Information Fusion 61 (2020) 36–47

Contents lists available at ScienceDirect

Information Fusion

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

A multiple *k*-means clustering ensemble algorithm to find nonlinearly separable clusters

Liang Bai[a], Jiye Liang[a,*], Fuyuan Cao[b]

[a] Institute of Intelligent Information Processing, Shanxi University, Taiyuan, 030006, Shanxi, China
[b] School of Computer and Information Technology, Shanxi University, Taiyuan, 030006, Shanxi, China

ARTICLE INFO

Keywords:
Cluster ensemble
Nonlinearly separable clustering
k-means
Local hypothesis

ABSTRACT

Cluster ensemble is an important research content of ensemble learning, which is used to aggregate several base clusterings to generate a single output clustering with improved robustness and quality. Since clustering is unsupervised, where the "accuracy" does not have a clear meaning, most of existing ensemble methods try to obtain the most consistent clustering result with base clusterings. However, it is difficult for these methods to realize "Multi-weaks equal to a Strong". For example, on a data set with nonlinearly separable clusters, if the base clusterings are produced by some linear clusterers, these methods generally cannot integrate them to obtain a good nonlinear clustering. In this paper, we select *k*-means as a base clusterer and provide an ensemble clusterer (algorithm) of multiple *k*-means clusterings based on a local hypothesis. In the new algorithm, we study the extraction of the local-credible labels from a base clustering, the production of different base clusterings, the construction of cluster relation and the final assignment of each object. The proposed ensemble clusterer not only inherits the scalability of *k*-means but also overcomes its limitation that it only can find linearly separable clusters. Finally, the experimental results illustrate its effectiveness and efficiency.

1. Introduction

Clustering is an important problem in statistical multivariate analysis, data mining and machine learning [1]. The goal of clustering is to group a set of objects into clusters so that the objects in the same cluster are highly similar but remarkably dissimilar with objects in other clusters [2]. To tackle this problem, various types of clustering algorithms have been developed in the literature (e.g., [3] and references therein), including partitional, hierarchical, density-based and grid-based clustering and so on.

However, there is no single clustering algorithm that is suitable to deal with all the clustering tasks. Each algorithm has its own strengths and weaknesses. On a given data set, different algorithms or the same algorithms with different input parameters often have distinct clusterings. Therefore, it is very difficult for users to determine which clustering is suitable for a data set. Recently, the concept of "cluster ensemble" or "clustering aggregation" is emerged [4,5] to integrate several clusterings into a final clustering with improved robustness and quality. The cluster ensemble is seen as an unsupervised ensemble learning. In machine learning, ensemble learning is a very important research content, which trains multiple learners to solve the same problem. In contrast to ordinary machine learning approaches which try to learn one hypothesis from training data, ensemble methods try to construct a set of hypotheses and combine them to use [6]. In cluster ensemble, each base clustering is seen as a learning result of a base learner or clusterer. Cluster ensemble methods are used to combine these base clusterings to produce a final clustering. Currently, several types of cluster ensemble methods, such as pairwise similarity, graph-based, relabeling-based, and feature-based methods, have been developed as effective solutions [7]. They already have good theoretical and practical contributions. A detailed review of cluster ensemble methods can be found in Section 2.

Cluster ensemble is different from supervised ensemble learning, where the "accuracy" has a clear meaning. Take a classification problem for example. The label information Y on a training data set is used as a prior guidance to integrate multiple weak classifiers and help users to judge which objects on a data set are well performed by a weak classifier. However, it is very difficult for cluster ensemble to recognize the major strength and weakness of a base clustering on an unlabeled data set [8]. Therefore, the ensemble objective of most existing cluster ensemble methods is to obtain the most consistent clustering with all the base clusterings. Their ensemble results strongly depend on the qualities of base clusterings. Thus, they cannot realize "Multi-weaks equal to a strong". Take a nonlinear clustering problem for example. Fig. 1(a) shows the data distribution of a synthetic data set, called Flame [9], with two clusters which have different shapes. According to the figure, we can see that this data set cannot be linearly separable. Fig. 1(b) shows that

[*] Corresponding author.
E-mail addresses: bailiang@sxu.edu.cn (L. Bai), ljy@sxu.edu.cn (J. Liang), cfy@sxu.edu.cn (F. Cao).

https://doi.org/10.1016/j.inffus.2020.03.009
Received 23 July 2018; Received in revised form 6 January 2020; Accepted 22 March 2020
Available online 31 March 2020

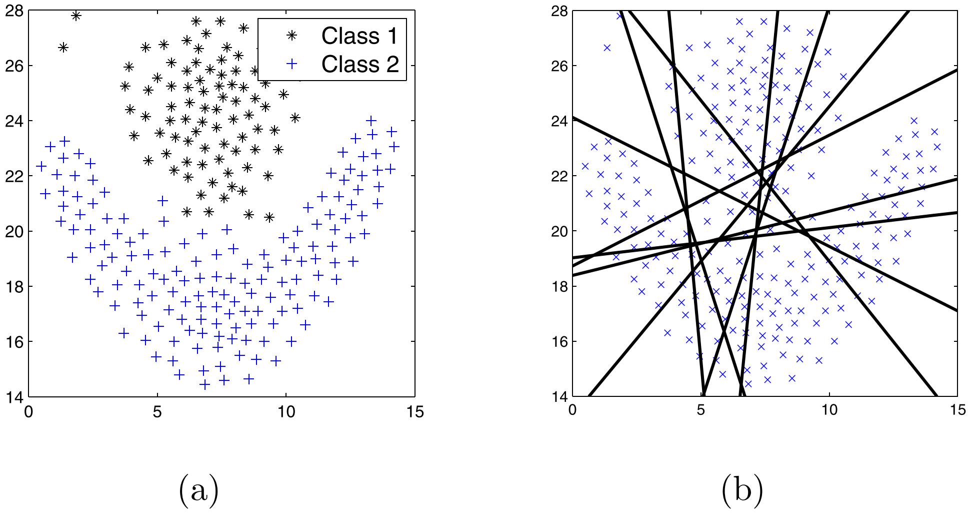


Fig. 1. (a) True class labels. (b) Multiple *k*-means clusterings.

the *k*-means algorithm [10] is employed to produce several base clusterings. We know that the *k*-means algorithm is a linear clusterer which is well-known for efficiency. It needs very low computing costs but is sensitive to data distributions [11].

If we do not consider the credibility of each label in these base clusterings, their most consistent result cannot recognize the nonlinearly separable clusters. This brings about a question: Why do not we directly use some nonlinear clustering algorithm? Indeed, currently, there are several nonlinear algorithms proposed in the literature. The representative methods include the spectral clustering [12,13] algorithms, the density-based spatial clustering of applications with noise (DBSCAN) [14], and the clustering by fast search and find of density peaks (CFSFDP) [15]. Although they can recognize clusters with any shapes, they need expensive time costs, i.e., the pairwise-objects distance calculations, which are not suitable for large-scale data sets. Compared to nonlinear clustering algorithms, linear clustering algorithms are generally efficient for dealing with large-scale data sets.

Therefore, it is a very key problem that how several linear clusterers are integrated to rapidly cluster data sets with different shapes, instead of a nonlinear clusterer. To solve the problem, we take *k*-means as a base clusterer and build an ensemble clusterer of multiple *k*-means clusterings to simulate a nonlinear clustering. The new algorithm need to address the following subproblems: (1) How to extract credible labels from a base clustering; (2) How to produce multiple different *k*-means clusterings to adequately describe the entire data; (3) How to build the relation between clusters to judge which clusters represent the same clusters; (4) How to determine the final label of each object. To solve these subproblems, we first assume that for a *k*-means clustering, the objects represented by a cluster center is credible in the local space. Based on the assumption, we propose a multiple *k*-means clustering algorithm with the local-credible constraint to produce multiple clusterings with different local-credible labels. Furthermore, we construct a relation graph for all the clusters from base clusterings based on the indirect overlap of their local-credible spaces. Finally, based on the label credibility function and relabeled base clusterings, we determine the final label of each object by maximizing the consistency of its labels. The main contributions of this paper are highlighted as follows.

- We define an evaluation function of cluster labels based on an local-credible assumption.
- We propose a multiple *k*-means clustering algorithm to rapidly solve the nonlinearly separable clustering problem.
- Experimental studies show the performance and scalability of the proposed algorithm for nonlinearly separable clustering.

The outline of the rest of this paper is as follows. Section 2 reviews the related work of the cluster ensemble problem. Section 3 presents an ensemble clusterer of multiple *k*-means clusterings. Section 4 demonstrates the performance of the proposed ensemble clusterer. Section 5 concludes the paper with some remarks.

2. Related work

Cluster ensemble, also called consensus clustering, is a kind of unsupervised ensemble learning. Currently, there are a large amount of literature on cluster ensemble. Generally speaking, cluster ensemble includes two major research tasks: (1) constructing a generator to produce a base clustering set Π and (2) devising an ensemble strategy to produce the final partition. Their results affect the performance of a cluster ensemble method. In the following, we will introduce the related work of the two tasks, respectively.

In ensemble learning, it is observed that the diversity among classification results of base classifiers or clusterers, to some extent, can enhance the performance of the ensemble learner. Currently, several heuristics have been proposed to produce different clusterings on a data set, which can be classified into three categories:

- Repeatedly run a single clustering algorithm with different initial sets of parameters to produce base clusterings [16–18]. Fred and Jain [16] applied *k*-means with the different numbers of clusters to produce a clustering set. Kuncheva and Vetrov [17] used *k*-means with randomly selected different cluster centers. Zhang et al. [18] run the spectral clustering algorithm with different kernel parameters.
- Run different types of clustering algorithms to produce base clusterings [5,19,20]. Gionis et al. [5] used several hierarchical clustering and *k*-means to produce a clustering set. Law et al. [19] applied multiple clustering algorithms with different objective functions as base clusterings and transformed a clustering ensemble problem as a multi-objective optimization. Yu et al. [20] studied how to integrate multiple types of fuzzy clusterings.
- Run one or more clustering algorithms on different subspaces or subsamples from a data set [21–23,23–29]. Fischer and Buhmann [21] applied the bootstrap method to obtain several data subsets. Fern and Brodley [25] used the random projection method to obtain several feature subspaces. Zhou et al. [26] used different kernel functions to describe the data. Yang et al. [29] proposed a novel hybrid sampling method for cluster ensemble by combining the strengths of boosting and bagging.

