A Sequential Three-Way Approach to Constructing a Co-association Matrix in Consensus Clustering

Mengjun Hu(✉), Xiaofei Deng, and Yiyu Yao

Department of Computer Science, University of Regina,
Regina, SK S4S 0A2, Canada
{hu258,deng200x,yyao}@cs.uregina.ca

Abstract. The main task in consensus clustering is to produce an optimal output clustering based on a set of input clusterings. The co-association matrix based consensus clustering methods are easy to understand and implement. However, they usually have high computational cost with big datasets, which restricts their applications. We propose a sequential three-way approach to constructing the co-association matrix progressively in multiple stages. In each stage, based on a set of input clusterings, we evaluate how likely two data points are associated and accordingly, divide a set of data-point pairs into three disjoint positive, negative and boundary regions. A data-point pair in the positive region is associated with a definite decision of clustering the two data points together. A pair in the negative region is associated with a definite decision of separating the two data points into different clusters. For a pair in the boundary region, we do not have sufficient information to make a definite decision. The decision on such a pair is deferred into the next stage where more input clusterings will be involved. By making quick decisions on early stages, the overall computational cost of constructing the matrix and the consensus clustering may be reduced.

Keywords: Sequential three-way decision · Consensus clustering
Co-association matrix

1 Introduction

Given a set of data points described by a set of attributes or features, the main task of clustering is to divide these data points into groups such that the data points in the same group are as similar as possible and those in different groups are as dissimilar as possible. Each group is called a cluster, and the family of all groups is called a clustering. The results of some popular clustering methods [2,4,5,8,16] depend on their initial configurations that involve a priori parameters such as a given number of clusters. In order to improve the robustness and accuracy, these methods are usually repeatedly applied with different

This work is partially supported by a Discovery Grant from NSERC, Canada.

H. S. Nguyen et al. (Eds.): IJCRS 2018, LNAI 11103, pp. 599–613, 2018.
https://doi.org/10.1007/978-3-319-99368-3_47

initial configurations. The family of produced clusterings are then combined into a single clustering via consensus clustering. This is one of the main motivations for consensus clustering that produces a final clustering by synthesizing a set of input clusterings.

The consensus clustering methods based on co-association matrix [6,7,12,13,21,22] are very popular and well studied in the literature. The first step in the main procedure is to synthesize the set of input clusterings into an $n \times n$ co-association matrix where n is the total number of data points. The values in the matrix reflect how likely the corresponding two data points are clustered together in the input clusterings. The second step is to obtain the final clustering by applying a basic clustering method to the matrix. These consensus clustering methods are easy to understand and implement. However, since they focus on all data-point pairs when constructing the matrix, they usually have high computational cost when applied to large datasets, which restricts their applications.

The consensus clustering can be viewed as a decision making process. In the co-association matrix based methods, we make decisions of whether to cluster two data points together or not based on the information provided by input clusterings. The theory of three-way decisions [23] offers a framework of decision making by dividing a set of objects into three disjoint decision regions according to some criterion. Each region is associated with a specific decision. Generally, the three regions include the positive, negative and boundary regions. The objects in the positive region are associated with an acceptance decision, that is, we accept that these objects satisfy the criterion. The objects in the negative region are associated with a rejection decision, that is, we decide that these objects do not satisfy the criterion. Those in the boundary region cannot be definitely determined to satisfy the criterion or not. They are associated with a third non-commitment decision due to the uncertainty. The theory of three-way decisions has been applied to basic clustering methods by researchers [27–30].

The sequential three-way decision model [26] iteratively applies the three-way decision model to refine the boundary region and reduce the uncertainty. Definite decisions (i.e., acceptance and rejection) are made on objects in each stage if sufficient information is available. Otherwise, the decision on the objects will be postponed into the next stage where more detailed and sufficient information will be involved. It has been applied to many real-world applications such as face recognition in [14,15]. Four modes of sequential three-way decisions are examined in [26], including multiple levels of granularity, probabilistic rough set theory, multiple models of classification, and ensemble classifications. Our presented approach in this paper follows a similar mode as ensemble classifications.

The presented approach integrates the sequential three-way decision model into the construction of a co-association matrix. In each stage, based on a set of input clusterings, we put a data-point pair into a positive region if the corresponding value in the matrix is high enough or into a negative region if the value

is low enough. The corresponding entry in the matrix is then updated with the largest value 1 or the smallest value 0, respectively. Otherwise, the pair is put into a third boundary region and the corresponding entry is to be determined in the next stage that involves more input clusterings. In this way, we determine the entries in the matrix and correspondingly, make quick decisions on the clustering of some data points in early stages. As a result, we may be able to reduce the overall computational cost of constructing the matrix.

The remaining part of this paper is arranged as follows. Section 2 reviews consensus clustering methods based on co-association matrix. The sequential three-way approach to constructing the matrix is presented in Sect. 3. Section 4 shows the experimental results. Section 5 concludes the paper and discusses possible directions for the future work.

2 A Review of Co-association Matrix Based Consensus Clustering Methods

The main task of consensus clustering is to combine different clusterings of a dataset into one single clustering, usually without referring to the original features or attributes of the data points. A general framework of consensus clustering includes two steps [20], namely, the Generation and Consensus steps. The Generation step generates the set of input clusterings for a given dataset. They can be produced by different basic clustering methods or multiple applications of the same method with different parameters. The Consensus step combines the input clusterings into a final consensus clustering according to a particular consensus function.

A co-association matrix based method includes two steps in the main procedure. The first step is to synthesize the input clusterings into an intermediate representation called a co-association matrix. Each entry in the matrix measures how many times the two corresponding data points are associated or clustered together in the input clusterings. The second step is to get the final consensus clustering by applying a basic clustering method to the matrix.

