Information Sciences 378 (2017) 389–409

Contents lists available at ScienceDirect

Information Sciences

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

Multigranulation information fusion: A Dempster-Shafer evidence theory-based clustering ensemble method☆

Feijiang Li[a], Yuhua Qian[a,b,*], Jieting Wang[a], Jiye Liang[b]

[a] School of Computer and Information Technology, Shanxi University, Taiyuan, 030006 Shanxi, China
[b] Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, Taiyuan, 030006 Shanxi, China

ARTICLE INFO

Article history:
Received 29 August 2015
Revised 21 September 2016
Accepted 1 October 2016
Available online 4 October 2016

Keywords:
Multigranulation
Information fusion
Clustering ensemble
Dempster-Shafer evidence theory

ABSTRACT

Clustering analysis is a fundamental technique in machine learning, which is also widely used in information granulation. Multiple clustering systems granulate a data set into multiple granular structures. Therefore, clustering ensemble can serve as an important branch of multigranulation information fusion. Many approaches have been proposed to solve the clustering ensemble problem. This paper focuses on the direct approaches which involve two steps: finding cluster correspondence and utilizing a fusion strategy to produce a final result. The existing direct approaches mainly discuss the process of finding cluster correspondence, while the fusing process is simply done by voting. In this paper, we mainly focus on the fusing process and propose a Dempster-Shafer evidence theory-based clustering ensemble algorithm. The advantage of the algorithm is that the information of an object's surrounding cluster structure is taken into consideration by using its neighbors to describe it. First, we find neighbors of each object and generate its label probability outputs in every base partition. Second, these label probability outputs are integrated based on DS theory. Theoretically, our method is superior to other voting methods. Besides, several experiments show that the proposed algorithm is statistically better than seven other clustering ensemble methods.

1. Introduction

Granular computing is efficient in designing intelligent systems [30]. As one of the three key issues (information granulation, organization and causation) in granular computing [11,41,43,44], information granulation is also regarded as one of the fundamental features of human cognitive ability. The task of information granulation is to granulate a data set into granules to obtain a granular structure. Many strategies of information granulation [13,24,28,29,42] have been proposed to meet different user demands, in which clustering analysis is a widely used strategy. Clustering analysis [40] is an interesting area in machine learning, whose task is to find the structure of data through dividing a data set into clusters. Good clustering often satisfies two requirements: one is that the objects share high similarity in the same cluster, the other is that the objects share high dissimilarity in different clusters. A clustering algorithm is only suitable for a particular data distribution and a

☆ This is an extended version of the paper presented at the International Conference of Machine Learning and Cybernetics (ICMLC) 2015, Guangzhou, China.

* Corresponding author.
E-mail addresses: feijiangli@email.sxu.edu.cn (F. Li), jinchengqyh@126.com (Y. Qian), jietingw@163.com (J. Wang), ljy@sxu.edu.cn (J. Liang).

http://dx.doi.org/10.1016/j.ins.2016.10.008
0020-0255/

particular data model. However, it is often hard to get data distribution and data model. Using a single clustering system to realize information granulation is not always satisfactory. Therefore, clustering ensemble comes into being. This evolution is similar to the development of multigranulation analysis [31–34]. Multigranulation analysis offers many new ideas for designing clustering algorithms [23,36,48]. In turn, multiple clustering systems explore a data set from different viewpoints, so that the data set can be granulated into multiple granular structures. Thus, integrating multiple clustering systems, the so-called clustering ensemble, is an important branch of multigranulation information fusion.

Clustering ensemble, which integrates multiple exploratory clustering results, has shown better performance than a single clustering system. Different from classifier ensemble, there is no training label in the clustering ensemble. The labels assigned by a clustering result are just symbols without practical meaning, so the design of a clustering ensemble method is much more difficult than that of a classifier ensemble method. Even so, many approaches [3–5,7–10,12,14,17–19,26,37,38,45–47] have been proposed to handle clustering ensemble problem. Based on different information matrix the consensus function used, the clustering ensemble algorithms can be summarized into four types: feature-based methods [26,38], co-association-based methods [9,12,14,37,45], graph-based methods [4,7,17–19,37] and direct methods [3,8,10,37,47]. The feature-based methods treat the clustering ensemble problem as clustering for categorical data. The co-association-based methods utilize the matrix of co-association frequency. The graph-based methods first build a graph, in which nodes are objects or clusters and edges are their relationship, and then utilize a graph partition technique to generate the final ensemble result. As for the direct methods, there are two steps: finding cluster correspondence and then utilizing a fusion strategy to generate the final ensemble result.

The direct methods are the most immediate and simplest approach, which are based on the major characteristic of the labels of base partitions' being merely symbols without any implication. Therefore, the direct methods mainly focus on finding the cluster correspondence. They firstly re-label the base partitions to unify the system of labels, and then employ a fusion strategy to generate the final partition. In fact, the fusion strategy has a direct effect on the final clustering result, so it is as important as the correspondence-finding step in direct method. However, most direct methods oversimplify this step by using voting. In other words, the reason for the poor performance of direct method may be the neglect of the fusion strategy. If so, there is a lot of room to improve the performance of the direct method, which inspires our work here.

Clustering analysis aims to place an object and its neighbors into the same cluster. In clustering ensemble, each base clustering result attempts to express the underlying distribution of data. However, the voting strategy does not take this essential relationship into account. If we introduce the neighbor information into each object, its surrounding underlying distribution will be taken into consideration. We denote each object in the form of label probabilities rather than a certain label. Thus, the integrated probabilities is a kind of ensemble with underlying structure. Since the label probability outputs are uncertain measures for each object, we introduce the Dempster-Shafer evidence theory to integrate these multiple uncertain labels.

Dempster-Shafer (DS) theory, also known as evidence theory, was firstly proposed by Dempster in 1967 in his book *A Mathematical Theory of Evidence* [35]. As an uncertainty reasoning method, DS theory has been widely used in expert system to handle risk assessment, information reliability evaluation, and uncertainty management. In addition, DS theory has provided a simple method, the Dempster's combination rule, to combine multiple evidences. Hence, its application is extended to the information fusion area [2,6,22,25,39].

The reasons for recommending DS theory lie in its simplicity and suitability. As for the simplicity, the Dempster's combination rule in DS theory provides a natural way to combine the label probability outputs. The only needed work is to construct the label probabilities, which are known as evidence or believe mass functions in DS theory. Because the essential operation of Dempster's combination rule is multiplication, a high combined believe mass will be given to the cluster which get high believe masses from all experts. It intuitively meets the demands of clustering ensemble, so DS theory is suitable.

In this study, we expect that the combination of neighbor information and DS theory can realize ensemble with underlying structure. To address this issue, one may consider the following four problems.

(1) How to use neighbor information to denote the objects in each base clustering and how to find the neighbors of each object in clustering ensemble?
(2) How to utilize DS theory to combine base partitions?
(3) Will the DS theory-based clustering ensemble algorithm get a better performance?
(4) In direct methods, is the fusion strategy important?

This paper is motivated by the above four problems. In this paper, we utilize a gray level image binaryzation technique to find the neighbors of each object without parameter., we first explore the label distribution around a measured object with its neighbor information, and then give its probabilities of belonging to every label. Then, the expression form of the base partitions is changed into probabilities. To combine the base partitions in the form of probability, a DS theory-based fusion strategy is proposed. This strategy takes the relationship between an object and its neighbors into consideration, so it may be more suitable as a fusion strategy for a clustering ensemble method. Utilizing DS-based fusion strategy, we propose a Dempster-Shafer evidence theory-based clustering ensemble algorithm (using DSCE for short) and show its performance both in theory and experiments. Comparing DSCE with the existing direct methods and some other type methods, we can conclude that the fusion strategy in direct methods is very important, that is, a suitable fusion strategy also can improve the performance of a direct method.

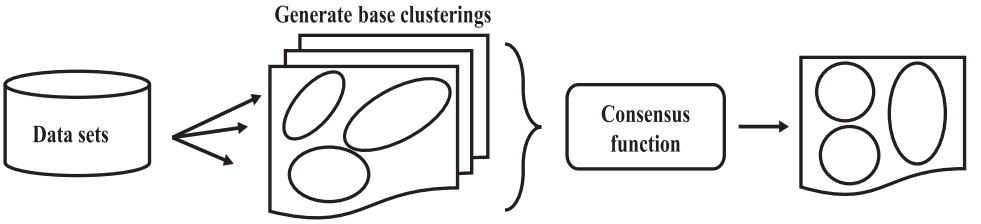


Fig. 1. The framework of clustering ensemble.

Table 1
Clustering ensemble process.

$$\begin{array}{c|cccc} X & A_1 & A_2 & \cdots & A_d \\ \hline x_1 & x_{11} & x_{12} & \cdots & x_{1d} \\ x_2 & x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_n & x_{n1} & x_{n2} & \cdots & x_{nd} \end{array} \xrightarrow{clustering} \begin{array}{c|cccc} \Pi & \pi^1 & \pi^2 & \cdots & \pi^h \\ \hline x_1 & \pi^1_{x_1} & \pi^2_{x_1} & \cdots & \pi^h_{x_1} \\ x_2 & \pi^1_{x_2} & \pi^2_{x_2} & \cdots & \pi^h_{x_2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_n & \pi^1_{x_n} & \pi^2_{x_n} & \cdots & \pi^h_{x_n} \end{array} \xrightarrow{F(\bullet)} \begin{array}{c|c} & \pi \\ \hline x_1 & \pi_{x_1} \\ x_2 & \pi_{x_2} \\ \vdots & \vdots \\ x_n & \pi_{x_n} \end{array}$$

The rest of the paper is organized as follows. In Section 2, we briefly describe the clustering ensemble problem, the notations used in this paper, and a direct method which includes two methods of finding correspondence for different situations and voting fusion strategy. In Section 3, we discuss on how to find neighbors and the DS-based fusion strategy. Then, DSCE and voting-based direct method is compared from theoretical perspective. The experimental analysis is reported in Section 4. Finally, we summarize this paper and list the future work in Section 5.

2. Related works

Clustering ensemble aims to integrate multiple clustering results into a unified result, which is superior to each base clustering result. As is shown in Fig. 1, there are two major tasks in clustering ensemble: generating base clusterings and designing consensus function. In this section, we list the notations used in the paper, introduce the method of generating base clusterings, and describe the two phases of the direct methods: finding cluster correspondence and fusing clusterings.

2.1. Notations

Generally, notations in the clustering ensemble problem can be defined as follows. Let $\mathcal{X} = \{x_1, x_2, \cdots, x_n\} \in R^d$ denote original data sets with n objects and each object is represented by d attributes. Every clustering result is denoted as a label vector $\pi \in \mathbb{N}^n$. The input of a clustering ensemble algorithm is a set of h base clustering results $\Pi = \{\pi^1, \pi^2, \cdots, \pi^h\}$. The number of meta-clusters in qth clustering can be denoted by $K^{(q)}$, then the qth clustering is expressed as $\pi^q = \{c_1, c_2, \cdots, c_{K^{(q)}}\}$. The label of ith object partitioned by jth algorithm is denoted by $\pi^j_{x_i}$. $\pi = F(\{\pi^1, \pi^2, \cdots, \pi^h\})$ indicates a clustering ensembles problem which integrates h base partitions by $F(\bullet)$. $F(\bullet)$ is the consensus function employed in the clustering ensembles method. Then, clustering ensemble process is formalized in Table 1.

2.2. Base weak partitions

It has been proved that clustering ensemble is effective when base partitions have high diversity. Base partitions which are generated by different clustering algorithms or different parameters of an algorithm are time-consuming and similar. Unfortunately, these two characteristics go against the demands of the base partitions in clustering ensemble. There is no strong requirement for the performance of base partitions, so weak partitions are more suitable. Although a weak partition is slightly better than random partition, it is diverse and economical. Alexander [38] proposed a method called random one-dimensional projection to generate different weak clustering results. This method firstly projects a d-dimensional data set into one dimension, then generates a partition by k-means clustering algorithm [15]. The steps of random 1D projection are shown in Algorithm 1:

Algorithm 1 Random 1D projection.

