Advances in Data Analysis and Classification
https://doi.org/10.1007/s11634-024-00588-4

REGULAR ARTICLE

Clustering ensemble extraction: a knowledge reuse framework

Mohaddeseh Sedghi[1] · Ebrahim Akbari[1] · Homayun Motameni[1] · Touraj Banirostam[2]

Received: 26 October 2022 / Accepted: 26 February 2024

Abstract

Clustering ensemble combines several fundamental clusterings with a consensus function to produce the final clustering without gaining access to data features. The quality and diversity of a vast library of base clusterings influence the performance of the consensus function. When a huge library of various clusterings is not available, this function produces results of lower quality than those of the basic clustering. The expansion of diverse clusters in the collection to increase the performance of consensus, especially in cases where there is no access to specific data features or assumptions in the data distribution, has still remained an open problem. The approach proposed in this paper, Clustering Ensemble Extraction, considers the similarity criterion at the cluster level and places the most similar clusters in the same group. Then, it extracts new clusters with the help of the Extracting Clusters Algorithm. Finally, two new consensus functions, namely Cluster-based extracted partitioning algorithm and Meta-cluster extracted algorithm, are defined and then applied to new clusters in order to create a high-quality clustering. The results of the empirical experiments conducted in this study showed that the new consensus function obtained by our proposed method outperformed the methods previously proposed in the literature regarding the clustering quality and efficiency.

✉ Ebrahim Akbari
ebrahimakbari30@yahoo.com

Mohaddeseh Sedghi
sedghi.mo@gmail.com

Homayun Motameni
h.motameni@yahoo.com

Touraj Banirostam
banirostam@iauctb.ac.ir

[1] Department of Computer Engineering, Sari Branch, Islamic Azad University, Sari, Iran

[2] Department of Technical and Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran

Published online: 27 March 2024

Keywords Clustering · Clustering ensemble · Knowledge reuse · Clustering ensemble extraction · Diversity

Mathematics Subject Classification 62H30

1 Introduction

Clustering is one of the most important tasks in data mining because it uncovers useful patterns in unlabeled datasets. In other words, it is the process of clustering data points in such a way that the individuals in the same cluster have the most similarity to one another and the least similarity to those in other clusters. In recent years, this issue has grown increasingly relevant in machine learning, pattern recognition, etc. (Tan et al. 2016). Many clustering methods have been created over the years, each of which uses different distance/similarity measurements and/or objective functions (Zhao et al. 2018). In general, different algorithms make predictions with varying degrees of accuracy (Fozieh Asghari et al. 2017). Choosing the best model is not always the best option because dismissing the findings of less effective models can result in the loss of potentially vital information (Fern and Lin 2008; Alizadeh et al. 2014). One of the most fundamental questions is: How should we choose one amid so many options? (Zhao et al. 2018; Alizadeh et al. 2015). There is no single best clustering algorithm for all cases, according to the Kleinberg theorem (Akbari et al. 2015). Clustering results vary depending on the method and the parameters used. One alternative response to this question is that we do not have to choose at all because we can generate different options by (1) producing a group of base clustering (BC) results, or (2) utilizing a consensus function to create a final clustering. Clustering Ensemble (CE) (Strehl and Ghosh 2002), as a knowledge reuse framework, is based on this theory and has become popular in the clustering community in recent years (Fern and Lin 2008; Topchy et al. 2005). CE is defined as the process of integrating numerous clustering results into final clusters without accessing the features or algorithms; it is a useful method for preserving privacy and reusing information (Strehl and Ghosh 2002). A consensus approach is used to combine clusterings (Akbari et al. 2015). CE can considerably increase the stability, robustness, and quality of a clustering solution in comparison with a single clustering technique (Li et al. 2019). Many clustering ensembles, such as categorical data (He et al. 2005; Iam-On et al. 2012), high dimensional data (Jing et al. 2015), noisy data (Yu et al. 2015), temporal data (Yang and Chen 2011), and feature selection, have been successfully handled using the CE technique (Elghazel and Aussem 2015). Scientists have proved that all the base clustering results do not contribute to the creation of the final clustering (Jia et al. 2011). Some methods can be used to analyze and select a subset of partitions that yield ensemble results similar to, or better than, those that are based on the entire set of partitions, in order to improve clustering quality. "Clustering Ensemble Selection" or "cluster ensemble selection" (CES) is the common name for these procedures (Jia et al. 2011). Previous research has demonstrated that such selection can greatly increase the final partition quality. As a result, there is a need for a thorough assessment of the candidate partitions for combination (Naldi et al. 2013). Diversity and quality play critical roles in the selection of the partitions to be combined. The main purpose of CES is to select a subset of large libraries and a consensus of base clusters for best performance so that the clusters could be of high quality while still being significantly different. In knowledge reuse, if the base clusterings are small and without diversity, CES cannot find a subset of base clusterings with appropriate diversity; therefore, it cannot achieve an acceptable result. For a variety of reasons, e.g., the unavailability of original features, the high quality of the label, the confidentiality of features, and the information distributed, various clusterings for the objects under consideration may already exist in several applications and one prefers to apply them to new clusterings instead of throwing them away. This is knowledge reuse. On the other hand, clusters may be available with low diversity and small size, without the main features. In this case, the question is how new clusters could be created by having clusters with such features so that they could have a good variety and their labels could be near to reality. In this paper, a new approach is proposed, namely Clustering Ensemble Extraction (CEE). Figure 1 depicts the overall flow of our approach. CEE involves four steps: first, basic clustering is provided in binary form. Then, with the use of the Jacquard similarity measure, a set of base clusters is selected from among the existing clusters. Afterward, a method is proposed to extract the clusters (see Fig. 4). By the iterative method, two clusters are selected from among those available and then, with the help of logical relations, the new clusters are extracted. Therefore, the quality of the cluster does not decrease. Then, at the last step, using the consensus function, final clustering is obtained. Table 1 summarizes the contributions of the proposed method in comparison with existing ones.

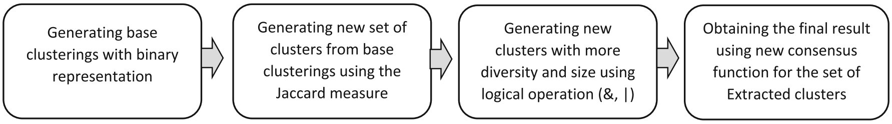

Generating base clusterings with binary representation
Generating new set of clusters from base clusterings using the Jaccard measure
Generating new clusters with more diversity and size using logical operation (&, |)
Obtaining the final result using new consensus function for the set of Extracted clusters

Fig. 1 The proposed approach is depicted in a flow diagram

Table 1 Contributions of the proposed method compared to the methods previously proposed in the literature

The proposed method	Previous methods
(1) It does not have access to the main features of the data, and due to the existing base clusterings, it produces a new set of clusters with high diversity (2) Despite the small library and low diversity, selecting a subset of clusters will not lead to clustering labeled 'close to reality'; to address this problem, we proposed the clusters extraction approach (3) It proposes a new consensus function for the set of extracted clusters instead of clusterings	(1) In previous methods, a large library of clusterings is available, which leads to the production of extremely diverse clusters (2) In most of the methods previously proposed, picking a subset of clusterings from a vast library of clusterings may increase diversity, but it generates the same unreal clusters in the chosen clusterings (3) In previous methods, the consensus function was applied to a collection of basis clusterings

In the experimental section, we explain the importance of the proposed extracted clusters and demonstrate its superiority regarding consensus quality over the well-known clustering group methods found in several datasets. The main contribution of this approach is generating a new set of clusters instead of clusterings in knowledge reuse. In addition, a set of diverse clusters is generated for base clusterings with small size and low diversity. In summary, the contributions of the proposed approach are as follows:

- Generating a new set of clusters from base clusterings using the Jaccard similarity measure.
- Extracting the new set of the generated clusters using the Boolean function with bigger size and higher diversity.
- Proposing two new consensus functions for the set of extracted clusters instead of clusterings.

The outline of the rest of the paper is as follows. Section 2 reviews the related work already existing in the literature. Then, Sect. 3 presents the background knowledge about the presentation of base clusters, diversity and quality measures, and the Jacquard similarity measure. Next, Sect. 4 introduces the proposed algorithm in detail. Afterward, Sect. 5 empirically demonstrates the performance of the proposed algorithm. Finally, Sect. 6 concludes the paper.

2 Related work

CE is a widely used method for improving the quality and stability of clustering results in the clustering research area. There may be high quality and diversity in different clusterings. On the other hand, in a certain clustering, a number of low-quality clusters may be included, which cannot be eliminated. Our main goal is to generate a diverse set of clusters with a significant level of diversity. The diversity stage (creating multiple clusters) and the consensus function (combining multiple clusters) are the two primary stages of CE) Akbari et al. 2015). The production of different clustering solutions involves five methods: (1) the initialization of various parameters such as cluster centers in the K-means clustering method to create a homogeneous set of data; (2) the use of different clustering algorithms called heterogeneous groups; (3) the creation of different subsets of features; (4) the implementation of different subsets of objects; and (5) projection to subspace. The second stage in the clustering ensemble (i.e., consensus function) refers to combining these solutions to obtain a final accurate result. The six families of techniques that make up the consensus function are hypergraph methods, voting approaches, information theoretic methods, co-association-based methods, mixture models, and evolutionary algorithms. Several CE approaches exist in the literature, which can be classified into four categories:

1. Vote approaches: The Hungarian method can be used to re-label the two basic clusters, and then the data is assigned to the clusters by a voting process.
2. Feature-based approaches: These methods identify the best subset of features. Attributes can be thought of as a middle ground where the algorithms perform clustering to get the optimum results in predicting class labels (Kuhn 1955).