Suppose $X = \{x_1, x_2, \cdots, x_n\}$ is a given dataset and $\mathbb{C}^{in} = \{\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_m\}$ is a set of input clusterings on X. In a co-association matrix based method, an input clustering $\mathcal{C}_k (1 \leq k \leq m)$ is commonly represented by an $n \times n$ matrix. Moreover, the input clusterings are widely assumed to be hard clusterings where a data point belongs to exactly one cluster. Thus, the entries in a matrix $\mathcal{C}_k (1 \leq k \leq m)$ are formally defined as: for $1 \leq i \leq n$ and $1 \leq j \leq n$,

$$\mathcal{C}_k(i,j) = \begin{cases} 1, & \text{if } x_i \text{ and } x_j \text{ are clustered together,} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Based on the set $\mathbb{C}^{in}$, a simple way to construct the co-association matrix $M_{n \times n}$ is to use the proportion of input clusterings where the two corresponding data points are associated, which is the evidence accumulation framework proposed

in [7]. Accordingly, M is constructed as: for $1 \leq i \leq n$ and $1 \leq j \leq n$,

$$M(i,j) = \frac{1}{m} \sum_{k=1}^{m} \mathcal{C}_k(i,j). \tag{2}$$

More complex measures are proposed to construct the matrix by taking into account more information. The Connected-Triple based Similarity (CTS) and SimRank based Similarity (SRS) [12] consider the transitivity property of clustering data points. A Weighted Co-Association Matrix is presented in [21] which takes into consideration the size of the clusters containing the two data points and the total number of clusters in the corresponding input clustering. The Probability Accumulation Matrix [22] considers the size of the clusters containing the two data points and the number of attributes used to describe the data points.

To cluster the data points based on the co-association matrix, two hierarchical clustering methods are proposed in [7,13]. A graph based method proposed in [19] generates a similarity graph based on the matrix and obtains the final clustering by partitioning the graph. Two threshold based methods are presented in [6,7].

The co-association matrix based methods are advantageous in several aspects. They use the co-association idea to avoid the labeling correspondence problem which is a common difficulty in some popular categories of current consensus clustering methods. For instance, in the relabeling and voting based methods [20], the first step is to relabel the input clusters in all the input clusterings where the labeling correspondence problem needs to be solved in order to find the correspondence between clusters in different clusterings. The labeling correspondence problem can only be solved, with certain accuracy, when the input clusterings have the same number of clusters, which is a very restrictive condition in these methods. Besides, the co-association matrix based methods are easy to understand and implement since the constructions of the matrix and the basic clustering methods are usually quite intuitive. However, since they need to compute the value for each data-point pair to construct the co-association matrix, they usually have high computational cost with big datasets, which restricts their applications.

3 A Sequential Three-Way Approach to Constructing a Co-association Matrix

Based on a general framework of sequential three-way decisions proposed in [26], we present a sequential three-way approach to progressively constructing a co-association matrix in multiple stages.

3.1 An (α, β)-cut of a Co-association Matrix

The values in a co-association matrix quantitatively evaluate how likely two data points are clustered together. In order to decide whether two data points should

be clustered together in the final clustering, it may be sufficient to qualitatively know whether they are likely enough to be associated, that is, whether the corresponding value in the matrix is large enough. Similarly, to decide whether they should be separated into different clusters, a qualitatively small enough value may be sufficient. Based on this idea, we can use a pair of thresholds to cut the values and divide the data-point pairs into three decision regions. The matrix is then updated by assigning different values to the pairs in different regions.

Suppose (α, β) is a pair of thresholds with $0 \leq \beta < \alpha \leq 1$ and $eval : X \times X \to [0, 1]$ is a measure to evaluate how likely two data points are associated based on a set of input clusterings (e.g., Eq. (2)). By using the pair (α, β) to cut the evaluation values, the set of data-point pairs $\mathbb{X} = X \times X$ is divided into three disjoint positive POS, negative NEG and boundary BND regions:

$$\begin{aligned} \mathrm{POS}(\mathbb{X}) &= \{(x_i, x_j) \in \mathbb{X} \mid eval(x_i, x_j) \geq \alpha\}, \\ \mathrm{NEG}(\mathbb{X}) &= \{(x_i, x_j) \in \mathbb{X} \mid eval(x_i, x_j) \leq \beta\}, \\ \mathrm{BND}(\mathbb{X}) &= \{(x_i, x_j) \in \mathbb{X} \mid \beta < eval(x_i, x_j) < \alpha\}. \end{aligned} \tag{3}$$

The entries in the co-association matrix $M_{n \times n}$ are accordingly determined as:

(M^{P}) If $(x_i, x_j) \in \mathrm{POS}(\mathbb{X})$, then $M(i, j) = 1$,
(M^{N}) If $(x_i, x_j) \in \mathrm{NEG}(\mathbb{X})$, then $M(i, j) = 0$,
(M^{B}) If $(x_i, x_j) \in \mathrm{BND}(\mathbb{X})$, then $M(i, j) = eval(x_i, x_j)$ or a constant value $v \in (0, 1)$.

As a result, for two data points x_i and x_j, if their evaluation value $eval(x_i, x_j)$ is high enough to indicate that they are associated (i.e., $eval(x_i, x_j) \geq \alpha$), then we cluster them together by assigning the largest evaluation value 1 to the entry $M(i, j)$. If the evaluation value is low enough to indicate that they are not associated (i.e., $eval(x_i, x_j) \leq \beta$), then we separate them into different clusters by assigning the smallest evaluation value 0 to the entry $M(i, j)$. Otherwise, we cannot make a definite decision due to insufficient information. The entry $M(i, j)$ may take the original evaluation value or a default constant value $v \in (0, 1)$ such as 0.5.

3.2 An *l*-stage Sequential Three-Way Approach to Constructing a Co-association Matrix

In the (α, β)-cut discussed in the previous subsection, a definite decision cannot be made on the data-point pairs in the boundary region due to insufficient information provided by the input clusterings. By involving more input clusterings, we may be able to refine the boundary region, which results in a sequential three-way approach to constructing a co-association matrix.

