Pattern Recognition 92 (2019) 93–106

Contents lists available at ScienceDirect

Pattern Recognition

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

Ensemble clustering based on evidence extracted from the co-association matrix

Caiming Zhong[a,*], Lianyu Hu[b], Xiaodong Yue[c], Ting Luo[a], Qiang Fu[a], Haiyong Xu[a]

[a] College of Science and Technology, Ningbo University, Ningbo 315211, China
[b] Faculty of Information Science and Engineering, Ningbo University, Ningbo 315211, China
[c] Department of Computer Science and Technology, Shanghai University, Shanghai 200444, China

ARTICLE INFO

Article history:
Received 6 September 2018
Revised 13 March 2019
Accepted 23 March 2019
Available online 23 March 2019

Keywords:
Clustering ensemble
Co-association matrix
Path-based distance

ABSTRACT

The evidence accumulation model is an approach for collecting the information of base partitions in a clustering ensemble method, and can be viewed as a kernel transformation from the original data space to a co-association matrix. However, cluster structure information may be partially lost in this transformation; hence, some methods proposed in the literature try to find the lost information and return it to the ensemble process. In this paper, an interesting phenomenon is introduced: remove some evidences from the co-association matrix, which can result in more accurate clustering results. The intuitive explanation for this is that some evidences in the original co-association matrix could be noise, with negative effects on the final clustering. However, it is difficult to detect those evidences practically, let alone remove them from the matrix. To remedy this problem, we remove multiple level evidences having low occurrence frequencies, because negative evidences do not normally occur regularly in the base partitions. Subsequently, we use normalized cut to achieve multiple clustering results. To discriminate the optimal ensemble result, an internal validity index, which uses only the co-association matrix, is specially designed for the clustering ensemble. The experimental results on 16 datasets demonstrate that the proposed scheme outperforms some state-of-the-art clustering ensemble approaches.

1. Introduction

Cluster analysis is a fundamental problem in pattern recognition, data mining, and machine learning communities. It normally deals with two kinds of problems: whether a cluster structure exists in a dataset, and what type of cluster structure exists in the dataset. The former problem is to evaluate clusterability [1], and the latter is to detect the clustering. Although many clustering algorithms have been proposed in the literature [2–10], there is not one currently that can deal with all clustering problems [11]. Compared with conventional clustering methods, such as hierarchical, partitional, and density-based clustering, clustering ensemble [12–17] is relatively more universal, robust, and accurate. A clustering ensemble scheme usually consists of two components: a base partition generator and a final clustering generator.

To generate a set of base partitions, two performance indices quality and diversity are usually focused upon. If a base partition is of good quality, then its cluster homogeneity rate is high and its heterogeneity rate is low. Here, the cluster homogeneity rate of a base partition is defined as the percentage of the number of data point pairs that have the same cluster label in both the base partition and the ground truth (SS pairs). The heterogeneity rate is defined as the percentage of the number of pairs that have the same cluster label in the base partition but different in the ground truth (SD pairs). Intuitively, SS pairs have a positive effect on final clustering, and SD pairs have a negative effect. Diversity of the base partitions means specifying how different views of the cluster structure are disclosed by different base partitions. If each base partition discloses the cluster structure from different viewpoints, their consensus may indicate the global image of the structure. However, Hadjitodorov et al. claimed that moderate diversity of the base partitions could be more effective for cluster ensemble [18]. This paper does not focus on this topic.

Base partitions are usually produced in three ways: the same clustering algorithm with different parameters (different initial status), different clustering algorithms, and different subspaces. For the first kind of approach, K-means is frequently used as a base partition generator [19–22]. K-means can capture the local information of clusters (especially when the number of clusters is

* Corresponding author.
E-mail address: zhongcaiming@nbu.edu.cn (C. Zhong).

https://doi.org/10.1016/j.patcog.2019.03.020
0031-3203/

large), and is computationally efficient and produces different clusterings in different runs. When different clustering algorithms are employed to produce the base partitions, these algorithms should be selected complementarily to each other so that the base partition can describe the cluster structure from different views. Yoon et al. used K-means, the hierarchical algorithm, and a principle component analysis-based algorithm to generate the base partitions [23]. For high dimensional datasets, different subspaces can be projected to generate different base partitions [24]. This is because the cluster structure of a high dimensional dataset may be embedded in a certain subspace.

After being generated, the base partitions will be represented so that the cluster structure can be easily disclosed. The widely used representations include co-association matrix [20], binary matrix [19], and hypergraph [25]. The co-association matrix records the frequency of each pair appearing in the same cluster. This frequency can discover the neighbourhood information; hence, it can act as the similarity of a pair. Moreover, the co-association matrix can be viewed as a kernel transformation of the original data. A notable fact is that the original base partitions cannot be derived from a given co-association matrix. This means some partition information could be missed during the transformation from the partitions to the matrix. The binary matrix focuses on the relationship between a data point and a base cluster. For example, an entry is 1 if the data point belongs to the corresponding cluster, and 0 otherwise. In addition, it can be considered as a space transformation that conveys all the information of the base partitions. Strehl and Ghosh in [25] represented the based clusterings as a hypergraph, of which a hyperedge is a base cluster. After the base partitions are represented, some clustering algorithms (or graph partition methods) can be applied to the representations to obtain the final clustering.

Some improvements have been proposed in the literature to refine the co-association and binary matrices. To refine the co-association matrix, Zhong et al. [22] observed that in the same base cluster, point pairs with different distances could have different weights to accumulate the similarity evidences, and replaced occurrence frequencies with occurrence probabilities. Furthermore, the stability of a base cluster is also considered [22]. Huang et al. explored the uncertainty of a base cluster, and refined the co-association matrix according to the uncertainty [14]. Iam-On et al. refined zero entries in the binary matrix by measuring the similarity of two base clusters in the same base partition; hence, some hidden information was discovered and used [19]. Moreover, Liu et al. directly applied K-means to the binary matrix with entropy-based utility function and KL-divergence distance measure [26].

In this paper, we focus on remodelling the co-association matrix by removing some information from the matrix so that it can more effectively portray the intrinsic cluster structure. The main difference between this work and [22] is that the former removes blurring cluster structure information from the matrix, while the latter discovers some extra information of depicting cluster structure and adds it into the matrix. In addition, a dedicated internal validity index is proposed for clustering ensemble, as it only uses the information of co-association matrix rather than the original dataset, which may be not available in a clustering ensemble scenario.

The rest of the paper is organised as follows. Section 2 provides a brief background of the co-association matrix and a visual assessment of cluster tendency. Section 3 details the proposed method, which contains the procedure of removing negative information from the co-association matrix and a new clustering validity index dedicated for clustering ensemble. The experimental results are presented in Section 4. Finally, Section 5 concludes the paper.

2. Background

2.1. Base clusterings

Let $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ be a dataset, where $\mathbf{x}_i = (x_{i1}, \ldots, x_{id})^T \in \mathcal{R}^d$, d is the dimensionality of X, and N is the number of data points. The base partitions are usually generated by K-means with a fixed number of clusters, i.e., $K = \sqrt{N}$ [22]. This is because, as a general rule, the number is not bigger than $\sqrt{N}$ [27]. In addition, the number of SD pairs is relatively small and the number of SS pairs is relatively large, and this situation is expected for ensemble. By comparison, the quantity of SD is more important than that of SS, because an SD pair has a negative effect on the ensemble process, while an SS pair has a positive effect. An example is illustrated in Fig. 1, in which the number of clusters K is set to 17, and both the quantities of SS and SD are close to the

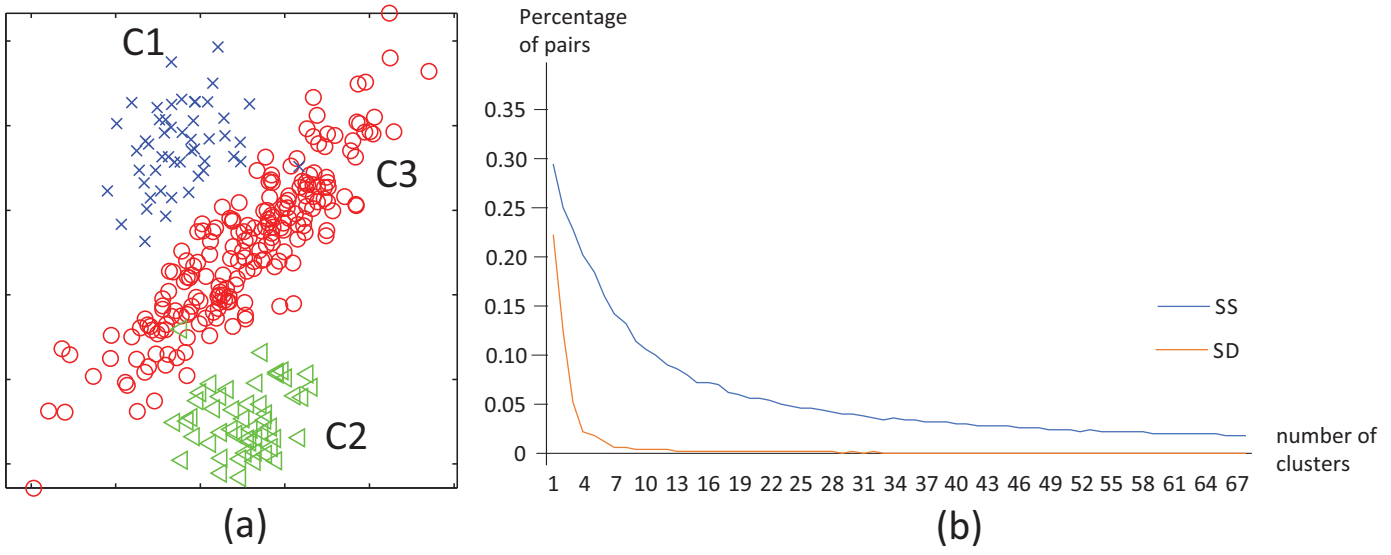

Fig. 1. (a) is a dataset with 300 points belonging to three Gaussian clusters, of which C_1 and C_2 have 50 data points each, and C_3 has 200 data points. Here, (b) is the average percentage of SS and SD pairs in the base partitions generated by K-means (10 runs) with different number of clusters K, where the iteration number of K-means is set to 4.

kneels of the curves of the percentage of pairs and the number of clusters.

The diversity of base partitions is considered by setting the iteration number of K-means. Normally, this is set to 100 in some tools (such as MATLAB). However, to generate base partitions for ensemble is not necessary for K-means to be convergent for the purpose of high diversity, and the iteration number could be relatively small. In Fig. 1, it is set to 4.

2.2. Co-association matrix

Let $P_1, \ldots, P_M$ denote the M base partitions, where $P_i = \{C_{i1}, \ldots, C_{iK_i}\}$, C_{ij} is the jth cluster of P_i, K_i is the number of clusters in P_i. Suppose $P = \{C_1, \ldots, C_K\}$ is the final clustering, and K is the number of clusters in P, C_i is a cluster of P.

A co-association matrix (CM) is defined as follows [20]:

$$CM(i,j) = \frac{1}{M}\sum_{m=1}^{M}\sum_{l=1}^{K_m}\mathcal{T}(i,j,C_{ml}) \tag{1}$$

where $CM(i,j)$ denotes an entry of CM, C_{ml} is the lth base cluster in P_m, and $\mathcal{T}(i,j,C_{ml})$ is an indicator:

$$\mathcal{T}(i,j,C_{ml}) = \begin{cases} 1, \text{if } \mathbf{x}_i \in C_{ml} \wedge \mathbf{x}_j \in C_{ml} \\ 0, \text{otherwise} \end{cases} \tag{2}$$

A co-association matrix depicts the frequency of a pair appearing in a similar base cluster, and it can be viewed as a similarity matrix for clustering. In this paper, we focus on how to detect the cluster structure according to this matrix.

2.3. Confusion of a pair

From the definition of $CM(i,j)$, the homogeneity (or heterogeneity) relationship of the pair $\mathbf{x}_i$ and $\mathbf{x}_j$ is deterministic when $CM(i,j)$ is 1 (or 0), but totally confused when $CM(i,j)$ is 0.5. Ren et al. defined an index, confusion, to depict the uncertainty of a pair, as follows [28]:

$$confusion(\mathbf{x}_i, \mathbf{x}_j) = CM(i,j) * (1 - CM(i,j)) \tag{3}$$

According to the confusion, the weight of a point is defined:

$$w(i) = \frac{w^*(i) + e}{1 + e} \tag{4}$$

where e is a small positive number, and

$$w^*(i) = \frac{4}{N}\sum_{j=1}^{N} confusion(\mathbf{x}_i, \mathbf{x}_j) \tag{5}$$

It is claimed that the points with large weights are difficult to cluster [28].

