Pattern Recognition 45 (2012) 3017–3033

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

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

A robust adaptive clustering analysis method for automatic identification of clusters

P.Y. Mok*, H.Q. Huang, Y.L. Kwok, J.S. Au

Institute of Textiles and Clothing, The Hong Kong Polytechnic University, Hunghom, Hong Kong

ARTICLE INFO

Article history:
Received 4 January 2010
Received in revised form
11 December 2011
Accepted 2 February 2012
Available online 16 February 2012

Keywords:
Cluster analysis
Cluster validity
Fuzzy clustering
Fuzzy C-Means
Cluster ensembles

ABSTRACT

Identifying the optimal cluster number and generating reliable clustering results are necessary but challenging tasks in cluster analysis. The effectiveness of clustering analysis relies not only on the assumption of cluster number but also on the clustering algorithm employed. This paper proposes a new clustering analysis method that identifies the desired cluster number and produces, at the same time, reliable clustering solutions. It first obtains many clustering results from a specific algorithm, such as Fuzzy C-Means (FCM), and then integrates these different results as a judgement matrix. An iterative graph-partitioning process is implemented to identify the desired cluster number and the final result. The proposed method is a robust approach as it is demonstrated its effectiveness in clustering 2D data sets and multi-dimensional real-world data sets of different shapes. The method is compared with cluster validity analysis and other methods such as spectral clustering and cluster ensemble methods. The method is also shown efficient in mesh segmentation applications. The proposed method is also adaptive because it not only works with the FCM algorithm but also other clustering methods like the *k*-means algorithm.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

Cluster analysis aims to partition a large number of data into different subsets or groups so that the requirements of homogeneity and heterogeneity are fulfilled. Homogeneity requires that data in the same cluster should be as similar as possible and heterogeneity means that data in different clusters should be as different as possible [1]. Typical clustering activity involves three sequential steps [2]: data/object representation, definition and computation of data proximity, and clustering/grouping, as shown in Fig. 1. Data/object representation refers to problem definition including the number of clusters, the number of available data, and the number, type, and scale of the data variables available to the clustering algorithm. Data proximity, also known as inter-object similarity, is usually measured by a distance function defined on pairs of data. A variety of distance measures are in use for different purposes. The grouping step can be performed in a number of ways, for instance hierarchical approach, partitional approach and other algorithms can be employed. Cluster analysis is widely used in areas such as market research, pattern recognition [3], image segmentation [4], and mesh segmentation [5].

Different clustering algorithms have been developed in the past, and some examples are shifting grid [6], SOFM neural networks [7] and Evidential C-Means [8]. The availability of such a vast collection of clustering algorithms in the literature can easily confuse users attempting to select algorithm for a specific problem. When presented with data, all clustering algorithms will produce clusters regardless of whether the data contain clusters or not. There is no clustering technique that is universally applicable in uncovering the variety of structures present in multidimensional data sets [2]. It is because clustering algorithms often contain implicit assumptions about cluster shape and grouping criteria used. Clustering algorithms must be carefully selected by evaluating (1) the manner in which clusters are formed, (2) the structure of the data, and (3) sensitivity of the clustering technique [2].

Although clustering is a useful and challenging problem with great potential in applications, its application must be cautiously handled. Otherwise, the technique can easily be abused or misapplied. Cluster number and similarity measure are the two most important assumptions of clustering analysis, which affect the overall quality of the results. In most of the automatic clustering algorithms, the cluster number must be first defined. This is true for most popular algorithms like the Fuzzy C-Means (FCM) clustering algorithm. Some researchers [9–13] have proposed cluster validity indices to validate the cluster results so as to obtain the optimal cluster number. Apart from identifying the

* Corresponding author. Tel.: +852 2766 4442; fax: +852 2773 1432.
E-mail address: tracy.mok@inet.polyu.edu.hk (P.Y. Mok).

0031-3203/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2012.02.003

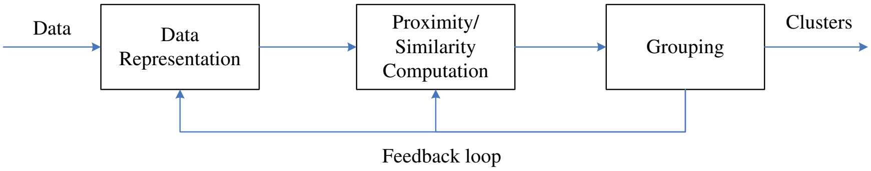


Fig. 1. Steps in clustering.

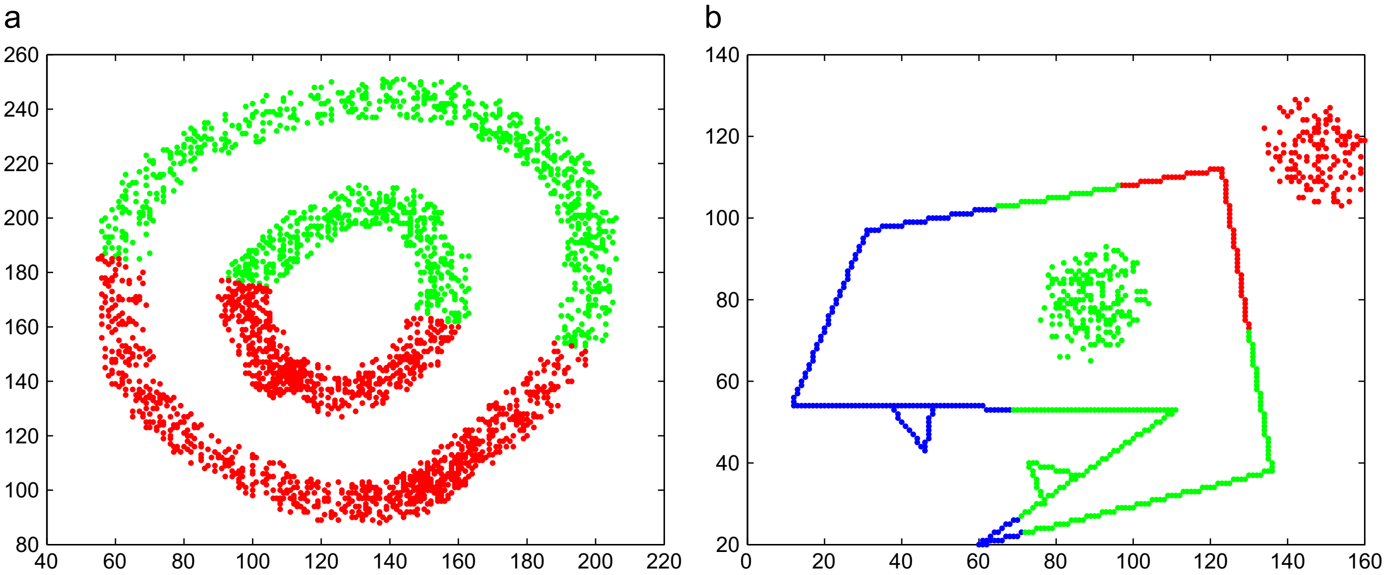


Fig. 2. Clustering data sets by Fuzzy C-Means clustering algorithm. (a) cluster number=2 and (b) cluster number=3.

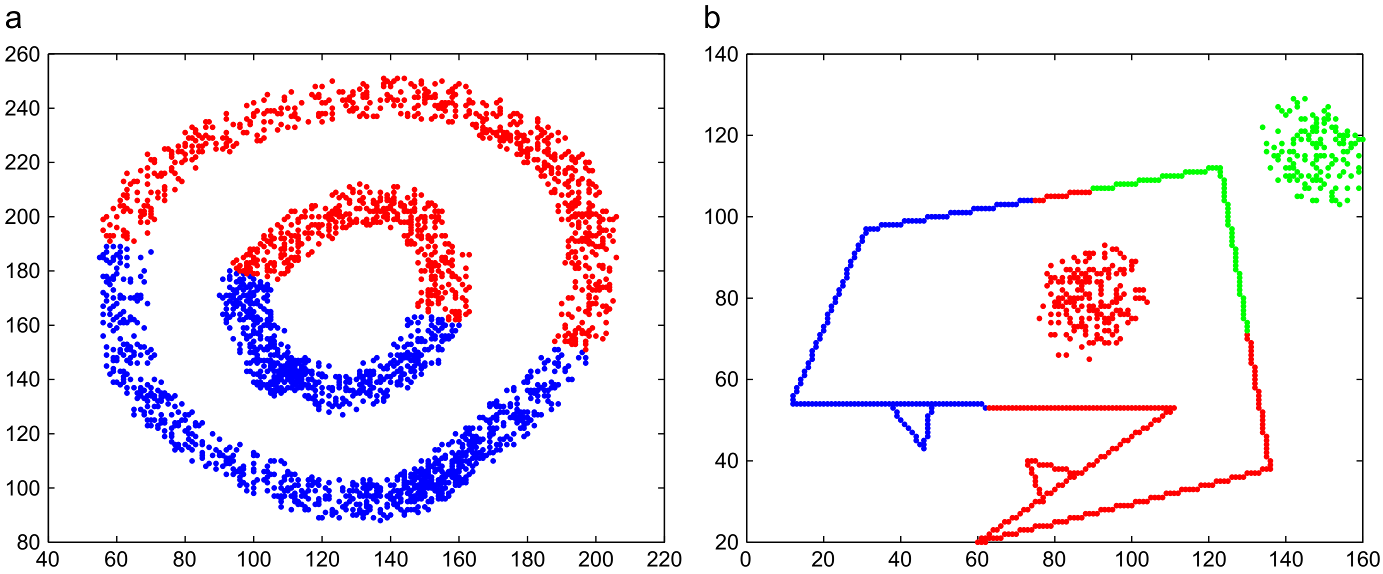


Fig. 3. Clustering data sets by *k*-means clustering algorithm. (a) cluster number=2 and (b) cluster number=3.

optimal cluster number, effective clustering requires the algorithm to be robust for data sets of different shapes. Sometimes, correct cluster numbers do not guarantee that a data set can be properly partitioned in the desired way. Most widely used clustering algorithms assumed distance based similarity measures [2], upon which the grouping process is carried out. There are varied types of distance based similarity measures, such as Euclidean distance, Manhanttan distance, and Mahalanobis distance. The similarity measure must be chosen carefully. For instance, as shown in Figs. 2 and 3, data sets are not well partitioned by either FCM or *k*-means algorithms, even though the correct cluster numbers are given. FCM and *k*-means algorithms use centroid-based distance as similarity measurement. In Figs. 2 and 3, different clusters are depicted in different colours, and all figures hereafter follow the same colour scheme to illustrate cluster results.

The objective of this paper is to propose a robust and adaptive clustering analysis method that produces reliable clustering results and simultaneously identifies the desired cluster number. This method integrates the results of a specific clustering algorithm with a range of cluster number initialisations, and then identifies the desired cluster number and the final result by a graph-partitioning process. The advantages of the proposed method includes: (1) identifying the desired cluster number and reliable result in an unsupervised manner; (2) requiring no additional validation step of the clustering result; and (3) adaptive nature of the method allowing effective integration with different clustering algorithms without modification.

The remaining part of this paper is organised as follows: Section 2 reviews different approaches of cluster analysis, the related work on cluster validity indices, and graph theoretic applications. Section 3 describes the proposed clustering method and its implementation. Section 4 shows some experimental results using both artificial and real-world data sets. An application to mesh segmentation based on the proposed clustering method is also included. Section 5 discusses that the current method is adaptable to implement with different clustering algorithms, and *k*-means algorithm is used as an example for illustration. The method is also compared with spectral clustering and cluster ensemble methods.

2. Related work

2.1. Clustering techniques

The clustering problem is to partition a data set into groups (clusters) so that the data within a cluster are more similar to each other than data in different clusters. Clustering algorithms can be classified as hierarchical, partitional, density-based and grid-based clustering [2,14]. Among which, hierarchical and partitional clustering are the two most widely used approaches [14]. *Hierarchical clustering* algorithms create clusters recursively. They merge smaller cluster into larger ones or split larger clusters into smaller ones. Hierarchical algorithms construct a sequence of nested clusters in a dendrogram, a graph of tree-like structure, representing the groups of patterns and similarity level at which groupings change. In contrast, partitional clustering does not involve treelike construction process. Instead, *Partitional clustering* algorithms decompose directly the data set into a set of disjoint clusters. They attempt to determine the desired partitions that optimise a certain criterion function (similarity measure).