1: generate a random d-dimensional vector u, s.t $|u| = 1$
2: $X' = X_{n\times d} \cdot u_{d\times 1}$
3: $\pi^q \leftarrow$ k-means (X')

The characteristic of random 1D projection is that it is sensitive to both projection vector and initial centers in k-means, which brings infinite settings of parameters. Run the algorithm several times, we will obtain a group of diverse base clusterings. In this paper, we apply this algorithm to generate base partitions.

2.3. The direct methods

2.3.1. Finding cluster correspondence

Clustering analysis is a kind of unsupervised learning technology, so the correspondence between every meta-cluster of different clustering results is unknown. Suppose that there are two partitions which are expressed as {1, 1, 2, 2, 3, 3} and {2, 2, 3, 3, 1, 1}. It is obvious that the two partitions indicate the same result but they belong to different label systems. Naturally, a solution to solve the clustering ensemble problem contains two steps: finding the correspondence between every meta-cluster in different clustering systems and generating a final integrated partition through a fusion strategy.

For different types of base partitions, the methods of finding cluster correspondence are different. Here, we consider the following two types of base partitions: one is that the number of clusters in each base partition is equal to the number of clusters in the target partition and the other is that at least one base partition has different number of clusters with the target result. For the first type, Zhou [47] introduced the align process to find cluster correspondence based on the assumption that similar clusters should share maximum overlapped objects. The align process selects a clustering randomly as the baseline for the remaining clusterings to match with. For each pair of clusterings, this approach builds a $K \times K$ overlap matrix, in which the clusters with the largest number of overlapped objects are re-labeled in the same way. When there are two clusterings π_1 and π_2, the align process is shown in Algorithm 2. Correspondence between the clusters in all clustering systems can be found by repeating this process. For the other type of base partitions, the method of finding cluster correspondence in Meta-Clustering Algorithm (MCLA) [37] is efficient. It is based on clustering clusters to find cluster correspondence. In MCLA, each cluster is treated as a hyperedge, and the weight is the similarity between each pair of clusters, then an undirected meta-graph can be built. The cluster correspondence is found by partitioning the meta-graph through a graph partitioning technique called METIS [20]. Suppose a base partitions set is $\Pi = \{\pi^1, \pi^2, \cdots, \pi^h\}$, the method of finding cluster correspondence in MCLA is shown in Algorithm 3. We do not focus on the process of finding cluster correspondence, so these two methods are directly employed in the proposed algorithm in Section 3.

Algorithm 2 Align process.

for $i = 1$ to K **do**
  for $j = 1$ to K **do**
    $OVERLAP(i, j) = |\{x|\pi_x^1 = i, \pi_x^2 = j\}|$;
  end for
end for
for $k = 1$ to K **do**
  $(u(k), v(k)) = location(max(OVERLAP))$
  $OVERLAP_{u(k)*} = -1$
  $OVERLAP_{*v(k)} = -1$
end for
re-label π^1 and π^2 based on u, v

Algorithm 3 Finding correspondence.

$C = \sum_{q=1}^{h} K^{(q)}$
for $i = 1$ to C **do**
  for $j = 1$ to C **do**
    $w(i, j) = \frac{c_i^{\mathrm{T}} c_j}{\|c_i\|_2^2 + \|c_j\|_2^2 - c_i^{\mathrm{T}} c_j}$
  end for
end for
$W = METIS(w)$
re-label Π based on W

2.3.2. Fusion strategy

After finding the correspondence of clusters between different partitions, the clustering ensemble problem can be transformed into a qualitative information fusion problem. The most popular fusion strategy used in clustering ensemble is the voting strategy, in which plurality voting, weighted voting and soft voting are three widely used techniques. For an object, plurality voting chooses the label which has the most votes as the final output. When two labels have equal votes, either

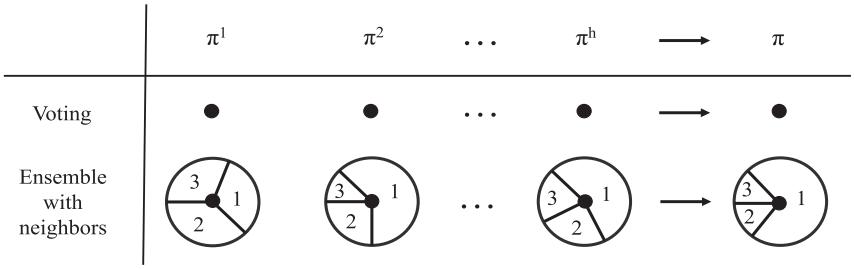


Fig. 2. Voting and ensemble with neighbors (• represents an object).

of them can be treated as the final result. The ensemble result obtained by plurality voting can be deemed as the partition which shares the most information with the base clusterings. Weighted voting considers that the quality of each base clustering is different, so the contribution of each base clustering to the final output would not be equal. High confidence believe should be given to high quality clustering result in weighted voting. Without ground truth, measuring the quality of a base clustering result is challenging. Most methods use stability, i.e., the average similarity between one clustering result with others, to substitute the measurement of quality in clustering ensemble [21]. Soft voting is applied when the output of an object in each base partition is label probabilities. The integrated output of soft voting is the average label probabilities of the base partitions. In the methods MCLA [37], Constr.Combin [3] and Unconstr.Combin [3], the combined output is formalized as the probabilities of an object's belonging to each cluster, and the final label is assigned to the cluster which has the highest probability.

The common practice in the second step of a direct method is simply utilizing an universal qualitative fusion strategy. It is interesting to design a particular clustering ensemble fusion strategy which takes the characteristics of clustering analysis into consideration.

3. Dempster-Shafer evidence theory-based clustering ensemble algorithm (DSCE)

Our approach contains the following four steps: (1) finding neighbors of each object, (2) finding cluster correspondence, (3) defining mass functions of each base partition and (4) applying Dempster's combination rule. How to find neighbors of each object will be described in Section 3.1. The second step in our algorithm is corresponding to the first step of the direct methods. Here, we use the existing methods which are described in Section 2.3.1. The last two steps correspond to the second step of the direct methods together, and they constitute a novel fusion strategy which is called DS-based fusion strategy. In Section 3.2, DS-based fusion strategy will be introduced in detail. In Section 3.3, the framework of DSCE will be given. Finally, we theoretically analyze the ensemble performance of DSCE in Section 3.4.

3.1. Finding neighbors

In clustering analysis, according to the demand that similar objects should be placed in the same cluster, an object and its neighbors often have same label. Then, using the labels of neighbors to estimate the label distribution around a measured object is an efficient approach. The label distribution can offer the confidence level of a measured object's degree of belonging to each label. The integration of label probabilities involves the underlying structure of a data set, which is very different from voting (see Fig. 2). So it may be more suitable to handle the clustering ensemble problem. To estimate the label distribution, neighbors of each object should be found first.

The distance between two objects is difficult to measure without the feature information, which makes finding neighbors a difficult problem in clustering ensemble. As a compromise, similarity between two objects can be represented by the frequency of their being assigned to the same cluster. Concretely, the similarity between two objects x_i and x_j is defined as:

$$p_{ij} = \frac{1}{h}\sum_{q=1}^{h} \mathbb{I}\left(\pi_{x_i}^q, \pi_{x_j}^q\right), \tag{1}$$

where

$$\mathbb{I}\left(\pi_{x_i}^q, \pi_{x_j}^q\right) = \begin{cases} 1, & \pi_{x_i}^q = \pi_{x_j}^q; \\ 0, & \pi_{x_i}^q \neq \pi_{x_j}^q. \end{cases}$$

Computing the similarity between each pair of objects forms a pairwise similarity matrix $S_{n \times n}$, in which each element can reflect an object's probability of being neighbor with another object. Intuitively, the higher the similarity between two objects is, the stronger their tendency of being neighbor is. To explicitly define the neighbor, a threshold should be established to segment this pairwise similarity matrix into a two-value matrix so that the values which are greater than the

Table 2
Finding neighbors.

$$\begin{array}{c|cccc} \Pi & \pi^1 & \pi^2 & \cdots & \pi^h \\ \hline x_1 & \pi^1_{x_1} & \pi^2_{x_1} & \cdots & \pi^h_{x_1} \\ x_2 & \pi^1_{x_2} & \pi^2_{x_2} & \cdots & \pi^h_{x_2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_n & \pi^1_{x_n} & \pi^2_{x_n} & \cdots & \pi^h_{x_n} \end{array} \xrightarrow{similarity} \begin{array}{c|cccc} S & x_1 & x_2 & \cdots & x_n \\ \hline x_1 & p_{11} & p_{12} & \cdots & p_{1n} \\ x_2 & p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_n & p_{n1} & p_{n2} & \cdots & p_{nn} \end{array} \xrightarrow{Otsu} \begin{array}{c|cccc} S' & x_1 & x_2 & \cdots & x_n \\ \hline x_1 & p'_{11} & p'_{12} & \cdots & p'_{1n} \\ x_2 & p'_{21} & p'_{22} & \cdots & p'_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_n & p'_{n1} & p'_{n2} & \cdots & p'_{nn} \end{array}$$

threshold will be defined as neighbor and marked as 1, while the remaining elements will be marked as 0. To establish the threshold, Otsu's algorithm [27], a widely used technique in selecting a threshold from a gray level image, is taken into consideration. It is shown as follows.

Let matrix $M_{n\times n}$ indicate a gray level image with L gray levels $[1, 2, \cdots, L]$, in which the ith level has n_i pixels correspondingly. Then, the total number of pixels equals to the sum of the number of every pixel, which can be expressed as $n^2 = n_1 + n_2 + \cdots + n_L$. The probability distribution of the gray-level histogram can be normalized as:

$$p_i = \frac{n_i}{n^2}, \quad p_i > 0, \ \sum_{i=1}^{L} p_i = 1. \tag{2}$$

Suppose that the pixels are separated into two classes by a threshold t, then the probability of pixels' being below t level, the mean level of below threshold and the total mean level, can be calculated by:

$$\omega(t) = \sum_{i=1}^{t} p_i, \quad \mu(t) = \sum_{i=1}^{t} i p_i, \quad \mu_L = \sum_{i=1}^{L} i p_i. \tag{3}$$

After searching all levels in the original gray level image, an optimal threshold t^* can be achieved by:

$$t^* = \arg\max_{1\le t\le L} \left(\frac{(\mu_L \cdot \omega(t) - \mu(t))^2}{\omega(t)\cdot(1-\omega(t))} \right). \tag{4}$$

Otsu's algorithm selects an optimal threshold automatically and stably with simple procedure, and it is widely applied not only in gray level images but also in histograms.

Matrix S has information of histogram and can be treated as a gray level image, so Otsu's algorithm is suitable to select an optimal threshold from S. With a threshold t^* selected by Otsu's algorithm, one can obtain a new matrix S' from similarity matrix S through:

$$p'_{ij} = \begin{cases} 1, & p_{ij} > t^*; \\ 0, & p_{ij} \le t^*. \end{cases} \tag{5}$$

The steps of finding neighbors based on a set of partitions Π is shown in Table 2. In matrix S', x_j is a neighbor of x_i when $p'_{ij} = 1$, so matrix S' indicates the neighbors of each object clearly.

3.2. DS-based fusion strategy

Having obtained the neighbors of each object, it is easy to estimate the label distribution around each object, which transforms the traditional absolute label into label probabilities. The label probability output is uncertain measures of an object. As an effective and widely used uncertainty reasoning method, Dempster-Shafer evidence theory is employed as a fusion strategy to handle the uncertain ensemble problem.

In Dempster-Shafer evidence theory, there are two important functions: the believe mass function and the Dempster's combination rule. How to define mass functions and how to use Dempster's combination rule will be introduced in this section. These two steps are processed after finding cluster correspondence, thus, the mass functions generated from every base partition will share an unified standard and the Dempster's combination rule will combine corresponding mass functions.