3. Pairwise approaches: These approaches produce a co-association (CA) matrix by comparing the number of cluster instances, which is then utilized to generate a new clustering algorithm (Fred and Jain 2005).
4. Graph-based approaches: These methods create the edge and then perform a weighting process based on the edges' similarity. Finally, they create the final partition by cutting the chart's edges (Strehl and Ghosh 2002; Hamidi et al. 2019; Ma et al. 2020).

The CES technique was adopted by Azimi and Fern (2009), Parvin et al. (2012), and Saidi et al. (2017) by selecting a subset of clusterings to improve the clustering ensemble solution. The diversity and quality of the base clusters are two essential elements that improve CE, to the researchers mentioned above. Different methods for the enhancement of quality and diversity have been considered in the literature. The accuracy between clusters is measured using two criteria: Normalized Mutual Information (NMI) (Strehl and Ghosh 2002) and Adjusted Round Index (ARI) (Akbari et al. 2015). According to Fern and Brodley (2004), low diversity reduces performance improvement; consequently, a high-diversity subset of partitions should be considered. They compared the effects of base partition diversity and stability on selective ensemble performance. Kuncheva and Hadjitodorov (2004) expanded on Fern and Brodley's work by suggesting that the number of clusters in each base division be set at random in order to be greater than the expected number. The best selective clustering ensemble algorithms rely on diversity to pick a subset of clustering results. Naldi et al. (2013) provided a number of clustering validity indicators. Alizadeh et al. (2014) developed an asymmetric criterion to evaluate the cluster-partition relationships. This criterion is used to determine which cluster is the best. The co-matrix was then constructed using the Extended Evidence Accumulation Clustering (EEAC) method. Akbari et al. (2015) proposed the Hierarchical Cluster Ensemble Selection approach (HCES), in which a subset of cluster members is determined using three techniques: average-linkage, single-linkage, and complete-linkage agglomerative. HCES determines the subset of cluster members by considering both diversity and quality as important factors. In addition, their research developed a novel relative diversity measure. The criteria of independence and diversity in cluster ensemble selection were introduced by Yousefnezhad et al. (2016). They calculated the independence of two basic clustering algorithms using an exploration criterion based on the technique of turning code into graphs in software testing. Furthermore, homogeneity, a novel similarity criterion, was proposed to assess the diversity of the underlying results. For semi-surveillance clustering, Fozieh Asghari et al. (2017) presented an approach based on the WOC theory. Their method included a semi-surveillance clustering algorithm, a strategy for evaluating and selecting early results based on the feedback mechanism, and a new criterion for evaluating the variability of the underlying results by decreasing the size of data based on monitoring information. In regard to consistency under constraints, diversity among group members, and overall set quality, Yang et al. (2017) provided a new approach to solving the constraint group set selection problem, referring to it as a hybrid optimization problem. Li et al. (2018) examined a technique considering the discrepancies between the objectives of the group selection stage and those of the group integration stage in the selected clustering set. Yang et al. (2017) looked at cluster ensemble selection in the context of various limitations. Three criteria were used: compatibility under limitations, ensemble quality, and diversity among ensemble members. They employed several ways to use restrictions in group selection rather than library production, taking quality and diversity into account. Huang et al. (2018) developed an ensemble clustering strategy based on the ensemble-driven cluster uncertainty assessment and a local weighting technique. They used the entropy criteria to estimate cluster uncertainty without having access to the original data features, by evaluating the cluster label throughout the full ensemble. Then, they developed a collaboration matrix with local weighting and provided two consensus functions: LWEA and LWGP. To evaluate the quality of clusters, cluster similarity, and object similarity, Bai et al. (2019) proposed a weighted consensus criterion based on the entropy of information. To generate a high-quality final cluster, they presented weighted feature consensus, weight labeling consensus, and pairwise similarity consensus methods. To increase diversity and quality in base clusters, Zhao et al. (2018) introduced a sequential clustering algorithm based on SECG, minimizing anticipated entropy, and normalized cross-information to achieve high-quality basic clustering results and variability for mixed data in group clustering. Unlike many other basic clustering production algorithms, the proposed algorithm takes into account correlations between different basic clusterings while producing such clusterings. The sample frequencies employed by Li et al. (2019) differed between clusters. They presented a clustering approach that took into account sample stability to reflect differences (CEs). The samples are categorized into two groups by their algorithms: core and halo. The cluster core samples are then used to find a clear structure; on the other hand, the samples from the cluster halo are gradually assigned to the clear structure. Ma et al. (2020) concentrated on clustering and multiple selection while taking quality and diversity into consideration. They provided the results of the MCAS approach with two main methods of combining the selected solutions: direct combination and clustering. Bagherinia et al. (2020) suggested a fuzzy clustering ensemble based on fuzzy cluster-level weighting without access to the object attributes base. They used the entropic criterion to assess cluster unreliability and the reliability cluster-based index to consider the reliability of fuzzy clusters. To build the final clustering, two consensus functions were provided: (1) the fuzzy weight correlation matrix built from the ensemble consensus, and (2) the fuzzy clustering reliability-based graph partitioning fuzzy clustering ensemble (RBGPFCE). Mahmoudi et al. (2021) proposed a two-level clustering-based consensus function method (CFTLC). Using the average of the hierarchical clusters on the cluster similarity matrix, their suggested method produced a collection of meta clusters. The basic BCs with the greatest cluster–cluster similarity are combined first via CFTLC. The object-cluster similarity is then used to assign each data point to a meta cluster. Finally, at the output of the clustering consensus, the meta cluster was considered the consensus cluster. Banerjee et al. (2021) proposed using several set selection procedures on weighted clusters after creating a new clustering set. To accomplish this, they first created a cluster-level surprising measure derived from the principle of agreement and disagreement among all clusters in the ensemble. In the second step, they calculated the merit of a clustering by adding up the cluster-level surprisal values of the constituent clusters and mathematically demonstrated that this clustering-level surprisal measure could be treated as a valid entropy measure. This measure can be used to prioritize clusterings in forming a quality consensus. In the final phase, they suggested a coupled ensemble selection method based on the clustering level surprise criteria. The iterative procedure took one of the clusters from the time set and built a consensus, keeping the quality of the consensus from dropping monotonically. The consensus was calculated in each iteration by applying the Weighted Hierarchical Agglomerative Clustering Ensemble (WHAC) algorithm to the correlation matrix, which was changed by the cluster level established in step one. In all the clustering methods presented so far, the main features of the data are available; as a result, the final clustering is created by selecting the appropriate criteria of the features, which leads to the quality and variety of clustering and selection of a subset of the data. Wang et al. (2022) developed a graph-based clustering model using row vectors as vertices and row vector similarity as edges. Then, they used the Markov process to derive the graph-based clustering model. Li et al. (2022) proposed a clustering framework based on density hierarchical clustering (AHC) methods, which includes clustering methodology and similarity evaluation. The proposed approach is a model selection-based Meta-Clustering Ensemble technique (MCEMS). To enhance Extracted Clusters (EC), MCEMS used a two-weight approach to address the model selection problem. Several individual AHC approaches clustered the data with different features to produce major clusters. They estimated the similarity between cases based on the findings of several methods. The MCEMS system was used for the re-clustering purposes. After clustering, the ideal number of ideal clusters was calculated by merging comparable clusters and considering a threshold. Finally, the similarity of samples with meta-clusters was determined. And each sample was assigned to a meta-cluster with the highest similarity to produce the final clusters. However, the problem is that if only the primary clusters are available and there is no access to the main features for various reasons, then the primary library will be small, and selecting a subset of the clusters for the new clustering will not be effective. This article proposes a way to generate new clustering with small libraries without access to basic features in such a way that the clustering labels could be close to reality.

3 Preliminaries

3.1 Representation of base clusterings

This section introduces the general formulation of the ensemble clustering problem. Let $o = \{o_1, o_2, \ldots, o_n\}$ represent a dataset where o_i, $i = 1, 2, \ldots, n$ denotes the i-th data object and n represents the number of objects in the$dataset$. Consider a dataset with l partitions (or clusterings), each of which is handled as a base clustering and contains a specific number of clusters. The ensemble of l base clusterings is formalized as follows: Let$H = \{h^1, h^2, \ldots, h^l\}$ be a set of clusterings where h^i, $i = 1, 2, \ldots,$ l is a clustering. Each clustering can be represented as a set, vector, or binary value.