2.4. Visual assessment of cluster tendency

Visual assessment of cluster tendency (VAT) is to reorder the pairwise dissimilarity matrix $\mathbf{D}$ of X, denoted by $\mathbf{D}^*$, so that the cluster structure information is represented by an image $I(\mathbf{D}^*)$. It can also be used directly to cluster the dataset by segmenting the image [29–32].

The idea of VAT is that the more similar a pair, the more closely they will be reordered. As VAT only reorders the dissimilarity matrix, the cluster structure remains unchanged (Algorithm 1).

Algorithm 1: Visual assessment of cluster tendency (VAT).

Input: $N \times N$ dissimilarity matrix $\mathbf{D} = [d_{ij}]$
Output: Reordered dissimilarity matrix $\mathbf{D}^* = [d^*_{ij}]$

$J \leftarrow \{1, 2, \ldots, N\}$, $I \leftarrow \emptyset$, $Q \leftarrow (0, 0, \ldots, 0)$
$[i, j] \leftarrow \arg\min_{p,q \in J}\{d_{pq}\}$
$Q(1) \leftarrow j$, $I \leftarrow I \cup \{j\}$, $J \leftarrow J - \{j\}$
for $t = 2, \ldots, N$ **do**
  $[i, j] \leftarrow \arg\min_{p \in I, q \in J}\{d_{pq}\}$
  $Q(t) \leftarrow j$, $I \leftarrow I \cup \{j\}$, $J \leftarrow J - \{j\}$
$d^*_{ij} \leftarrow d_{Q(i)Q(j)}$, for $1 \le i, j \le N$

3. The proposed method

3.1. Overview of the proposed method

An overview of the proposed method is illustrated in Fig. 2. The input is a set of base partitions, and the output is the final clustering. The concrete steps are described as follows. As a self-contained method, the base partitions are generated by K-means with the number of clusters fixed to $\sqrt{N}$. Then, the co-association matrix is created, and each element of the matrix is between 0 and 1. Remodelled matrices are produced by setting those elements less than different thresholds to 0, and Ncut is applied to these

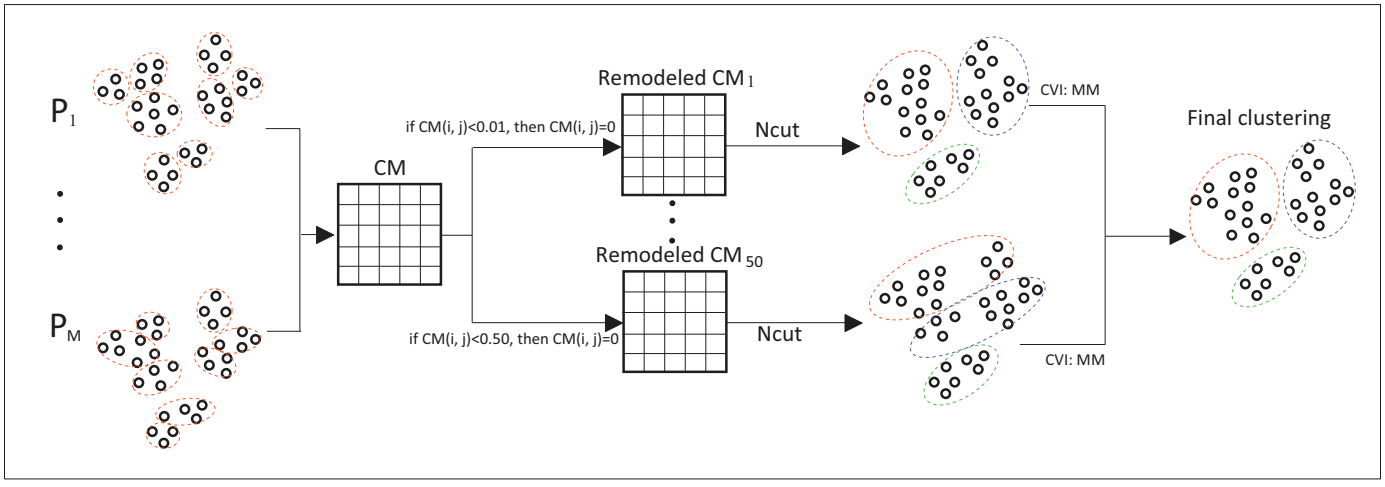

Fig. 2. The overview of the proposed method.

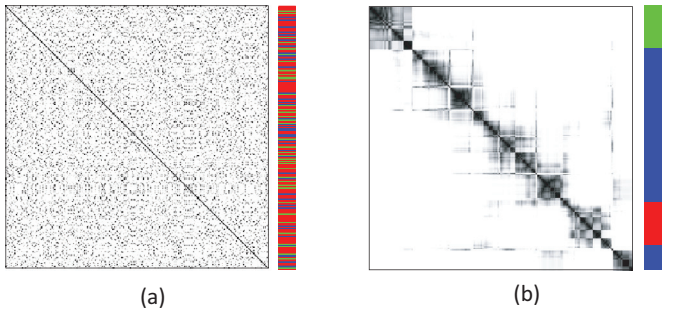

Fig. 3. The dataset in Fig. 1 is represented by a co-association matrix and the VAT matrix. In (a), the left is the co-association matrix and the data points in the matrix are randomly permuted; to the right are the corresponding cluster labels of the ground truth. The left in (b) is the co-association matrix reordered by VAT, and to the right are the corresponding cluster labels of the ground truth.

matrices to obtain multiple clusterings. The final clustering is selected by a new internal validity index MM.

3.2. Cluster structure and VAT matrix

As the final clustering is generated from the co-association matrix, this matrix should precisely represent the cluster structure. However, it is difficult to discern directly how accurate the matrix depicts the structure. One way could be to reorder the original co-association matrix by the VAT algorithm. The reordered version is then called the VAT matrix, from which the cluster structure can be observed directly (to some extent).

In Fig. 3, for example, we randomly permute the data points from the dataset illustrated in Fig. 1(a); however, no cluster structure information can be perceived from Fig. 3(a). When the VAT algorithm is applied to the original co-association matrix, the cluster structure emerges in Fig. 3(b). Please note, for some clustering algorithms such as normalised cut (Ncut) [33], the original co-association matrix and the matrix reordered by VAT have the same capability to depict the cluster structure, as they have only different data point orders, which have nothing to do with the cluster structure.

Although the VAT matrix may not depict the exact cluster structure directly, the exact structure can be discovered by observing the clues presented by the matrix.

3.3. VAT matrix's clue: Negative evidence

The dataset in Fig. 1 is analysed by Ncut with the co-association matrix. In the left of Fig. 4(a), VAT applied on the original co-association matrix divides the dataset into five blocks. For a pair in the VAT matrix, darker equates to being more similar. The clustering result on the matrix is illustrated to the right of Fig. 4(a). This result is unexpected, because a part of the data points of C_3 (namely, C_{12}) is clustered into C_1.

Looking into the dotted rectangle r1 in the left of Fig. 4(b) (which represents the similarities of block b4 to blocks b2 and b3), some similarities can be observed between b4 and b2. The similarities between b4 and b2 are caused by the fact that some data points from C_{11} and C_{12} are partitioned into the same base clusters. Similarly, b1 and b2 have some similarities marked in r2, but

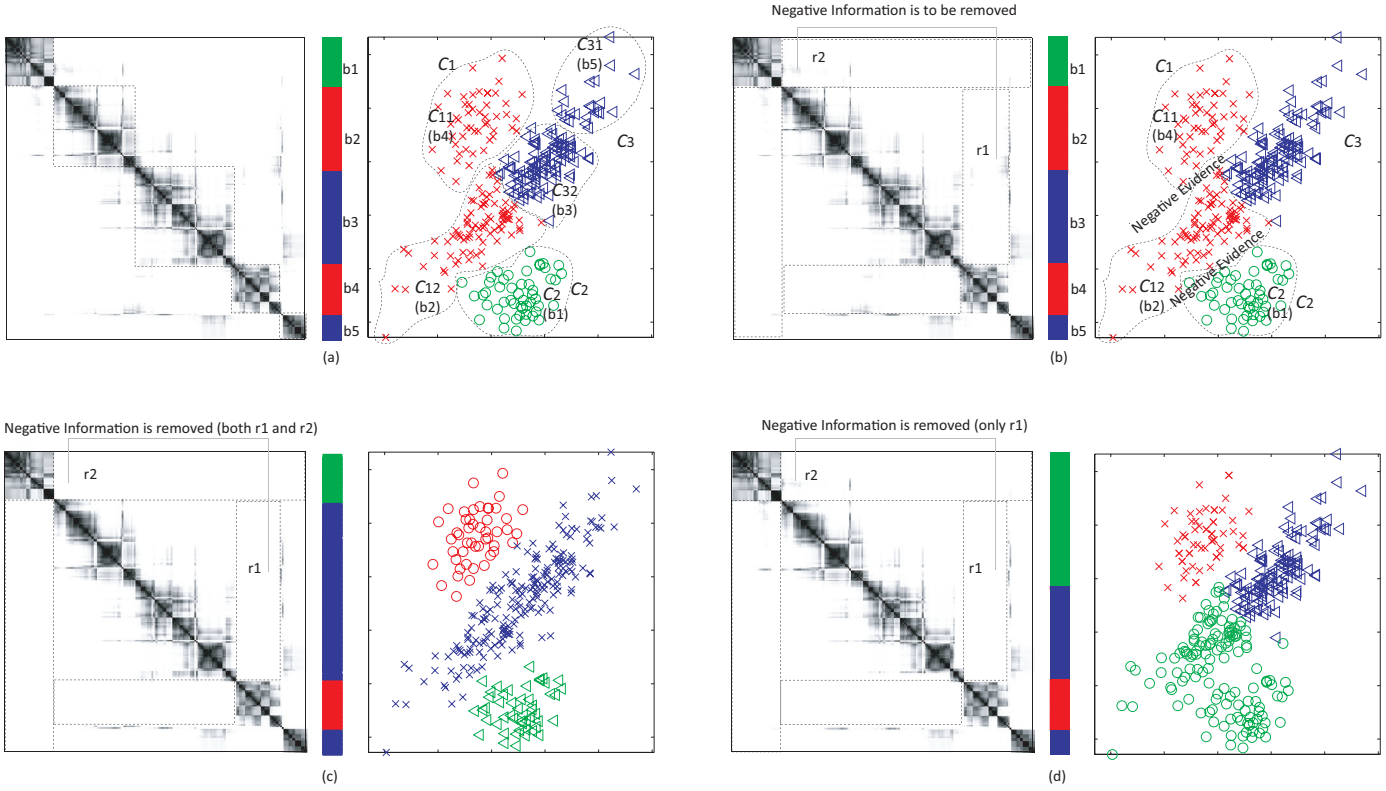

Fig. 4. The dataset in Fig. 1 is analysed by the VAT matrix and modified VAT matrix. The left in (a) is the VAT matrix, the right is the clustering result generated by Ncut towards this matrix, and the bar in the middle is the corresponding cluster labels. This bar is composed of five blocks, which are denoted from b1 to b5. The cluster C_1 contains C_{11} and C_{12}, which correspond to blocks b4 and b2, respectively. The cluster C_3 consists of C_{31} and C_{32}, which correspond to blocks b5 and b3, respectively. The cluster C_2 corresponds to block b1. In the left of (b), negative evidences are marked in the dotted rectangles. From the right of (b), it can be seen that the negative evidences are from block pairs b4 and b2, and b1 and b2. In the left of (c), the negative evidences are removed from the VAT matrix; the similarities of pairs in the regions are set to zero. The right in (c) is the clustering result generated by Ncut on the modified VAT matrix, and the bar in the middle is the corresponding cluster labels. The right of (d) is the clustering result with only negative evidences in r1 removed.

are fewer than those between b4 and b2. This is why C_{12} and C_{11} are partitioned into the same final cluster by Ncut.