Both hierarchical and partitional schemes have advantages and limitations. Hierarchical methods allow user to examine a wide range of solutions, obtained from a defined measure of similarity, in an efficient manner. However, hierarchical methods have a few shortcomings. Firstly, the sequential process of hierarchical clustering may prevent arriving at optimal clusters due to undesirable early combinations/divisions of clusters. Secondly, hierarchical methods are susceptible to outliers in the data. Partitional schemes can better manage large data sets comparing to hierarchical approach, because the construction of a dendrogram in hierarchical clustering for large data set is computationally prohibitive. Different algorithms have been proposed for partitional clustering, and some are optimisation problems. These algorithms can produce the optimal result with respect to a defined criterion, namely the input parameter of cluster number and the defined similarity measure. However, the result of a clustering algorithm can be very different on the same data set for different inputs because input parameters can extremely modify the behaviour and execution of the algorithm [14]. Jain et al. [2] pointed out that a key problem accompanying the use of partitional algorithms is the choice of cluster number, according to some practical, objective or theoretical basis.

Some researchers recommend a hybrid approach integrating both hierarchical and partitional clustering. A hierarchical technique is first used to select the number of clusters and the profile cluster centres that serve as initial cluster seeds in the partitional approach. Next, partitional method is used to cluster all data using the seed points [15]. However, even though the hybrid approach helps the identification of cluster number, the final results are optimised based on the defined similarity measure. As shown in Figs. 2 and 3, distance based measures may not fit data with special shape.

2.2. Cluster validity indices

A critical issue of performing clustering analysis, by either hierarchical or partitional method, is determining the number of clusters most representative of the data structure. Unfortunately, no standard objective selection procedure of cluster number exists [15]. The study on identifying the optimal cluster number can sometimes be referred as cluster validity study. Cluster validity issues are concerned with determining the correct number of clusters and checking the quality of clustering results. Many difference indices of cluster validity have been proposed, such as the Bezedek's partition coefficient (PC) and partition entropy (PE) [16,17], the Xie-Beni's index [18], and the Fukuyama and Sugeno index [19]. See [20,21] for comprehensive reviews of cluster validity. In the case of fuzzy clustering, some validity indices use only the information of fuzzy membership to evaluate the clustering results [16,17], and other indices make use of not only the fuzzy memberships but also the structure of the data [18,19]. Zhang [11] proposed a validity index for obtaining the optimal cluster number by evaluating both the compactness and separation of the cluster result. Compactness indicates the difference or variation of data points within the same cluster, and separation represents the strength of separation between clusters. A good clustering result is expected to have a low degree of compactness and a large distance of separation. However, cluster validity indices have certain drawbacks that such indices are defined only for fuzzy partitioning or a specific group of clustering algorithm. In other words, the cluster validity indices must be modified if other types of cluster algorithm are used. In this paper, the clustering results obtained from the proposed method are compared with the cluster validity analysis results in Section 4.2.

2.3. Application of graph theory in clustering analysis

Graph theory has long been applied in cluster analysis. The best-known graphic–theoretic divisive clustering algorithm is based on construction of the minimal spanning tree of the data, then deleting the tree edges with the largest lengths to generate clusters [2]. The hierarchical clustering methods are applications of graph–theoretic clustering. Traditional, graphs are used to indicate the similarity (or dissimilarity) between data in the clustering results. In this paper, adjacency graphs are used to represent the degree of association between data, i.e., the connectivity relationship, rather than their similarity.

This paper aims to develop a new approach of clustering for simultaneously obtaining the desired cluster number and effective clustering result. The proposed method is described in detailed in the next section and the method is implemented with the FCM algorithm. Nevertheless, it is important to know that the proposed method can be easily applied to other clustering algorithms. An example of integrating the method with the *k*-means clustering algorithm is described in Section 5.1.

2.4. Spectral clustering and clustering ensembles

Spectral clustering method is to partition data by clustering the *k*-largest eigenvectors of the Laplacian matrix derived from a given data set [22]. Different algorithms were proposed [23–25]. The main differences between these spectral algorithms are (1) the construction of the similarity graph *W*, (2) the computation of the Laplacian matrix *L*, and (3) the operation of the *k*-largest vectors [26]. The clustering method proposed in this paper is compared with the spectral clustering algorithm [24].

Cluster ensemble is a technique to combine several runs of different clustering algorithms or independent runs of the same clustering algorithm in order to get a better cluster partition of the original data set. Although research on cluster ensemble has not been widely recognised [27], several research efforts have been done independently [28–30]. In [29], the authors formally defined the cluster ensemble problem as an optimisation problem. Generally speaking, cluster ensembles have two approaches, co-association methods [28] and hyper-graph methods [29]. In co-association methods, object relationship is represented in a co-association matrix, and then consensus partition is obtained by applying a clustering algorithm. In hyper-graph methods, different data partitions are represented as hyper-graphs, and the final cluster result is obtained by hyper-graph-partitioning algorithm. Strehl and Ghosh proposed three efficient heuristics to partition the hyper-graphs in [30], including Cluster-based Similarity Partitioning Algorithm (CSPA), Hyper Graph Partitioning Algorithm (HPGA), and Meta-Clustering Algorithm (MCLA). Fred and Jain [28] adopted co-association approach and developed an evidence accumulation clustering (EAC) method based on single-link (SL) and average-link (AL) hierarchical clustering. Two other cluster ensemble methods, called weighted cluster ensemble using a kernel consensus function (denoted as WKF) [30] and Generalised WKF (denoted as GWKF) [31], were also proposed following the co-association approach.

In this paper, the proposed clustering method is compared with spectral clustering and cluster ensembles in Section 5.

3. A robust adaptive clustering method

3.1. Method overview and matrix definitions

The proposed clustering method follows an unsupervised approach to obtain the desired cluster number and final partitioned result. Inspiration comes from the human decision making process. People usually make decisions after integrating diverse opinions and comparing different alternatives. Similar to hierarchical clustering, a number of clustering results are first analysed, from which final results are obtained by manipulating the graph of the resulted matrix. Different from hierarchical clustering, the clustering process is not a sequential process, where the cluster solutions later in the process are not based on the clusters formed early. Instead, the cluster solutions being integrate to obtain the final results are indeed cluster results obtained from a clustering algorithm with varied initial settings. In other words, the proposed method integrates both hierarchical and partitional approaches in order to obtain reliable clustering results. The whole process contains three phases:

Phase 1: A data set is clustered by a clustering algorithm a number of times with a range of cluster number initialisations.
Phase 2: The clustering results are combined and investigated by an iterative graph partitioning process.
Phase 3: The desired cluster number and clustering results are identified by evaluating the distribution of graph partitioning results.

The proposed method can lead to reliable clustering result by summarising and evaluating a number of clustering solutions in the form of a matrix and adjacency graph. The definitions of these matrices and graphs are given as follows:

Definition 1. *Observation matrix* O_C is an $N \times N$ matrix obtained by clustering a data set $X=\{x_1, \ldots, x_N\} \subset R^m$ into C clusters with a given clustering algorithm. Each entry of the observation matrix, o_{ij}, represents whether or not the two data points x_i and x_j belong to the same cluster.

Definition 2. *Judgment matrix* J is the sum of all observation matrices O_C for different values of C, $C \in [2,k]$.

Definition 3. *Judgment graph* G_J is defined as the adjacency graph of judgment matrix J.

3.2. Clustering with varied initializations and result integration

If $X=\{x_1, \ldots, x_N\}$ is a given data set of N samples, each sample, $x_i \in R^m$ is an m-dimensional data point. The proposed clustering approach can be integrated with any clustering algorithms, but it is first illustrated using Fuzzy C-Means (FCM) algorithm in this paper. FCM proposed by Bezdek [3] is one of the most widely used clustering algorithms with applications in social surveys, engineering, chemistry and so forth. Until recently, further development and new applications of the algorithm are still being actively studied [13].

FCM algorithm partitions X into C clusters by minimising an objective function

$$F=\sum_{i=1}^{N}\sum_{j=1}^{C}u_{ij}^{p}\|x_i-c_j\|^2, \quad p>1 \tag{1}$$

The cluster number C is first defined by users in the initialisation process. The centres (centroids) of cluster, $c_{j,}$ with the following definition,

$$c_j=\frac{\sum_{i=1}^{N}u_{ij}^{p}\cdot x_i}{\sum_{i=1}^{N}u_{ij}^{p}}, \quad 1 \le j \le C \tag{2}$$

can be obtained by optimising the objective function (Eq. (1)) iteratively, where u_{ij}^{p} is the Fuzzy C-Means (FCM), which defines the membership degree of data point x_i in cluster j as follows

$$u_{ij}=\left[\sum_{k=1}^{C}\left(\frac{\|x_i-c_j\|}{\|x_i-c_k\|}\right)^{2/p-1}\right]^{-1}, \quad 1 \le i \le N, 1 \le j \le C \tag{3}$$

In this way, the FCM provides a degree of membership for each data point whereby a data point can belong to more than one cluster during the iteration. The parameter p in the above equations controls the fuzziness of membership of every data point. In FCM, cluster number C determines the overall quality of the clustering results, but the value of C is unknown.

By clustering the data set X using the FCM algorithm, the following $C \times N$ membership matrix U is obtained:

$$U=\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1N} \\ u_{21} & u_{22} & \cdots & u_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ u_{C1} & u_{C2} & \cdots & u_{CN} \end{bmatrix} \tag{4}$$

The sum of each column of membership values equals 1, i.e. $\sum_{i=1}^{C} u_{ij}=1$.

A row vector $L=[l_1\ l_2\ \ldots\ l_N]$ is then defined so that l_j is the row number of the largest membership value u_{ij} in each column of U, i.e. $l \in L = \left\{ i \,\middle|\, \max_{i=1}^{C} u_{ij} \right\}$. Each l_j represents a label indicating to which cluster data point x_j belongs. The observation matrix $O=[o_{ij}]_{N\times N}$ is computed based on the vector L as follows

$$o_{ij} = \begin{cases} 1 & l_i = l_j (i \neq j) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $o_{ij}=1$ if x_i and x_j have the same label l in vector L, otherwise $o_{ij}=0$. In other words, observation matrix O shows the relationships of data points, $o_{ij}=1$ means that x_i and x_j belong to the same cluster, while $o_{ij}=0$ implies the two data points are in different clusters. Therefore, O is a diagonal symmetric matrix representing an observation result of clustering X in C clusters. Besides, O can also be viewed as an adjacency matrix of an undirected graph, whose adjacent edge lengths are all 1.

With the above matrix operation, the clustering result of a given cluster number is represented by an observation matrix. In order to obtain reliable clustering result, different observation matrices are computed with the cluster number C ranging from 2 to k, where k is a number smaller than N. Different k values will influence the reliability of the cluster number and this will be discussed later in Section 4.1. The lower bound of C is 2 because a data set is usually clustered into more than one group. To integrate different clustering results, a judgment matrix J is defined as

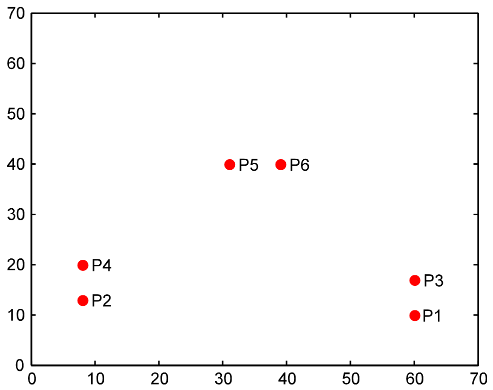
$$J = \sum_{C=2}^{k} O_C \tag{6}$$

where O_C is the observation matrix for clustering the data set into C clusters. The judgement matrix J can be viewed as an weighted adjacency graph representing the algebraic connectivity of the given data points.

The judgement matrix contains all the connectivity relationship between data points. Larger the entry value of matrix J, stronger the relationship between the corresponding data points, higher chance they are being grouped in the same cluster in clustering process. As mentioned, matrix J represents the weighed adjacency graph, G_J. Clustering can therefore be performed by a simple graph partitioning process [32] on G_J. Fig. 4 indicates that data with or without noise can be clustered based on a defined cluster number. It is important to note that adjacency graphs are used to represent the possibility that data/objects being grouped in the same cluster in the proposed method, which is different from the dissimilarity (the 'distance') between data in traditional graph–theoretic clustering.