Table 2 Illustrative cluster ensemble problem with$l = 4$, and $m = 3$. original label vectors (left) and equivalent hypergraph representation with 12 hyperedges (right)

	p_1	p_2	p_3	p_4
o_1	1	1	1	1
o_2	1	1	1	1
o_3	1	2	2	1
o_4	2	2	2	1
o_5	2	2	3	2
o_6	2	2	3	2

(a)

	h^1			h^2			h^3			h^4		
	c_1	c_2	c_3	c_1	c_2	c_3	c_1	c_2	c_3	c_1	c_2	c_3
o_1	1	0	0	1	0	0	1	0	0	1	0	0
o_2	1	0	0	1	0	0	1	0	0	1	0	0
o_3	1	0	0	0	1	0	0	1	0	1	0	0
o_4	0	1	0	0	1	0	0	1	0	1	0	0
o_5	0	1	0	0	1	0	0	0	1	0	1	0
o_6	0	1	0	0	1	0	0	0	1	0	1	0

(b)

3.1.1 Set representation

Let $h^m = \{c_1^m, c_2^m, \ldots., c_{k^m}^m\}$ be a set of base clusterings, where $m = 1, 2, \ldots .l$, c_i^m is the i-th cluster inh^m, and k^m denotes the number of clusters inh^m. Each cluster is a set of homogeneous objects. As an example, let$o = \{o_1, o_2, o_3, o_4, o_5, o_6\}$. Four clusterings are presented in the dataset o by set representation, which are$h^1 = \{\{o_1, o_2, o_3\}, \{o_4, o_5, o_6\}\}$,$h^2 = \{\{o_1, o_2\}, \{o_3, o_4, o_5, o_6\}\}$, $h^3 = \{\{o_1, o_2\}, \{o_3, o_4\}, \{o_5, o_6\}\}$ and $h^4\{\{o_1, o_2, o_3, o_4\}, \{o_5, o_6\}\}$.Obviously, the entire dataset is represented by the union of all clusters in the same base clustering, i.e., $\forall\, h^m \in$ H, $\bigcup_{i=1}^{n^m} c_i^m = o$. In the same base clustering, different clusters do not overlap: $\forall c_i^m, c_j^m \in h^m$ s.t. $i \neq j$, $c_i^m \bigcap c_j^m = \emptyset$. Let $cls^m(o_i)$ denote the cluster in $h^m \in H$ to which object o_i belongs. That is, if o_i belongs to the k-th cluster in h^m, i.e., $o_i \in c_k^m$, then we have $cls^m(o_i) = c_k^m$.

3.1.2 Vector representation

In the case of order objects, the label vector $p \epsilon N^n$ is a vector of integer numbers corresponding to the objects, such that if $o_i \in c^m$, $c^m(x_i) = m$. The label vectors of the above examples are $p_1 = (1, 1, 1, 2, 2, 2)$, $p_2 = (1, 1, 2, 2, 2, 2)$, $p_3 = (1, 1, 2, 2, 3, 3)$, and $p_4 = (1, 1, 1, 1, 2, 2)$, respectively (see Table 2(a)).

3.1.3 Binary representation

The binary representation indicates a binary matrix$H = \{\, h^{ij} | i = 1, 2, \ldots, n;\ j = 1, 2, \ldots, k\} \in \{0, 1\}^{n*k}$,

$$\text{where}\, h^{ij} = \begin{Bmatrix} 1\; if\; object\; i\; include\; j - th\; cluster \\ 0\; else \end{Bmatrix} \tag{1}$$

For example, the binary representations of $h^1, h^2, h^3 and h^4$ are shown in Table 2 in the example above (b).

3.2 Diversity and quality measures

This section briefly discusses the concepts of base cluster quality and accuracy, as well as their implications for base clustering. After a set of base clustering findings, a number of samples stay in one category consistently, while others alternate between groups on a regular basis. The tendency of a sample to change its group can be used to describe this phenomenon. This tendency is useful for a variety of clustering ensemble tasks, such as assessing the quality of the base clustering results (Li et al. 2019). The cluster set's quality and diversity of fundamental partitions are two important factors; the former reflects the accuracy of the group's members, while the latter measures the diversity of the group's predictions. If the labels on one partition do not match the labels on the other, the two partitions are different. To quantify the diversity or quality of partitions, NMI (Strehl and Ghosh 2002) and ARI (Akbari et al. 2015) are often used. Definition 1 formally defines NMI and ARI.

Definition 1 Let $h^a = \{c_1^a, c_2^a, \ldots, c_{ka}^a\}$ and $h^b = \{c_1^b, c_2^b, \ldots, c_{kb}^b\}$ be two base clusterings for the data set X; ARI and NMI between them are given as follow:

$$ARI(h^a, h^b) = \frac{\sum_{i=1}^{k_a}\sum_{j=1}^{k_b}\binom{n_{ij}}{2} - t_3}{\frac{1}{2}(t_1 + t_2) - t_3} \tag{2}$$

where $t_1 = \sum_{i=1}^{ka}\binom{n_{ia}}{2}$, $t_2 = \sum_{j=1}^{kb}\binom{n_{jb}}{2}$, $t_3 = \frac{2t_1t_2}{n(n-1)}$

and

$$NMI(h^a, h^b) = \frac{-2\sum_{i=1}^{ka}\sum_{j=1}^{kb} n_{ij}\log\left(\frac{n \cdot n_{ij}}{n_{ia} \cdot n_{bj}}\right)}{\sum_{i=1}^{ka} n_{ia}\log\left(\frac{n_{ia}}{n}\right) + \sum_{j=1}^{kb} n_{bj}\log\left(\frac{n_{bj}}{n}\right)} \tag{3}$$

where in both equations,h^a $=\{c_1^a, c_2^a, \ldots, c_{ka}^a\}$ and $h^b = \{c_1^b, c_2^b, \ldots, c_{kb}^b\}$ with k_a and k_b clusters, respectively, are two clusterings on dataset D with n samples; n_{ij} signifies the number of common objects in cluster c_i in clustering h^a and in cluster c_j in clustering h^b; n_{ia} denotes the number of objects in cluster c_i in clustering h^a; and n_{bj} stands for the number of objects in cluster c_j in clustering h^b. NMI affects both clustering performance and is easy to calculate. The lower the NMI value, the higher the diversity is. NMI can help express a reliable estimate of how much information is shared between any two clustering solutions. Some samples remain consistently in one category after achieving a set of base clustering results, whereas others regularly switch between groups. The tendency of a sample to change its group can be used to describe this phenomenon. External and internal diversities are two types of diversity metrics. When class labels are provided, the external diversity metric is defined as follows, using a quality measure such as NMI or ARI.

Definition 2 Given the external diversity, the measure is computed as follows:

$$\text{Diversity}\,(\overline{h}, h_i) = 1 - \text{quality}(\overline{h}, h_i) \tag{4}$$

where $\overline{h}$ is the known class label and h_i, $i = 1, 2, \ldots, l$ is clustering. Remember that NMI in this study is employed as a measure of quality. The average of diversity is

$$\text{De} = \frac{1}{L}\sum_{i=1}^{L} \text{Diversity}\ (\overline{h}, h_i) \tag{5}$$

Pair-wise and non-pair-wise diversities are two types of internal diversity. Each clustering is implicitly picked as a class label in pair-wise diversity, and other clusterings are measured by the chosen class label. The following formula is used to calculate diversity:

$$\text{Diversity}\ (h_i, h_j) = 1 - \text{quality}(h_i, h_j) \tag{6}$$

where $i \neq j$, $i = 1, 2, \ldots, l$. The diversity measure can be averaged as follows:

$$D_p = \frac{1}{l(l-1)}\sum_{i=1}^{l} \sum_{\substack{j = 1 \\ j \neq i}}^{l} \text{Diversity}\ (h_i, h_j) \tag{7}$$

The following is the definition of the non-pair-wise diversity measure:

$$\text{Diversity}\ (h^*, h_i) = 1 - \text{quality}(h^*, h_i) \tag{8}$$

where $i = 1, 2, \ldots, l$ and h^* stands for the result of using a consensus function.

The following is the average of diversity:

$$D_{np} = \frac{1}{l}\sum_{i=1}^{l} \text{Diversity}\ (h^*, h_i) \tag{9}$$

According to Kuncheva and Hadjitodorov (2004), ensembles with a larger variety of individual diversities perform better than those with a smaller range. As a result, a new relative diversity measure is proposed in this research based on the h_n derived by a consensus function and h_i, $i = 1, 2, \ldots.$, l that are ensemble members:

$$\text{Diversity}\ (h^*, h_i) = |quality(h^*, h_i) - quality(h^*, h_j)| \tag{10}$$

In comparison with the reference consensus partition,h^*, the relative diversity measure determines the absolute distance between qualities of h_i and h_j.

3.3 Jaccard measure

The Jaccard similarity measure is also used in this study. The Jaccard resemblance index (also known as the Jaccard similarity coefficient) analyzes members from two sets to see which are similar and which are distinct (Sulaiman and Mohamad 2012). This measurement is the same for both data sets, and it ranges from 0 to 100%. The closer the two populations are, the greater the percentage. Although it is simple to use, it has a high sensitivity to sample size and may produce incorrect results, particularly with very small samples or data sets with missing observations. The formal definition of the Jaccard similarity measure is provided in Definition 3.

Definition 3 The Jaccard similarity pair of sets S and T is defined as:

$$\text{Jaccard similarity} = \frac{\left|S \bigcap T\right|}{|S|+|T|-\left|S \bigcap T\right|} \tag{11}$$

where $\left|S \bigcap T\right|$ calculates the number of members that are common to both sets, $|S|$ is the total number of members in set S, and $|T|$ is the total number of members in set T.

4 Cluster ensembles extraction for knowledge reuse

The aim of ensemble clustering is to produce a stronger and more robust clustering by combining numerous base clusterings in the ensemble H. In this paper, a new set of clusters (instead of clusterings) is generated with greater size and diversity. For example, a chain store with M distributed stores divides its customers into three clusters: active, intermediate, and inactive. Each store may have defined these three clusters based on different criteria. Therefore, there will be three clusters with different criteria for n customers. The number of stores is small; thus, with low clusterings, the basic data that feature the customers is not available. The current study aims to increase the diversity of clusters by extracting them from base clusters. So far, researchers working in this domain have first created a large library and then selected different clusters from among them. But if the library is small, selecting a subset of the library as well as the consensus function will not be suitable for this library. For that reason, our purpose will be to create a new library consisting of clusters from the old library, which includes clustering. The present paper proposes a new CE approach. Note that since compilers can only examine cluster labels and do not have original features, it is a framework for reusing knowledge. Cluster tags are symbolic, so the communication problem must be solved. In addition, the form of the given clusters might vary depending on the clustering method and the particular perspective of the data used in that method. Therefore, in this study, first, the main clustering was changed to binary. Next, the set information was used to estimate the cluster similarity based on the Jacquard similarity measurements. A new set of clusters was selected and created from the main cluster. The existence of similar clusters in a group, according to this article, can be a useful indicator for cluster extraction. Thus, new clusters were created with logical relationships. Finally, the new clusters were extracted with the consensus function. The framework of the cluster group extraction method used in this paper is shown in Fig. 1.