For ensemble strategy, there are several representative methods which can be classified into the following categories:

- *The pairwise similarity approach* that makes use of co-occurrence relationships between all pairs of data objects to aggregate multiple clusterings [30–34]. Fred and Jain [30] proposed an ensemble algorithm based on evidence accumulation and constructed a co-association (CO) matrix. Yang et al. [31] made use of clustering validity functions as weights to construct a weighted similarity matrix. Iam-On et al. [32,33] defined a link-based similarity matrix which sufficiently considers the similarity between clusters. Huang et al. [34] proposed an enhanced co-association (ECA) matrix, which is able to simultaneously capture the object-wise co-occurrence relationship as well as the multi-scale cluster-wise relationship in ensembles.
- *The graph-based approach* that expresses the base clustering information as an undirected graph and then derives the ensemble clustering via graph partitioning [4,35–38]. Strehl et al. [4] proposed three hypergraph ensemble algorithms CSPA, HGPA, and MCLA. Brodley et al. [35] proposed the HBGF algorithm where vertices represent both objects and clusters. Yu et al. [37] proposed using a distribution-based normalized hypergraph cut algorithm to generate the final clustering. This algorithm fully consider the representative of cluster structures of base clusterings and select appropriate cluster structures to participate in the cluster ensemble. Huang et al. [38] proposed a graph-based algorithm based on random walk to recognize uncertain links in cluster ensemble.
- *The relabeling-based approach* that expresses the base clustering information as label vectors and then aggregates via label alignment [22,23,39–41]. Its representative methods can be classified into two types: crisp label correspondence and soft label correspondence. The crisp methods [22,23,39] transfer the relabeling problem into a minimum cost one-to-one assignment problem. Long et al. [40] used an alternating optimization strategy to solve the soft label alignment problem.
- *The feature-based approach* that treats the problem of cluster ensemble as the clustering of categorical data [42–48]. Cristofor and Simovici [42] integrated the information theory and genetic algorithms to search for the most consistent clustering. Topchy et al. [43] proposed a probabilistic framework and used the EM algorithm for finding the consensus clustering. Nguyen et al. [46] made use of the *k*-modes [47] as the consensus function for cluster ensemble. In [48], we proposed an information-theoretical framework for cluster ensemble, which uses information entropy as a validity function to evaluate the effectiveness of cluster ensemble.
- *The semi-supervised approach* that makes use of few supervision information to enhance the effectiveness of the cluster ensemble. Representative works can be found in the literature [49–51]. Yu et al. sufficiently exploited the supervision information to deal with high-dimensional data.

Most existing algorithms mainly focus on how to obtain the most consistent clustering from base clusterings, which can improve the clustering quality and robustness. Since the base clusterings are not required to be from some particular clustering algorithm, they have good generalization, i.e., they could be applied to different situations of cluster ensemble. However, everything has two sides. Since their base clusterings may be from different types of clustering algorithms, they cannot recognize the credibility of each label. Thus, they do not easily integrate multiple “weak” clusterings to simulate a “strong” clustering. For example, if all the base clusterings are produced by linear clusterers, it is very difficult for them to produce a good nonlinear clustering. For the label credibility and fast nonlinear clustering problems, Huang et al. did some innovative works, as shown in the literature [38,52,53]. For example, they estimated the uncertainty of a cluster label [52] and the uncertainty between clusters [38], which fully considers the consensus of the cluster with respect to all the base clusterings. Furthermore, they proposed a fast approximation method for spectral clustering algorithm and use it as base clusterings to integrate multiple approximate result [53]. However, *It is worth noting that* the research objective of this paper is different from those of existing cluster ensemble algorithms. Our research object is specified as *k*-means. The aim of our ensemble clusterer is to integrate multiple *k*-means clusterings to simulate a nonlinear clustering and realize “Multi-weaks equal to a Strong”. The ensemble clusterer can overcome the limitation of *k*-means and rapidly discover nonlinearly separable clusters.

Table 1
Description of the main symbols used in this paper.

Symbol	Description
X	A data set
$\mathbf{x}_i$	The ith object in X
N	The number of objects in X
Π	A base clustering set
π_h	The hth base clustering
T	The number of base clusterings in Π
$T_{\max}$	The maximum number of base clusterings in Π
π^*	The final clustering
k	The final number of clusters
C_{hl}	The lth cluster in π_h
V	A set of all the cluster centers in Π
v_h	A set of all the cluster centers in π_h
$\mathbf{v}_{hl}$	The cluster centers of C_{hl}
$B(\mathbf{x}_i)$	The ε-neighborhood of $\mathbf{x}_i$
$d(\mathbf{x}_i, \mathbf{x}_j)$	The distance between $\mathbf{x}_i$ and $\mathbf{x}_j$
K	A set of the numbers of clusters in Π
k_h	The number of clusters in π_h
λ_h	The label credibility function of the hth clustering
E	A consensus objective function
Z	A objective function of producing base clusterings
Q	A objective function of the graph cuts problem
R	A relabeled base clustering set
G	A weighted graph of clusters
w_{xy}	The weight between cluster C_x and C_y
A	A set of all the cluster labels in Π
Ω	A partition of A in G
$L(C_x)$	The label of the subgraph which C_x belongs to

3. New cluster ensemble algorithm

3.1. Cluster ensemble problem

Let $X = \{\mathbf{x}_i\}_{i=1}^{N}$ be a set of N objects, $\Pi = \{\pi_h\}_{h=1}^{T}$ be a set of T base clusterings, $\pi_h = \{C_{hl}\}_{l=1}^{k_h}$ be the hth base clustering where k_h is the number of clusters and C_{hl} is the lth cluster in π_h, and $K = \{k_h\}_{h=1}^{T}$ be a set of the number of clusters in each base clustering. $\pi_h(\mathbf{x}_i)$ is the cluster label of object $\mathbf{x}_i$ in the clustering π_h. $\pi_h(\mathbf{x}_i) = l$ denotes that object $\mathbf{x}_i$ belongs to cluster C_{hl}. The cluster ensemble problem aims to finding out a final clustering π^* of data set X based on the clustering set Π. The main symbols used in this paper are summarized in Table 1.

In this paper, we select the *k*-means algorithm [10] as a base clusterer. Its objective function F is described as

$$F(\pi_h, v_h) = \sum_{l=1}^{k_h} \sum_{\pi_h(\mathbf{x}_i)=l, \mathbf{x}_i \in X} d(\mathbf{x}_i, \mathbf{v}_{hl})^2,$$

where $v_h = \{\mathbf{v}_{hl}\}_{l=1}^{k_h}$ and $\mathbf{v}_{hl}$ is the lth cluster center and $d(\mathbf{x}_i, \mathbf{v}_{hl}) = \sqrt{\|\mathbf{x}_i - \mathbf{v}_{hl}\|^2}$ is Euclidean distance between the object $\mathbf{x}_i$ and the center $\mathbf{v}_{hl}$ of the lth cluster. *k*-means makes use of alternatively updating π_h and v_h to solve the problem of minimizing F in finding cluster solutions. Its clustering results are often different, while it runs with different initial cluster centers. Therefore, we attempt to produce multiple base clusterings by *k*-means and integrate them to rapidly generate a good clustering result on data sets with nonlinearly separable clusters.

Given a base clustering set Π, we define the optimization problem of cluster ensemble as

$$\max_{\pi^*} \left[E(\pi^*) = \sum_{i=1}^{N} \sum_{h=1}^{T} \lambda_h(\mathbf{x}_i) I(\pi_h(\mathbf{x}_i) = \pi^*(\mathbf{x}_i)) \right], \tag{1}$$

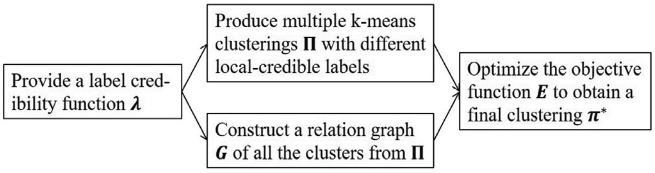

Provide a label credibility function λ
Produce multiple k-means clusterings Π with different local-credible labels
Construct a relation graph G of all the clusters from Π
Optimize the objective function E to obtain a final clustering π^*

Fig. 2. Solving process of the cluster ensemble problem.

where

- $I(.)$ is an indicator function which is used to measure the consistency of two labels. $I(.)$ takes 1 if two labels are equal, and 0 otherwise.
- $\lambda_h(.)$ is a boolean variable which is used to reflect the label credibility in the hth base clustering. If. is credible, $\lambda_h(.)$ takes 1, and 0 otherwise.

The objective function E is used to reflect the overall consistency between a final clustering π^* and base clusterings. We wish to maximize it to obtain the most consensus π^*. However, there are three important factors which often affect the effectiveness of the optimization problem as follows.

- *The credibility of each label*. In a base clustering, there are some objects whose labels are not correct. If these objects have consistently incorrect labels in the base clusterings, these labels are combined into the final clustering, which leads to reducing the effectiveness of ensemble. It is a key task for enhancing the ensemble effectiveness to provide an evaluation criterion for label credibility. Thus, in the objective function E, we use the variable $\lambda_h(.)$ to show the credibility of a label and reduce the effect of the incredible labels.
- *The difference among base clusterings*. In cluster ensemble, people wish each of base clusterings is different to some extent. The ensemble learning uses the difference to find out a robust clustering result. If most base clusterings in Π are very similar, it is not worth optimizing the objective function E. Thus, optimizing the objective function E is based on the difference of base clusterings. We wish to obtain multiple complementary k-means clusterings to adequately describe the entire data.
- *The relation of clusters*. Unlike classification, each base clustering may have a different representation of labels. Thus, we need to judge which cluster labels represent the same clusters. Obtaining a good relation of clusters is the prerequisite to optimize the objective function E. It is noted that the relation of clusters is different from that of most existing relabeling methods. Since the clusters from the same clustering also may represent the same cluster, the relation reflects all the clusters from the same and different base clusterings.

