Neurocomputing 329 (2019) 407–423

Contents lists available at ScienceDirect

Neurocomputing

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

Possibilistic fuzzy clustering with high-density viewpoint

Yiming Tang[a,b], Xianghui Hu[a], Witold Pedrycz[b,c,d,*], Xiaocheng Song[a]

[a] Anhui Province Key Laboratory of Affective Computing and Advanced Intelligent Machine, School of Computer and Information, Hefei University of Technology, Hefei, Anhui 230601, China
[b] Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6R 2V4, Canada
[c] Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia
[d] Systems Research Institute, Polish Academy of Sciences, Warsaw 01-447, Poland

ARTICLE INFO

Article history:
Received 30 May 2018
Revised 20 September 2018
Accepted 2 November 2018
Available online 7 November 2018

Communicated by Dr. Nianyin Zeng

Keywords:
Fuzzy clustering
Knowledge-driven
Density
Viewpoint
Clustering center initialization

ABSTRACT

Fuzzy clustering algorithms are usually data-driven. Recently, knowledge has been introduced into these methods to form knowledge-driven and data-driven fuzzy clustering algorithms. However, these algorithms still have the problems of sensitivity to clustering center initialization and a lack of robustness, in general. There is a genuine need for a sound acquisition of viewpoints. In this study, a new fuzzy clustering algorithm driven by data and knowledge named Density Viewpoint-induced Possibilistic Fuzzy C-Means (DVPFCM) is put forward. To begin with, we propose a new method to calculate the density radius, which determines the density range of each data point. Based on this, we establish a Hypersphere Density-based Clustering Center Initialization method (HDCCI), which can obtain the initial clustering centers located in the denser region of the dataset. Furthermore, the high density point obtained by the HDCCI method is taken as a new viewpoint and integrated into the clustering algorithm. The new viewpoint helps to speed up the convergence of the algorithm. It can also guide the clustering algorithm to discover the data structure. Finally, on the basis of the HDCCI method, the idea of high-density viewpoint is introduced, and the advantages of FCM (Fuzzy C-Means) and PFCM (Possibilistic Fuzzy C-Means) are combined, and then the DVPFCM algorithm is proposed. Through experimental studies including some comparative analyses, it is demonstrated that the DVPFCM algorithm is better in several different ways in terms of initializing clustering centers and values of some performance indexes. It also exhibits better performance in determining the distance between the computed clustering centers and the reference centers.

1. Introduction

Cluster analysis aims at discovering the structure in a multivariate dataset, which partitions the data into a number of subsets. The data in the same subset exhibit a maximal similarity, and the data in the different subsets possesses is totally dissimilar. Cluster analysis has been applied widely to numerous areas, such as geology, taxonomy, business, engineering systems, medicine and image processing [1–7]. Clustering assign each object to a single category. For instance, the RLM algorithm is an example of clustering [8]. Then, fuzzy clustering has become a new and hot direction in this field, which uses fuzzy memberships to describe the extent to which each sample point belongs to a class.

Fuzzy C-Means (FCM) is the generic algorithm of fuzzy clustering [9]. Its goal is to maximize the compactness within the class and the separation between classes. However, the FCM algorithm is not only sensitive to the initialization of clustering centers, but also susceptible to noise points. Krishnapuram and Keller [10] proposed the Possibilistic C-Means algorithm (PCM). PCM uses a possibilistic partition matrix, which reflects the typicality of a data point relative to a clustering center. In addition, it relaxes the constraint presented in FCM stating that the sum of membership values is equal to 1. Compared with FCM, PCM has a better ability to deal with noise points. However, the PCM algorithm is also sensitive to initialization. Without appropriate initialization, one may have overlapping classes or fewer clusters than the present number of clusters [11,12]. The Possibilistic Clustering Algorithm (PCA) [13] and the Cutset-type Possibilistic C-Means clustering algorithm (C-PCM) [14] were improvements of PCM at this point. Pal et al. [15] proposed the Possibilistic Fuzzy C-Means algorithm (PFCM), which took into account both the memberships and the possibilities. PFCM solves the problem of overlapping clustering centers

* Corresponding author at: Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6R 2V4, Canada.
E-mail addresses: ymtang@hfut.edu.cn (Y. Tang), xhhu306@163.com (X. Hu), wpedrycz@ualberta.ca (W. Pedrycz), xcsong120@163.com (X. Song).

https://doi.org/10.1016/j.neucom.2018.11.007
0925-2312/

and has strong noise immunity. Until now, there were many improvements of the FCM algorithm [16–21]. For instance, the Euclidean distance was replaced by a kernel-guided distance function to enhance the anti-noise performance of the algorithm [17,18]. Genetic algorithm was introduced into the FCM to improve its performance [19]. Askari et al. [20] proposed a Generalized Entropy based Possibilistic C-Means algorithm (GEPFCM) combining the Entropy C-Means (ECM) algorithm with the PFCM algorithm, which obtained more accurate clustering centers.

