Computer Science Review 28 (2018) 1–25

Contents lists available at ScienceDirect

Computer Science Review

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

Review article

Cluster ensembles: A survey of approaches with recent extensions and applications

Tossapon Boongoen, Natthakan Iam-On [*]

IQ-D Research Unit, School of Information Technology, Mae Fah Luang University, Tasud, Muang District, Chiang Rai 57100, Thailand

ARTICLE INFO

Article history:
Received 10 April 2017
Received in revised form 26 December 2017
Accepted 29 January 2018
Available online 10 February 2018

Keywords:
Data clustering
Cluster ensemble
Theoretical extension
Domain specific application

ABSTRACT

Cluster ensembles have been shown to be better than any standard clustering algorithm at improving accuracy and robustness across different data collections. This meta-learning formalism also helps users to overcome the dilemma of selecting an appropriate technique and the corresponding parameters, given a set of data to be investigated. Almost two decades after the first publication of a kind, the method has proven effective for many problem domains, especially microarray data analysis and its downstreaming applications. Recently, it has been greatly extended both in terms of theoretical modelling and deployment to problem solving. The survey attempts to match this emerging attention with the provision of fundamental basis and theoretical details of state-of-the-art methods found in the present literature. It yields the ranges of ensemble generation strategies, summarization and representation of ensemble members, as well as the topic of consensus clustering. This review also includes different applications and extensions of cluster ensemble, with several research issues and challenges being highlighted.

Contents

1. Introduction

Cluster analysis is usually employed in the initial stage of understanding a raw data, especially for new problems where prior

[*] Corresponding author.
E-mail addresses: tossapon.boo@mfu.ac.th (T. Boongoen), natthakan@mfu.ac.th (N. Iam-On).

https://doi.org/10.1016/j.cosrev.2018.01.003
1574-0137/

knowledge is minimal. Also, in the pre-processing stage of supervised learning, it is exploited to identify outliers and possible object classes for the following expert-directed labelling process. This is crucial when the complexity of modern-age information is generally overwhelming for a human investigation. The need to acquire knowledge or learn from the excessive amount of data is hence a major driving force for making clustering a highly active research subject. Data clustering is applied to a variety of problem domains such as biology [1], customer relationship management [2], information retrieval [3,4], image processing [5,6], marketing [7,8], psychology [9] and recommender systems [10]. In addition, the recent development of clustering cancer gene expression data has attracted a lot of interests amongst computer scientists, biological and clinical researchers [11–13].

Principally, the core of cluster analysis is the clustering process which divides data objects into groups or clusters such that objects in the same cluster are more similar to each other than to those belonging to different clusters [14]. Objects under examination are normally described in terms of object-specific (e.g., attribute values) or relative measurements (e.g., pairwise dissimilarity). Unlike supervised learning to which classification is categorized, clustering is ‘unsupervised’ and does not require class information, which is typically achieved through a manual tagging of category labels on data objects, by a domain expert (or through the consensus of multiple experts). Given its potential, a large number of research studies focus on several aspects of cluster analysis: for instance, clustering algorithms and extensions for particular data type [15], dissimilarity (or distance) metric [16], optimal cluster numbers [17], relevance of data attributes per cluster or subspace clustering [18], evaluation of clustering results [19], and cluster ensembles [20].

Specific to this survey, the practice of cluster ensembles is motivated by the fact that the performance of most clustering techniques are highly data dependent. A particular clustering model may produce an acceptable result for one dataset, but possibly become ineffective for others. Generally, there are two major challenges inherent to clustering algorithms. First, different techniques discover different structures (e.g., cluster size and shape) from the same set of data objects [21–23]. For example, *k*-means that is probably the best known technique is suitable for spherical-shape clusters, while single-linkage hierarchical clustering is effective to detect connected patterns. This is due to the fact that each individual algorithm is designed to optimize a specific criterion. Second, a single clustering algorithm with different parameter settings can also reveal various structures on the same dataset. A specific setting may be good for a few, but not all datasets. Users encounter these challenges, which consequently make the selection of a proper clustering technique very difficult.

A solution to this dilemma remains an ultimate goal. In order to accomplish this, researchers invented the methodology of combining different clusterings into a single consensus clustering. This process which is widely known as ‘cluster ensembles’ can provide more robust and stable solutions across different domains and datasets [20,22,24]. However, modelling a mechanism (usually referred to as a ‘consensus function’) that is effective for integrating multiple data partitions in a cluster ensemble is far from trivial. This task is difficult since there is no well defined correspondence between the different clustering results. The further challenges arising from the need to combine data partitions and generate a better clustering result without prior knowledge are of high interest amongst researchers.

The rest of this survey is organized as follows. To set the scene for concepts and discussion presented here, Section 2 introduces the basis of cluster ensembles, including formal definition, framework and different ensemble generation strategies. Then, four major approaches to find a consensus clustering are illustrated in Section 3. In addition, Section 4 provides applications and recent theoretical extensions of those cluster ensemble techniques, especially the use of ensemble information as a data transformation approach for classification task. The survey is concluded in Section 5 with future research directions.

2. The problem of cluster ensembles

This paper first presents the fundamental concepts of data clustering including a number of benchmark algorithms that have been employed for various problem domains. Each of these conventional techniques are designed on a particular assumption(s), which is normally realized via input parameters. Generally, there is no clustering algorithm, or the algorithm with distinct parameter settings, that performs well for every set of data. To overcome the difficulty with identifying a proper alternative, the methodology of cluster ensemble which is the focus of this review has been continuously developed in the past decade. The second part of this section includes details of general framework and an overview of cluster ensemble methods found in the literature.

2.1. Data clustering and conventional techniques

Data clustering is one of the fundamental and effective tools for understanding the structure of a given dataset. It plays a crucial, foundational role in machine learning, data mining, information retrieval and pattern recognition. Clustering aims to categorize data into groups or clusters such that the data in the same cluster are more similar to each other than to those in different clusters. Similarity or proximity is measured using the attribute values that represent objects (data points) in the dataset [14]. Clustering is branded an unsupervised learning approach as the measurement of similarity is conducted without knowledge of class assignment. This knowledge-free scenario brings about a series of difficult decisions, hence the corresponding research studies, with respect to selecting appropriate algorithm, similarity measure, criterion function, and initial parameter condition [21,23]. Clustering is widely recognized as an ideal candidate for research and development [25], given its benefits and possible advances to be made in this field. There are a large number of clustering algorithms developed in the literature. Examples of well-known techniques are explained in this section.

***k*-means** is perhaps, the best known clustering technique that partitions data points into clusters. Its name comes from representing each of *k* clusters by the mean of its members or so-called ‘centroid’. *k*-means is an iterative algorithm that exploits a square-error as a criterion function (i.e., the total distance between each data point and its cluster centre, [26]). It begins with initializing centroids randomly and then allocates data points to clusters such that the square-error is minimized. This criterion function tends to work well with separated and compact clusters. Given a dataset X, the square-error e^2 of a clustering $\pi = \{C_i, \ldots, C_k\}$ with k clusters is defined as

$$e^2(X, \pi) = \sum_{p=1}^{k} \sum_{\forall x \in C_p} \|x - \overline{c_p}\|^2, \tag{1}$$

where $\|.\|$ denotes the Euclidean norm and $\overline{c_p}$ is the centre of the pth cluster. A general description of the *k*-means algorithm is given as follows:

1. *k* data points are first randomly selected as initial cluster centres.
2. Repeat:
   (a) Assign each data point to its closest cluster centre. The Euclidean metric is commonly used to compute the distance between data points and centroids.

  (b) The centroid of each cluster is updated as the mean of all current data points in that cluster.

3. Until the termination criteria are met.

The examples of termination criteria are: (i) no changes are made to the cluster centres (i.e., no reassignment of any data point from one cluster to another), (ii) the maximum number of iterations is exceeded, and (iii) there is no improvements in the objective function such as decrease in the square-error. The k-means algorithm is popular largely due to its efficiency, with time complexity of $O(Nkr)$, where N is the number of data points, k is the number of clusters and r is the number of iterations. However, it is sensitive to the choices of initial cluster centres (i.e., different initial states can lead to different output partitions). One might have to run the algorithm multiple times with various initial partitions and chooses the resulting clustering that offers the minimum square-error. Yet, k-means does not work well on noisy data and non-convex cluster shapes.

***k*-modes** is introduced by Huang [27] as an extension of the conventional k-means technique for clustering categorical data. It iteratively refines k cluster representatives, each as a vector of attribute values that has the minimal distance to all the data points in a cluster (i.e., the cluster's most frequent attribute values). k-modes uses a simple similarity measure that is determined by the number of common categorical attributes shared by two data points.

Formally, let X be a set of N data points $\{x_1, \ldots, x_N\}$ described by D categorical attributes, i.e., $x_i = (x_{i1}, \ldots, x_{iD}), i = 1 \ldots N$. The distance $d(x_i, \overline{c_p})$ between data point x_i and centroid of the pth cluster is defined by

$$d(x_i, \overline{c_p}) = \sum_{j=1}^{D} \delta(x_{ij}, \overline{c_{pj}}), \tag{2}$$

where

$$\delta(y, z) = \begin{cases} 0 & if\ y = z \\ 1 & otherwise \end{cases} \tag{3}$$

Similar to the k-means algorithm, k-modes is also affected by the initialization step and requires the number of clusters k to be specified in advance. However, it is still efficient with the computational complexity of $O(Nkr)$, where N is the number of data points, k is the number of required clusters and r is the number of iterations.

***k*-prototypes**: extends both k-means and k-modes to clustering mixed numeric and categorical data [16]. The clustering method is similar to the k-means algorithm except that it uses the k-modes approach to update the categorical attribute values of cluster prototypes (i.e., centroids). It employs a heterogeneous distance function to compute the dissimilarity between data points and cluster prototypes. This cost function combines distance measure on both numerical and categorical attributes. While the Euclidean distance is used for numeric attributes, the categorical dissimilarity is derived from the number of mismatches between categorical values. In addition, this function requires different weights for the contribution of numeric versus categorical attributes to avoid favouring either type of attribute. Specifically, the distance between data point $x_i \in X$ (described by D_n numeric attributes and D_c categorical attributes) and cluster prototype $\overline{c_p}$ is estimated by

$$d(x_i, \overline{c_p}) = \sum_{j=1}^{D_n} (x_{ij} - \overline{c_{pj}})^2 + \gamma \sum_{g=1}^{D_c} \delta(x_{ig}, \overline{c_{pg}}), \tag{4}$$

where $\delta(y, z) = 0$ if $y = z$ and 1, otherwise. In addition, γ is a weight for categorical attributes. A large weight parameter γ means that the clustering process favours the categorical attributes, while a small γ indicates that numerical attributes are emphasized.

Agglomerative Hierarchical Clustering begins by considering each data point as a cluster (i.e., singleton cluster), and then gradually merges similar clusters until all the clusters are combined into one big group (i.e., the top node of a dendrogram). The resulting dendrogram can be cut at any level to obtain the desired data partitions [28]. The main differences among agglomerative clustering methods are the definitions of distance between two clusters, which are used to determine how data points in the dataset should be grouped into clusters. The well-known agglomerative hierarchical techniques are:

- *Single-Linkage (SL)*: defines the distance between two clusters to be the minimum distance between all pairs of data points, taken one from each cluster. Let C_p and C_q be clusters, the single-linkage distance between these two clusters $D_{C_pC_q}$ is defined by

$$D_{C_pC_q} = \min_{\forall x \in C_p, x' \in C_q} d(x, x'), \tag{5}$$

  where $d(x, x')$ is usually the Euclidean distance between data points $x, x' \in X$.
- *Complete-Linkage (CL)*: determines the dissimilarity between clusters via the largest distance between data points in the clusters under examination. Formally, the distance between two clusters $D_{C_pC_q}$ is defined as follows:

$$D_{C_pC_q} = \max_{\forall x \in C_p, x' \in C_q} d(x, x') \tag{6}$$

- *Average-Linkage (AL)*: uses the average value of all pair-wise distance among data points in the two clusters as the cluster distance. In particular, the distance between clusters C_p and C_q is estimated by

$$D_{C_pC_q} = \frac{1}{N_pN_q} \sum_{\forall x \in C_p} \sum_{\forall x' \in C_q} d(x, x'), \tag{7}$$

  where N_p and N_q are the number of data points in cluster C_p and C_q, respectively.

These methods provide visualizations on how data points are grouped in different levels on a dendrogram, which can help users to analyse data more easily or select desired groups of data at some fixed level of proximity. However, the main drawback of such techniques is their complexity, $O(N^2)$ to $O(N^3)$. Therefore, they are impractical for large datasets.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) relies on the density of data points in the neighbourhood of each data point [29]. This density with respect to a definite data point is determined by the number of the other objects within a hypersphere area around it. The key idea is that, for each object of a cluster, its neighbourhood of a given radius (ϵ) has to contain at least a minimum number of instances ($MinPts$).

Several definitions used in DBSCAN are defined, based on the input parameters ϵ and $MinPts$, as follows:

- An ϵ-*neighbourhood* of $x_i \in X$ is a set of data objects that have distance from x_i less than or equal to ϵ (i.e., ϵ indicates the neighbourhood radius).
- A *core object* is a data point with a neighbourhood consisting of more than $MinPts$ data points.
- An object $x_j \in X$ is *directly density-reachable* from a core object x_i if x_j is within ϵ-neighbourhood of x_i.
- An object x_j is *density-reachable* from a core object x_i if there exists a finite sequence of core objects between x_i and x_j such that each connecting core belongs to an neighbourhood of its predecessor.

- Two data points x_i and x_j are *density-connected* if they are density-reachable from a common core.

To find a cluster, DBSCAN checks the ϵ-neighbourhood of each data point. If the ϵ-neighbourhood of $x_i \in X$ contains more than *MinPts* members, a new cluster with x_i as a core object is created. Then it iteratively collects directly density-reachable objects from these core objects, which may involve merging a few density-reachable clusters. This process terminates when no new object can be added to any cluster. DBSCAN can find arbitrary shapes of clusters, identify outliers, and determine the number of clusters automatically. However, the main disadvantage is that the quality of the resulting clusters is sensitive to the user-defined parameters, ϵ and *MinPts*, which are difficult to determine. The computational complexity of DBSCAN is $O(N \log N)$ if a spatial index is used. Otherwise, it is $O(N^2)$, where N is the number of data points.

SOM (Self-Organizing Map) is a very useful and well-known tool [30] for a range of applications, including dimensional reduction, sampling, classification and data clustering [31,32]. SOM makes extensive use of the neural network technology, with the basic idea of mapping the data patterns onto a multi-dimensional grid of neurons or units. That grid forms the output space, as opposed to the input space where the original data patterns are. The underlying mapping attempts to preserve the 'topological' relations, i.e., those patterns that are close in the input grid are to be mapped to units that are close in the output grid. In other words, the information regarding neighbourhoods of the patterns under examination are preserved through the mapping process. Many SOM variants have been introduced in the literature, see Furukawa [33] and Tokunaga and Furukawa [34] for examples of the recent development.

Specific to data clustering, the basic SOM algorithm [32] can be described as follows, where X denotes a dataset of N data points:

1. Initialize a set of cluster centre, i.e., $\overline{C} = \{\overline{c_1}, \dots, \overline{c_k}\}$
2. For each data point $x_i \in X$:

   (a) Assign x_i to a cluster C_j that provides the minimum Euclidean distance between x_i and its cluster centre $\overline{c_j}$.

   (b) Update the cluster centre $\overline{c_j}$, which is the weight vector of SOM's output units, using the following equation.

$$\overline{c_j} = \overline{c_j} + h[x_i - \overline{c_j}], \tag{8}$$

where $h \in [0, 1]$ is the degree of neighbourhood. In addition to updating the centre $\overline{c_j}$ of the cluster that x_i belongs to, all the cluster centres that are in the neighbourhood of $\overline{c_j}$ on the grid map are also updated. This neighbourhood-based propagation is controlled by h, which can be specified using the neighbourhood functions such as the bell-shaped (Gaussian-like) and the square (or bubble). This process is repeated for all $x_i \in X$.

Despite its innovative concept and reported success, the major disadvantage of SOM is the long processing time, especially with a large dataset.

2.2. Basis of cluster ensembles

Although, a large number of clustering algorithms have been introduced for a variety of application areas [14], the No Free Lunch theorem [35] suggests[1] there is no single clustering algorithm that performs best for all datasets [36], i.e., no algorithm is able to discover all types of cluster shapes and structures presented in data [21–23]. Each algorithm has its own strengths and weaknesses. For a particular dataset, different algorithms, or even the same algorithm with different parameters usually provide distinct solutions. Therefore, it is extremely difficult for users to decide which algorithm would be the *proper* alternative for a given set of data. As identified in the previous section, the use of those conventional clustering algorithms such as k-means, k-modes and k-prototypes can be complicated due to their settings of k, distance metric or initial centroids. Likewise, agglomerative hierarchical clustering methods also encounter the problem of selecting and appropriate k, while this is naturally solved within the process of DBSCAN. Nonetheless, the latter suffers greatly from configuring several parameters.

Recently, cluster ensembles have emerged as an effective solution that is able to overcome these limitations, and improve the robustness as well as the quality of clustering results. The main objective of cluster ensembles is to combine different decisions of various clusterings in such a way to achieve the accuracy superior to those of individual clustering. Examples of well-known ensemble methods are: (i) the *feature-based* approach that treats the problem of cluster ensembles as the clustering of categorical data, i.e., cluster labels [24,37–40], (ii) the *direct* approach that finds the final partition through relabelling the results of base clustering [41,42], (iii) *graph-based* algorithms that employ a graph partitioning methodology [43–45], and (iv) the *pairwise-similarity* approach that makes use of co-occurrence relationships between all pairs of data points [22,46–51]. The following subsections will introduce three fundamental concepts of problem definition, ensemble generation and consensus function, respectively.

2.2.1. Problem formulation

Let $X = \{x_1, \dots, x_N\}$ be a set of N data points, where each $x_i \in X$ is represented by a vector of D attribute values, i.e., $x_i = (x_{i,1}, \dots, x_{i,D})$. Also, let $\Pi = \{\pi_1, \dots, \pi_M\}$ be a cluster ensemble with M base clusterings, each of which is referred to as an 'ensemble member'. Each base clustering returns a set of clusters $\pi_g = \{C_1^g, C_2^g, \dots, C_{k_g}^g\}$, such that $\bigcup_{t=1}^{k_g} C_t^g = X$, where k_g is the number of clusters in the gth clustering. For each $x_i \in X$, $C^g(x_i)$ denotes the cluster label in the gth base clustering to which data point x_i belongs, i.e., $C^g(x_i) = $ 't' (or 'C_t^g') if $x_i \in C_t^g$. The problem is to find a new partition $\pi^* = C_1^*, \dots, C_K^*$, where K denotes the number of clusters in the final clustering result, of a dataset X that summarizes the information from the cluster ensemble Π.

The general framework of cluster ensembles is shown in Fig. 1. Essentially, solutions achieved from different base clusterings are aggregated to form a final partition. This meta-level method involves two major tasks of: (i) generating a cluster ensemble, and (ii) producing the final partition (normally referred to as a *consensus function*).

2.2.2. Ensemble generation strategies

It has been shown that ensembles are most effective when constructed from a set of predictors whose errors are dissimilar [52]. To a great extent, diversity amongst ensemble members is introduced to enhance the result of an ensemble [53]. Specific to data clustering, the results obtained with any single algorithm over many iterations are usually very similar. In such circumstance when all ensemble members agree on how a dataset should be partitioned, aggregating the base clustering results will show no improvement over any of the constituent members. Several heuristics have been proposed to introduce artificial instabilities in clustering algorithms, hence the diversity within a cluster ensemble. The following ensemble generation methods yield different clusterings of the same data, by exploiting different cluster models and different data partitions.

[1] The No Free Lunch theorem seems to apply here because the problem of clustering can be reduced to an optimization problem — we are seeking to find the optimal set of clusters for a given dataset via an algorithm.

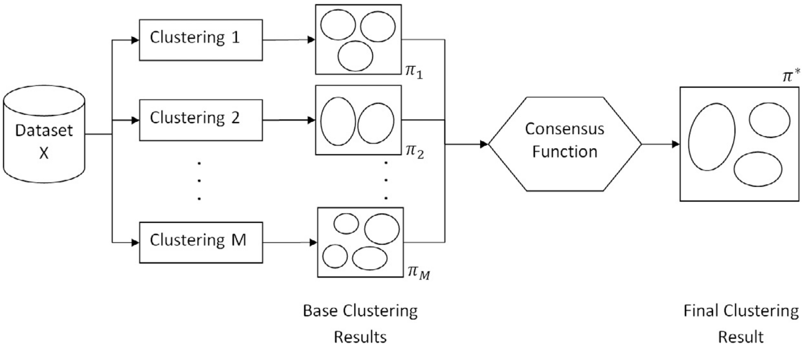


Fig. 1. The basic process of cluster ensembles. It first applies multiple base clusterings to a dataset X to obtain diverse clustering decisions ($\pi_1 \ldots \pi_M$). Then, these solutions are combined to establish the final clustering result (π^*) using a consensus function.