With this observation in mind, we try to remove the similarities in r1 and r2 manually. After the removal, Ncut can provide a satisfied clustering result, which is illustrated in Fig. 4(c). Please note, if only similarities in r1 are removed, then C_{12} will be partitioned into C_2 because the similarities between b1 and b2 still exist in r2. This situation is illustrated in Fig. 4(d).

We call the similarity evidences in the dotted rectangles negative evidences, as they have some negative effects on the final clustering. Next, we consider why the removal of negative evidences might lead to a good clustering result. Considering that Ncut is applied to the negative information removed by the VAT matrix, we present the following theorem:

Theorem 1. *Suppose X is composed of two clusters A and B, which are reordered by VAT into m and n sub-clusters as $A_1, \ldots, A_m$ and $B_1, \ldots, B_n$ respectively. If no similarity evidence exists between A_i and B_j ($1 \le i \le m$, $1 \le j \le n$), but there is some between A_i and A_j ($1 \le i \le m$, $1 \le j \le m$) or between B_i and B_j ($1 \le i \le n$, $1 \le j \le n$), Ncut will find the optimal clustering.*

Proof. The objective function of Ncut is:

$$J_{Ncut}(A,B) = \frac{S(A,B)}{S(A,A)+S(A,B)} + \frac{S(A,B)}{S(B,B)+S(A,B)} \tag{6}$$

where $S(A,B) = \sum_{i\in A}\sum_{j\in B} w_{ij}$, w_{ij} is the similarity of $\mathbf{x}_i$ and $\mathbf{x}_j$, and $J_{Ncut}(A, B)$ is to be minimised. In this paper, $CM(i, j)$ is regarded as w_{ij}.

According to the supposition of the claim, $S(A_i, B_j) = 0$ ($1 \le i \le m$, $1 \le j \le n$), then $S(A, B) = 0$. For any clustering, if $\exists B_i \subset A$ or $\exists A_i \subset B$, then $S(A, B) > 0$. Therefore, if the supposition is satisfied, the optimal clustering will be achieved. □

When kernel K-means [34] is applied to the negative information removed by the VAT matrix, the following theorem holds:

Theorem 2. *Suppose X is composed of two clusters C_1 and C_2, if no similarity evidence exists between C_1 and C_2, and the average similarity of objects inside C_1 is equal to that inside C_2, the kernel K-means will find the optimal clustering.*

Proof. Let d_{nk} denote the distance between $\mathbf{x}_n$ and the centre of C_k, and it can be computed as in [34]:

$$d_{nk} = \kappa(\mathbf{x}_n,\mathbf{x}_n) - \frac{2}{N_k}\sum_{m=1}^{N} z_{mk}\kappa(\mathbf{x}_n,\mathbf{x}_m) + \frac{1}{N_k^2}\sum_{m=1}^{N}\sum_{r=1}^{N} z_{mk}z_{rk}\kappa(\mathbf{x}_m,\mathbf{x}_r), \tag{7}$$

where $\kappa(\mathbf{x}_n, \mathbf{x}_m)$ is a kernel function, and z_{mk} denotes whether $\mathbf{x}_m$ is in C_k.

$\kappa(\mathbf{x}_n, \mathbf{x}_m)$ can be replaced by the similarity of $\mathbf{x}_n$ and $\mathbf{x}_m$ under a certain similarity measure. For example, when the kernel function is replaced by a Gaussian kernel, we may use $CM(n, m)$ to replace $\kappa(\mathbf{x}_n, \mathbf{x}_m)$.

Suppose for any $\mathbf{x}_i \in C_1$ and $\mathbf{x}_j \in C_2$, $CM(i, j) = 0$, then we have the following:

$$d_{i1} = CM(i,i) - \frac{2}{N_1}\sum_{m=1}^{N} z_{m1}CM(i,m) + \frac{1}{N_1^2}\sum_{m=1}^{N}\sum_{r=1}^{N} z_{m1}z_{r1}CM(m,r), \tag{8}$$

and

$$d_{i2} = CM(i,i) + \frac{1}{N_2^2}\sum_{m=1}^{N}\sum_{r=1}^{N} z_{m2}z_{r2}CM(m,r) \tag{9}$$

As the average similarity of objects inside C_1 is equal to that inside C_2, $d_{i1} < d_{i2}$. Similarly, $d_{j2} < d_{j1}$. That is to say, kernel K-means can find the optimal clustering. □

The above two theorems indicate that the removal of the similarity evidence may lead to a good clustering result. However, it is difficult to determine which evidences should be removed. Fortunately, the removed evidence has a prominent characteristic: the corresponding similarity is small. This phenomenon is reasonable: for a clustering, if a pair of data points is in the same cluster but has different cluster labels in ground truth, then the corresponding entry in *CM* should be small. This can be explained from the viewpoint of SD pair and confusion.

In the base clusters, we assume that the frequencies of occurrence of SD pairs are usually small; otherwise, the quality of the base partitions is low. If this assumption holds, to remove the similarity evidence in the dotted regions is to remove SD pairs. Moreover, in any ensemble scheme, SD pairs have a negative contribution to the final clustering; meaning small frequencies of occurrence should be removed.

Ren et al. defined a confusion and weight index, and claimed that it would be difficult to cluster a point with a large weight [28]. Confusion of a pair means the uncertainty of a pair of data points being in the same cluster. For any pair, the confusion is maximised when the normalised frequency is 0.5. When a frequency is much less than 0.5, the corresponding pair could be regarded as an SD pair.

3.4. Collect the candidate clusterings

According to the above discussion, although it is difficult to determine the negative evidence regions, we can collect multiple candidates of the final clustering by gradually removing the similarity evidences of *CM* in [0, 0.5] with step of 0.01, and then applying Ncut to the changed *CM*s.

The algorithm is described in Algorithm 2. In line 8, a candidate clustering is achieved by using Ncut on the similarity matrix CM', which is a modified *CM*. Please note that when Ncut is applied, CM' is not transformed with the Gaussian kernel again; hence, *K* is the specified number of clusters in a clustering.

Algorithm 2: Collect the candidate clusterings.

Input: Co-association matrix *CM*, number of clusters *K*
Output: Collection of candidate clusterings $\mathcal{C}$

```
1 C ← ∅
2 S ← [0, 0.01, 0.02, ..., 0.5]
3 for each s ∈ S do
4     CM' ← CM
5     for each CM'(i, j) do
6         if CM'(i, j) ≤ s then
7             CM'(i, j) ← 0
8     C ← C ∪ Ncut(CM', K)
```

After the candidate clusterings are collected, the best clustering is selected as the final clustering. To achieve this, an effective internal clustering validity index is needed. However, the well-known internal validity indices, such as DB [35], Dun [36], are not so effective for some datasets with complex cluster structures. Moreover, some of these indices require the information of the original dataset, which may not be available in a clustering ensemble scenario. In this paper, we propose a new index that only uses the co-association matrix and is more effective than some popular indices.

3.5. Determine the best clustering

3.5.1. Internal validity index

An internal validity index is a criterion that measures the clustering quality without any extra information, except for the

dataset. In general, it can be used for two tasks: to select the best clustering (or clustering algorithm), and to determine the optimal number of clusters. The latter depends on a hypothesis: for clusterings produced by the same algorithm, the one with the optimal number of clusters is better than those with a non-optimal number of clusters. However, in this paper, we only focus on the first task of distinguishing the best clusterings.

Two factors are generally considered for designing an internal validity index: compactness and separation [37]. Compactness measures distances of data points in the same cluster, and it is usually computed by the sum-of-squared error. Separation measures the distances of data points in different clusters, and is normally computed by the distance of two cluster centres. These two factors may come from the definition of clustering: data points in the same cluster are similar, and those in different clusters are dissimilar. However, validity indices designed from this definition fail when the dataset has a complex structure.

In addition, traditional indices are designed for normal clustering algorithms, but not for clustering ensemble. In some cases, only base partitions are available, and some traditional indices do not work because they need information from the original dataset. In this paper, a new validity index, called MM index, is proposed. It is based on the following cluster definition: a high-density region separated by low-density regions, and can be applied to clustering ensemble.

3.5.2. Minimax similarity

Suppose $G = (X, E)$ is an undirected graph of X, and $\mathcal{S}_X$ is the similarity matrix of X. The minimax similarity [38] is defined as follows:

Let $\mathscr{P}_{ij}^X$ denote the set of all possible paths between vertex $\mathbf{x}_i \in X$ and $\mathbf{x}_j \in X$. The *minimax* similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ with respect to $\mathcal{S}_X$ is:

$$Sim(\mathbf{x}_i, \mathbf{x}_j, \mathcal{S}_X) = \max_{\mathcal{P} \in \mathscr{P}_{ij}^X} \left\{ \min_{1 \le m < |\mathcal{P}|} s(\mathcal{P}[m], \mathcal{P}[m+1]) \right\} \tag{10}$$

where $Sim(\mathbf{x}_i, \mathbf{x}_j, \mathcal{S}_X)$ is the minimax similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$, $\mathcal{P}$ is a path from vertex $\mathbf{x}_i$ to $\mathbf{x}_j$, $\mathcal{P}[m]$ is the mth vertex along the path, and $s(\mathbf{x}_p, \mathbf{x}_q)$ is the similarity of $\mathbf{x}_p$ and $\mathbf{x}_q$ from $\mathcal{S}_X$.

This minimax similarity can be computed efficiently by using the minimum spanning tree [39].

Chang et al. proposed a robust minimax similarity to rule out the effect of noise data [38]. The robust minimax similarity is defined as follows:

$$RSim(\mathbf{x}_i, \mathbf{x}_j, \mathcal{S}_X, l) = \max_{\mathcal{P} \in \mathscr{P}_{ij}^X} \left\{ \min_{1 \le m < |\mathcal{P}|} s(\mathcal{P}[m], \mathcal{P}[m+1]) w_m w_{m+1} \right\} \tag{11}$$

where $w_m = \sum_{\mathbf{x}_p \in \mathcal{N}(\mathbf{x}_m, l)} s(\mathbf{x}_m, \mathbf{x}_p) / \max_{\mathbf{x}_q \in X} (\sum_{\mathbf{x}_k \in \mathcal{N}(\mathbf{x}_q, l)} s(\mathbf{x}_q, \mathbf{x}_k))$, and $\mathcal{N}(\mathbf{x}, l)$ is the set of l nearest neighbours of $\mathbf{x}$.

3.5.3. MM

Suppose $C = \{C_1, \ldots, C_K\}$ is the clustering to be measured. The minimax similarity based internal validity index, MM, is defined as:

$$MM = \sum_{1 \le i \le K} cohesion(C_i, X \setminus C_i) / stability(C_i) \tag{12}$$

where $cohesion(C_i, C_j)$ is the cohesion between cluster C_i and C_j, and $stability(C_i)$ is the stability of C_i. The former focuses on the density-based connectivity of C_i with other clusters, and the latter focuses on the inner density-based connectivity of C_i.

The robust minimax similarity matrix is used to define the cohesion and the stability as follows:

$$cohesion(C_i, C_j) = \max_{\mathbf{x}_p \in C_i, \mathbf{x}_q \in C_j} RSim(\mathbf{x}_p, \mathbf{x}_q, \mathcal{S}_X, l) \tag{13}$$

$$stability(C_i) = \min_{\mathbf{x}_p \in C_{i1}, \mathbf{x}_q \in C_{i2}} RSim(\mathbf{x}_p, \mathbf{x}_q, \mathcal{S}_{C_i}, l) \tag{14}$$

where C_{i1} and C_{i2} are produced by bi-partitioning C_i with a clustering algorithm, and Ncut is used in this paper. Please note, when $stability(C_i)$ is computed, the paths are only composed of edges from C_i. This is to remove the effects of those data points not in C_i.

To obtain a good clustering, it is expected that the cohesion of C_i and $X \backslash C_i$ is weak and the stability of C_i is strong. Therefore, the clustering with the smallest MM is expected.

The intuition behind MM is that in a clustering with high quality, a cluster is a high-density region separated by some low-density regions. This conforms to the definition of a cluster in [40]: data points are likely in the same cluster if there is a path connecting them passing through regions of high density only.

The parameter l in MM is the number of the nearest neighbours, which can convey the local density information. If there is a path connecting the data points only passing through high-density regions, it can be disclosed by the minimax similarity combined with this local density information. In all the experiments, parameter l is set to 3. The discussion of l is in Section 4.5.

The algorithm of MM is described in Algorithm 3. In line 1,

Algorithm 3: Compute MM measure.

