSPA
CES based on relative validity indexes (Naldi et al., 2013)	✓			Multiple criteria	Fixed	k-means	CSPA, HGPA, MCLA, Average linkage
HCSS (Yu et al., 2014)		✓		Squared-Error distortion, Disassociation	Fixed	k-means, Spectral clustering	Normalized cut algorithm
CES based on APMM criterion (Alizadeh et al., 2014a)		✓		APMM	Fixed	k-means	Average linkage
HCES (Akbari et al., 2015)	✓	✓	✓	NMI	Automatic	k-means	CSPA, HGPA
CES with constraints (Yang et al., 2017)	✓	✓		NMI	Fixed	k-means	CSPA
TCES (Shi et al., 2018)	✓		✓	NMI, ARI	Fixed	k-means, Spectral clustering	Normalized cut algorithm
MMSC (Wang and Liu, 2018)	✓	✓		NMI, CH, SI, Tanimoto	Fixed	k-means, hierarchical clustering algorithm	Single linkage
CES considering quality and diversity (Abbasi et al., 2019)		✓		NMI	Fixed	k-means	CSPA, HGPA, MCLA, Average linkage
MCAS (Ma et al., 2020)	✓			NMI	Fixed	k-means, Spectral clustering	Normalized cut algorithm

clustering partitions according to their quality and diversity using hybrid multi-modal metrics. All base clusterings are ranked using Eq. (57). Then, a subset of clustering partitions with better performance is chosen.

$$Weight(p_i) = \beta \times Quality(p_i) + (1 - \beta) \times Diversity(p_i) \quad (57)$$

where  $\beta$  signifies the weight ratio of quality; and  $p_i$  is a clustering partition. The consensus function is finally applied to the selected clusterings and the final solution is obtained.

**CES considering quality and diversity:** Abbasi et al. (2019) developed a method for choosing a subset of more effective clusters. The authors suggested a criterion based on NMI named Edited NMI (ENMI). Additionally, a new method for the formation of the co-association matrix, i.e., Extended Evidence Accumulation Clustering (EEAC), was suggested in their paper. In EEAC, each entry of the co-association matrix is calculated as follows:

$$C_{ij} = \frac{n_{ij}}{n_i + n_j} \quad (58)$$

where  $n_i$  and  $n_j$  stand for the numbers of selected clusters containing  $i$ th and  $j$ th objects, and  $n_{ij}$  denotes the number of selected clusters shared by  $i$ th and  $j$ th objects. Finally, the hierarchical clustering method is applied to extracting the consensus partition.

**Multiple clustering and selecting (MCAS):** MCAS was introduced by Ma et al. (2020) with taking into consideration different original clustering solutions and also with considering diversity and quality factors. In addition, they suggested two combining strategies, namely direct combining and clustering combining to aggregate the results selected by MCAS. At first, different clustering algorithms were used to produce ensemble members. Next, several selecting algorithms applied to obtain different subset of solutions. Afterwards, a combining strategy is used to combine these subsets in a way to obtain the selected solutions. Then, a consensus matrix  $W$  is constructed. Each entry in  $W$  is computed as follows:

$$w_{ij} = \frac{T(x_i, x_j)}{|S|} \quad (59)$$

where  $T(x_i, x_j)$  stands for the number of times the data points  $x_i$  and  $x_j$  appear together in all selected solutions, and  $|S|$  represents the

number of selected solutions. Finally, the normalized cut algorithm (Shi and Malik, 2000) was used as consensus function for the purpose of generating the final solution.

**Table 4** shows the comparison of the aforementioned approaches. As illustrated in **Table 4**, different clustering ensemble selection methods applied various algorithms to generate base clusterings and then select subset of ensemble members using pairwise, non-pairwise, or hybrid approaches based on a diversity measure. Finally, different consensus functions employed to produce the final solution.