Homogeneous ensembles: Base clusterings are created using repeated runs of a single clustering algorithm, with several sets of parameter. In particular, the k-means technique has often been employed with a random initialization of cluster centres [22,40,42,49,50,54]. An ensemble of k-means is computational efficient as its time complexity is $O(kNM)$, where k, N and M denote the number of clusters, the number of data points and the number of base clusterings, respectively. Other non-deterministic clustering techniques (results of multiple runs are dissimilar) such as PAM [55] can be used to form a homogeneous ensemble.

However, as compared with k-means, the ensembles of PAM are less efficient with time complexity of $O(Mk(N-k)^2)$ and $O(M(ks^2+k(N-k)))$, respectively. Note that s denotes the sample size ($s < N$). Unlike the aforementioned alternatives of base clustering, hierarchical clustering techniques (e.g., single-linkage (SL), complete-linkage (CL) and average-linkage (AL)) are deterministic with the identical result being obtained from multiple runs for any given number of clusters, k. Hence, such methods cannot generate diversity within a homogeneous ensemble.

This method is widely practised to create base clusterings each with distinct parameters and hence output. Selecting such parameters is problematic for a basic clustering approach, but with the ensemble method, a variation of setting can be randomly chosen for generate diverse versions of clustering model.

Different-k: The output of clustering algorithms is dependent on the initial choice of the number of clusters k. To acquire an ensemble diversity, base clusterings are created using a specific value of k or randomly selected k from a pre-specified interval. Intuitively, k should be greater than the expected number of clusters and the common rule-of-thumb is $k=\sqrt{N}$ [22,53,54,56]. This generation scheme allows a large number of clustering algorithms, both partitioning and hierarchical, to be used as base clustering. However, k-means is still often employed for the efficiency reason. It is noteworthy that the time complexity of creating a cluster ensemble with a hierarchical clustering technique being used as base clusterings is $O(N^3M)$.

Random subspacing/sampling: A cluster ensemble can also be achieved by applying manifold subsets of initial data to base clusterings. It is assumed that each clustering algorithm can provide different levels of performance for different partitions of a dataset [43]. In practice, data partitions can be obtained by projecting data onto different subspaces [47,57], choosing different subsets of features [45,58], or using data sampling techniques [41,59,60].

Let a matrix $X \in R^{N\times D}$ represents a dataset of N data points each of which is specified by D attributes/features. An artificial diversity within an ensemble Π can be achieved by generating base clustering results from different perturbed variations of X. To this extent, a random projection method [61] is objectively used in [47] and [57] to create such a transformed data matrix $X' \in R^{N\times D'}$ from the original X, where $D' < D$.

It is also possible to create different data subspaces each of which contains a randomly selected subset of original attributes [58,62]. Each data subspace X' is generated by firstly defining D':

$$D' = D'_{min} + \lfloor \alpha(D'_{max} - D'_{min}) \rfloor, \tag{9}$$

where $\alpha \in [0,1]$ is a uniform random variable, D'_{min} and D'_{max} are the lower and upper bounds of the generated subspace, respectively. Following Yu et al. [58], D'_{min} and D'_{max} are set to $0.75D$ and $0.85D$. An attribute is selected one by one from the pool of D attributes, until the collection of D' is obtained. In particular, the index of each randomly selected attribute is determined as follows:

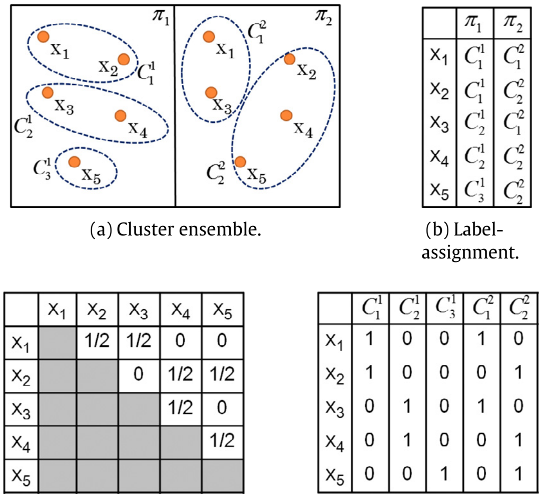
$$h = \lfloor 1 + \beta D \rfloor, \tag{10}$$

where h denotes the hth attribute in the pool of D attributes and $\beta \in [0,1)$ is a uniform random variable.

In addition to using data subspaces, an ensemble can also be created by applying a selected clustering algorithm(s) to a set of data perturbations. In the studies of Dudoit and Fridyand [59] and Fischer and Buhmann [41], perturbed datasets (of the same size as the original data) are obtained using the bootstrapping (or bagging) resampling scheme [63], whereby data points are sampled with replacement from the original dataset. Despite its effectiveness, especially for classifier ensembles, bootstrapping produces datasets with duplicated data points, which artificially distort the actual data compactness. An alternative to overcome this shortcoming is a subsampling technique, whereby a subset of data points is sampled without replacement from the original dataset. Specific to the strategy employed by Kim and Lee [64] and Monti et al. [51], each base clustering is obtained with a data subset that contains randomly selected 80% of original data points.

Heterogeneous ensembles: As an alternative, heterogeneous ensembles may be exploited, where diversity is induced by allowing each base clustering to be generated using different clustering algorithms [46,65,66]. The key idea is that each clustering technique has its own benefits and drawbacks, and is suitable for different types of dataset. Multiple algorithms can provide different decisions on data partitions and complement each other. Thus, combining different clustering results based on multiple clustering techniques can assure better data clusterings. This approach is adapted by many ensemble algorithms, for example, the clustering aggregation proposed by Gionis et al. [42] applies single linkage, average linkage, complete linkage, Ward's clustering and k-means to generate the ensembles.

Mixed heuristics: In addition to using one of the aforementioned methods, any combination of them can be applied as well.



(a) Cluster ensemble.

	π_1	π_2
x_1	C_1^1	C_1^2
x_2	C_1^1	C_2^2
x_3	C_2^1	C_1^2
x_4	C_2^1	C_2^2
x_5	C_3^1	C_2^2

(b) Label-assignment.

	x_1	x_2	x_3	x_4	x_5
x_1		1/2	1/2	0	0
x_2			0	1/2	1/2
x_3				1/2	0
x_4					1/2
x_5					

(c) Pairwise similarity.

	C_1^1	C_2^1	C_3^1	C_1^2	C_2^2
x_1	1	0	0	1	0
x_2	1	0	0	0	1
x_3	0	1	0	1	0
x_4	0	1	0	0	1
x_5	0	0	1	0	1

(d) Binary cluster-association.

Fig. 2. Examples of (a) cluster ensemble, (b) label-assignment matrix, (c) pairwise similarity matrix and (d) binary cluster-association (BA) matrix. Note that $X = \{x_1, \ldots, x_5\}$, $\Pi = \{\pi_1, \pi_2\}$, $\pi_1 = \{C_1^1, C_2^1, C_3^1\}$ and $\pi_2 = \{C_1^2, C_2^2\}$.

This can be found in [45], where several clusters (i.e., clustering algorithms) are used with multiple subspaces of data. Similarly, Monti et al. [51] apply hierarchical clustering with average-linkage and the self organizing map (SOM) with different sets of sampled data, while Domeniconi and Al-Razgan [43] generate the ensembles using their subspace clustering algorithm (LAC) with different input conditions. In addition, Fred and Jain [67] employ all strategies to construct their ensembles by applying three algorithms (k-means, single-linkage and spectral algorithm), with various initial settings, to multiple subsampled data. In the experiments of Nguyen and Caruana [39], the ensembles are produced using weighting k-means and k-means with different random restarts.

2.2.3. Consensus functions

Having obtained the cluster ensemble, a variety of consensus functions have been developed and made available for deriving the ultimate data partition. Each consensus function utilizes a specific form of information matrix, which summarizes the base clustering results. From the cluster ensemble shown in Fig. 2(a), three general types of such ensemble-information matrix can be constructed. Firstly, the label-assignment matrix (Fig. 2(b)), of size $N \times M$, represents cluster labels that are assigned to each data point by different base clusterings. Secondly, the pairwise similarity matrix (Fig. 2(c)), of size $N \times N$, summarizes co-occurrence statistics amongst data points. Furthermore, the binary cluster-association (BA) matrix (see Fig. 2(d) for an example) provides a cluster-specific view of the original label-assignment matrix. The association degree that a data point belonging to a specific cluster is either 1 or 0.

Given this background, a large number of different consensus functions found in the literature can be described and classified to four major categorizations: direct, feature-based, pairwise-similarity based and graph-based approaches, respectively. Examples of cluster ensemble methods belonging to these families will be provided in the next section.

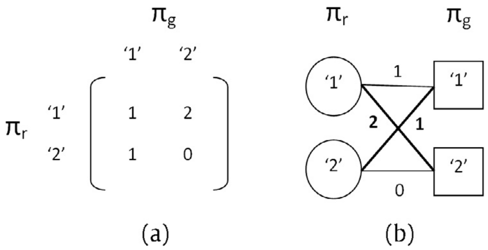


Fig. 3. An example of (a) contingency matrix Ω and (b) the corresponding weighted bipartite graph.

3. Cluster ensemble methods

3.1. Direct approach

The first family of cluster ensemble methods is characterized by the use of a combination strategy such as 'voting', which has proven effective for classifier ensembles [68,69]. However, such practice is not directly applicable to the cluster ensemble problem where a priori class information is not available. The cluster labels in different data partition (i.e., base clustering) $\pi_g, g = 1 \ldots M$ are arbitrary. As a result, a mechanism that finds 'label correspondence' and re-labels each partition in accordance with a reference partition, is necessary for developing such a voting model. Most methods in this category require the number of clusters in each base partition to be K, i.e., $k_g = K, g = 1 \ldots M$.

Simple Voting: Based on the analysis of Topchy et al. [70], the underlying re-labelling process is equivalent to the problem of maximum weight bipartite matching. This starts with the creation of a contingency matrix $\Omega \in R^{K \times K}$ from the reference π_r and to-be re-labelled π_g partitions, where K is the number of clusters in each partition. Each entry $\Omega(l, l')$ that denotes the co-occurrence statistics between labels $l \in \pi_r$ and $l' \in \pi_g$, is defined by

$$\Omega(l, l') = \sum_{\forall x_i \in X} \omega(x_i), \tag{11}$$

where $\omega(x_i) = 1$ if $(C^r(x_i) = l) \wedge (C^g(x_i) = l')$, otherwise $\omega(x_i) = 0$. Having obtained Ω, the label correspondence is solved by maximizing

$$\sum_{l=1}^{K} \sum_{l'=1}^{K} \Omega(l, l')\Theta(l, l'), \tag{12}$$

where $\Theta \in R^{K \times K}$ is another matrix representing correspondences amongst labels of partitions π_r and π_g. An entry $\Theta(l, l') = 1$ if label $l \in \pi_r$ corresponds to label $l' \in \pi^g$, 0 otherwise. Note also that

$$\sum_{l=1}^{K} \Theta(l, l') = \sum_{l'=1}^{K} \Theta(l, l') = 1 \tag{13}$$

The solution to this optimization problem can be found using the Hungarian algorithm [71]. Fig. 3(a) presents an example of the contingency matrix Ω which is created from two sets of labels given to four data points $X = \{x_1, \ldots, x_4\}$ within partitions $\pi_r = \{1, 1, 1, 2\}$ and $\pi_g = \{1, 2, 2, 1\}$. As shown in Fig. 3(b), the Ω matrix can also be presented as a weighted bipartite graph, in which the maximum matchings are identified as bold edges. This suggests re-labelling the label '1' of partition π_g as '2', and the label '2' as '1'.

Particularly to the study of Topchy et al. [70], the reference partition π_r is randomly selected from the pool of M partitions in an ensemble Π, i.e., $\pi_r \in \Pi$. Then, each of the $M - 1$ remaining partitions is re-labelled with respect to the chosen π_r, by following the aforementioned steps. Hence, a globally consistent label set is employed by all partitions. Then, a plurality voting can be employed to determine the consensus label of each data point $x_i \in X$. The consensus methods of Dudoit and Fridyand [59] and Fischer and Buhmann [41] also implement a similar voting model. However, the reference partition $\pi_r \notin \Pi$ is obtained from the original data (X), while partitions in an ensemble ($\pi_1 \ldots \pi_M \in \Pi$) are acquired on subsets of X. That is, each partition $\pi_g, g = 1 \ldots M$ is re-labelled in accordance with π_r that is not part of the ensemble itself.

Incremental Voting: The incremental (or cumulative) voting model is initially developed in the studies of Ayad and Kamel [72], Dimitriadou et al. [73,74], Frossyniotis et al. [75], and later generalized by Ayad and Kamel [76,77]. Unlike the simple voting previously discussed, data partitions are added to the underlying ensemble one by one, with the voting statistics being repeatedly updated. Let $P_g \in R^{N\times K}$ be the matrix representing the gth partition, i.e., $\pi_g \in \Pi$. Each $P_g(x_i, C_t^g)$ is 1 if data point $x_i \in X$ belongs to cluster $C_t^g \in \pi_g$, 0 otherwise. Also let $V_g \in R^{N\times K}$ be the matrix presenting the result of combining the first g partitions $(\pi_1, \ldots, \pi_g)$, and $V_g(x_i, L_j) \in \{0, 1, \ldots, g\}$ is the accumulative frequency (or the number of partitions) that the label L_j is assigned to data point x_i. Note that, initially, $V_1 = P_1$.

At the $(g+1)$th step ($1 < (g+1) \leq M$) where the $(g+1)$th partition is added to cluster ensemble, the relabelling algorithm such as Hungarian is exploited to find the correspondence between cluster labels (or columns) of the matrices V_g and P_{g+1}, with the first set being the reference partition. This begins with the creation of the contingency matrix, $\Omega \in R^{K\times K}$, where each $\Omega(l, l')$ is estimated as

$$\Omega(l, l') = \sum_{\forall x_i \in X} \omega(x_i), \tag{14}$$

where $\omega(x_i) = 1$ if $(V_g(x_i, l) \geq 1) \wedge (P_g(x_i, l') = 1)$, otherwise $\omega(x_i) = 0$. The optimal label (column) correspondence can be found as the maximum matching in the weighted bipartite graph created from Ω.

After that, the matrix V_{g+1} that represents the result of merging $g+1$ partitions is generated such that an entry $V_{g+1}(x_i, l)$ is calculated as

$$V_{g+1}(x_i, l) = V_g(x_i, l) + P_{g+1}(x_i, l'), \tag{15}$$

where the label or column l' of P_{g+1} corresponds to the column l of V_g.

To decide the final label $C^*(x_i)$ of each data point $x_i \in X$ from the incremental combination of M data partitions, the V_M matrix is exploited as follows:

$$C^*(x_i) = \underset{l}{\operatorname{argmax}}\, V_M(x_i, l) \tag{16}$$

Label Correspondence Search (LCS): Instead of relying on the combination strategy inherited from the task of classifier ensembles, the method introduced by Boulis & Ostendorf [37] searches for 'label correspondence', which has been specifically modelled as an optimization problem. Let $U_t^g \in R^{N\times 1}$ be a vector that represents the posteriors of cluster C_t^g for N data points. With respect to data point $x_i \in X$, the ith entry $U_t^g(x_i) = p(C_t^g|x_i)$. Specifically to a crisp partition, $U_t^g(x_i) = 1$ if $x_i \in C_t^g$, 0 otherwise.

The agreement $G(C_t^g, C_{t'}^{g'})$ between clusters C_t^g and $C_{t'}^{g'}$ in the partitions $\pi^g, \pi^{g'} \in \Pi$, respectively, can be defined as

$$G(C_t^g, C_{t'}^{g'}) = \left(U_t^g\right)^T \cdot U_{t'}^{g'} \tag{17}$$

This allows the correspondence between clusters to be identified. Hence, LCS makes use of such measure to formulate the following 'goodness' function, F^Λ. It is objectively maximized to generate meta-clusters $C_m^*, m = 1 \ldots K$ from clusters in a given ensemble Π.

$$F^\Lambda = \sum_{m=1}^{K} \sum_{g=1}^{M} \sum_{t=1}^{K} \Lambda(C_t^g, C_m^*) \times S(C_t^g, C_m^*), \tag{18}$$

subjected to

$$\sum_{m=1}^{K} \Lambda(C_t^g, C_m^*) = 1,\ \forall g, t \tag{19}$$

here $\Lambda(C_t^g, C_m^*) = 1$ if the cluster $C_t^g \in \pi^g$ is assigned to the meta-cluster C_m^* and 0 otherwise. Also, $S(C_t^g, C_m^*)$ denotes the score of assigning the cluster C_t^g to the meta-cluster C_m^*, provided that $C_t^g \in C_m^*$ if $\Lambda(C_t^g, C_m^*) \neq 0$.

$$S(C_t^g, C_m^*) = \frac{1}{|C_m^*|} \sum_{\forall cl \in C_m^*, cl \neq C_t^g} G(C_t^g, cl) \tag{20}$$

Optionally, the objective function F^Λ can be maximized with respect to the additional constraint of:

$$\sum_{t=1}^{K} \Lambda(C_t^g, C_m^*) = 1,\ \forall m, g \tag{21}$$

According to [37], two models can be derived by omitting the constraint given in Eq. (21) or including it in the optimization process. This results in the 'Unconstrained' and 'Constrained' methods, respectively. The outcome of these models is the matrix $F \in R^{N\times K}$ where an entry $F(x_i, C_m^*)$ denotes the association between data point $x_i, i = 1 \ldots N$ and meta-cluster $C_m^*, m = 1 \ldots K$. Each column $F_m, m = 1 \ldots K$ represents the centroid of the meta-cluster C_m^*. In particular, F_m is defined as

$$F_m = \frac{1}{|C_m^*|} \sum_{\forall\, C_t^g \in C_m^*} \left(U_t^g\right)^T \tag{22}$$

The final label $C^*(x_i)$ of data point $x_i \in X$ is α given that

$$F(x_i, C_\alpha^*) = \max_{m=1\ldots K} F(x_i, C_m^*) \tag{23}$$

Another method, named 'Singular Value Decomposition Combination', to create the output matrix F has also been proposed in [37]. Let $p_g(t|i)$ be the posterior of cluster C_t^g in the partition $\pi^g \in \Pi$ for data point $x_i \in X$. Each entry $F(x_i, C_m^*), \forall\, i = 1 \ldots N, \forall\, m = 1 \ldots K$ is estimated by

$$F(x_i, C_m^*) = \sum_{g=1}^{M} \lambda(f^g(C_m^*), C_m^*) \times p_g(h(f^g(C_m^*))|i), \tag{24}$$

where $f^g(C_m^*)$ denotes the function that aligns the meta-cluster C_m^* and clusters in the partition π^g. This is estimated using the SVD (Singular Value Decomposition) technique which finds the most correlated pair of clusters. In addition, λ weighting function provides a soft alignment of clusters in question, while the function $h(l) = l - K\lfloor l/K \rfloor$.

3.2. Feature-based approach

Several techniques that are categorized into this group are based similarly on the categorical/nominal data presented in the label-assignment matrix (see example in Fig. 2(b)). Unlike the direct approach previously discussed, feature-based methods cluster data points using the nominal information that is originally obtained from an ensemble, without searching for correspondence amongst labels or re-labelling. Details of such models are given below:

Iterative Voting Consensus (IVC): This feature-based method was recently introduced by Nguyen and Caruana [39]. It aims to obtain the consensus partition π^* of data points X from the label-assignment or categorical data matrix that is induced by a cluster ensemble $\Pi = \{\pi_1, \ldots, \pi_M\}$. IVC makes use of the set of label vectors $Y = (y_1, \ldots, y_N)$, with N denoting the number of data points and $y_i, i = 1 \ldots N$ being specified as

$$y_i = (y_{i1} = C^1(x_i), \ldots, y_{iM} = C^M(x_i)), \tag{25}$$

where $C^g(x_i)$ represents a label of specific cluster in clustering $\pi_g, g = 1 \ldots M$, to which data point $x_i \in X$ belongs. Note that y_i corresponds to the ith row in the label-assignment matrix.

In each iteration, IVC first estimates the centre of each cluster in π^*. Note that each cluster $C_p^*, p = 1 \ldots K$ in the target clustering π^* has a cluster centre $\overline{C}_p^* = \{mode(X_p, \pi_1), \ldots, mode(X_p, \pi_M)\}$, where $X_p \subset X$ is the set of data points belonging to the cluster C_p^* and $mode(X_p, \pi_g)$ denotes the majority labels (in the clustering π_g) of members in X_p. Having obtained these centres, IVC then reassigns each data point to its closest cluster centre. This is possible using the Hamming distance between M-dimensional vectors that represent data points and cluster centres. The iterative process continues until there is no change with the target clustering π^*. It is noteworthy that a consensus function similar to IVC has also been developed in the study of Luo et al. [78].

Mixture Model: Similar to IVC, this method of Topchy et al. [24] also generates the final clustering π^* from the label matrix Y. In particular, it is based on a finite mixture model for the probability of the cluster labels $y_i = (C^1(x_i), \ldots, C^M(x_i))$ of data point $x_i \in X$, which is acquired from the ensemble $\Pi = \{\pi_1, \ldots, \pi_M\}$. Label vectors y_i representing data points x_i are specified as random variables generated from a probability distribution. This can be described as a mixture of multivariate component densities.

$$P(y_i|\Theta) = \sum_{t=1}^{K} \varphi_t P_t(y_i|\theta_t), \tag{26}$$

where $\Theta = \{\varphi_1, \ldots, \varphi_K, \theta_1, \ldots, \theta_K\}$ is the collection of components. The tth components are identified with respect to the cluster $C_t^*, t = 1 \ldots K$ in the final clustering π^*. In this model, each data point $y_i, i = 1 \ldots N$ is presumed to be created by: first, drawing a component in according to the mixing coefficient φ_t, then, sampling a data point from the distribution $P_t(y_i|\theta_t)$.