Furthermore, domain knowledge can be introduced to support clustering mechanisms and enhance the realization of the overall processing. These knowledge guidelines are proposed by users or experts and can assist the process of clustering. Our research group proposed a knowledge-driven fuzzy clustering algorithm with viewpoints (called V-FCM shortly) [22]. This algorithm implements a personalized because the "viewpoints" represent the content that users are interested in, such as the average, maximum, minimum, etc. Viewpoints are regarded as an integral part of the structure in data so that the clustering results are what the user wants to know. If a viewpoint is the average, we can get the centralized distribution in data. And if the viewpoint is the extreme value, we can obtain the distribution of the data boundary area. Fazendeiro and Valente De Oliveira proposed an observer-based fuzzy clustering algorithm in [23], which simulated the scene in which people observe things in reality. The closer the observer observes a pile of things, the more details the observer can get. Conversely, the farther away the observer is, the less details are observed. But we can know the overall structure at a proper distance. Therefore, the authors express the position of the observer by focal point. The parameter ζ standing in the objective function is used to control the distance from the observer to the target object. Then the number of clusters decreases from $\sqrt{N}$while the ζ increases from zero until the clustering validity index XB gets the optimal value. In this way, the reasonable structure of the observed target is gotten. Experiments showed that without the parameter of cluster numbers given in advance, the observer-based clustering algorithm can automatically find the number of clusters in data. It becomes possible given to the introduction of the "observer" knowledge.

The idea of adopting viewpoints in the V-FCM algorithm is sound and the results are promising. However, there are still several problems with the V-FCM algorithm.

As a result, this study is focused on the following two issues:

- The problem of viewpoints selection. Selected viewpoints in the V-FCM algorithm are directly specified by the user. Therefore, whether or not an algorithm can select a more reasonable viewpoint is an issue that we need to focus on. In particular, the V-FCM algorithm uses average or boundary values that are of interest to the user to represent the viewpoints. The purpose of clustering is to make the distances between classes as far as possible and the intra-class distances as close as possible. For this reason, we use the concept of the density in the RLM algorithm. That means the viewpoint used is the point with the highest density. The densest point is at the center of the data distribution dense area, which is consistent with the characteristics of the clustering centers in the data structure. So it is reasonable to use it as a part of the prototype matrix.
- The noise immunity problem of the algorithm. The V-FCM algorithm introduces the viewpoints into the FCM algorithm, but the FCM algorithm is sensitive to noise points. How to improve this is a problem that we need to consider about. In order to better play the supporting role of the viewpoint, we have extended it to the PFCM algorithm to improve the robustness of the algorithm.

In this study, we propose a Density Viewpoint-induced Possibilistic Fuzzy C-Means (DVPFCM) algorithm. Firstly, in terms of viewpoint selection, the previous density function in the RLM algorithm is improved. The point with the highest density is used, and a new density-based clustering center initialization method is proposed, which is called the Hypersphere Density-based Clustering Center Initialization (HDCCI) method. The HDCCI method can reduce the iteration times of the algorithm and improve the accuracy of the algorithm. Furthermore, this method obtains the data point with the highest density and is not a noise point as the "viewpoint" to guide the algorithm to find the clustering centers more accurately. Secondly, we combine the advantages of FCM and PCM, while avoid the issue of noise sensitivity of the FCM algorithm and of which PCM tends to produce consistent clustering centers. In addition, a new viewpoint is integrated into the algorithm to get the DVPFCM algorithm. The new algorithm results in a more reasonable clustering result. Position the research in a general setting, one can refer to the roadmap displayed in Fig. 1.

The paper is organized as follows. Section 2 describes related works and problems with some algorithms. Section 3 shows the proposed DVPFCM algorithm. Section 4 presents experimental results of DVPFCM and other algorithms on artificial data sets and some data sets of machine learning. Section 5 gives a conclusion and suggestions for further research.