Suppose we have the following sequence of sets of input clusterings:

$$\mathbb{C}_1^{in} \subsetneq \mathbb{C}_2^{in} \subsetneq \cdots \subsetneq \mathbb{C}_l^{in}. \tag{4}$$

The proper subset relationship $\mathbb{C}_k^{in} \subsetneq \mathbb{C}_{k+1}^{in} (1 \leq k < l)$ ensures that $\mathbb{C}_{k+1}^{in}$ contains at least one more input clustering than $\mathbb{C}_k^{in}$, which gives more information about the clustering of data points. By using these sets one by one, we can obtain an l-stage sequential three-way approach to constructing the co-association matrix. Suppose X is the given dataset and $\mathbb{X}_k$ is the set of data-point pairs considered in the kth stage. The three regions in the kth stage are constructed as: let $\mathbb{X}_1 = X \times X$ and $\mathbb{X}_k = \mathrm{BND}_{k-1}(\mathbb{X}_{k-1}) (1 < k \leq l)$,

$$\begin{aligned}
\mathrm{POS}_k(\mathbb{X}_k) &= \{(x_i, x_j) \in \mathbb{X}_k \mid eval(x_i, x_j | \mathbb{C}_k^{in}) \geq \alpha_k\}, \\
\mathrm{NEG}_k(\mathbb{X}_k) &= \{(x_i, x_j) \in \mathbb{X}_k \mid eval(x_i, x_j | \mathbb{C}_k^{in}) \leq \beta_k\}, \\
\mathrm{BND}_k(\mathbb{X}_k) &= \{(x_i, x_j) \in \mathbb{X}_k \mid \beta_k < eval(x_i, x_j | \mathbb{C}_k^{in}) < \alpha_k\},
\end{aligned} \tag{5}$$

where $eval(x_i, x_j | \mathbb{C}_k^{in})$ is the evaluation value of x_i and x_j calculated based on the set $\mathbb{C}_k^{in}$, and the thresholds satisfy the condition $0 \leq \beta_k < \alpha_k \leq 1$. Accordingly, the entries in the co-association matrix $M_{n \times n}$ are determined as follows:

$$\begin{aligned}
&(M_k^{\mathrm{P}}) \ \text{If } (x_i, x_j) \in \mathrm{POS}_k(\mathbb{X}_k), \text{ then } M(i,j) = 1, \\
&(M_k^{\mathrm{N}}) \ \text{If } (x_i, x_j) \in \mathrm{NEG}_k(\mathbb{X}_k), \text{ then } M(i,j) = 0, \\
&(M_k^{\mathrm{B}}) \ \text{If } (x_i, x_j) \in \mathrm{BND}_k(\mathbb{X}_k), \text{ then } M(i,j) = eval(x_i, x_j | \mathbb{C}_k^{in}).
\end{aligned}$$

One may take special actions to deal with a nonempty final boundary region $\mathrm{BND}_l(\mathbb{X}_l)$ instead of using the original evaluation values. For example, one may use a two-way process with a threshold r (e.g., 0.5) to clean up the boundary region or use a fixed value (e.g., 0.5) to replace the original evaluation values.

There are several assumptions in the above sequential three-way approach. Firstly, it is assumed that we are more biased towards putting the data-point pairs into the boundary region in an early stage where limited information is available. It leads to the relationships of all the thresholds [25]: $0 \leq \beta_1 \leq \beta_2 \leq \cdots \leq \beta_l < \alpha_l \leq \alpha_{l-1} \leq \cdots \leq \alpha_1 \leq 1$. By using a more restrictive pair of thresholds in an early stage, a data-point pair is more likely to be put into the boundary region, which indicates a more conservative opinion due to limited information. A third assumption is that we do not go back to update the positive and negative regions constructed in earlier stages. In other words, the definite decisions associated with these regions are not updated although they might be inappropriate when more input clusterings are available in some stage later on. Consequently, in each stage, we only focus on refining the boundary region constructed in the previous stage.

Example 1. We illustrate the construction of a co-association matrix by the presented approach. Suppose the data set is $X = \{o_1, o_2, o_3, o_4, o_5, o_6, o_7, o_8, o_9, o_{10}\}$. The set $\mathbb{C}^{in}$ of all input clusterings on X includes the following ten clusterings:

$$\begin{aligned}
\mathcal{C}_1 &= \big\{\{o_1, o_2, o_8\}, \{o_3, o_9, o_{10}\}, \{o_4, o_6, o_7\}, \{o_5\}\big\},\\
\mathcal{C}_2 &= \big\{\{o_1, o_4, o_6\}, \{o_2, o_5, o_8\}, \{o_3, o_7, o_9, o_{10}\}\big\},\\
\mathcal{C}_3 &= \big\{\{o_1, o_4, o_6\}, \{o_2, o_8\}, \{o_3, o_5, o_9, o_{10}\}, \{o_7\}\big\},\\
\mathcal{C}_4 &= \big\{\{o_1, o_2, o_7, o_8\}, \{o_3, o_5, o_9, o_{10}\}, \{o_4, o_6\}\big\},\\
\mathcal{C}_5 &= \big\{\{o_1, o_2, o_7, o_8\}, \{o_3, o_9, o_{10}\}, \{o_4, o_5, o_6\}\big\},\\
\mathcal{C}_6 &= \big\{\{o_1, o_4, o_6\}, \{o_2, o_3, o_5, o_9\}, \{o_7\}, \{o_8, o_{10}\}\big\},\\
\mathcal{C}_7 &= \big\{\{o_1, o_4, o_6, o_7\}, \{o_2, o_3, o_8\}, \{o_5, o_9, o_{10}\}\big\},\\
\mathcal{C}_8 &= \big\{\{o_1, o_3, o_7, o_9, o_{10}\}, \{o_2, o_8\}, \{o_4, o_6\}, \{o_5\}\big\},\\
\mathcal{C}_9 &= \big\{\{o_1, o_2, o_4\}, \{o_3, o_5, o_9, o_{10}\}, \{o_6, o_7, o_8\}\big\},\\
\mathcal{C}_{10} &= \big\{\{o_1, o_4, o_6\}, \{o_2, o_7, o_8\}, \{o_3, o_5, o_9, o_{10}\}\big\}.
\end{aligned}$$