A numerical example is given below to illustrate the process of integrating different cluster results as observation matrices, O, and then of evaluating the overall data point relationship as a judgment matrix J. Let data set X be a sample of six 2D data points:

$$X = \begin{bmatrix} 15 & 58 & 15 & 58 & 30 & 36 \\ 18 & 20 & 23 & 26 & 39 & 39 \end{bmatrix}$$



The data set is clustered into 2 to 4 clusters using the FCM clustering algorithm, which generates membership matrices U and the corresponding row vectors L as follows

$$U_{2\times 6} = \begin{bmatrix} 0.9283 & 0.0151 & 0.9640 & 0.0050 & 0.8161 & 0.6468 \\ 0.0717 & 0.9849 & 0.0360 & 0.9950 & 0.1839 & 0.3532 \end{bmatrix} \quad \text{for } C=2$$

$$U_{3\times 6} = \begin{bmatrix} 0.0033 & 0.9862 & 0.0033 & 0.9841 & 0.0085 & 0.0119 \\ 0.0081 & 0.0090 & 0.0107 & 0.0112 & 0.9760 & 0.9770 \\ 0.9887 & 0.0048 & 0.9860 & 0.0047 & 0.0155 & 0.0112 \end{bmatrix} \quad \text{for } C=3$$

$$U_{4\times 6} = \begin{bmatrix} 0.9854 & 0.0000 & 0.9827 & 0.0000 & 0.0152 & 0.0112 \\ 0.0033 & 1.0000 & 0.0033 & 0.0000 & 0.0076 & 0.0104 \\ 0.0080 & 0.0000 & 0.0106 & 0.0000 & 0.9681 & 0.9651 \\ 0.0032 & 0.0000 & 0.0033 & 1.0000 & 0.0091 & 0.0134 \end{bmatrix} \quad \text{for } C=4$$

$$L_{C=2} = \begin{bmatrix} 1 & 2 & 1 & 2 & 1 & 1 \end{bmatrix} \quad L_{C=3} = \begin{bmatrix} 3 & 1 & 3 & 1 & 2 & 2 \end{bmatrix}$$

$$L_{C=4} = \begin{bmatrix} 1 & 2 & 1 & 4 & 3 & 3 \end{bmatrix}$$

From the vectors L, observation matrices are computed showing the relationships between these six data points:

$$O_{C=2} = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}, O_{C=3} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

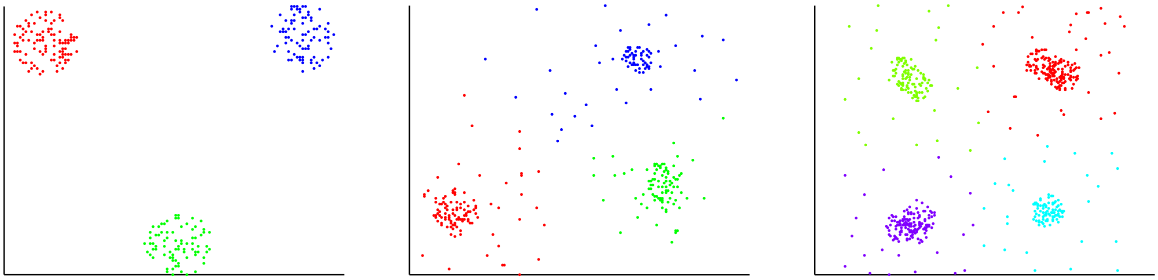


Fig. 4. Partitioning on the judgment matrix J by pre-defining a cluster number.

$$J = \sum_{i=2}^{4} O_i = \begin{bmatrix} 0 & 0 & 3 & 0 & 1 & 1 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 3 & 0 & 0 & 0 & 1 & 1 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 3 \\ 1 & 0 & 1 & 0 & 3 & 0 \end{bmatrix}$$

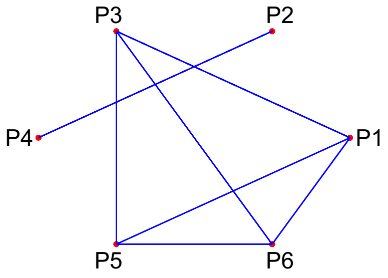


Fig. 5. An example of the judgment matrix J and its adjacency graph. The value of each entry j_{ij} of judgment matrix J indicates the strength of the connection between data points P_i and P_j.

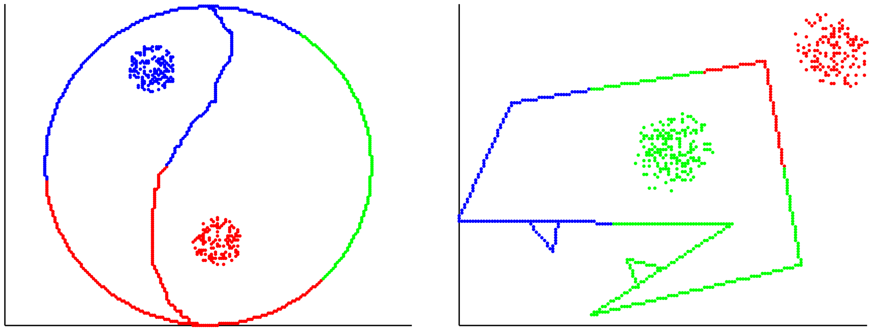


Fig. 6. Simple graph partitioning on the judgment matrix does not always provide good results.

$$O_{C=4} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Finally, a judgment matrix J is computed by Eq. (6). The corresponding adjacency graph G_J of J is shown in Fig. 5. In this figure, the value of an entry in matrix J represents the connection strength of the corresponding data points. For an instance, $J(1,3)$ with the value 3 means the connection between $P1$ and $P3$ is stronger than that of $P1$ and $P5$ because the value $J(1,5)$ is 1.

3.3. Iterative graph partitioning

Even though clustering can be achieved by graph-partitioning the judgment matrix [33,34], a specific cluster number must be first defined. Furthermore, experimental results show that a simple graph-partitioning method [32] on the judgment matrix does not always provide desired results with a fixed cluster number, such as the examples shown in Fig. 6. Therefore, an iterative graph partitioning process is proposed to *decrease* the values of the matrix J by one degree so as to gradually break off the connection among these data points. The partitioning divides the original graph G_J into sub-graphs. By *counting* these adjacency sub-graphs, cluster numbers can be identified. In the current method, the iterative graph partitioning process comprises two steps, namely decreasing matrices and counting sub-graphs.

Decreasing matrices is a procedure to decrease the judgment matrix J by one degree in an iterative way until all entries of the matrix become zero. This procedure can be represented as

$$t_{ij}^{new} = \begin{cases} t_{ij}^{previous}-1 & \text{if } t_{ij}^{previous} > 0 \\ 0 & \text{otherwise} \end{cases}, t_{ij}^{new} \in J_{ij}^{new}, t_{ij}^{previous} \in J_{ij}^{previous} \tag{6}$$

where a new matrix J^{new} is obtained by deducting 1 from every entry of the previous matrix $J^{previous}$. The purpose of the decreasing matrices procedure is to gradually break weak connections between data points.

Table 1
Graph-partitioning algorithm on judgment matrix to identify desired cluster number and final result.

```
Input: A judgment matrix $J$

$n=0$;
$G^{new}[0]=J$. BSF_traversal ();
$J^{previous}=J$;
Do
    $n=n+1$;
    $J^{new}$=Decreasing ($J^{previous}$);
    $G_{subgraphs}[n]$=BSF_traversal($J^{new}$)
    ClusterNumber[$n$]= $G_{subgraphs}[n]$.getSubGraphNumber();
    $J^{previous}=J^{new}$;
Until (all entries in $J^{previous}$ previous are 0)
Return ClusterNumber[$n$] and $G_{subgraphs}$ [$n$];

Output: An array of cluster numbers-ClusterNumber[$n$] and a set of sub-graphs- $G_{subgraphs}$
```

Counting sub-graphs is the procedure to count the connected sub-graphs of every decreased matrix and to evaluate the clustering situation during the process of decreasing. A graph traversal algorithm is implemented to count the graph G^{new} of the matrix J^{new}, i.e. breadth first search (BSF), which is denoted as:

$$G_{subgraphs} = \text{BSF\_traversal}(J^{new})$$

The judgment matrix J aggregating from the k-1 observation matrices would serve as the input to the graph-partitioning process. During each step of decreasing $J^{previous}$, a new matrix J^{new} and its adjacency graph G^{new} are generated. Each G^{new} contains a number of sub-graphs. The number of these sub-graphs is in fact the cluster number. The partitioning process will stop until all

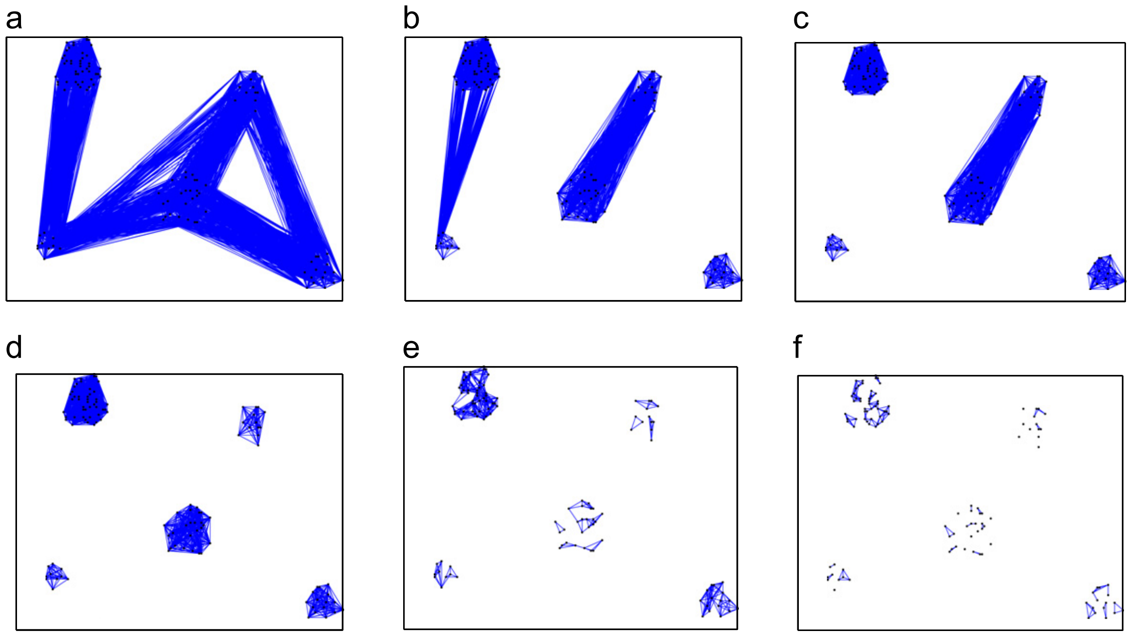
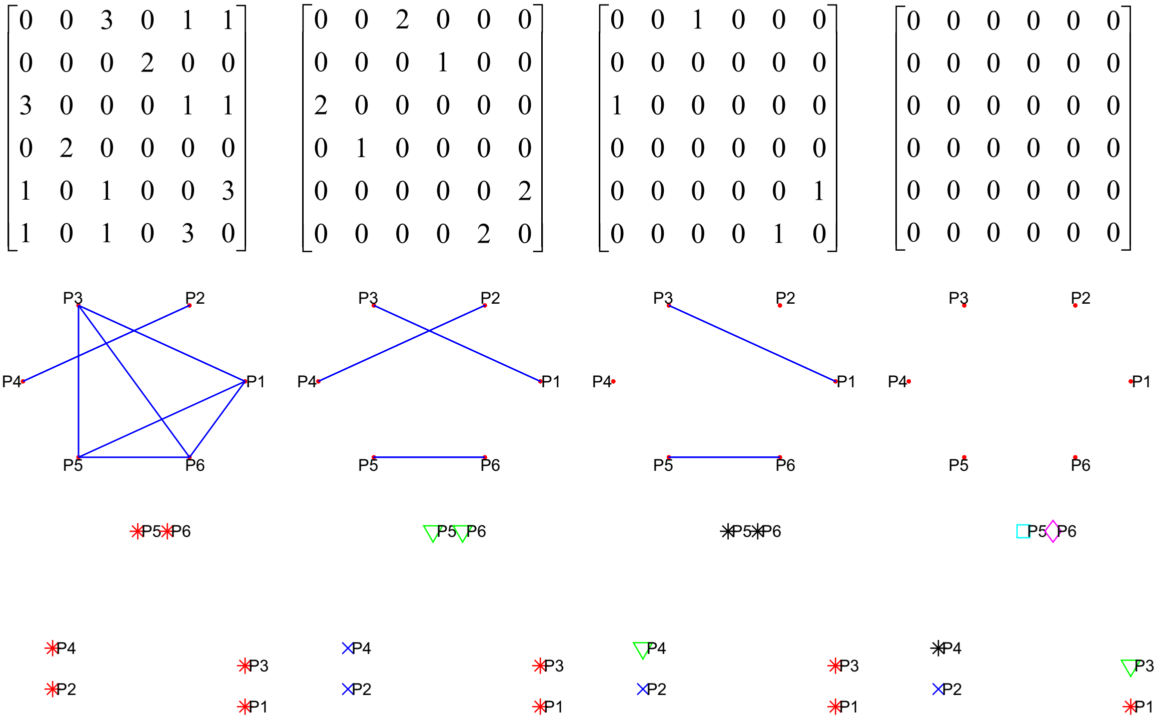