4.1 Generating base clusterings

It has been demonstrated that diversity is essential for improving ensemble learning performance (Jia et al. 2011) and has an impact on the quality of the final solution. The K-means algorithm creates various clusters with different initial values and a choice of k between $\left[2, \sqrt{n}\right]$ (k number of clusters in a cluster). This is useful for creating different types of clusters (Hamidi et al. 2019). Available evidence indicates that if the size of clusters and their variety are small, the clusters created by K-means are variable, but with low accuracy. The question is how we can produce clustering with high accuracy if there is a set of clusterings with small size and low diversity. Since there are no real small base clusterings, the current study attempted to produce it experimentally with the K-means algorithm as follows. The K-means algorithm was employed with constant value of$k$, and different number of initial clusters size varying from 30 to 50 (see Fig. 3). Since in the next steps there are only the cluster labels and these labels are symbolic, the cluster labels are merged into binaries to integrate them.

4.2 Selecting set of clusters

Our goal in comparing the sets is to see how similar they are in composition, and the Jaccard Index is the most straightforward approach to this task. This index is a proportion of how many objects two sets share out of the total number of objects they have, and is the measurement of asymmetric information on binary variables. This index is used to calculate the separation of the clusters due to the shape of the clusters and their overlap. The generic formulation of the ensemble clustering problem is presented in this section. Assume $O = \{o_1, \ldots, o_N\}$ is a dataset, o_i refers to the $i\text{-}th$ data object, and the number of objects in O is N. Consider the dataset$O$, which has l partitions (or clusterings), each of which is treated as a base clustering and has a specified number of clusters. Formally, the ensemble of l base clusterings is denoted as follows: $H = \{h^1,h^2, \ldots, h^l\}$, where $h^k = \{c_1^k, c_2^k, \ldots, c_{n^k}^k\}$ is the k-th base clustering, and c_i^k denotes the $i\text{-}th$ cluster inh^k, and n^k is the number of clusters inh^k. Each cluster consists of a collection of data elements. The union of all clusters in the same base clustering, obviously, encompasses the complete dataset, i.e., $\forall\ h^k \in H$, $\bigcup_{i=1}^{n^k} c_i^k = O$. Within the same base clustering, different clusters do not intersect with each other, i.e., $\forall c_i^k, c_j^k \in h^k$ s.t. $i \neq j$, $c_i^k \bigcap c_j^k = \emptyset$. Let $cls^k(o_i)$ denotes the cluster in $h^k \in H$ to which object o_i belongs. That is, if object o_i belongs to the $m\text{-}th$ cluster inh^k, i.e., $o_i \in c_m^k$, then we have $cls^k(o_i)$=c_m^k.

The Jaccard index frequently outperforms other methods for evaluating binary two-vector similarity as well as in high-dimensional data sets (Strehl and Ghosh 2002). In this work, each cluster is treated as a binary vector, and the degree of similarity between two clusters was computed using this index. For example, according to Fig. 2, $c_1^1 = (1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0)$, and $c_1^2 = (1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0)$ therefore the jaccard $(c_1^1, c_1^2) = \frac{5}{7}$. At this point, the cluster's similarity to all other clusters in the clustering is determined. For this aim, formula (12) was

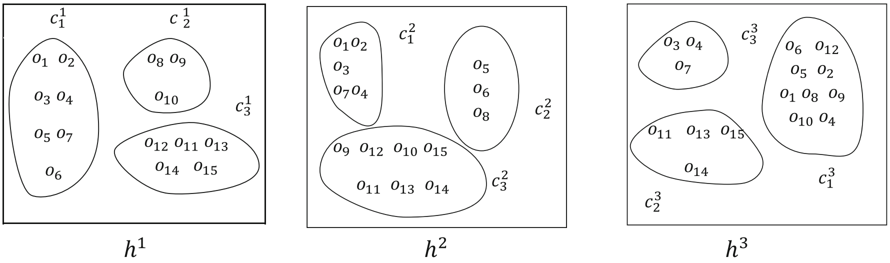


Fig. 2 An ensemble of three base clusterings: h^1, h^2, and h^3

presented to compute similarity using Jaccard, and the average similarity of the clusters was utilized as a criterion to choose the selected clusters.

Definition 4 Given the base clusterings H, the similarity of cluster C_j with the entire ensemble H is computed as follows:

$$w(c_j^k, h^k) = \frac{1}{l-1} \sum_{\substack{r=1 \\ r \neq n}}^{l} \max\{jacard(c_j^k, c_s^r)\} \tag{12}$$

Inc_j^k,$j = 1, 2, \ldots, n^k$,$k = 1, 2, ..., l$, and$s = 1, 2, ..., n^r$. Figure 3 and Table 3 show how to compute cluster similarity using an ensemble of three base clusterings. For the dataset $o = \{o_1, o_2, \ldots .o_{15}\}$ with 15 data objects, three base clusterings (h^1, $h^2 and h^3$) are created, each consisting of three clusters (see Fig. 2). Of the three clusters in h^1, c_1^1 contains seven objects, c_1^2 three objects, and c_1^3 five objects. On the other hand, of the three clusters in h^2, c_2^1 contains six objects, c_2^2 four objects, and c_2^3 five objects. Cluster c_1^1 contains eight objects that belong to three different clusters inh^2. Then, the similarity of the three clusters in h^1 and h^2 was computed. According to Definition 1, with Jaccard (c_1^1, c_1^2) = 0.75, then Jaccard (c_1^1, c_2^2) = 0.25 and Jaccard (c_1^1, c_3^2) = 0. hence the maximum similarity of c_1^1 in base clustering h^2 is 0.75. Similarity Jaccard (c_1^1,c_m^3) can be obtained. The cluster c_1^1 has the highest similarity jaccard with

Fig. 3 Illustration of clusters extraction from base clustering

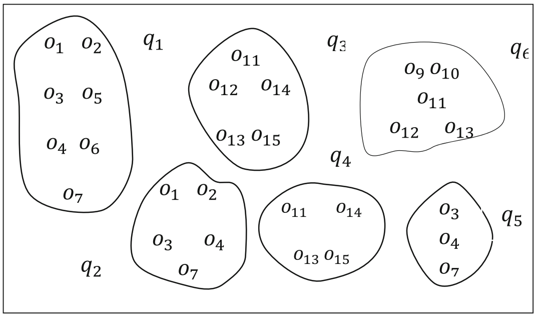


Table 3 Computation of cluster similarity for the clusters in the ensemble shown in Fig. 2

Base clustering	Cluster	Max similarity	Cluster similarity	θ	Result

h^1	c_1^1	Max(Jaccard(c_1^1, c_1^2), Jaccard(c_1^1, c_2^2), Jaccard(c_1^1, c_3^2))
		Max(Jaccard(c_1^1, c_1^3), Jaccard(c_1^1, c_2^3), Jaccard(c_1^1, c_3^3))
			W(c_1^1, h^1) = 0.56	0.5	0.56

	c_2^1	Max(Jaccard(c_2^1, c_1^2), Jaccard(c_2^1, c_2^2), Jaccard(c_2^1, c_3^2))
		Max(Jaccard(c_2^1, c_1^3), Jaccard(c_2^1, c_2^3), Jaccard(c_2^1, c_3^3))
			W(c_2^1, h^1) = 0.31		

	c_3^1	Max(Jaccard(c_3^1, c_1^2), Jaccard(c_3^1, c_2^2), Jaccard(c_3^1, c_3^2))
		Max(Jaccard(c_3^1, c_1^3), Jaccard(c_3^1, c_2^3), Jaccard(c_3^1, c_3^3))
			W(c_3^1, h^1) = 0.75		0.75

h^2	c_1^2	Max(Jaccard(c_1^2, c_1^1), Jaccard(c_1^2, c_2^1), Jaccard(c_1^2, c_3^1))
		Max(Jaccard(c_1^2, c_1^3), Jaccard(c_1^2, c_2^3), Jaccard(c_1^2, c_3^3))
			W(c_1^2, h^2) = 0.64	0.5	0.64

	c_2^2	Max(Jaccard(c_2^2, c_1^1), Jaccard(c_2^2, c_2^1), Jaccard(c_2^2, c_3^1))
		Max(Jaccard(c_2^2, c_1^3), Jaccard(c_2^2, c_2^3), Jaccard(c_2^2, c_3^3))
			W(c_2^2, h^2) = 0.31		

Table 3 (continued)