We use Eq. (2) to calculate the evaluation values, which is a symmetric measure. Thus, we need to compute the entries in the top right half of the matrix, not including the diagonal line. Suppose $\mathbb{C}_1^{in} = \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4, \mathcal{C}_5, \mathcal{C}_6\}$. The evaluation values are given in Table 1(a). By using thresholds $(1, 0)$, the entries with grey background are in the boundary region and the remaining entries are in either the positive or negative region. In stage 2, $\mathbb{C}_2^{in} = \mathbb{C}_1^{in} \cup \{\mathcal{C}_7\}$. The evaluation values for the previous boundary region are modified and given in Table 1(b). By using thresholds $(0.9, 0.1)$, the previous boundary region stays the same. In stage 3, $\mathbb{C}_3^{in} = \mathbb{C}_2^{in} \cup \{\mathcal{C}_8\}$ and the evaluation values are given in Table 1(c). By using thresholds $(0.8, 0.2)$, some entries in the previous boundary region are moved to either the positive or negative region and the corresponding values in the matrix are changed to either 1 or 0. This process goes on with stage 4 using $\mathbb{C}_4^{in} = \mathbb{C}_3^{in} \cup \{\mathcal{C}_9\}$ and thresholds $(0.7, 0.3)$ and stage 5 using $\mathbb{C}_5^{in} = \mathbb{C}^{in}$ and thresholds $(0.6, 0.4)$. If we do not allow overlap between clusters (i.e., we consider the hard clusterings) and assume that two data points are clustered together if they are both clustered together with a third data point, then the nonempty boundary region in stage 5 can be cleaned up and the final consensus clustering is $\big\{\{o_1, o_4, o_6\}, \{o_2, o_8\}, \{o_3, o_5, o_9, o_{10}\}\big\}$.

3.3 Two Issues in the Presented Approach

The first issue in the presented sequential three-way approach is to avoid an easy agreement on a definite decision in early stages where we have limited input clusterings. In other words, the data-point pairs should be less likely to be put into the positive and negative regions in early stages. There are at least two possible solutions to this issue. One solution is to use very restrictive thresholds in early stages, such as $(1, 0)$ in the first few stages. Another solution is to carefully select the input clusterings used in an early stage so that it is not easy

Table 1. The construction of a co-association matrix in Example 1

	o_1	o_2	o_3	o_4	o_5	o_6	o_7	o_8	o_9	o_{10}
o_1		$\frac{3}{6}$	**0**	$\frac{3}{6}$	**0**	$\frac{3}{6}$	$\frac{2}{6}$	$\frac{3}{6}$	**0**	**0**
o_2			$\frac{1}{6}$	**0**	$\frac{2}{6}$	**0**	$\frac{2}{6}$	$\frac{5}{6}$	$\frac{1}{6}$	**0**
o_3				**0**	$\frac{3}{6}$	**0**	$\frac{1}{6}$	**0**	**1**	$\frac{5}{6}$
o_4					$\frac{1}{6}$	**1**	$\frac{1}{6}$	**0**	**0**	**0**
o_5						$\frac{1}{6}$	**0**	$\frac{1}{6}$	$\frac{3}{6}$	$\frac{2}{6}$
o_6							$\frac{1}{6}$	**0**	**0**	**0**
o_7								$\frac{2}{6}$	$\frac{1}{6}$	$\frac{1}{6}$
o_8									**0**	$\frac{1}{6}$
o_9										$\frac{5}{6}$
o_{10}										

(a) Stage 1: $\mathbb{C}_1^{in} = \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4, \mathcal{C}_5, \mathcal{C}_6\}$

	o_1	o_2	o_3	o_4	o_5	o_6	o_7	o_8	o_9	o_{10}
o_1		$\frac{3}{7}$	**0**	$\frac{3+1}{7}$	**0**	$\frac{3+1}{7}$	$\frac{2+1}{7}$	$\frac{3}{7}$	**0**	**0**
o_2			$\frac{1+1}{7}$	**0**	$\frac{2}{7}$	**0**	$\frac{2}{7}$	$\frac{5+1}{7}$	$\frac{1}{7}$	**0**
o_3				**0**	$\frac{3}{7}$	**0**	$\frac{1}{7}$	**0**	**1**	$\frac{5}{7}$
o_4					$\frac{1}{7}$	**1**	$\frac{1+1}{7}$	**0**	**0**	**0**
o_5						$\frac{1}{7}$	**0**	$\frac{1}{7}$	$\frac{3+1}{7}$	$\frac{2+1}{7}$
o_6							$\frac{1+1}{7}$	**0**	**0**	**0**
o_7								$\frac{2}{7}$	$\frac{1}{7}$	$\frac{1}{7}$
o_8									**0**	$\frac{1}{7}$
o_9										$\frac{5+1}{7}$
o_{10}										

(b) Stage 2: $\mathbb{C}_2^{in} = \mathbb{C}_1^{in} \cup \{\mathcal{C}_7\}$

	o_1	o_2	o_3	o_4	o_5	o_6	o_7	o_8	o_9	o_{10}
o_1		$\frac{3}{8}$	**0**	$\frac{4}{8}$	**0**	$\frac{4}{8}$	$\frac{3+1}{8}$	$\frac{3}{8}$	**0**	**0**
o_2			$\frac{2}{8}$	**0**	$\frac{2}{8}$	**0**	$\frac{2}{8}$	$\frac{6+1}{8} \Rightarrow$**1**	$\frac{1}{8} \Rightarrow$**0**	**0**
o_3				**0**	$\frac{3}{8}$	**0**	$\frac{1+1}{8}$	**0**	**1**	$\frac{5+1}{8}$
o_4					$\frac{1}{8} \Rightarrow$**0**	**1**	$\frac{2}{8}$	**0**	**0**	**0**
o_5						$\frac{1}{8} \Rightarrow$**0**	**0**	$\frac{1}{8} \Rightarrow$**0**	$\frac{4}{8}$	$\frac{3}{8}$
o_6							$\frac{2}{8}$	**0**	**0**	**0**
o_7								$\frac{2}{8}$	$\frac{1+1}{8}$	$\frac{1+1}{8}$
o_8									**0**	$\frac{1}{8} \Rightarrow$**0**
o_9										$\frac{6+1}{8} \Rightarrow$**1**
o_{10}										