According to the above analysis, we see that a cluster ensemble problem is a multi-objective optimization problem. Before optimizing the objective function E, we need to solve several subproblems produced by the three factors. Fig. 2 summarizes a solving process of the clustering ensemble problem.

In the following, we will propose an ensemble clusterer of multiple k-means clusterings which can provide new solving methods for the above subproblems.

3.2. Label credibility function

In k-means, a cluster center is used to represent a cluster. However, if a cluster is nonlinearly separable with other clusters, the objects represented by a cluster center may come from different clusters. Take a clustering of k-means shown in Fig. 3 for example. We can see that Cluster 1 consists of objects from different "true" clusters. Thus, the cluster center obtained by k-means is not suitable to represent a nonlinear cluster. According to Fig. 3, we also can find that as the size of a local space represented by the cluster center is gradually reduced, the "true" cluster labels of objects in the local space are more consistent. Therefore, we evaluate the credibility of a cluster label based on a local hypothesis that the label of an object should be consistent with most of its neighbors. We assume that if the objects represented by a cluster center fall into its local space, they are thought to have credible labels. In this, the ε-neighborhood of the cluster center is seen as its local space.

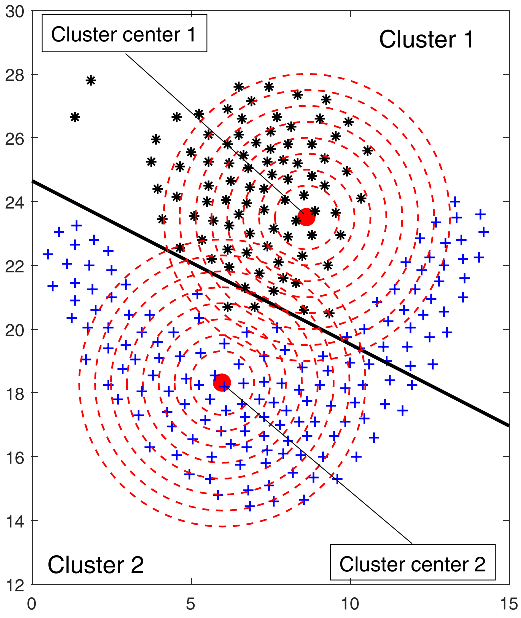


Fig. 3. A clustering of k-means.

Based on the local hypothesis, a label credibility function is formally defined as

$$\lambda_h(\mathbf{x}_i) = \begin{cases} 1, & \text{if} \quad \mathbf{x}_i \in B(\mathbf{v}_{hl}), \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

where $l = \pi_h(\mathbf{x}_i)$ and $B(\mathbf{v}_{hl}) = \{\mathbf{x}_j \in X | d(\mathbf{x}_j, \mathbf{v}_{hl}) \le \varepsilon\}$ is the ε-neighborhood of the cluster center $\mathbf{v}_{hl}$ which is also called as the local-credible space of the cluster C_{hl}, for $1 \le i \le N$ and $1 \le h \le T$. The definition shows that we only retain the label information of the objects in the ε-neighborhood of a cluster center.

3.3. Production of multiple base clusterings

Since the difference among base clusterings is a precondition for obtaining a good result of cluster ensemble. Therefore, in the following, we discuss how to obtain multiple k-means clusterings with different local-credible labels.

We first define an optimization problem of producing base clusterings as follows.

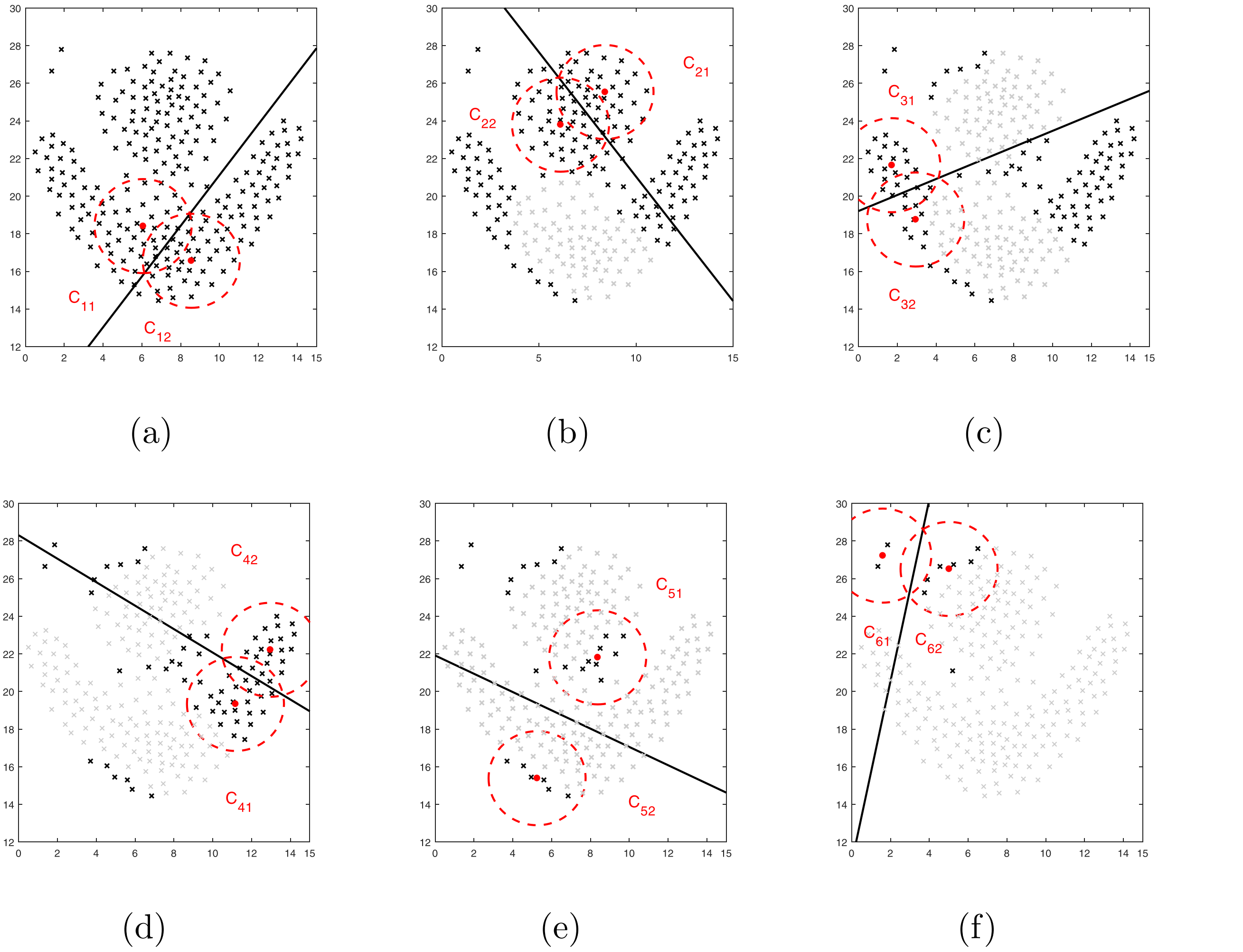
$$\min_{\Pi} \left[Z(\Pi) = \sum_{h=1}^{T} \sum_{i=1}^{N} \theta_h(\mathbf{x}_i)\lambda_h(\mathbf{x}_i) d(\mathbf{x}_i, \mathbf{v}_{h\pi_h(\mathbf{x}_i)}) \right], \tag{3}$$

subject to

$$\sum_{h=1}^{T} \theta_h(\mathbf{x}_i)\lambda_h(\mathbf{x}_i) = 1, 1 \le i \le N, \tag{4}$$

where $\theta_h(\mathbf{x}_i)$ is a boolean variable which takes 1 if object $\mathbf{x}_i$ plays a part in producing the hth base clustering, and 0 otherwise. It is used to control times each object plays a part. The constraint (4) requires each object to only once participate in producing a base clustering where it has a local-credible label. The aim of minimizing the objective function Z is to make the objects with the local-credible labels in a base clustering as different as possible from other base clusterings.

We propose an incremental learning method to solve the optimization problem. The method gradually produces multiple base clusterings



(a) (b) (c)

(d) (e) (f)

Fig. 4. Example about a running procedure of the MKM algorithm. (a) The 1st clustering. (b) The 2nd clustering. (c) The 3rd clustering. (d) The 4th clustering. (e) The 5th clustering. (f) The 6th clustering.

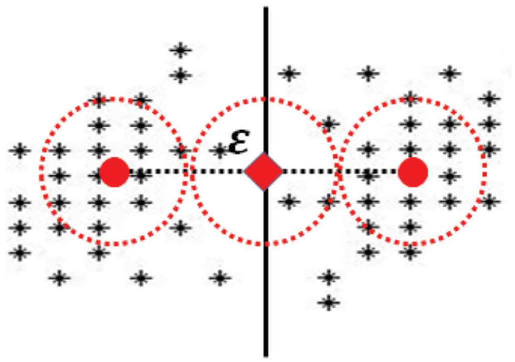


Fig. 5. A latent cluster between clusters.

by trying to optimize an incremental problem at each stage. The incremental problem is described as follows. Given $\Pi^{'}$ including the first gth obtained base clusterings $(0 \le g < T)$,

$$\min_{\pi_{g+1}} Z(\Pi^{'} \cup \{\pi_{g+1}\}), \tag{5}$$

subject to

$$\theta_{g+1 i} = \begin{cases} 1, & if \ \ \sum_{h=1}^{g} \lambda_h(\mathbf{x}_i) = 0, \\ 0, & otherwise, \end{cases} \tag{6}$$

for $1 \le i \le N$. According to the constraint (6), we see that the objects which do not have local-credible labels in $\Pi^{'}$ are required to play a part in producing the $g+1$th base clustering.