3.2.1. Defining mass function

As an uncertainty reasoning method, DS theory gives degree of belief by mass function at first. Mass function is confidence evaluation of an expert to a problem. The definition of mass function is described as follows.

Definition 1 (Mass Function)**.** Let Ⓗ be an proposition space and $\mathscr{P}(Ⓗ)$ is all subsets of Ⓗ. Mass function m is a projection $\mathscr{P}(Ⓗ) \to [0, 1]$ which satisfies

(1) $m(\emptyset) = 0$,

(2) $\sum_{A \subseteq \mathscr{P}(Ⓗ)} m(A) = 1$.

In a mass function, $m(A)$ indicates the confidence evaluation of an expert. In the clustering ensemble problem, each base clustering is treated as an expert's evaluation and A is a subset of labels.

Every clustering result gives each object an absolute label. It is arbitrary to trust the obtained label of a measured object. In general, the prediction of an expert to a problem is uncertain. Through the labels of an object's neighbors, one can compute the object's possibility within each cluster. What calls for special attention is the situation that the belief mass of a label is zero, which will happen when the label does not exist in neighbors of an object. A problem arising from the situation is that the belief mass of this label will always be zero when we use Dempster's combination rule to combine with other partitions in the next step. That is to say, the label of an object may be determined by one clustering result, and if the prediction of the object in this clustering result happens to be wrong, there is no chance to correct the final result. To avoid this, we divide the measured object into K objects which belong to different labels to guarantee that the neighbors of the object contain all labels. Then the neighbors of an object will include the real neighbors and the fictitious divided objects but without the measured object itself. For object x_i, the K mass functions based on partition π^q are defined as:

$$m(\pi^q_{x_i}) \begin{cases} m\left(\pi^q_{x_i}, 1\right) = \frac{|X^q_{i1}|+1}{\sum_{j=1}^{n} p'_{ij}+K-1}, \\ m\left(\pi^q_{x_i}, 2\right) = \frac{|X^q_{i2}|+1}{\sum_{j=1}^{n} p'_{ij}+K-1}, \\ \vdots \\ m\left(\pi^q_{x_i}, K\right) = \frac{|X^q_{iK}|+1}{\sum_{j=1}^{n} p'_{ij}+K-1}, \end{cases} \tag{6}$$

where $X^q_{ik} = \{x_j | p'_{ij} = 1,\ \pi^q_{x_j} = k,\ j = 1, 2, \cdots, n,\ j \neq i\}$, and $|a|$ is the cardinality of set a. In the fraction, the denominator indicates the numbers of an object's neighbors and the numerator counts the number of the object's neighbors with label k. Obviously, the object in a partition π^q has K mass functions whose sum is equal to 1, so this construction method satisfies the definition of mass function.

3.2.2. Applying Dempster's combination rule

Undoubtedly, different experts will provide different mass functions. Dempster's combination rule is used to combine evaluations of different experts in DS theory. Dempster's combination rule is defined as:

Theorem 1 (Dempster's combination rule)**.** *Suppose m_1 and m_2 are two mass functions of Ⓗ, m is a mass function combining m_1 and m_2, which satisfies*

(1) $m(\emptyset) = 0$,

(2) $m(A) = \frac{1}{N_{12}} \sum_{E \bigcap F = A} m_1(E) m_2(F), \qquad A \neq \emptyset \ \text{ and } \ A, E, F \subseteq \mathscr{P}(Ⓗ)$;

where $N_{12} = \sum_{E \bigcap F \neq \emptyset} m_1(E) m_2(F)$.

Theorem 1 gives a frame to combine evaluations of two experts. The consequence is still mass functions which can be treated as an evaluation of an expert as well. Thus, when there are more than two experts, one can use Dempster's combination rule repeatedly to generate a final judgement. In clustering ensemble, for object x_i, using Dempster's combination rule to combine two mass functions which are generated by partitions π^p and π^q can be formulated as:

$$m(\pi^p_{x_i}) \oplus m(\pi^q_{x_i}) = m(\pi^{pq}_{x_i}) \begin{cases} m\left(\pi^{pq}_{x_i}, 1\right) = \left(m(\pi^p_{x_i}, 1) m(\pi^q_{x_i}, 1)\right)/N_{pq}, \\ m\left(\pi^{pq}_{x_i}, 2\right) = \left(m(\pi^p_{x_i}, 2) m(\pi^q_{x_i}, 2)\right)/N_{pq}, \\ \vdots \\ m\left(\pi^{pq}_{x_i}, K\right) = \left(m(\pi^p_{x_i}, K) m(\pi^q_{x_i}, K)\right)/N_{pq}, \end{cases} \tag{7}$$

where $N_{pq} = \sum_{k=1}^{K} m(\pi^p_{x_i}, k) m(\pi^q_{x_i}, k)$.

For an object, one can give a group of mass functions based on all partitions, and then use Dempster's combination rule $h-1$ times (as Formula (8)) to obtain integrated mass functions, in which the cluster with the maximum value of belief mass is the cluster that the object should be assigned to.

$$m(\pi_{x_i}) = \left(\left(\left(m(\pi^1_{x_i}) \oplus m(\pi^2_{x_i})\right) \oplus m(\pi^3_{x_i})\right) \oplus \cdots m(\pi^{h-1}_{x_i})\right) \oplus m(\pi^h_{x_i}). \tag{8}$$

Table 3
Mass functions of all partitions.

$m(\pi_x^1)$	$m(\pi_x^2)$	$\cdots$	$m(\pi_x^h)$
$m(\pi_x^1, 1)$	$m(\pi_x^2, 1)$	$\cdots$	$m(\pi_x^h, 1)$
$m(\pi_x^1, 2)$	$m(\pi_x^2, 2)$	$\cdots$	$m(\pi_x^h, 2)$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$m(\pi_x^1, K)$	$m(\pi_x^2, K)$	$\cdots$	$m(\pi_x^h, K)$

3.3. The framework of the algorithm

The frame of Dempster-Shafer theory-based clustering ensembles algorithm is shown in Algorithm 4.

Algorithm 4 Dempster-Shafer theory-based clustering ensembles algorithm (DSCE).

INPUT: cluster results set Π, number of clusters K
OUTPUT: final label vector π
STEP.1 Find neighbors
call for process in Table 2
STEP.2 Find cluster correspondence
call for process in Section 2.3
STEP.3 Create mass functions
for $q = 1$ to h **do**
  for $i = 1$ to n **do**
    create mass functions use Formula (6)
  end for
end for
STEP.4 Apply Dempster's combination rule
for $i = 1$ to n **do**
  $m(\pi_{x_i}) = m(\pi_{x_i}^1)$
  for $q = 2$ to h **do**
    $m(\pi_{x_i}) = m(\pi_{x_i}) \oplus m(\pi_{x_i}^q)$
  end for
end for
STEP.5 Label
for $i = 1$ to n **do**
  $\pi(x_i) = \arg\max_{k=1,2,\cdots K} m(\pi_{x_i})$
end for

Finding cluster correspondence process in Step 2 can also be operated at beginning of the algorithm when the base partitions have a fixed K which equals to the true class number. In this situation, process of finding neighbors will generate the same result no matter which step goes first.

3.4. Analysis of DSCE

We conduct the analysis after the mass functions of object x in each base partition have been obtained as Table 3. Then, the integration of the h mass functions can be calculated as Formula (9):

$$m(\pi_x)\begin{cases} m(\pi_x, 1) = \prod_{p=1}^{h} m(\pi_x^p, 1)/N_\Pi, \\ m(\pi_x, 2) = \prod_{p=1}^{h} m(\pi_x^p, 2)/N_\Pi, \\ \qquad\vdots \\ m(\pi_x, K) = \prod_{p=1}^{h} m(\pi_x^p, K)/N_\Pi, \end{cases} \tag{9}$$

where $N_\Pi = \sum_{k=1}^{K} \prod_{p=1}^{h} m(\pi_x^p, k)$.

Without loss of generality, we assume that $\Pi_h = \{\pi^1, \pi^2, \ldots, \pi^h\}$ is a finite set with h predictions. These predictions are correct with probability $\{g_{1k}, g_{2k}, \ldots g_{hk}\}$ for object x in cluster k. For the other $(K-1)$ clusters in prediction p, the probabilities of incorrectness are equivalent, which is $(1-g_{pk})/(K-1)$. It can be deemed that the objects around x have the same prediction probability as x. Then, for x in prediction $\pi^p \in \Pi_h$, the mass functions which are defined by neighbors can be calculated as:

$$m\left(\pi_x^p\right)\begin{cases} m\left(\pi_x^p, 1\right) = (1-g_{pk})/(K-1), \\ \qquad\vdots \\ m\left(\pi_x^p, k\right) = g_{pk}, \\ \qquad\vdots \\ m\left(\pi_x^p, K\right) = (1-g_{pk})/(K-1), \end{cases} \tag{10}$$

Based on Formula (9), the combined mass functions is:

$$m(\pi_x^{\Pi_h})\begin{cases} m(\pi_x^{\Pi_h},1)=\prod_{p=1}^{h}(\frac{1-g_{pk}}{K-1})/N_{\Pi_h}, \\ \vdots \\ m(\pi_x^{\Pi_h},k)=\prod_{p=1}^{h}(g_{pk})/N_{\Pi_h}, \\ \vdots \\ m(\pi_x^{\Pi_h},K)=\prod_{p=1}^{h}(\frac{1-g_{pk}}{K-1})/N_{\Pi_h}, \end{cases} \tag{11}$$

where $N_{\Pi_h}=\frac{\prod_{p=1}^{h}(1-g_{pk})}{(K-1)^{h-1}}+\prod_{p=1}^{h}g_{pk}$.

We define g_k as the accuracy of the worst prediction, i.e., $g_k=\min\{g_{1k},g_{2k},\ldots g_{hk}\}$. For the convenience of calculation, we weaken the accuracy of every prediction in Π_h to the worst situation. It can be deemed as constructing a weak prediction set Π'_h based on Π_h. Specifically, the predictions in Π'_h are correct with probability $\{g'_{1k},g'_{2k},\ldots g'_{hk}\}$, in which $g'_{1k}=g'_{2k}=\ldots=g'_{hk}=g_k$. Thus, the conservative combined mass function can be calculated as:

$$m\left(\pi_x^{\Pi'_h}\right)\begin{cases} m(\pi_x^{\Pi'_h},1)=((1-g_k)/(K-1))^h/N_{\Pi'_h}, \\ \vdots \\ m(\pi_x^{\Pi'_h},k)=g_k^h/N_{\Pi'_h}, \\ \vdots \\ m(\pi_x^{\Pi'_h},K)=((1-g_k)/(K-1))^h/N_{\Pi'_h}, \end{cases} \tag{12}$$

where $N_{\Pi'_h}=g_k^h+(1-g_k)^h/(K-1)^{h-1}$.

Note that, the value of $m(\pi_x^{\Pi'_h},k)$ is smaller than that of $m(\pi_x^{\Pi_h},k)$, while the mass values of the other $(k-1)$ clusters in $m(\pi_x^{\Pi'_h})$ are larger than that in $m(\pi_x^{\Pi_h})$. Therefore, we can use Formula (12) to describe the worst situation of Formula (11).

In addition, the ensemble prediction of x is the cluster assigned with the largest mass function, i.e.

$$\pi(x)=\arg\max_{k=1,2,\cdots K} m(\pi_x,k).$$

Then the result of DSCE is correct if

$$\frac{g_k^h}{g_k^h+(1-g_k)^h/(K-1)^{h-1}}>\frac{((1-g_k)/(K-1))^h}{g_k^h+(1-g_k)^h/(K-1)^{h-1}}. \tag{13}$$

Through the above inequality (13), we obtain that $g_k>1/K$. That is to say, if each base partition is better than random partition, DSCE will generate a satisfactory solution.