Input: Similarity matrix $\mathcal{S}_X$, number of nearest neighbours l, clustering to be measured $C = \{C_1, \ldots, C_K\}$
Output: Quality measured by MM

```
1  RS_X ← RSim(x_p, x_q, S_X, l)|_{x_p,x_q∈X}
2  mm ← 0
3  for i=1, 2,..., K do
4      if |C_i| < l then
5          return INF
6      coh ← max_{x_p∈C_i, x_q∈X\C_i} RS_X
7      S_{C_i} ← S_X(C_i)
8      RS_{C_i} ← RSim(x_p, x_q, S_{C_i}, l)|_{x_p,x_q∈C_i}
9      [C_i1, C_i2] ← Ncut(RS_{C_i}, 2)
10     st ← min_{x_p∈C_i1, x_q∈C_i2} RS_{C_i}
11     mm ← mm + coh/st
12 return mm
```

RS_X is the robust minimax similarity matrix of X. For simplicity, in lines 4 and 5, when the cardinality of C_i is less than the number of neighbours, this partition is directly discarded, as C_i is a spurious cluster. In line 7, the similarity matrix of C_i is formed by simply extracting corresponding rows and columns from $\mathcal{S}_X$.

The final clustering ensemble result is achieved by selecting the candidate clustering with minimum MM value; see Algorithm 4.

Algorithm 4: Negative evidence removed clustering ensemble (NegMM).

Input: Collection of candidate clusterings $\mathcal{C}$
Output: Final clustering $\mathcal{C}_{final}$

```
1  C_final ← NULL
2  for each C_i ∈ C do
3      if C_final is NULL or MM(C_final) > MM(C_i) then
4          C_final ← C_i
```

3.6. Computational complexity

Given a set of base partitions, the computational complexity of the proposed method is analysed as follows:

The computational time can be composed of two parts: T_1 and T_2. The computational time of producing the candidate clusterings from a set of base partitions is denoted as T_1, and computational time of selecting the final clustering from the candidates is denoted as T_2. Further, T_1 includes the time of computing CM and producing the candidate clusterings. The time to compute CM from M base partitions is as follows:

$$T_{CM} = \sum_{i=1}^{M} \sum_{C_{ij} \in P_i} \binom{|C_{ij}|}{2} \tag{15}$$

For simplicity, we suppose each base partition has $\sqrt{N}$ clusters, and the clusters have the same size. Then, $T_{CM} \approx \frac{1}{2} M * N^{\frac{3}{2}}$.

To generate a candidate clustering, we first set the entries of CM that is less than a given threshold to 0, which takes $O(N^2)$. Then, a candidate clustering is produced by Ncut with time complexity of $O(N^3)$ [41]. Therefore, the computational complexity of producing candidate clusterings from the given base partitions T_1 is $O(N^3 + N^2 + \frac{1}{2} M * N^{\frac{3}{2}})$.

To select the final clustering from the candidates, we use the proposed internal validity index MM. The two components of MM are *cohesion* and *stability*, which are both computed from the robust path-based similarity. As the time complexity of the robust path-based similarity is $O(N^2)$ [42], MM has the same complexity: $T_2 = O(N^2)$.

Considering T_1 and T_2 simultaneously, the proposed method has a computational complexity of $O(N^3)$.

4. Experimental analysis

4.1. Compared internal clustering indices

In this section, the compared internal indices are introduced, which are listed in Table 1.

1. Silhouette index [43]. It is defined as:

$$Sil(C) = \frac{1}{N} \sum_{C_i \in C} \sum_{\mathbf{x} \in C_i} \frac{b(\mathbf{x}, C_i) - a(\mathbf{x}, C_i)}{\max(a(\mathbf{x}, C_i), b(\mathbf{x}, C_i))} \tag{16}$$

where $C = \{C_1, \ldots, C_K\}$, $a(\mathbf{x}, C_i) = \frac{1}{|C_i|} \sum_{\mathbf{x}' \in C_i} d(\mathbf{x}, \mathbf{x}')$, and $b(\mathbf{x}, C_i) = \min_{C_j \in C \setminus C_i} \{\frac{1}{|C_j|} \sum_{\mathbf{x}' \in C_j} d(\mathbf{x}, \mathbf{x}')\}$. $d(\mathbf{x}, \mathbf{x}')$ is the distance between the pair of data points $\mathbf{x}$ and $\mathbf{x}'$.
This index measures the normalised difference between the intracluster and intercluster average distances, where $a(\mathbf{x}, C_i)$ is the compactness of C_i, and $b(\mathbf{x}, C_i)$ represents the separation of C_i. Its value can be from −1 to 1, where 1 represents the best.
2. The Davies-Bouldin index [35] measures the intracluster similarity by the average distance from objects to the cluster centre, and the intercluster similarity by the distance between cluster centres. It is defined as follows:

$$DB(C) = \frac{1}{K} \sum_{C_i \in C} \max_{C_j \in C \setminus C_i} \frac{S_i + S_j}{d(\text{mean}(C_i), \text{mean}(C_j))} \tag{17}$$

where $S_i = \frac{1}{|C_i|} \sum_{\mathbf{x} \in C_i} d(\mathbf{x}, \text{mean}(C_i))$, and $\text{mean}(C_i)$ is the centre of C_i.
3. The Calinski-Harabasz index [44] is defined as follows:

$$CH(C) = \frac{N - K}{K - 1} \frac{\sum_{C_i \in C} (|C_i| d(\text{mean}(C_i), \text{mean}(X)))}{\sum_{C_i \in C} \sum_{\mathbf{x} \in C_i} d(\mathbf{x}, \text{mean}(C_i))} \tag{18}$$

where $\text{mean}(X)$ is the centre of X.
4. The Dunn index [36] is defined as follows:

$$Dunn(C) = \frac{\min_{C_i \in C, C_j \in C \setminus C_i} \delta(C_i, C_j)}{\max_{C_k \in C} \Delta(C_k)} \tag{19}$$

where $\delta(C_i, C_j) = \min_{\mathbf{x} \in C_i, \mathbf{x}' \in C_j} d(\mathbf{x}, \mathbf{x}')$, and $\Delta(C_k) = \max_{\mathbf{x}, \mathbf{x}' \in C_k} d(\mathbf{x}, \mathbf{x}')$.
5. S_Dbw index [45]. Suppose the standard deviation of a cluster is as $\sigma(C_i) = \frac{1}{|C_i|} \sum_{\mathbf{x} \in C_i} (\mathbf{x} - \text{mean}(C_i))^2$, and the standard deviation of clustering is as $\text{stdev}(C) = \frac{1}{K} \sqrt{\sum_{C_i \in C} \|C_i\|}$, $\|\mathbf{x}\| = \sqrt{\mathbf{x}^T \mathbf{x}}$ is defined as follows:

$$S\_Dbw(C) = Scat(C) + Den(C) \tag{20}$$

where

$$Scat(C) = \frac{1}{K} \sum_{C_i \in C} \frac{\sigma(C_i)}{\sigma(X)}$$

$$Den(C) = \frac{1}{K(K-1)} \sum_{C_i \in C} \sum_{C_j \in C \setminus C_i} \frac{den(C_i \cup C_j)}{\max(den(C_i), den(C_j))}$$

$$den(C_i) = \sum_{\mathbf{x} \in C_i} f(\mathbf{x}, \text{mean}(C_i))$$

and $f(\mathbf{x}, \text{mean}(C_i)) = 0$, if $d(\mathbf{x}, \text{mean}(C_i)) > \text{stdev}(C)$, and 1, otherwise.
6. CVNN index [37].

$$CVNN(C) = Sep(C, l) + Com(C) \tag{21}$$

where $Sep(C, l) = \max_{C_i \in C} \sum_{\mathbf{x} \in C_i} ((|\mathcal{N}(\mathbf{x}, l) \setminus C_i|)/l)$, $Com(C) = \sum_{C_i \in C} \frac{2}{|C_i| \times (|C_i| - 1)} \sum_{\mathbf{x}, \mathbf{x}' \in C_i} d(\mathbf{x}, \mathbf{x}')$, and $\mathcal{N}(\mathbf{x}, l)$ is the set of l nearest neighbors of $\mathbf{x}$.
7. $\mathcal{I}$ index [46].

$$\mathcal{I}(C) = \left(\frac{1}{K} \times \frac{\sum_{\mathbf{x} \in X} d(\mathbf{x}, \text{mean}(X))}{\sum_{C_i \in C} \sum_{\mathbf{x} \in C_i} d(\mathbf{x}, \text{mean}(C_i))} \times D_K \right)^p \tag{22}$$

where $D_K = \max_{C_i, C_j \in C} d(\text{mean}(C_i), \text{mean}(C_j))$.

Among the above 7 indices, CVNN and S_Dbw take density information into account, while the others only employ Euclidean distance information.

Table 1
The internal validity indices to be compared.

Internal validity indices	Optimal value	Refs
Silhouette	Min	[43]
Davies-Bouldin	Max	[35]
Calinski-Harabasz	Max	[44]
Dunn	Min	[36]
S_Dbw	Max	[45]
CVNN	Max	[37]
$\mathcal{I}$	Min	[46]

4.2. Experimental datasets and compared clustering ensemble algorithms

The proposed method is tested on 16 datasets, of which 8 are synthetic and 8 are real. The synthetic two-dimensional datasets are shown in Fig. 5. Detailed descriptions of these 16 datasets are in Table 2.

Dataset path based is composed of two Gaussian clusters and an unclosed ring cluster. It is difficult to detect the cluster structure by a single compactness-based or connectivity-based objective function. Dataset Spiral has three link-based clusters, and can be easily dealt with by a single-linkage algorithm. However, this

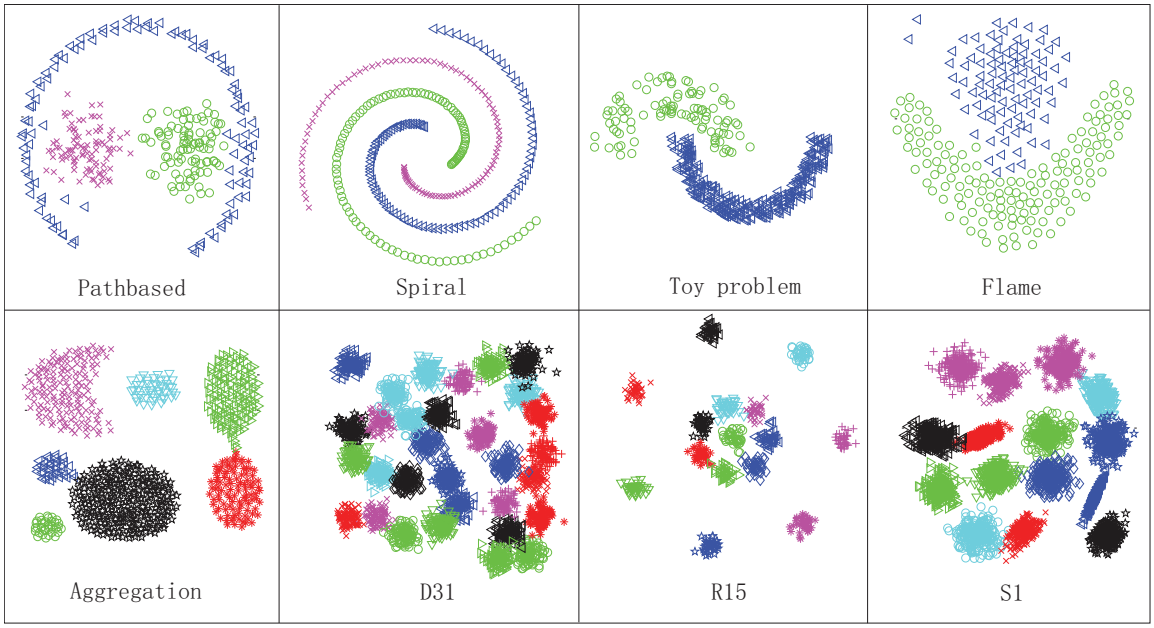

Fig. 5. Synthetic datasets.

Table 2
The description of the datasets.