## 6. Summary and research directions

Overall, the partitional algorithms such as k-means have low complexity but they are suitable for special datasets (convex shape). On the other hand, they produce clusters with low accuracy. On the contrary, hierarchical methods have high complexity ( $O(n^2)$ ) but they produce clusters with high accuracy. Also, these methods do not have a global objective function for optimization. Many partitional and hierarchical clustering algorithms affect the result by taking input parameters such as k-means. If these parameters chose badly then the results are bad. All of single methods focus on specific areas of data. Therefore, they have a good efficiency on specific data. Even, the selection of the best clustering methods with correct parameter values is not suitable for every case of datasets. The most sophisticated clustering methods produce low quality for large datasets with the outlier. Developing a clustering method that produces accurate solutions for every case of datasets and finds results with low complexity has remained a challenge.

Clustering ensemble approaches can effectively increase the accuracy and stability levels through combining the results of various clusterings. Clustering ensemble offers two main problems: diversity and consensus function. In knowledge reuse issues, the raw data is not available. A few number of ensemble members with low diversity that are considered as the input of consensus function lead to final solution with non-acceptable quality. Thus, generating new diversity concerning the quality using the few number of base clusterings is a challenge.

Consensus functions are sensitive to ensemble size, base clusterings quality and diversity. Some consensus functions like HGPA are more sensitive to diversity, whereas others like CSPA are more sensitive to quality. Developing a consensus function with lower sensitivity to ensemble size, quality and diversity is a challenge.

The goal of the cluster ensemble selection is improving the CE performance. However, there were some drawbacks in CES approach that are including: The size of an appropriate subset of BC is uncertain, the proposed diversity measures are loosely defined, since the labels of BC are virtual, and this approach finds the subset of a large library of BC, in this case, if the set of BC has small size, the CES cannot improve the CE result. On the other hand, consensus function is also an important factor in improving BC. The relationship between diversity and quality of base clusterings is yet uncertain.

Based on the review done in this paper, the problems in consensus function include accuracy, complexity, estimating the number of true clusters, and obtaining outlier.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This work was supported by “Human Resources Program in Energy Technology” of the Korea Institute of Energy Technology Evaluation and Planning (KETEP), South Korea, granted financial resource from the Ministry of Trade, Industry & Energy, Republic of Korea

(No. 20204010600470). This research was also supported by “Research Base Construction Fund Support Program” funded by Chon Buk National University in 2019, South Korea, and a grant (No: 2019R1A2C108690412) of National Research Foundation of Korea (NRF), South Korea funded by the Korea government (MSP).