Fig. 7. Decreasing matrices and counting their adjacency graphs: The graph of the original judgment matrix J is shown in (a), after several iterations, the graph is partitioned as three groups of nodes in (b), four groups of nodes in (c), five groups of nodes in (d), and more number of groups after further iterations in (e) and (f).



$$\begin{bmatrix}0&0&3&0&1&1\\0&0&0&2&0&0\\3&0&0&0&1&1\\0&2&0&0&0&0\\1&0&1&0&0&3\\1&0&1&0&3&0\end{bmatrix}\quad\begin{bmatrix}0&0&2&0&0&0\\0&0&0&1&0&0\\2&0&0&0&0&0\\0&1&0&0&0&0\\0&0&0&0&0&2\\0&0&0&0&2&0\end{bmatrix}\quad\begin{bmatrix}0&0&1&0&0&0\\0&0&0&0&0&0\\1&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&1\\0&0&0&0&1&0\end{bmatrix}\quad\begin{bmatrix}0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\end{bmatrix}$$

Fig. 8. Decreased matrices, corresponding adjacency graphs, and cluster results.

entries of $J^{previous}$ are zero. The algorithm is summarised in Table 1.

Fig. 7 visualises the part of the decreasing and counting process. The adjacency graph of the original judgment matrix is depicted in Fig. 7(a). It is shown that all the data points are connected and therefore they belong to only one cluster. In Fig. 7(b), three sub-graphs are shown after a few iterations of graph partitioning, in which the connections between the three sub-graphs are broken off. It means that the connections between these data points are comparatively weaker than those of connecting data points. Three clusters are hence identified. In Fig. 7(c) and (d), four and five clusters are found after further graph partitioning iterations. Even more sub-graphs are generated if the process continues as shown in Fig. 7(e) and (f). As a result, every decreasing and counting step breaks the connections of these sub-graphs and a number of clustering results are found. Nevertheless, it has yet to give answers to the question how the final clustering result and the desired cluster number are obtained.

With reference to the numeric example in Section 3.2, Fig. 8 demonstrates the complete process of decreasing matrices and counting sub-graphs. The first row of Fig. 8 shows the decreasing process in which the judgment matrix is decreased by one degree every time until all entries become zero. The second row shows the adjacency graphs of the resulted matrices. The connections between data points are gradually broken off. The last row of Fig. 8 shows the corresponding clustering results. As shown, many cluster numbers and results are obtained during the decreasing and counting operations. It is explained in the next section how the desired cluster number and the final result can be identified.

3.4. *Desired cluster number and final result identification*

Many works [9–12,16,17,19–21,32,33,35–44] have been reported for identifying optimal cluster numbers and results. Optimal cluster numbers can be found by minimising or maximising clustering validity indices. In the current method, each iteration of the graph partitioning process generates a number of sub-graphs, and the number of sub-graph is exactly the cluster number of that iteration. In the sub-graph number distribution, as shown in Fig. 9, the number of sub-graphs may remain stable for a number of iterations then experience a sharp change after that stable level. The desired cluster number is accordingly defined as the most stable sub-graph number in the distribution because under this sub-graph number, the connections of the data points within the sub-graphs are the strongest and hardest to break. Fig. 9 displays an example of the sub-graph number distribution and the final clustering result. It is obvious from the figure that 4 is the desired cluster number from both the stability of the distribution (a) and from the clustered result (b).

It is important to note that the proposed method does not rely on a single metric for clustering but verifies the relationships of data points in different clustering attempts, in which varied assumptions are considered. It avoids any bias caused by inappropriate assumptions (clustering initialisation). In the proposed method, the relationships of the data points, namely the connections between data points in graphs, are derived from many clustering results.

3.5. *Adaptive implementation*

The proposed clustering method follows an adaptive approach because it can easily work with other clustering algorithms to achieve reliable results and identify desired cluster numbers. Table 2 provides a generic workflow for applying this idea to other algorithms. Examples of integrating the proposed method with *k*-means clustering algorithm is demonstrated in later Section 5.1.

4. Experimental results

The proposed clustering method concludes with the desired cluster number and clustering result by verifying the distribution of the number of sub-graphs. In the section, eight data sets were investigated. The eight data sets include five artificial 2-dimensional data sets with various data shapes, namely Dataset_1, Dataset_2, Dataset_3, Dataset_4 and Dataset_5, and three widely used data sets, Iris data set, Breast Cancer Wisconsin (Diagnostic) data set and Wine data set, which are available on the Internet [45]. An application of the proposed method to mesh segmentation was suggested.

4.1. *Clustering results and cluster number stability*

Dataset_1, Dataset_2, Dataset_3, and Dataset_4 were firstly partitioned by the FCM clustering algorithm with the cluster number ranging from 2 to *k* so as to obtain their judgment matrices. By graph-partitioning the adjacency graphs of these matrices, cluster number distribution and clustering results are shown in Figs. 10, 11, 12 and 13. In these figures, based on the stability of the distribution, the desired cluster numbers can be easily identified. In Fig. 10, for example, the cluster number is

Table 2
Generic workflow for applying the proposed method to other clustering algorithm (OCA).

Input: A data set $Y=\{y_1, \ldots, y_M\}$ in n-dimensional space

1. Using OCA to cluster the data set Y with cluster number ranging from *2* to k;
2. Every clustering result can be represented by a vector $L=[l_1\ l_2\ \ldots\ l_M]$, where $l_i=l_j=c$ if both y_i and y_j belongs to the c-th cluster;
3. An observation matrix O can be computed from the vector L according to $o_{ij}=\begin{cases}1 & l_i=l_j(i\neq j)\\ 0 & \text{otherwise}\end{cases}$;
4. A judgment matrix J can be obtained as the sum of k-1 observation matrices O, i.e., $J=\sum_{C=2}^{k} O_C$;
5. The adjacency graph of J is iteratively partitioned so as to identify the desired cluster number and clustering results according to the distribution of the number of sub-graphs.

Output: The desired cluster number and final clustering result.

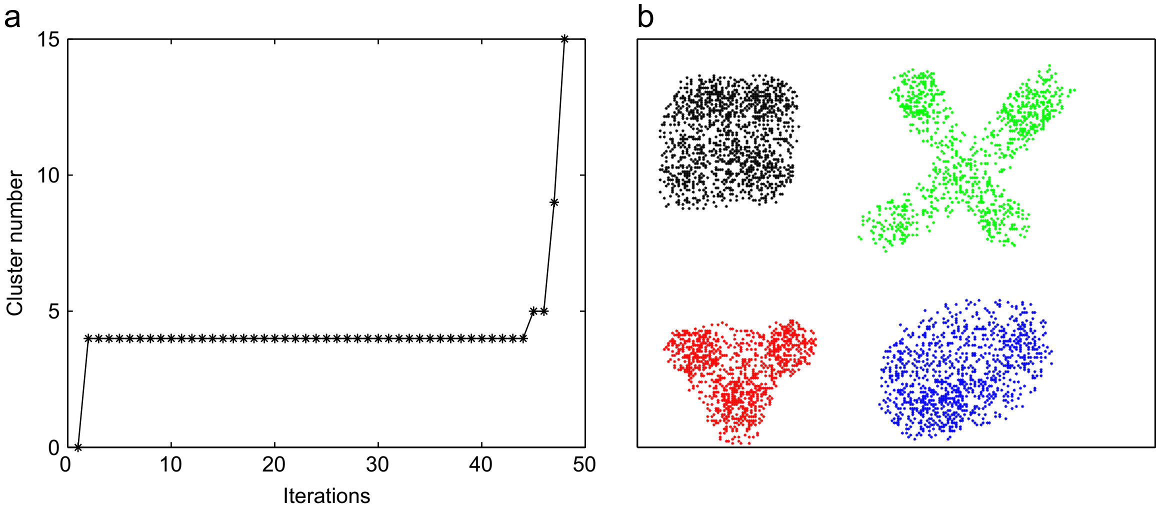


Fig. 9. An example of displaying the distribution of cluster number and the identification of final result. (a) Distribution of the number of sub-graphs and (b) Clustering result.

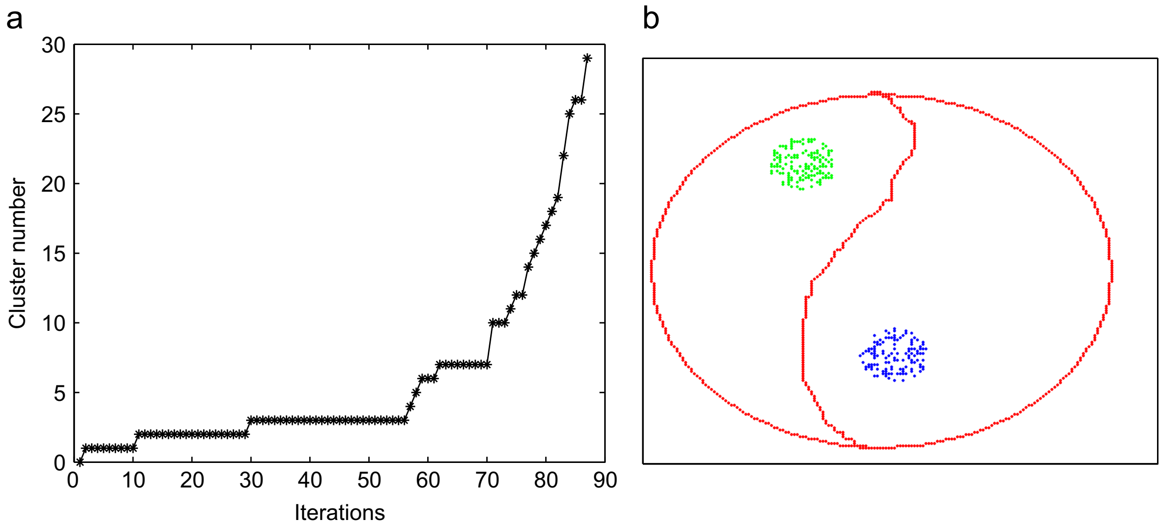


Fig. 10. Dataset_1 with 889 data points and the optimal cluster number=3. (a) Cluster number distribution of Dataset_1 and (b) Clustering result and Dataset_1.

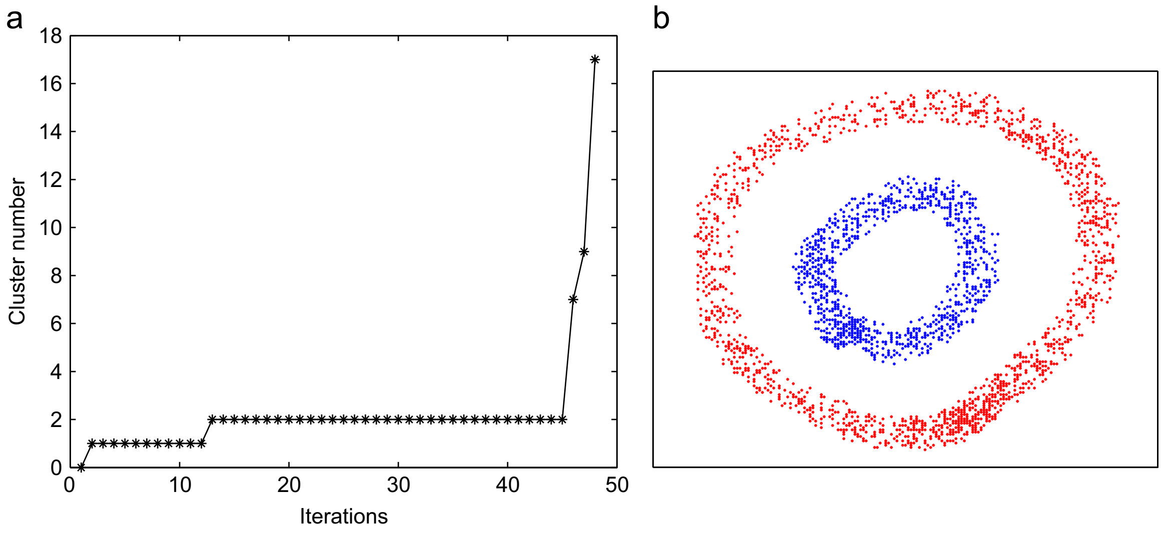


Fig. 11. Dataset_2 with 2298 data points and the optimal cluster number=2. (a) Cluster number distribution of Dataset_2 and (b) Clustering result and Dataset_2.