(c) Stage 3: $\mathbb{C}_3^{in} = \mathbb{C}_2^{in} \cup \{\mathcal{C}_8\}$

	o_1	o_2	o_3	o_4	o_5	o_6	o_7	o_8	o_9	o_{10}
o_1		$\frac{3+1}{9}$	**0**	$\frac{4+1}{9}$	**0**	$\frac{4}{9}$	$\frac{4}{9}$	$\frac{3}{9}$	**0**	**0**
o_2			$\frac{2}{9} \Rightarrow$**0**	**0**	$\frac{2}{9} \Rightarrow$**0**	**0**	$\frac{2}{9} \Rightarrow$**0**	**1**	**0**	**0**
o_3				**0**	$\frac{3+1}{9}$	**0**	$\frac{2}{9} \Rightarrow$**0**	**0**	**1**	$\frac{6+1}{9} \Rightarrow$**1**
o_4					**0**	**1**	$\frac{2}{9} \Rightarrow$**0**	**0**	**0**	**0**
o_5						**0**	**0**	**0**	$\frac{4+1}{9}$	$\frac{3+1}{9}$
o_6							$\frac{2+1}{9}$	**0**	**0**	**0**
o_7								$\frac{2+1}{9}$	$\frac{2}{9} \Rightarrow$**0**	$\frac{2}{9} \Rightarrow$**0**
o_8									**0**	**0**
o_9										**1**
o_{10}										

(d) Stage 4: $\mathbb{C}_4^{in} = \mathbb{C}_3^{in} \cup \{\mathcal{C}_9\}$

	o_1	o_2	o_3	o_4	o_5	o_6	o_7	o_8	o_9	o_{10}
o_1		$\frac{4}{10} \Rightarrow$**0**	**0**	$\frac{5+1}{10} \Rightarrow$**1**	**0**	$\frac{4+1}{10}$	$\frac{4}{10} \Rightarrow$**0**	$\frac{3}{10} \Rightarrow$**0**	**0**	**0**
o_2			**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**
o_3				**0**	$\frac{4+1}{10}$	**0**	**0**	**0**	**1**	**1**
o_4					**0**	**1**	**0**	**0**	**0**	**0**
o_5						**0**	**0**	**0**	$\frac{5+1}{10} \Rightarrow$**1**	$\frac{4+1}{10}$
o_6							$\frac{3}{10} \Rightarrow$**0**	**0**	**0**	**0**
o_7								$\frac{3+1}{10} \Rightarrow$**0**	**0**	**0**
o_8									**0**	**0**
o_9										**1**
o_{10}										

(e) Stage 5: $\mathbb{C}_5^{in} = \mathbb{C}^{in}$

for them to agree on a definite decision. This involves the determination of a proper total number of input clusterings and the selection of the basic clustering methods to generate the input clusterings. Intuitively, the group of input clusterings should be large enough since a small group is more likely to agree on a definite decision. The basic clustering methods that are used to generate the input clusterings should be as various as possible so that we can capture different views of clustering the data points. Repeated applications of the same method, such as k-means, are likely to produce similar clusterings although they start with different initial configurations. We should involve basic clustering methods in various categories, such as density-based clustering methods [5] that model clusters as areas with high density and EM algorithms [2] that model clusters as probability distributions.

The second issue is the determination of thresholds. The computation and interpretation of thresholds have been studied with respect to one-step three-way decisions, such as a probabilistic approach proposed in [24], a game-theoretic approach proposed in [9], and a decision-theoretic approach proposed in [3]. In order to apply these studies in the presented approach, we need to generalize the current methods with respect to the sequential case and the specific topic of consensus clustering.

These two issues can also be empirically solved by tuning related parameters in the experiments. For instance, one may use a fixed decreasing step and a fixed increasing step to update α and β in each stage. The two step lengths can be tuned though experiments to find the optimal lengths.

4 Experiments

The experiments are implemented using R Studio (IDE) based on Microsoft R Open 3.4.2. The implemented algorithm, which is called a Sequential THREE-Way algorithm to Consensus Clustering based on Co-Association Matrix (S3WCC-CAM), constructs a co-association matrix based on a set of input matrices representing the input clusterings and applies a hierarchical clustering method to generate the final clustering. The main procedure in S3WCC-CAM is given as follows.

Input:

- A set $\mathbb{C}^{in}$ of $n \times n$ matrices where n is the number of data points in the dataset. The values in these matrices are in the unit interval [0,1].
- A number m of input matrices to be used in the first iteration.
- A number $r (r \geq 1)$ used to refine the thresholds.

Output: A hierarchical final clustering $\mathcal{HC}$ of the dataset.

Step 1: Construct the co-association matrix $M_{n \times n}$.

(1) Generate a sequence Seq of thresholds refined by r.

(2) Initialize all the entries in the co-association matrix $M_{n \times n}$ to be N/A (i.e., not available) and the subset $\mathbb{C}_{it}^{in}$ of input matrices used in the next iteration to be empty. As a result, $\mathbb{C}_{it}^{in}$ is the set of visited input matrices in $\mathbb{C}^{in}$ and $(\mathbb{C}^{in} - \mathbb{C}_{it}^{in})$ is the set of non-visited input matrices.