The mixture model is formulated as a maximum likelihood estimation problem, which aims to find the best fitting mixture density for a given data Y. This is obtained by maximizing the following likelihood function with respect to the unknown Θ.

$$\Theta^* = \underset{\Theta}{\operatorname{argmax}} \ \log L(\Theta|Y) \tag{27}$$

By assuming that all data points $y_1, \ldots, y_N$ are independent and identically distributed, the previous function can be simplified to

$$\begin{aligned} \log L(\Theta|Y) &= \log \prod_{i=1}^{N} P(y_i|\Theta) \\ &= \sum_{i=1}^{N} \log \sum_{t=1}^{K} \varphi_t P_t(y_i|\theta_t) \end{aligned} \tag{28}$$

Following that, for each data point, the corresponding density distribution $P_t(y_i|\theta_t)$ is defined as follows, where a conditional independence assumption is made for the components of y_i.

$$P_t(y_i|\theta_t) = \prod_{g=1}^{M} P_t^g(y_{ig}|\theta_t^g), \tag{29}$$

given that

$$P_t^g(y_{ig}|\theta_t^g) = \prod_{l=1}^{k_g} (\vartheta_{gt}(l))^{\delta(y_{ig},l)}, \tag{30}$$

where k_g is the number of clusters in the clustering $\pi^g \in \Pi$, $\vartheta_{gt}(l), l = 1 \ldots k_g$ are probabilities of the outcomes with $\sum_{l=1}^{k_g} \vartheta_{gt}(l) = 1, \forall g = 1 \ldots M, \forall t = 1 \ldots K$, and $\delta(y_{ig}, l)$ is defined as

$$\delta(y_{ig}, l) = \begin{cases} 1 & if\ y_{ig} = l \\ 0 & otherwise \end{cases} \tag{31}$$

The EM algorithm is adopted to optimize the likelihood function given in Eq. (28). For such purpose, the existences of hidden data Z and the likelihood of complete data (Y, Z) are hypothesized. It is possible to identify which of K mixture components has been exploited to generate data point y_i if the corresponding $z_i = (z_{i1}, \ldots, z_{iK})$ is known. Specifically, $z_{it} = 1$ if y_i belongs to the tth component (i.e., $y_i \in C_t^*, t = 1 \ldots K$), otherwise $z_{it} = 0$. As a result, the likelihood function of Eq. (28) is modified as

$$\log L(\Theta|Y, Z) = \sum_{i=1}^{N} \sum_{t=1}^{K} z_{it} \log \varphi_t P_t(y_i|\theta_t) \tag{32}$$

The resulting EM process begins with an initial guess of the model parameters $\Theta' = \{\varphi_1', \ldots, \varphi_K', \theta_1', \ldots, \theta_K'\}$. Then the following steps are repeated until the convergence criterion is satisfied. As suggested by Topchy et al. [24], the stability of the assignment of data points Y (or equivalently X) can be employed as a convergence criterion in practice.

1. Compute expected values $E[z_{it}], \forall i = 1 \ldots N, t = 1 \ldots K$:

$$E[z_{it}] = \frac{\varphi_t' \prod_{g=1}^{M} \prod_{l=1}^{k_g} (\vartheta_{gt}'(l))^{\delta(y_{ig},l)}}{\sum_{s=1}^{K} \varphi_s' \prod_{g=1}^{M} \prod_{l=1}^{k_g} (\vartheta_{gs}'(l))^{\delta(y_{ig},l)}} \tag{33}$$

2. Re-estimate the parameters:

$$\varphi_t = \frac{\sum_{i=1}^{N} E[z_{it}]}{\sum_{i=1}^{N} \sum_{t=1}^{K} E[z_{it}]} \tag{34}$$

$$\vartheta_{gt}(l) = \frac{\sum_{i=1}^{N} \delta(y_{ig}, l) E[z_{it}]}{\sum_{i=1}^{N} \sum_{l=1}^{k_g} \delta(y_{ig}, l) E[z_{it}]} \tag{35}$$

Having obtained the final (or converged) Z, the consensus cluster label of each data point x_i (or $y_i, i = 1 \ldots N$) can be defined as $C^*(x_i) = \xi$ provided that

$$E[z_{i\xi}] = \max_{t=1 \ldots K} E[z_{it}] \tag{36}$$

Clustering Aggregation (AGG): The problem of clustering aggregation [42] is to find a clustering that minimizes the 'disagreements' with ensemble members. Formally, a measure of disagreement between two clustering $\pi_a, \pi_b \in \Pi$ with respect to two specific data points $x_i, x_j \in X$ can be defined as follows:

$$d_{x_i,x_j}(\pi_a, \pi_b) = \begin{cases} 1 & if\ \left(C^a(x_i) = C^a(x_j) \wedge C^b(x_i) \neq C^b(x_j)\right) \vee \\ & \quad \left(C^a(x_i) \neq C^a(x_j) \wedge C^b(x_i) = C^b(x_j)\right) \\ 0 & otherwise \end{cases}, \tag{37}$$

where $C^g(x_i)$ denotes the label that is assigned to data point $x_i \in X$ in the clustering $\pi_g \in \Pi$. Note that such information is summarized by the label-assignment matrix, see Fig. 2(b) for example.

Given the set of data points $X = \{x_1, \ldots, x_N\}$, the distance/proximity between two clusterings $\pi_a, \pi_b \in \Pi$ is specified by

$$d_X(\pi_a, \pi_b) = \sum_{\forall (x_i, x_j) \in X \times X} d_{x_i, x_j}(\pi_a, \pi_b) \tag{38}$$

With the cluster ensemble $\Pi = \{\pi_1, \ldots, \pi_M\}$ of data points X, the aim of clustering aggregation is to search for a median partition π^* that minimizes the following objective function:

$$D(\pi^*) = \sum_{g=1}^{M} d_X(\pi_g, \pi^*) \tag{39}$$

According to Gionis et al. [42], this problem can be generalized to 'correlation clustering' [79] that sets to minimize the cost function of:

$$d(\pi^*) = \sum_{\substack{\forall (x_i, x_j), \\ C^*(x_i) = C^*(x_j)}} DA(x_i, x_j) + \sum_{\substack{\forall (x_i, x_j), \\ C^*(x_i) \neq C^*(x_j)}} 1 - DA(x_i, x_j), \tag{40}$$

where $DA \in \mathbb{R}^{N \times N}$ is the matrix of distance amongst N data points and each $DA(x_i, x_j)$ can be estimated by

$$DA(x_i, x_j) = \frac{\sum_{g=1}^{M} \beta^g(x_i, x_j)}{M}, \tag{41}$$

here $\beta^g(x_i, x_j) = 1$ if $C^g(x_i) \neq C^g(x_j)$, and 0 otherwise.

Based on the aforementioned basis, a number of algorithms have been proposed to find the partition π^* by applying conventional clustering techniques to the discovered DA matrix. In particular, three algorithms of AGG_F, AGG_{LSR} and AGG_{LSF}, apparently the most effective (see further details in [42]), are included in this review.

At the outset, the AGG_F algorithm makes use of the Furthest-First traversal (FF) method of Hochbaum and Shmoys [80]. AGG_F begins with a single cluster that contains all data points. It then searches for the pair of data points ($x_i, x_j \in X$) which are furthest apart; in other words, $DA(x_i, x_j) = \max_{\forall (x_p, x_q) \in X \times X} DA(x_p, x_q)$. These data points become new clusters' centres and the remaining data points are assigned to the closest cluster (i.e., the closest cluster centre). This process is iterated such that, at each step, a new cluster centre which is the furthest from the existing centres is selected. The data points are re-assigned to the centre that incurs the least cost. At the end of each step, the cost of a new solution is calculated using Eq. (40). If it is lower than that of the prior step, the aforementioned procedure continues. Otherwise, the algorithm terminates and outputs the previous solution.

Another algorithm called 'Local Search' is also introduced for the clustering aggregation problem. It begins with an initial partition of data points, which can be obtained randomly or from the result of another model such as AGG_F. The resulting methods are referred to as AGG_{LSR} and AGG_{LSF} with the former and the latter setting, respectively. They similarly determine the cluster, one of the existing clusters or a new singleton cluster, that each data point should belong to with the minimum cost. This procedure is repeated until an additional alteration cannot further decrease the cost.

In particular to a data point $x_i \in X$ and the data partition $\pi^* = \{C_1^*, \ldots, C_K^*\}$, the cost $d(x_i, C_p^*)$ of assigning x_i to $C_p^* \in \pi^*$ is defined as

$$d(x_i, C_p^*) = \sum_{\forall x_j \in C_p^*} DA(x_i, x_j) + \sum_{\forall x_l \notin C_p^*} 1 - DA(x_i, x_l) \tag{42}$$

Similarly, the cost of assigning x_i to a new singleton cluster $C_{single} \notin \pi^*$ can be given by

$$d(x_i, C_{single}) = \sum_{\forall x_j \in X} 1 - DA(x_i, x_j) \tag{43}$$

Quadratic Mutual Information (QMI): The cluster ensemble method of Topchy et al. [24] searches for a 'median' partition that is the most similar to those data partitions generated by ensemble members. This is achieved by maximizing the measure of 'Quadratic mutual information (QMI)' which determines the quality of the final clustering result. In particular, QMI and CU (Category Utility) that is employed by the conceptual clustering (COBWEB) algorithm [81] give the same consensus clustering criterion (see proofs and further details in the study of Topchy et al. [24]). Also, it has been demonstrated by Mirkin [82] that the maximization of CU is equivalent to minimization of the square-error criterion of k-means if the number of clusters in target partition is fixed. In particular, the label-assignment matrix (Fig. 2(b)) acquired from the cluster ensemble under examination is firstly converted into its equivalent Binary Cluster-Association (BA) matrix (Fig. 2(d)) counterpart. It is then transformed to another numerical variation (TMB) to which k-means can be effectively applied. The value of each $TMB(x_i, cl)$, $\forall x_i \in X$, $\forall cl \in \pi_g$, $g = 1 \ldots M$ can be defined by

$$TMB(x_i, cl) = BA(x_i, cl) - p(cl), \tag{44}$$

where $p(cl)$ is simply estimated as follows, given that N is the number of data points.

$$p(cl) = \frac{\sum_{j=1}^{N} BA(x_j, cl)}{N} \tag{45}$$

Refined K-Means (RKM): The RKM method of Bradley and Fayyad [83] provides a general intuition of combining multiple clustering results. Given an ensemble Π of M members, each base clustering $\pi_g \in \Pi$ of K clusters is obtained by applying k-means to the dataset X (or perhaps, a subset of X). Let $\overline{C_t^g}$ represents the centroid of the cluster C_t^g, $t = 1 \ldots K$ in π^g. RKM considers the set of centroids CM that is obtained from the underlying ensemble ($CM = \{\overline{C_1^1}, \ldots, \overline{C_K^1}, \ldots, \overline{C_1^M}, \ldots, \overline{C_K^M}\}$) as features for the next clustering stage, i.e., clustering clusters.

Particularly, each collection FM_p, $p = 1 \ldots M$ of refined centroids is created by applying k-means to CM using $\{\overline{C_1^p}, \ldots, \overline{C_K^p}\} \subset CM$ as the initial K centroids. Note that FM_p consists of K refined centroids $\{\overline{F_1^p}, \ldots, \overline{F_K^p}\}$. Following that, the best set BM of refined centroids is selected from $\{FM_1, \ldots, FM_M\}$, using the distortion measure Φ:

$$BM = \operatorname*{argmin}_{FM_p} \Phi(FM_p, CM), \tag{46}$$

and $\Phi(FM_p, CM)$ is defined as

$$\Phi(FM_p, CM) = \sum_{\forall \overline{cl} \in FM_p} \sum_{\forall \overline{cl'} \in CM} d(\overline{cl}, \overline{cl'}), \tag{47}$$

where $d(\overline{cl}, \overline{cl'})$ is the Euclidean distance between centroids $\overline{cl} \in FM_p$ and $\overline{cl'} \in CM$. Having obtained BM, k-means is applied to X to generate the final clustering π^*, using the best refined centroids of BM to initialize the clustering process.

3.3. Pairwise-similarity based approach

This specific category of cluster ensemble methods is based principally on the pairwise similarity amongst data points [22]. A number of different consensus functions have been applied to such similarity matrix to generate the final clustering result.

Agglomerative Hierarchical Clustering Models: Given a set of data points $X = \{x_1, \ldots, x_N\}$, it first generates a cluster ensemble $\Pi = \{\pi_1, \ldots, \pi_M\}$ by applying M base clusterings to the dataset X. Following that, an $N \times N$ similarity matrix is constructed for each base clustering, denoted as $S_g, g = 1 \ldots M$. Each entry in this matrix represents the relationship between two data points. If they are assigned to the same cluster, the entry will be 1, 0 otherwise. More precisely, the similarity between two data points $x_i, x_j \in X$ from the gth ensemble member can be computed as

$$S_g(x_i, x_j) = \begin{cases} 1 & if\ C^g(x_i) = C^g(x_j) \\ 0 & otherwise \end{cases} \tag{48}$$

Following that, M similarity matrices of $S_1, \ldots, S_M$ are merged to form a 'co-association (CO)' matrix [22], which is also called consensus matrix [51], similarity matrix [45] and agreement matrix [84] — see Fig. 2(c) for an example. Each entry in the CO matrix represents the similarity between any two data points, which is a ratio of a number of ensemble members in which they are assigned to the same cluster to the total number of ensemble members. Formally, each entry of such a matrix $CO(x_i, x_j), x_i, x_j \in X$ is defined as

$$CO(x_i, x_j) = \frac{1}{M} \sum_{g=1}^{M} S_g(x_i, x_j) \tag{49}$$

Since the CO matrix is a similarity matrix, any similarity-based clustering algorithm (as a consensus function) can be applied to this matrix to yield the final partition π^* [22]. Amongst several existing similarity-based techniques, the most well-known is agglomerative hierarchical clustering algorithm. Specifically, the SL (single-linkage) agglomerative hierarchical clustering is similarly used as a consensus function by EAC-SL [22], MULTI-K [12] and FKNNCE [85] cluster ensemble methods. In addition, the AL (average-linkage) clustering has also been exploited to create the final data partition by EAC-AL [22] and CC$_{HC}$ [51] methods.

Hierarchical Clustering on Normalized Edges (HCNE): Instead of applying a conventional hierarchical clustering to the CO matrix directly, the HCNE method of Li et al. [86] formulates a new hierarchical clustering procedure based on the concept of 'normalized edges'. At the outset, an undirected graph $G = (V, E)$ is created such that each vertex $v_i \in V$ corresponds to data point $x_i, \in X$, whilst an unweighted edge $e_{ij} \in E$ connecting vertices $v_i, v_j \in V$ exists only when $CO(x_i, x_j) > \theta$. θ is a user-defined parameter in the range of [0, 1]. The measure $Edge(C_p, C_q)$ is specified as the number of distinct edges between data points in clusters C_p and C_q. Given that $x_i, x_j \in X$,

$$Edge(C_p, C_q) = \sum_{\forall\ x_i \in C_p} \sum_{\forall\ x_j \in C_q} I(e_{ij}), \tag{50}$$

where $I(e_{ij}) = 1$ if $e_{ij} \in E$, 0 otherwise.

By following Guha et al. [87], this initial measure is modified to 'normalized edges (NE)' such that it is robust to clusters' sizes and shapes.

$$NE(C_p, C_q) = \frac{Edge(C_p, C_q)}{(n_p + n_q)^{1+f(\theta)} - n_p^{1+f(\theta)} - n_q^{1+f(\theta)}}, \tag{51}$$

where n_p denotes the number of data points in cluster C_p, while $n_p^{1+f(\theta)}$ and $n_q^{1+f(\theta)}$ are the expected number of edges within clusters C_p and C_q, respectively. Also,

$$f(\theta) = \frac{1 - \theta}{1 + \theta} \tag{52}$$

Having obtained such means to estimate the similarity between clusters, a conventional hierarchical clustering process is used with the two most similar clusters being merged in each iteration.

Fuzzy Ensemble Clustering (FEC): Unlike many cluster ensemble methods that focus on combining the results of crisp clustering, the FEC model of Avogadri and Valentini [62] has been introduced for aggregating soft data partitions each of which is obtained by applying a fuzzy clustering algorithm (such as fuzzy c-means; [88,89]). For a partition $\pi_g \in \Pi$ where $\pi_g = \{C_1^g, \ldots, C_{k_g}^g\}$, $U_p^g(x_i) \in [0, 1]$ is the membership degree of data point $x_i \in X$ belonging to cluster $C_p^g \in \pi_g$, provided that $\sum_{p=1}^{k_g} U_p^g(x_i) = 1$.

Following that, the CO alike matrix, CO', is created from soft data partitions $\pi_g, g = 1 \ldots M$, such that $CO'(x_i, x_j)$ is estimated by

$$CO'(x_i, x_j) = \frac{1}{M} \sum_{g=1}^{M} \sum_{p=1}^{kg} \tau(U_p^g(x_i), U_p^g(x_j)), \tag{53}$$

where τ is a fuzzy t-norm operator. In particular to the study of Avogadri and Valentini [62] an algebraic product is selected as t-norm, i.e., $\tau(U_p^g(x_i), U_p^g(x_j)) = U_p^g(x_i) \times U_p^g(x_j)$.

To generate the soft final partition $\pi^* = \{C_1^*, \ldots, C_K^*\}$, the fuzzy c-means technique is applied to clustering rows of the CO' matrix. The membership degree that data point x_i belongs to cluster C_q^* is denoted as $U_q^*(x_i)$. Given this, a crisp clustering result can be achieved by determining the most appropriate cluster $C_q^* \in \pi^*$ to which each data point should belong. Formally, $x_i \in C_q^*$ if

$$U_q^*(x_i) = \max_{s=1 \ldots K} U_s^*(x_i) \tag{54}$$

3.4. Graph-based approach

This family of algorithms makes use of the graph representation to solve the cluster ensemble problem [43–45,58]. In this approach, a weighted graph is first constructed from the clustering ensemble. Then, the graph is partitioned in to K parts to produce the final clustering using any graph partitioning techniques.

Graph-based Consensus Clustering (GCC): This method of Yu et al. [58] transforms the CO matrix into a graph $G = (V, W)$, where V and W are the sets of vertices and weighted edges, respectively. Each vertex $v_i \in V$ corresponds to a specific data point $x_i \in X$, while the weight of edge $w_{ij} \in W$ connecting vertices $v_i, v_j \in V$ equals to the value of entry $CO(x_i, x_j)$. The resulting graph is undirected such that $w_{ij} = w_{ji}, \forall v_i, v_j \in V$. In order to obtain the final clustering π^*, the GCC approach applies the normalized cut algorithm [90] to the graph G.

Cluster-based Similarity Partitioning Algorithm (CSPA): Similar to GCC, the CSPA method [45] also creates a similarity graph $G = (V, W)$ from the CO matrix. Afterwards, a multi-level k-way graph partitioning called METIS [91] is used to partition the graph G into K clusters of roughly equal size. METIS handles multi-constraint graph partitioning in three phases: (i) coarsening phase, the size of the graph is successively decreased; (ii) initial partitioning phase, a k-way partition of the smaller graph is computed; and (iii) uncoarsening phase, the partitioning is successively refined as it is projected to the larger graphs. More details of METIS can also be found in the reports of Karypis and Kumar [92] and Karypis and Kumar [93].

Shared Nearest Neighbours-Based Combiner (SNNC): Another graph-based method that also makes use of the CO matrix is developed by Ayad and Kamel [46]. It first modifies a given similarity matrix such that only entries of a value above the pre-specified threshold μ are maintained. In other words, for any $x_i, x_j \in X$, $CO(x_i, x_j)$ remains unchanged if $CO(x_i, x_j) > \mu$, 0 otherwise. Following that, data point x_j belongs to a set of nearest neighbours $N_{x_i} \subset X$ of data point x_i if $CO(x_i, x_j) > 0$.

A weighted graph $G = (V, W)$ is then created where V is a set of weighted vertices and W is a set of weighted edges. In particular, the weight of edge w_{ij} connecting vertices $v_i, v_j \in V$ (corresponding to data points x_i and x_j) can be estimated by

$$w_{ij} = 2 \times \frac{|N_{x_i} \cap N_{x_j}|}{|N_{x_i}| + |N_{x_j}|}, \tag{55}$$

where $|A|$ denotes the size of set A. In addition, each vertex $v_i \in V$ that represents data point $x_i, i = 1 \dots N$ is given the following weight:

$$v_i = \frac{N_{x_i}}{N} \tag{56}$$

Similar to the CSPA method, SNNC also exploits METIS to generate the final data partition π^*. Note that, to reflect the majority of voting amongst ensemble members, it is suggested by Ayad and Kamel [46] that the threshold μ should be around 0.5.