Next, we will simply compare the performance of DSCE with that of the simple voting when $K=2$ in theory.

Denote the accuracy probability of each base clustering as g, when $K=2$, the probability of accuracy by voting is

$$p_{c\_V}(g)=\sum_{i=\lceil\frac{h+1}{2}\rceil}^{h}\frac{h!}{i!(h-i)!}g^i(1-g)^{h-i}, \tag{14}$$

where h is the number of base clusterings.

Based on Formula (12), when $K=2$, the accuracy probability by using DS-based fusion strategy is

$$p_{c\_D}(g)=\frac{g^h}{g^h+(1-g)^h}. \tag{15}$$

If $p_{c\_D}(g)>p_{c\_V}(g)$, we could deem that DSCE is better than voting. Based on the following property, it can be concluded that if each base partition is better than random partition, i.e., $g>1/2$, the accuracy of DS-based fusion strategy is higher than that of voting.

Property 1. *If* $1/2<g<1$ *and integer* $h>1$, *then.*

$$p_{c\_D}(g)>p_{c\_V}(g). \tag{16}$$

Proof. Based on Formula (14) and Formula (15),

$$p_{c\_D}(g)>p_{c\_V}(g)$$

$$\Leftrightarrow\frac{g^h}{g^h+(1-g)^h}>\sum_{i=\lceil\frac{h+1}{2}\rceil}^{h}\frac{h!}{i!(h-i)!}g^i(1-g)^{h-i}$$

$$\Leftrightarrow 1 > \left(\sum_{i=\lceil\frac{h+1}{2}\rceil}^{h}\frac{h!}{i!(h-i)!}g^i(1-g)^{h-i}\right)\left(\frac{g^h+(1-g)^h}{g^h}\right)$$

$$\Leftrightarrow 1 > \sum_{i=\lceil\frac{h+1}{2}\rceil}^{h}\frac{h!}{i!(h-i)!}g^i(1-g)^{h-i}+\left(\sum_{i=\lceil\frac{h+1}{2}\rceil}^{h}\frac{h!}{i!(h-i)!}g^i(1-g)^{h-i}\right)\left(\frac{1-g}{g}\right)^h$$

$$\Leftrightarrow 1-\sum_{i=\lceil\frac{h+1}{2}\rceil}^{h}\frac{h!}{i!(h-i)!}g^i(1-g)^{h-i} > \sum_{i=\lceil\frac{h+1}{2}\rceil}^{h}\frac{h!}{i!(h-i)!}g^{i-h}(1-g)^{2h-i}. \tag{17}$$

Note that, the probability of accuracy by voting is the summation of the second half in the binomial expansion. Actually, the left part in the inequality (17) is the summation of the first half in the binomial expansion. Then, the left part of the inequality (17) has the following characteristic.

When h is odd number, we have

$$1-\sum_{i=\lceil\frac{h+1}{2}\rceil}^{h}\frac{h!}{i!(h-i)!}g^i(1-g)^{h-i}=\sum_{i=\lceil\frac{h+1}{2}\rceil}^{h}\frac{h!}{i!(h-i)!}(1-g)^ig^{h-i}.$$

When h is even number, we have

$$\begin{aligned}1-&\sum_{i=\lceil\frac{h+1}{2}\rceil}^{h}\frac{h!}{i!(h-i)!}g^i(1-g)^{h-i}\\ &=\sum_{i=\lceil\frac{h+1}{2}\rceil}^{h}\frac{h!}{i!(h-i)!}(1-g)^ig^{h-i}+\frac{h!}{(h/2)!(h/2)!}(1-g)^{\frac{h}{2}}g^{\frac{h}{2}}\\ &>\sum_{i=\lceil\frac{h+1}{2}\rceil}^{h}\frac{h!}{i!(h-i)!}(1-g)^ig^{h-i}.\end{aligned}$$

Then, the inequality (17) is equivalent to

$$\sum_{i=\lceil\frac{h+1}{2}\rceil}^{h}\frac{h!}{i!(h-i)!}(1-g)^ig^{h-i} > \sum_{i=\lceil\frac{h+1}{2}\rceil}^{h}\frac{h!}{i!(h-i)!}g^{i-h}(1-g)^{2h-i}. \tag{18}$$

For each i, if $(1-g)^ig^{h-i} > g^{i-h}(1-g)^{2h-i}$, Formula (17) holds. Then, we have the following derivation:

$$\begin{aligned}&(1-g)^ig^{h-i} > g^{i-h}(1-g)^{2h-i}\\ \Leftrightarrow&\frac{g^{h-i}(1-g)^i}{g^{i-h}(1-g)^{(2h-i)}} > 1\\ \Leftrightarrow&\left(\frac{g}{1-g}\right)^{2(h-i)} > 1.\end{aligned} \tag{19}$$

If $g > 1/2$ and $i < h$, we can obtain inequality (19). Based on the above derivation, the property holds.

This completes the proof. □

This property guarantees that DSCE is more suitable than voting in the application of the binary clustering ensemble problem.

To further illustrate the superiority of DS-based fusion strategy, we will visualize the ensemble accuracy curve of the DS-based fusion strategy and that of voting when $K = 2$ and $K = 3$.

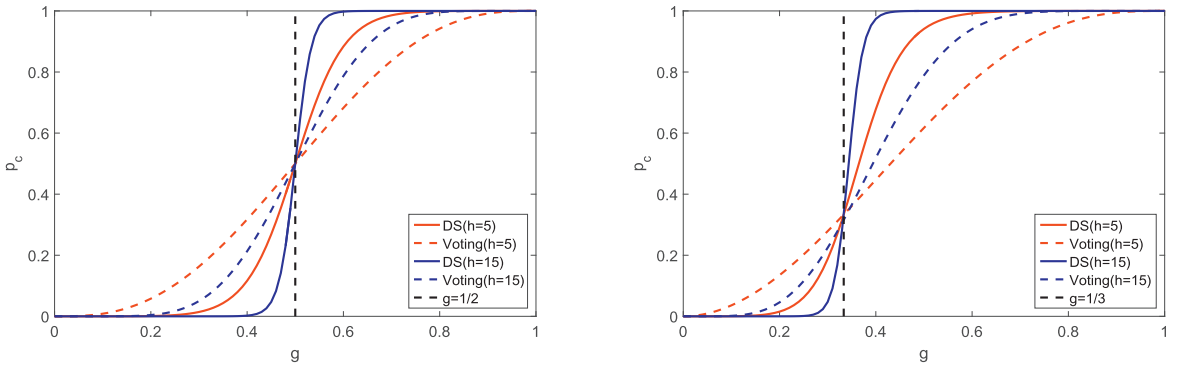
When $K = 3$, the ensemble accuracy by voting is:

$$p_{c\_V}(g)=\sum_{i=\lceil\frac{h+1}{2}\rceil}^{h}\frac{h!}{i!(h-i)!}g^i(1-g)^{h-i}+\sum_{i=\lceil\frac{h+1}{3}\rceil}^{\lceil\frac{h+1}{2}\rceil-1}\sum_{j=h-2i+1}^{i-1}\frac{h!}{i!j!(h-i-j)!}g^i\left(\frac{1-g}{2}\right)^{h-i},$$

where h is the number of partitions and g is the accuracy of each base partition.

For DS-based fusion strategy, the accuracy is:

$$p_{c\_D}(g)=\frac{g^h}{g^h+(1-g)^h/2^{h-1}}.$$



(a) K=2 (b) K=3

Fig. 3. Accuracy curve of voting and DSCE.

Table 4
Data sets.

ID	Data set	Size	Attribute	Class
1	Iris	150	4	3
2	Wine Recognition data	178	13	3
3	Seeds Data Set	210	7	3
4	User Knowledge Modeling	258	5	3
5	Protein Localization Sites	272	7	3
6	Johns Hopkins University Ionosphere database	351	34	2
7	Wisconsin Diagnostic Breast Cancer (WDBC)	569	30	2
8	Pima Indians Diabetes Database	768	8	2
9	Mammographic Mass Data	830	5	2
10	Vehicle Silhouettes	846	18	4
11	Statlog (Landsat Satellite) Data Set-test	1004	36	7
12	Waveform Database Generator 21	5000	21	3
13	Waveform Database Generator 40	5000	40	3
14	Blocks Classification	5473	10	5
15	Pen Digits	10,992	16	10
16	Letter Recognition Data Set	20,000	16	26

Fig. 3 shows the accuracy curves of voting and DSCE in terms of accuracy of base partitions when $K = 2$ (left) and $K = 3$ (right). It is obvious that when $g > 1/K$, the curve of DSCE is higher than that of voting. It can be concluded that DS-based fusion strategy is better than voting only if the base partitions are better than a random partition.

So far, we have made it clear that if the accuracy of base partitions is slightly better than random partition, DSCE will generate a satisfactory decision which is better than that of voting when $K = 2$ and $K = 3$. The superiority of DSCE may owe to the neighbor information of objects used in the definition of mass functions and the multiplication rule of DS theory.

4. Experimental analysis

4.1. Data sets

Sixteen data sets from UCI Machine Learning Repository [1] are used to evaluate the performance of DSCE. The detailed information of these data sets are shown in Table 4. They are all numerical data sets. The size of these data sets ranges from 150 to 20,000 and the number of attribute ranges from 4 to 40. In Table 4, the data sets are sorted by the number of objects.

4.2. Experimental setting

Experiments are set up to compare DSCE with CSPA, HGPA, MCLA, Voting, Weighted Voting (W-V), Selected Voting (S-V) and Selected Weighted Voting (S-W-V). The first three methods are graph-based methods and their codes are available at http://strehl.com/. CSPA is also a co-association-based method. MCLA is also a direct approach. After finding correspondence between each pair clusters, MCLA votes meta-clusters to generate the final clustering. The remaining methods are direct methods which are based on the same method of finding cluster correspondence. Their difference lies in the different voting techniques. Voting method utilizes the plurality voting strategy. W-V computes the weight for each base partition based on the average NMI [37] value, and votes base partitions with weights. S-V votes the base partitions whose weights are bigger

Table 5
Overlap table of U and U'.

$U\backslash U'$	U'_1	U'_2	$\cdots$	U'_K	Sums
U_1	n_{11}	n_{12}	$\cdots$	n_{1K}	b_1
U_2	n_{21}	n_{22}	$\cdots$	n_{2K}	b_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
U_K	n_{K1}	n_{K2}	$\cdots$	n_{KK}	b_K
Sums	b'_1	b'_2	$\cdots$	b'_K	n

than a threshold, which is set as the average weight in the experiment. S-W-V votes the selected partitions with weights. In experiment, DSCE has the same approach to finding cluster correspondence with Voting, W-V, S-V, and S-W-V. By comparing DSCE with Voting, W-V, S-V, S-W-V, we expect to find whether utilizing DS-based fusion strategy in the direct methods is suitable or not. The comparison between DSCE and the seven other methods will show whether DSCE is superior to other methods.

Accuracy (AC) [26] and Adjusted Rand Index (ARI) [16] are employed to evaluate the performance of DSCE and the seven other algorithms. These two indices evaluate the similarity between two clustering results. Suppose $U = \{U_1, U_2, \cdots, U_K\}$ indicates the clustering result and $U' = \{U'_1, U'_2, \cdots, U'_K\}$ is the true label. Table 5 shows the overlap matrix of U and U', in which n_{ij} is the overlapped objects of U_i and U'_j, b_i is the number of objects in cluster U_i and b'_i is the number of objects in cluster U'_i.