## References

- Abbasi, S.-o., Nejatian, S., Parvin, H., Rezaie, V., Bagherifard, K., 2019. Clustering ensemble selection considering quality and diversity. *Artif. Intell. Rev.* 52 (2), 1311–1340.
- Aggarwal, C.C., 2001. Re-designing distance functions and distance-based applications for high dimensional data. *ACM SIGMOD Rec.* 30 (1), 13–18.
- Agrawal, A.K., Goldfarb, A., 2006. Restructuring Research: Communication Costs and the Democratization of University Innovation. Technical Report, National Bureau of Economic Research.
- Akbari, E., Dahlan, H.M., Ibrahim, R., Alizadeh, H., 2015. Hierarchical cluster ensemble selection. *Eng. Appl. Artif. Intell.* 39, 146–156.
- Alizadeh, H., Minaei-Bidgoli, B., Parvin, H., 2014a. Cluster ensemble selection based on a new cluster stability measure. *Intell. Data Anal.* 18 (3), 389–408.
- Alizadeh, H., Minaei-Bidgoli, B., Parvin, H., 2014b. To improve the quality of cluster ensembles by selecting a subset of base clusters. *J. Exp. Theor. Artif. Intell.* 26 (1), 127–150.
- Arbelaez, P., Maire, M., Fowlkes, C., Malik, J., 2011. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 33 (5), 898–916.
- Ayad, H.G., Kamel, M.S., 2008. Cumulative voting consensus method for partitions with variable number of clusters. *IEEE Trans. Pattern Anal. Mach. Intell.* 30 (1), 160–173.
- Ayad, H.G., Kamel, M.S., 2010. On voting-based consensus of cluster ensembles. *Pattern Recognit.* 43 (5), 1943–1953.
- Azimi, J., Fern, X., 2009a. Adaptive cluster ensemble selection. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence, Vol. 9. Morgan Kaufmann Publishers Inc., pp. 992–997.
- Azimi, J., Fern, X., 2009b. Adaptive cluster ensemble selection. In: International Joint Conferences on Artificial Intelligence, Vol. 9, pp. 992–997.
- Bae, E., Bailey, J., 2006. CoalA: A novel approach for the extraction of an alternate clustering of high quality and high dissimilarity. In: Sixth International Conference on Data Mining (ICDM). IEEE, pp. 53–62.
- Bagherinia, A., Minaei-Bidgoli, B., Hosseinzadeh, M., Parvin, H., 2020. Reliability-based fuzzy clustering ensemble. *Fuzzy Sets and Systems*.
- Bailey, T.L., Elkan, C., et al., 1994. Fitting a mixture model by expectation maximization to discover motifs in bipolymers. In: Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology, Vol. 2, pp. 28–36.
- Barthelemy, J.-P., Brucker, F., 2001. NP-Hard approximation problems in overlapping clustering. *J. Classification* 18 (2), 159–183.
- Beauchemin, M., 2015. A density-based similarity matrix construction for spectral clustering. *Neurocomputing* 151, 835–844.
- Berikov, V., 2014. Weighted ensemble of algorithms for complex data clustering. *Pattern Recognit. Lett.* 38, 99–106.
- Berikov, V., Pestunov, I., 2017. Ensemble clustering based on weighted co-association matrices: Error bound and convergence properties. *Pattern Recognit.* 63, 427–436.
- Bezdek, J.C., 1981. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers.
- Bobrowski, L., Bezdek, J.C., 1991. C-means clustering with the  $l_1$  and  $l_\infty$  norms. *IEEE Trans. Syst. Man Cybern.* 21 (3), 545–554.
- Brown, R.D., Martin, Y.C., 1996. Use of structure-activity data to compare structure-based clustering methods and descriptors for use in compound selection. *J. Chem. Inf. Comput. Sci.* 36 (3), 572–584.
- Caruana, R., Elhaway, M., Nguyen, N., Smith, C., 2006. Meta clustering. In: Sixth International Conference on Data Mining, ICDM. IEEE, pp. 107–118.
- Chakraborty, T., Pierazzi, F., Subrahmanian, V., 2017. Ec2: Ensemble clustering and classification for predicting android malware families. *IEEE Trans. Dependable Secure Comput.* 17 (2), 262–277.
- Correa-Morris, J., 2013. An indication of unification for different clustering approaches. *Pattern Recognit.* 46 (9), 2548–2561.
- Davies, D.L., Bouldin, D.W., 1979. A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-1 (2), 224–227.
- Deb, K., Agrawal, S., Pratap, A., Meyarivan, T., 2000. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: International Conference on Parallel Problem Solving from Nature. Springer, pp. 849–858.
- Domeniconi, C., Gunopulos, D., Ma, S., Yan, B., Al-Razgan, M., Papadopoulos, D., 2007. Locally adaptive metrics for clustering high dimensional data. *Data Min. Knowl. Discov.* 14 (1), 63–97.
- Drineas, P., Frieze, A., Kannan, R., Vempala, S., Vinay, V., 2004. Clustering large graphs via the singular value decomposition. *Mach. Learn.* 56 (1–3), 9–33.
- Dudoit, S., Fridlyand, J., 2003. Bagging to improve the accuracy of a clustering procedure. *Bioinformatics* 19 (9), 1090–1099.
- Dunn, J.C., 1973. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *J. Cybern.* 3 (3), 32–57.