Hyper-Graph Partitioning Algorithm (HGPA): Based on the binary cluster-association (BA) matrix (see an example in Fig. 2(d)), HGPA [45] constructs a hyper-graph, where vertices represent data points and the same-weighted hyper-edges represent clusters in the ensemble. Then, HMETIS [94] is applied to partition the underlying hyper-graph into K parts with roughly of the same size.

Meta-Clustering Algorithm (MCLA): This graph-based method [45] generates a graph that represents the relationships among clusters in the ensemble. In this meta-level graph, each vertex corresponds to each cluster in the ensemble and each edge's weight between any two cluster vertices is computed using the binary Jaccard measure (i.e., the ratio of the intersection to the union of the sets of objects belonging to the two clusters). METIS is also employed to partition the meta-level graph into K meta-clusters. Effectively, each data point has a specific association degree to each meta-cluster. This can be estimated from the number of original clusters, to which the data point belongs, in the underlying meta-cluster. The final clustering π^* is produced by assigning each data point to the meta-cluster with which it is most frequently associated (i.e., with the highest association degree).

Hybrid Bipartite Graph Formulation (HBGF): HBGF [44] is introduced with the purpose to improve the previous models of CSPA and MCLA that considers only either the associations between data points or those amongst clusters. In particular, a bipartite graph $G = (V, W)$ is used by the HBGF method, where $V = V^X \cup V^C$ is the set of vertices corresponding to data points (V^X) and clusters (V^C). The weight of edge $w_{ij} \in W$ between vertices $v_i, v_j \in V^X$ or that of edge w_{pq} connecting $v_p, v_q \in V^C$ is zero. On the other hand, the weight of edge w_{ip} connecting vertices $v_i \in V^X$ and $v_p \in V^C$ can be obtained from the BA matrix.

$$w_{ip} = BA(x_i, C_p), x_i \in X, C_p \in \Pi \tag{57}$$

This graph is undirected such that w_{ip} is equivalent to w_{pi}. The spectral graph partitioning algorithm of Ng et al. [95] and METIS are exploited to obtain the final clustering from this graph.

Weighted Similarity Partitioning Algorithm (WSPA): This cluster ensemble technique is developed as the by-product of a new soft-subspace clustering model [43]. It creates a BA-alike information matrix, *WDM*, from which the final clustering can be effectively determined. Unlike the conventional BA in which each entry is determined by the underlying label assignment, an entry $WDM(x_i, cl)$ is estimated from the distance between data point $x_i \in X$ and centre of the cluster $cl \in \Pi$. For each base clustering $\pi_g \in \Pi$ where $\pi_g = \{C_1^g, \dots, C_{k_g}^g\}$, the value of $WDM(x_i, cl), \forall cl \in \pi_g$ can be defined by

$$WDM(x_i, cl) = \frac{D_i - d(x_i, \overline{cl}) + 1}{k_g D_i + k_g - \sum_{\forall cl' \in \pi_g} d(x_i, \overline{cl'})}, \tag{58}$$

where k_g denotes the number of clusters in the base clustering $\pi_g \in \Pi$ and $d(x_i, \overline{cl})$ is the distance between data point x_i and $\overline{cl}$, that is centre (or centroid) of the cluster cl. In addition, D_i can be specified as

$$D_i = \max_{\forall cl \in \pi_g} d(x_i, \overline{cl}) \tag{59}$$

According to Domeniconi and Al-Razgan [43], the distance $d(x_i, \overline{cl})$ can be defined as follows, where D is the number of attributes, $w_{cl,s} \in [0, 1]$ is the weight of the sth attribute that is specific to the cluster $cl \in \pi_g$, $x_{i,s}$ denotes value of the sth attribute of data x_i, and $\overline{cl}_s$ denotes the sth attribute value of the cluster centre $\overline{cl}$.

$$d(x_i, \overline{cl}) = \sqrt{\sum_{s=1}^{D} w_{cl,s}(x_{i,s} - \overline{cl}_s)^2} \tag{60}$$

For any $cl \in \pi_g$,

$$\sum_{s=1}^{D} w_{cl,s} = 1 \tag{61}$$

The set of cluster-specific weights is systematically obtained from a so-called 'soft subspace clustering' technique such as LAC (Locally Adaptive Clustering; [96]). This method extends the conventional k-means by iteratively revising cluster-specific attribute weights that allow more compact clusters to be obtained. Let $X = \{x_1, \dots, x_N\}$ be a set of data points and each object $x_i = (x_{i,1}, \dots, x_{i,D}), i = 1 \dots N$ is characterized by a set of attribute $F = \{f_1, \dots, f_D\}$. LAC searches for the partition $\pi = \{C_1, \dots, C_k\}$ of X into k clusters that minimizes the following objective function.

$$J(U, Z, W) = \sum_{l=1}^{k} \sum_{s=1}^{D} \left[w_{l,s} O_{l,s} + h\, w_{l,s} \log w_{l,s} \right], \tag{62}$$

where

$$\sum_{l=1}^{k} u_{i,l} = 1 \tag{63}$$

and $U \in R^{N \times K}$ is a matrix in which each entry $u_{i,l}$ represents a membership degree that data point $x_i \in X$ has with cluster $C_l \in \pi$ ($u_{i,l} \in \{0, 1\}$ and $u_{i,l} \in [0, 1]$ for crisp and soft clustering, respectively). In addition, $Z = \{z_1, \dots, z_k\}$ denotes a vectors representing the centroids of k clusters, $|C_l|$ is the cardinality of the cluster C_l, with $O_{l,s}$ being defined by the following.

$$O_{l,s} = \frac{1}{|C_l|} \sum_{\forall x_i \in C_l} (x_{i,s} - z_{l,s})^2, \tag{64}$$

while $h \geq 0$ is the constant that controls the relative differences between dimension weights. In each iteration of the k-means alike process, W is updated by

$$w_{l,s} = \frac{\exp\left(\frac{-O_{l,s}}{h}\right)}{\sum_{t=1}^{D} \exp\left(\frac{-O_{l,t}}{h}\right)} \tag{65}$$

In the study of Domeniconi and Al-Razgan [43], the resulting *WDM* matrix is used to design the graph-based ensemble methods of WSPA (Weighted Similarity Partitioning Algorithm) and WBPA (Weighted Bipartite Partitioning Algorithm). Particularly to WSPA, it creates an $N \times N$ pairwise-similarity matrix S_g from a given *WDM*, for each clustering $\pi_g, g = 1 \dots M$ where $\pi_g = \{C_1^g, \dots, C_{k_g}^g\}$. Let P_i^g be a vector of entries in the *WDM* matrix that corresponds to data point $x_i \in X$ and clusters in π_g.

$$P_i^g = \left(WDM(x_i, C_1^g), \dots, WDM(x_i, C_{k_g}^g) \right) \tag{66}$$

Each entry $S_g(x_i, x_j)$ that represents the similarity between data points $x_i, x_j \in X$ can be estimated by the following cosine measure:

$$S_g(x_i, x_j) = \frac{P_i^g P_j^g}{\|P_i^g\|\|P_j^g\|}, \tag{67}$$

where $\|P_i^g\|$ is estimated by

$$\sqrt{WDM(x_i, C_1^g)^2 + \cdots + WDM(x_i, C_{k_g}^g)^2} \tag{68}$$

For an ensemble of M clustering, the overall similarity measures is presented by the matrix S, which can be specified as

$$S = \frac{1}{M}\sum_{g=1}^{M} S_g \tag{69}$$

Similar to CSPA, this similarity matrix, S, is transformed to a weighted graph, which is later partitioned into K clusters using METIS [92].

Weighted Bipartite Partitioning Algorithm (WBPA): Unlike the WSPA method, WBPA transforms the underlying *WDM* matrix to a bipartite graph, which is partitioned into clusters using spectral graph partitioning (SPEC; [95]) or METIS. Following the representation scheme used by HBGF, the bipartite graph $G = (V, W)$ consists of the set of vertices $V = V^X \cup V^C$ corresponding to data points (V^X) and clusters (V^C), and the set of weighted edges W. The weight of edge $w_{ij} \in W$ between vertices $v_i, v_j \in V^X$ or w_{pq} between $v_p, v_q \in V^C$ is zero, whilst the weight of edge w_{ip} connecting vertices $v_i \in V^X$ and $v_p \in V^C$ can be obtained directly from the *WDM* matrix, i.e., $w_{ip} = WDM(x_i, C_p), x_i \in X, C_p \in \Pi$.

Connected-Triple Similarity (CTS) Algorithm: To enhance the performance of CSPA [45] that makes use of the conventional CO matrix, the link-based algorithms of Connected-Triple Similarity (CTS) and SimRank-based Similarity (SRS) have been introduced to refine the evaluation of similarity measures among data samples [54]. Despite its simplicity, the CO matrix fails drastically to handle a large number of 'unknown' relations, each of which is presented with '0'. This information matrix can expose only a small proportion of pairwise similarity between data points, which may be better discovered by bringing in additional information regarding relations between clusters in an ensemble. As a result, S_{CTS} and S_{SRS} matrices are established with substantially less unknown entries, as compared to the CO counterpart.

Specifically to the CTS method, the similarity between clusters in ensemble Π are assessed from the weighted graph $G = (V, W)$, where V is the set of vertices each representing a cluster and W is a set of weighted edges between clusters. Formally, the weight assigned to the edge $w_{pq} \in W$, that connects clusters $C_p, C_q \in V$, is estimated by

$$w_{pq} = \frac{|L_p \cap L_q|}{|L_p \cup L_q|}, \tag{70}$$

where $L_p \subset X$ denotes the set of samples belonging to cluster $C_p \in V$.

Given this network formalism, the Weighted Connected-Triples (WCT) measure is employed to disclose the similarity between any pair of clusters. It extends the Connected-Triple method of Reuther and Walter [97] that has been originally developed to identify ambiguous author names within publication databases. In particular, the similarity of any $C_p, C_q \in V$ can be estimated by counting the number of Connected-Triples (i.e. triples) they are part of. Formally, a triple, $Triple = (V_{Triple}, W_{Triple})$, is a subgraph of G containing three vertices $V_{Triple} = \{C_p, C_q, C_o\} \subset V$ and two non-zero edges $W_{Triple} = \{w_{po}, w_{qo}\} \subset W$, with $w_{pq} = 0$. This simple counting is sufficient for any indivisible object, e.g. sample, but becomes inappropriate for clusters, i.e., a set of samples. As a result, the WCT measure of clusters $C_p, C_q \in V$ with respect to each triple $C_o \in V$, is estimated as

$$WCT_{pq}^{o} = \min(w_{po}, w_{qo}), \tag{71}$$

where $w_{po}, w_{qo} \in W$ are weights of the edges connecting clusters C_p and C_o, and clusters C_q and C_o, respectively. The count of all triples $(1 \ldots \alpha)$ between clusters C_p and C_q can be calculated as follows:

$$WCT_{pq} = \sum_{o=1}^{\alpha} WCT_{pq}^{o} \tag{72}$$

Then, the similarity between clusters C_p and C_q can be estimated by

$$Sim_{WCT}(C_p, C_q) = \frac{WCT_{pq}}{WCT_{max}} \times DC, \tag{73}$$

where WCT_{max} is the maximum WCT_{st} value of any two clusters $C_s, C_t \in V$ and $DC \in [0, 1]$ is a constant decay factor (i.e. confidence level of accepting two non-identical clusters as being similar).

Following that, the S_{CTS} matrix is generated as follows. For each ensemble member $\pi_g, g = 1 \ldots M$, the similarity between samples $x_i, x_j \in X$ is estimated as

$$S_g(x_i, x_j) = \begin{cases} 1 & if\ C^g(x_i) = C^g(x_j) \\ Sim_{WCT}(C^g(x_i), C^g(x_j)) & otherwise \end{cases} \tag{74}$$

Each entry in the S_{CTS} matrix can be computed by

$$S_{CTS}(x_i, x_j) = \frac{1}{M}\sum_{g=1}^{M} S_g(x_i, x_j) \tag{75}$$

Similar to CSPA, the similarity matrix, S_{CTS}, is transformed to the weighted graph, from which the final clustering result π^* is generated using METIS.

SimRank-based Similarity (SRS) Algorithm: Besides considering a cluster ensemble as a network of clusters only (as for the CTS algorithm), the SRS method [54] utilizes a bipartite graph representation and SimRank measure [98] to reveal hidden relations. Given a cluster ensemble Π, the bipartite graph $G = (V, W)$ can be constructed, where $V = V^X \cup V^C$ is a set of vertices representing both data samples (V^X) and clusters (V^C) in the ensemble, and W denotes a set of edges between samples and the clusters to which they are assigned. In particular, the weight of edge w_{ip} connecting sample $x_i \in X$ and cluster $C_p \in \Pi$ is 1 if $x_i \in C_p$, 0 otherwise.

Let $S_{SRS} \in R^{N\times N}$ and $S'_{SRS} \in R^{P\times P}$ be the pairwise similarity matrices amongst N samples and P clusters, respectively. An entry $S_{SRS}(x_i, x_j)$ that represents the similarity between samples $x_i, x_j \in X$ equals to 1 if $x_i = x_j$, otherwise

$$S_{SRS}(x_i, x_j) = \frac{DC}{|N_{x_i}||N_{x_j}|}\sum_{\forall C_p \in N_{x_i}}\sum_{\forall C_q \in N_{x_j}} S'_{SRS}(C_p, C_q), \tag{76}$$

where $DC \in [0, 1]$ is a constant decay factor and $N_{x_i} \subset V^C$ denotes the set of cluster vertices connecting to the sample vertex $x_i \in V^X$, i.e. $w_{ip} = 1, \forall C_p \in N_{x_i}$.

Likewise, any entry $S'_{SRS}(C_p, C_q)$ is 1 if $C_p = C_q$, otherwise

$$S'_{SRS}(C_p, C_q) = \frac{DC}{|N_{C_p}||N_{C_q}|}\sum_{\forall x \in N_{C_p}}\sum_{\forall x' \in N_{C_q}} S_{SRS}(x, x'), \tag{77}$$

In fact, both S_{SRS} and S'_{SRS} matrices can be correctly achieved through the iterative refinement process. In particular to the S_{SRS} matrix,

$$\lim_{r\to\infty} S_{SRS_r}(x_i, x_j) = S_{SRS}(x_i, x_j) \tag{78}$$

Let $S_{SRS_r}(x_i, x_j)$ be a similarity degree between $x_i, x_j \in X$ at the rth iteration, the estimation of the similarity score at the next iteration $r+1$ is defined as

$$S_{SRS_{r+1}}(x_i, x_j) = \frac{DC}{|N_{x_i}||N_{x_j}|} \sum_{C_p \in N_{x_i}} \sum_{C_q \in N_{x_j}} S'_{SRS_r}(C_p, C_q) \tag{79}$$

Note that, initially, $S_{SRS_0}(x_i, x_j) = 1$ if $x_i = x_j$ and 0 otherwise. This updating procedure is applicable to the case of S'_{SRS} matrix, where $S'_{SRS_0}(C_p, C_q) = 1$ if $C_p = C_q$, else 0. Once the similarity matrix S_{SRS} is obtained, it is transformed to the weighted graph that is similar to those used by CSPA and CTS techniques. Again, METIS is exploited to partition this graph into the final clustering result.

Link-based Cluster Ensembles (LCE) Algorithm: To improve the efficiency of previous link-based methods (CTS and SRS) to cluster ensemble problem, LCE [20] focuses on refining the BA matrix that is less expensive to build than the pairwise similarity alternative. It extends the HBGF method that is based on information presented in the conventional BA matrix, where each entry $BA(x_i, C_p) \in \{0, 1\}$ represents a 'crisp' association degree between sample $x_i \in X$ and cluster $C_p \in \Pi$. Similar to the case of CO matrix, a large number of entries in the BA are 'unknown', each presented with '0'. These hidden or unknown associations can be estimated upon the similarity amongst clusters, discovered from a link network of clusters.

In particular, the refined cluster-association (RA) matrix is put forward as the enhanced variation of the original BA. Its aim is to approximate value of unknown associations ('0') from known ones ('1'), whose association degrees are preserved within the RA. In other words,

$$BA(x_i, C_p) = 1 \rightarrow RA(x_i, C_p) = 1 \tag{80}$$

For each clustering $\pi_g, g = 1 \ldots M$ and their corresponding clusters $C_1^g, \ldots, C_{k_g}^g$ (where k_g is the number of clusters in the clustering π_g), the association degree $RA(x_i, cl) \in [0, 1]$ that sample $x_i \in X$ has with each cluster $cl \in \{C_1^g, \ldots, C_{k_t}^g\}$ is estimated as follows:

$$RA(x_i, cl) = \begin{cases} 1 & if\ cl = C^g(x_i) \\ Sim(cl, C^g(x_i)) & otherwise \end{cases} \tag{81}$$

where $C^g(x_i)$ is a cluster label (corresponding to a particular cluster of the clustering π_g) to which the sample x_i belongs. In addition, $Sim(C_p, C_q) \in [0, 1]$ denotes the similarity between clusters $C_p, C_q \in \Pi$, which can be discovered using the Weighted Connected-Triples algorithm (see Eqs. (70)–(73), for details).

Having obtained the RA matrix, a graph-based partitioning method is exploited to obtain the final clustering. Similar to HBGF, this consensus function requires the underlying matrix to be initially transformed into a weighted bipartite graph $G = (V, W)$, where $V = V^X \cup V^C$ is the set of vertices corresponding to samples (V^X) and clusters (V^C). The weight of edge $w_{ij} \in W$ between $v_i, v_j \in V^X$ or that of edge w_{pq} between $v_p, v_j \in V^C$ is zero. On the other hand, the weight of edge w_{ip} connecting vertices $v_i \in V^X$ and $v_p \in V^C$ can be obtained from the RA matrix, i.e. $w_{ip} = RA(x_i, C_p), x_i \in X, C_p \in \Pi$. The spectral graph partitioning algorithm [95] is finally applied to G to acquire π^*.

4. Recent extensions and applications

Soon after 2010, a large number of research studies have published new concepts and findings related to several issues of cluster ensemble. Some introduce theoretical improvement and extensions to the previous approaches to ensemble generation, representation and consensus clustering. Others focus on the application side, where existing methods are exploited for real problems and different data-mining tasks. The section is to provide details and a useful insight of these exciting developments.

4.1. Theoretical improvement and extensions

The following three subsections explore the literature for new developments of conceptual components within cluster ensemble, which commonly aim to promote the quality of final clustering results.

4.1.1. Ensemble generation

It is known that the goodness of the ensemble decision is highly subjected to both diversity within the ensemble and accuracy of those ensemble members. Also, Fern and Lin [99] have recommended to form a smaller but better-performing cluster ensemble with a subset of members, than using all primary alternatives. In addition to the collection of general approaches to ensemble generation discussed in Section 2, this part provides details of more up-to-date methods to reach the aforementioned goal.

- **Validity index based generation**: Similar to the study of Fern and Lin [99], Alizadeh et al. [100] introduce an ensemble clustering framework, which makes use of a subset of initial members in the ensemble, instead of employing all like before. As such, the quality metric of Normalized Mutual Information or NMI is exploited for the determination of these target clusterings. Of course, setting an appropriate NMI threshold is data dependent and requires domain knowledge. About the same time, Zhang et al. [101] make use of Adjusted Rand Index (ARI) to control the formation of cluster ensemble. In particular, this classical validity metric is generalized to new measures of ARImp and ARImm. The former compares the similarity between base clusterings and the consensus matrix that summarizes the entire ensemble, while the other computes the similarity between any pair of primary partitions. The NMI metric is also exploited by Parvin and Minaei-Bidgoli [102] to determine a good subset of base clusterings, which are initially generated using the weighted locally adaptive clustering (WLAC) algorithm. Following that, a new asymmetric criterion named Alizadeh–Parvin–Moshki–Minaei (APMM) has been brought forward as the alternative to NMI to control the process of ensemble selection [103]. Likewise, the measure of cluster stability and dataset simplicity are coupled to assess the quality of subsets of base partitions [104].

  In addition to the aforementioned, the comparative study of Naldi et al. [105] reports the use of different relative clustering validity indexes to select ensemble members. A major finding reveals that each index can be more suitable for a specific data conformation. As such, a combination of distinct relative indexes is proposed based on the intuition that the majority of indices may compensate the poor performance caused by some within the group. Another approach called Cluster Ensemble Selection (CES) is recently proposed to identify good clusterings that should be parts of the desirable ensemble [106]. A collection of pre-generated clusterings are represented as a multiplex network, in which slices are formed based on clustering dissimilarity indices. Provided this, a community detection algorithm is deployed to deliver communities in the aforementioned slices. Then, for each community, select the best clustering with respect to quality and diversity indexes. These are finally combined to form the target ensemble.
- **Heuristic based generation**: One of the recent extensions model the ensemble generation based on a concept called The Wisdom of Crowds [107]. It is a phenomenon founded in social science that suggests criteria applicable to group behaviour. Intuitively, with these criteria being satisfied, the group decisions may often be better than those of individual

members. As a result, Wisdom of Crowds Cluster Ensemble (WOCCE) is introduced with the capability to analyse conditions necessary for an ensemble to exhibit its collective wisdom. These include decentralization condition for generating base clusterings, independence condition among base algorithms, and diversity condition within the ensemble. Besides, Jia et al. [108] also implement a new selection strategy based on the rule of nearest neighbours. Their method called SELective Spectral Clustering Ensemble (SELSCE) promotes the diversity through random scaling parameter, Nyström approximation and random initialization of k-means. Before the application of neighbour-based heuristic, the set of primary decisions are filtered using a measure integrating diversity and quality.