Datasets	Classes	Objects	Dimensions	Refs
Pathbased	3	300	2	[38]
Spiral	3	312	2	[38]
Toy problem	2	373	2	[48]
Flame	2	240	2	[49]
Aggregation	7	788	2	[50]
D31	31	3100	2	[51]
R15	15	600	2	[51]
S1	15	5000	2	[52]
Iris	3	150	4	[53]
Ionosphere	2	351	34	[53]
Wine	3	178	13	[53]
Diabetes	2	768	8	[53]
Segmentation	7	2130	19	[53]
Glass	6	214	9	[53]
WDBC	2	569	30	[53]
WPBC	2	194	33	[53]

dataset is difficult for K-means, even for clustering ensemble algorithms that take K-means as a base partition algorithm. The Toy problem includes two clusters with different densities. Flame has two clusters and two noise data points, and is not favoured by compactness-based or connectivity-based algorithms. The other four synthetic datasets can be handled relatively easily because the clusters follow a Gaussian distribution.

To compare with clustering ensemble algorithms, two well-known approaches are selected: Link-based Clustering Ensemble (LCE) [19] and Strehl's algorithms [25]. The former has three variants: Weighted Connected-Triple (WCT), Weighted Triple-Quality (WTQ) and Combined Similarity Measure (CSM). The latter also has three variants: Cluster-based Similarity Partitioning Algorithm (CSPA), Hyper Graph Partitioning Algorithm (HGPA), and Meta-Clustering Algorithm (MCLA).

4.3. Clustering measures

To measure the clustering results, we employ three criteria: Hubert's Γ statistic [47], normalised mutual information [25], and CA [22].

In statistics, a null hypothesis is used to test a parameter against a specific value, but it is expressed in a slightly different way [47]. It can also be used to measure a clustering result by testing whether or not the dataset possesses a structure, and Hubert's Γ statistic is a typical example [47]. Suppose $P' = \{C'_1, \cdots, C'_{K'}\}$ is the ground truth of X, $P = \{C_1, \cdots, C_K\}$ is a clustering result. Consider a pair $(\mathbf{x}_i, \mathbf{x}_j)$. Let a denote the number of pairs where the two objects of each are in the same cluster with respect to P and P'. Let b denote the number of pairs where the two objects of each are in the same cluster with respect to P but in different clusters with respect to P'. Let c denote the number of pairs where the two objects of each are in different clusters with respect to P and in the same cluster with respect to P'. Finally, let d denote the number of pairs where the two objects are in different clusters with respect to both P and P'. Hubert's Γ statistic [47] is defined as follows:

$$\widehat{\Gamma} = (Ma - m_1 m_2)/\sqrt{(m_1 m_2 - (M - m_1)(M - m_2))}; \tag{23}$$

where $M = a + b + c + d$, $m_1 = a + b$, and $m_2 = a + c$.

As mutual information can depict the shared information of a pair of clusterings, the normalised mutual information in [25] is usually used as an external validity criterion, which is defined as follows:

$$NMI(P, P') = \frac{\sum_{i=1}^{K}\sum_{j=1}^{K'} |C_i \cap C'_j| \log(\frac{N|C_i \cap C'_j|}{(|C_i||C'_j|)})}{\sqrt{(\sum_{i=1}^{K} |C_i| \log\frac{|C_i|}{N})(\sum_{j=1}^{K'} |C'_j| \log\frac{|C'_j|}{N})}} \tag{24}$$

where $|C|$ denotes the number of objects in C.

CA is another external criterion for measuring a clustering result by computing the error rate [22]. It is defined as follows:

$$CA(P, P') = \frac{1}{N}\sum_{C_i \in P} |C_i \cap mode(C_i, P')| \tag{25}$$

where $mode(C_i, P') = \arg\max_{C'_j \in P'} |C_i \cap C'_j|$.

4.4. Experimental results

4.4.1. Compared to the selected ensemble approaches

As the 16 datasets in the experiments have labels, the qualities of all the experimental results are measured by Hubert's Γ statis-

Table 3
The qualities of the clustering results are measured by *CA*. The highest quality of clustering results in each row is highlighted as the bold item(s). The numbers in the brackets are the corresponding standard deviations.

Dataset	Best candidate	Clustering ensemble algorithms						
		NegMM	WCT	WTQ	CSM	CSPA	HGPA	MCLA
Pathbased	0.99	**0.98** (0.02)	0.92 (0.07)	0.90 (0.07)	0.88 (0.06)	0.93 (0.05)	0.92 (0.01)	0.82 (0.11)
Spiral	0.66	**0.63** (0.03)	0.40 (0.05)	0.41 (0.05)	0.38 (0.03)	0.38 (0.03)	0.60 (0.07)	0.44 (0.09)
Toy problem	1.00	**1.00** (0.00)	0.79 (0.10)	0.89 (0.11)	0.78 (0.08)	0.76 (0.00)	0.74 (0.00)	0.77 (0.01)
Flame	0.98	**0.98** (0.01)	0.96 (0.02)	0.96 (0.01)	0.95 (0.05)	0.83 (0.01)	0.86 (0.05)	0.84 (0.02)
Aggregation	1.00	**0.99** (0.00)	0.91 (0.01)	0.87 (0.03)	0.91 (0.00)	0.82 (0.02)	0.87 (0.01)	0.84 (0.02)
D31	0.98	**0.98** (0.00)	0.97 (0.00)	0.97 (0.01)	0.97 (0.01)	0.97 (0.00)	0.93 (0.02)	**0.98** (0.00)
R15	1.00	**1.00** (0.00)	**1.00** (0.00)	**1.00** (0.00)	0.99 (0.02)	**1.00** (0.00)	**1.00** (0.00)	**1.00** (0.00)
S1	1.00	**0.99** (0.00)	**0.99** (0.00)	**0.99** (0.00)	**0.99** (0.00)	0.97 (0.01)	0.89 (0.07)	0.97 (0.01)
Iris	0.98	0.96 (0.04)	0.94 (0.04)	0.95 (0.04)	0.91 (0.03)	0.96 ()0.05	**0.97**(0.00)	0.96 (0.05)
Ionosphere	0.88	**0.83** (0.06)	0.68 (0.04)	0.71 (0.01)	0.66 (0.03)	0.66 (0.01)	0.68 (0.02)	0.71 (0.04)
Wine	0.73	**0.72** (0.00)	**0.72** (0.01)	**0.72** (0.02)	**0.72** (0.01)	0.71 (0.00)	**0.72** (0.00)	**0.72** (0.00)
Diabetes	0.65	**0.65** (0.00)	**0.65** (0.00)	**0.65** (0.00)	**0.65** (0.00)	**0.65** (0.00)	**0.65** (0.00)	**0.65** (0.00)
Segmentation	0.69	0.64(0.16)	**0.69** (0.04)	0.63 (0.02)	0.68 (0.04)	**0.69** (0.05)	0.67 (0.05)	0.68 (0.07)
Glass	0.62	**0.62** (0.01)	0.60 (0.01)	0.60 (0.03)	0.61 (0.02)	0.61 (0.01)	0.61 (0.02)	0.58 (0.04)
WDBC	0.90	**0.86** (0.04)	0.73 (0.03)	0.78 (0.02)	0.71 (0.05)	0.76 (0.08)	0.82 (0.00)	0.81 (0.04)
WPBC	0.76	**0.76** (0.00)	**0.76** (0.00)	**0.76** (0.00)	**0.76** (0.00)	**0.76** (0.00)	**0.76** (0.00)	**0.76** (0.00)

tic [47], normalised mutual information [25], and *CA* [22]. In all experiments, the parameters are set as follows: because clustering results of each clustering ensemble algorithm are not unique, we run each algorithm repeatedly 50 times, and the quality of averages is presented. The number of clusters for each base partition is $\sqrt{N}$. The number of the base partitions is set to 500 for NegMM, CSPA, HGPA, and MCLA, and 10 for WCTspec, WTQspec, and CSMspec. Meanwhile, the decay factor parameters *DC* of CTspec, WTQspec, and CSMspec are set to 0.9. The number of the nearest neighbours is set to 3 for MM[1].

The results measured by *CA* are shown in Table 3. In the second column, the best clusterings are selected from the candidates generated by Algorithm 2, and the qualities are measured by CA. For the Path based dataset, the best candidate is similar to the ground truth, and the quality of the result of NegMM is better than those of others. As the cluster structure of this dataset is quite complex, it can be said that NegMM has the potential capability of dealing with datasets with complex structures.

For the Spiral dataset, even the best candidate is far from the ground truth. This is because the base clusterings are of low quality when the number of base clusters is set to $\sqrt{N}$, which is not enough for this link-based dataset. For example, if the number is set to $2 * \sqrt{N}$, the ground truth can be found in the candidates. Even if the number is $\sqrt{N}$, NegMM produces a better result than the others. The Toy problem and Flame datasets have arbitrary shapes and complex structures, and NegMM gives the best clusterings. The Aggregation dataset contains two chain-linked Gaussian clusters, but NegMM is robust and outperforms others on this dataset. For the other three synthetic datasets (D31, R15, and S1), only NegMM has the best results simultaneously.

The proposed NegMM has good performance on Iris, although HGPA provides the best performance. Concerning the Ionosphere dataset, the best candidate is of high quality. To our knowledge, there is no clustering ensemble algorithm in the literature that can produce clustering with a similar quality on this dataset. NegMM can find a relatively good clustering from the candidates. For the other six real datasets, the only one not favoured by NegMM is Segmentation.

The results measured by Hubert's Γ statistic and NMI are shown in Tables 4 and 5, respectively. The evaluations of these two measures on the experimental results are similar to that of CA, especially for those clustering with high qualities. For example, from Tables 3, 4 and 5, it can be seen that the three measures present a similar idea towards the clustering results of the eight synthetic datasets (except for Spiral), and these seven clustering results generated by NegMM are of high quality. When clustering results are of low quality (for example, those of Spiral and the eight real datasets), the three measures have significantly different views. However, from the number of the best clusterings, the three measures give the same evaluation result: NegMM outperforms the compared state-of-the-art methods.

To test NegMM extensively, we compare NegMM with the other six methods on 160 synthetic datasets[2]. These are listed in Table 6. The number of dimensions of the datasets range from 2 to 100, and the number of clusters ranges from 4 to 40. As the datasets are generated with Gaussian and Ellipsoidal cluster generators, some noisy data may be included. The clustering results are shown in Fig. 6.

From this figure, it can be seen that NegMM is better than the compared six methods with respect to the three measures, because in each line chart most of the red line is at the top. In Fig. 6(a), the clusterings are measured by *CA*, and NegMM generates better results than the compared methods on 102 of 160 datasets. In Fig. 6(b), the clusterings are measured by Hubert's Γ statistic, and NegMM outperforms the compared methods on 116 of 160 datasets. In Fig. 6(c), the clusterings are measured by NMI, and NegMM also outperforms the compared methods on 116 of 160 datasets. Although Hubert's Γ statistic and NMI suggest that NegMM outperforms the compared methods on the same number of datasets, 4 different ones exist between the two groups of 116 datasets. This means the two measures are similar.

4.4.2. Compared to the selected internal validity indices

In the above sub-section, the overall performance of the proposed method has been compared to some well-known ensemble approaches. In this sub-section, we focus on evaluating the proposed internal validity index MM. As mentioned previously, MM is designed to measure the clusterings produced by clustering ensemble algorithms, and it only uses the information of the base partitions, rather than that of the original dataset.

For each dataset, the proposed clustering ensemble algorithm first produces 50 candidate clusterings, and then selects the final one by MM. In the comparison, we measure the candidates by CA

[1] The Matlab code is available at https://github.com/zhongcaiming/clustering-ensemble.

[2] https://personalpages.manchester.ac.uk/staff/Julia.Handl/data.tar.gz.

Table 4
The qualities of the clustering results are measured by **Hubert's Γstatistic**. The highest quality of clustering results in each row is highlighted as the bold item(s). The numbers in the brackets are the corresponding standard deviations.