The incremental learning method, called the multiple k-means clustering (MKM) algorithm, proceeds as follows. We initially set $h = 1$, $\theta_h(\mathbf{x}_i) = 1$ for $1 \le i \le N$ and $S = X$. At each stage, we randomly select k_h objects as initial cluster centers from S and apply k-means with a constraint to cluster it. The constraint denotes that the cluster centers are updated by only considering the objects in their ϵ-neighborhoods, which makes the final obtained cluster centers better represent the objects in their local-credible spaces. After k-means runs, we assign each of objects in $X - S$ to the cluster represented by its nearest cluster center. Furthermore, we update $S = S - S'$, where S' is a set of the objects which have local-credible labels in the hth base clustering, $h = h + 1$ and $\theta_h(\mathbf{x}_i) = 1$ if $\mathbf{x}_i \in S$ and 0 otherwise, for $1 \le i \le N$. The above procedure is repeated until the number of the objects in S is less than k_h^2 or the number of base clusterings is equal to $T_{\max}$ which is the desired maximum number of base clusterings. The incremental procedure makes the final cluster centers obtained at each time represent different data subsets. Here, we need to explain why to set the end condition $|S| < k_h^2$. Many scholars pointed out in the literature [54,55] a rule of thumb that the maximum number of clusters on a set S of objects should be less than $\sqrt{|S|}$. Thus, while the number of objects in S is less than k_h^2, we assume that S cannot be partitioned into k_h clusters. In this case, although the number of clusterings maybe less than $T_{\max}$, we still terminate the iteration.

The formal description of the incremental method is shown in Algorithm 1. Next, let us continue taking the data set Flame for example to illustrate the running procedure of the MKM algorithm. We set $\varepsilon = 2.5$ and obtain six base clusterings on this data set. Fig. 3 shows the procedure of gradually producing these base clusterings. In these figures, gray objects indicate they do not play a part in producing a new base clustering. We see that these base clusterings have different local-credible labels, which is beneficial to cluster ensemble.

Algorithm 1: The MKM algorithm.

```
Input: $X$, $K$, $\varepsilon$, $T_{\max}$
Output: $\Pi$, $V$
Initialize $\Pi = \emptyset$, $V = \emptyset$, $S = X$, $h = 0$, and $\theta_{h+1}(\mathbf{x}_i) = 1$ for
$1 \leq i \leq N$;
while $|S| \geq k_h^2 \wedge h \leq T_{\max}$ do
    Set $F = 0$, $F' = 1$, $h = h + 1$ and $v_h$ is made up of randomly
    selected $k_h$ objects on $S$;
    while $F < F'$ do
        $F' = F$;
        for each $\mathbf{x}_i \in S$ do
            $\pi_h(\mathbf{x}_i) = \arg\min_{l=1}^{k_h} d(\mathbf{x}_i, \mathbf{v}_{hl})$;
        for $1 \leq l \leq k_h$ do
            $D = \{\pi_h(\mathbf{x}_i) = l \wedge \mathbf{x}_i \in B(\mathbf{v}_{hl}), \mathbf{x}_i \in S\}$;
            $\mathbf{v}_{hl} = \frac{\sum_{\mathbf{x}_i \in D} \mathbf{x}_i}{|D|}$;
        $F = \sum_{l=1}^{k_h} \sum_{\pi_h(\mathbf{x}_i)=l, \mathbf{x}_i \in S} d(\mathbf{x}_i, \mathbf{v}_{hl})^2$;
    for each $\mathbf{x}_i \in X - S$ do
        $\pi_h(\mathbf{x}_i) = \arg\min_{l=1}^{k_h} d(\mathbf{x}_i, \mathbf{v}_{hl})$;
    $S' = \{\lambda_h(\mathbf{x}_i) = 1, \mathbf{x}_i \in S\}$;
    for $i = 1 : N$ do
        if $\mathbf{x}_i \in S'$ then
            $\theta_{h+1}(\mathbf{x}_i) = 0$;
        else
            $\theta_{h+1}(\mathbf{x}_i) = \theta_h(\mathbf{x}_i)$;
    Update $\Pi = \Pi \bigcup \{\pi_h\}$, $V = V \bigcup v_h$, and $S = S - S'$;
```

The time complexity of the MKM algorithm is $O(N \sum_{h=1}^{T} t_h k_h)$, where t_h is the number of iterations of k-means in the process of producing the hth base clustering and T is the number of the produced base clusterings. The outputs of the algorithm are a base clustering set $\Pi = \{\pi_h, 1 \leq h \leq T\}$ and a cluster center set $V = \{v_h, 1 \leq h \leq T\}$. Note that if $T_{\max}$ is set to a very large value, T depends on the parameter ε. The T value generally increases as the ε value decreases. The main reason is that a small ε value indicates each base clustering includes few local-credible labels. Thus, in this case, we need more base clusterings to describe the entire data set. Therefore, how to set ε depends on the requirement of users. The users can regulate the parameter to control the number of base clusterings, according to own need.

3.4. Construction of cluster relation

Unlike classification where the class labels represent specific classes, the cluster labels only express grouping characteristics of the data and are not directly comparable across different clusterings in cluster analysis. Therefore, in cluster ensemble, the labels of different clusterings should be aligned. Besides, since the k-means algorithm only can recognize linearly separable clusters, two clusters from a base clustering may represent the same cluster. Therefore, we also need to analyze their relation.

Currently, there are several similarity or dissimilarity measures between clusters proposed in existing cluster ensemble algorithms [6]. Among these measures, the degree of overlap between two clusters, i.e., the number of their common objects, is widely used to reflect their similarity, which can be seen in the graph-based algorithms proposed by Strehl et al. [4] and the relabeling-based algorithms proposed by Zhou et al. [23]. However, this measure cannot be used to evaluate the similarity between clusters from the same clusterings, since they have no common objects. To solve the problem, Iam-On et al. [32] proposed a link-based similarity measure between clusters, which compares the overlap of them with other clusters. Although these existing measures already have good practical contributions, they do not consider the credibility of cluster labels. The objects with incredible labels generally affect the performance of these measures. Therefore, we need to design a new similarity measure to overcome the shortcoming.

According to the MKM algorithm, we know that the produced base clusterings Π are with different local-credible labels. Thus, we want to measure the overlap between the local-credible spaces of two clusters to reflect their similarity. Let C_{hl} and C_{gj} be two clusters, $\mathbf{v}_{hl}$ and $\mathbf{v}_{gj}$ be their cluster centers. If $d(\mathbf{v}_{hl}, \mathbf{v}_{gj})$ is no more than 2ε, their local-credible spaces are overlapped. However, for any two clusters, the overlap of their local-credible spaces is generally small or null, due to the producing mechanism of the base clusterings by the MKM algorithm. Therefore, we introduce a latent cluster to evaluate their "indirect" overlap. Next, we need to answer a question: How do we judge whether the local-credible spaces of two clusters are indirect overlapped? Let $\frac{\mathbf{v}_{hl}+\mathbf{v}_{gj}}{2}$ be the midpoint of the two centers $\mathbf{v}_{hl}$ and $\mathbf{v}_{gj}$. We assume there is a latent cluster C_z whose cluster center is $\frac{\mathbf{v}_{hl}+\mathbf{v}_{gj}}{2}$. If $d(\mathbf{v}_{hl}, \mathbf{v}_{gj})$ is no more than 4ε, the local-credible spaces of both the clusters C_{hl} and C_{gj} are overlapped with that of the latent cluster C_z, which can be seen in Fig. 5. In this case, the local-credible spaces of C_{hl} and C_{gj} are thought to be indirectly overlapped with respect to the latent cluster.

Furthermore, we consider the following two factors to measure the similarity between clusters C_{hl} and C_{gj} as follows.

- *The distance between their cluster centers.*
- *The number of objects in the local-credible space of the latent cluster.*

We know that the smaller $d(\mathbf{v}_{hl}, \mathbf{v}_{gj})$ is, the more overlapped the local-credible spaces between them and C_z are. Therefore, we think their similarity should be inversely proportional to $d(\mathbf{v}_{hl}, \mathbf{v}_{gj})$. Besides, since the k-means algorithm is a linear clusterer, the spaces of any two clusters are separated by the midline between their cluster centers. If the surrounding area of their midpoint includes few objects, they can be clearly distinguished.

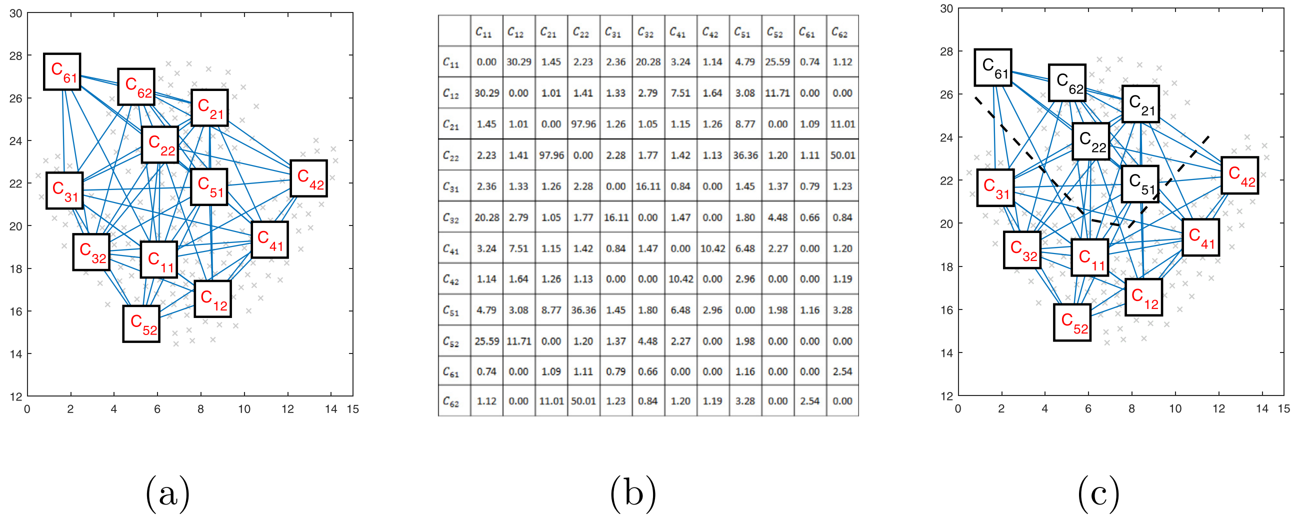
According to the above analysis, we think that their similarity should be proportional to the number of objects in the local-credible space of the latent cluster. Therefore, the similarity measure for two clusters is formally defined as follows.

$$\delta(C_{hl}, C_{gj}) = \begin{cases} \frac{|B(\frac{\mathbf{v}_{hl}+\mathbf{v}_{gj}}{2})|}{d(\mathbf{v}_{hl}, \mathbf{v}_{gj})}, & if\ d(\mathbf{v}_{hl}, \mathbf{v}_{gj}) \leq 4\varepsilon, \\ 0, & otherwise. \end{cases} \tag{7}$$

Based on the similarity measure, we construct a undirected and weighted graph $G =< A, W >$ to reflect the relation of these clusters. In the graph G, A is a set of vertices each representing a cluster label from Π. Thus, A is also seen as a set of all the cluster labels in Π. W is a weight set of edges between clusters. For any two clusters, we use their similarity as the weight of the edges between them, i.e., $w_{xy} = \delta(C_x, C_y)$, $x, y \in A$. The larger similarity they are, the more possibly they represent the same cluster.