- **Hierarchical clustering based selection**: Since the relationship between diversity and quality is uncertain, Akbari et al. [109] has proposed the Hierarchical Cluster Ensemble Selection (HCES) method and the diversity measure to determine the effect of diversity and quality on final results. In particular, HCES employs single-linkage, average-linkage, and complete-linkage agglomerative techniques to select members hierarchically. It is reported that the proposed diversity metric leads to more diverse members than that of the pairwise diversity counterpart. This claim is supported with empirical studies on two benchmark ensemble methods of CSPA and HGPA.
- **Soft clustering based generation**: Parvin and Minaei-Bidgoli [110] is one among several researchers that approach the ensemble generation through fuzzy clustering. With Fuzzy Weighted Locally Adaptive Clustering (FWLAC) algorithm, it is possible to produce a diverse and accurate ensemble using the weighting scheme for differentiating informative and uninformative features. Specific to the problem of tumour clustering, Yu et al. [111] propose the random double clustering based fuzzy cluster ensemble framework (RDCFCE). It first creates a set of representative features using a randomly selected clustering algorithm in the ensemble. Then, data points are assigned to appropriate clusters based on the grouping results. These assignments are turned into a fuzzy consensus matrix, from which the final decision is obtained using the normalized cut algorithm.

  As the core part of granular computing, the rough set theory that deals with dealing with uncertain or vague information, has also been applied for the problem of cluster ensemble [112]. A work published last year by Hu et al. [113] demonstrates such an idea, where a hierarchical cluster ensemble model based on knowledge granulation is proposed with a novel rough distance to measure the dissimilarity between base partitions.
- **Model-initialization based generation**: For an homogeneous ensemble like that of k-means members, model initialization plays a crucial part in producing diversity. Wu et al. [114] has suggested a number of desired conditions for K-means-based consensus clustering (KCC), including the criteria for initialization. Another interesting concept of co-initialization [115] has been investigated, with the results suggesting that the quality of clusterings can often be improved when a set of diverse clustering techniques provides initializations for each other.
- **Re-sampling**: The initial work of Fern and Lin [99] has been generalized by the SELective Spectral Clustering Ensemble (SELSCE) method [116]. Primary components are first created using spectral clustering (SC), with Nyström approximation to perturb the results of SC. Then, these base decisions are manipulated through the bagging process, which is usually applied in supervised learning. At last, the components are ranked by aggregating multiple NMI or ARI values, which have been obtained from random comparisons between individual components and the consensus matrix. Similarly inspired by bagging and boosting algorithms in classification, other studies by Parvin et al. [117] and Minaei-Bidgoli et al. [118] examines the non-weighing and weighing-based sampling approaches to ensemble generation. And recently, this line of research has continued to cover a new framework called Weighted-Object Ensemble Clustering (WOEC) with the co-association matrix being employed to represent the ensemble information [119].

  In addition, a novel cluster generation method based on random sampling or RS-NN is introduced, where the nearest neighbour strategy is adopted to fill the category information of the missing samples [120]. According to the evaluation against a typical random projection method (Random Feature Subset or FS) and another random sampling method (Random Sampling based on Nearest Centroid or RS-NC), it is found that RS-NN is able to produce base clusterings with a good balance between quality and diversity, thus achieving significant improvement over the counterparts. Note that FS usually generates more diverse partitions, while RS-NC delivers high-quality partitions. Yang and Jiang [121] propose a novel hybrid sampling-based clustering ensemble by combining the strengths of boosting and bagging. The base partitions are iteratively created via a hybrid process exhibiting characteristics of both boosting and bagging.
- **Re-using feature selection/transformation techniques**: It is also possible to view clustering solutions as features, such that existing feature selection algorithms can be employed to selection a subset of primary features (or solutions). With this in mind, Yu et al. [122] propose a hybrid clustering solution selection strategy (HCSS) to aggregate different feature selection techniques for identifying the suitable subset of ensemble members. Similar to this work, the use of data transformation operators has also been investigated [123]. In particular, two new data transformation operators are developed to create new datasets in the ensemble. These are known as probabilistic based data sampling operator and probabilistic based attribute sampling operator. Following that, three new random transformation models are proposed, including the random combination of transformation operators in the data dimension, in the attribute dimension, and in both dimensions, respectively.
- **Multiple distance functions and pruning**: Yu et al. [124] introduced a cluster ensemble framework named as AP2CE, which is claimed to be noise immune. This is feasible with the use of affinity propagation algorithm (AP) and multiple distance functions. In that, a set of new data matrices is produced with respect to the subspaces consists of representative attributes obtained by AP. In order to enhance the quality of ensemble, diversity is increased through removing the redundant base partitions [125]. The significance of attribute founded in rough set theory is adopted as a heuristic to select the subset of ensemble members.
- **Multiple data modalities**: Specific to biomedical data analysis, a new method called Complementary ensemble clustering (CEC) is presented as an weighted extension of co-association or CO matrix [126]. In that, base partitions are obtained from separate clusterings of different data modalities, e.g., text and images.

4.1.2. Representation and summarization of multiple clusterings

In addition to the generation aspect, there have been several studies devoting to the topic of representing and summarizing base partitions. These include:

- **String representation**: A collection of new cluster ensemble methods design the problem of combining primary partitions as an optimization process (see the next section on consensus functions for more details). As such, the information among ensemble members is formulated in 0–1 bit strings. With this terminology, Alizadeth et al. [127] introduce a constrained nonlinear objective function called fuzzy string objective function (FSOF), to search for a median partition. This is achieved by maximizing the agreement between the ensemble members and minimizes the disagreement at the same time.
- **Tree representation**: In the attempt to improve the problem with time and memory complexity of CO based methods, Wang [128] proposes a hierarchical structure called a coassociation tree or CA-tree, which is built using the base cluster labels. At a given threshold, the corresponding cut of this tree creates a preliminary partition of the data into disjoint groups or pre-clusters. Then, the CO matrix is derived from the representatives of these groups.
- **Graph representation**: Du et al. [129] argued that existing approaches to represent cluster ensemble is either by multiple co-association pairwise relations or cluster based features. Given this background, a unified framework is put forward to integrate the two representation schemes by means of weighted graph regularized nonnegative matrix factorization. It is also reported that such a combination outperforms both of the individuals, with respect to clustering accuracy and stability. Another work by Huang et al. [130] propose a new approach named as ensemble clustering using factor graph (ECFG). In that, the concept of superobject is founded as a compact and adaptive representation for the ensemble data. Based on probabilistic modelling, the problem with approximated data is formulated as a binary linear programming (BLP) problem. In order to solve this optimization, an efficient solver based on factor graph is established.
- **Data fragments**: Instead of applying the ensemble clustering to the entire data points, it is feasible to separately analyse data fragments that represent subsets of the original data [131]. Of course, this help scaling up the model to large datasets. The concept of clustering aggregation or AGG is reused to generate the final results from base partitions of these fragments. It is reported with empirical results that the proposed approach is more efficient than the existing AGG methods (Agglomerative, Furthest, and LocalSearch), without sacrificing the accuracy. See the study of Chung and Dai [132] for a recent extension.
- **Relation and link-based representation**: Wang et al. [133] invent the framework for coupled clustering ensembles (CCE) to overcome the problem of explicating the dependency between base partitions and between data points. In fact, it integrates the two coupling relationships, which are presented as the intra-coupling within one base clustering and the inter-coupling between different base clusterings. Besides this invention, the following two works extend the concept of link-based formation introduced by LCE approach [134]. The first introduces a new method termed WETU that is capable of refining the data cluster association matrix with a link-based similarity measure. Unlike LCE, the matrix is acquired from the similarity of clusters among all base clusterings, not from any specific one. As such, WETU can provide more discriminative information than the original counterpart. The other proposes the use of crowd agreement estimation and multi-granularity link analysis to improve the quality of cluster ensemble [135]. At the outset, base partitions are weighted using the normalized crowd agreement index (NCAI). Following that, the relationship between clusters is explored with the application of source aware connected triple (SACT) similarity, which encodes information regarding common neighbours and the source reliability. Based on these, two novel consensus clusterings are provided, weighted evidence accumulation clustering (WEAC) and graph partitioning with multi-granularity link analysis (GP-MGLA), respectively.
- **Extensions to CO matrix**: Several attempts have been made to extend the representation and application of the so-called CO matrix. Some of these can be summarized in this heading. Duarte et al. [136] make use of the information represented in CO matrix to determine degrees of confidence associated with data points. These confidence values dictate the likelihood of data points being included in the consensus clustering process. Another work by Lourenco et al. [137] identifies the fact that a differentiation among the base partitions can lead to improved quality of the consensus clustering. In particular, the framework of CO matrix is modified to implement a weighting mechanism that represents the importance of different ensemble members. With a similar intuition, Ren et al. [138] also present the Weighted-Object Ensemble Clustering (WOEC) method, which embed the weights associated to data points in the conventional CO matrix. These weights are initially obtained through the application of boosting during the clustering process. After that, another approach to representing weights in the CO matrix is introduced via the notion of competence, which reflects the quality of different algorithms used to create base members [139]. The efficiency of this method is demonstrated with Monte-Carlo modelling.

  A path-based approach is developed to refine the CO matrix such that distinct contributions of individual data points and base partitions to the ensemble can be represented [140]. The path-based similarity allows more global information of the cluster structure to be incorporated into the matrix, from which the final clustering is generated using spectral clustering. Built upon the paradigm of CO matrix, Louren et al. [141] devise the new probabilistic approach to assign data points to clusters. This is achieved via minimizing a Bregman divergence between the observed co-association frequencies and the corresponding co-occurrence probabilities expressed as functions of the unknown assignments. Unlike the works described so far, another attempt by Liu et al. [142] focuses on the efficiency aspect of CO matrix based solution. As a result, an efficient Spectral Ensemble Clustering method is proposed, where it is theoretical equivalent to weighted k-means process, thus vastly reducing the algorithmic complexity.

4.1.3. *Consensus clustering*

According to the four general categories of cluster ensemble approaches, this part will provide details of consensus functions recently developed in the literature.

- **Direct approach**: In contrary to the conventional voting method that is compatible with hard base clusterings, Wang et al. [143] propose an alternative to aggregate soft partitions. It is called Soft-Voting Clustering Ensemble (SVCE), which provides better flexibility and generalization than the hard counterpart. Similarly, Sevillano et al. [144] introduces a new consensus function to consolidate the outcomes of multiple fuzzy clusterings into a single fuzzy partition. This is achieved through the application of positional and confidence voting techniques. Another work by Zhang et al.

[145] focuses on the re-labelling process within the two-layer clustering framework.

Several recent extensions represent the problem of consensus clustering as an optimization or search for the median partition. For that, a framework to learn a low-rank matrix via optimization is examined [146], with a block coordinate descent algorithm being employed to solve the problem. Li et al. [147] make use of simulated annealing method named BV-RSA to solve the problem of ensemble clustering. In addition, intracluster criteria such as Minimum-Sum-of-Squares-Clustering (MSSC) is also exploited to formulate the objective function [148]. Later, Chatterjee and Mukhopadhyay [149] models this as a multiobjective optimization problem and a multiobjective evolutionary algorithm (MOECEA). The final clustering is generated from input partitions by optimizing two criteria simultaneously. One is to maximize the similarity of the resultant clustering with all the input clusterings. The other minimizes the standard deviation among the similarity scores. This can help to prevent the evolving solution to be very similar with one of the input clusterings. Besides, Gullo et al. [150] extends the concept of Projective Clustering Ensemble (PCE), where a single-objective formulation is effective to allow both sample-based and feature-based cluster representations to be jointly considered.

Franek and Jiang [151] reduce the complexity of cluster ensemble to the well-known Euclidean median problem. This is solved by the Weiszfeld algorithm and an inverse transformation that maps the Euclidean median back into the clustering domain. Besides, Bhatnagar et al. [152] claim to obtain robust clustering using discriminant analysis. It kicks off with re-labelling input partitions using the Hungarian algorithm, followed by applying discriminant analysis to construct of a label matrix. At last, clustering scheme is refined to deliver robust and stable outcome. Along this line of research, some studies complement the aforementioned with the reduction of search space. One of these attempts to find the best subspace to derive the consensus partition [153]. In addition, Vega-Pons and Avesani [154] introduce a new pruning technique that allows a dramatic reduction of the search space.

- **Feature based approach**: At first, Lock and Dunson [155] propose the BCC (Bayesian consensus clustering) as an integrative statistical model, which combines input clusterings of the objects from different data sources. This is applied to the problem of identifying subtypes of breast cancer tumour samples, based on public data from The Cancer Genome Atlas. Following that, the Gaussian mixture model based cluster structure ensemble framework (GMMSE) is presented as a novel probabilistic approach [156]. In particular, GMMSE employs a number of Gaussian mixture models to capture cluster structures embedded in the data. Through the process of Expectation Maximization (EM), components of the Gaussian mixture models are estimated and then viewed as new data samples. These are used to create the matrix representing the relations among components. In that, the Bhattacharyya distance function is used to calculate the similarity between two components corresponding to their respective Gaussian distributions. Lastly, GMMSE builds a graph to represent new data samples and the aforementioned matrix, and looks for the most representative cluster structure.

  In addition to the existing works belonging to this category, the theory of belief functions is introduced to the problem of cluster ensemble [157]. A number of belief functions can be defined on the lattice of interval partitions of samples to represent degree of confidence. Provided this, the consensus belief function is obtained using a suitable combination rule. Likewise, Wu et al. [158] introduce a new approach that utilizes Dempster–Shafer (DS) evidence theory and Gaussian Mixture Modelling (GMM) technique to combine the base partitions. Another group of new consensus methods concentrates on developing a new proximity metric that can be effective at ensemble-level for summarizing similarities among samples. See the studies of Zheng et al. [159] and Aidos and Fred [160] for examples.
- **Pairwise similarity based approach**: One of the most exciting works that adopt concepts invented in other fields is the model termed Cluster Forests or CF [161]. Implied by the name, it is inspired by the success of Random Forests (RF) in the context of classification. CF aims to obtain good local partitions through randomly probing a high-dimensional data. Based on a cluster quality measure kappa, CF gradually obtains improved local clustering in a manner that resembles RF tree growth. Another work on Dual-Similarity Clustering Ensemble (DSCE) initially establishes core clusters based on similarity among objects, then clusters may be merged in accordance with their member-based similarity. Besides these, there have been a vast amount of applications of CO matrix or pairwise similarity scheme (see the next section for more details).
- **Graph based approach**: In the work of Xiao et al. [162], an ensemble is created using multiple trials of CHAMELEON, with a CO matrix being employed to summarize ensemble information. Then, the matrix is modelled as a similarity graph, to which METIS is applied to acquire p sub-graphs, where $p \gg k$. After that, these are combined along the process of hierarchical clustering, to get the final clustering. A similar model has also been introduced by Mimaroglu and Erdil [163], with a specific advantage of obtaining the number of clusters in the final partition automatically. Based on the terminology and application for image segmentation, Abdala et al. [164] has adapted a random walker (RW) algorithm to work with cluster ensembles. It first generates a graph representation of the ensemble, from which the similarity between objects can be inferred using the RW technique.
- **Other new approaches**: Apart from the extensions belonging to the four conventional families, there are still several new approaches worth mentioning here. One of these is presented as the Gravitational ensemble clustering (GEC) method, which is designed to aggregate the results obtained from weak algorithms like k-means [165]. Moreover, Du et al. [166] propose a self-supervised learning framework for the problem of cluster ensemble. In particular, base partitions are treated as pseudoclass labels, each of which a classifier can be learned. With this, the relationships between these input partitions can be exhibited by adding priors to the parameters of the corresponding classifiers. Yet another concept called enhanced splitting merging awareness tactics (E-SMART) is employed specifically to determine the appropriate number of clusters, which remains a major problem to many state-of-the-art consensus clustering methods [167].

In 2016, Teng et al. [168] publish the work on a cluster ensemble framework based on the group method of data handling (CE-GMDH), which consists of three components of an initial solution, a transfer function and an external criterion, respectively. Provided this, a number of models can be formulated using different types of transfer functions and external criteria. Examples of the transfer function include least squares and semidefinite programming.

In the context of image segmentation, Ammour and Alajlan [169] make use of an hybrid cluster ensemble, in which ensemble members are created using fuzzy c-means and fuzzy local information c-means algorithms with different parameter settings. The consensus clustering is performed by the ordered weighted averaging (OWA) method that is normally exploited for group-based decision making. A similar idea of engaging aggregation operators to combine multiple hierarchical clusterings has also been recently reported by Rashedi et al. [170]. In particular, desired properties of different aggregators for hierarchical clustering have been elaborated and assessed. This study is motivated by the initial finding that weighted combination of hierarchical clusterings perform better than other combination methods, e.g., averaging [171]. Again, with the goal of combining base hierarchical clusterings, Rashedi et al. [172] make use of Renyi and Jensen–Shannon Divergences as the measures to shape the aggregation of data matrices, each representing an input hierarchy.

4.2. Applications of cluster ensembles

In addition to the extensions elaborated earlier, this section looks into different applications of cluster ensemble, with respect to two viewpoints. The first part explores the applications to specific problem domains such as time series analysis. The second emphasizes the use of cluster ensemble for other data-mining tasks, e.g., transfer learning and classification.

4.2.1. Specific problem domains

Since 2010, various applications of cluster ensembles have been implemented and deployed in different domains, including a number of interesting areas such as transportation and cybersecurity. These can be categorized as follows.

- **Transportation**: Since the security issues of High Speed Train (HST) have recently been the centre of attention, a cluster ensemble method called CECH algorithm [162] is used for the diagnosis of running gear faults. The study is based on vibration data collected by sensors that reflect the operation condition. In addition, Fiori et al. [173] also apply the concept of consensus clustering to disclose the transportation network knowledge. The method called DeCoClu (Density Consensus Clustering) is introduced to analyse GPS data to infer geographical locations of stops and other information captured by the vehicles during their work.
- **Time series analysis**: The analysis of time series becomes another important area of research with numerous applications. Particularly to manufacturing, ability to recognize slide alterations is needed as indicative of a malfunction. It is known that manual monitoring can be tedious, yet require experts' undistracted attention. Hence, an automated alternative called control chart pattern recognition (CCPR) model has been proposed with the use of consensus clustering [174]. In addition, Ramasso et al. [175] introduce a cluster ensemble approach for unsupervised pattern recognition in acoustic emission (AE) time-series issued from composite materials. It is able to emphasize sudden growths of damages in composites under solicitations. Furthermore, a HMM-based partitioning ensemble is proposed for temporal data clustering [176]. The resulting model provides several benefits such as: (i) the model initialization problem can be solved through the ensemble setting; (ii) the appropriate number of clusters is automatically determined; and (iii) no parameter re-estimation is required for a pair of clusters to be merged, which helps the HMM agglomerative clustering to be much more efficient.
- **Image processing and computer vision**: The task of image segmentation is the initial and one of the most critical stage in image analysis. There exist various segmentation techniques each of which naturally requires an optimal setting of parameters. In practice, this is achieved by an application of supervised parameter learning to derive the desired setting. On the other hand, a new research direction leads to the combination of different segmentations into a final consensus solution. To reach the goal, the methodology of cluster ensemble is exploited to aggregate the results of different segmentation algorithms and parameter settings [177]. Similar study by Kim et al. [178] has investigated the use of hierarchical segmentation ensemble clustering for the partitioning of images into foreground and background regions. In addition to the aforementioned attempts, Wang et al. [179] also introduce a cluster ensemble-based image segmentation algorithm, which overcomes several problems of traditional methods. In particular, the ensemble framework is exploited to fuse the segmentation results from different types of visual features. As a result, it can deliver a better final result and achieve a much more stable performance for broad image categories.

  Along this line of research, Akbarizadeh and Rahmani [180] report the study that integrates spectral clustering and Gabor feature clustering, which can lead to improved segmentation results. Specific to the task of segmenting a satellite image, a hybridization of fuzzy-based cluster ensemble and a supervised learning technique like support vector machine (SVM) is developed to improve the accuracy [181]. Firstly, multiple partitions are generated using a fuzzy clustering technique. These solutions are then separately improved by a classifier-directed process, and finally combined to form the final data partition. Another interesting application of cluster ensemble to shape decomposition is reported by Lewin et al. [182]. Moreover, some researchers develop and utilize visual words (i.e., vector-quantized local descriptors) for category-level object and activity recognition. These vocabularies are frequently built by using a local feature such as SIFT and a single clustering algorithm. It is possible to lift the quality of visual recognition by aggregating heterogeneous codebooks via consensus clustering [183]. This idea has been investigated with the problems of identifying objects and scenes in very challenging datasets.
- **Biometrics**: In the research of Lourenco et al. [184], Electrocardiography (ECG) that has typically been employed for patient monitoring, is investigated as a biometrics trait. The EAC framework of consensus clustering adopted for the analysis of ECG signals in the context of ECG-based biometrics.
- **Voice processing**: With a rapid increase in the volume of recorded speech, e.g., television and audio broadcasting, meeting recordings and voice mails; a growing need for automatically processing of such repository has arisen. However, attempts to content organization, navigation, browsing, and search have been constrained by the data size. One approach to tackle this is speaker segmentation and speaker clustering, where cluster ensembles have proven effective [185].
- **Chemoinformatics**: Research related to chemoinformatics aims to obtain chemical knowledge through representation and organization of chemical data. It is commonly employed for drug discovery and design, especially the process of High-Throughput Screening (HTS) that screens available compounds for useful information. A consensus clustering method is exploited to reduce cost and time for this screening [186]. As such, it leads to the selection of a representative

subset of all the compounds, with the chance of producing redundant information being minimized. Following the previous attempt, Saeed et al. [187] has introduced the information theory and voting based algorithm (Adaptive Cumulative Voting-based Aggregation Algorithm A-CVAA) for the analysis of chemical structures. This is assessed MDL Drug Data Report (MDDR) and Maximum Unbiased Validation (MUV) datasets, based on the ability to separate active from inactive molecules in each cluster [188].