AC can be defined as:

$$AC = \sum_{i=1}^{K} \frac{\max_{j=1,2,\ldots,K} n_{ij}}{n}. \tag{20}$$

ARI is defined as:

$$ARI = \frac{r_0 - r_3}{\frac{1}{2}(r_1 + r_2) - r_3}, \tag{21}$$

where

$$r_0 = \sum_{i=1}^{K}\sum_{j=1}^{K}\binom{n_{ij}}{2}, \quad r_1 = \sum_{i=1}^{K}\binom{b_i}{2}, \quad r_2 = \sum_{j=1}^{K}\binom{b'_j}{2}, \quad r_3 = \frac{2r_1 r_2}{n(n-1)}.$$

The experiments contain two parts: base partitions with fixed K, which is equal to the true class number, and base partitions with random K, which is selected from $2{:}\sqrt{n}$ randomly. In the first part, for DSCE, Voting, W-V, S-V and S-W-V, the approach to finding correspondence is Algorithm 2. In the second part, these five algorithms use method in MCLA (Algorithm 3) to find cluster correspondence. Other experiment settings are as follows: (1) Random 1D projection algorithm is used to generate base partitions. (2) The size of the base clusterings h is set as 10 and 20 in fixed K experiment, and set as 50 in random K experiment. (3) To eliminate randomness in the experiment, we run each method 50 times to obtain an average value and standard deviation.

4.3. Experimental results

Experimental results are shown from Tables 6–11. Tables 6 and 7 are results of AC and ARI when there are ten base 10 partitions with fixed K, respectively. Tables 8 and 9 are results of AC and ARI when there are 20 base partitions with fixed K, respectively. Tables 10 and 11 are results based on 50 weak partitions with random K. The maximum value for each data set is underlined and printed in bold type, while the second maximum value is printed in bold type. The last row in each table is the average rank of the algorithms on each data set.

From Tables 6–11, it is easy to find that DSCE is the method with the largest number of bold index, which manifests that DSCE is competent to integrate base partitions with both fixed K and random K. In detail, the average rank of DSCE ranks first in every circumstance, which declares DSCE is high-efficient. From Tables 10 and 11, it can be found that the graph-based methods (CSPA, HGPA and MCLA) are much better than the other direct methods except DSCE, which means that DSCE improves the direct methods by utilizing an efficient fusion strategy.

To show the evaluation results in the above tables more clearly, Figs. 4–6 visualize a statistical value of AC and ARI. This statistical value is the difference between the number of times that a method performs better than the others and the number of times it performs worse than the others. The better and worse are calculated at 95% confidence level. Let $[L_{(i,\,a,\,d)}, U_{(i,\,a,\,d)}]$ be the 95% confidence interval of index $i \in \{AC, ARI\}$, in which the values of index i are obtained by running algorithm $a \in A$ (A is the set of the 8 clustering ensemble algorithms in this paper) n ($n = 50$ in the experiment) times on

Table 6
Index of AC with fixed K when $h = 10$.

	CSPA	HGPA	MCLA	Voting	W-V	S-V	S-W-V	DSCE
1	0.8280 ± 0.0276	0.5533 ± 0.0834	**0.8653 ± 0.0128**	0.8493 ± 0.0153	0.8507 ± 0.0209	0.7560 ± 0.1648	0.7253 ± 0.2262	**0.8627 ± 0.0138**
2	**0.7393 ± 0.0176**	0.5809 ± 0.0914	0.7112 ± 0.0050	0.7124 ± 0.0259	0.6899 ± 0.0279	0.6652 ± 0.0361	0.6247 ± 0.1263	**0.7191 ± 0.0191**
3	0.8343 ± 0.0114	0.5857 ± 0.1726	**0.8410 ± 0.0120**	0.8352 ± 0.0080	0.8257 ± 0.0170	0.7190 ± 0.1376	0.7048 ± 0.1254	**0.8457 ± 0.0072**
4	0.6674 ± 0.0069	0.6628 ± 0.0000	**0.7256 ± 0.0215**	0.7233 ± 0.0245	0.7171 ± 0.0346	0.7008 ± 0.0364	0.7124 ± 0.0544	**0.7264 ± 0.0364**
5	0.7537 ± 0.0225	0.5971 ± 0.0955	0.7596 ± 0.0252	**0.7603 ± 0.0204**	0.7559 ± 0.0278	0.6993 ± 0.0365	0.7154 ± 0.0465	**0.7838 ± 0.0416**
6	**0.6883 ± 0.0025**	0.6410 ± 0.0000	0.6872 ± 0.0037	0.6849 ± 0.0025	0.6866 ± 0.0040	0.6701 ± 0.0065	0.6798 ± 0.0069	**0.6872 ± 0.0024**
7	0.7209 ± 0.0390	0.6274 ± 0.0000	**0.8538 ± 0.0008**	0.8510 ± 0.0034	0.8496 ± 0.0042	0.8394 ± 0.0064	0.7989 ± 0.0959	**0.8524 ± 0.0018**
8	0.6805 ± 0.0024	0.6510 ± 0.0000	**0.6919 ± 0.0054**	0.6901 ± 0.0081	**0.6919 ± 0.0061**	0.6901 ± 0.0143	0.6862 ± 0.0113	**0.6927 ± 0.0048**
9	0.7892 ± 0.0047	0.5145 ± 0.0000	0.7995 ± 0.0023	**0.7998 ± 0.0010**	0.7990 ± 0.0010	0.7872 ± 0.0212	0.7867 ± 0.0212	**0.7998 ± 0.0010**
10	0.3991 ± 0.0191	0.2589 ± 0.0000	**0.4189 ± 0.0251**	0.3671 ± 0.0429	0.3792 ± 0.0426	0.3629 ± 0.0564	0.3962 ± 0.0355	**0.4035 ± 0.0264**
11	0.6305 ± 0.0080	0.5697 ± 0.0491	**0.6601 ± 0.0161**	0.5998 ± 0.0505	0.5980 ± 0.0578	0.5857 ± 0.0769	0.6120 ± 0.0534	**0.6592 ± 0.0160**
12	0.3849 ± 0.0218	0.3510 ± 0.0084	0.3784 ± 0.0153	0.4173 ± 0.0209	0.4335 ± 0.0274	**0.4411 ± 0.0144**	**0.4583 ± 0.0317**	0.3929 ± 0.0237
13	0.3597 ± 0.0090	0.3498 ± 0.0087	0.3618 ± 0.0130	0.3704 ± 0.0263	0.3829 ± 0.0392	**0.3914 ± 0.0431**	**0.4091 ± 0.0511**	0.3731 ± 0.0335
14	0.8977 ± 0.0000	0.8977 ± 0.0000	0.8977 ± 0.0000	**0.8990 ± 0.0007**	0.8988 ± 0.0008	0.8988 ± 0.0007	0.8985 ± 0.0007	**0.8997 ± 0.0005**
15	0.2366 ± 0.0097	0.2004 ± 0.0160	0.2379 ± 0.0149	**0.2411 ± 0.0144**	0.2401 ± 0.0133	0.2381 ± 0.0164	0.1865 ± 0.0255	**0.2457 ± 0.0064**
16	0.6933 ± 0.0301	0.1125 ± 0.0000	0.7375 ± 0.0481	**0.7411 ± 0.0302**	0.7350 ± 0.0375	0.6741 ± 0.0479	0.6018 ± 0.1154	**0.7393 ± 0.0044**
rank	4.9375	7.875	3.1875	3.375	3.9375	5.3125	5.4375	1.9375

Table 7
Index of ARI with fixed K when $h = 10$.

	CSPA	HGPA	MCLA	Voting	W-V	S-V	S-W-V	DSCE
1	0.5868 ± 0.0566	0.1831 ± 0.1299	**0.6614 ± 0.0289**	0.6277 ± 0.0333	0.6285 ± 0.0472	0.4759 ± 0.2615	0.4545 ± 0.2787	**0.6575 ± 0.0282**
2	**0.4221 ± 0.0192**	0.1941 ± 0.1255	0.3765 ± 0.0076	**0.3957 ± 0.0449**	0.3663 ± 0.0474	0.3302 ± 0.0518	0.2793 ± 0.1562	0.3955 ± 0.0407
3	0.5838 ± 0.0242	0.2612 ± 0.2353	**0.5945 ± 0.0216**	0.5819 ± 0.0178	0.5651 ± 0.0351	0.4075 ± 0.2076	0.3853 ± 0.1887	**0.6044 ± 0.0142**
4	0.0741 ± 0.0222	0.0272 ± 0.0160	**0.1542 ± 0.0348**	0.1492 ± 0.0387	0.1344 ± 0.0590	0.0874 ± 0.0650	0.1096 ± 0.0952	**0.1620 ± 0.0441**
5	0.3726 ± 0.0414	0.0897 ± 0.1373	**0.4582 ± 0.0476**	0.4459 ± 0.0491	0.4219 ± 0.0604	0.2921 ± 0.0232	0.3402 ± 0.0672	**0.4881 ± 0.0954**
6	**0.1396 ± 0.0038**	−0.0026 ± 0.0000	**0.1379 ± 0.0056**	0.1344 ± 0.0037	0.1370 ± 0.0060	0.1134 ± 0.0088	0.1271 ± 0.0099	0.1378 ± 0.0036
7	0.1987 ± 0.0698	−0.0016 ± 0.0000	**0.4904 ± 0.0023**	0.4821 ± 0.0099	0.4780 ± 0.0124	0.4485 ± 0.0184	0.3644 ± 0.2040	**0.4862 ± 0.0052**
8	0.1293 ± 0.0034	−0.0012 ± 0.0000	0.1446 ± 0.0080	0.1417 ± 0.0112	**0.1448 ± 0.0093**	0.1425 ± 0.0214	0.1351 ± 0.0165	**0.1456 ± 0.0072**
9	0.3337 ± 0.0109	−0.0012 ± 0.0000	0.3581 ± 0.0056	**0.3587 ± 0.0024**	0.3569 ± 0.0024	0.3306 ± 0.0464	0.3295 ± 0.0466	**0.3587 ± 0.0024**
10	0.0807 ± 0.0088	−0.0035 ± 0.0000	**0.0934 ± 0.0265**	0.0593 ± 0.0333	0.0683 ± 0.0356	0.0537 ± 0.0373	0.0770 ± 0.0303	**0.0879 ± 0.0126**
11	0.2691 ± 0.0098	0.2106 ± 0.0518	**0.3180 ± 0.0209**	0.2329 ± 0.0586	0.2245 ± 0.0516	0.2290 ± 0.0779	0.2506 ± 0.0737	**0.3308 ± 0.0361**
12	0.0115 ± 0.0066	0.0009 ± 0.0011	0.0081 ± 0.0044	0.0265 ± 0.0124	0.0390 ± 0.0204	**0.0436 ± 0.0187**	**0.0640 ± 0.0370**	0.0127 ± 0.0070
13	0.0029 ± 0.0024	0.0008 ± 0.0010	0.0034 ± 0.0028	0.0070 ± 0.0088	0.0137 ± 0.0174	**0.0194 ± 0.0228**	**0.0322 ± 0.0395**	0.0087 ± 0.0152
14	0.0011 ± 0.0003	0.0044 ± 0.0008	−0.0010 ± 0.0002	0.0156 ± 0.0131	**0.0193 ± 0.0200**	0.0146 ± 0.0434	−0.0012 ± 0.0061	**0.0206 ± 0.0175**
15	0.1305 ± 0.0057	0.0981 ± 0.0083	0.1289 ± 0.0090	0.1324 ± 0.0073	**0.1331 ± 0.0075**	0.1298 ± 0.0098	0.0694 ± 0.0178	**0.1343 ± 0.0026**
16	0.5175 ± 0.0323	0.0000 ± 0.0000	0.5758 ± 0.0421	**0.5847 ± 0.0340**	0.5776 ± 0.0273	0.5271 ± 0.0489	0.4358 ± 0.1216	**0.5966 ± 0.0016**
rank	4.7500	7.7500	3.3125	3.6563	3.7500	5.2500	5.6250	1.9063

Table 8
Index of AC with fixed K when $h = 20$.