After the weighted graph is obtained, the problem of constructing a cluster relation can be transferred to a normalized graph cuts problem which is described as follows [12].

$$\min_{\Omega} \left[Q(\Omega) = \frac{1}{k} \sum_{l=1}^{k} \frac{\sum_{x \in A_l, y \in A - A_l} w_{xy}}{\sum_{x \in A_l, z \in A} w_{xz}} \right], \tag{8}$$



	C_{11}	C_{12}	C_{21}	C_{22}	C_{31}	C_{32}	C_{41}	C_{42}	C_{51}	C_{52}	C_{61}	C_{62}
C_{11}	0.00	30.29	1.45	2.23	2.36	20.28	3.24	1.14	4.79	25.59	0.74	1.12
C_{12}	30.29	0.00	1.01	1.41	1.33	2.79	7.51	1.64	3.08	11.71	0.00	0.00
C_{21}	1.45	1.01	0.00	97.96	1.26	1.05	1.15	1.26	8.77	0.00	1.09	11.01
C_{22}	2.23	1.41	97.96	0.00	2.28	1.77	1.42	1.13	36.36	1.20	1.11	50.01
C_{31}	2.36	1.33	1.26	2.28	0.00	16.11	0.84	0.00	1.45	1.37	0.79	1.23
C_{32}	20.28	2.79	1.05	1.77	16.11	0.00	1.47	0.00	1.80	4.48	0.66	0.84
C_{41}	3.24	7.51	1.15	1.42	0.84	1.47	0.00	10.42	6.48	2.27	0.00	1.20
C_{42}	1.14	1.64	1.26	1.13	0.00	0.00	10.42	0.00	2.96	0.00	0.00	1.19
C_{51}	4.79	3.08	8.77	36.36	1.45	1.80	6.48	2.96	0.00	1.98	1.16	3.28
C_{52}	25.59	11.71	0.00	1.20	1.37	4.48	2.27	0.00	1.98	0.00	0.00	0.00
C_{61}	0.74	0.00	1.09	1.11	0.79	0.66	0.00	0.00	1.16	0.00	0.00	2.54
C_{62}	1.12	0.00	11.01	50.01	1.23	0.84	1.20	1.19	3.28	0.00	2.54	0.00

Fig. 6. Example about a procedure of constructing cluster relation. (a) A graph of cluster relation. (b) A pairwise-clusters similarity matrix. (c) A min-cut of graph.

where $\Omega = \{A_l\}_{l=1}^{k}$ is a partition of vertices in the graph G and A_l is the lth subset of A. we wish to obtain such a partition by minimizing the objective function Q that the vertices in the same subsets have very high similarity but are very dissimilar with vertices in other subsets. In order to solve the optimization problem, we apply the normalized spectral clustering (NSC) algorithm [13] to obtain a final partition of A. The vertices in the same subsets are used to represent a cluster. Thus, let $L(C_x)$ be the label of the subset which C_x belongs to, we have

$$L(C_x) = l,\ if\ C_x \in A_l, \tag{9}$$

for $1 \le l \le k$ and $x \in A$. The time complexity of constructing cluster relation is $O(N(\sum_{h=1}^{T} k_h)^2)$. Let us continue considering the example of the data set Flame to show a procedure of constructing cluster relation. In Fig. 4, the MKM algorithm produces 12 clusters. Fig. 6(a) and (b) show their relation graph and their similarity matrix, respectively. We employ the NSC algorithm to obtain a min-cut of this graph which is shown in Fig. 6(c). All the clusters in each subgraph are used to represent the same cluster.

3.5. Generation of final clustering

After relabeling the clusters from base clusterings, Π can be transformed into a relabeled base clustering set R as follows.

$$R_h(\mathbf{x}_i) = L(C_{h\pi_h(\mathbf{x}_i)}), \tag{10}$$

for $1 \le i \le N$ and $1 \le h \le T$. Given R, the consensus function E can be rewritten as follows.

$$E(\pi^*) = \sum_{i=1}^{N}\sum_{h=1}^{T} \lambda_h(\mathbf{x}_i) I(R_h(\mathbf{x}_i) = \pi^*(\mathbf{x}_i)). \tag{11}$$

We can maximize the objective function E by the following equation

$$\pi^*(\mathbf{x}_i) = \arg\max_{l=1}^{k} |\{\lambda_h(\mathbf{x}_i) R_h(\mathbf{x}_i) = l, 1 \le h \le T\}|, \tag{12}$$

for $1 \le i \le N$. The time complexity of generating the final clustering is $O(NT)$.

3.6. Overall implementation

We integrate the above steps to form a new multiple k-means clustering ensemble (KMCE) algorithm. This algorithm is described in Algorithm 2 . The overall time complexity of the KMCE algorithm is $O(N\sum_{h=1}^{T} t_h k_h + N\sum_{h=1}^{T} k_h + N(\sum_{h=1}^{T} k_h)^2 + NT)$. We see that the time complexity is linear with the number of objects. Generally, $\left(\sum_{h=1}^{T} k_h\right)^2 \ll N$. In this case, the time complexity is less than $O(N^2)$. We know that the time complexities of most nonlinear clustering algorithms are no less than $O(N^2)$. This indicates that the KMCE algorithm is suitable to deal with large-scale data sets, compared to other nonlinear clustering algorithms.

Algorithm 2: The KMCE algorithm.

```
Input: $X$, $k$, $K$, $\varepsilon$, $T_{\max}$
Output: $\pi^*$
$\Pi = \arg\min Z(\Pi)$ by Algorithm 1;
for $i = 1 : N$ do
    for $h = 1 : |\Pi|$ do
        Compute $\lambda_h(\mathbf{x}_i)$ by Eq.~(2);
$A$ = a set including all the cluster labels in Π;
for $x, y \in A$ do
    $w_{xy} = \delta(C_x, C_y)$;
Obtain a graph $G =< A, W >$ where $W = \{w_{xy}\}_{x,y \in A}$;
$\Omega = \arg\min Q(\Omega)$ by the NSC algorithm;
Obtain the relabeled base clustering set $R$ by Eq.~(10);
$\pi^* = \arg\max E(\pi^*)$ by Eq.~(12);
```

4. Experimental analysis

In this section, we carry out the KMCE algorithm on 5 synthetic and 6 real data sets and evaluate its effectiveness by two validity measures and time costs.

4.1. Data sets

Table 2 shows the details of these tested data sets. The data distributions of the synthetic data sets are shown in Fig. 7. The synthetic and real data sets are downloaded from https://github.com/deric/clustering-benchmark and http://www.ics.uci.edu/mlearn/MLRepository.html, respectively.

4.2. Evaluation criteria

We employ the two widely-used external criteria ARI [56] and NMI [57] to measure the similarity between the clustering result and the true

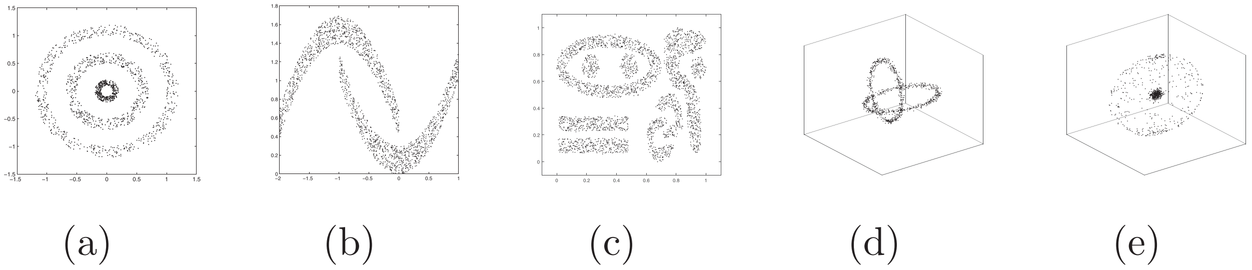


Fig. 7. Data distribution of synthetic data. (a) Ring. (b) Banana. (c) Complex. (d) Chainlink. (e) Atom.

Table 2
Description of data sets: Number of Data Objects (N), Number of Dimensions (D), Number of Clusters (k).

	Data set	N	D	k
Synthetic data	Ring	1500	2	3
	Banana	2000	2	2
	Complex	3031	2	9
	Chainlink	1000	3	2
	Atom	800	3	2
Real data	Iris	150	4	3
	Wine	178	13	3
	Breast	569	30	2
	Digits	5620	63	10
	Statlog	6435	36	7
	KDD99	1,048,576	39	2

Table 3
Notation for the contingency table for comparing two partitions.

C\P	p_1	p_2	$\cdots$	$p_{k'}$	*Sums*
c_1	n_{11}	n_{12}	$\cdots$	$n_{1k'}$	b_1
c_2	n_{21}	n_{22}	$\cdots$	$n_{2k'}$	b_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
c_k	n_{k1}	n_{k2}	$\cdots$	$n_{kk'}$	b_k
Sums	d_1	d_2	$\cdots$	$d_{k'}$	

partition on a data set. Given a data set X with N objects and two partitions of these objects, namely $C = \{c_1, c_2, \ldots, c_k\}$ (the clustering result) and $P = \{p_1, p_2, \ldots, p_{k'}\}$ (the true partition), the overlappings between C and P can be summarized in a contingency table (Table 3) where n_{ij} denotes the number of common nodes of groups c_i and p_j: $n_{ij} = |c_i \cap p_j|$.