- Dunn, J.C., 1974. Well-separated clusters and optimal fuzzy partitions. *J. Cybern.* 4 (1), 95–104.
- Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D., 1998. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci.* 95 (25), 14863–14868.
- Elghazel, H., Aussem, A., 2013. Unsupervised feature selection with ensemble learning. *Mach. Learn.* 1–24.
- Faceli, K., De Carvalho, A.C., De Souto, M.C., 2007. Multi-objective clustering ensemble. *Int. J. Hybrid Intell. Syst.* 4 (3), 145–156.
- Fern, X.Z., Brodley, C.E., 2003. Random projection for high dimensional data clustering: A cluster ensemble approach. In: Proceedings of the Twentieth International Conference on International Conference on Machine Learning, Vol. 3, pp. 186–193.
- Fern, X.Z., Brodley, C.E., 2003. Random projection for high dimensional data clustering: A cluster ensemble approach. In: Proceeding of the 20th International Conference on Machine Learning (ICML), Vol. 3, pp. 186–193.
- Fern, X.Z., Brodley, C.E., 2004. Solving cluster ensemble problems by bipartite graph partitioning. In: Proceedings of the Twenty-First International Conference on Machine Learning. ACM, p. 36.
- Fern, X.Z., Lin, W., 2008. Cluster ensemble selection. *Stat. Anal. Data Min.* 1 (3), 128–141.
- Fischer, B., Buhmann, J.M., 2003. Bagging for path-based clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* 25 (11), 1411–1415.
- Franek, L., Jiang, X., 2014. Ensemble clustering by means of clustering embedding in vector spaces. *Pattern Recognit.* 47 (2), 833–842.
- Fred, A.L., Jain, A.K., 2002. Data clustering using evidence accumulation. In: Object Recognition Supported By User Interaction for Service Robots, Vol. 4. IEEE, pp. 276–280.
- Fred, A.L., Jain, A.K., 2005. Combining multiple clusterings using evidence accumulation. *IEEE Trans. Pattern Anal. Mach. Intell.* 27 (6), 835–850.
- Ghosh, J., Acharya, A., 2011. Cluster ensembles. *Wiley Interdiscip. Rev.: Data Min. Knowl. Discov.* 1 (4), 305–315.
- Gionis, A., Mannila, H., Tsaparas, P., 2007. Clustering aggregation. *ACM Trans. Knowl. Discov. Data (TKDD)* 1 (1), 4–es.
- Guénoche, A., Hansen, P., Jaumard, B., 1991. Efficient algorithms for divisive hierarchical clustering with the diameter criterion. *J. Classification* 8 (1), 5–30.
- Hadjitodorov, S.T., Kuncheva, L.I., Todorova, L.P., 2006. Moderate diversity for better cluster ensembles. *Inf. Fusion* 7 (3), 264–275.
- Hagen, L., Kahng, A.B., 1992. New spectral methods for ratio cut partitioning and clustering. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* 11 (9), 1074–1085.
- Hamidi, S.S., Akbari, E., Motameni, H., 2019. Consensus clustering algorithm based on the automatic partitioning similarity graph. *Data Knowl. Eng.* 124, 101754.
- Han, E.-H., Karypis, G., Kumar, V., Mobasher, B., 1997. Clustering Based on Association Rule Hypergraphs. University of Minnesota, Department of Computer Science.
- Han, J., Pei, J., Kamber, M., 2011. Data Mining: Concepts and Techniques. Elsevier.
- Hinneburg, A., Aggarwal, C.C., Keim, D.A., 2000. What is the nearest neighbor in high dimensional spaces? In: Proceedings of the 26th International Conference on Very Large Data Bases. Morgan Kaufmann Publishers Inc., pp. 506–515.
- Hong, Y., Kwong, S., Chang, Y., Ren, Q., 2008. Unsupervised feature selection using clustering ensembles and population based incremental learning algorithm. *Pattern Recognit.* 41 (9), 2742–2756.
- Hong, Y., Kwong, S., Wang, H., Ren, Q., 2009. Resampling-based selective clustering ensembles. *Pattern Recognit. Lett.* 30 (3), 298–305.