(3) Perform the following steps iteratively until either the boundary region or the set $(\mathbb{C}^{in} - \mathbb{C}^{in}_{it})$ is empty:
   - Get the next pair of thresholds (α, β) from the sequence Seq.
   - If it is the first iteration, select a set of m matrices from $(\mathbb{C}^{in} - \mathbb{C}^{in}_{it})$ and add them to $\mathbb{C}^{in}_{it}$. Otherwise, select one matrix from $(\mathbb{C}^{in} - \mathbb{C}^{in}_{it})$ and add it to $\mathbb{C}^{in}_{it}$.
   - Based on the set $\mathbb{C}^{in}_{it}$, update the evaluation values of all data-point pairs in the current boundary region, divide these pairs into three regions and update the entries in M accordingly.

(4) If the boundary region is not empty, update all entries in the boundary region with 0.5.

Step 2: Generate the hierarchical clustering $\mathcal{HC}$ by applying a hierarchical clustering method to M.

The input matrices in $\mathbb{C}^{in}$ are produced by applying basic clustering algorithms to a dataset. These basic clustering algorithms include 12 algorithms implemented in the package diceR [1], namely, AP, BLOCK, CMEANS, GMM, SC, SOM, DIANA_Euclidean, HC_Euclidean, HDBSCAN, KM_Euclidean, NMF_Scd (or NMF_Lee), and PAM_Euclidean. Every clustering algorithm can be repeatedly applied with different sets of tuning parameters, such as a given number of clusters and a distance measure. In the current implementation, we only consider Euclidean distance and run each algorithm three times with the number of clusters as 3, 4, and 5, respectively. In total, they produce 36 clusterings represented by 36 $n \times n$ matrices that comprise the input set $\mathbb{C}^{in}$.

The sequence Seq of thresholds starts from the most restrictive pair $(1, 0)$. The other pairs are generated according to two step lengths, one δ_α for decreasing α and another δ_β for increasing β. In the current implementation, we consider a simple case where $\delta_\alpha = \delta_\beta = \delta$. The step length δ is calculated as:

$$\delta = \frac{1}{2 * (|\mathbb{C}^{in}| - m + 1) - 1} \cdot \frac{1}{r}, \tag{6}$$

where the number $|\mathbb{C}^{in}| - m + 1$ is the maximum number of iterations.

Each iteration in (3) of Step 1 represents a stage in the presented sequential three-way approach. In order to use as various input clusterings as possible, when selecting matrices from $(\mathbb{C}^{in} - \mathbb{C}^{in}_{it})$, we prefer the matrices produced by non-visited clustering algorithms, that is, these algorithms do not produce any matrix in $\mathbb{C}^{in}_{it}$ that is the set of visited matrices. If there are more candidate matrices than required, we randomly select a required number of matrices from them. To deal with a nonempty boundary region after the iterations, we update all the entries in the boundary region with a value 0.5. The hierarchical clustering method used in Step 2 adopts an agglomerative strategy using the average linkage (UPGMA) [18] to find and merge similar clusters, which is implemented in the package diceR [1].

The algorithm S3WCC-CAM is applied to two datasets, that is, iris[1] from UCI and hgsc[2] from the diceR package. The dataset iris includes 150 data points described by 4 attributes. A fifth attribute of class labels is ignored in the clustering process and used as an external reference in the evaluations. The dataset hgsc includes 489 data points described by 321 attributes without an attribute of class labels. Due to the limitation of our experimental environments, the algorithm is not applied with large datasets in the current experiments. This might be a direction of our future work. The evaluation value of a data-point pair is computed as the proportion of times that the two data points are clustered together out of the times that they are chosen in the bootstrap resampling [1], which is implemented in the package diceR. Table 2 lists the configurations of m and r considered in our experiments.

Table 2. Configurations of m and r in the experiments

id	m	r	id	m	r	id	m	r	id	m	r
c1	3	1	c5	6	1	c9	9	1	c13	12	1
c2	3	3	c6	6	3	c10	9	3	c14	12	3
c3	3	6	c7	6	6	c11	9	6	c15	12	6
c4	3	9	c8	6	9	c12	9	9	c16	12	9

The results of S3WCC-CAM are compared with Cluster-based Similarity Partitioning Algorithm (CSPA) [19] and Link-based Cluster Ensemble method (LCE) [11]. The clustering results are measured by both internal and external indices implemented in the package diceR [1]. The internal indices include avg_within that measures the average distance within clusters, avg_between that measures the average distance between clusters and avg_silwidth that measures the average distance between clusters based on Silhouette width. Thus, a smaller avg_within, a bigger avg_between and a bigger avg_silwidth indicate a better clustering. The external indices measure the similarity of two clusterings by using the class labels as an external reference. The two external indices used in our experiments are the corrected Rand index (corrected_rand) [10] and Meila's variation index (vi) [17]. The corrected Rand index ranges from -1 to 1 with -1 indicating no agreement and 1 indicating perfect agreement. The Meila's variation index measures the variation of information for two clusterings based on mutual information. It has an upper bound $\log n$ where n is the number of data points in the dataset. A smaller Meila's variation index indicates a better clustering. Table 3 summarizes the results of all the above indices. Besides, Table 3 also shows the run time (run_time) and the percentage of boundary region when the iterations stop (BND_perc) in S3WCC-CAM. Since the dataset hgsc does not contain the class labels, only internal indices are evaluated.

[1] https://archive.ics.uci.edu/ml/datasets/Iris.
[2] https://www.rdocumentation.org/packages/diceR/versions/0.3.2/topics/hgsc.