Dataset	Best candidate	Clustering ensemble algorithms						
		NegMM	WCT	WTQ	CSM	CSPA	HGPA	MCLA
Path based	0.96	**0.95** (0.01)	0.80 (0.02)	0.75 (0.02)	0.70 (0.02)	0.80 (0.05)	0.80 (0.01)	0.60 (0.11)
Spiral	0.32	**0.23** (0.05)	0.03 (0.05)	0.02 (0.05)	0.01 (0.03)	0.01 (0.03)	0.22 (0.07)	0.06 (0.09)
Toy problem	1.00	**1.00** (0.00)	0.23 (0.10)	0.57 (0.03)	0.15 (0.05)	0.26 (0.00)	0.18 (0.00)	0.30 (0.01)
Flame	0.96	**0.92** (0.01)	0.86 (0.02)	0.85 (0.01)	0.81 (0.05)	0.42 (0.01)	0.52 (0.05)	0.47 (0.02)
Aggregation	1.00	**0.99** (0.00)	0.67 (0.01)	0.59 (0.03)	0.67 (0.00)	0.54 (0.02)	0.64 (0.01)	0.56 (0.02)
D31	0.97	**0.95** (0.00)	0.94 (0.00)	0.94 (0.01)	0.94 (0.01)	**0.95** (0.00)	0.88 (0.02)	**0.95** (0.00)
R15	1.00	**1.00** (0.00)	**1.00** (0.00)	**1.00** (0.00)	0.99 (0.02)	**1.00** (0.00)	**1.00** (0.00)	**1.00** (0.00)
S1	1.00	**0.99** (0.00)	**0.99** (0.00)	**0.99** (0.00)	**0.99** (0.00)	0.93 (0.01)	0.83 (0.07)	0.94 (0.01)
Iris	0.97	0.89 (0.04)	0.83 (0.04)	0.86 (0.04)	0.78 (0.03)	0.90 (0.05)	**0.91**(0.00)	0.89 (0.05)
Ionosphere	0.35	**0.23** (0.06)	0.12 (0.04)	0.17 (0.01)	0.07 (0.03)	0.11 (0.01)	0.13 (0.02)	0.17 (0.04)
Wine	0.50	**0.40** (0.00)	0.00 (0.01)	0.00 (0.02)	0.00 (0.01)	0.39 (0.00)	**0.40** (0.00)	**0.40** (0.00)
Diabetes	0.05	**0.02** (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
Segmentation	0.50	0.47(0.16)	**0.54** (0.04)	0.47 (0.02)	0.53 (0.04)	0.53 (0.05)	0.51 (0.05)	0.51 (0.07)
Glass	0.31	**0.23** (0.01)	0.18 (0.01)	0.19 (0.03)	0.18 (0.02)	0.18 (0.01)	0.19 (0.02)	0.17 (0.04)
WDBC	0.69	**0.63** (0.04)	0.21 (0.03)	0.32 (0.02)	0.18 (0.05)	0.27 (0.08)	0.42 (0.00)	0.39 (0.04)
WPBC	0.01	0.00 (0.00)	**0.02** (0.00)	**0.02** (0.00)	**0.02** (0.00)	0.00 (0.00)	0.01 (0.00)	**0.02** (0.00)

Table 5
The qualities of the clustering results are measured by **NMI**. The highest quality of clustering results in each row is highlighted as the bold item(s). The numbers in the brackets are the corresponding standard deviations.

Dataset	Best candidate	Clustering ensemble algorithms						
		NegMM	WCT	WTQ	CSM	CSPA	HGPA	MCLA
Path based	0.95	**0.93** (0.01)	0.79 (0.02)	0.80 (0.01)	0.70 (0.02)	0.56 (0.02)	0.58 (0.01)	0.56 (0.10)
Spiral	0.35	0.34 (0.03)	0.03 (0.04)	0.02 (0.04)	0.01 (0.03)	**0.56** (0.03)	0.56 (0.06)	0.55 (0.09)
Toy problem	1.00	**1.00** (0.00)	0.28 (0.11)	0.57 (0.03)	0.36 (0.05)	0.47 (0.01)	0.48 (0.00)	0.47 (0.01)
Flame	0.90	**0.86** (0.01)	0.83 (0.02)	0.80 (0.01)	0.79 (0.03)	0.45 (0.01)	0.46 (0.05)	0.45 (0.02)
Aggregation	1.00	**0.99** (0.00)	0.83 (0.01)	0.79 (0.03)	0.83 (0.00)	0.69 (0.02)	0.71 (0.01)	0.71 (0.02)
D31	0.97	**0.97** (0.00)	0.96 (0.00)	0.96 (0.01)	0.96 (0.01)	0.91 (0.00)	0.90 (0.02)	0.92 (0.00)
R15	1.00	**1.00** (0.00)	**1.00** (0.00)	**1.00** (0.00)	0.99 (0.02)	0.93 (0.00)	0.93 (0.00)	0.93 (0.00)
S1	1.00	**0.99** (0.00)	**0.99** (0.00)	**0.99** (0.00)	**0.99** (0.00)	0.80 (0.01)	0.78 (0.07)	0.81 (0.01)
Iris	0.90	**0.88** (0.04)	0.85 (0.04)	0.86 (0.04)	0.81 (0.03)	0.64 (0.05)	0.65(0.00)	0.65 (0.05)
Ionosphere	0.30	0.18 (0.06)	0.13 (0.04)	0.13 (0.01)	0.11 (0.03)	**0.49** (0.01)	0.43 (0.02)	0.48 (0.04)
Wine	0.50	0.40 (0.00)	0.39 (0.01)	0.38 (0.02)	0.36 (0.01)	0.62 (0.00)	**0.65** (0.00)	**0.65** (0.00)
Diabetes	0.05	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	**0.46** (0.00)	**0.46** (0.00)	**0.46** (0.00)
Segmentation	0.64	0.62(0.16)	0.60 (0.04)	0.61 (0.02)	0.57 (0.04)	**0.68** (0.05)	**0.68** (0.05)	0.67 (0.07)
Glass	0.39	0.37 (0.01)	0.33 (0.01)	0.32 (0.03)	0.33 (0.02)	0.73 (0.01)	0.72 (0.02)	**0.75** (0.04)
WDBC	0.62	**0.54** (0.04)	0.29 (0.03)	0.37 (0.02)	0.27 (0.05)	0.45 (0.08)	0.47 (0.00)	0.46 (0.04)
WPBC	0.01	0.00 (0.00)	0.03 (0.00)	0.03 (0.00)	0.03 (0.00)	0.48 (0.00)	**0.50** (0.00)	0.48(0.00)

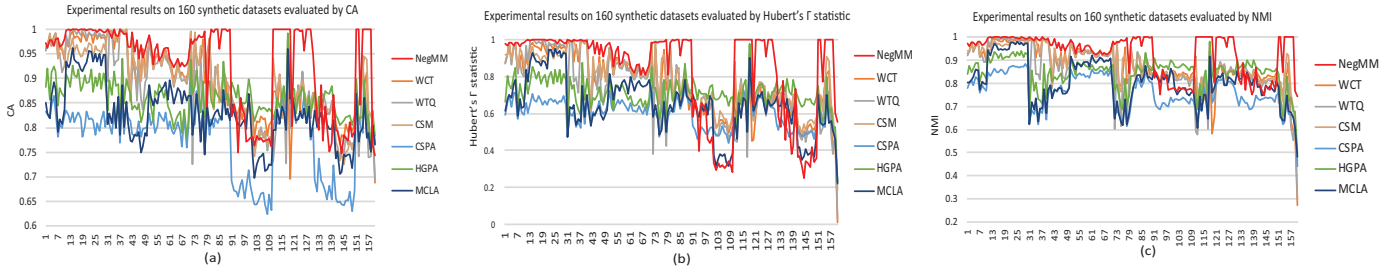

Fig. 6. Clustering results on 160 synthetic datasets.

as the baseline, and use MM and the 7 indices in Table 1 to select the final clusterings. The qualities of the selected clusterings are measured with CA. The results are shown in Table 7, where the number of the nearest neighbours is set to 3 for both MM and CVNN.

In Table 7, the second column is the same as that in Table 3, and is regarded as the baseline. The qualities of this are better than or equal to those of the selected clusterings by the eight internal indices. Among the 8 indices, MM has the best performance on 14 of 16 datasets, while the other indices have the best performance on less than or equal to 9 datasets.

For the Aggregation dataset (D31, R15, S1, Diabetes, and WPBC), the 8 tested internal validity indices have the same performance, and the best clusterings are detected. However, for these 6 datasets, the quality differences between the worst and best candidates generated by Algorithm 2 are very small. This means the performance of the internal validity indices cannot be discriminated by the 6 datasets.

For the Path based dataset, there are six indices: MM, Silhouette, DB, Dunn, S_Dbw, and $\mathcal{I}$. These can almost distinguish the best clusterings, which represent the clustering structure; however, CH and CVNN cannot. For Spiral, three indices find the clusterings

Table 6
The name list of 160 synthetic datasets.

No.	Dataset name	No.	Dataset name	No.	Dataset name	No.	Dataset name
1	10d-10c-no0.dat	41	2d-10c-no0.dat	81	ellipsoid.100d10c.1.dat	121	ellipsoid.50d10c.1.dat
2	10d-10c-no1.dat	42	2d-10c-no1.dat	82	ellipsoid.100d10c.10.dat	122	ellipsoid.50d10c.10.dat
3	10d-10c-no2.dat	43	2d-10c-no2.dat	83	ellipsoid.100d10c.2.dat	123	ellipsoid.50d10c.2.dat
4	10d-10c-no3.dat	44	2d-10c-no3.dat	84	ellipsoid.100d10c.3.dat	124	ellipsoid.50d10c.3.dat
5	10d-10c-no4.dat	45	2d-10c-no4.dat	85	ellipsoid.100d10c.4.dat	125	ellipsoid.50d10c.4.dat
6	10d-10c-no5.dat	46	2d-10c-no5.dat	86	ellipsoid.100d10c.5.dat	126	ellipsoid.50d10c.5.dat
7	10d-10c-no6.dat	47	2d-10c-no6.dat	87	ellipsoid.100d10c.6.dat	127	ellipsoid.50d10c.6.dat
8	10d-10c-no7.dat	48	2d-10c-no7.dat	88	ellipsoid.100d10c.7.dat	128	ellipsoid.50d10c.7.dat
9	10d-10c-no8.dat	49	2d-10c-no8.dat	89	ellipsoid.100d10c.8.dat	129	ellipsoid.50d10c.8.dat
10	10d-10c-no9.dat	50	2d-10c-no9.dat	90	ellipsoid.100d10c.9.dat	130	ellipsoid.50d10c.9.dat
11	10d-20c-no0.dat	51	2d-20c-no0.dat	91	ellipsoid.100d20c.1.dat	131	ellipsoid.50d20c.1.dat
12	10d-20c-no1.dat	52	2d-20c-no1.dat	92	ellipsoid.100d20c.10.dat	132	ellipsoid.50d20c.10.dat
13	10d-20c-no2.dat	53	2d-20c-no2.dat	93	ellipsoid.100d20c.2.dat	133	ellipsoid.50d20c.2.dat
14	10d-20c-no3.dat	54	2d-20c-no3.dat	94	ellipsoid.100d20c.3.dat	134	ellipsoid.50d20c.3.dat
15	10d-20c-no4.dat	55	2d-20c-no4.dat	95	ellipsoid.100d20c.4.dat	135	ellipsoid.50d20c.4.dat
16	10d-20c-no5.dat	56	2d-20c-no5.dat	96	ellipsoid.100d20c.5.dat	136	ellipsoid.50d20c.5.dat
17	10d-20c-no6.dat	57	2d-20c-no6.dat	97	ellipsoid.100d20c.6.dat	137	ellipsoid.50d20c.6.dat
18	10d-20c-no7.dat	58	2d-20c-no7.dat	98	ellipsoid.100d20c.7.dat	138	ellipsoid.50d20c.7.dat
19	10d-20c-no8.dat	59	2d-20c-no8.dat	99	ellipsoid.100d20c.8.dat	139	ellipsoid.50d20c.8.dat
20	10d-20c-no9.dat	60	2d-20c-no9.dat	100	ellipsoid.100d20c.9.dat	140	ellipsoid.50d20c.9.dat
21	10d-40c-no0.dat	61	2d-40c-no0.dat	101	ellipsoid.100d40c.1.dat	141	ellipsoid.50d40c.1.dat
22	10d-40c-no1.dat	62	2d-40c-no1.dat	102	ellipsoid.100d40c.10.dat	142	ellipsoid.50d40c.10.dat
23	10d-40c-no2.dat	63	2d-40c-no2.dat	103	ellipsoid.100d40c.2.dat	143	ellipsoid.50d40c.2.dat
24	10d-40c-no3.dat	64	2d-40c-no3.dat	104	ellipsoid.100d40c.3.dat	144	ellipsoid.50d40c.3.dat
25	10d-40c-no4.dat	65	2d-40c-no4.dat	105	ellipsoid.100d40c.4.dat	145	ellipsoid.50d40c.4.dat
26	10d-40c-no5.dat	66	2d-40c-no5.dat	106	ellipsoid.100d40c.5.dat	146	ellipsoid.50d40c.5.dat
27	10d-40c-no6.dat	67	2d-40c-no6.dat	107	ellipsoid.100d40c.6.dat	147	ellipsoid.50d40c.6.dat
28	10d-40c-no7.dat	68	2d-40c-no7.dat	108	ellipsoid.100d40c.7.dat	148	ellipsoid.50d40c.7.dat
29	10d-40c-no8.dat	69	2d-40c-no8.dat	109	ellipsoid.100d40c.8.dat	149	ellipsoid.50d40c.8.dat
30	10d-40c-no9.dat	70	2d-40c-no9.dat	110	ellipsoid.100d40c.9.dat	150	ellipsoid.50d40c.9.dat
31	10d-4c-no0.dat	71	2d-4c-no0.dat	111	ellipsoid.100d4c.1.dat	151	ellipsoid.50d4c.1.dat
32	10d-4c-no1.dat	72	2d-4c-no1.dat	112	ellipsoid.100d4c.10.dat	152	ellipsoid.50d4c.10.dat
33	10d-4c-no2.dat	73	2d-4c-no2.dat	113	ellipsoid.100d4c.2.dat	153	ellipsoid.50d4c.2.dat
34	10d-4c-no3.dat	74	2d-4c-no3.dat	114	ellipsoid.100d4c.3.dat	154	ellipsoid.50d4c.3.dat
35	10d-4c-no4.dat	75	2d-4c-no4.dat	115	ellipsoid.100d4c.4.dat	155	ellipsoid.50d4c.4.dat
36	10d-4c-no5.dat	76	2d-4c-no5.dat	116	ellipsoid.100d4c.5.dat	156	ellipsoid.50d4c.5.dat
37	10d-4c-no6.dat	77	2d-4c-no6.dat	117	ellipsoid.100d4c.6.dat	157	ellipsoid.50d4c.6.dat
38	10d-4c-no7.dat	78	2d-4c-no7.dat	118	ellipsoid.100d4c.7.dat	158	ellipsoid.50d4c.7.dat
39	10d-4c-no8.dat	79	2d-4c-no8.dat	119	ellipsoid.100d4c.8.dat	159	ellipsoid.50d4c.8.dat
40	10d-4c-no9.dat	80	2d-4c-no9.dat	120	ellipsoid.100d4c.9.dat	160	ellipsoid.50d4c.9.dat