- **Ontology**: An automatic ontology alignment tool performs the matching between concepts belonging to two ontologies. In that, it provides a similarity measure for each pair of the aligned concepts. Despite the development for this issue, none of the existing alternatives is absolutely accurate, with different tools generating distinct similarity values for a specific alignment. Instead of throwing away the results of some methods that seem less appropriate, Chowdhury and Dou [189] propose an ensemble moel of ontology alignment that aggregates multiple alignment outcomes.
- **Text mining**: Document clustering is used in the context of text mining to set groups of similar documents. A specific model called Gravitational Ensemble Clustering (GEC) is introduced for this task [190]. With a similar objective, Costa and Ortale [191] also exploits a cluster ensemble method for the partitioning of XML corpus. This allows the inherently difficult problem of catching structural and content relationships among XML documents into a number of simpler subproblems, whose results will be combined to form the final solution. Another application of consensus clustering to the field of text mining is to improving the quality of subtopic retrieval [192].
- **Emotion recognition**: A method that is capable of automatically detecting a person's emotion state is in great demand for human–machine interaction and other fields like psychology and psychiatry. For this purpose Aidos et al. [193] put forward a voting-based approach of cluster ensemble to analyse a dataset containing EEG signals from subjects who performed a stress-inducing task. In particular, the study focuses on six different feature spaces obtained from band power features and phase-locking factors.
- **Remote sensing**: For an attempt to develop a weather-wise classification system, Mahrooghy et al. [194] introduces High resolution Satellite Precipitation Estimation (SPE), which is based on the Precipitation Estimation from Remotely Sensed Imagery using an Artificial Neural Network Cloud Classification (PERSIANN-CCS) framework. This model consists of four steps: (i) segmentation of infrared cloud images into patches; (ii) extracting features from cloud patches; (iii) clustering cloud patches using the consensus clustering method of LCE; and (4) deriving interpretation through dynamic application of brightness temperature and rain-rate relationships, respectively. Besides this work, another investigation has recently proposed a sampling based approximate spectral clustering ensemble (SASCE) for unsupervised land cover identification using large remote sensing images [195]. To be efficient with large datasets, a simple voting approach is implemented for the generation of final clustering. For agricultural and environmental monitoring, cluster analysis of high spatial resolution remote-sensing images exhibits a crucial role in land-cover identification. To this end, Tasdemir et al. [196] has developed an approximate spectral-clustering ensemble (ASCE2) to fuse data partitions acquired by image clustering with different similarity representations.
- **Geospatial data analysis**: Despite the fact that geospatial clustering emerges as one of the important topics in spatial analysis, existing techniques still analyse only at data level without taking into account domain knowledge as well as users' goals. Regarding the limitation, Gu et al. [197] has invented an ontology-based geospatial cluster ensemble method to generate good clustering results.
- **Bioinformatics**: For successful diagnosis and treatment of cancer, discovering cancer types accurately becomes essential. The difficulty arises as gene expression profiles normally possess a large number of genes, with many are noisy. In order to overcome this, two new consensus clustering frameworks, named as triple spectral clustering-based consensus clustering (SC3) and double spectral clustering-based consensus clustering (SC2Ncut), are proposed [198]. Apart from the analysis of microarray data, the task of detecting protein complexes from protein–protein interaction (PPI) networks is challenging in the field of bioinformatics. In spite of a vast number of computational methods developed for this course, almost all concentrate on a single aspect of the PPI network, hence the limited collection of features for cluster analysis. To overcome such as deficit, a Bayesian Nonnegative Matrix Factorization (NMF)-based cluster ensemble method is used to aggregate clustering results, which are derived from features of different PPI aspects [199]. Another work by Wang et al. [200] has demonstrated the use of consensus clustering for determining the subtype for a breast cancer patient, through the integration of multiple modalities of data. These range from genotypes to multiple levels of phenotypes.

  Besides, Lock and Dunson [155] report the work on an integrative ensemble model that fuses separate clusterings of the objects for each data source. It makes use of a Bayesian framework for simultaneous estimation of both the consensus clustering and the source-specific clusterings. This is evaluated with the task of identifying subtype of breast cancer tumour samples using publicly available data from the Cancer Genome Atlas.

  A semi-supervised consensus clustering algorithm has also been implemented for electrocardiography (ECG) pathology classification [201]. Yang et al. [202] presents a specific use of cluster ensemble in the context of microbial community responses to human habitats. This is a significant task, as establishing baselines of human microbiome is essential in understanding its role in human disease and health. The study investigates a microbial similarity network that integrates 1920 metagenomic samples from three body habitats.
- **Environment and natural resources**: Provided the global concern of water scarcity, a large number of hydrology researchers have worked on forecasting of water quantity and quality, as well as regionalization of river basins. As such, the need to enhance the quality of prediction of yield in river basins arises. In response, Ahuja [203] publishes research findings based on the data of Godavari basin, which is regionalized using a cluster ensemble method. The method of consensus clustering is also implemented for characterizing flow patterns in soils [204]. It is known that the quality of both surface water and groundwater is directly subjected to flow paths in the vadose zone. This leads to studies that aim to visualize flow patterns in soils. In general, it requires image classification of stained and non-stained parts and the calculation of the dye coverage, which can be interpreted against depth to determine flow types.

- **Cybersecurity**: Along the advance development of Internet technology and applications, the subject of cybersecurity has gained an enormous impact on both information integrity and privacy. One of the emerging threats encountered around the globe is malware that has been induced by the extensive use of mobile communication. With the aim to detect this harmful program, Ye et al. [205] propose the Automatic Malware Categorization System (AMCS) that automatically groups malware samples into sets that share some common characteristics using a cluster ensemble. The underlying analysis is based on features related to instruction frequency and function-based instruction sequences. Following that, an ensemble clustering system called DUET is introduced for the same task [206], providing a learning platform to combine static instruction features and dynamic behaviour features.

  In fact, determining class boundaries of overlapping malware families is a difficult goal to accomplish. As a response, Hou et al. [207] create an intelligent malware detection system that can resolve this using cluster-oriented ensemble classifiers. It is evaluated with Windows Application Programming Interface (API) calls extracted from the file samples. In comparison with malware incidents, phishing website fraud is a relatively new threat. However, shared properties exist: (i) as driven by economic benefits, both malware samples and phishing websites are created at a rate of thousands per day; and (ii) phishing websites represented by the term frequencies possess similar characteristics with malware samples represented by the instruction frequencies. An example of using cluster ensemble for phishing website detection is given in the work of Zhuang et al. [208], where domain knowledge in the form of website-level constraints can be naturally incorporated into the ensemble framework.
- **Network analysis**: Revealing the modules in complex networks is significant to the understanding of systems. An ensemble clustering method is employed to incorporate node groupings of various sizes, with sequential removal of weak links between nodes that are rarely grouped together [209]. It has been successfully applied to several cases, e.g., hierarchical random networks and the American college football network, each with known modular structures. Moreover, Lancichinetti and Fortunato [210] make use of consensus clustering to study the community structure of complex networks. This can help to reveal organization of the discovered communities and hidden relationships among their constituents.
- **Business process management**: With respect to the study of Zhao et al. [211], resource allocation has been regarded as a multi-criteria decision problem, which can be solved by a clustering ensemble approach. This is obtained through the analysis of resource characteristics and task preference patterns from the previous process executions. As such, the right resources may well be recommended, thus improving resource utility.
- **Cloud computing**: As size and complexity of cloud infrastructure increase, scalability has become troublesome for process monitoring and management. This is the case as all virtual machines (VMs) are separately treated, thus producing huge amounts of data to handle. The problem can be tackled by leveraging the similarity between VMs with respect to resource usage patterns [212]. For that, a cluster ensemble framework is created to group similar VMs, without knowledge of the software active in these sessions.
- **Smart living**: A methodology of cluster ensemble has also been investigated for activity monitoring systems, which incorporate sensor-based technology within the smart living scheme [213]. In particular, activities are designed as groups or clusters built on different subsets of extracted features. To classify a new incident, it is assigned to the cluster with the smallest proximity measure from the one under examination.

4.2.2. *Application to other data mining tasks*

Despite the fact that cluster ensemble has been established for unsupervised learning, the method has recently been exploited for other tasks related to data analysis. These include the followings.

- **Transfer learning**: Conventional supervised learning usually assumes that both training and test data are from a common distribution. Thus a challenge arises in transfer learning, where training and test distributions may be mismatched. The problem is even worse when the test data is actually from a different domain and without labels. In order to resolve this, Acharya et al. [214] introduce an optimization framework, which takes as input one or more classifiers learned on the source domain as well as the results of a cluster ensemble operating solely on the target domain, and yields a consensus labelling of the data in the target domain.
- **Detecting ambiguity in data**: Another novel application scheme of cluster ensemble is to identify uncertain or ambiguous regions in the data under examination [215]. Following the detection, two approaches have been suggested for the treatment of such uncertainty. Firstly, the simplest way is to ignore ambiguous patterns prior to the consensus clustering, thus preserving the non-ambiguous data as good prototypes for any further modelling. The other alternative is to use the ensemble solution obtained by the first to train a supervised model that is later applied to reallocate the ambiguous clusters.
- **Dimensionality reduction**: With large amounts of data being generated in various domains such as bioinformatics and social networks, dimensionality reduction remains a challenging task for data-mining researchers. The concept of cluster ensemble is recently exploited for this problem with the use of genetic algorithm [216]. Based on the validation with conventional classification methods and benchmark data collections, its performance is promising with the accuracy on par with the latest approaches proposed in the literature.
- **Semi-supervised learning**: For the analysis of gene expression data, Wang and Pan [217] introduce semi-supervised consensus clustering (SSCC) that integrate the LCE model [20] with semi-supervised clustering process. The clustering quality can be improved when prior knowledge (in terms of must-link and cannot-link constraints) is provided in addition to a typical proximity metric. This study follows the line of research initially brought about by Yu et al. [218] and Yang et al. [219]. As for the former, a new cluster ensemble method named knowledge based cluster ensemble (KCE) is proposed where prior knowledge of data is included into the cluster ensemble framework. Specific to this, pairwise constraints among data points are encoded as confidence factors between base clusterings. Later, these will be concluded in the form of consensus matrix from which the final result is generated. In the latter, an improved Cop-Kmeans (ICop-Kmeans) algorithm has been put forward to tackle the violation of pairwise constraints usually encountered with the original Cop-Kmeans model. Likewise, Zhang et al. [220] and Yu et al. [221] contribute

to this subject by proposing the semi-supervised clustering ensemble model based on collaborative training (SCET) and the incremental semi-supervised clustering ensemble framework (ISSCE), respectively.

- **Data classification**: In spite of the difference between unsupervised and supervised learning, the use of cluster analysis in classification tasks has shown to be effective to raise the classification accuracy [222]. This is pretty much with the fact that data clusters can provide supplementary constraints that may yield the generalization capability of a classifier. In the work of Nguyen et al. [223], the use of clustering information in addition to the original data attributes has been reported to improve the accuracy of intrusion detection problem. Two other works of Sang-Woon [224] and Nasierding et al. [225] have combined cluster labels and conventional supervised algorithms for face recognition and image annotation, respectively.

  Following these, Iam-On and Boongoen [226] present an investigation of employing the information of cluster ensemble for classification modelling. In particular, the ensemble-information matrix created by link-based ensemble clustering or LCE [20] is evaluated as the transformed data for classifier development. In that, the refined sample-cluster association matrix can be considered as the representation of samples in the transformed space, which is discovered from multiple clusterings in the setting of original features. Having accomplished this, the initial data dimensions are reduced to a set of cluster labels, with which each sample associates to a certain degree. Given the common conclusion that a combination of multiple classifiers is able to increase classification accuracy, a new classifier combination scheme is proposed based on the Decision Templates Combiner [222]. It represents the classifiers decision as a vector in an intermediate feature space, then creates decision templates using cluster ensembles.

5. Challenges and conclusion

This survey has presented classical and recently developed approaches to cluster ensemble. It kicks off with the formal terminology by which the problem is defined. Four basic categories of consensus clustering methods are then discussed in depth with illustrative examples. After that, it provides details of extensions to three main components of a cluster ensemble framework: ensemble generation, representation and summarization, and consensus function, respectively. Given the superior capability to deliver accurate data partitions, many cluster ensemble techniques have been exploited for a wide range of applications and domain problems. In addition, the use of this meta-learning approach for other data-mining tasks such as classification has been studied. The attention received by this subject has consistently increased over the years, especially after 2010 that is the focus of this survey. Based on the statistics shown in Fig. 4, the numbers of Google scholar search results for "cluster ensemble" or "consensus clustering" are 1240, 1660 and 2800 for the periods of 2011–12, 2013–14 and 2015–16, respectively. It is clearly illustrated that these counts are much higher than those belonging to the intervals before 2010.

The aforementioned observation is greatly due to the maturity of basic practice to cluster ensemble and a flourish of its applications. From the early period with most of the works relating only to bioinformatics, especially microarray data analysis, the application landscape has largely expanded over the past few years. It covers several new challenges to the modern age such as cybersecurity and time-series data analysis. The followings summarize potential challenges of cluster ensemble in the near future.

- **Heterogeneous data analysis**: Despite the long history of development, most of cluster ensemble methods have been directed to numerical data analysis. Only a handful of publications report findings with other types of data. Specific to the work of Iam-On et al. [227], the link-based method or LCE is adopted for the clustering of biological samples. Each of these can be expressed by both continuous variables extracted from microarray data, and nominal variables obtained from clinical or pathological data of the samples under examination. This so-called integrative approach to biological data analysis has shown to improve the accuracy of prognostic outcome, as compared to those obtained by using one of the aforementioned factors alone. However, given the fact that the aforementioned model is based simply on k-prototype algorithm, its performance is highly subjected to parameter setting (i.e., weights given to continuous and nominal variables).

  A gap of improvement in terms of clustering quality and model robustness exists especially for implementing new inventions of mixed-type data clustering in the ensemble context. For instance, Blomstedt et al. [228] recently introduce a model-based algorithm for clustering attributes of mixed type, which is based a Bayesian predictive framework. Provided that clustering solutions represent random data partitions, the posterior probability for a partition can be determined using conjugate analysis. Another approach applies unsupervised feature learning (UFL) to mixed-type data in order to acquire a sparse representation. As a result, it becomes easier for clustering algorithms to disclose data partitions [229]. While conventional UFL techniques are designed for homogeneous data, the aforementioned works with the mixed-type data using fuzzy adaptive resonance theory (ART). In the biomedical domain, Abidin and Westhead [230] also point out the need for accurate cluster analysis of mixed type data. This commonly appears as a mixture of binary or nominal data (e.g. presence of mutations, binding and epigenetic marks) and continuous data (e.g. gene expression and metabolite levels). As such, a generic clustering method is proposed and evaluated with genetic regulation and the clustering of cancer samples.

- **Big data analysis**: Common applications on office and social based platforms have facilitated the vast amount of data being generated daily. Analysing this so-called big data has been a major trend and challenge within the community of data mining. To better appreciate this, see the comparative study of Fahad et al. [231], where several classical clustering techniques are assessed against big datasets. In particular to an ensemble model, it may face the problem of scaling up, despite the quality it produces. In response, Su et al. [232] introduce a novel cluster ensemble approach for fuzzy clustering, especially for big data. It first builds fuzzy base clusters with respect to each data feature. Then, it makes use of a fuzzy hierarchical graph to represent relationships between those base clusters. Based on this representation scheme, the final result is generated using hierarchical clustering as the consensus function. This work follows an initial attempt to mitigate the practice of cluster ensemble to large data [233]. In that, ECCA (Ensemble of Combined Clustering Algorithms) is invented as a framework of ensemble of algorithms with fixed uniform grids. The final collective solution is based on pairwise classification of the elements of the grid structure. Another study attempts to deal with the curse of dimensionality in big data, especially for cluster ensemble [234]. In particular, a new fuzzy c-means (FCM) algorithm with random projection has been created as the basis of novel consensus clustering, which scales linearly

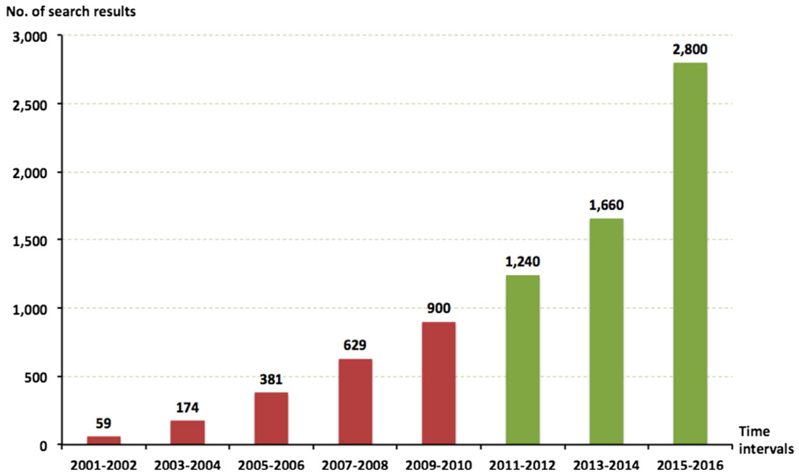


Fig. 4. A comparison of Google scholar search results of "cluster ensemble" or "consensus clustering", over different time intervals from 2001 to 2016.

with data size. This is achieved through calculating spectral embedding of data with cluster-centre based representation.

Analysing the big data has become a major challenge, especially to those web-based organizations such as Google and Facebook. They commonly develop a customized variation of non-relational systems not only to overcome the limitations of efficient storage and retrieval, but also pave the way for data analytics [235]. Some of the new approaches to analysing big data gain a great deal of attention amongst commercial and academic researchers, e.g., Google's MapReduce framework, Hadoop and Hive. According to the report of Dean and Ghemawat [236], MapReduce has been the most popular solution for parallel and analysis of large amount of data. Within the community of data mining, implementations of several techniques using MapReduce have been presented in the past few years. For instance, Liu et al. [237] introduce a MapReduce based parallel back-propagation neural network (MR-BPNN). As for data clustering, a MapReduce-based artificial bee colony (MR-ABC) is developed for a clustering method similar to k-means [238]. This ABC implementation helps to optimize the assignment of the large data objects to clusters. However, for cluster ensemble, such an implementation has rarely been reported in the literature. In fact, one recent publication kicks off this research direction, with the introduction of a new parallel k-means clustering based on MapReduce framework for aspect based summary generation [239]. Of course, an opportunity to coupling existing cluster ensemble methods with MapReduce or other big-data platforms is obvious. This may further boost its application that is in line with the new challenges encountered by big data scientists.

- **Repository of tools**: Ever since its introduction in the early 2000s, the scope of end users of cluster ensemble or consensus clustering is rather limited. As compared to conventional clustering algorithms like k-means or DBSCAN that are available in several well-known data mining tools (e.g., Weka[2] and RapidMiner[3]), implementations of those ensemble models appear to be harder to obtain. Most of them are provided as a supplementary to the publication, which can disappear over time. Yet, this is typically not user friendly as it has been customized in a specific programming environment. As a result, it is also significant to make this family of methods known to a wider public, perhaps as an extension to the well-established tools. This may help broaden the application domain to cover interesting problems in the new era of data intensive industry and society.

Acknowledgements

This work is funded by ST/P005594/1 - Newton STFC-NARIT: Using astronomy surveys to train Thai researchers in Big Data analysis.

References

[1] D. Jiang, C. Tang, A. Zhang, Cluster analysis for gene expression data: A survey, IEEE Trans. Knowl. Data Eng. 16 (2004) 1370–1386.

[2] R.C. Wu, R.S. Chen, C.C. Chang, J.Y. Chen, Data mining application in customer relationship management of credit card business, in: Proceedings of international conference on Computer software and applications, 2005, pp. 39–40.

[3] S.K. Bhatia, J.S. Deogun, Conceptual clustering in information retrieval, IEEE Trans. Syst. Man Cybern. 28 (1998) 427–436.

[4] J. Zhang, J. Mostafa, H. Tripathy, Information retrieval by semantic analysis and visualisation of the concept space of D-Lib magazine, D-Lib Mag. 8 (2002).

[5] J.A.F. Costa, M. de Andrade Netto, Cluster analysis using self-organising maps and image processing techniques, Proc. IEEE Int. Conf. Syst. Man Cybern. 5 (1999) 367–372.

[6] H. Tao, T.S. Huang, Color image edge detection using cluster analysis, in: Proceedings of IEEE International Conference on Image Processing, 1997, pp. 834–836.