The adjusted rand index [56] is defined as

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - [\sum_i \binom{b_i}{2}\sum_j \binom{d_j}{2}]/\binom{N}{2}}{\frac{1}{2}[\sum_i \binom{b_i}{2} + \sum_j \binom{d_j}{2}] - [\sum_i \binom{b_i}{2}\sum_j \binom{d_j}{2}]/\binom{N}{2}}$$

where n_{ij}, b_i, d_j are values from the contingency table (Table 3). The normalized mutual information (NMI) [57] is defined as

$$NMI = \frac{2\sum_i \sum_j n_{ij} \log \frac{n_{ij} N}{b_i d_j}}{-\sum_i b_i \log \frac{b_i}{N} - \sum_j d_j \log \frac{d_j}{N}}.$$

If a clustering result is close to the true partition, then its ARI and NMI values are high.

4.3. Compared methods

In order to properly examine the performance of the proposed algorithm, we compare it with the following cluster ensemble algorithms. The codes of these compared algorithms are open and accessible, which can be found from the personal homepage of these authors.

- *Pairwise similarity algorithms* include the co-association similarity matrix (CO) proposed by Fred and Jain [16] and the three link-based similarity matrices WCT, WTQ and CSM proposed by Iam-On et al.[32]. The single-link (SL) and the average-link (AL) algorithms are used to derive the final solution.
- *Graph-based algorithms* include the cluster-based similarity partitioning algorithm (CSPA), hyper graph partitioning algorithm (HGPA) and meta-clustering (MCLA) algorithm proposed by Strehl and Ghosh [4].
- *Relabeling-based algorithms* include the selectively un-weighted and weighted ensemble algorithms SV and SWV proposed by Zhou and Tang [23].
- *Feature-based algorithms* include the expectation maximization (EM) algorithm for cluster ensemble proposed by Topchy et al. [43] and the iterative voting consensus (IVC) algorithm proposed by Nguyen et al. [46].

Besides, we compare KMCE with three nonlinear clustering algorithms including the normalized spectral clustering algorithm (NSC) [13], the density-based spatial clustering of applications with noise (DBSCAN) [14] and the clustering by fast search and find of density peaks (CFSFDP) [15]. The aim of the comparison is to show the simulation of KMCE for nonlinear clustering.

4.4. Experimental settings

To ensure that the comparisons are in a uniform environmental condition, several settings of these compared algorithms are listed as follows.

- For existing cluster ensemble algorithms, we run k-means T times, each with a random and different initialization of cluster centers, to produce base clusterings on a data set. The number of clusters k_h in each base clustering is equal to the true number of classes on each of the given data sets. For the parameter T, we test each of these algorithms with $T = 10, 20, 30, 40, 50$ respectively, and select the highest ARI and NMI values for comparison. For other parameters of these algorithms, we set them, according to the suggestions of the authors.
- The DBSCAN, CFSFDP and KMCE algorithms are required to input the parameter ε. We estimate the ε value by using $\bar{d} = \frac{1}{n}\sum_{i=1}^{nd}(\mathbf{x}_i, =\mathbf{x})$ where $=\mathbf{x} = \sum_{j=1}^{n}\frac{\mathbf{x}_j}{n}$. However, each of these algorithms may need different ε values on each data set. Thus, we test each of these algorithms with 10 different values, i.e., $\varepsilon = \bar{d}, \bar{d}/2, \bar{d}/3, \bar{d}/4, \bar{d}/5, \bar{d}/6, \bar{d}/7, \bar{d}/8, \bar{d}/9$, and $\bar{d}/10$ and select the highest ARI and NMI values on each data set for comparison. However, different from DBSCAN and CFSFDP, the KMCE algorithm has a certain randomness. Therefore, we need to run the KMCE algorithm 50 times on each data set and compute the average ARI and NMI values for comparison. Besides, we set $T_{\max} = 50$ for the KMCE algorithm.
- For the NSC algorithm, we use Gaussian kernel to obtain a pairwise-objects similarity matrix and set the kernel parameter δ^2 in the interval [0.1,2] with the step size as 0.1. In these parameters, we select the highest ARI and NMI values for comparison.

Table 4
ARI measures of different methods on synthetic data sets.

Methods	Synthetic data sets				
	Ring	Banana	Complex	Chainlink	Atom
CO-SL	0.5002	0.5039	0.6267	0.0927	0.1456
CO-AL	0.1305	0.5039	0.3726	0.0927	0.1456
WCT-SL	0.0259	0.5039	0.6268	0.0927	0.1456
WCT-AL	0.1382	0.5039	0.3635	0.0927	0.1456
WTQ-SL	0.4115	0.5039	0.6158	0.0927	0.1456
WTQ-AL	0.1389	0.5039	0.3705	0.0927	0.1456
CSM-AL	0.0046	0.5039	0.5878	0.0927	0.1456
CSM-SL	0.1448	0.5039	0.4199	0.0927	0.1456
CSPA	0.3163	0.4926	0.3418	0.0927	0.0021
HGPA	0.0004	-0.0004	0.1966	-0.0010	-0.0013
MCLA	0.0004	0.5039	0.3736	0.0927	0.1554
SV	0.0847	0.5039	0.1406	0.1002	0.1736
SWV	0.1809	0.5039	0.1966	0.1002	0.1736
EM	0.0302	0.0031	0.3240	0.0896	0.2617
IVC	0.3231	0.5039	0.4097	0.0927	0.1178
NSC	**1.0000**	**1.0000**	0.9848	**1.0000**	**1.0000**
DBSCAN	**1.0000**	**1.0000**	0.8513	0.4947	0.3786
CFSFDP	0.3227	**1.0000**	0.8043	0.6853	0.4154
KMCE	**1.0000**	**1.0000**	**0.9879**	**1.0000**	**1.0000**

Table 5
ARI measures of different methods on real data sets.

Methods	Real data sets					
	Iris	Wine	Breast	Digits	Statog	KDD99
CO-SL	0.7302	0.8471	0.7302	0.1651	0.3248	0.9584
CO-AL	0.7302	0.8471	0.7302	0.6050	0.5700	0.9584
WCT-SL	0.7302	0.8471	0.7302	0.1047	0.4101	0.9584
WCT-AL	0.7302	0.8471	0.7302	0.6046	0.5699	0.9584
WTQ-SL	0.7302	0.8471	0.7302	0.1651	0.4101	0.9584
WTQ-AL	0.7302	0.8471	0.7302	0.6049	0.5699	0.9584
CSM-AL	0.7302	0.8471	0.7302	0.0000	0.4101	0.9584
CSM-SL	0.7302	0.8471	0.7302	0.6146	0.5699	0.9584
CSPA	0.6521	0.7808	0.3414	0.7573	0.4329	0.9370
HGPA	0.1026	0.1286	-0.0007	0.3750	0.2619	-0.0005
MCLA	0.7302	0.8471	0.7302	0.6935	0.5127	0.9584
SV	0.0067	0.8685	0.7302	0.3244	0.4533	0.9584
SWV	0.0002	0.8685	0.7302	0.4641	0.4546	0.9584
EM	0.6008	0.7855	0.6328	0.6205	0.5074	0.7652
IVC	0.5970	0.6875	0.0487	0.6006	0.4188	0.7425
NSC	0.7455	**0.9310**	0.7493	0.7536	0.5308	0.9604
DBSCAN	0.5162	0.3587	0.0478	0.5052	0.4319	**0.9793**
CFSFDP	0.7028	0.7414	0.7305	0.7584	0.4963	0.9604
KMCE	**0.7565**	0.8687	**0.7700**	**0.7841**	**0.6211**	0.9715

4.5. Experimental results

4.5.1. Performance analysis

We first test these algorithms on the given data sets to compare their clustering accuracies. Due to the fact that the KDD-CUP'99 data set is very large, some algorithms cannot be implemented on the entire data set. Therefore, we sample a subset which includes 5000 normal-connected and 5000 abnormal-connected records from the data set for the accuracy comparison.

Tables 4 –7 show the ARI and NMI values of different algorithms on synthetic and real data sets, respectively. According to these tables, we see that the clustering accuracies of the KMCE algorithm are obviously superior to other cluster ensemble algorithms on these synthetic data sets. The experimental results conclude that: (1) While clustering nonlinearly separable data sets, the base clusterings produced by k-means include lots of incredible labels. However, since the existing ensemble algorithms do not evaluate these label credibilities, they cannot integrate them to recognize nonlinear clusters. (2) The proposed ensemble algorithm can effectively discover nonlinearly separable clusters and improve the performance of the k-means algorithm. On the real data sets, the KMCE algorithm also has better performance, compared to other cluster ensemble algorithms.

Besides, these tables also show the comparison results of the KMCE algorithm with three nonlinear clustering algorithms on the given data sets. We can see that the clustering validity of the KMCE algorithm is superior or close to the best results of these algorithms. The experiments tell us that the proposed algorithm can well simulate nonlinear clustering results.

Due to the fact that the KMCE algorithm has a certain randomness, we test it 50 times on each data sets. Tables 8 and 9 show the standard deviation (std) of the ARI and NMI values for its 50 clustering results. We can see that the std value is less than 0.1 on each data set. This indicates that the randomness has limited impact on the performance of the KMCE algorithm.

Furthermore, we compare the efficiency of the KMCE algorithm with these nonlinear algorithms on the KDD-CUP'99 data set. In the experiment, we fix $k = 2$ and $\varepsilon = 0.14$. Fig. 8 shows the running time of these algorithms with different numbers of objects. We can see that the proposed algorithm is very efficient, compared to other algorithms. This indicates that the KMCE algorithm is a good choice for clustering large-scale data sets.

Table 6
NMI measures of different methods on synthetic data sets.