	CSPA	HGPA	MCLA	Voting	W-V	S-V	S-W-V	DSCE
1	0.8133 ± 0.0208	0.5567 ± 0.0510	**0.8560 ± 0.0064**	0.8420 ± 0.0290	0.8280 ± 0.0589	0.7453 ± 0.0666	0.7213 ± 0.1183	**0.8533 ± 0.0089**
2	**0.7416 ± 0.0038**	0.5562 ± 0.0000	0.7096 ± 0.0023	0.7056 ± 0.0025	0.7006 ± 0.0018	0.6848 ± 0.0056	0.6837 ± 0.0047	**0.7118 ± 0.0020**
3	0.8390 ± 0.0035	0.5619 ± 0.0000	**0.8448 ± 0.0036**	0.8295 ± 0.0027	0.8271 ± 0.0898	0.7714 ± 0.0895	0.7681 ± 0.0897	**0.8410 ± 0.0027**
4	0.6771 ± 0.0042	0.6690 ± 0.0000	0.7182 ± 0.0094	**0.7217 ± 0.0101**	0.7178 ± 0.0094	0.7043 ± 0.0185	0.6984 ± 0.0162	**0.7271 ± 0.0111**
5	0.7665 ± 0.0063	0.5827 ± 0.0445	**0.7713 ± 0.0151**	0.7673 ± 0.0183	0.7632 ± 0.0195	0.7107 ± 0.0572	0.7188 ± 0.0556	**0.7820 ± 0.0172**
6	**0.6877 ± 0.0080**	0.6410 ± 0.0976	0.6875 ± 0.0090	0.6869 ± 0.0132	0.6872 ± 0.0165	0.6863 ± 0.0261	0.6866 ± 0.0280	**0.6883 ± 0.0119**
7	0.7529 ± 0.0173	0.6274 ± 0.0592	**0.8540 ± 0.0066**	0.8511 ± 0.0879	0.8501 ± 0.0784	0.8404 ± 0.0194	0.8406 ± 0.0352	**0.8522 ± 0.0071**
8	0.6799 ± 0.0195	0.6510 ± 0.0196	**0.6915 ± 0.0186**	0.6911 ± 0.0285	0.6900 ± 0.0307	0.6849 ± 0.0400	0.6848 ± 0.0411	**0.6918 ± 0.0250**
9	0.7893 ± 0.0134	0.5145 ± 0.0000	0.7967 ± 0.0033	**0.7970 ± 0.0234**	0.7700 ± 0.0309	0.7583 ± 0.0452	0.7589 ± 0.0642	**0.7978 ± 0.0100**
10	0.4067 ± 0.0193	0.2589 ± 0.0744	**0.4317 ± 0.0027**	0.4152 ± 0.0128	0.4116 ± 0.0205	0.3766 ± 0.0032	0.3719 ± 0.0060	**0.4251 ± 0.0038**
11	0.6219 ± 0.0196	0.5321 ± 0.0000	**0.6697 ± 0.0006**	0.6058 ± 0.0022	0.6028 ± 0.0026	0.5873 ± 0.0016	0.5677 ± 0.0020	**0.6600 ± 0.0019**
12	0.3764 ± 0.0000	0.3532 ± 0.0000	0.3652 ± 0.0000	0.3586 ± 0.0008	0.3584 ± 0.0009	**0.3970 ± 0.0008**	**0.4005 ± 0.0008**	0.3722 ± 0.0005
13	0.3559 ± 0.0200	0.3470 ± 0.0033	0.3577 ± 0.0123	0.3646 ± 0.0110	0.3648 ± 0.0092	**0.3770 ± 0.0102**	**0.3776 ± 0.0125**	0.3561 ± 0.0121
14	0.8977 ± 0.0126	0.8977 ± 0.0050	0.8977 ± 0.0110	**0.8991 ± 0.0112**	0.8990 ± 0.0071	0.8988 ± 0.0265	0.8990 ± 0.0231	**0.8997 ± 0.0134**
15	0.2339 ± 0.0057	0.1983 ± 0.0059	0.2405 ± 0.0131	**0.2460 ± 0.0090**	0.2442 ± 0.0089	0.2444 ± 0.0131	0.1949 ± 0.0208	**0.2477 ± 0.0021**
16	0.6806 ± 0.0206	0.1125 ± 0.0000	0.7334 ± 0.0128	0.7364 ± 0.0154	**0.7370 ± 0.0159**	0.7016 ± 0.0304	0.5727 ± 0.0391	**0.7367 ± 0.0036**
rank	4.8125	7.8750	2.9375	3.3125	4.2188	5.2500	5.6563	1.9375

Table 9
Index of ARI with fixed K when $h = 20$.

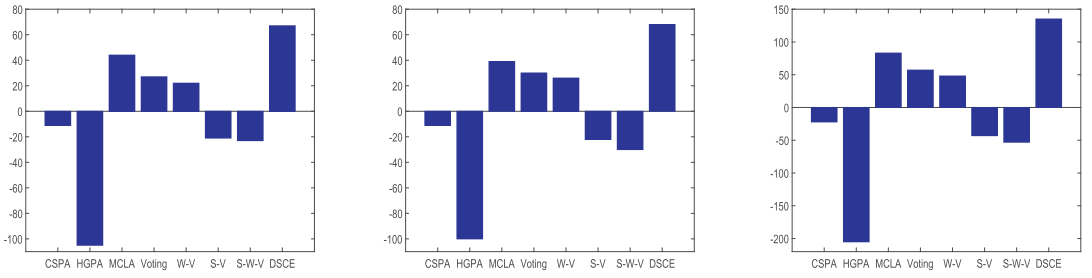
	CSPA	HGPA	MCLA	Voting	W-V	S-V	S-W-V	DSCE
1	0.5562 ± 0.0439	0.1746 ± 0.0722	**0.6457 ± 0.0123**	0.6118 ± 0.0706	0.5877 ± 0.1195	0.4225 ± 0.1301	0.4017 ± 0.1772	**0.6393 ± 0.0183**
2	**0.4358 ± 0.0271**	0.1583 ± 0.1047	0.3740 ± 0.0040	0.3731 ± 0.0100	0.3700 ± 0.0160	0.3564 ± 0.0067	0.3530 ± 0.0094	**0.3850 ± 0.0245**
3	0.5937 ± 0.0180	0.2149 ± 0.1621	**0.6032 ± 0.0175**	0.5743 ± 0.0212	0.5716 ± 0.0288	0.4914 ± 0.0377	0.4852 ± 0.0404	**0.5974 ± 0.0216**
4	0.0856 ± 0.0277	0.0540 ± 0.0322	**0.1550 ± 0.0262**	0.1499 ± 0.0427	0.1394 ± 0.0531	0.0919 ± 0.0700	0.1018 ± 0.0630	**0.1560 ± 0.0628**
5	0.3915 ± 0.0134	0.0730 ± 0.0478	**0.4827 ± 0.0298**	0.4625 ± 0.0423	0.4546 ± 0.0433	0.3373 ± 0.0968	0.3500 ± 0.1111	**0.4973 ± 0.0329**
6	**0.1388 ± 0.0058**	−0.0026 ± 0.0000	0.1383 ± 0.0036	0.1374 ± 0.0037	0.1378 ± 0.0027	0.1366 ± 0.0083	0.1370 ± 0.0069	**0.1395 ± 0.0030**
7	0.2560 ± 0.0395	−0.0016 ± 0.0000	**0.4909 ± 0.0016**	0.4826 ± 0.0065	0.4795 ± 0.0077	0.4515 ± 0.0046	0.4520 ± 0.0059	**0.4857 ± 0.0057**
8	0.1286 ± 0.0059	−0.0012 ± 0.0000	**0.1444 ± 0.0140**	0.1437 ± 0.0149	0.1418 ± 0.0137	0.1343 ± 0.0260	0.1342 ± 0.0232	**0.1448 ± 0.0167**
9	0.3340 ± 0.0081	−0.0012 ± 0.0000	0.3515 ± 0.0086	**0.3521 ± 0.0065**	0.3199 ± 0.1125	0.2949 ± 0.1163	0.2963 ± 0.1169	**0.3541 ± 0.0063**
10	0.0842 ± 0.0076	−0.0035 ± 0.0000	**0.1077 ± 0.0023**	0.0949 ± 0.0155	0.0906 ± 0.0276	0.0560 ± 0.0327	0.0641 ± 0.0374	**0.1035 ± 0.0135**
11	0.2611 ± 0.0211	0.1757 ± 0.0438	**0.3143 ± 0.0055**	0.2462 ± 0.0771	0.2380 ± 0.0677	0.2272 ± 0.0398	0.2038 ± 0.0525	**0.3246 ± 0.0211**
12	0.0082 ± 0.0079	0.0013 ± 0.0006	0.0037 ± 0.0027	0.0019 ± 0.0013	0.0017 ± 0.0016	**0.0145 ± 0.0067**	**0.0150 ± 0.0077**	0.0050 ± 0.0035
13	0.0019 ± 0.0019	0.0003 ± 0.0005	0.0026 ± 0.0025	0.0038 ± 0.0025	0.0038 ± 0.0017	**0.0082 ± 0.0094**	**0.0084 ± 0.0068**	0.0022 ± 0.0026
14	0.0012 ± 0.0006	0.0034 ± 0.0014	0.0005 ± 0.0018	0.0039 ± 0.0105	**0.0116 ± 0.0268**	0.0100 ± 0.0331	**0.0276 ± 0.0430**	0.0104 ± 0.0143
15	0.1304 ± 0.0051	0.0928 ± 0.0070	0.1312 ± 0.0053	0.1384 ± 0.0037	0.1386 ± 0.0033	**0.1403 ± 0.0052**	0.0605 ± 0.0151	**0.1406 ± 0.0013**
16	0.5124 ± 0.0247	0.0000 ± 0.0000	0.5967 ± 0.0167	0.6063 ± 0.0217	**0.6071 ± 0.0223**	0.5561 ± 0.0468	0.3931 ± 0.0406	**0.6064 ± 0.0026**
rank	4.8750	7.8125	3.0000	3.6563	4.0313	5.2500	5.3750	2.0000

Table 10
Index of AC with random K when $h = 50$.