Base clustering	Cluster	Max similarity	Cluster similarity	θ	Result
	c_3^2	Max(Jaccard(c_3^2, c_1^1), Jaccard(c_3^2, c_2^1), Jaccard(c_3^2, c_3^1)) Max(Jaccard(c_3^2, c_1^3), Jaccard(c_3^2, c_2^3), Jaccard(c_3^2, c_3^3))	W(c_3^2, h^2) = 0.67		0.67
h^3	c_1^3	Max(Jaccard(c_1^3, c_1^1), Jaccard(c_1^3, c_2^1), Jaccard(c_1^3, c_3^1)) Max(Jaccard(c_1^3, c_1^2), Jaccard(c_1^3, c_2^2), Jaccard(c_1^3, c_3^2))	W(c_1^3, h^3) = 0.37	0.5	
	c_2^3	Max(Jaccard(c_2^3, c_1^1), Jaccard(c_2^3, c_2^1), Jaccard(c_2^3, c_3^1)) Max(Jaccard(c_2^3, c_1^2), Jaccard(c_2^3, c_2^2), Jaccard(c_2^3, c_3^2))	W(c_2^3, h^3) = 0.51		0.51
	c_3^3	Max(Jaccard(c_3^3, c_1^1), Jaccard(c_3^3, c_2^1), Jaccard(c_3^3, c_3^1)) Max(Jaccard(c_3^3, c_1^2), Jaccard(c_3^3, c_2^2), Jaccard(c_3^3, c_3^2))	W(c_3^3, h^3) = 0.68		0.68

the cluster in which it is placed; the similarity value here is 1. Therefore, the similarity Jaccard of clusterc_1^1, on the entire ensemble H can be computed as: Max (Jaccard (c_1^1, c_1^1), Jaccard (c_1^1, c_2^1), Jaccard (c_1^1, c_3^1)) + Max (Jaccard (c_1^1, c_1^2), Jaccard (c_1^1, c_2^2), Jaccard (c_1^1, c_3^2)) + Max (Jaccard(c_1^1, c_1^3), Jaccard (c_1^1, c_2^3), Jaccard (c_1^1, c_3^3)))/($m-1$) = (1 + 0.75 + 0.42)/(3 − 1) = 0.56, where M is the number of base clustering in$H$. Accordingly, the Jaccard similarity of the other clusters in H can be calculated (see Table 2). Algorithm 1 summarizes the general ASC algorithm for greater clarity. As shown in Table 2, of the nine clusters in$H$, we compared the average similarity of the clusters with the value of θ. Given the similarity value in the table, the average similarity of some clusters will be found very low. It could indicate that, for example,

Table 4 Similarity clusters in clustering HN

Cluster	W(1)	W(2)	W(3)	W(4)	W(5)	W(6)
Similarity	0.56	0.75	0.64	0.67	0.51	0.68

these clusters have more noise. Therefore, θ is set as a threshold for similarity, then the clusters that are less similar to θ are removed. Next, new clusters (HN) are created with the remaining clusters (Fig. 3). The similarity of the remaining clusters is presented in Table 4.

Algorithm 1 Average similarity of two clusters (ASC)

```
Input: $H = \{h^1, h^2, \ldots, h^l\}$ set of $l$ base clusterings
Output: Similarity matrix ($w$)
Initialization: Let $w = \emptyset$ denote the empty set
for $i = 1$ to $l$
    for $j = 1$ to $n^k$
        $w(c_j^k, h^k) = \frac{1}{l-1}\sum_{\substack{r=1 \\ r \neq n}}^{l} \max\{Jaccard\ (c_j^k, c_s^r)\}_{s=1}^{n^r}$
    end for
end for
```

4.3 Clusters extraction

This section describes the framework of the extraction method. Two clusters are randomly selected with the help of the Jaccard measure in $w(i)$ from$HN = \{q_1, q_2, \ldots, q_r\}$, then the similarity of these two clusters is calculated. If the similarity of these two clusters is more than the threshold, these two clusters are compared to make sure that the clusters are not completely opposite. To compare the clusters, they are summarized logically (see Table 5(a)). If the result of the logical sum of the clusters is zero, it means that two clusters are not completely different (Table 5(b)). At the next step, the clusters are mutated. To create mutations, two methods are used: logic combination (OR) and logic multiplication (AND). In the former, a new cluster, called$q\prime$, is produced from the logic combination of two clusters. With the help of the Jaccard measure, the similarity of $q\prime$ with two parent clusters is determined. If the similarity of $q\prime$ and each of the parent clusters is greater than or equal to the similarity of the two parent clusters, then $q\prime$ is added as a new cluster to the set of children. In the latter, AND, after logically multiplying two clusters, a new cluster, calledq'', is produced. Similar to the previous step, the similarity of q'' with the parent clusters is calculated. If it is greater than or equal to the similarity of two parent clusters,q'' will also be added to the set of

Table 5 Computing the logical sum of clusters (a), removed clusters whose logical sum is zero (b)

	q_1					q_2				q_3			q_4		q_5
	q_2	q_3	q_4	q_5	q_6	q_3	q_4	q_5	q_6	q_4	q_5	q_6	q_5	q_6	q_6
x_1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
x_2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
x_3	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0
x_4	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0
x_5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
x_6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
x_7	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
x_8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
x_9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
x_{10}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
x_{11}	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0
x_{12}	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
x_{13}	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0
x_{14}	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0
x_{15}	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0

(a)

	q_1		q_2	q_3		q_4
	q_2	q_5	q_5	q_4	q_6	q_6
x_1	1	0	0	0	0	0
x_2	1	0	0	0	0	0
x_3	1	1	1	0	0	0
x_4	1	1	1	0	0	0
x_5	0	0	0	0	0	0
x_6	0	0	0	0	0	0
x_7	0	1	0	0	0	0
x_8	0	0	0	0	0	0
x_9	0	0	0	0	0	0
x_{10}	0	0	0	0	0	0
x_{11}	0	0	0	1	1	1
x_{12}	0	0	0	0	1	0
x_{13}	0	0	0	1	1	1
x_{14}	0	0	0	1	1	1
x_{15}	0	0	0	1	1	1

(b)

child clusters as a new cluster. For example, with$i = 3$, 4; the similarity of q_3 and q_4 in HN with Jaccard measure equaled 0.8. Since the similarity of these two clusters is higher than the threshold, two logical operations (AND and OR) are applied. In the first step,q_3 and q_4 are logically combined and a new cluster like $q\prime$ is created; Jaccard (q_3, $q\prime$) = 1, and Jaccard (q_4, $q\prime$) = 0.8. Since the similarity of $q\prime$ is greater than or equal to that of both parent clusters (q_3,q_4), the cluster $q\prime$ is added to the set of extraction clusters. At the next step, q_3 and q_4 are logically multiplied. A new cluster like $q^{''}$ is created. The similarity between $q^{''}$ and both parents are calculated with the help of the Jaccard. If the similarity is greater than or equal to that of both parents, then $q^{''}$ is added to the set of extraction clusters. The previous steps are continued until the number of children is equal to the number of parents in the initial cluster set (Table 6). The framework of the cluster extraction approach is given in Fig. 4.

Table 6 Computing the similarity of clusters

	q_1		q_2	q_3		q_4
Similarity	q_2	q_5	q_5	q_4	q_6	q_6
	0.57	0.42	0.4	0.8	0.71	0.57

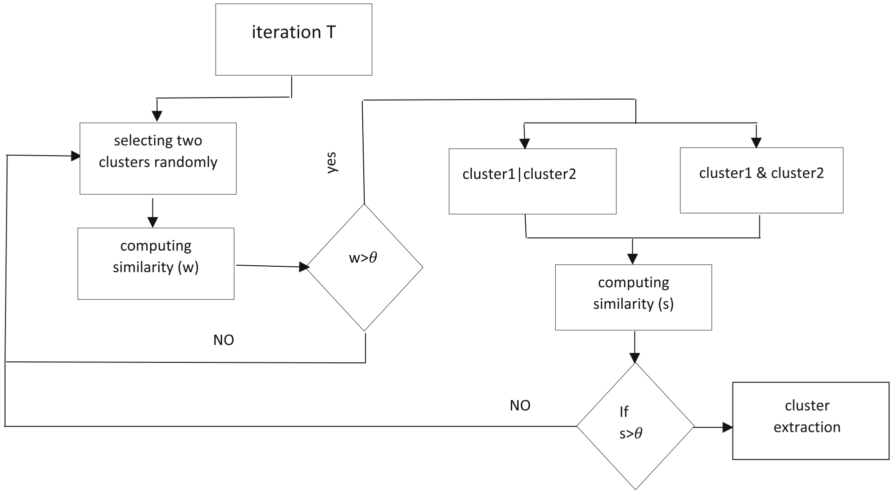


Fig. 4 Framework of the cluster extraction method

Algorithm 2 Extracting Clusters Algorithm (ECA)

```
Input: $HN = \{ q_1, q_2, \dots, q_r\}$, set of clusters, and $w$, similarity matrix, both obtained from Alg1.
Output: $EHC$={ $eq_1$ , $eq_2$, …. , $eq_s$ } set of $s$ extracted clusters.
Initialization:
Let p= $\emptyset$ denote the empty set
for r = 1 to M
    i, j= {Produce two random numbers in the range of 1 to the maximum number of arrays w(i)}
    D=$q_i|q_j$
    if $D \neq 0$ then
        w = Jaccard ( $q_i$, $q_j$)
    if w>= threshold then
        $eq$= $q_i$ & $q_j$
    if  Jaccard ( $eq$, $q_i$ ) >= Jaccard ( $q_i$, $q_j$)  OR  Jaccard ( $eq$, $q_j$ ) >= Jaccard ( $q_i$, $q_j$)  then
        $EHC$ =$EHC$ $\cup$ { $eq$ }
    $eq$= $q_i$ | $q_j$
    if  Jaccard ( $eq$, $q_i$ ) >= Jaccard ($q_i$, $q_j$)  OR  Jaccard ( $eq$, $q_j$ ) >= Jaccard ($q_i$, $q_j$)  then
        $EHC$ =$EHC$ $\cup$ { $eq$ }
end for
```

4.4 Consensus function for a set of different clusters

The goal of consensus function is to find a better consensus cluster result and also directly determining the quality of the final solution. As a result, it is regarded as the most important component of an ensemble. The consensus functions that have been presented so far use a combination of clustering members to generate the final partition. Based on the information provided by the CE members, we propose two novel consensus functions in this article: the Cluster-based Extracted Partitioning Algorithm (CEPA) and the Meta-Cluster Extracted Algorithm (MCEA).