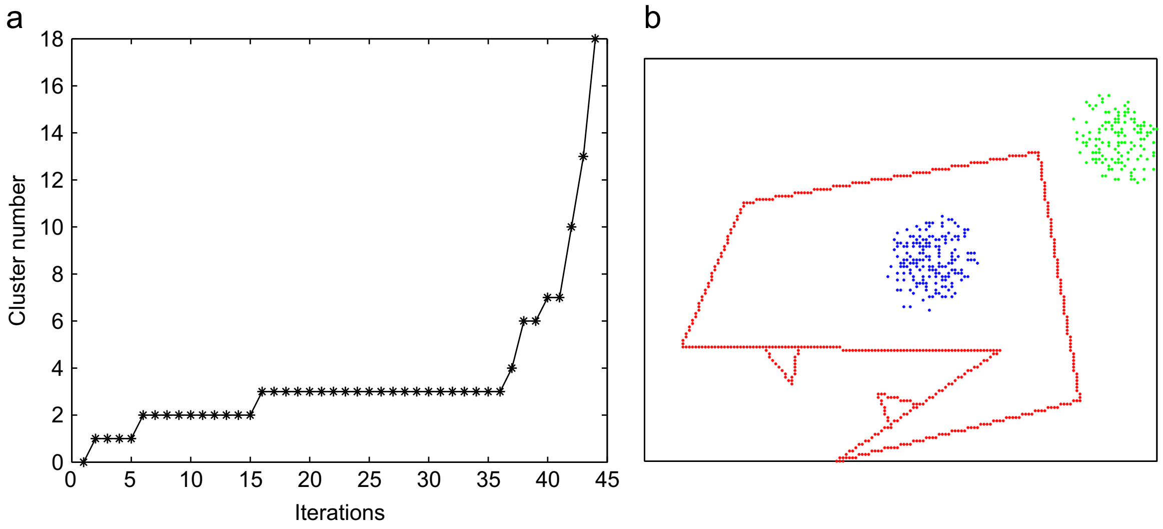


Fig. 12. Dataset_3 with 781 data points and the optimal cluster number=3. (a) Cluster number distribution of Dataset_3 and (b) Clustering result and Dataset_3.

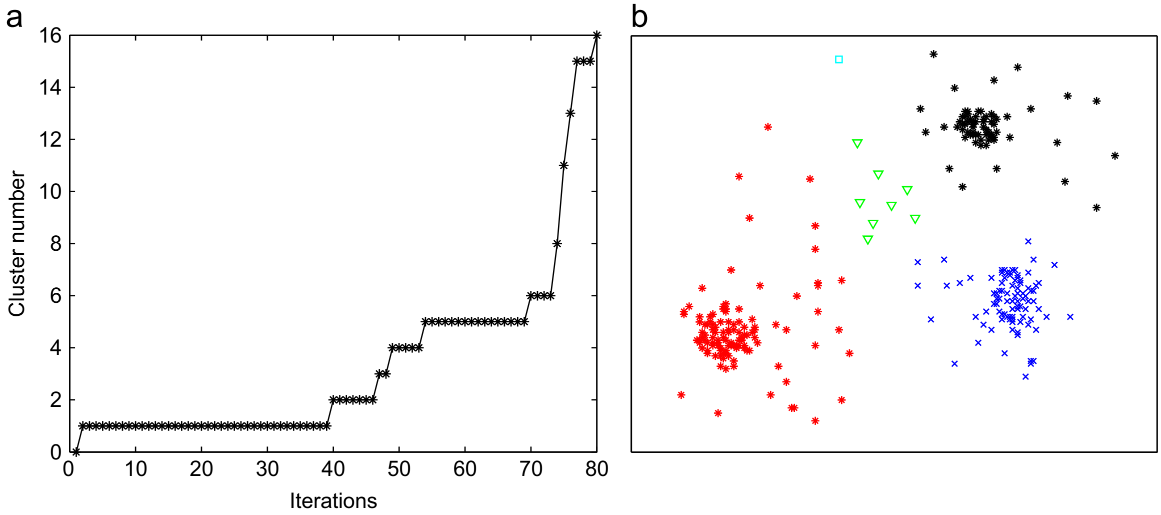


Fig. 13. Dataset_4 with 278 data points and the optimal cluster number=5. (a) Cluster number distribution of Dataset_4 and (b) Clustering result and Dataset_4.

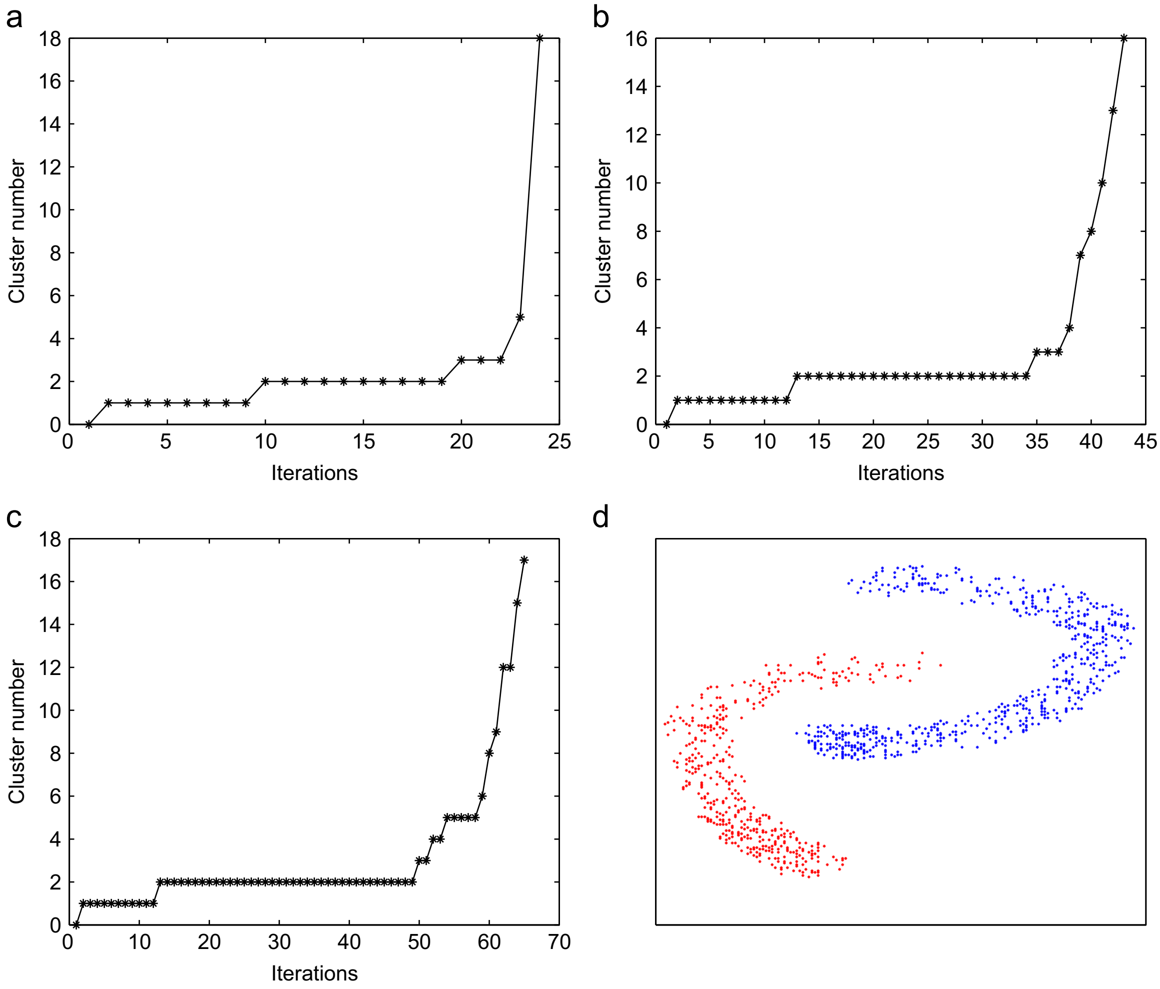


Fig. 14. Effect of different k values on the distribution of cluster number. (a) Cluster number distribution when $k=30$. (b) Cluster number distribution $k=60$. (c) Cluster number distribution when $k=100$ and (d) Clustering results of Datasets_5 of 1035 data points.

stable on 3 and therefore we conclude that 3 is the desired cluster number for Dataset_1. Moreover, the clustering results of the data sets displayed in Figs. 11 and 12 are far more promising than those in Fig. 2, which were partitioned by direct implementation of FCM clustering. FCM is known as being sensitive to outliers or noises. The current method can overcome this limitation. Fig. 13 shows the clustering result of a data set with much noise. Because of data noise, 1 was considered as the most stable number. In clustering analysis, data are usually grouped into more than one cluster. Therefore, the second-most stable number, which is 5 in this case, is thus chosen as the desired cluster number. If the data set contains some outliers, it is expected that the distribution should be stepwise at the beginning, but the desired cluster number can still be identified by checking the most stable cluster number in the distribution. These experiments show that the proposed clustering method is robust and independent of the data shapes, needing no assumption on the data sets.

It is interesting to find out how the parameter k, namely the upper bound of the cluster number, is defined. In related work [18,35] of cluster validity index, $k \leq \sqrt{N}$ were chosen. It is observed from the experiments that increasing the value of k (k must be smaller than N) would generate a more reliable cluster number distribution for easy identification of the desired cluster number. Fig. 14 shows the different distributions when Dataset_5 was clustered using different k values. When $k=30$ in Fig. 14(a), the distribution shows 2 could be a choice of desired cluster number. When k is increased to 60 and 100 in Fig. 14(c) and (d), cluster number 2 becomes overwhelming as the desired cluster number. Consequently, when the distribution of cluster number is not stable enough to give the desired number, increasing the upper bound of cluster number can help identify the desired cluster number. This can be proved again by clustering real-world datasets in the following section. According to iterative graph-partitioning process, the maximum value of k should be equal to the maximum entry value of the obtained judgement matrix J.

4.2. Verification with real-world data sets

More clustering evaluations were conducted on three widely used data sets including the Iris data set, Breast Cancer Wisconsin (Diagnostic) data set and Wine data set. These real world multi-dimensional data sets are used to verify the proposed clustering method. The verification included two parts. It first verified the distribution stability of the cluster number, from which the desired cluster numbers were identified. The identified desired cluster numbers were then compared to different cluster validity index methods.