	CSPA	HGPA	MCLA	Voting	W-V	S-V	S-W-V	DSCE
1	0.8143 ± 0.0004	**0.8183 ± 0.0003**	0.8080 ± 0.0005	0.8107 ± 0.0005	0.8057 ± 0.0004	0.6687 ± 0.0039	0.5190 ± 0.0301	**0.8250 ± 0.0003**
2	**0.6463 ± 0.0002**	0.6236 ± 0.0003	0.6250 ± 0.0000	0.6292 ± 0.0000	0.6270 ± 0.0000	0.6028 ± 0.0004	0.5073 ± 0.0128	**0.6430 ± 0.0002**
3	**0.8405 ± 0.0001**	0.8390 ± 0.0001	0.8393 ± 0.0001	0.8393 ± 0.0001	0.8386 ± 0.0001	0.8112 ± 0.0005	0.6583 ± 0.0424	**0.8424 ± 0.0001**
4	0.6698 ± 0.0002	0.6632 ± 0.0000	**0.6973 ± 0.0003**	0.6746 ± 0.0002	0.6711 ± 0.0001	0.6628 ± 0.0000	0.6647 ± 0.0001	**0.6760 ± 0.0003**
5	0.7614 ± 0.0000	**0.7662 ± 0.0001**	0.7619 ± 0.0000	0.7651 ± 0.0000	0.7653 ± 0.0000	0.7594 ± 0.0001	0.6511 ± 0.0136	**0.7673 ± 0.0000**
6	0.6833 ± 0.0000	0.6776 ± 0.0001	0.6842 ± 0.0001	**0.6843 ± 0.0000**	0.6838 ± 0.0000	0.6799 ± 0.0001	0.6801 ± 0.0001	**0.6858 ± 0.0000**
7	0.8490 ± 0.0000	**0.8662 ± 0.0001**	0.8540 ± 0.0000	0.8489 ± 0.0000	0.8489 ± 0.0000	0.8476 ± 0.0004	0.8463 ± 0.0003	**0.8554 ± 0.0000**
8	**0.6792 ± 0.0000**	0.6758 ± 0.0001	0.6777 ± 0.0000	0.6756 ± 0.0000	0.6758 ± 0.0000	0.6753 ± 0.0001	0.6762 ± 0.0001	**0.6796 ± 0.0000**
9	0.7942 ± 0.0000	**0.8064 ± 0.0000**	0.7948 ± 0.0000	0.7929 ± 0.0000	0.7931 ± 0.0000	0.7956 ± 0.0000	0.7975 ± 0.0000	**0.7977 ± 0.0000**
10	0.3533 ± 0.0000	**0.3665 ± 0.0001**	0.3541 ± 0.0000	0.3520 ± 0.0000	0.3524 ± 0.0000	0.3540 ± 0.0002	0.3468 ± 0.0007	**0.3563 ± 0.0001**
11	0.6344 ± 0.0001	0.6242 ± 0.0002	**0.6505 ± 0.0002**	0.6348 ± 0.0001	0.6362 ± 0.0001	0.6249 ± 0.0009	0.4657 ± 0.0120	**0.6497 ± 0.0001**
12	**0.3935 ± 0.0015**	0.3646 ± 0.0002	0.3588 ± 0.0001	0.3585 ± 0.0001	0.3558 ± 0.0001	0.3884 ± 0.0007	**0.4222 ± 0.0029**	0.3675 ± 0.0004
13	0.3628 ± 0.0002	**0.3630 ± 0.0001**	0.3553 ± 0.0001	0.3558 ± 0.0001	0.3545 ± 0.0001	0.3627 ± 0.0001	**0.3857 ± 0.0015**	0.3576 ± 0.0001
14	0.8977 ± 0.0000	0.8977 ± 0.0000	0.8977 ± 0.0000	0.8977 ± 0.0000	0.8977 ± 0.0000	0.8977 ± 0.0000	0.8977 ± 0.0000	0.8977 ± 0.0000
15	0.2422 ± 0.0073	0.2033 ± 0.0154	0.2635 ± 0.0170	**0.2658 ± 0.0176**	0.2657 ± 0.0174	0.2646 ± 0.0140	0.2350 ± 0.0139	**0.2678 ± 0.0184**
16	**0.7132 ± 0.0228**	0.2715 ± 0.0726	0.6917 ± 0.0576	0.6862 ± 0.0573	0.6894 ± 0.0602	0.6836 ± 0.0566	0.6409 ± 0.0804	**0.7166 ± 0.0264**
rank	3.8438	4.5625	3.9375	4.7188	5.0938	5.8438	5.9688	2.0313

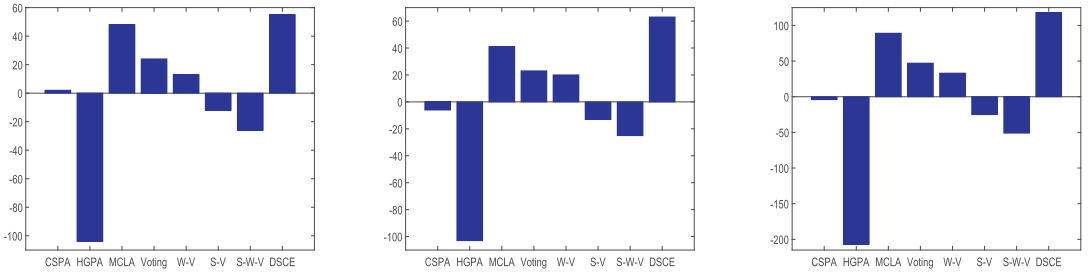
Table 11
Index of ARI with random K when $h = 50$.

	CSPA	HGPA	MCLA	Voting	W-V	S-V	S-W-V	DSCE
1	0.5463 ± 0.0016	**0.5626 ± 0.0019**	0.5293 ± 0.0027	0.5366 ± 0.0026	0.5272 ± 0.0024	0.3130 ± 0.0081	0.1819 ± 0.0327	**0.5761 ± 0.0015**
2	**0.3063 ± 0.0003**	0.2845 ± 0.0005	0.2589 ± 0.0002	0.2755 ± 0.0002	0.2703 ± 0.0002	0.2224 ± 0.0011	0.1203 ± 0.0162	**0.3170 ± 0.0006**
3	**0.5995 ± 0.0003**	0.5905 ± 0.0002	0.5934 ± 0.0002	0.5948 ± 0.0003	0.5941 ± 0.0003	0.5482 ± 0.0019	0.3845 ± 0.0597	**0.6016 ± 0.0003**
4	0.0835 ± 0.0004	0.0691 ± 0.0003	**0.1285 ± 0.0004**	**0.0925 ± 0.0003**	0.0878 ± 0.0002	0.0478 ± 0.0003	0.0367 ± 0.0019	0.0878 ± 0.0004
5	0.3740 ± 0.0001	**0.4250 ± 0.0006**	0.3628 ± 0.0001	0.3733 ± 0.0001	0.3735 ± 0.0001	0.3736 ± 0.0006	0.2106 ± 0.0408	**0.3914 ± 0.0002**
6	0.1322 ± 0.0000	0.1240 ± 0.0001	0.1336 ± 0.0001	**0.1338 ± 0.0001**	0.1329 ± 0.0001	0.1274 ± 0.0002	0.1277 ± 0.0002	**0.1358 ± 0.0001**
7	0.4865 ± 0.0002	**0.5359 ± 0.0012**	0.5006 ± 0.0001	0.4860 ± 0.0002	0.4863 ± 0.0001	0.4839 ± 0.0030	0.4801 ± 0.0026	**0.5047 ± 0.0003**
8	**0.1274 ± 0.0000**	0.1227 ± 0.0002	0.1253 ± 0.0000	0.1223 ± 0.0001	0.1226 ± 0.0000	0.1224 ± 0.0003	0.1234 ± 0.0002	**0.1281 ± 0.0001**
9	0.3453 ± 0.0000	**0.3750 ± 0.0002**	0.3468 ± 0.0000	0.3424 ± 0.0001	0.3428 ± 0.0000	0.3489 ± 0.0002	0.3535 ± 0.0002	**0.3536 ± 0.0001**
10	0.0563 ± 0.0000	**0.0576 ± 0.0000**	0.0572 ± 0.0000	0.0560 ± 0.0000	0.0563 ± 0.0000	0.0550 ± 0.0000	0.0500 ± 0.0002	**0.0598 ± 0.0000**
11	0.2701 ± 0.0000	0.2659 ± 0.0002	0.2858 ± 0.0001	0.2717 ± 0.0000	0.2728 ± 0.0001	**0.3076 ± 0.0027**	0.1603 ± 0.0136	**0.2878 ± 0.0002**
12	**0.0183 ± 0.0006**	0.0035 ± 0.0000	0.0020 ± 0.0000	0.0020 ± 0.0000	0.0015 ± 0.0000	0.0147 ± 0.0002	**0.0363 ± 0.0013**	0.0054 ± 0.0000
13	**0.0034 ± 0.0000**	0.0033 ± 0.0000	0.0017 ± 0.0000	0.0018 ± 0.0000	0.0015 ± 0.0000	**0.0034 ± 0.0000**	**0.0153 ± 0.0003**	0.0023 ± 0.0000
14	0.0110 ± 0.0000	**0.0298 ± 0.0004**	0.0159 ± 0.0000	0.0025 ± 0.0000	−0.0002 ± 0.0000	−0.0028 ± 0.0000	0.0167 ± 0.0054	**0.0218 ± 0.0001**
15	0.1415 ± 0.0070	0.0993 ± 0.0075	0.1493 ± 0.0113	0.1504 ± 0.0119	**0.1506 ± 0.0118**	0.1505 ± 0.0110	0.1062 ± 0.0206	**0.1509 ± 0.0084**
16	0.5649 ± 0.0255	0.1015 ± 0.0546	**0.5687 ± 0.0485**	0.5592 ± 0.0506	0.5615 ± 0.0527	0.5556 ± 0.0542	0.4939 ± 0.0719	**0.5945 ± 0.0261**
rank	3.8125	4.2500	4.2813	5.0938	5.1875	5.4688	6.0000	1.9063



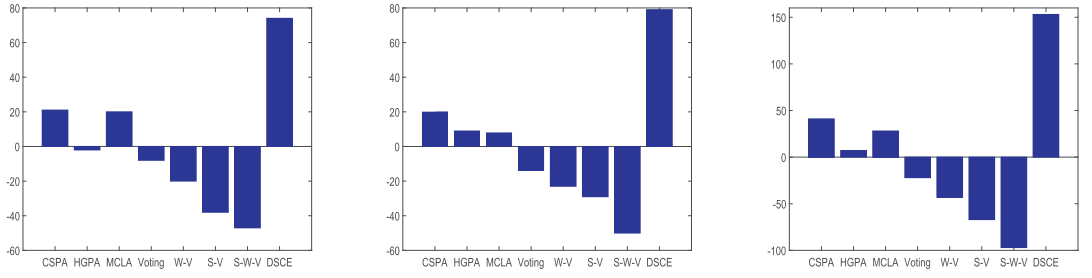
(a) Statistical performance of AC (b) Statistical performance of ARI (c) Total statistical performance

Fig. 4. Statistical performance when K is fixed and $h = 10$.



(a) Statistical performance of AC (b) Statistical performance of ARI (c) Total statistical performance

Fig. 5. Statistical performance when K is fixed and $h = 20$.



(a) Statistical performance of AC (b) Statistical performance of ARI (c) Total statistical performance

Fig. 6. Statistical performance when K is random and $h = 50$.

data set $d \in D$ (D is the set of 16 data sets in this paper). $L_{(i,\,a,\,d)}$ and $U_{(i,\,a,\,d)}$ are calculated by:

$$\begin{aligned} L_{(i,a,d)} &= \mu_{(i,a,d)} - 1.96\frac{\sigma_{(i,a,d)}}{\sqrt{n}}, \\ U_{(i,a,d)} &= \mu_{(i,a,d)} + 1.96\frac{\sigma_{(i,a,d)}}{\sqrt{n}}, \end{aligned} \tag{22}$$

where $\mu_{(i,\,a,\,d)}$ and $\sigma_{(i,\,a,\,d)}$ indicate the average value and the standard deviation based on parameters (i, a, d) respectively. The number of times that an algorithm performs better than others in the sense of index $i \in \{AC, ARI\}$ is expressed as $B_{(i,\,a)}$:

$$B_{(i,a)} = \sum_{d \in D} \sum_{a^* \in A, a^* \neq a} better_{(i,d)}(a, a^*), \tag{23}$$

where

$$better_{(i,d)}(a, a^*) = \begin{cases} 1, & L_{(i,a,d)} > U_{(i,a^*,d)}; \\ 0, & L_{(i,a,d)} \leq U_{(i,a^*,d)}. \end{cases}$$

In the same way, the number of times that an algorithm performs worse than others in the sense of index $i \in \{AC, ARI\}$ is defined as $W_{(i,\,a)}$:

$$W_{(i,a)} = \sum_{d \in D} \sum_{a^* \in A, a^* \neq a} worse_{(i,d)}(a, a^*), \tag{24}$$

where

$$worse_{(i,d)}(a, a^*) = \begin{cases} 1, & U_{(i,a,d)} > L_{(i,a^*,d)}; \\ 0, & U_{(i,a,d)} \leq L_{(i,a^*,d)}. \end{cases}$$

Statistical performance of clustering ensemble algorithm $a \in A$ with respect to index $i \in \{AC, ARI\}$ is computed by:

$$(B - W)_{(i,a)} = B_{(i,a)} - U_{(i,a)}. \tag{25}$$

Total statistical performance integrating indices AC and ARI of an algorithm $a \in A$ is calculated by:

$$(B - W)_a = \sum_{i \in \{AC, ARI\}} (B_{(i,a)} - U_{(i,a)}). \tag{26}$$

This performance statistics assesses the average performance of an algorithm, which can avoid the possible situation that an algorithm runs abnormally at one time [17]. Figs. 4–6 show this statistical value based on Tables 6–11. As shown in Figs. 4 and 5, DSCE possesses the maximum value both in single index and total statistical performance when the K value of the base partitions is fixed. When the K values of the base partitions are random, it is easy to see from Fig. 6 that DSCE is significantly much better than the other seven clustering ensemble methods. In this experiment, DSCE, MCLA, Voting, W-V, S-V and S-W-V share the same approach to finding cluster correspondence, which illustrates that the DS-based fusion strategy is much better than voting on meta-cluster, voting, weighted voting, selected voting and selected weighted voting. In Fig. 6, without considering DSCE, the graph-based methods show better performance than the direct methods. As a direct method, DSCE is much better than the three graph-based methods, which illustrates that an efficient fusion strategy can make up for the shortage in finding cluster correspondence. Further, the experiment illustrates that the fusion strategy has a great influence on the performance of a direct method.