- Houle, M.E., Kriegel, H.-P., Kröger, P., Schubert, E., Zimek, A., 2010. Can shared-neighbor distances defeat the curse of dimensionality? In: Scientific and Statistical Database Management. Springer, pp. 482–500.
- Huang, Z., 1997a. Clustering large data sets with mixed numeric and categorical values. In: Proceedings of the 1st Pacific-Asia Conference on Knowledge Discovery and Data Mining, (PAKDD), Singapore, pp. 21–34.
- Huang, Z., 1997b. A fast clustering algorithm to cluster very large categorical data sets in data mining. *DMKD* 3 (8), 34–39.
- Huang, D., Wang, C.-D., Lai, J.-H., 2017. LWMC: A locally weighted meta-clustering algorithm for ensemble clustering. In: International Conference on Neural Information Processing. Springer, pp. 167–176.
- Huang, D., Wang, C.-D., Lai, J.-H., 2018. Locally weighted ensemble clustering. *IEEE Trans. Cybern.* 48 (5), 1460–1473.
- Hubert, L., Arabie, P., 1985. Comparing partitions. *J. Classification* 2 (1), 193–218.
- Iam-On, N., Boongoen, T., Garrett, S., 2008. Refining pairwise similarity matrix for cluster ensemble problem with cluster relations. In: International Conference on Discovery Science. Springer, pp. 222–233.
- Iam-on, N., Boongoen, T., Garrett, S., 2010. LCE: a link-based cluster ensemble method for improved gene expression data analysis. *Bioinformatics* 26 (12), 1513–1519.
- Jackowski, K., 2018. New diversity measure for data stream classification ensembles. *Eng. Appl. Artif. Intell.* 74, 23–34.
- Jain, A.K., 2010. Data clustering: 50 years beyond K-means. *Pattern Recognit. Lett.* 31 (8), 651–666.
- Jain, A.K., Dubes, R.C., et al., 1988. Algorithms for Clustering Data, Vol. 6. Prentice hall Englewood Cliffs.
- Jain, A.K., Murty, M.N., Flynn, P.J., 1999. Data clustering: a review. *ACM Comput. Surv.* 31 (3), 264–323.
- Jia, J., Xiao, X., Liu, B., Jiao, L., 2011. Bagging-based spectral clustering ensemble selection. *Pattern Recognit. Lett.* 32 (10), 1456–1467.
- Karypis, G., Kumar, V., 1998. Multilevel k-way partitioning scheme for irregular graphs. *J. Parallel Distrib. Comput.* 48 (1), 96–129.
- Kaufman, L., Rousseeuw, P.J., 2009. Finding Groups in Data: An Introduction to Cluster Analysis, Vol. 344. John Wiley & Sons.
- Khedairia, S., Khadir, M.T., 2019. A multiple clustering combination approach based on iterative voting process. *J. King Saud Univ.-Comput. Inf. Sci.*
- King, B., 1967. Step-wise clustering procedures. *J. Amer. Statist. Assoc.* 62 (317), 86–101.
- Kleinberg, J., 2002. An impossibility theorem for clustering. In: Proceedings of the 15th International Conference on Neural Information Processing Systems, Vol. 15, pp. 463–470.
- Kuhn, H.W., 1955. The hungarian method for the assignment problem. *Nav. Res. Logist. Q.* 2 (1–2), 83–97.
- Kuncheva, L.I., 2005. Diversity in multiple classifier systems. *Inf. Fusion* 6 (1), 3–4.
- Kuncheva, L.I., Hadjitodorov, S.T., 2004. Using diversity in cluster ensembles. In: International Conference on Systems, Man and Cybernetics, Vol. 2. IEEE, pp. 1214–1219.
- Larsen, B., Aone, C., 1999. Fast and effective text mining using linear-time document clustering. In: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 16–22.
- Law, M.H., Topchy, A.P., Jain, A.K., 2004. Multiobjective data clustering. In: Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, Vol. 2. IEEE, pp. 418–424.
- Levine, E., Domany, E., 2001. Resampling method for unsupervised estimation of cluster validity. *Neural Comput.* 13 (11), 2573–2593.
- Li, H., Liu, X., Li, T., Gan, R., 2020. A novel density-based clustering algorithm using nearest neighbor graph. *Pattern Recognit.* 102, 107206.