The Iris data set has 150 samples of 4-dimentional data points. In this data set, two of the three clusters are hardly distinguishable and the third one is well separated from the other two [11]. Fig. 15 shows the cluster number distribution of the Iris data set. As shown in Fig. 15(a), the cluster number is stable on 2 and 3 equally, when k is 80. In Figs. 15(b), 3 becomes slightly more dominant than 2 when k is increased to 140. Therefore, choosing $C=2$ or 3 could be both acceptable for the Iris data set. The Breast Cancer (Diagnostic) data set has is 569 samples, each with 32

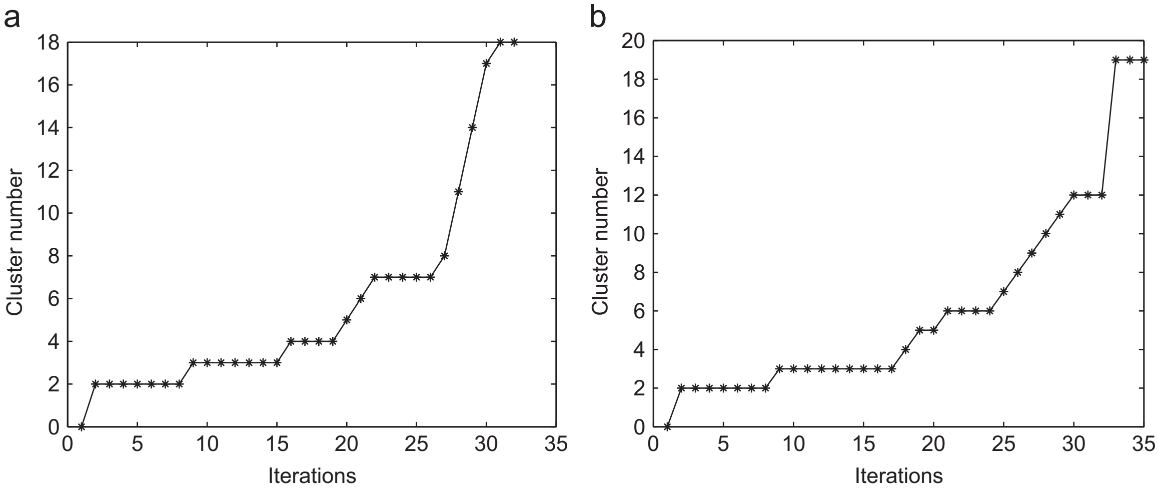


Fig. 15. Cluster number distribution of Iris data set with two different k. (a) $k=80$ and (b) $k=140$.

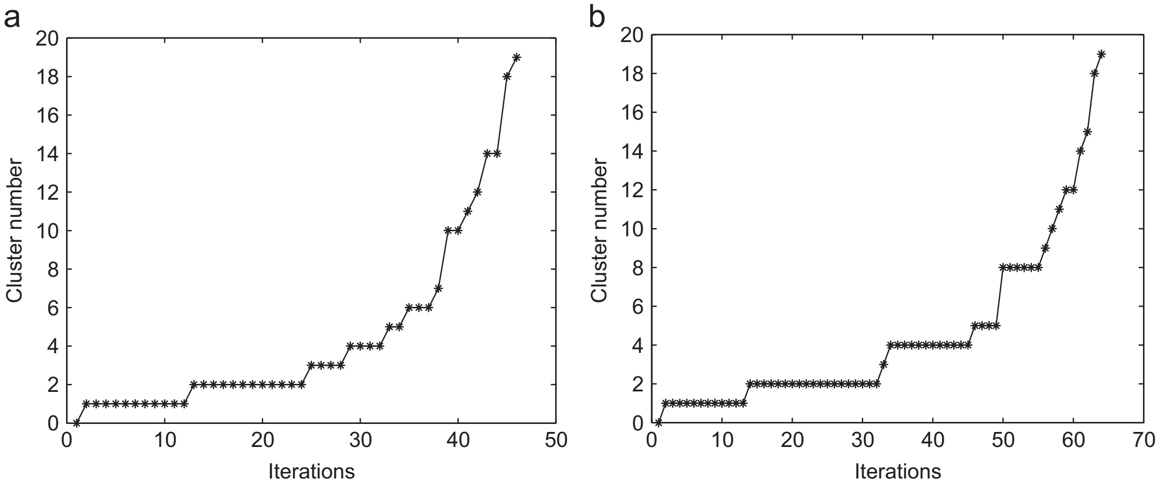


Fig. 16. Cluster number distribution of Breast Cancer Wisconsin (Diagnostic) data set with two different k. (a) $k=80$ and (b) $k=160$.

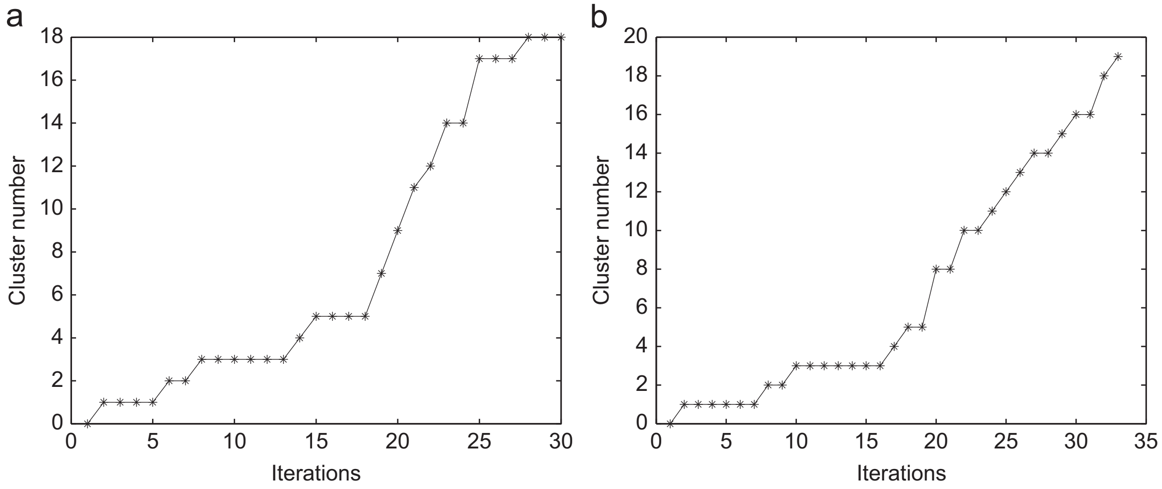


Fig. 17. Cluster number distribution of Wine data set with two different k. (a) k=60 and (b) k=160

Table 3
Cluster numbers obtained by V_{PC} [16], V_{PE} [17], V_{XE} [18], V_{FS} [19], V_W [11] and the proposed method with two different k values, k1 and k2.

Data sets	C^*	V_{PC}	V_{PE}	V_{XB}	V_{FS}	V_W	k1	k2
Iris	2 or 3	2	2	2	5	3	2 or 3 (k=80)	3 (k=140)
Breast cancer (Diagnostic)	2	2	2	2	4	2	2 (k=80)	2 (k=160)
Wine	3	2	2	3	13	3	3 or 5 (k=60)	3 (k=160)

attributes (2 are non-numeric). Fig. 16 indicates that 2 is always overwhelming and can be concluded as the desired cluster number when k is 80 or 160. The Wine data set has 178 samples in 13-dimensional space. Similarly, when k is 60, there are two stable levels in Figs. 17, and 3 is slightly more stable than 5. When k is increased to 160, 3 is shown to be the desired cluster number for the Wine data set.

In Table 3, cluster numbers computed from optimising cluster validity indices of V_{PC} [16], V_{PE} [17], V_{XB} [18], V_{FS} [19], V_W [11] are compared with the results of the current method. The column C^* shows the actual cluster numbers of the three data sets [45]. The desired cluster numbers obtained by the proposed method with two different values of k are shown in the table. It is shown that the proposed method is effective in identifying the optimal cluster numbers in all three real-world multi-dimensional data sets.

4.3. Application to unsupervised mesh partitioning

Triangular mesh partitioning, as a clustering application, is described in this section to evaluate the practical use of the proposed method. Mesh partitioning is defined as segmenting a shape into a number of patches that are uniform with respect to some properties, for example curvature or distance to a fitting plane. Katz and Tal [5] proposed a fuzzy clustering method to segment the triangular mesh based on the angular distance and geodesic distance between two triangles on the mesh surface. But manually specifying a cluster number is a necessary step in their segmentation process. In the current method, the desired cluster number can be identified automatically. A 3D cube and a triangular bi-pyramid were tested. The metrics for clustering were triangle normals on the surface. In Fig. 18(a) and (c), the cluster numbers are dominantly stable on 6, which means that the six different regions of the mesh surface can be identified, as shown in (b) and (d).

5. Discussions

5.1. Adaptive implementation with k-means clustering algorithm

It has been demonstrated that the proposed clustering method is effective in clustering data sets of various shapes and in mesh partitioning applications. As discussed in Section 3.5, the proposed method has an adaptable nature that it can work with other clustering algorithms. K-means clustering method is used as an example in this section to explain this adaptive approach. K-means clustering aims to partition any given data set Y_M into C ($C<M$) partitions S=S_1, S_2, …, S_C so that the sum-of-squares,

$$\arg\min_S \sum_{i=1}^{C} \sum_{y_j \in S_i} \|y_j - \mu_i\|^2 \tag{7}$$

is minimised. μ_i is the geometric centroid of the data points in S_i. According to the described workflow in Table 2, the data set is first clustered according to the k-means algorithm with cluster numbers ranging from 2 to k to obtain vectors L. k-1 observation matrices are computed from these vectors and the judgment matrix J is then obtained. Finally, an iterative graph-partitioning process is conducted to identify cluster number and clustering result.

Fig. 19 shows the cluster number distributions and results obtained by implementing the proposed method with k-means algorithm. Comparing with Fig. 3, it is shown that the proposed method can generate more reliable clustering results. It is thus demonstrated that the proposed method is flexible and adaptive, can be implement with different clustering algorithms to generate results and identify desired cluster numbers. If the clustering algorithm to implement with the current method does not involve the calculation of membership functions, Eqs. (1–4) are not used. The clustering results obtained from a specific

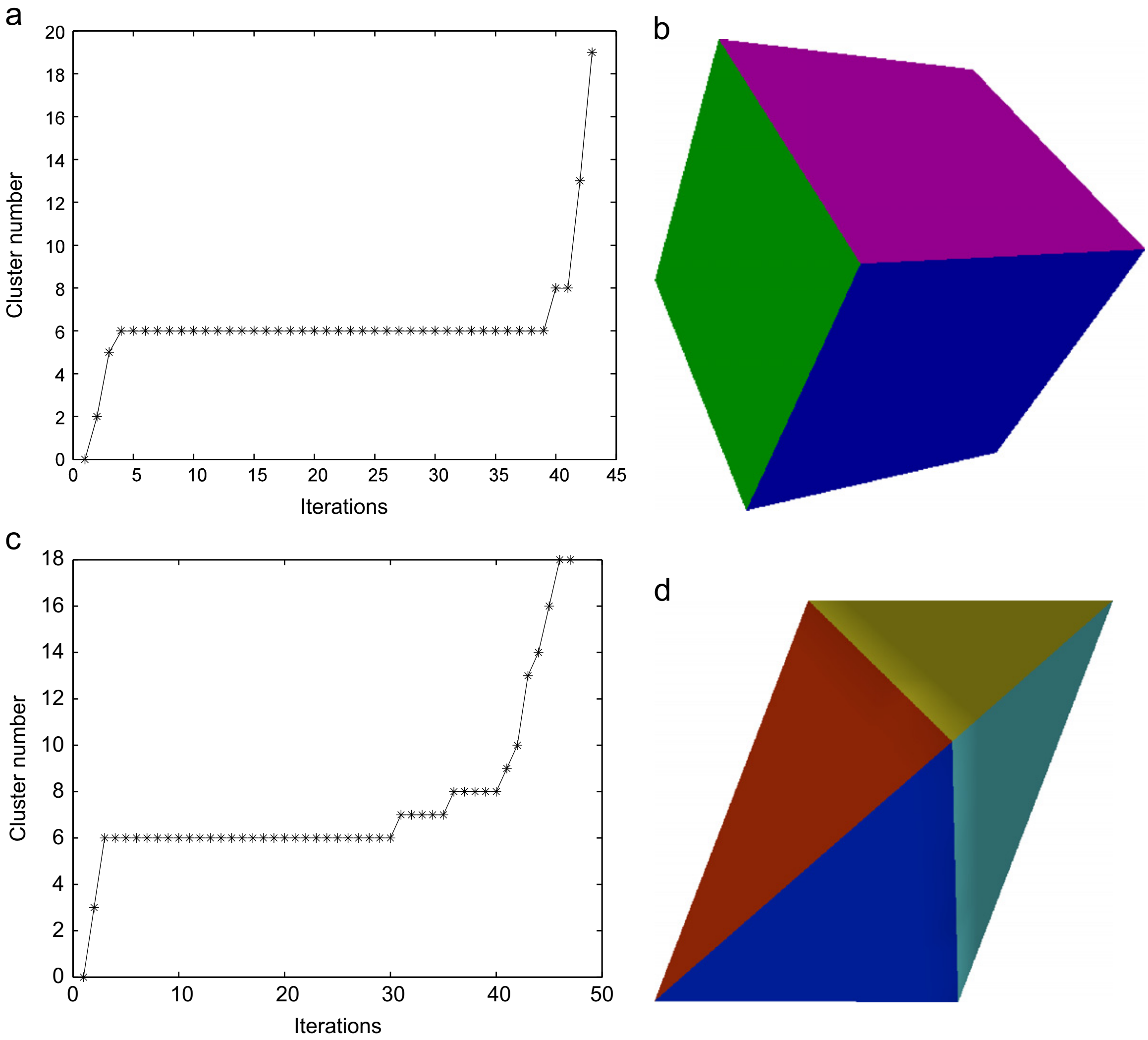


Fig. 18. Clustering application to mesh partitioning with optimal cluster number automatically identified. (a) Cluster number distribution of a 3D cube (b) Mesh partitioning result (c) Cluster number distribution of a triangular bi-pyramid and (d) Mesh partitioning result.

algorithm are summarised in vectors *L*, from which observation matrices *O* and judgement matrix *J* are calculated.