Table 7
The clusterings are selected by different internal validity indices from the candidates produced by Algorithm 2, and the qualities of the selected clustering are measured by CA. The highest quality of clustering results in each row is highlighted as the bold item(s). The numbers in the brackets are the corresponding standard deviations.

Dataset	Best candidate	Internal validity indices							
		MM	Silhouette	DB	CH	Dunn	S_Dbw	CVNN	$\mathcal{I}$
Path based	0.99 (0.01)	**0.98** (0.02)	**0.98** (0.03)	0.97 (0.04)	0.89 (0.13)	**0.98** (0.01)	**0.98** (0.02)	0.93 (0.04)	**0.98** (0.03)
Spiral	0.66 (0.01)	0.63 (0.03)	0.54 (0.05)	0.50 (0.03)	0.58 (0.04)	0.52 (0.04)	0.62 (0.06)	**0.64** (0.04)	0.53 (0.04)
Toy problem	1.00 (0.00)	**1.00** (0.00)	0.88 (0.05)	0.90 (0.07)	0.85 (0.08)	0.83 (0.05)	0.91 (0.06)	0.91 (0.06)	0.90 (0.07)
Flame	0.98 (0.00)	**0.98** (0.00)	0.62 (0.01)	0.62 (0.01)	0.88 (0.12)	0.62 (0.01)	**0.98** (0.00)	0.74 (0.04)	0.62 (0.01)
Aggregation	1.00 (0.00)	**1.00** (0.00)	**1.00** (0.00)	**1.00** (0.00)	**1.00** (0.00)	**1.00** (0.00)	**1.00** (0.00)	**1.00** (0.00)	**1.00** (0.00)
D31	0.98 (0.00)	**0.98** (0.00)	**0.98** (0.00)	**0.98** (0.00)	**0.98** (0.00)	**0.98** (0.00)	**0.98** (0.00)	**0.98** (0.00)	**0.98** (0.00)
R15	1.00 (0.00)	**1.00** (0.00)	**1.00** (0.00)	**1.00** (0.00)	**1.00** (0.00)	**1.00** (0.00)	**1.00** (0.00)	**1.00** (0.00)	**1.00** (0.00)
S1	1.00 (0.00)	**0.99** (0.00)	**0.99** (0.01)	**0.99** (0.01)	**0.99** (0.01)	**0.99** (0.01)	**0.99** (0.01)	**0.99** (0.01)	**0.99** (0.01)
Iris	0.98 (0.00)	**0.96** (0.04)	0.87 (0.06)	0.88 (0.05)	0.88 (0.04)	0.86 (0.06)	0.87 (0.06)	0.89 (0.07)	0.87 (0.07)
Ionosphere	0.88 (0.01)	**0.83** (0.06)	0.70 (0.06)	0.72 (0.09)	0.66 (0.01)	0.67 (0.06)	0.67 (0.05)	0.80 (0.09)	0.72 (0.09)
Wine	0.73 (0.00)	**0.72** (0.00)	**0.72** (0.01)	0.71 (0.01)	**0.72** (0.00)	0.71 (0.01)	**0.72** (0.01)	0.71 (0.02)	**0.72** (0.01)
Diabetes	0.65 (0.00)	**0.65** (0.00)	**0.65** (0.00)	**0.65** (0.00)	**0.65** (0.00)	**0.65** (0.00)	**0.65** (0.00)	**0.65** (0.00)	**0.65** (0.00)
Segmentation	0.69 (0.01)	**0.64** (0.01)	**0.64** (0.01)	**0.64** (0.02)	0.60 (0.01)	0.63 (0.01)	0.63 (0.01)	0.63 (0.01)	0.62 (0.01)
Glass	0.62 (0.00)	**0.62** (0.00)	0.61 (0.00)	0.61 (0.00)	0.60 (0.01)	0.61 (0.01)	0.60 (0.02)	0.60 (0.01)	0.61 (0.01)
WDBC	0.90 (0.02)	0.86 (0.04)	0.82 (0.02)	0.82 (0.02)	0.86 (0.04)	0.82 (0.02)	0.88 (0.01)	**0.90** (0.02)	0.82 (0.02)
WPBC	0.76 (0.00)	**0.76** (0.00)	**0.76** (0.00)	**0.76** (0.00)	**0.76** (0.00)	**0.76** (0.00)	**0.76** (0.00)	**0.76** (0.00)	**0.76** (0.00)

that are close to the best. For Flame, index, MM, and S_Dbw display a good performance. For the Toy problem, Iris, Ionosphere, and Glass datasets, only MM can find the best clustering. For WDBC, MM is slightly worse than CVNN and S_Dbw.

4.5. Number of nearest neighbours for MM and CVNN

In the proposed internal validity index MM, the nearest neighbours are used to depict the density information. However, how to

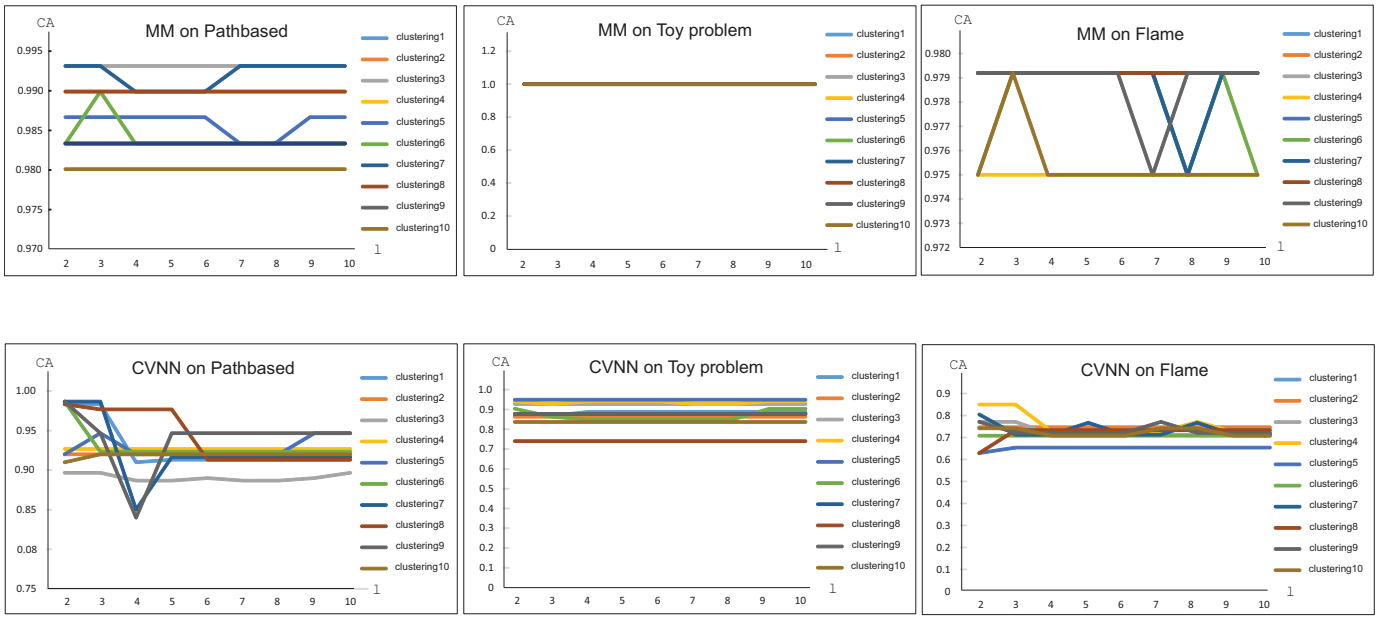

Fig. 7. Index MM and CVNN are tested with different numbers (2–10) of nearest neighbours.

select a suitable number of the nearest neighbours is not a trivial task. In fact, index CVNN has the same problem. In this subsection, we first test how the number of the nearest neighbours affects the performance of MM and CVNN, and then suggest a preferred number.

Three datasets are selected to test the effect of the number of nearest neighbours: Path based, Toy problem, and Flame. These are selected because for each of these datasets, the difference in CA values between the best and the worst candidate clustering is relatively significant; hence, the effect of different numbers on nearest neighbours could be disclosed more easily.

For each dataset, 2 to 10 nearest neighbours are tested, and NegMM runs 10 times for each one. The results are shown in Fig. 7. From the figure, different numbers of nearest neighbours have a slight effect on the performance of the MM index. When the number of the nearest neighbours is set to 3, the performance of index MM is relatively better. However, for index CVNN, the effects are more obvious compared to MM. For example, using the Path based dataset, when the number is changed from 2 to 10 the biggest difference (clustering 6) of CA is less than 0.01 for MM, but almost 0.15 for CVNN. Similarly, when the number is set to 2 or 3, the performance of CVNN is relatively better. Therefore, in the experiments we set the number of nearest neighbours to 3.

4.6. Scalability of the proposed method

From the analysis of the computational complexity in Section 3.6, we can see the scalability of the proposed method is determined by two main factors: the scalability of both Ncut and MST. In the literature, the scalability of the two problems have been intensively studied [41,42,54].

In [41], Cai and Chen proposed a landmark-based spectral clustering, in which $p(\ll n)$ representative objects are selected as the landmarks and represent the original objects as sparse linear combinations of these landmarks. The spectral embedding of the dataset can then be efficiently obtained with the landmark-based representation. The computational cost is linear with N. Jia et al. presented an approximate normalised cut without the Eigen-decomposition method [54], of which the computational complexity is $O(m^3 + m^2 * N + m * N * K * t)$, where $m \ll N$, and t is the maximum iteration number. Therefore, Ncut can be speeded up to less than $O(N^2)$.

Zhong et al. employed a divide-and-conquer strategy to design a fast MST method [42], and its time complexity is $O(N^{1.5})$.

It is evident that the proposed method in this paper can scale up to large-scale datasets if the above methods are used.