- Li, F., Qian, Y., Wang, J., Dang, C., Jing, L., 2019a. Clustering ensemble based on sample's stability. *Artificial Intelligence* 273, 37–55.
- Li, F., Qian, Y., Wang, J., Dang, C., Jing, L., 2019b. Clustering ensemble based on sample's stability. *Artificial Intelligence* 273, 37–55.
- Li, F., Qian, Y., Wang, J., Liang, J., 2017. Multigranulation information fusion: A Dempster-Shafer evidence theory-based clustering ensemble method. *Inform. Sci.* 378, 389–409.
- Li, Z., Wu, X.-M., Chang, S.-F., 2012. Segmentation using superpixels: A bipartite graph partitioning approach. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 789–796.
- Li, Y., Yu, J., Hao, P., Li, Z., 2007. Clustering ensembles based on normalized edges. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, pp. 664–671.
- Lim, N.J.S., Durrant, R.J., 2020. A diversity-aware model for majority vote ensemble accuracy. In: 23rd International Conference on Artificial Intelligence and Statistics (AISTATS), Vol. 108. Addison-Wesley, pp. 4078–4086.
- Liu, H.Q., Zhang, Q., Zhao, F., 2018. Interval fuzzy spectral clustering ensemble algorithm for color image segmentation. *J. Intell. Fuzzy Systems* 35 (5), 5467–5476.
- Lu, X., Yang, Y., Wang, H., 2013. Selective clustering ensemble based on covariance. In: Multiple Classifier Systems. Springer, pp. 179–189.
- Ma, T., Yu, T., Wu, X., Cao, J., Al-Abdulkarim, A., Al-Dhelaan, A., Al-Dhelaan, M., 2020. Multiple clustering and selecting algorithms with combining strategy for selective clustering ensemble. *Soft Comput.* 1–13.
- MacQueen, J., et al., 1967. Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1, Oakland, CA, USA, pp. 281–297.
- Manning, C.D., Raghavan, P., Schütze, H., et al., 2008. Introduction to Information Retrieval, Vol. 1. Cambridge university press.
- Mimaroglu, S., Erdil, E., 2011. Combining multiple clusterings using similarity graph. *Pattern Recognit.* 44 (3), 694–703.
- Mimaroglu, S., Erdil, E., 2013a. An efficient and scalable family of algorithms for combining clusterings. *Eng. Appl. Artif. Intell.* 26 (10), 2525–2539.
- Mimaroglu, S., Erdil, E., 2013b. An efficient and scalable family of algorithms for combining clusterings. *Eng. Appl. Artif. Intell.* 26 (10), 2525–2539.
- Mimaroglu, S., Yagci, M., 2012. CLICOM: Cliques for combining multiple clusterings. *Expert Syst. Appl.* 39 (2), 1889–1901.
- Minaei-Bidgoli, B., Parvin, H., Alinejad-Rokny, H., Alizadeh, H., Punch, W.F., 2014. Effects of resampling method and adaptation on clustering ensemble efficacy. *Artif. Intell. Rev.* 41 (1), 27–48.
- Minaei-Bidgoli, B., Topchy, A., Punch, W.F., 2004. Ensembles of partitions via data resampling. In: Proceedings of International Conference on Information Technology: Coding and Computing, 2004. ITCC 2004., Vol. 2. IEEE, pp. 188–192.
- Monti, S., Tamayo, P., Mesirov, J., Golub, T., 2003. Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Mach. Learn.* 52 (1–2), 91–118.
- Naldi, M., Carvalho, A., Campello, R., 2013. Cluster ensemble selection based on relative validity indexes. *Data Min. Knowl. Discov.* 27 (2), 259–289.
- Niu, H., Khozouie, N., Parvin, H., Alinejad-Rokny, H., Beheshti, A., Mahmoudi, M.R., 2020. An ensemble of locally reliable cluster solutions. *Appl. Sci.* 10 (5), 1891.
- Olson, C.F., 1995. Parallel algorithms for hierarchical clustering. *Parallel Comput.* 21 (8), 1313–1325.
- Pakhira, M.K., Bandyopadhyay, S., Maulik, U., 2004. Validity index for crisp and fuzzy clusters. *Pattern Recognit.* 37 (3), 487–501.