Table 3. A summary of the experiment results

		internal indices			external indices		run_time(s)	BND_perc(%)
		avg_within	avg_between	avg_silwidth	corrected_rand	vi		
S3WCC-CAM	c1	0.132±0.096	0.991±0.007	0.877±0.089	0.739±0.078	0.468±0.075	0.227±0.017	1.701±0.714
	c2	0.093±0.043	0.993±0.003	0.916±0.036	0.747±0.020	0.461±0.030	0.720±0.024	0.457±0.180
	c3	0.161±0.042	0.969±0.017	0.830±0.043	0.750±0.024	0.453±0.033	0.835±0.026	13.610±2.325
	c4	0.210±0.044	0.943±0.021	0.750±0.050	0.751±0.025	0.446±0.043	0.837±0.028	20.914±3.404
	c5	0.126±0.074	0.990±0.008	0.882±0.067	0.746±0.013	0.467±0.028	0.158±0.014	3.014±1.113
	c6	0.102±0.039	0.993±0.003	0.908±0.034	0.747±0.007	0.470±0.025	0.497±0.024	1.225±0.353
	c7	0.130±0.003	0.989±0.000	0.879±0.003	0.745±0.001	0.483±0.002	0.780±0.028	8.480±0.172
	c8	0.186±0.015	0.959±0.004	0.793±0.016	0.749±0.007	0.464±0.029	0.783±0.028	18.013±1.180
	c9	0.126±0.062	0.987±0.011	0.881±0.056	0.748±0.008	0.466±0.027	0.098±0.010	5.817±1.667
	c10	0.112±0.058	0.991±0.005	0.896±0.053	0.746±0.009	0.469±0.026	0.275±0.015	1.566±0.501
	c11	0.098±0.031	0.993±0.002	0.912±0.028	0.747±0.007	0.471±0.026	0.546±0.023	1.643±0.357
	c12	0.106±0.000	0.992±0.000	0.903±0.000	0.745±0.000	0.483±0.000	0.727±0.030	5.241±0.015
	c13	0.482±0.028	0.872±0.058	0.194±0.045	0.477±0.053	0.790±0.128	0.031±0.000	61.272±5.289
	c14	0.162±0.051	0.972±0.018	0.834±0.049	0.748±0.008	0.463±0.028	0.053±0.001	12.911±2.732
	c15	0.150±0.046	0.976±0.017	0.849±0.044	0.749±0.007	0.461±0.029	0.075±0.002	11.956±2.699
	c16	0.116±0.055	0.990±0.007	0.892±0.049	0.747±0.007	0.470±0.026	0.124±0.005	4.330±1.145
CSPA		0.687±0	8.464±0	0.898±0	0.745±0	0.483±0	0.615±0.020	N/A
LCE		0.339±0	6.133±0	0.906±0	0.759±0	0.422±0	1.438±0.034	N/A

(a) iris

		internal indices			run_time(s)	BND_perc(%)
		avg_within	avg_between	avg_silwidth		
S3WCC-CAM	c1	0.272±0.091	0.932±0.025	0.695±0.085	0.349±0.021	3.323±0.654
	c2	0.254±0.042	0.932±0.017	0.714±0.038	1.069±0.043	0.899±0.120
	c3	0.293±0.062	0.910±0.022	0.642±0.060	1.291±0.064	26.236±4.778
	c4	0.334±0.067	0.888±0.036	0.570±0.081	1.308±0.069	36.903±6.814
	c5	0.252±0.080	0.929±0.021	0.710±0.071	0.268±0.020	6.470±0.785
	c6	0.251±0.058	0.934±0.015	0.723±0.042	0.770±0.028	2.514±0.191
	c7	0.307±0.015	0.922±0.003	0.655±0.012	1.239±0.043	17.703±0.408
	c8	0.324±0.031	0.885±0.017	0.569±0.027	1.274±0.046	36.643±1.549
	c9	0.275±0.075	0.933±0.022	0.689±0.063	0.182±0.017	11.348±1.292
	c10	0.258±0.058	0.932±0.019	0.703±0.056	0.451±0.029	3.335±0.290
	c11	0.259±0.047	0.935±0.015	0.709±0.044	0.854±0.039	3.298±0.194
	c12	0.275±0.004	0.918±0.002	0.678±0.004	1.151±0.047	11.190±0.033
	c13	0.484±0.009	0.685±0.046	0.110±0.048	0.084±0.001	91.439±8.501
	c14	0.281±0.054	0.909±0.024	0.649±0.048	0.125±0.022	26.261±2.971
	c15	0.286±0.064	0.910±0.028	0.650±0.050	0.163±0.022	25.775±2.708
	c16	0.259±0.065	0.933±0.019	0.703±0.054	0.238±0.027	8.938±0.836
CSPA		2.485±0	11.752±0	0.710±0	20.341±0.099	N/A
LCE		2.401±0	10.508±0	0.677±0	11.261±0.070	N/A

(b) hgsc

As shown in Table 3, S3WCC-CAM generally produces as good clustering results as CSPA and LCE based on the internal and external indices. In terms of the run time, S3WCC-CAM outperforms LCE with all the configurations and CSPA with most configurations, especially on the dataset hgsc. Different configurations of m and r in S3WCC-CAM have a significant influence on run_time and BND_perc. A further study, either experimental or theoretical, on the optimal configuration is necessary and might be a direction for future work.

5 Conclusions and Future Work

We present a sequential three-way approach to progressively constructing a co-association matrix in multiple stages. In each stage, we calculate the evaluation values based on a set of input clusterings. A pair of thresholds is then used to cut the evaluation values, and accordingly, the data-point pairs are divided into three disjoint positive, negative and boundary regions. The entries in the co-association matrix corresponding to the positive and negative regions are updated with the highest evaluation value 1 and the lowest evaluation value 0, respectively. Accordingly, a definite decision of either clustering two data points together or separating them is associated. By gradually involving more input clusterings, we are able to refine the evaluation values in the boundary regions and make a definite decision if possible. By determining some entries to be 1 or 0 once sufficient information can be obtained from the input clusterings, the presented approach makes quick definite decisions on the clustering of some data points in early stages. In this way, we may reduce the overall computational cost of constructing the co-association matrix and obtaining the final clustering.

One direction of the future work is to solve the two issues in the presented approach as mentioned. A second direction is to generalize the presented sequential approach with respect to other consensus clustering methods that do not use co-association matrix. A third direction is a further experimental study, including the optimal configuration of S3WCC-CAM as well as its applications on larger datasets.