Methods	Synthetic data sets				
	Ring	Banana	Complex	Chainlink	Atom
CO-SL	0.6948	0.4035	0.6888	0.0686	0.2631
CO-AL	0.2112	0.4035	0.6343	0.0686	0.2631
WCT-SL	0.1207	0.4035	0.6887	0.0686	0.2631
WCT-AL	0.2162	0.4035	0.6302	0.0686	0.2631
WTQ-SL	0.5407	0.4035	0.6781	0.0686	0.2631
WTQ-AL	0.2174	0.4035	0.6370	0.0686	0.2631
CSM-AL	0.0218	0.4035	0.7166	0.0686	0.2631
CSM-SL	0.2211	0.4035	0.6630	0.0686	0.2631
CSPA	0.3785	0.3927	0.6071	0.0686	0.0024
HGPA	0.0008	0.0000	0.3656	0.0000	0.0000
MCLA	0.0013	0.4035	0.6334	0.0686	0.2713
SV	0.1758	0.4035	0.2049	0.0743	0.2863
SWV	0.2487	0.4035	0.4339	0.0743	0.2863
EM	0.1495	0.0042	0.5730	0.0663	0.3404
IVC	0.3813	0.4035	0.6467	0.0686	0.1942
NSC	**1.0000**	**1.0000**	0.9853	**1.0000**	**1.0000**
DBSCAN	**1.0000**	**1.0000**	0.8719	0.4828	0.2773
CFSFDP	0.3792	**1.0000**	0.8451	0.6544	0.4592
MKCE	**1.0000**	**1.0000**	**0.9892**	**1.0000**	**1.0000**

Table 7
NMI measures of different methods on real data sets.

Methods	Real data sets					
	Iris	Wine	Breast	Digits	Statog	KDD99
CO-SL	0.7582	0.8347	0.6231	0.5145	0.5263	0.9263
CO-AL	0.7582	0.8347	0.6231	0.7307	0.6322	0.9263
WCT-SL	0.7582	0.8347	0.6231	0.4119	0.5526	0.9263
WCT-AL	0.7582	0.8347	0.6231	0.7305	0.6321	0.9263
WTQ-SL	0.7582	0.8347	0.6231	0.5145	0.5526	0.9263
WTQ-AL	0.7582	0.8347	0.6231	0.7306	0.6321	0.9263
CSM-AL	0.7582	0.8347	0.6231	0.0032	0.5526	0.9263
CSM-SL	0.7582	0.8347	0.6231	0.7309	0.6321	0.9263
CSPA	0.6803	0.7771	0.2981	0.7857	0.5425	0.8816
HGPA	0.1609	0.1705	0.0007	0.4932	0.326	0.0000
MCLA	0.7582	0.8347	0.6231	0.7627	0.5903	0.9263
SV	0.0183	0.8529	0.6231	0.3782	0.4481	0.9263
SWV	0.0110	0.8529	0.6231	0.6085	0.5248	0.9263
EM	0.6727	0.7980	0.5400	0.7271	0.5837	0.7388
IVC	0.6801	0.7281	0.0415	0.7208	0.5256	0.7425
NSC	0.7980	**0.9016**	0.6328	0.8119	0.6243	0.9291
DBSCAN	0.5904	0.4451	0.0303	0.7163	0.5021	**0.9584**
CFSFDP	0.7277	0.7528	0.6152	**0.8645**	0.5644	0.9291
MKCE	**0.8042**	0.8542	**0.6667**	0.8593	**0.6646**	0.9466

Table 8
Standard deviation of the KMCE algorithm for the ARI and NMI measures on synthetic data sets.

Indices	Synthetic data sets				
	Ring	Banana	Complex	Chainlink	Atom
ARI(std)	0.0000	0.0000	0.0121	0.0000	0.0000
NMI(std)	0.0000	0.0000	0.0127	0.0000	0.0000

Table 9
Standard deviation of the KMCE algorithm for the ARI and NMI measures on real data sets.

Indices	Real data sets					
	Iris	Wine	Breast	Digits	Statog	KDD99
ARI(std)	0.0993	0.0897	0.0670	0.0531	0.013	0.0170
NMI(std)	0.8071	0.0689	0.0646	0.0269	0.008	0.0289

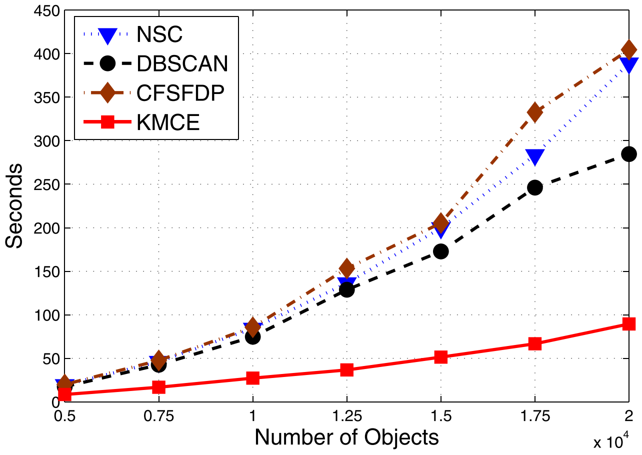
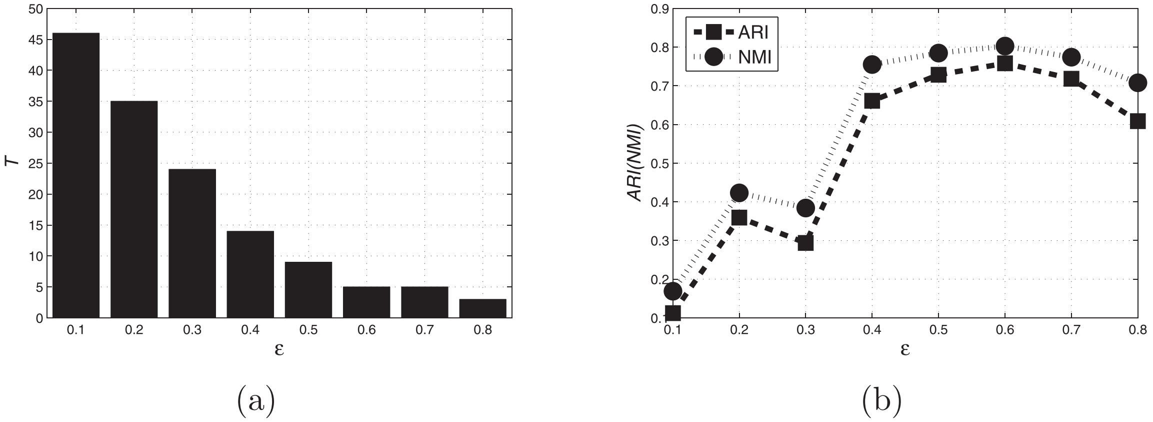


Fig. 8. Time comparison on the KDD99 data set.

4.5.2. Parameter analysis

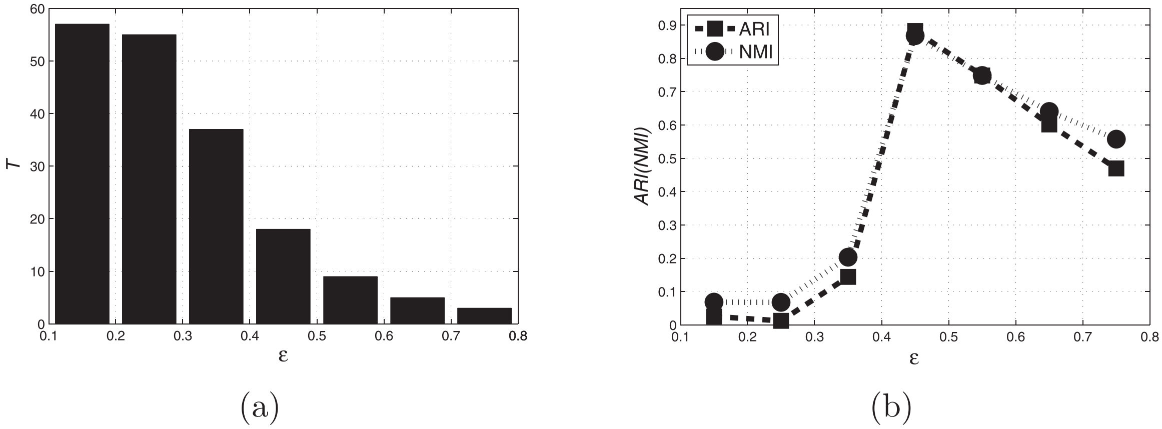
In this part, we analyze the effect of the parameter ε on the performance of the KMCE algorithm by the experiments. We know that the number T of base clusterings depends on the selection of the parameters ε and $T_{\max}$. Thus, we set $T_{\max} = 1000$, which reduces the effect of $T_{\max}$ on T and makes the MKM algorithm produce as many base clusterings as possible. We take the iris and wine data for example. According to Figs. 9(a) and 10(a), we see that the number of the base clusterings produced by the MKM algorithm decreases as the ε value increases. However, Figs. 9(b) and 10(b) illustrate that the clustering accuracy does not increase, after the ε value is growing to a certain extent. This experimental result tells us that the number of the base clusterings is too large or small to obtain a good ensemble result. Thus, we should select a suitable value of ε to control the number of base clusterings on each data set. It is an important issue for many nonlinear algorithms including KMCE to select the parameter value. However, there are few theoretical guidelines for setting the parameter. We wish to further study the problem in next research work. In this paper, we provide a rule of thumb that the parameter value is selected from the interval $[\bar{d}/10, \bar{d}]$ where $\bar{d}$ is the average distance between each object and the center of a data set. We tested the DBSCAN, CFSFDP, and KMCE with different parameter values on the given data sets. We found that these algorithms can obtain better clustering results if the parameter is selected from the interval.



(a)

(b)

Fig. 9. Effect of the parameter ε on the iris data. (a) The numbers T of produced base clusterings with respect to different values of ε. (b) The ARI and NMI values of ensemble results with respect to different values of ε.



(a)

(b)

Fig. 10. Effect of the parameter ε on the wine data. (a) The numbers T of produced base clusterings with respect to different values of ε. (b) The ARI and NMI values of ensemble results with respect to different values of ε.

5. Conclusions

K-means is a widely-used clustering algorithm for its low computational cost. However, it is a linear clusterer and its performance tends to be affected by data distributions. In this paper, we have proposed a new cluster ensemble algorithm by using multiple *k*-means, which is called KMCE. The new algorithm includes four main steps: producing multiple k-means clusterings, evaluating the local credibility of each label, building the relation between clusters, and generating the final clustering. It improves the robustness and quality of *k*-means and can rapidly recognize nonlinearly separable clusters. In the experimental analysis, we have compared the KMCE algorithm with eleven existing cluster ensemble algorithms and three nonlinear clustering algorithms on synthetic and real data sets. The comparison results have illustrated that the performance of the proposed algorithm is very effective. Furthermore, we have analyzed the efficiency of the KMCE algorithm which is suitable to deal with large-scale data sets.