- Rafiee, G., Dray, S.S., Woo, W.L., 2013. Region-of-interest extraction in low depth of field images using ensemble clustering and difference of Gaussian approaches. *Pattern Recognit.* 46 (10), 2685–2699.
- Ramasso, E., Placet, V., Boubakar, M.L., 2015. Unsupervised consensus clustering of acoustic emission time-series for robust damage sequence estimation in composites. *IEEE Trans. Instrum. Meas.* 64 (12), 3297–3307.
- Rashedi, E., Mirzaei, A., 2013. A hierarchical clusterer ensemble method based on boosting theory. *Knowl.-Based Syst.* 45, 83–93.
- Saeed, F., Salim, N., Abdo, A., 2012. Voting-based consensus clustering for combining multiple clusterings of chemical structures. *J. Cheminform.* 4 (1), 37.
- Sarkar, J.P., Saha, I., Maulik, U., 2019. Improved fuzzy clustering using ensemble based differential evolution for remote sensing image. In: TENCON 2019-2019 IEEE Region 10 Conference (TENCON). IEEE, pp. 880–885.
- Sesmero, M.P., Alonso-Weber, J.M., Giuliani, A., Armano, G., Sanchis, A., 2018. Measuring diversity and accuracy in ANN ensembles. In: Conference of the Spanish Association for Artificial Intelligence. Springer, pp. 108–117.
- Sharma, K.K., Seal, A., 2020. Clustering analysis using an adaptive fused distance. *Eng. Appl. Artif. Intell.* 96, 103928.
- Shi, J., Malik, J., 2000. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 22 (8), 888–905.
- Shi, Y., Yu, Z., Chen, C.P., You, J., Wong, H.-S., Wang, Y., Zhang, J., 2018. Transfer clustering ensemble selection. *IEEE Trans. Cybern.*
- Sibson, R., 1973. SLINK: an optimally efficient algorithm for the single-link cluster method. *Comput. J.* 16 (1), 30–34.
- Sinaga, K.P., Yang, M.-S., 2020. Unsupervised K-means clustering algorithm. *IEEE Access* 8, 80716–80727.
- Sirbu, A., Kerr, G., Crane, M., Ruskin, H.J., 2012. RNA-Seq vs dual-and single-channel microarray data: sensitivity analysis for differential expression and clustering. *PLoS One* 7 (12), e50986.
- Sneath, P.H., Sokal, R.R., et al., 1973. Numerical Taxonomy. In: The Principles and Practice of Numerical Classification, W H Freeman & Co.
- Stoltz, T., Huertas, M.E., Mendoza, A., 2020. Assessment of air quality monitoring networks using an ensemble clustering method in the three major metropolitan areas of Mexico. *Atmos. Pollut. Res.* 11 (8), 1271–1280.
- Strehl, A., Ghosh, J., 2003. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.* 3, 583–617.
- Tan, T.Y., Zhang, L., Lim, C.P., 2020. Adaptive melanoma diagnosis using evolving clustering, ensemble and deep neural networks. *Knowl.-Based Syst.* 187, 104807.
- Topchy, A., Jain, A.K., Punch, W., 2003. Combining multiple weak clusterings. In: Third IEEE International Conference on Data Mining, ICDM. IEEE, pp. 331–338.
- Topchy, A.P., Jain, A.K., Punch, W.F., 2004a. A mixture model for clustering ensembles. In: Proceedings of the Fourth SIAM International Conference on Data Mining (SDM). SIAM, pp. 379–390.
- Topchy, A., Jain, A.K., Punch, W., 2005. Clustering ensembles: Models of consensus and weak partitions. *IEEE Trans. Pattern Anal. Mach. Intell.* 27 (12), 1866–1881.
- Topchy, A., Minaei-Bidgoli, B., Jain, A.K., Punch, W.F., 2004b. Adaptive clustering ensembles. In: Proceedings of the 17th International Conference on Pattern Recognition, ICPR, Vol. 1. IEEE, pp. 272–275.
- Tumer, K., Agogino, A.K., 2008. Ensemble clustering with voting active clusters. *Pattern Recognit. Lett.* 29 (14), 1947–1953.
- Vega-Pons, S., Ruiz-Shulcloper, J., 2011. A survey of clustering ensemble algorithms. *Int. J. Pattern Recognit. Artif. Intell.* 25 (03), 337–372.