4.4.1 Cluster-based extracted partitioning algorithm

In this section, a similarity matrix is proposed based on the extracted clustering solutions in an ensemble, in which each pair of data points is clustered together. Definition 5 formally defines Cluster-based Extracted Partitioning Algorithm (CEPA) as follows:

Definition 5

$$EHC = \{eq_1, eq_2, \ldots., eq_s\}$$

where eq_i denotes the set of clusters.

A well-acknowledged viewpoint is that two items have a similarity of 1 when they are in the same cluster; otherwise, they have a similarity of 0. As a result, each clustering might have a matrix with n rows and n columns. The entry-wise maximum of r in matrices that represent the r sets of groupings will produce a finer-resolution overall similarity matrix (matrix M). S is the fraction of clusterings in which two objects are in the same cluster, and it may be obtained in one sparse matrix multiplication:

$$S = \max(sum(EHC(i, :)))$$

where $i = 1, 2, \ldots, n$, and ":" stands for all columns.

$$M = \frac{1}{s}(EHC \cdot EHC^t) \tag{13}$$

Now, the final clustering can be created using the cut graph on the matrix M (vertex = cluster, edge weight = similarity).

4.4.2 Meta-cluster extracted partitioning algorithm

In this subsection, the second algorithm is to group and collapse related hyperedges and assign each object to the collapsed hyperedge in which it participates most strongly. The Meta-Cluster Extracted Partitioning Algorithm (MCEA) is based on the EHN set's clusters, which are extracted using clustering principles. MCEA operates by grouping and collapsing similar hyperedges before assigning each object to the collapsed hyperedge with the most active edge. Objects are used as graph nodes in the construction of a meta graph. The weight of the edges is proportional to the vertices' similarity. The Jaccard binary measurement is the appropriate similarity criterion for the graph's edge. Formally, the edge weight w_{eq_i,eq_j} between two vertices eq_i, eq_j, as defined by the binary Jaccard measure of the associated indicator vectors eq_i andeq_j, is formalized as follows: $w_{eq_i,eq_j} = \frac{eq_i^t.eq_j}{||eq_i||_2^2 - ||eq_j||_2^2 - eq_i^t.eq_j}$. At this stage, the clusters are divided into k clusterings using the METIS (Karypis and Kumar 1998) diagram partitioning package. In other words, each meta-graph cluster contains a set of clusters. A meta-clustering represents a set of appropriate labels since each vertex in the meta-graph represents a separate clustering label.

5 Experiments

This section summarizes the results of the study's trials on a number of real-world datasets in order to compare the proposed strategy with state-of-the-art ensemble clustering approaches.

5.1 Datasets and evaluation methods

Based on the NMI values, we compared CEE solutions to BC solutions in our experiments. The studies were conducted on real data sets with known true natural clusters. Because our data sets were labeled, we could use external criteria to evaluate the quality of the clustering solutions (Hadjitodorov et al. 2006). Although CEE extracts new clusterings from BC without accessing the data, we need access to the original features to generate BC. In this paper, the K-means algorithm with different initial cluster center locations generates BC with nearly identical qualities (Akbari et al. 2015). The mismatch between the structure defined by clustering and the structure defined by class labels was measured using external criteria. All of the experiments were repeated ten times, with the findings averaged across each dataset. Ten genuine data sets were used to assess the performance of CEE. The real data sets came from the UCI machine learning repository.[1] Table 7 contains the specifics of these data sets. BC was obtained by K-means with true k and 50 iterations (the number of BC was set to $L = 50$). The ASC algorithm was used to extract BCs of the size ($M = L = 40$) at the first stage of our experiment. The ECA algorithm was then used to extract different ECs in the second half, with the number of ECs ranging from 10 to 100 with incremental steps of 10. This approach led to a variety of qualities and diversities. It is worth noting that diversity and quality are two essential aspects that influence the quality of the final solution. To measure the success of clusterings, two extensively used assessment measures, normalized mutual information (NMI) (Strehl and Ghosh 2002) and adjusted rand index (ARI), are used. It should be noted that higher NMI and ARI values indicate better clustering results (Huang et al. 2018). In this work, for each benchmark dataset, 100 possible clusterings were randomly formed. The range of NMI was set to 0–1.

[1] http://www.ics.uci.com/mlearn/MLRespository.html.

Table 7 Properties of selected UCI data sets

Number	Data set	Data size (n)	Dimension (d)	$No.clusters(k)$
1	Iris	150	4	3
2	Ecoli	336	7	8
3	IS	2310	7	7
4	Landsat	6435	36	6
5	Leukemia	72	3572	2
6	Splice	3190	60	3
7	Wine	178	13	4
8	Yeast	1484	8	10
9	Satimage	6435	36	7
10	User knowledge modeling	145	5	4

The proposed methods and the baseline methods are evaluated by their average performance over a large number of runs, where the clustering ensemble for each run is constructed by randomly selecting M-based clusterings from the pool in order to rule out the factor of getting lucky occasionally and provide a fair comparison. $M = 40$ is commonly used as the ensemble size.

5.2 Choices of Parameter θ

Parameter θ determines the average similarity between clusters. If the average similarity is not less than the standard value, it leads to a stronger cluster similarity effect. Since the K-means algorithm leads to the production of diverse clusters, the average cluster similarity will also be dynamic. For each dynamic value of θ, the proposed average similarity was performed between the base clusters 20 times. The average scores of the similarity of clusters with different dynamic parameters θ are shown in Fig. 5. We selected the value t at a point from which the density of similar clusters was high and before that, the density of similar clusters was low. For example, in the Iris database, the value of θ was set to 0.65, and in the Ecoli database, it was set to 0.1. Therefore, in different samples, the value of t differed.

5.3 Comparison with base clusterings

Clustering ensemble is an approach that combines basic clusters to achieve a consensus clustering that could probably improve the quality and robustness of the final results. In this section, a comparison is made between the Extracted Clusters (EC) and the base clusterings. For each benchmark dataset, the proposed cluster extraction method was run 10 times. Each time, the ensemble of base clusterings was picked at random from the pool. Figure 6 shows the average NMI scores, variances of extracted clusters, and basis clusterings. The proposed method outperformed the basic clusterings by a significant margin. The benefit of the proposed method over the base clusterings