References

1. Chiu, D.S., Talhouk, A.: diceR: an R package for class discovery using an ensemble driven approach. BMC Bioinform. **19**, 11–18 (2018)
2. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm (with discussion). J. Royal Stat. Soc. Ser. B **39**, 1–38 (1977)
3. Deng, X.F., Yao, Y.Y.: An information-theoretic interpretation of thresholds in probabilistic rough sets. In: Li, T., et al. (eds.) RSKT 2012. LNCS (LNAI), vol. 7414, pp. 369–378. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31900-6_46
4. Donath, W.E., Hoffman, A.J.: Algorithms for partitioning of graphs and computer logic based on eigenvectors of connection matrices. IBM Tech. Discl. Bull. **15**, 938–944 (1972)
5. Ester, M., Kriegel, H.P., Sander, J., Xu, X.W.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Simoudis, E., et al. (eds.) KDD 1996, pp. 226–231. AAAI Press (1996)
6. Fred, A.: Finding consistent clusters in data partitions. In: Kittler, J., Roli, F. (eds.) MCS 2001. LNCS, vol. 2096, pp. 309–318. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-48219-9_31
7. Fred, A., Jain, A.K.: Combining multiple clustering using evidence accumulation. IEEE Trans. Pattern Anal. Mach. Intell. **27**, 835–850 (2005)
8. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd edn. Springer, New York (2009). https://doi.org/10.1007/978-0-387-84858-7

9. Herbert, J.P., Yao, J.T.: Game-theoretic rough sets. Fundamenta Informaticae **108**, 267–286 (2011)
10. Hubert, L., Arabie, P.: Comparing partitions. J. Classif. **2**, 193–218 (1985)
11. Iam-on, N., Boongoen, T., Garrett, S.: LCE: a link-based cluster ensemble method for improved gene expression data analysis. Bioinformatics **26**, 1513–1519 (2010)
12. Iam-on, N., Boongoen, T., Garrett, S.: Refining pairwise similarity matrix for cluster ensemble problem with cluster relations. In: Jean-Fran, J.-F., Berthold, M.R., Horváth, T. (eds.) DS 2008. LNCS, vol. 5255, pp. 222–233. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88411-8_22
13. Li, Y., Yu, J., Hao, P., Li, Z.: Clustering ensembles based on normalized edges. In: Zhou, Z.-H., Li, H., Yang, Q. (eds.) PAKDD 2007. LNCS, vol. 4426, pp. 664–671. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-71701-0_71
14. Li, H.X., Zhang, L.B., Huang, B., Zhou, X.Z.: Sequential three-way decision and granulation for cost-sensitive face recognition. Knowl. Based Syst. **91**, 241–251 (2016)
15. Li, H.X., Zhang, L.B., Zhou, X.Z., Huang, B.: Cost-sensitive sequential three-way decision modeling using a deep neural network. Int. J. Approx. Reason. **85**, 68–78 (2017)
16. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297. University of California Press (1967)
17. Meila, M.: Comparing clusterings - an information based distance. J. Multivar. Anal. **98**, 873–895 (2007)
18. Sokal, R., Michener, C.: A statistical method for evaluating systematic relationships. Univ. Kansas Sci. Bull. **38**, 1409–1438 (1958)
19. Strehl, A., Ghosh, J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res. **3**, 583–617 (2002)
20. Vega-Pons, S., Ruiz-Shulcloper, J.: A survey of clustering ensemble algorithms. Int. J. Pattern Recogn. Artif. Intell. **25**, 337–372 (2011)
21. Vega-Pons, S., Ruiz-Shulcloper, J.: Clustering ensemble method for heterogeneous partitions. In: Bayro-Corrochano, E., Eklundh, J.-O. (eds.) CIARP 2009. LNCS, vol. 5856, pp. 481–488. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10268-4_56
22. Wang, X., Yang, C., Zhou, J.: Clustering aggregation by probability accumulation. Pattern Recogn. **42**, 668–675 (2009)
23. Yao, Y.Y.: An outline of a theory of three-way decisions. In: Yao, J.T., et al. (eds.) RSCTC 2012. LNCS, vol. 7413, pp. 1–17. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32115-3_1
24. Yao, Y.Y.: Probabilistic rough set approximations. Int. J. Approx. Reason. **49**, 255–271 (2008)
25. Yao, Y.Y., Deng, X.F.: Sequential three-way decisions with probabilistic rough sets. In: Wang, Y., et al. (eds.) ICCI-CC 2011, pp. 120–125 (2011)
26. Yao, Y.Y., Hu, M., Deng, X.F.: Modes of sequential three-way classifications. In: Medina, J., Ojeda-Aciego, M., Verdegay, J.L., Pelta, D.A., Cabrera, I.P., Bouchon-Meunier, B., Yager, R.R. (eds.) IPMU 2018. CCIS, vol. 854, pp. 724–735. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91476-3_59
27. Yao, Y.Y., Lingras, P., Wang, R., Miao, D.: Interval set cluster analysis: a reformulation. In: Sakai, H., Chakraborty, M.K., Hassanien, A.E., Ślęzak, D., Zhu, W. (eds.) RSFDGrC 2009. LNCS, vol. 5908, pp. 398–405. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10646-0_48

28. Yu, H.: A framework of three-way cluster analysis. In: Polkowski, L., et al. (eds.) IJCRS 2017. LNCS, vol. 10314, pp. 300–312. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-60840-2_22
29. Yu, H., Wang, X., Wang, G.: A semi-supervised three-way clustering framework for multi-view data. In: Polkowski, L., et al. (eds.) IJCRS 2017. LNCS, vol. 10314, pp. 313–325. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-60840-2_23
30. Yu, H., Zhang, H.: A three-way decision clustering approach for high dimensional data. In: Flores, V., et al. (eds.) IJCRS 2016. LNCS, vol. 9920, pp. 229–239. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47160-0_21