- Wang, X., Han, D., Han, C., 2013. Rough set based cluster ensemble selection. In: Proceedings of the 16th International Conference on Information Fusion. IEEE, pp. 438–444.
- Wang, H., Liu, G., 2018. Two-level-oriented selective clustering ensemble based on hybrid multi-modal metrics. *IEEE Access* 6, 64159–64168.
- Wang, X., Yang, C., Zhou, J., 2009. Clustering aggregation by probability accumulation. *Pattern Recognit.* 42 (5), 668–675.
- Wu, X., Ma, T., Cao, J., Tian, Y., Alabdulkarim, A., 2018. A comparative study of clustering ensemble algorithms. *Comput. Electr. Eng.* 68, 603–615.
- Wu, M., Zhao, Y., Zhang, L., Wang, J., Xu, H., Wei, D., 2017. Ensemble clustering model of hyperspectral image segmentation. In: 2017 9th International Conference on Advanced Infocomm Technology (ICAIT). IEEE, pp. 356–360.
- Xiao, W., Yang, Y., Wang, H., Li, T., Xing, H., 2016. Semi-supervised hierarchical clustering ensemble and its application. *Neurocomputing* 173, 1362–1376.
- Yang, F., Li, X., Li, Q., Li, T., 2014. Exploring the diversity in cluster ensemble generation: Random sampling and random projection. *Expert Syst. Appl.* 41 (10), 4844–4866.
- Yang, F., Li, T., Zhou, Q., Xiao, H., 2017. Cluster ensemble selection with constraints. *Neurocomputing* 235, 59–70.
- Yao, T., Liu, C., Deng, Z., Liu, X., Liu, J., 2017. Adaptive ensemble clustering for image segmentation in remote sensing. In: International Conference in Communications, Signal Processing, and Systems. Springer, pp. 1608–1613.
- Ye, M., Liu, W., Wei, J., Hu, X., 2016. Fuzzy-means and cluster ensemble with random projection for big data clustering. *Math. Probl. Eng.* 2016.
- Yi, J., Yang, T., Jin, R., Jain, A.K., Mahdavi, M., 2012. Robust ensemble clustering by matrix completion. In: 2012 IEEE 12th International Conference on Data Mining. IEEE, pp. 1176–1181.
- Yousefnezhad, M., Reihanian, A., Zhang, D., Minaei-Bidgoli, B., 2016. A new selection strategy for selective cluster ensemble based on diversity and independency. *Eng. Appl. Artif. Intell.* 56, 260–272.
- Yu, Z., Chen, H., You, J., Han, G., Li, L., 2013. Hybrid fuzzy cluster ensemble framework for tumor clustering from biomolecular data. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 10 (3), 657–670.
- Yu, Z., Li, L., Gao, Y., You, J., Liu, J., Wong, H.-S., Han, G., 2014. Hybrid clustering solution selection strategy. *Pattern Recognit.* 47 (10), 3362–3375.
- Yu, Z., Wong, H.-S., 2009. Class discovery from gene expression data based on perturbation and cluster ensemble. *IEEE Trans. NanoBiosci.* 8 (2), 147–160.
- Yu, Z., Wong, H.-S., Wang, H., 2007. Graph-based consensus clustering for class discovery from gene expression data. *Bioinformatics* 23 (21), 2888–2896.
- Zadeh, R.B., Ben-David, S., 2009. A uniqueness theorem for clustering. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. AUAI Press, pp. 639–646.
- Zarikas, V., Poulopoulos, S.G., Gareiou, Z., Zervas, E., 2020. Clustering analysis of countries using the COVID-19 cases dataset. *Data Brief* 31, 105787.
- Zhang, Y., Rong, C., Huang, Q., Wu, Y., Yang, Z., Jiang, J., 2017. Based on multi-features and clustering ensemble method for automatic malware categorization. In: 2017 IEEE Trustcom/BigDataSE/ICESS. IEEE, pp. 73–82.
- Zhao, W., Liu, H., Dai, W., Ma, J., 2016. An entropy-based clustering ensemble method to support resource allocation in business process management. *Knowl. Inf. Syst.* 48 (2), 305–330.
- Zhou, P., Du, L., Liu, X., Shen, Y.-D., Fan, M., Li, X., 2020. Self-paced clustering ensemble. *IEEE Trans. Neural Netw. Learn. Syst.*