This paper mainly focused on cluster ensemble of *k*-means. For future research, we would like to investigate the label credibility of different clustering algorithms. Furthermore, we plan to propose a general cluster ensemble framework for fast nonlinearly separable clustering.

Declaration of Competing Interest

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

CRediT authorship contribution statement

Liang Bai: Writing - original draft, Methodology, Conceptualization, Methodology, Writing - original draft. **Jiye Liang:** Writing - review & editing, Supervision, Project administration, Funding acquisition. **Fuyuan Cao:** Writing - review & editing.

Acknowledgement

The authors are very grateful to the editors and reviewers for their valuable comments and suggestions. This work is supported by the National Natural Science Foundation of China (Nos. 61773247, 61876103, 61976128, 61902227), the Technology Research Development Projects of Shanxi (No. 201901D211192) and the 1331 Engineering Project of Shanxi Province, China.

References

[1] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, 2001.

[2] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice Hall, 1988.

[3] A.K. Jain, Data clustering: 50 years beyond k-means, Pattern Recognit. Lett. 31 (8) (2010) 651–666.

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

[5] A. Gionis, H. Mannila, P. Tsaparas, Clustering aggregation, ACM Trans. Knowl. Discov. Data 1 (1) (2007) 1–30.

[6] Z. Zhou, Ensemble Methods: Foundations and Algorithms, CRC Press, 2012.

[7] N. Iam-On, T. Boongoen, Comparative study of matrix refinement approaches for ensemble clustering, Mach. Learn. 98 (2015) 269–300.

[8] E. Gonzalez, J. Turmo, Unsupervised ensemble minority clustering, Mach. Learn. 98 (2015) 217–268.

[9] L. Fu, E. Medico, A novel fuzzy clustering method for the analysis of DNA microarray data, BMC Bioinformatics 8 (1) (2007) 3.

[10] J.B. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proc. of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, 1967, pp. 281–297. 1

[11] H. Xiong, J.J. Wu, J. Chen, k-means clustering versus validation measures: adata-distribution perspective, IEEE Trans. Syst. Man Cybern. Part B 39 (2) (2009) 318–331.

[12] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 888–905.

[13] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm, in: T.G. Dietterich, S. Becker, Z. Ghahramani (Eds.), Advances in Neural Information Processing Systems, vol. 14, MIT Press, Cambridge, MA, 2002.

[14] M. Ester, H. Kriegel, J. Sander, X. Xu, U.M. Fayyad, A density-based algorithm for discovering clusters in large spatial databases with noise, in: E. Simoudis, J. Han (Eds.), Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), AAAI Press, 1996, pp. 226–231.

[15] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (6191) (2014) 1492–1496.

[16] A. Fred, A. Jain, Combining multiple clusterings using evidence accumulation, IEEE Trans. Pattern Anal. Mach. Intell. 27 (6) (2005) 835–850.

[17] L. Kuncheva, D. 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.

[18] X. Zhang, L. Jiao, F. Liu, L. Bo, M. Gong, Spectral clustering ensemble applied to SAR image segmentation, IEEE Trans. Geosci. Remote Sens. 46 (7) (2008) 2126–2136.

[19] M. Law, A. Topchy, A. Jain, Multiobjective data clustering, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2004.

[20] Z. Yu, H. Chen, J. You, Hybrid fuzzy cluster ensemble framework for tumor clustering from bio-molecular data, IEEE/ACM Trans. Comput. Biol. Bioinf. 10 (3) (2013) 657–670.

[21] B. Fischer, J. Buhmann, Bagging for path-based clustering, IEEE Trans. Pattern Anal. Mach. Intell. 25 (11) (2003) 1411–1415.

[22] A. Topchy, B. Minaei-Bidgoli, A. Jain, Adaptive clustering ensembles, in: Proc. the 17th International Conference on Pattern Recognition, 2004.

[23] Z. Zhou, W. Tang, Clusterer ensemble, Knowl. Based Syst. 19 (1) (2006) 77–83.

[24] Y. Hong, S. Kwong, H. Wang, Q. Ren, Resampling-based selective clustering ensembles, Pattern Recognit. Lett. 41 (9) (2009) 2742–2756.

[25] X. Fern, C. Brodley, Random projection for high dimensional data clustering: a cluster ensemble approach, in: Proc. International Conference on Machine Learning, 2003.

[26] P. Zhou, L. Du, L. Shi, H. Wang, et al., Learning a robust consensus matrix for clustering ensemble via Kullback-Leibler divergence minimization, in: Proc. the 25th International Joint Conference on Artificial Intelligence, 2015.

[27] Z. Yu, L. Li, J. Liu, et al., Adaptive noise immune cluster ensemble using affinity propagation, IEEE Trans. Knowl. Data Eng. 27 (19) (2015) 3176–3189.

[28] F. Gullo, C. Domeniconi, Metacluster-based projective clustering ensembles, Mach. Learn. 98 (1–2) (2013) 1–36.

[29] Y. Yang, J. Jiang, Hybrid sampling-based clustering ensemble with global and local constitutions, IEEE Trans. Neural Netw. Learn. Syst. 27 (5) (2016) 952–965.

[30] A. Fred, A.K. Jain, Data clustering using evidence accumulation, in: Proc. the 16th International Conference on Pattern Recognition, 2002, pp. 276–280.

[31] Y. Yang, K. Chen, Temporal data clustering via weighted clustering ensemble with different representations, IEEE Trans. Knowl. Data Eng. 23 (2) (2011) 307–320.

[32] N. Iam-On, T. Boongoen, S. Garrett, C. Price, A link-based approach to the cluster ensemble problem, IEEE Trans. Pattern Anal. Mach. Intell. 33 (12) (2011) 2396–2409.

[33] N. Iam-On, T. Boongoen, S. Garrett, C. Price, A link-based cluster ensemble approach for categorical data clustering, IEEE Trans. Knowl. Data Eng. 24 (3) (2010) 413–425.

[34] D. Huang, C. Wang, H. Peng, J. Lai, C. Kwoh, Enhanced ensemble clustering via fast propagation of cluster-wise similarities, IEEE Trans. Syst. Man. Cybern. (2018), doi:10.1109/TSMC.2018.2876202.

[35] X. Fern, C. Brodley, Solving cluster ensemble problems by bipartite graph partitioning, in: Proc. of the 21st International Conference on Machine Learning, 2004.

[36] M. Selim, E. Ertunc, Combining multiple clusterings using similarity graph, Pattern Recognit. 44 (3) (2011) 694–703.

[37] Z. Yu, X. Zhu, H. Wong, J. You, J. Zhang, G. Han, Distribution-based cluster structure selection, IEEE Trans. Cybern. 47 (11) (2017) 3554–3567.

[38] D. Huang, J. Lai, C. Wang, Robust ensemble clustering using probability trajectories, IEEE Trans. Knowl. Data Eng. 28 (5) (2016) 1312–1326.

[39] P. Hore, L.O. Hall, B. Goldgo, A scalable framework for cluster ensembles, Pattern Recognit. 42 (5) (2009) 676–688.

[40] B. Long, Z. Zhang, P.S. Yu, Combining multiple clusterings by soft correspondence, in: Proc. the 4th IEEE International Conference on Data Mining, 2005.

[41] C. Boulis, M. Ostendorf, Combining multiple clustering systems, in: Proc. European Conf. Principles and Practice of Knowledge Discovery in Databases, 2004.

[42] D. Cristofor, D. Simovici, Finding median partitions using information theoretical based genetic algorithms, J. Univers. Comput. Sci. 8 (2) (2002) 153–172.

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

[44] H. Wang, H. Shan, A. Banerjee, Bayesian cluster ensembles, Stat. Anal. Data Min. 4 (1) (2011) 54–70.

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

[46] N. Nguyen, R. Caruana, Consensus clusterings, in: Proc. IEEE Int. Conf. Data Mining, 2007, pp. 607–612.

[47] Z. Huang, Extensions to the k-means algorithm for clustering large data sets with categorical values, Data Min. Knowl. Discov. 2 (3) (1998) 283–304.

[48] L. Bai, J. Liang, H. Du, Y. Guo, An information-theoretical framework for cluster ensemble, IEEE Trans. Knowl. Data Eng. 31 (8) (2019) 1464–1477.

[49] Z. Yu, P. Luo, J. Liu, H. Wong, J. You, G. Han, J. Zhang, Semi-supervised ensemble clustering based on selected constraint projection, IEEE Trans. Knowl. Data Eng. 30 (12) (2018) 2394–2407.

[50] Z. Yu, Z. Kuang, J. Liu, H. Chen, J. Zhang, J. You, H. Wong, G. Han, Adaptive ensembling of semi-supervised clustering solutions, IEEE Trans. Knowl. Data Eng. 29 (8) (2017) 1577–1590.

[51] Z. Yu, P. Luo, J. You, H. Wong, H. Leung, S. Wu, J. Zhang, G. Han, Incremental semi–supervised clustering ensemble for high dimensional data clustering, IEEE Trans. Knowl. Data Eng. 28 (3) (2016) 701–714.

[52] D. Huang, C. Wang, J. Lai, Locally weighted ensemble clustering, IEEE Trans. Cybern. 48 (5) (2018) 1460–1473.

[53] D. Huang, C. Wang, J. Wu, J. Lai, C. Kwoh, Ultra-scalable spectral clustering and ensemble clustering, IEEE Trans. Knowl. Data Eng. (2020), doi:10.1109/TKDE.2019.2903410.

[54] J.C. Bezdek, N.R. Pal, Some new indexes of cluster validity, IEEE Trans. Syst. Man Cybern.Part B 28 (3) (1998) 301–315.

[55] N.R. Pal, J.C. Bezdek, On cluster validity for the fuzzy c-means model, IEEE Trans. Fuzzy Syst. 3 (3) (1995) 370–379.

[56] W.M. Rand, Objective criteria for the evaluation of clustering methods, J. Am. Stat. Assoc. 66 (336) (1971) 846–850.

[57] W.H. T. S. A. V. W. T. Press, B.P. Flannery, Conditional Entropy and Mutual Information, Numerical Recipes: The Art of Scientific Computing (3rd ed.), Cambridge University Press, New York, 2007.