[7] G.S. Day, R.M. Heeler, Using cluster analysis to improve marketing experiments, J. Market. Res. 8 (1971) 340–347.

[8] A.G. Sheppard, The sequence of factor analysis and cluster analysis: Differences in segmentation and dimensionality through the use of raw and factor scores, Tourism Anal. 1 (1996) 49–57.

[9] D.B. Henry, P.H. Tolan, D. Gorman-Smith, Cluster analysis in family psychology research, J. Family Psychol. 19 (2005) 121–132.

[10] K. Kim, H. Ahn, A recommender system using GA K-means clustering in an online shopping market, Expert Syst. Appl. 34 (2008) 1200–1209.

[11] M. Bredel, C. Bredel, D. Juric, G. Harsh, H. Vogel, L. Recht, B. Sikic, Functional network analysis reveals extended gliomagenesis pathway maps and three novel MYC-interacting genes in human gliomas, Cancer Res. 65 (2005) 8679–8689.

[12] E. Kim, S. Kim, D. Ashlock, D. Nam, MULTI-K: Accurate classification of microarray subtypes using ensemble k-means clustering, BMC Bioinform. 10 (2009) 260.

[13] T. Sorlie, R. Tibshirani, J. Parker, T. Hastie, J. Marron, A. Nobel, S. Deng, H. Johnsen, R. Pesich, S. Geisler, Repeated observation of breast tumor subtypes in independent gene expression data sets, Proc. Natl. Acad. Sci. USA 100 (2003) 8418–8423.

[14] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: A review, ACM Comput. Survey 31 (1999) 264–323.

[15] A. Ahmad, L. Dey, A k-mean clustering algorithm for mixed numeric and categorical data, Data Knowl. Eng. 63 (2007) 503–527.

[2] http://www.cs.waikato.ac.nz/ml/weka/.

[3] https://rapidminer.com/.

[16] Z. Huang, Claustering large data sets with mixed numeric and categorical values, in: Proceedings of the First Pacific Asia Knowledge Discovery and Data Mining Conference, 1997, pp. 21–34.

[17] S. Dudoit, J. Fridyand, A prediction-based resampling method for estimating the number of clusters in a dataset, Genome Biol. 3 (2002) RESEARCH0036.

[18] T. Boongoen, Q. Shen, Nearest-neighbour guided evaluation of data reliability and its applications, IEEE Trans. Syst. Man Cybern. B 40 (2010) 1622–1633.

[19] W.M. Rand, Objective criteria for the evaluation of clustering methods, J. Amer. Statist. Assoc. 66 (1971) 846–850.

[20] N. Iam-On, T. Boongoen, S. Garrett, LCE: A link-based cluster ensemble method for improved gene expression data analysis, Bioinformatics 26 (2010) 1513–1519.

[21] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, second ed., Wiley-Interscience, 2000.

[22] A.L.N. Fred, A.K. Jain, Combining multiple clusterings using evidence accumulation, IEEE Trans. Pattern Anal. Mach. Intell. 27 (2005) 835–850.

[23] H. Xue, S. Chen, Q. Yang, Discriminatively regularized least-squares classification, Pattern Recognit. 42 (2009) 93–104.

[24] A.P. Topchy, A.K. Jain, W.F. Punch, Clustering ensembles: Models of consensus and weak partitions, IEEE Trans. Pattern Anal. Mach. Intell. 27 (2005) 1866–1881.

[25] A.K. Jain, R. Duin, J. Mao, Statistical pattern recognition: A review, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 4–37.

[26] J. McQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281–297.

[27] Z. Huang, A bfast clustering algorithm to cluster very large categorical data sets in data mining, in: Proceedings of the SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, 1997, pp. 1–8.

[28] P.N. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining, first ed., Addison Wesley, 2005.

[29] M. Ester, H.P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of International Conference on Knowledge Discovery and Data Mining, 1996, pp. 226–231.

[30] T. Kohonen, Self-organized formation of topologically correct feature maps, Biol. Cybernet. 43 (1982) 59–69.

[31] A. Flexer, On the use of self-organizing maps for clustering and visualization, Intell. Data Anal. 5 (2001) 373–384.

[32] J. Vesanto, E. Alhoniemi, Clustering of the self-origanising map, IEEE Trans. Neural Netw. 11 (2000) 586–600.

[33] T. Furukawa, Som of soms, Neural Netw. 22 (2009) 463–478.

[34] K. Tokunaga, T. Furukawa, Modular network som, Neural Netw. 22 (2009) 82–90.

[35] D.H. Wolpert, W.G. Macready, No free lunch theorems for search, in: Technical Report SFI-TR-95-02-010, Santa Fe Institute, 1995.

[36] L.I. Kuncheva, S.T. Hadjitodorov, Using diversity in cluster ensembles, in: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 2004, pp. 1214–1219.

[37] C. Boulis, M. Ostendorf, Combining multiple clustering systems, in: Proceedings of European Conference on Principles and Practice of Knowledge Discovery in Databases, 2004, pp. 63–74.

[38] D. Cristofor, D. Simovici, Finding median partitions using information-theoretical-based genetic algorithms, J. Univ. Comput. Sci. 8 (2002) 153–172.

[39] N. Nguyen, R. Caruana, Consensus clusterings, in: Proceedings of IEEE International Conference on Data Mining, 2007, pp. 607–612.

[40] A.P. Topchy, A.K. Jain, W.F. Punch, A mixture model for clustering ensembles, in: Proceedings of SIAM International Conference on Data Mining, 2004, pp. 379–390.

[41] B. Fischer, J.M. Buhmann, Bagging for path-based clustering, IEEE Trans. Pattern Anal. Mach. Intell. 25 (2003) 1411–1415.

[42] A. Gionis, H. Mannila, P. Tsaparas, Clustering aggregation, ACM Trans. Knowl. Discov. Data 1 (2007) 4–ex.

[43] C. Domeniconi, M. Al-Razgan, Weighted cluster ensembles: Methods and analysis, ACM Trans. Knowl. Discov. Data 2 (2009) 1–40.

[44] X.Z. Fern, C.E. Brodley, Solving cluster ensemble problems by bipartite graph partitioning, in: Proceedings of International Conference on Machine Learning, 2004, pp. 36–43.

[45] A. Strehl, J. Ghosh, Cluster ensembles: A knowledge reuse framework for combining multiple partitions, J. Mach. Learn. Res. 3 (2002) 583–617.

[46] H. Ayad, M. Kamel, Finding natural clusters using multiclusterer combiner based on shared nearest neighbors, in: Proceedings of International Workshop on Multiple Classifier Systems, 2003, pp. 166–175.

[47] X.Z. Fern, C.E. Brodley, Random projection for high dimensional data clustering: A cluster ensemble approach, in: Proceedings of International Conference on Machine Learning, 2003, pp. 186–193.

[48] A.L.N. Fred, Finding consistent clusters in data partitions, in: Multiple Classifier Systems, 2001, pp. 309–318.

[49] A.L.N. Fred, A.K. Jain, Data clustering using evidence accumulation, in: Proceedings of International Conference on Pattern Recognition, pp. 276–280.

[50] A.L.N. Fred, A.K. Jain, Robust data clustering, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2003, pp. 128–136.

[51] S. Monti, P. Tamayo, J.P. Mesirov, T.R. Golub, Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data, Mach. Learn. 52 (2003) 91–118.

[52] J. Kittler, M. Hatef, R. Duin, J. Matas, On combining classifiers, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1998) 226–239.

[53] L.I. Kuncheva, D. Vetrov, Evaluation of stability of k-means cluster ensembles with respect to random initialization, IEEE Trans. Pattern Anal. Mach. Intell. 28 (2006) 1798–1808.

[54] N. Iam-On, T. Boongoen, S. Garrett, Refining pairwise similarity matrix for cluster ensemble problem with cluster relations, in: Proceedings of Eleventh International Conference on Discovery Science, 2008, pp. 222–233.

[55] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, Wiely Publishers, New York, 1990.

[56] S.T. Hadjitodorov, L.I. Kuncheva, L.P. Todorova, Moderate diversity for better cluster ensembles, Inform. Fusion 7 (2006) 264–275.

[57] A.P. Topchy, A.K. Jain, W.F. Punch, Combining multiple weak clusterings, in: Proceedings of IEEE International Conference on Data Mining, 2003, pp. 331–338.

[58] Z. Yu, H.-S. Wong, H. Wang, Graph-based consensus clustering for class discovery from gene expression data, Bioinformatics 23 (2007) 2888–2896.

[59] S. Dudoit, J. Fridyand, Bagging to improve the accuracy of a clustering procedure, Bioinformatics 19 (2003) 1090–1099.

[60] B. Minaei-Bidgoli, A. Topchy, W. Punch, A comparison of resampling methods for clustering ensembles, in: Proceedings of the International Conference on Machine Learning: Models, Technologies and Applications, 2004, pp. 939–945.

[61] E. Bingham, H. Mannila, Random projection in dimensionality reduction: Applications to image and text data, in: Proceedings of International Conference on Knowledge Discovery and Data Mining, 2001, pp. 245–250.

[62] R. Avogadri, G. Valentini, Fuzzy ensemble clustering based on random projections for DNA microarray data analysis, Artif. Intell. Med. 45 (2009) 173–183.

[63] L. Breiman, Bagging predictors, Mach. Learn. 24 (1996) 123–140.

[64] S. Kim, J. Lee, Ensemble clustering method based on the resampling similarity measure for gene expression data, in: Statistical Methods in Medical Research, 2007, pp. 1–26.

[65] X. Hu, I. Yoo, Cluster ensemble and its applications in gene expression analysis, in: Proceedings of Asia-Pacific Bioinformatics Conference, 2004, pp. 297–302.

[66] M. Law, A. Topchy, A.K. Jain, Multiobjective data clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2004 pp. 424–430.

[67] A.L.N. Fred, A.K. Jain, Learning pairwise similarity for data clustering. in: Proceedings of International Conference on Pattern Recognition, 2006, pp. 925–928.

[68] E. Bauer, R. Kohavi, An empirical comparison of voting classification algortihms: Bagging, boosting, and variants, Mach. Learn. 36 (1999) 105–139.

[69] L. Lam, C.Y. Suen, Application of majority voting to pattern recognition: An analysis of its behavior and performance, IEEE Trans. Syst. Man Cybern. 22 (1997) 553–568.

[70] A.P. Topchy, M.H.C. Law, A.K. Jain, A.L. Fred, Analysis of consensus partition in cluster ensemble, in: Proceedings of IEEE International Conference on Data Mining, 2004, pp. 225–232.

[71] J. Munkres, Algorithms for the assignment and transportation problems, J. SIAM 5 (1957) 32–38.

[72] H. Ayad, M. Kamel, Cluster-based cumulative ensembles, in: Proceedings of International Workshop on Multiple Classifier Systems, 2005, pp. 236–245.

[73] E. Dimitriadou, A. Weingessel, K. Hornik, Voting-merging: An ensemble method for clustering, in: Proceedings of International Conference on Artificial Neural Networks, 2001, pp. 217–224.

[74] E. Dimitriadou, A. Weingessel, K. Hornik, A combination scheme for fuzzy clustering, Int. J. Pattern Recognit. Artif. Intell. 16 (2002) 901–912.

[75] D. Frossyniotis, M. Pertselakis, A. Stafylopatis, A multi-clustering fusion algorithm, in: Proceedings of Hellenic Conference on AI, 2002, pp. 225–236.

[76] H. Ayad, M. Kamel, Cumulative voting consensus method for partitions with a variable number of clusters, IEEE Trans. Pattern Anal. Mach. Intell. 30 (2008) 160–173.

[77] H. Ayad, M. Kamel, On voting-based consensus of cluster ensembles, Pattern Recognit. 43 (2010) 1943–1953.

[78] H. Luo, F. Kong, Y. Li, Combining multiple clusterings via k-modes algorithm, in: Proceedings of International Conference on Advanced Data Mining and Applications, 2006, pp. 308–315.

[79] N. Bansal, A. Blum, S. Chawla, Correlation clustering, Mach. Learn. 56 (2004) 89–113.

[80] D.S. Hochbaum, D.B. Shmoys, A best possible heuristic for the k-center problem, Math. Oper. Res. 10 (1985) 180–184.

[81] D.H. Fisher, Knowledge acquisition via incremental conceptual clustering, Mach. Learn. 2 (1987) 139–172.

[82] B. Mirkin, Reinterpreting the category utility function, Mach. Learn. 45 (2001) 219–228.
[83] P.S. Bradley, U.M. Fayyad, Refining initial points for k-means clustering, in: Proceedings of the International Conference on Machine Learning, 1998, pp. 91–99.
[84] S. Swift, A. Tucker, V. Vinciotti, N. Martin, C. Orengo, X. Liu, P. Kellam, Consensus clustering and functional interpretation of gene-expression data, Genome Biol. 5 (2004) R94.
[85] F. Weng, Q. Jiang, L. Chen, Z. Hong, Clustering ensemble based on the fuzzy KNN algorithm, in: Proceedings of International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007, pp. 1001–1006.
[86] Y. Li, J. Yu, P. Hao, Z. Li, Clustering ensembles based on normalized edges, in: Proceedings of Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, 2007, pp. 664–671.
[87] S. Guha, R. Rastogi, K. Shim, ROCK: A robust clustering algorithm for categorical attributes, Inf. Syst. 25 (2000) 345–366.
[88] J. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.
[89] J. Bezdek, R. Hathaway, Recent convergence results for the fuzzy c-means clustering algorithms, J. Classification 5 (1988) 237–247.
[90] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 888–905.
[91] G. Karypis, V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs, SIAM J. Sci. Comput. 20 (1998) 359–392.
[92] G. Karypis, V. Kumar, Multilevel k-way partitioning scheme for irregular graphs, J. Parallel Distrib. Comput. 48 (1998) 96–129.
[93] G. Karypis, V. Kumar, A parallel algorithm for multilevel graph-partitioning and sparse matrix ordering, J. Parallel Distrib. Comput. 48 (1998) 71–95.
[94] G. Karypis, R. Aggarwal, V. Kumar, S. Shekhar, Multilevel hypergraph partitioning: Applications in vlsi domain, IEEE Trans. VLSI Syst. 7 (1999) 69–79.
[95] A. Ng, M. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, Adv. Neural Inf. Process. Syst. 14 (2001) 849–856.
[96] C. Domeniconi, D. Gunopulos, B. Yan, M. Al-Razgan, D. Papadopoulos, Locally adaptive metrics for clustering high-dimensional data, Data Mining Knowl. Discov. 14 (2007) 63–97.
[97] P. Reuther, B. Walter, Survey on test collections and techniques for personal name matching, Int. J. Metadata Semant. Ontol. 1 (2006) 89–99.
[98] G. Jeh, J. Widom, SimRank: A measure of structural-context similarity, in: Proceedings of International Conference on Knowledge Discovery and Data Mining, 2002, pp. 538–543.
[99] X.Z. Fern, W. Lin, Cluster ensemble selection, Statist. Anal. Data Mining 1 (2008) 128–141.
[100] H. Alizadeh, H. Parvin, S. Parvin, A framework for cluster ensemble based on a max metric as cluster evaluator, IAENG Int. J. Comput. Sci. 39 (2012) 10–19.
[101] S. Zhang, H.S. Wong, Y. Shen, Generalized adjusted rand indices for cluster ensembles, Pattern Recognit. 45 (2012) 2214–2226.
[102] H. Parvin, B. Minaei-Bidgoli, A clustering ensemble framework based on elite selection of weighted clusters, Adv. Data Anal. Classif. 7 (2013) 181–208.
[103] H. Alizadeh, B. Minaei-Bidgoli, H. Parvin, Cluster ensemble selection based on a new cluster stability measure, Intell. Data Anal. 18 (2014) 389–408.
[104] H. Alizadeh, B. Minaei-Bidgoli, H. Parvin, To improve the quality of cluster ensembles by selecting a subset of base clusters, J. Experiment. Theoret. Artif. Intell. 26 (2014) 127–150.
[105] M. Naldi, A. Carvalho, R. Campello, Cluster ensemble selection based on relative validity indexes, Data Mining Knowl. Discov. 27 (2013) 259–289.
[106] P. Rastin, R. Kanawati, A multiplex-network based approach for clustering ensemble selection, in: Proceedings of IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 2015, pp. 1332–1339.
[107] H. Alizadeh, M. Yousefnezhad, B.M. Bidgoli, Wisdom of crowds cluster ensemble, Intell. Data Anal. 19 (2015) 485–503.
[108] J. Jia, X. Xiao, B. Liu, Similarity-based spectral clustering ensemble selection, in: Proceedings of International Conference on Fuzzy Systems and Knowledge Discovery, 2012, pp. 1071–1074.
[109] E. Akbari, H.M. Dahlana, R. Ibrahima, H. Alizadeh, Hierarchical cluster ensemble selection, Eng. Appl. Artif. Intell. 39 (2015) 146–156.
[110] H. Parvin, B. Minaei-Bidgoli, A clustering ensemble framework based on selection of fuzzy weighted clusters in a locally adaptive clustering algorithm, Pattern Anal. Appl. 18 (2015) 87–112.
[111] Z. Yu, H. Chen, J. You, J. Liu, H.S. Wong, G. Han, L. Li, Adaptive fuzzy consensus clustering framework for clustering analysis of cancer data, IEEE/ACM Trans. Comput. Biol. Bioinform. 12 (2015) 887–901.
[112] P. Lingras, F. Haider, Partially ordered rough ensemble clustering for multi granular representations, Intell. Data Anal. 19 (2015) 103–116.
[113] J. Hu, T. Li, H. Wang, H. Fujita, Hierarchical cluster ensemble model based on knowledge granulation, Knowl.-Based Syst. 91 (2016) 179–188.
[114] J. Wu, H. Liu, H. Xiong, J. Cao, J. Chen, K-means-based consensus clustering: A unified view, IEEE Trans. Knowl. Data Eng. 27 (2014) 155–169.
[115] H. Zhang, Z. Yang, E. Oja, Improving cluster analysis by co-initializations, Pattern Recognit. Lett. 45 (2014) 71–77.
[116] J. Jia, X. Xiao, B. Liu, L. Jiao, Bagging-based spectral clustering ensemble selection, Pattern Recognit. Lett. 32 (2011) 1456–1467.
[117] H. Parvin, B. Minaei-Bidgoli, H. Alinejad-Rokny, W.F. Punch, Data weighing mechanisms for clustering ensembles, Comput. Electr. Eng. 39 (2013) 1433–1450.
[118] B. Minaei-Bidgoli, H. Parvin, H. Alinejad-Rokny, H. Alizadeh, W.F. Punch, Effects of resampling method and adaptation on clustering ensemble efficacy, Artif. Intell. Rev. 41 (2014) 27–48.
[119] Y.Z. Ren, C. Demeniconi, G. Zhang, G.X. Yu, Weigted object ensemble clustering: Methods and analysis, Knowl. Inform. Syst. 51 (2017) 661–689.
[120] F. Yang, X. Li, Q. Li, T. Li, Exploring the diversity in cluster ensemble generation: Random sampling and random projection, Expert Syst. Appl. 41 (2014) 4844–4866.
[121] Y. Yang, J. Jiang, Hybrid sampling-based clustering ensemble with global and local constitutions, IEEE Trans. Neural Netw. Learn. Syst. 27 (2015) 952–965.
[122] Z. Yu, L. Li, Y. Gao, J. You, J. Liu, H.S. Wong, G. Han, Hybrid clustering solution selection strategy, Pattern Recognit. 47 (2014) 3362–3375.
[123] Z. Yu, H.S. Wong, J. You, G. Yu, G. Han, Hybrid cluster ensemble framework based on the random combination of data transformation operators, Pattern Recognit. 45 (2012) 1826–1837.
[124] Z. Yu, L. Li, J. Liu, J. Zhang, G. Han, Adaptive noise immune cluster ensemble using affinity propagation, IEEE Trans. Knowl. Data Eng. 27 (2015) 3176–3189.
[125] X. Wang, D. Han, C. Han, Rough set based cluster ensemble selection, in: Proceedings of International Conference on Information Fusion, 2013, pp. 438–444.
[126] S.J. Fodeh, C. Brandt, T.B. Luong, A. Haddad, M. Schultz, T. Murphy, M. Krauthammer, Complementary ensemble clustering of biomedical data, J. Biomed. Inform. 46 (2013) 436–443.
[127] H. Alizadeh, B. Minaei-Bidgoli, H. Parvin, Optimizing fuzzy cluster ensemble in string representation, Int. J. Pattern Recognit. Artif. Intell. 27 (2013) 1–22.
[128] T. Wang, Ca-tree: A hierarchical structure for efficient and scalable coassociation-based cluster ensembles, IEEE Trans. Syst. Man Cybern. B 41 (2011) 1083–4419.
[129] L. Du, X. Li, Y.D. Shen, Cluster ensembles via weighted graph regularized nonnegative matrix factorization, in: Proceedings of International Conference on Advanced Data Mining and Applications, 2011, pp. 215–228.
[130] D. Huang, J. Lai, C.D. Wang, Ensemble clustering using factor graph, Pattern Recognit. 50 (2016) 131–142.
[131] O. Wu, W. Hu, S.J. Maybank, M. Zhu, B. Li, Efficient clustering aggregation based on data fragments, IEEE Trans. Syst. Man Cybern. B 40 (2012) 913–926.
[132] C.H. Chung, B.R. Dai, A fragment-based iterative consensus clustering algorithm with a robust similarity, Knowl. Inform. Syst. 41 (2014) 591–609.
[133] C. Wang, Z. She, L. Cao, Coupled clustering ensemble: Incorporating coupling relationships both between base clusterings and objects, in: Proceedings of IEEE International Conference on Data Engineering, 2013, pp. 374–385.
[134] N. Iam-On, T. Boongoen, S. Garrett, C. Price, A link-based approach to the cluster ensemble problem, IEEE Trans. Pattern Anal. Mach. Intell. 33 (2011) 2396–2409.
[135] D. Huang, J.H. Lai, C.D. Wang, Combining multiple clusterings via crowd agreement estimation and multi-granularity link analysis, Neurocomputing 170 (2015) 240–250.
[136] J.M. Duarte, A. Fred, F. Jorge, F. Duarte, Adaptive evidence accumulation clustering using the confidence of the objects' assignments, in: Proceedings of PAKDD, 2013, pp. 70–87.
[137] A. Lourenco, S.R. Bulo, A. Fred, M. Pelillo, Consensus clustering with robust evidence accumulation, in: Proceedings of International Conference on EMM-CVPR, 2013, pp. 307–320.
[138] Y. Ren, C. Domeniconi, G. Zhang, G. Yu, Weighted-object ensemble clustering, in: Proceedings of IEEE International Conference on Data Mining, 2013, pp. 627–636.
[139] V. Berikov, Weighted ensemble of algorithms for complex data clustering, Pattern Recognit. Lett. 38 (2014) 99–106.
[140] C. Zhong, X. Yue, Z. Zhang, J. Lei, A clustering ensemble: Two-level-refined co-association matrix with path-based transformation, Pattern Recognit. 48 (2015) 2699–2709.
[141] A. Louren, S.R. Bulo, N. Rebagliati, A. Fred, M. Figueiredo, M. Pelillo, Probabilistic consensus clustering using evidence accumulation, Mach. Learn. 98 (2015) 331–357.
[142] H. Liu, T. Liu, J. Wu, D. Tao, Y. Fu, Spectral ensemble clustering, in: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 715–724.
[143] H. Wang, Y. Yang, H. Wang, D. Chen, Soft-voting clustering ensemble, in: Proceedings of International Workshop on Multiple Classifier Systems, 2013, pp. 307–318.
[144] X. Sevillano, F. Alas, J.C. Socoro, Positional and confidence voting-based consensus functions for fuzzy cluster ensembles, Fuzzy Sets and Systems 193 (2012) 1–32.
[145] L. Zhang, X.H. Ling, J.W. Yang, X.Q. Wang, F.Z. Li, Cascaded cluster ensembles, Int. J. Mach. Learn. Cybern. 3 (2012) 335–343.