5.2. *Comparison with the spectral clustering method*

In this section, the proposed clustering method is compared with spectral clustering algorithms. Similar to the proposed method, spectral clustering algorithms have a strong connection with graph theory. A complete, weight undirected graph is constructed for a given dataset, where the nodes correspond to data and the edges define the adjacency relationship between data. In such framework, clustering is a graph cutting problem to separate a set of nodes. Spectral clustering method relaxes the complexity of the graph cutting optimisation problem by spectral decomposition of the Lapalcian matrix of the given dataset. It gives useful information about the properties of the graph, in which particular eigenvalue of the Laplacian matrix relates to graph cut and the corresponding eigenvector can cluster together similar data [22]. Different from using graph to represent data relationship in spectral clustering methods, the current method uses graph to consolidate a number of clustering results obtained based on a chosen algorithm, and graph cutting is then used to obtain a more reliable clustering result by consolidating and verifying various clustering results. In this section, some artificial datasets are used to compare the current method with spectral clustering algorithm [22].

Fig. 20 shows the clustered results obtained from the spectral clustering method (a, b, and c) and the current method (d, e, and f). Firstly, (d–f) shows better clustered results than those in (a–c). Secondly, the proposed method in this paper (d–f) is an unsupervised process. No cluster number is needed to define. In the spectral clustering, cluster numbers are required as input parameters, yet the results are comparatively less promising.

The current method is adaptable to different clustering algorithms. The current method is implemented with the spectral algorithm [24]. Firstly, the spectral algorithm is used to cluster data, resulting in a set of vectors *L*, from which observation matrices *O* are computed. Secondly, the judgment matrix *J* is computed by aggregating these observation matrices. Fig. 21 demonstrates that the proposed method can generate reliable results by integrating with clustering algorithms, which are not built upon membership functions.

5.3. *Comparison with cluster ensembles*

By reviewing the literature, it is found that the current method has a similar concept as cluster ensemble methods. It is therefore interesting to compare the current method with other cluster ensemble methods. In this section, the proposed clustering

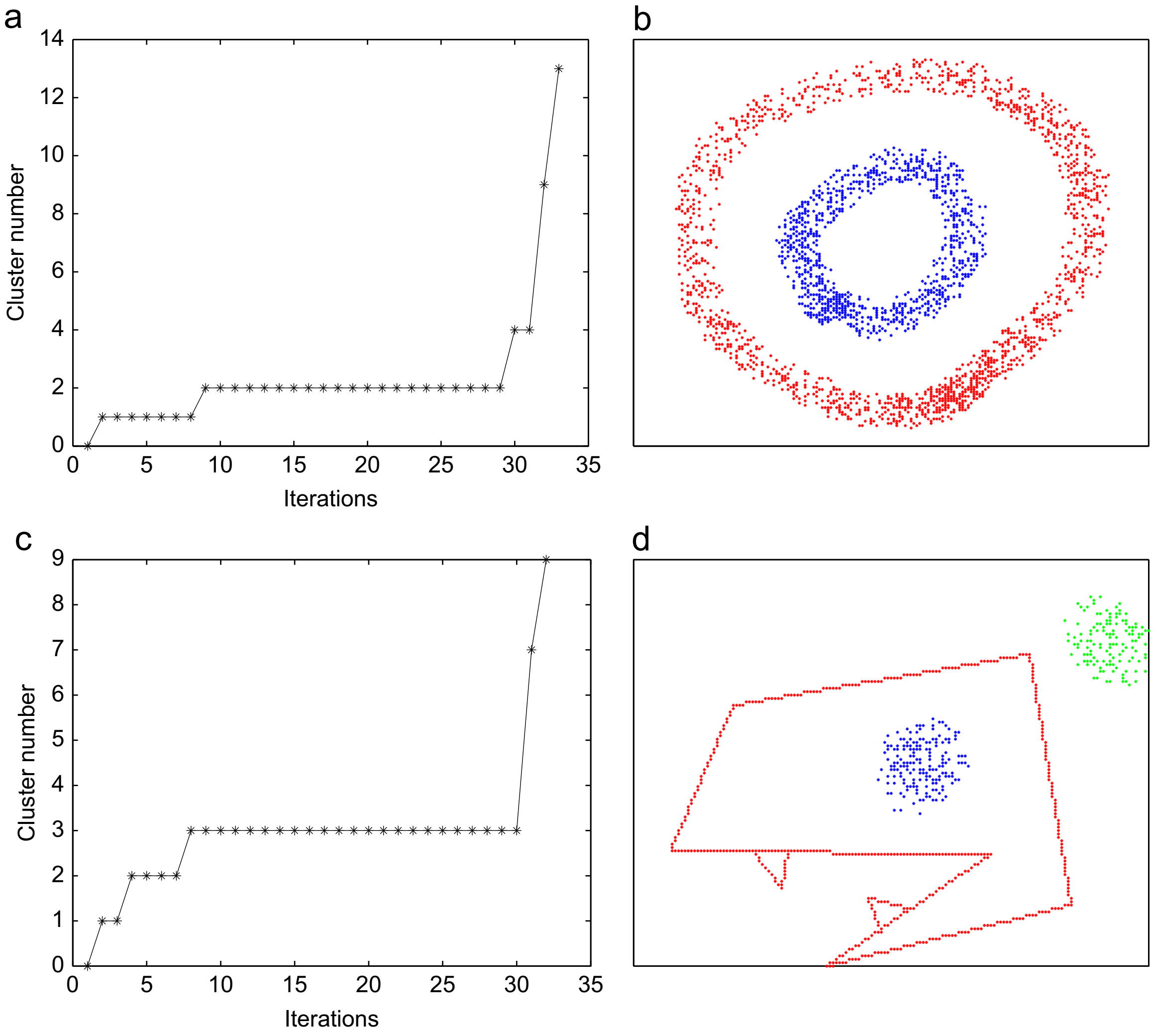


Fig. 19. Clustering results and cluster number distributions for Dataset_2 and Dataset_3 using *k*-means clustering algorithm. (a) Cluster number distribution of Dataset_2 (b) Clustering result and Dataset_2 (c) Cluster number distribution of Dataset_3 and (d) Clustering result and Dataset_3.

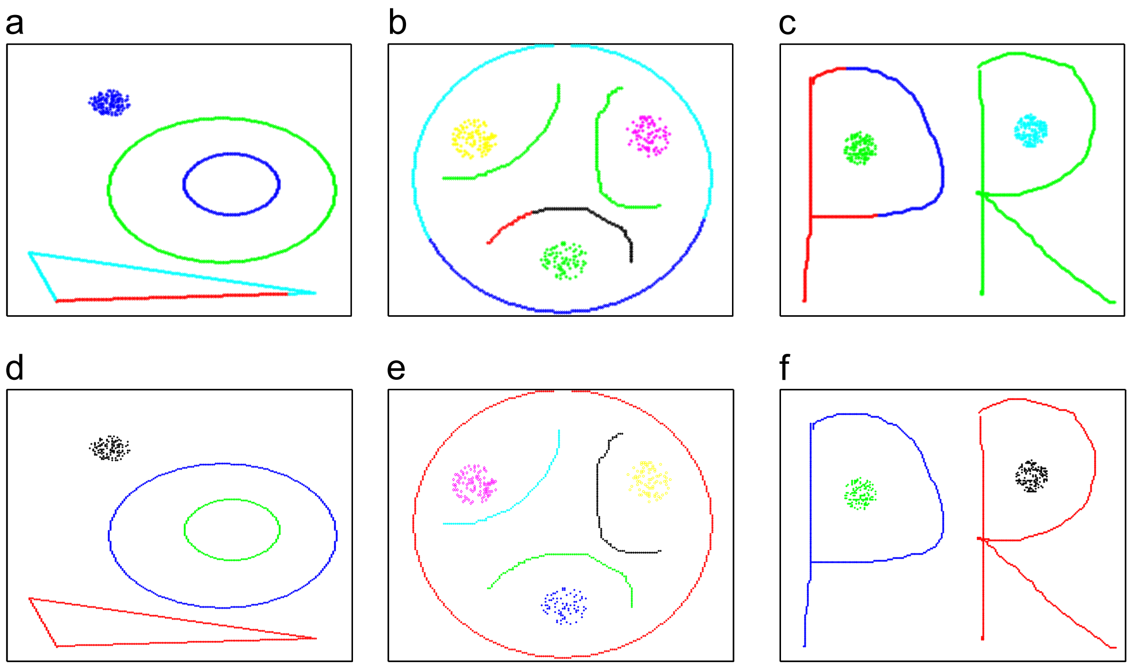


Fig. 20. Clustering results comparison: (a–c) results from the spectral clustering method [24]; (d–f) results from the proposed method.

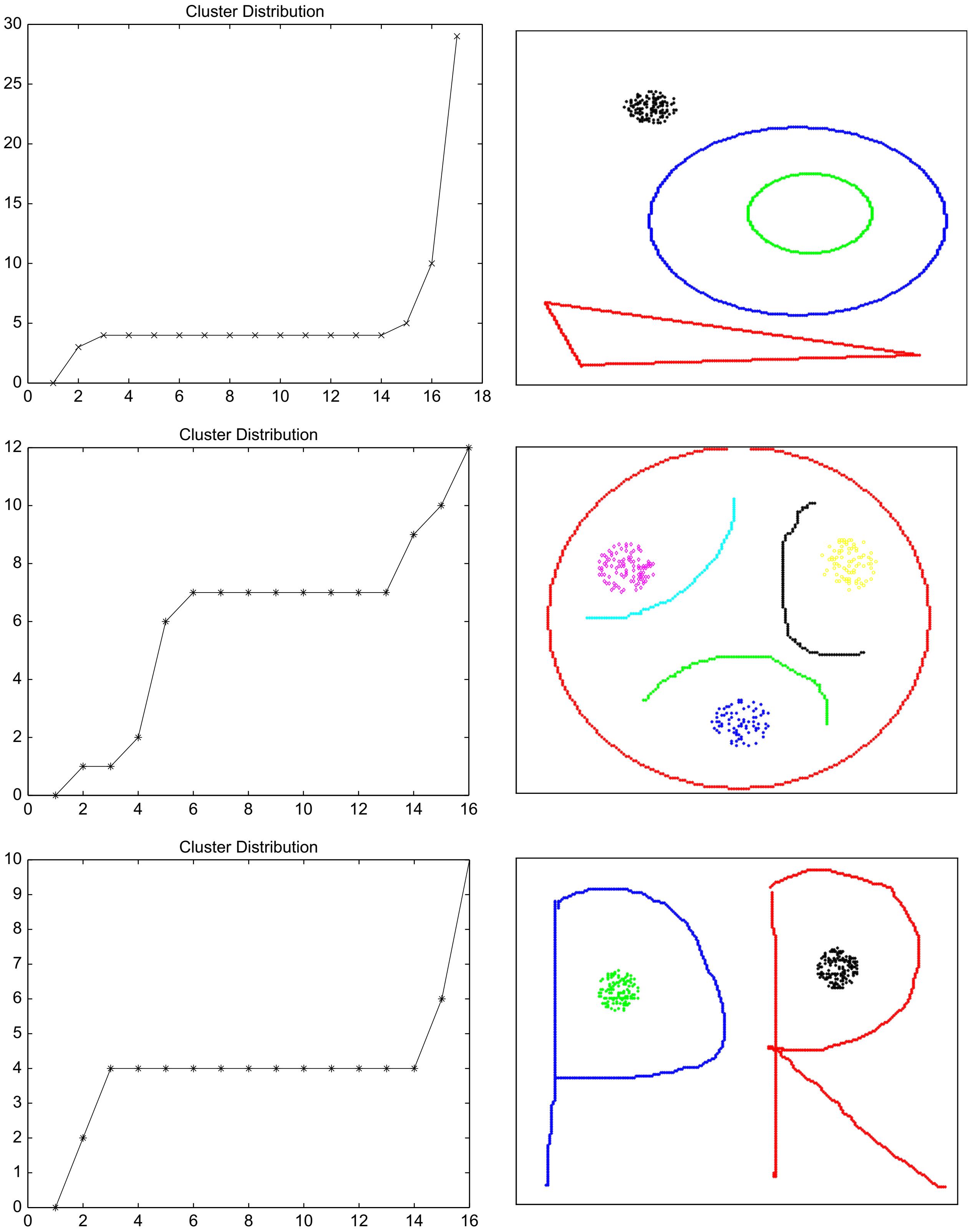


Fig. 21. Clustering results and cluster number distributions based on adaptation of the spectral algorithm [24].

method is compared with hyper-graph ensemble clustering algorithms including Cluster-based Similarity Partitioning Algorithm (CSPA), Hyper Graph Partitioning Algorithm (HPGA), and Meta-Clustering Algorithm (MCLA) [30], and co-association algorirthms including single link (SL) and average link (AL) based evidence accumulation clustering (EAC) [28], weighted cluster ensemble using a kernel consensus function (denoted as WKF) [30] and Generalised WKF (denoted as GWKF) [31].