5. Conclusions

Multiple clustering systems can granulate a data set into multiple granular structures. Thus, clustering ensemble is a branch of multigranulation information fusion. Many clustering ensemble methods have been proposed. This paper focuses on the direct methods. Considering that most of the direct methods only pay attention to the process of finding cluster correspondence but neglect the fusion process, this paper has proposed a Dempster-Shafer-based clustering ensemble method (DSCE) which mainly pays attention to the fusion process. The proposed algorithm contains four steps: (1) finding neighbors of each object, (2) finding cluster correspondence, (3) defining mass functions and (4) employing Dempster's combination rule.

The characteristics of clustering analysis indicate that an object and its neighbors are often in the same cluster. With the neighbor information, it is easy to evaluate an object's probability within each cluster, which is called label probabilities. The benefit of using neighbors to describe the measured object is that the surrounding underlying structure is considered and blind trust in the obtained label is avoided. The introduced DS-based fusion strategy aims at combining the base clustering results with label probabilities after the correspondence is determined. Both theoretical analysis and experiments indicate that this algorithm performs well. Thus, we can draw a conclusion that the fusion strategy in a direct method has a great influence on the final result.

In the future, DSCE could be improved from the following two aspects. (1) The process of finding neighbors in DSCE is time-consuming, because it will take a lot time to construct the co-association matrix. How to improve the speed of finding neighbors will be an interesting and challenging research in the future. (2) It is feasible to develop a method of finding cluster correspondence based on neighbor information, which will make DSCE more systematic.

In addition, there are many elements that can affect the performance of a clustering ensemble method, such as characteristics of base partitions, expression of information matrix, fusion strategies, etc. This paper simply explores the influence of fusion strategy in the direct methods. It is interesting to explore the influence of each listed element. This series of work is also helpful to explain why integrating multiple clustering systems can generate a much better result than a single clustering.

Acknowledgements

This work was supported by National Natural Science Fund of China (Nos. 61672332, 61322211, 61432011, and U1435212), Program for New Century Excellent Talents in University (No. NCET-12-1031), Program for the Outstanding Innovative Teams of Higher Learning Institutions of Shanxi, and Program for the Young San Jin Scholars of Shanxi.

References

[1] A. Asuncion, D. Newman, UCI Machine Learning Repository, 2007, http://www.ics.uci.edu/mlearn/MLRepository.html, School of Information and Computer Science, University of California Irvine.
[2] I. Bloch, Some aspects of Dempster-Shafer evidence theory for classification of multi-modality medical images taking partial volume effect into account, Pattern Recognit. Lett. 17 (8) (1996) 905–916.
[3] C. Boulis, M. Ostendorf, Combining multiple clustering systems, Knowl. Discov. Databases 3202 (2004) 63–74.
[4] C. Domeniconi, M. Al-Razgan, Weighted cluster ensembles: methods and analysis, ACM Trans. Knowl. Discov. Data 2 (4) (2009) 1–40.
[5] L. Du, Y.D. Shen, Z.Y. Shen, J.Y. Wang, Z.W. Xu, A self-supervised framework for clustering ensemble, in: Proceedings of the 14th International Conference on Web-Age Information Management, 2013, pp. 253–264. Beidaihe, China
[6] P.T. Edlefsen, C.H. liu, A.P. Dempster, Estimating limits from poisson counting data using Dempster-Shafer analysis, Ann. Appl. Stat. 3 (2009) 764–790.
[7] X.Z. Fern, C.E. Brodley, Solving cluster ensemble problems by Bipartite graph partitioning, in: proceedings of the Twenty-first International Conference on Machine Learning, 2004, pp. 36–43. Banff, Alberta, Canada
[8] B. Fischer, J.M. Buhmann, Bagging for path-based clustering, IEEE Trans. Pattern Anal. Mach. Intell. 25 (11) (2003) 1411–1415.
[9] A.L.N. Fred, A.K. Jain, Combining multiple clusterings using evidence accumulation, IEEE Trans. Pattern Anal. Mach. Intell. 27 (6) (2005) 835–850.
[10] A. Gionis, H. Mannila, P. Tsaparas, Clustering aggregation, ACM Trans. Knowl. Discov. Data 1 (1) (2007) 341–352.
[11] J.R. Hobbs, Granularity, in: Proceedings of the Ninth International Joint Conference on Artificial Intelligence, 1985, pp. 432–435. Los Angeles, California
[12] J. Hu, T.R. Li, H.J. Wang, H. Fujita, Hierarchical cluster ensemble model based on knowledge granulation, Knowl.-Based Syst. 91 (2016) 179–188.
[13] Q.H. Hu, D.R. Yu, J.F. Liu, C.X. Wu, Neighborhood rough set based heterogeneous feature subset selection, Inf. Sci. 178 (18) (2008) 3577–3594.
[14] S.D. Huang, H.J. Wang, D.C. Li, Y. Yang, T.R. Li, Spectral co-clustering ensemble, Knowl.-Based Syst. 84 (2015) 46–55.
[15] Z.X. Huang, Extensions to the k-means algorithm for clustering large data sets with categorical values, Data Min. Knowl. Discov. 2 (3) (1998) 283–304.
[16] L. Hubert, P. Arabie, Comparing partitions, J. Classif. 2 (1) (1985) 193–218.
[17] N. Iam-On, T. Boongoen, Comparative study of matrix refinement approaches for ensemble clustering, Mach. Learn. 98 (1–2) (2015) 269–300.
[18] N. Iam-On, T. Boongoen, S. Garrett, A link-based approach to the cluster ensemble problem, IEEE Trans. Pattern Anal. Mach. Intell. 33 (12) (2011) 2396–2409.
[19] N. Iam-On, T. Boongoen, S. Garrett, A link-based cluster ensemble approach for categorical data clustering, IEEE Trans. Knowl. Data Eng. 24 (3) (2012) 413–425.
[20] G. Karypis, V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs, SIAM J. Sci. Comput. 20 (1) (1998) 359–392.
[21] L.T. Kuncheva, D.P. Vetrov, Evaluation of stability of k-means cluster ensembles with respect to random initialization, IEEE Trans. Pattern Anal. Mach. Intell. 28 (11) (2006) 1798–1808.
[22] T. Lee, J.A. Richards, P.H. Swain, Probabilistic and evidential approaches for multisource data analysis, IEEE Trans. Geosci. Remote Sens. 25 (3) (1987) 283–293.
[23] J.Q. Liang, Y.H. Han, Q.H. Hu, Semi-supervised image clustering with multi-modal information, Multimedia Syst. 22 (2) (2016) 149–160.
[24] J.Y. Liang, Z.Z. Shi, Y. LiD, M.J. Wierman, The information entropy, rough entropy and knowledge granulation in incomplete information system, Int. J. Gen. Syst. 35 (6) (2006) 641–654.
[25] G.P. Lin, J.Y. Liang, Y.H. Qian, An information fusion approach by combining multigranulation rough sets and evidence theory, Inf. Sci. 314 (1) (2015) 184–199.
[26] N. Nguyen, R. Caruana, Consensus clusterings, in: Proceedings of the Seventh IEEE International Conference on Data Mining, 2007, pp. 607–612. Omaha, NE
[27] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Syst. Man Cybern. 9 (1) (1979) 62–66.
[28] Z. Pawlak, Rough sets: theoretical aspects of reasoning about data, system theory, Knowledge Engineering and Problem Solving, 9, Kluwer, Dordrecht, 1991.
[29] Z. Pawlak, A. Skowron, Rudiments of rough sets, Inf. Sci. 177 (2007) 3–27.
[30] W. Pedrycz, Granular Computing: Analysis and Design of Intelligent Systems, CRC Press/Francis Taylor, Boca Raton, 2013.
[31] Y.H. Qian, J.Y. Liang, C.Y. Dang, Incomplete multigranulation rough set, IEEE Trans. Syst. Man Cybern. Part A 40 (2) (2010a) 420–431.
[32] Y.H. Qian, J.Y. Liang, W. Pedrycz, Positive approximation: an accelerator for attribute reduction in rough set theory, Artif. Intell. 174 (9) (2010b) 597–618.
[33] Y.H. Qian, J.Y. Liang, W.Z. Wu, C.Y. Dang, Information granularity in fuzzy binary GrC model, IEEE Trans. Fuzzy Syst. 19 (2) (2011) 253–264.
[34] Y.H. Qian, J.Y. Liang, Y.Y. Yao, C.Y. Dang, MGRS: a multi-granulation rough set, Inf. Sci. 180 (6) (2010) 949–970.
[35] G. Shafer, A Mathematical Theory of Evidence, Princeton University Press, Princeton, 1976.
[36] H. Shi, Y. Li, Y.H. Han, Q.H. Hu, Cluster structure preserving unsupervised feature selection for multi-view tasks, Neurocomputing 175 (2016) 686–697.
[37] A. Strehl, J. Ghosh, Cluster ensembles: a knowledge reuse framework for combining multiple partitions, J. Mach. Learn. Res. 3 (2002) 583–617.
[38] A.P. Topchy, A.K. Jain, W.F. Punch, Clustering ensembles: models of consensus and weak partitions, IEEE Trans. Pattern Anal. Mach. Intell. 27 (12) (2005) 1866–1881.
[39] J.V. Cleynenbreugel, S.A. Osinga, F. Fierens, P. Suetens, A. Oosterlinck, Road extraction from multi-temporal satellite images by an evidential reasoning approach, Pattern Recognit. Lett. 12 (1991) 371–380.
[40] R. Xu, D. Wunsch, Survey of clustering algorithms, IEEE Trans. Neural Netw. 16 (3) (2005) 645–678.
[41] R.R. Yager, D. Filev, Operations for granular computing: mixing words with numbers, in: Proceedings of 1998 IEEE International Conference on Fuzzy Systems, 1998, pp. 123–128. Anchorage, AK
[42] X.B. Yang, J.Y. Yang, C. Wu, D.J. Yu, Dominance-based rough set approach and knowledge reductions in incomplete ordered information system, Inf. Sci. 178 (4) (2008) 1219–1234.
[43] L.A. Zadeh, Advances in Fuzzy Set Theory and Applications, Elsevier Science Ltd, Amsterdam, 1979. November
[44] L.A. Zadeh, Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems, Soft Comput. 2 (1998) 23–25.
[45] C.M. Zhong, X.D. Yue, Z.H. Zhang, J.S. Lei, A clustering ensemble: two-level-refined co-association matrix with path-based transformation, Pattern Recognit. 48 (8) (2015) 2699–2709.
[46] 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 the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentinean, 2015, pp. 4112–4118.
[47] Z.H. Zhou, W. Tang, Clusterer ensemble, Knowl.-Based Syst. 19 (1) (2006) 77–83.
[48] P.F. Zhu, Q.H. Hu, W.M. Zuo, M. Yang, Multi-granularity distance metric learning via neighborhood granule margin maximization, Inf. Sci. 282 (2014) 321–331.