5. Conclusion

To improve clustering performance, co-association matrix-based clustering ensemble algorithms usually refine this matrix by mining some hidden information of the base partitions and fusing them back to the matrix. In this paper, we aimed to achieve the same goal, but approached it from the opposite direction, to remove some information from the co-association matrix. In fact, adding extra positive information into the matrix or removing negative evidences out of the matrix has the same effect; both approaches can make the matrix depict the clustering structure accurately.

As a co-association matrix describes the frequency of a pair being in the same base cluster, adding positive information means increasing the frequency of a pair of data points that are in the same ground truth cluster, while removing negative evidences means decreasing the frequency of a pair of data points that are in different ground truth clusters.

Although it is difficult to find the negative evidences directly, it can be observed that negative evidences have relatively low values in the co-association matrix. Therefore, an ensemble scheme is designed in this paper to generate multiple clustering candidates by setting multiple level frequencies of the co-association matrix to zero.

When the multiple clustering candidates are generated, the best one should be selected as the final clustering. This can only be achieved by employing a certain internal validity index. Because in some clustering ensemble scenarios the original dataset information is not available, we design an internal validity index MM, which only uses the information of the co-association matrix (but not of the original dataset). The experimental results indicate that the proposed method is effective.

Future work could be focused on the relation between removing negative evidences and data transformation, and simultaneously considering removing negative and adding positive evidences in the same ensemble scheme. For the former topic, removing multiple level frequencies is similar to tuning the parameters in a Gaussian transformation, as both change the nearest neighbour bound of a data point. For the latter topic, the hybrid of removing negative and adding positive evidences could intuitively be more effective for detecting the clustering structure.

Robustness against noise is also an important performance metric of a clustering algorithm [55]. However, for a clustering ensemble scheme, this performance is almost determined by the base partitions. While the main contribution of this paper is to claim that removal of negative information from the co-association matrix may lead to a good clustering, in future work we will focus on robustness against noise for a clustering ensemble.

Acknowledgements

The work was supported by Natural Science Foundation of China (No. 61175054, 61573235), and sponsored by K.C. Wong Magna Fund in https://doi.org/10.13039/501100004387.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.patcog.2019.03.020.

References

[1] A. Adolfsson, M. Ackerman, N. Brownstein, To cluster, or not to cluster, An analysis of clusterability methods, Pattern Recognition 88 (2019) 13–26.
[2] L. Bai, X. Cheng, J. Liang, H. Shen, Y. Guo, Fast density clustering strategies based on the k-means algorithm, Pattern Recognit. 71 (2017) 375–386.
[3] P. Zhu, W. Zhu, Q. Hu, C. Zhang, W. Zuo, Subspace clustering guided unsupervised feature selection, Pattern Recognit. 66 (2017) 364–374.
[4] Y. Qin, Z.L. Yu, C.D. Wang, Z. Gu, Y. Li, A novel clustering method based on hybrid k-nearest-neighbor graph, Pattern Recognit. 74 (2018) 1–14.
[5] M. Brbić, I. Kopriva, Multi-view low-rank sparse subspace clustering, Pattern Recognit. 3 (2018) 247–258.
[6] L. Houthuys, R. Langone, J. Suykens, Multi-view kernel spectral clustering, Inform. Fusion 44 (2018) 46–56.
[7] X. Wang, Z. Lei, X. Guo, C. Zhang, H. Shi, S. Li, Multi-view subspace clustering with intactness-aware similarity, Pattern Recognit. 88 (2019) 50–63.
[8] I. Maraziotis, S. Perantonis, A. Dragomir, D. Thanos, K-Nets: clustering through nearest neighbors networks, Pattern Recognit. 88 (2019) 470–481.
[9] S. Huang, Z. Kang, I. Tsang, Z. Xu, Auto-weighted multi-view clustering via kernelized graph learning, Pattern Recognit. 88 (2019) 174–184.
[10] L. Huang, H. Chao, C. Wang, Multi-view intact space clustering, Pattern Recognit. 86 (2019) 344–353.
[11] R. Xu, D.W.I. I., Survey of clustering algorithms, IEEE Trans. Neural Netw. 16 (2005) 645–678.
[12] D. Huang, J. Lai, C.D. Wang, Ensemble clustering using factor graph, Pattern Recognit. 50 (2016) 131–142.
[13] D. Huang, J. Lai, C.D. Wang, Robust ensemble clustering using probability trajectories, IEEE Trans. Knowl. Data Eng. 28 (2016) 1312–1326.
[14] D. Huang, C.D. Wang, J. Lai, Locally weighted ensemble clustering, IEEE Trans. Cybern. 48 (2018) 1460–1473.
[15] X. Zhao, J. Liang, C. Dang, Clustering ensemble selection for categorical data based on internal validity indices, Pattern Recognit 69 (2017) 150–168.
[16] E. Ivannikova, H. Park, T. Hämäläinen, K. Lee, Revealing community structures by ensemble clustering using group diffusion, Inform. Fusion 42 (2018) 24–36.
[17] N. Sandes, A. Coelho, Clustering ensembles: a hedonic game theoretical approach, Pattern Recognit. 81 (2018) 95–111.
[18] S. Hadjitodorov, L. Kuncheva, L. Todorova, Moderate diversity for better cluster ensembles, Inform. Fusion 7 (2006) 264–275.
[19] N. Iam-On, T. Boongoen, S. Garrett, C. Price, A link-based approach to the cluster ensemble problem, IEEE Trans. Pattern Anal. Mach. Intell. 33 (2011) 2396–2409.
[20] A.L.N. Fred, A.K. Jain, Combining multiple clusterings using evidence accumulation, IEEE Trans. Pattern Anal. Mach. Intell. 27 (2005) 835–850.
[21] J. Wu, H. Liu, H. Xiong, J. Cao, J. Chen, K-Means-based consensus clustering: a unified view, IEEE Trans. Knowl. Data Eng. 27 (2015) 1041–4347.
[22] C. Zhong, X. Yue, Z. Zhang, J. Lei, A clustering ensemble: two-level-refined co-association matrix with path-based transformation, Pattern Recognit. 48 (2015) 2699–2709.
[23] H.S. Yoon, S.Y. Ahn, S.H. Lee, S.B. Cho, J. Kim, Heterogeneous Clustering Ensemble Method for Combining Different Cluster Results, Workshop on Data Mining for Biomedical Applications, 2006.
[24] Z. Yu, H.S. Wong, H. Wang, Graph-based consensus clustering for class discovery from gene expression data, Bioinformatics 23 (2007) 2888–2896.
[25] A. Strehl, J. Ghosh, Cluster ensembles - a knowledge reuse framework for combining multiple partitions, J. Mach. Learn. Res. 3 (2003) 583–617.
[26] H. Liu, R. Zhao, H. Fang, F. Cheng, Y. Fu, Y.Y. Liu, Entropy-based consensus clustering for patient stratification, Bioinformatics (2017) 1–8, doi:10.1093/bioinformatics/btx167.
[27] J. Bezdek, N. Pal, Some New Indexes of Cluster Validity, in: IEEE Transactions on Systems, Man, and Cybernetics, Part B, volume 28, 1998, pp. 301–315.
[28] Y. Ren, C. Domeniconi, G. Zhang, G. Yu, Weighted-object ensemble clustering: methods and analysis, Knowl. Inf. Syst. 51 (2017) 661–689.
[29] R. Hathaway, J. Bezdek, J. Huband, Scalable visual assessment of cluster tendency for large data sets, Pattern Recognit. 39 (2006) 1315–1324.
[30] J. Bezdek, R. Hathaway, J. Huband, Visual assessment of clustering tendency for rectangular dissimilarity matrices, IEEE Trans. Fuzzy Syst. 15 (2007) 890–903.
[31] L. Wang, C. Leckie, T. Havens, K. Ramamohanarao, J. Bezdek, Automatically determining the number of clusters in unlabeled data sets, IEEE Trans. Knowl. Data Eng. 21 (2009) 335–350.
[32] T. Havens, J. Bezdek, An effcient formulation of the improved visual assessment of cluster tendency (iVAT) algorithm, IEEE Trans. Knowl. Data Eng. 24 (2012) 813–822.
[33] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 888–905.
[34] M. Girolami, S. Rogers, A first course in machine learning, 2nd, Taylor & Francis Group, 6000 Broken Sound Parkway NW, Suite 300, 2017.
[35] D. Davies, D. Bouldin, A clustering separation measure, IEEE Trans. Pattern Anal. Mach. Intell. 1 (1979) 224–227.
[36] J. Dunn, A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters, J. Cybern. 3 (1973) 32–57.
[37] Y. Liu, Z. Li, H. Xiong, X. Gao, J. Wu, S. Wu, Understanding and enhancement of internal clustering validation measures, IEEE Trans. Cybern. 43 (2013) 982–994.
[38] H. Chang, D.Y. Yeung, Robust path-based spectral clustering, Pattern Recognit. 41 (2008) 191–203.
[39] C. Zhong, X. Yue, J. Lei, Visual hierarchical cluster structure: a refined co-association matrix based visual assessment of cluster tendency, Pattern Recognit. Lett. 59 (2015) 48–55.
[40] O. Chapelle, J. Weston, B. Schölkopf, Cluster Kernels for Semi-supervised Learning, in: Proceedings of the 2002 Neural Information Processing Systems, 2002.
[41] D. Cai, X. Chen, Large scale spectral clustering via landmark-based sparse representation, IEEE Trans. Cybern. 45 (2015) 1669–1680.
[42] C. Zhong, M. Malinen, D. Miao, P. Fränti, Fast minimum spanning tree algorithm based on k-means, Inf. Sci. (Ny) 295 (2015) 1–17.
[43] P. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, J. Comput. Appl. Math. 20 (2015) 53–65.
[44] T. Calinski, J. Harabasz, A dendrite method for cluster analysis, Commun. Stat. 3 (1974) 1–27.
[45] M. Halkidi, M. Vazirgiannis, Clustering Validity Assessment: Finding the Optimal Partitioning of a Data Set, in: Proceedings of the First IEEE International Conference on Data Mining (ICDM'01), California, USA, 2001.
[46] U. Maulik, S. Bandyopadhyay, Performance evaluation of some clustering algorithms and validity indices, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 1650–1654.
[47] S. Theodoridis, K. Koutroumbas, Pattern Recognit., 4th, Elsevier (USA), 525 B Street, Suite 1900, San Diego, CA 92101-4495, USA, 2009.
[48] A. Jain, M. Law, Data Clustering: A User's Dilemma, Pattern Recognition and Machine Intelligence, 2005.
[49] L. Fu, E. Medico, Flame, a novel fuzzy clustering method for the analysis of dna microarray data, in: BMC bioinformatics, volume 8.
[50] A. Gionis, H. Mannila, P. Tsaparas, Clustering aggregation, ACM Trans. Knowl. Discov. Data 1 (2007) 1–30.
[51] C. Veenman, M. Reinders, E. Backer, A maximum variance cluster algorithm, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 1273–1280.
[52] P. Fränti, O. Virmajoki, Iterative shrinking method for clustering problems, Pattern Recognit. 39 (2006) 761–775.
[53] A. Asuncion, D. Newman, UCI machine learning repository. http://www.ics.uci.edu/~mlearn/MLRepository.html.
[54] H. Lia, S. Ding, M. Du, Y. Xue, Approximate normalized cuts without eigen-decomposition, Inf. Sci. (Ny) 374 (2016) 135–150.
[55] R.C. Amorim, C. Hennig, Recovering the number of clusters in data sets with noise features using feature rescaling factors, Inf. Sci. (Ny) 324 (2015) 126–145.

Caiming Zhong is a professor in College of Science and Technology, Ningbo University, Ningbo, China. His research interests include cluster analysis, manifold learning and image segmentation.

Lianyu Hu is a master student in Ningbo University, Ningbo, China. His research interests include cluster analysis, machine learning.

Xiaodong Yue received his Ph.D. degree in 2010 from Tongji University. From 2016 he has been an associate professor of Department of Computer Science and Technology, Shanghai University, Shanghai, China. His research interests include image analysis, pattern recognition and machine learning.

Ting Luo is an associate professor in College of Science and Technology, Ningbo University, Ningbo, China. His research interests include image processing, manifold learning.

Qiang Fu is an associate professor in College of Science and Technology, Ningbo University, Ningbo, China. His research interests include soft computing, Swarm intelligence.

Haiyong Xu is an associate professor in College of Science and Technology, Ningbo University, Ningbo, China. His research interests include image processing, manifold learning.