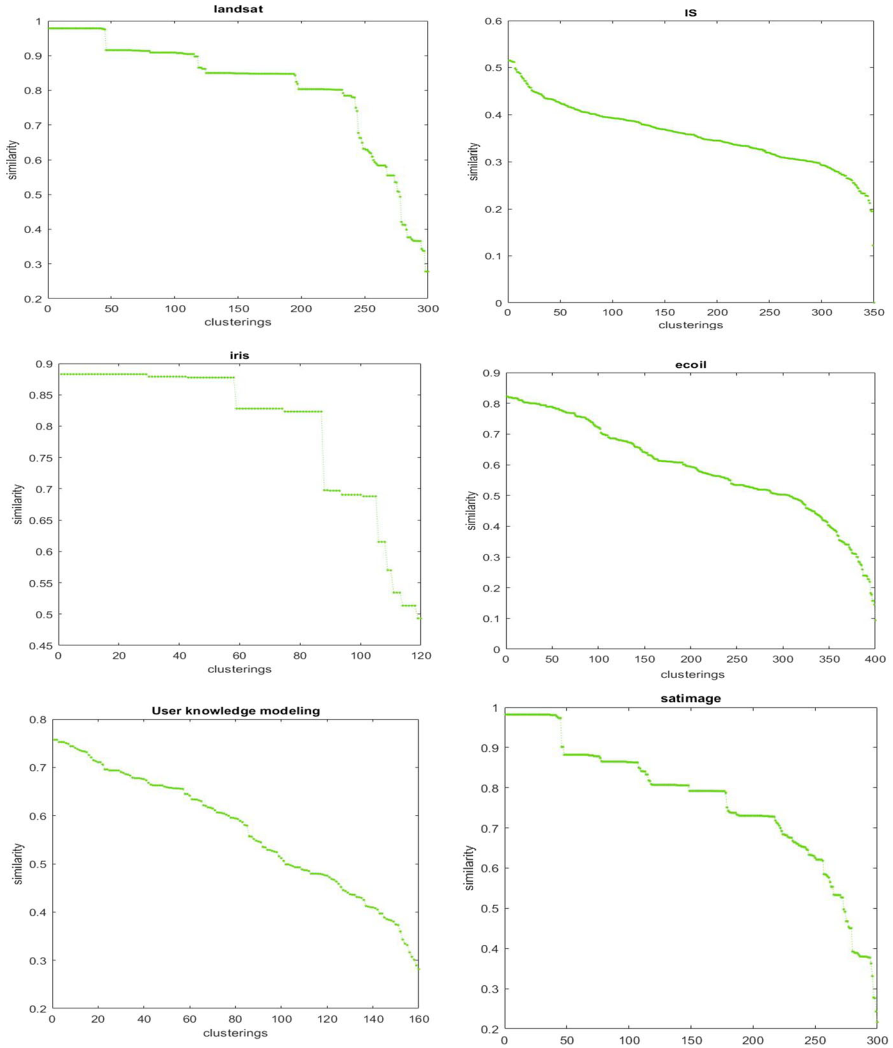
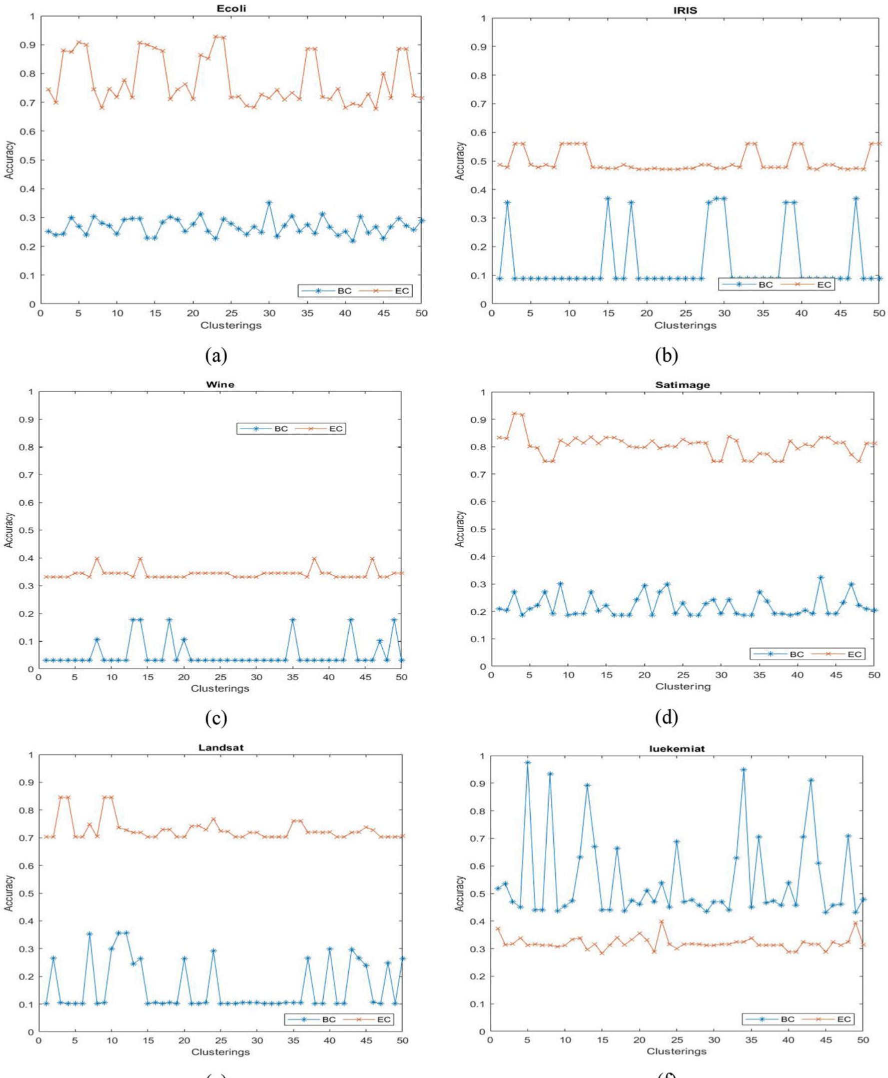


Fig. 5 Quality measure for varying parameters θ

was much greater for the Ecoli, IS, Iris, Landsat, luekemia, Satimage, Splice, user knowledge modeling, Wine, and Yeast datasets.

5.4 Comparison with other ensemble clustering methods

This section compares the proposed MCEA and CEPA methods with nine ensemble clustering methods, namely, CSPA, HGPA, MCLA (Topchy et al. 2004), hybrid bipartite graph formulation (HBGF) (Li et al. 2019), weighted evidence accumulation clustering (WEAC) (Minaei et al. 2014), K-means-based consensus clustering (KCC)



(a)

(b)

(c)

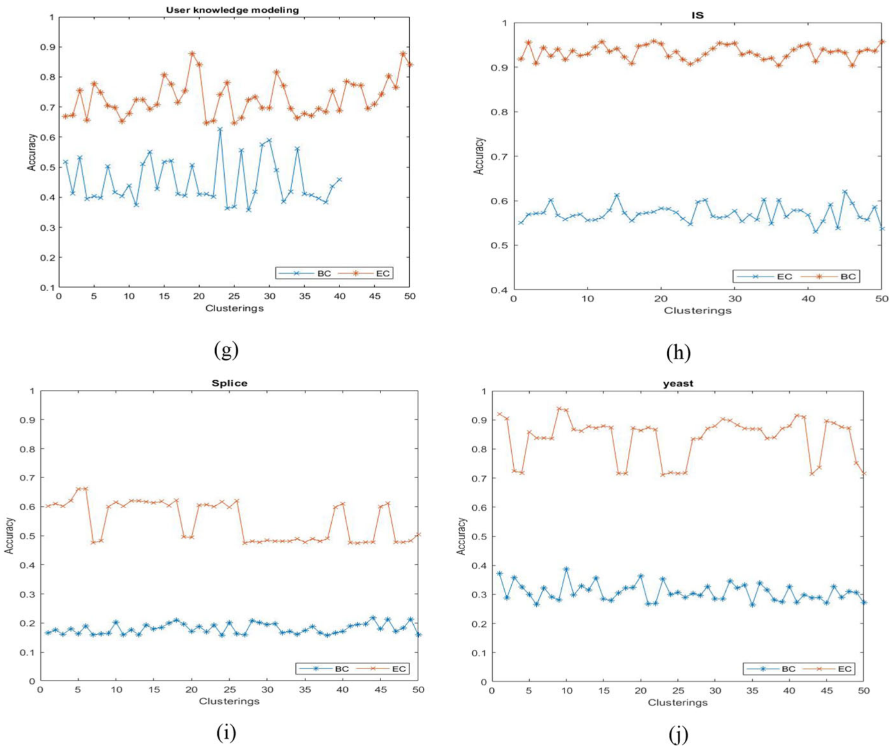
(d)

(e)

(f)

Fig. 6 Comparing the diversity of BC and EC

(Naldi et al. 2013), spectral ensemble clustering (SEC) (Lourenco 2013), Locally Weighted Evidence Accumulation (LWGP), and Locally Weighted Graph Partitioning (LWEA) (Huang et al. 2018). The true-*k* criterion is used to determine the number of clusters for the consensus clustering for each of the suggested and baseline methods. For the true-k, each approach was based on the number of classes in the dataset. Each of the suggested approaches and the baseline methods were run 100 times using the ensembles randomly constructed from the base clustering pool to ensure a fair comparison. Table 8 shows the average performance and standard deviations of dif-



(g)

(h)

(i)

(j)

Fig. 6 continued

ferent methods across 100 runs. As shown in the table, our proposed method (MCEA) obtained the best data in the Ecoli, Iris, Leukemia, Satimage, Splice, Wine, and Yeast datasets. However, CEPA obtained the best data in the Ecoli, Iris, Landsat, Leukemia, User knowledge Modeling, Wine, Yeast, and IS data sets.

Table 8 Over the course of 20 runs, several approaches with varying ensemble sizes of M were used to compute the average performance (NMI)

Method	Ecoli	Iris	Landsat	Leukemia	Satimage
CSPA	0.5157 ± 0.013	0.6523 ± 0.012	0.4543 ± 0.017	0.2131 ± 0.012	0.4513 ± 0.014
MCLA	0.5264 ± 0.012	0.7419 ± 0.014	0.4984 ± 0.012	0.2177 ± 0.015	0.6117 ± 0.012
HBGF	0.5287 ± 0.013	0.6205 ± 0.011	0.4536 ± 0.021	0.2245 ± 0.013	0.4527 ± 0.015
WEAC	0.5301 ± 0.012	0.7434 ± 0.009	0.4605 ± 0.009	0.2338 ± 0.012	0.4675 ± 0.011
KCC	0.5254 ± 0.011	0.5675 ± 0.013	0.4465 ± 0.018	0.2182 ± 0.011	0.4511 ± 0.008
SEC	0.5232 ± 0.014	0.6256 ± 0.012	0.4538 ± 0.013	0.2262 ± 0.016	0.4518 ± 0.016
LWGP	0.5305 ± 0.011	0.577 ± 0.011	0.4752 ± 0.012	0.2092 ± 0.013	0.4542 ± 0.012
LWEA	0.5254 ± 0.012	0.6162 ± 0.014	**0.4871 ± 0.013**	0.1942 ± 0.013	0.5455 ± 0.011
WHAC	0.5491 ± 0.014	0.7324 ± 0.010	0.4693 ± 0.015	0.2234 ± 0.012	**0.5554 ± 0.012**
MCEA	**0.5615 ± 0.014**	**0.7582 ± 0.012**	0.4796 ± 0.012	**0.3101 ± 0.014**	**0.6122 ± 0.011**
CEPA	**0.5478 ± 0.013**	**0.7338 ± 0.013**	**0.4912 ± 0.011**	**0.2342 ± 0.013**	0.4914 ± 0.014