The current method can indeed be viewed as a clustering ensemble method because the judgment matrix is integrated

Table 4
Clustering error rate comparison with k-means method and different cluster ensemble methods.

Dataset|Method	k-means$_{\text{avg}}$	CSPA	HPGA	MCLA	EAC-SL	EAC-AL	WKF	GWKF	RAC-FCM
Iris	18.1	13.3	37.3	11.2	11.1	11.1	10.6	10.8	2.7
Breast cancer (Original)	3.9	17.3	49.9	3.8	4.0	4.0	3.7	3.7	3.5

from many clustering solutions and also the final result is computed from this matrix. Compared with other cluster ensemble methods, the current method adopts a different way of integrating or consolidating cluster results, a judgement matrix is defined to represent the relationship between data points but not simple distance-base similarity between data points, but the definition of this relationship matrix is different from co-association matrices in [28,30]. In addition, the matrix is partitioned differently, in the current method it is done by an iterative graph partitioning process while in [28] the matrix is analysed by the single-link agglomerative approach.

In order to compare the current method with the other cluster ensemble methods, the Iris and Breast Cancer (Original) datasets [45] are used for evaluation. In Table 4, the error rates are calculated by comparing the clustering results of various methods with true underlying groups of the data in the dataset. The average error rates of k-means algorithms with 20 different parameter initialisations are also shown in the table for comparison. The current method is implemented based on FCM algorithm, thus denoted as RAC-FCM. It is shown form the table that the current method obtains lower error rates than other ensemble methods, thus it is effective in clustering Iris and Brest Cancer (Original) datasets.

6. Conclusions

In this paper, a robust adaptive clustering method has been proposed and implemented. The idea stems from the human decision process. In general, people make decisions after collecting and evaluating different opinions. An opinion, from a clustering analysis perspective, is a partitioned result obtained from a specific clustering algorithm with certain parameter settings. An observation matrix has been defined to represent a possible clustering result. A number of observation matrices are then computed by conducting Fuzzy C-Mean clustering with different parameter settings. A matrix, called the judgment matrix, has been defined to summarise different observation matrices, from which the desired cluster number and final result can be identified by an iterative graph-partitioning procedure.

Eight data sets, including five artificial 2D data sets of various data shapes and three real-world multi-dimensional data sets, have been used to evaluate the proposed clustering method. This has demonstrated that the proposed method is robust in a sense that data set of different structures can be clustered. The proposed method is shown to be more effective than the Fuzzy C-Means and k-means algorithms, even the optimal cluster numbers are predefined for the latter. Experiments have also indicated that the proposed clustering method is effective in mesh segmentation applications. The method has also been compared with the spectral clustering method and different cluster ensemble methods. It has been demonstrated that the method can integrate with other clustering algorithms like k-means. Therefore, the method is not only robust but also adaptive as it can be readily implemented with other clustering algorithms.

In conclusion, the proposed clustering approach not only identifies the desired cluster number but also ensures reliable clustering results.

Acknowledgments

This work was partly supported by the Research Grants Council of the Hong Kong Special Administrative Region, China (Grant no. PolyU 5254/08E) and the Hong Kong Polytechnic University (Grant nos. RGZQ and A-PD1H). The authors thank all reviewers' for their valuable comments and suggestions.

References

[1] F. Hoppner, F. Klawonn, R. Kruse, T. Runkler, Fuzzy Cluster Analysis, Wiley Press, Chichester, 1999.
[2] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Computing Surveys 31 (1999) 264–323.
[3] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.
[4] W. Cai, S. Chen, D. Zhang, Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation, Pattern Recognition 30 (3) (2007) 825–838.
[5] S. Katz, A. Tal, Hierarchical mesh decomposition using fuzzy clustering and cuts, ACM Transactions on Graphics (SIG_GRAPH) 22 (3) (2003) 954–961.
[6] W.M.E. Ma, W.S.T. Chow, A new shifting grid clustering algorithm, Pattern Recognition 37 (3) (2004) 503–514.
[7] J.H. Wen, K.W. Meng, H.Y. Wu, Z.F. Wu, A novel clustering algorithm based upon a SOFM neural network family, in: J. Wang, X. Liao, Z. Yi (Eds.), Lecture Notes in Computer Sciences, 3497, 2005, pp. 69–74.
[8] M. Masson, T. Denoeux, ECM: An evidential version of the fuzzy c-means algorithm, Pattern Recognition 41 (4) (2008) 1384–1397.
[9] D.W. Kim, K.H. Lee, D. Lee, On cluster validity index for estimation of the optimal number of fuzzy clusters, Pattern Recognition 37 (2004) 2009–2025.
[10] W. Wang, Y. Zhang, On fuzzy cluster validity indices, Fuzzy Sets and Systems 158 (19) (2007) 2095–2117.
[11] Y. Zhang, W. Wang, X. Zhang, Y. Li, A cluster validity index for fuzzy clustering, Informat. Sci 178 (2008) 1205–1218.
[12] K. Wu, M. Yang, J. Hsieh, Robust cluster validity indexes, Pattern Recognition 42 (2009) 2541–2550.
[13] I. Berget, B. Mevik, T. Naes, New modifications and applications of fuzzy c-means methodology, Computational Statistics and Data Analysis 52 (5) (2008) 2403–2418.
[14] F. Kovács, C. Legány, A. Babos, Cluster Validity Measurement Techniques, in: Proceedings of the fifth WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases, Madrid, Spain, (2006), pp. 388–393.
[15] J.F. Hair, Multivariate Data Analysis, Prentice Hall/Pearson, Upper Saddle River, NY, 2010.
[16] J.C. Bezdek, Cluster validity with fuzzy sets, Journal of Cybernet 3 (1974) 58–72.
[17] J.C. Bezdek, Numerical taxonomy with fuzzy sets, Journal of Mathematical Biology 1 (1974) 57–71.
[18] X.L. Xie, G. Beni, A validity measure for fuzzy clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (8) (1991) 841–847.
[19] Y. Fukuyama, M. Sugeno, A New Method of Choosing the Number of Clusters for the Fuzzy C-Means Method, Proceedings of Fifth Fuzzy Systems Symposium, 1989, pp. 247–250.
[20] M. Halkidi, Y. Batistakis, M. Vazirgiannis, On clustering validation techniques, Journal of Intelligent Information Systems 17 (2001) 107–145.
[21] J.C. Bezdek, N.R. Pal, Some new indices of cluster validity, IEEE Transactions on Systems, Man and Cybernet 28 (1998) 301–315.
[22] M. Filippone, F. Camastra, F. Masulli, S. Rovetta, A survey of kernel and spectral methods for clustering, Pattern Recognition 41 (2008) 176–190.
[23] Y. Weiss, Segmentation Using Eigenvectors: A Unifying View, in: International Conference on Computer Vision, 1999.
[24] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 888–905.
[25] A. Ng, M. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm In Advances in Neural Information Processing System (2002) 849–856.
[26] U. Luxburg, A tutorial on spectral clustering, Statistics and Computing, Kluwer Academic Publishers 17 (2007) 395–416.
[27] Z. He, X. Xu, S. Deng, A cluster ensemble method for clustering categorical data, Information Fusion 6 (2005) 143–151.

[28] A.L.N. Fred, A.K. Jain, Combining multiple clusterings using evidence accumulation, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2005) 835–850.

[29] A. Strehl, J. Ghosh, Cluster ensembles: a knowledge reuse framework for combining multiple partitions, Journal of Machine Learning Research 3 (2002) 583–617.

[30] S. Vega-Pons, J. Correa-Morris, J. Ruiz-Shulcloper, Weighted cluster ensemble using a kernel consensus function, in: J. Ruiz-Shulcloper, W.G. Kropatsch (Eds.), CIARP 2008, 5197, Lecture Notes in Computer Science, 2008, pp. 195–202.

[31] S. Vega-Pons, J. Ruiz-Shulcloper, Clustering ensemble method for heterogeneous partitions, in: E. Bayro-Corrochano, J.-O. Eklundh (Eds.), CIARP 2009, 5856, Lecture Notes in Computer Science, 2009, pp. 481–488.

[32] J. Hespanha, An efficient MATLAB Algorithm for Graph Partitioning. Technical Report, University of California, 2004.

[33] B.W. Kernighan, S. Lin, An efficient heuristic procedure for partitioning graphs, The Bell System Technical Journal (1970) 291–307.

[34] G. Karypis, V. Kumar, Multilevel-way Partitioning Scheme for Irregular Graphs, Journal of Parallel and Distributed Computing 48 (1) (1998) 96–129.

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

[36] R. Kothari, D. Pitts, On finding the number of clusters, Pattern Recognition Letters 20 (1999) 405–416.

[37] N. Zahid, M. Limouri, A. Essaid, A new cluster-validity for fuzzy clustering, Pattern Recognition 32 (1999) 1089–1097.

[38] M.K. Pakhira, S. Bandyopadhyay, U. Maulik, Validity index for crisp and fuzzy clusters, Pattern Recognition 37 (2004) 481–501.

[39] M.R. Rezaee, B.P.F. Lelieveldt, J.H.C. Reiber, A new cluster validity index for the fuzzy c-mean, Pattern Recognition Letters 19 (1998) 237–246.

[40] H.B. Silva, P. Brito, J.P.D. Costa, A partitional clustering algorithm validated by a clustering tendency index based on graph theory, Pattern Recognition 39 (2006) 776–788.

[41] G.E. Tsekouras, H. Sarimveis, A new approach for measuring the validity of the fuzzy c-means algorithm, Advances in Engineering Software 35 (2004) 567–575.

[42] M.P. Windham, Cluster validity for fuzzy clustering algorithms, Fuzzy Sets and Systems 5 (1981) 177–185.

[43] K.L. Wu, M.S. Yang, A cluster validity index for fuzzy clustering, Pattern Recognition Letters 26 (2005) 1275–1291.

[44] J. Yu, C.X. Li, Novel cluster validity index for FCM algorithm, Journal of Computer Science and Technology 21 (1) (2006) 137–140.

[45] ⟨http://archive.ics.uci.edu/ml/⟩.

P.Y. MOK graduated from the University of Hong Kong with a PhD in engineering in 2002. Before joining the Hong Kong Polytechnic University in 2005, she conducted different research projects in artificial intelligence. Her current research interests are garment pattern engineering, fashion CAD and cloth simulations, and RFID applications.

H.Q. HUANG currently is a PhD candidate in the Institute of Textiles and Clothing of The Hong Kong Polytechnic University. He received his MSc degree in Computer Science from University of Bath, UK, in 2006 and BEng degree in Clothing Design and Engineering from Beijing Institute of Clothing Technology in 2001. His research interests are human body data visualisation, garment 3D CAD, clothing ergonomics, and body recognition.

Y.L. KWOK gained her PhD from the University of Leeds in 1993. She is currently an Associate Professor specialized in product development, clothing technology, clothing properties, pattern engineering and pattern construction.

J.S. AU gained his PhD from Institute of Textiles and Clothing, The Hong Kong Polytechnic University in 2003 and Master of Design degree in College of Fine Arts of The University of New South Wales in 1998. He received his Associate in Applied Science Degree in Fashion Institute of Technology, The State University of New York. His current research interests are fashion design theory and fashion pattern technology.