[146] P. Zhou, L. Du, H. Wang, L. Shi, Y.D. Shen, Learning a robust consensus matrix for clustering ensemble via kullback-leibler divergence minimization, in: Proceedings of International Joint Conference on Artificial Intelligence, 2015, pp. 4112–4118.

[147] H. Li, H. Lin, J. Wu, G. Cheng, Bv-rsa: A rapid simulated annealing model for ensemble clustering, in: Proceedings of International Conference on Service Systems and Service Management, 2015, pp. 1–6.

[148] I.T. Christou, Coordination of cluster ensembles via exact methods, IEEE Trans. Pattern Anal. Mach. Intell. 33 (2011) 279–293.

[149] S. Chatterjee, A. Mukhopadhyay, Clustering ensemble: A multiobjective genetic algorithm based approach, in: Proceedings of International Conference on Computational Intelligence: Modeling Techniques and Applications, 2013, pp. 443–449.

[150] F. Gullo, C. Domeniconi, A. Tagarelli, Metacluster-based projective clustering ensembles, Mach. Learn. 98 (2015) 181–216.

[151] L. Franek, X. Jiang, Ensemble clustering by means of clustering embedding in vector spaces, Pattern Recognit. 47 (2014) 833–842.

[152] V. Bhatnagar, S. Ahuja, S. Kaur, Discriminant analysis-based cluster ensemble, Int. J. Data Mining Modell. Manage. 7 (2015) 83–107.

[153] S. Xu, X. Li, R. Chen, S. Wu, J. Ni, Subspace similarity-based algorithm for combine multiple clustering, in: Proceedings of International Conference on Internet Computing for Engineering and Science, 2013, pp. 69–76.

[154] S. Vega-Pons, P. Avesani, On pruning the search space for clustering ensemble problems, Neurocomputing 150 (2015) 481–489.

[155] E.F. Lock, D.B. Dunson, Bayesian consensus clustering, Bioinformatics 29 (2013) 2610–2616.

[156] Z. Yu, L. Li, H.S. Wong, J. You, G. Han, Y. Gao, G. Yu, Probabilistic cluster structure ensemble, Inform. Sci. 267 (2014) 16–34.

[157] M.H. Masson, T. Denoeux, Ensemble clustering in the belief functions framework, Internat. J. Approx. Reason. 52 (2011) 92–109.

[158] Y. Wu, X. Liu, L. Guo, A new ensemble clustering method based on dempster-shafer evidence theory and gaussian mixture modeling, 2014, pp. 1–8.

[159] L. Zheng, T. Li, C. Ding, A framework for hierarchical ensemble clustering, ACM Trans. Knowl. Discov. Data 9 (2014) 9.

[160] H. Aidos, A. Fred, Consensus of clusterings based on high-order dissimilarities, in: Partitional Clustering Algorithms, 2015, pp. 313–351.

[161] D. Yan, A. Chen, M.I. Jordan, Cluster forests, Comput. Statist. Data Anal. 66 (2013) 178–192.

[162] W. Xiao, Y. Yang, H. Wang, T. Li, H. Xing, Semi-supervised hierarchical clustering ensemble and its application, Neurocomputing 173 (2016) 362–1376.

[163] S. Mimaroglu, E. Erdil, Combining multiple clusterings using similarity graph, Pattern Recognit. 44 (2011) 694–703.

[164] D.D. Abdala, P. Wattuya, X. Jiang, Ensemble clustering via random walker consensus strategy, in: Proceedings of International Conference on Pattern Recognition, 2010, pp. 1433–1436.

[165] A.H. Sadeghian, H. Nezamabadi-pour, Gravitational ensemble clustering, 2014, pp. 1–6.

[166] L. Du, Y.D. Shen, Z. Shen, J. Wang, Z. Xu, A self-supervised framework for clustering ensemble, in: Proceedings of International Conference on Web-Age Information Management, 2013, pp. 253–264.

[167] R. Fa, B. Abu-Jamous, D.J. Roberts, A.K. Nandi, Coce-smart: Consensus clustering based on enhanced splitting-merging awareness tactics. in: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, 2015, pp. 2011–2015.

[168] G. Teng, C. He, J. Xiao, Y. He, B. Zhu, X. Jiang, Cluster ensemble framework based on the group method of data handling, Appl. Soft Comput. 43 (2016) 35–46.

[169] N. Ammour, N. Alajlan, A dynamic weights owa fusion for ensemble clustering, Signal Image Video Process. 9 (2015) 727–734.

[170] E. Rashedi, A. Mirzaei, M. Rahmati, Optimized aggregation function in hierarchical clustering combination, Intell. Data Anal. 20 (2016) 281–291.

[171] S.M. Vahidipour, A. Mirzaei, M. Rahmati, Comparing weighted combination of hierarchical clustering based on cophenetic measure, Intell. Data Anal. 18 (2014) 547–559.

[172] E. Rashedi, A. Mirzaei, M. Rahmati, An information theoretic approach to hierarchical clustering combination, Neurocomputing 148 (2015) 487–497.

[173] A. Fiori, A. Mignone, G. Rospo, Decoclu: Density consensus clustering approach for public transport data, Inform. Sci. 328 (2016) 378–388.

[174] S. Haghtalab, P. Xanthopoulos, K. Madani, A robust unsupervised consensus control chart pattern recognition framework, Expert Syst. Appl. 42 (2015) 6767–6776.

[175] E. Ramasso, V. Placet, M.L. Boubakar, Unsupervised consensus clustering of acoustic emission time-series for robust damage sequence estimation in composites, IEEE Trans. Instrum. Meas. 64 (2015) 3297–3307.

[176] Y. Yanga, J. Jiang, Hmm-based hybrid meta-clustering ensemble for temporal data, Knowl.-Based Syst. 36 (2014) 299–310.

[177] L. Franek, D.D. Abdala, S. Vega-Pons, X. Jiang, Image segmentation fusion using general ensemble clustering methods, in: Asian Conference on Computer Vision, 2010, pp. 373–384.

[178] H. Kim, J.J. Thiagarajan, P. Bremer, Image segmentation using consensus from hierarchical segmentation ensembles, in: Proceedings of IEEE International Conference on Image Processing 2014, pp. 3272 –3276.

[179] X. Wang, J. Du, S. Wu, X. Li, F. Li, Cluster ensemble-based image segmentation, Int. J. Adv. Robot. Syst. 10 (2013) 297.

[180] G. Akbarizadeh, M. Rahmani, A new ensemble clustering method for polsar image segmentation, in: Proceedings of International Conference on Information and Knowledge Technology, 2015, pp. 1–4.

[181] I. Saha, U. Maulik, S. Bandyopadhyay, D. Plewczynski, Svmefc: Svm ensemble fuzzy clustering for satellite image segmentation, IEEE Geosci. Remote Sensing Lett. 9 (2011) 52–55.

[182] S. Lewin, X. Jiang, A. Clausing, A clustering-based ensemble technique for shape decomposition. in: Proceedings of Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition, 2012, pp. 153–161.

[183] R.J. Lopez-Sastre, J. Renes-Olalla, P. Gil-Jimenez, S. Maldonado-Bascon, S. Lafuente-Arroyo, Heterogeneous visual codebook integration via consensus clustering for visual categorization, IEEE Trans. Circuits Syst. Video Technol. 23 (2013) 1358–1368.

[184] A. Lourenco, C. Carreiras, S.R. Bulo, A. Fred, Ecg analysis using consensus clustering, in: Proceedings of European Conference on Signal Processing, 2014, pp. 511–515.

[185] N. Bassiou, V. Moschou, C. Kotropoulos, Speaker diarization exploiting the eigengap criterion and cluster ensembles, IEEE Trans. Audio Speech Language Process. 18 (2010) 2134–2144.

[186] F. Saeed, N. Salim, A. Abdo, Voting-based consensus clustering for combining multiple clusterings of chemical structures, J. Cheminform. 4 (2012) 37.

[187] F. Saeed, N. Salim, A. Abdo, Information theory and voting based consensus clustering for combining multiple clusterings of chemical structures, Mole. Inform. 32 (2013) 591–598.

[188] F. Saeed, A. Ahmed, M. Shamsir, N. Salim, Weighted voting-based consensus clustering for chemical structure databases, J. Comput. Aided Mol. Des. 28 (2014) 675–684.

[189] N.A. Chowdhury, D. Dou, Improving the accuracy of ontology alignment through ensemble fuzzy clustering. in: Proceedings of International Symposium on Distributed Objects, Middleware and Applications, 2011, p. 826–833.

[190] A.H. Sadeghian, H. Nezamabadi-pour, Document clustering using gravitational ensemble clustering, in: Proceedings of International Symposium on Artificial Intelligence and Signal Processing, 2015, pp. 240–245.

[191] G. Costa, R. Ortale, Developments in partitioning xml documents by content and structure based on combining multiple clusterings, in: Proceedings of IEEE International Conference on Tools with Artificial Intelligence, 2013, pp. 477–482.

[192] C. Carpineto, G. Romano, Consensus clustering based on a new probabilistic rand index with application to subtopic retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 34 (2012) 2315–2326.

[193] H. Aidos, C. Carreiras, H. Silva, A. Fred, Evidence accumulation approach applied to eeg analysis, in: Proceedings of International Conference on Pattern Recognition Applications and Methods, 2013, pp. 479–484.

[194] M. Mahrooghy, N.H. Younan, V.G. Anantharaj, J. Aanstoos, S. Yarahmadian, On the use of a cluster ensemble cloud classification technique in satellite precipitation estimation, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 5 (2012) 1356–1363.

[195] Y. Moazzen, B. Yalcin, K. Tasdemir, Sampling based approximate spectral clustering ensemble for unsupervised land cover identification. in: Proceedings of IEEE International Symposium on Geoscience and Remote Sensing, 2015, pp. 2405–2408.

[196] K. Tasdemir, Y. Moazzen, I. Yildirim, An approximate spectral clustering ensemble for high spatial resolution remote-sensing images, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 8 (2015) 1996–2004.

[197] W. Gu, Z. Zhang, B. Wang, X. Wang, Use of ontology and cluster ensembles for geospatial clustering analysis, in: Proceedings of Canadian Conference on Artificial Intelligence, 2014, pp. 119–130.

[198] Z. Yu, L. Li, J. You, H.S. Wong, G. Han, Sc3: Triple spectral clustering-based consensus clustering framework for class discovery from cancer gene expression profiles, IEEE/ACM Trans. Comput. Biol. Bioinform. 9 (2012) 1751–1765.

[199] L.O. Yang, D.Q. Dai, X.F. Zhang, Protein complex detection via weighted ensemble clustering based on bayesian nonnegative matrix factorization, PLOS ONE 8 (2013) e62158.

[200] C. Wang, R. Machiraju, K. Huang, Breast cancer patient stratification using a molecular regularized consensus clustering method, Methods 67 (2014) 304–312.

[201] H. Aidos, A. Lourenco, D. Batista, S.R. Bulo, A. Fred, Semi-supervised consensus clustering for ecg pathology classification, in: European Conference on Machine Learning and Knowledge Discovery in Databases, 2015, pp. 150–164.

[202] P. Yang, X. Su, L.O. Yang, H.N. Chua, X.L. Li, K. Ning, Microbial community pattern detection in human body habitats via ensemble clustering framework, BMC Syst. Biol. 8 (2014) 57.

[203] S. Ahuja, Regionalization of river basins using cluster ensemble, J. Water Resour. Protect. 4 (2012) 560–566.

[204] C. Bognera, B.T. Widemann, H. Lange, Characterising flow patterns in soils by feature extraction and multiple consensus clustering, Ecol. Inform. 15 (2013) 44–52.

[205] Y. Ye, T. Li, Y. Chen, Q. Jiang, Automatic malware categorization using cluster ensemble. in: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2010, pp. 95–104.

[206] X. Hu, K.G. Shin, Duet: integration of dynamic and static analyses for malware clustering with cluster ensembles, in: Proceedings of Annual Conference on Computer Security Applications, 2013, pp. 79–88.

[207] S. Hou, L. Chen, E. Tas, I. Demihovskiy, Y. Ye, Cluster-oriented ensemble classifiers for intelligent malware detection, in: IEEE International Conference on Semantic Computing, 2015, pp. 189–196.

[208] W. Zhuang, Y. Ye, Y. Chen, T. Li, Ensemble clustering for internet security applications, IEEE Trans. Syst. Man Cybern. C 42 (2012) 1784–1796.

[209] E.Y. Kim, D.U. Hwang, T.W. Ko, Multiscale ensemble clustering for finding modules in complex networks, Phys. Rev. E 85 (2012) 026119.

[210] A. Lancichinetti, S. Fortunato, Consensus clustering in complex networks, Sci. Rep. 2 (2012) 336.

[211] W. Zhao, H. Liu, W. Dai, J. Ma, An entropy-based clustering ensemble method to support resource allocation in business process management, Knowl. Inform. Syst. 48 (2016) 305–330.

[212] C. Canali, R. Lancellotti, Exploiting ensemble techniques for automatic virtual machine clustering in cloud systems, Automat. Softw. Eng. 21 (2014) 319–344.

[213] A. Jurek, C. Nugent, Y. Bi, S. Wu, Clustering-based ensemble learning for activity recognition in smart homes, Sensors 14 (2014) 12285–12304.

[214] A. Acharya, E.R. Hruschka, J. Ghosh, S. Acharyya, Transfer learning with cluster ensembles, in: Proceedings of Workshop on Unsupervised and Transfer Learning, 2012, pp. 123–133.

[215] A. Albalate, A. Suchindranath, M.M. Soenmez, D. Suendermann, On ambiguity detection and postprocessing schemes using cluster ensembles, in: Proceedings of International Conference on Agents and Artificial Intelligence, 2010, pp. 623–630.

[216] D. Rani, T. Rani, S. Bhavani, Consensus clustering for dimensionality reduction, in: Proceedings of International Conference on Contemporary Computing, 2014, pp. 148–153.

[217] Y. Wang, Y. Pan, Semi-supervised consensus clustering for gene expression data analysis, BMC BioData Mining 7 (2014) 7.

[218] Z. Yu, H.S. Wong, J. You, Q. Yang, H. Liao, Knowledge based cluster ensemble for cancer discovery from biomolecular data, IEEE Trans. NanoBiosci. 10 (2011) 76–85.

[219] Y. Yang, W. Tan, T. Li, D. Ruan, Consensus clustering based on constrained self-organizing map and improved cop-kmeans ensemble in intelligent decision support systems, Knowl.-Based Syst. 32 (2012) 101–115.

[220] J. Zhang, Y. Yang, H. Wang, A. Mahmood, F. Huang, Semi-supervised clustering ensemble based on collaborative training, in: Proceedings of International Conference on Rough Sets and Knowledge Technology, 2012, pp. 450–455.

[221] Z. Yu, P. Luo, J. You, H.S. Wong, H. Leung, S. Wu, J. Zhang, G. Han, Incremental semi-supervised clustering ensemble for high dimensional data clustering, IEEE Trans. Knowl. Data Eng. 28 (2016) 701–714.

[222] M.A. Duval-Poo, J. Sosa-Garcia, A. Guerra-Gandon, S. Vega-Pons, J. Ruiz-Shulcloper, A new classifier combination scheme using clustering ensemble. in: Proceedings of Iberoamerican Congress on Pattern Recognition, 2012, pp. 154–161.

[223] H.H. Nguyen, N. Harbi, J. Darmont, An efficient fuzzy clustering-based approach for intrusion detection, in: Proceedings of IEEE International Conference on Data Mining, 2011, pp. 607–612.

[224] K. Sang-Woon, A pre-clustering technique for optimizing subclass discriminant analysis, Pattern Recognit. Lett. 31 (2010) 462–468.

[225] G. Nasierding, G. Tsoumakas, A.Z. Kouzani, Clustering based multi-label classification for image annotation and retrieval, in: Proceedings of IEEE International Conference on System, Man and Cybernetics, 2009, pp. 4514–4519.

[226] N. Iam-On, T. Boongoen, Revisiting link-based cluster ensembles for microarray data classification, in: Proceedings of IEEE Systems, Man and Cybernetics, 2013, pp. 4543–4548.

[227] N. Iam-On, T. Boongoen, S. Garrett, C. Price, New cluster ensemble approach to integrative biological data analysis, Int. J. Data Mining Bioinform. 8 (2013) 150–168.

[228] P. Blomstedt, J. Tang, J. Xiong, C. Granlund, J. Corander, A bayesian predictive model for clustering data of mixed discrete and continuous type, IEEE Trans. Pattern Anal. Mach. Intell. 37 (2015) 489–498.

[229] D. Lam, M. Wei, D. Wunsch, Clustering data of mixed categorical and numerical type with unsupervised feature learning, IEEE Access 3 (2015) 1605–1613.

[230] F.N.Z. Abidin, D.R. Westhead, Flexible model-based clustering of mixed binary and continuous data: Application to genetic regulation and cancer, Nucleic Acids Res. gkw1270 (2016) 1–11.

[231] A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A.Y. Zomaya, S. Foufou, A. Bouras, A survey of clustering algorithms for big data: Taxonomy and empirical analysis, IEEE Trans. Emerg. Top. Comput. 2 (2014) 267–279.

[232] P. Su, C. Shang, Q. Shen, A hierarchical fuzzy cluster ensemble approach and its application to big data clustering, J. Intell. Fuzzy Syst. 28 (2015) 2409–2421.

[233] I.A. Pestunova, V.B. Berikovb, E.A. Kulikovaa, S.A. Rylova, Ensemble of clustering algorithms for large datasets, Optoelectron. Instrum. Data Process. 47 (2011) 45–252.

[234] M. Ye, W. Liu, J. Wei, X. Hu, Fuzzy c-means and cluster ensemble with random projection for big data clustering, Math. Probl. Eng. 2016 (2016) 1–13.

[235] J.R. Swedlow, G. Zanetti, C. Best, Channeling the data deluge, Nat. Methods 8 (2011) 463–465.

[236] J. Dean, S. Ghemawat, Mapreduce: A flexible data processing tool, Commun. ACM 53 (2010) 72–77.

[237] Y. Liu, J. Yang, Y. Huang, L. Xu, S. Li, M. Qi, Mapreduce based parallel neural networks in enabling large scale machine learning, Comput. Intell. Neurosci. 2015 (2015) 1–13.

[238] A. Banharnsakun, A mapreduce-based artificial bee colony for large-scale data clustering, Pattern Recognit. Lett. 93 (2017) 78–84.

[239] V. Priya, K. Umamaheswari, Ensemble based parallel k means using map reduce for aspect based summarization, in: Proceedings of the International Conference on Informatics and Analytics, 2016, pp. 1–9.