Method	Splice	User knowledge modeling	Wine	Yeast	IS
CSPA	0.1878 ± 0.013	0.3447 ± 0.012	0.3952 ± 0.013	0.2109 ± 0.012	0.3073 ± 0.015
MCLA	0.2077 ± 0.012	0.3523 ± 0.016	0.4226 ± 0.011	0.2251 ± 0.011	0.3262 ± 0.013
HBGF	0.1885 ± 0.014	0.3615 ± 0.013	0.3876 ± 0.011	0.2012 ± 0.015	0.3214 ± 0.017
WEAC	**0.1932 ± 0.009**	0.3628 ± 0.015	0.3943 ± 0.012	0.2143 ± 0.013	0.3245 ± 0.012
KCC	0.1734 ± 0.011	0.3549 ± 0.011	0.3821 ± 0.009	0.2001 ± 0.016	0.3138 ± 0.013
SEC	0.1883 ± 0.014	0.3573 ± 0.023	0.3853 ± 0.008	0.2004 ± 0.012	0.3187 ± 0.022
LWGP	0.1764 ± 0.012	0.3661 ± 0.012	0.386 ± 0.012	0.2019 ± 0.011	0.3234 ± 0.011
LWEA	0.1853 ± 0.013	0.3832 ± 0.016	0.3861 ± 0.013	0.2241 ± 0.013	0.3256±0.018
WHAC	0.1878 ± 0.016	**0.4036 ± 0.016**	0.4087 ± 0.012	0.2352 ± 0.012	**0.3329 ± 0.024**
MCEA	**0.2119 ± 0.012**	0.3845 ± 0.015	**0.4228 ± 0.012**	**0.2567 ± 0.013**	0.3316 ± 0.016
CEPA	0.1892 ± 0.015	**0.3934 ± 0.014**	**0.4435 ± 0.011**	**0.2388 ± 0.012**	**0.3528 ± 0.018**

Bold face indicates best performance

5.5 Robustness to ensemble sizes

Additionally, in this study, different ensemble sizes, M, were used to compare the performance of the proposed approaches with that of the baseline methods. The proposed approaches and baseline methods were run 20 times on each benchmark dataset for each ensemble size M, with the ensemble of M base clusterings being randomly determined each time. The proposed methods had better performance on the Leukemia data set at size M = 30 compared with other sizes of M, but in other datasets, size M = 40 significantly exceeded the other sizes. compared to the basic methods, the proposed methods often obtained the most consistent and robust functions with size M = 40 in the datasets.

6 Conclusion

This paper proposed a new CEE clustering approach based on extracting new clusters from base clustering. In the proposed approach, two issues are considered: (1) Knowledge could be reused, meaning that there are different clusters in a library, which could be used to create new clustering; and (2) Traditional CE or CES needs a large library of clusters; however, if the library is small and the diversity is low, it will not have acceptable results. The proposed method is based on a set of clusters, not clustering. CEE can generate a large library of clustering on the cluster extraction. In this study, at the first step, the most effective clusters were identified. Then, the labels of the effective clusters (which were symbolic) were converted to binary, and new binaries were used to extract the new clusters. In this way, a large library of clusters was produced. At the next step, two new consensus functions, i.e., CEPA and MCEA, were created. Extensive experiments were conducted on a variety of real-world datasets. Experimental results (which were compared to those of advanced approaches) showed the superiority of the proposed methods regarding the clustering quality and efficiency.

A worthwhile future work could be focused on the use of different methods to extract new clusters with sizes different from those of the base clusters. For example, we propose the following five-step approach: (1) select a random subset from the base n clusters to create new clusters; (2) obtain the h_n consensus cluster solution using the consensus function; (3) find clusters that are of higher quality; (4) obtain the clustering composition from the previous step; and finally, (5) obtain group solutions with community function in the new clustering combination.

References

Akbari E et al (2015) Hierarchical cluster ensemble selection. Eng Appl Artif Intell 39:146–156

Alizadeh H, Minaei B, Parvin H (2014) Cluster ensemble selection based on a new cluster stability measure. Intell Data Anal 18:389–408

Alizadeh H, Yousefnezhad M, Minaei B (2015) Wisdom of crowds cluster ensemble. Intell Data Anal 19:485–503

Azimi J, Fern X (2009) Adaptive cluster ensemble selection. In: Proceedings of the 21st international joint conference on artificial intelligence. Morgan Kaufmann Publishers Inc., Pasadena, California, USA, pp 992–997

Bagherinia A et al (2020) Reliability-based fuzzy clustering ensemble. Fuzzy Sets Syst 413:1–28

Bai L et al (2019) An information-theoretical framework for cluster ensemble. IEEE Trans Knowl Data Eng 31:1464–1477

Banerjee A et al (2021) A new method for weighted ensemble clustering and coupled ensemble selection. Connect Sci 33:1–22

Elghazel H, Aussem A (2015) Unsupervised feature selection with ensemble learning. Mach Learn 98(1–2):157–180

Fern XZ, Brodley CE (2004, July) Solving cluster ensemble problems by bipartite graph partitioning. In: Proceedings of the twenty-first international conference on Machine learning, p 36

Fern XZ, Lin W (2008) Cluster ensemble selection. In: Proceedings of the 2008 SIAM international conference on data mining (SDM). Society for Industrial and Applied Mathematics, pp 787–797

Fozieh Asghari P, Saber N, Muhammad Y (2017) Wised semi-supervised cluster ensemble selection: a new framework for selecting and combing multiple partitions based on prior knowledge. J Adv Comput Res 8(1):67–88

Fred A, Jain A (2005) Combining Multiple Clusterings Using Evidence Accumulation. IEEE Trans Pattern Anal Mach Intell 27:835–850

Hadjitodorov ST, Kuncheva LI, Todorova LP (2006) Moderate diversity for better cluster ensembles. Information Fusion 7(3):264–275

Hamidi SS, Akbari E, Motameni H (2019) Consensus clustering algorithm based on the automatic partitioning similarity graph. Data Knowl Eng 124:101754

He Z, Xu X, Deng S (2005) A cluster ensemble method for clustering categorical data. Information Fusion 6(2):143–151

Huang D, Wang C-D, Lai J-H (2018) Locally weighted ensemble clustering. IEEE Trans Cybern 48:1460–1473

Iam-On N et al (2012) A Link-Based Cluster Ensemble Approach for Categorical Data Clustering. IEEE Trans Knowl Data Eng 24(3):413–425

Jia J et al (2011) Bagging-based spectral clustering ensemble selection. Pattern Recogn Lett 32(10):1456–1467

Jing L, Tian K, Huang J (2015) Stratified feature sampling method for ensemble clustering of high dimensional data. Pattern Recogn 48:3688–3702

Karypis G, Kumar V (1998) Multilevel k-way partitioning scheme for irregular graphs. J Parallel Distrib Comput 48(1):96–129

Kuhn HW (1955) The Hungarian method for the assignment problem. Naval Research Logistics Quarterly 2(1–2):83–97

Kuncheva L, Hadjitodorov S (2004) Using diversity in cluster ensembles, vol 2, pp 1214–1219

Li F et al (2018) Cluster's quality evaluation and selective clustering ensemble. ACM Trans Knowl Discov Data 12:1–27

Li F et al (2019) Clustering ensemble based on sample's stability. Artif Intell 273:37–55

Li T, Rezaeipanah A, Tag El Din EM (2022) An ensemble agglomerative hierarchical clustering algorithm based on clusters clustering technique and the novel similarity measurement. J King Saud Univ Comput Inf Sci 34(6, Part B):3828–3842

Lourenco A et al (2013) Probabilistic consensus clustering using evidence accumulation. Mach Learn 98:331–357

Ma T et al (2020) Multiple clustering and selecting algorithms with combining strategy for selective clustering ensemble. Soft Comput 24(20):15129–15141

Mahmoudi MR et al (2021) Consensus function based on cluster-wise two level clustering. Artif Intell Rev 54(1):639–665

Minaei B et al (2014) 2.02. Effects of resampling method and adaptation on clustering ensemble efficacy. Artif Intell Rev 41:27–48

Naldi M, Carvalho A, Campello RJGB (2013) Cluster ensemble selection based on relative validity indexes. Data Min Knowl Disc 27:259–289

Parvin H et al (2012) 2.03. A new classifier ensemble methodology based on subspace learning. J Exp Theor Artif Intell 25:1–27

Saidi M et al (2017) Instances selection algorithm by ensemble margin. J Exp Theor Artif Intell 30:1–22

Strehl A, Ghosh J (2002) Cluster Ensembles - A Knowledge Reuse Framework for Combining Multiple Partitions. J Mach Learn Res 3:583–617

Sulaiman NH, Mohamad D (2012) A Jaccard-based similarity measure for soft sets. In: 2012 IEEE symposium on humanities, science and engineering research

Tan P-N, Steinbach M, Kumar V (2016) Introduction to data mining. Pearson Education India

Topchy A, Jain AK, Punch W (2004) A mixture model for clustering ensembles. In: Proceedings of the 2004 SIAM international conference on data mining (SDM). Society for Industrial and Applied Mathematics, pp 379–390

Topchy A, Jain AK, Punch W (2005) Clustering ensembles: models of consensus and weak partitions. IEEE Trans Pattern Anal Mach Intell 27(12):1866–1881

Wang L et al (2022) Markov clustering ensemble. Knowl-Based Syst 251:109196

Yang Y, Chen K (2011) Temporal Data Clustering via Weighted Clustering Ensemble with Different Representations. Knowledge and Data Engineering, IEEE Transactions on 23:307–320

Yang F et al (2017) Cluster ensemble selection with constraints. Neurocomputing 235:59–70

Yousefnezhad M et al (2016) A new selection strategy for selective cluster ensemble based on Diversity and Independency. Eng Appl Artif Intell 56:260–272

Yu Z et al (2015) Adaptive Noise Immune Cluster Ensemble Using Affinity Propagation. IEEE Trans Knowl Data Eng 27(12):3176–3189

Zhao X, Cao F, Liang J (2018) A sequential ensemble clusterings generation algorithm for mixed data. Appl Math Comput 335:264–277

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.