2. Related studies

In this section, we briefly review the "classical" FCM [9] and PFCM [15] algorithms, as well as four other improved algorithms such as PCA [13], GEPFCM [20], V-FCM [22] and C-PCM [14]. The last one is RLM [8] related to our proposed cluster center initialization method (HDCCI).

Assuming that the application object is a set of N samples represented by $X=\{x_k\}_{k=1}^{N}$. We aim to classify it into C classes to get a set of clustering centers $V=\{v_i\}_{i=1}^{C}$. Each sample x_k and clustering center v_i are positioned in the R^l space where l is the data dimension. We set the iteration threshold ε and the maximum number of iterations $iterMax$ in advance, and the initial value of the iteration number is $iter = 0$.

2.1. Classical and advanced clustering algorithms

The FCM algorithm is a classical algorithm of the clustering methods based on the objective function. It employs fuzzy memberships to characterize the extent to which each sample point belongs to a class, and then obtains the final clustering results through the membership matrix. The minimized objective function of the FCM algorithm is expressed as:

$$J_{FCM}=\sum_{k=1}^{N}\sum_{i=1}^{C}u_{ik}^{m}d_{ik}^{2} \tag{1}$$

where u_{ik} is the membership grade of x_k in cluster i, which ranges from 0 to 1 and has the constrain $\sum_{i=1}^{C}u_{ik}=1$ $(k=1,2,\ldots,N)$. $d_{ik}^2=\|x_k-v_i\|^2$ denotes the Euclidean distance between the ith clustering center and the kth sample. $v_i\in R^l$ is the centroid of the ith cluster $(i=1,\ldots,C)$. And $m\in(1,+\infty)$ is a parameter that controls the fuzziness of membership for each sample k.

PFCM is a combination of PCM1 [10] and FCM. Krishnapuram and Keller proposed PCM1 to overcome the sensitivity to noise or outliers presented in the FCM algorithm. PCM1 relaxed the constraint about the fuzzy membership matrix and the membership grade is replaced with typicality of x_k relative to cluster i. The PFCM algorithm contains both membership and typicality

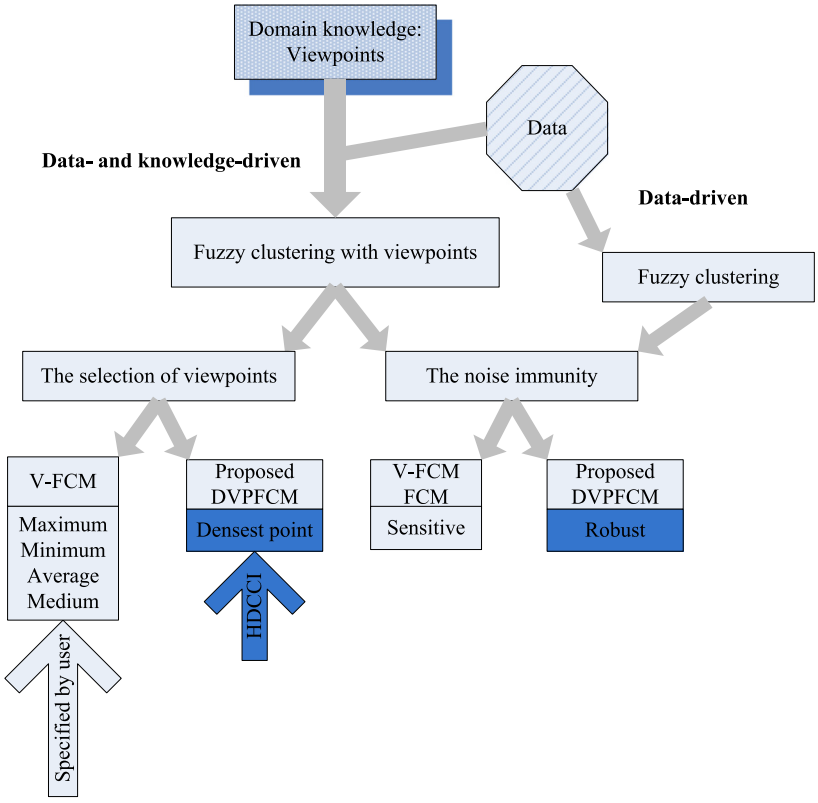


Fig. 1. Overall roadmap of research on clustering methods.

components. Its objective function reads as follows:

$$J_{PFCM}=\sum_{k=1}^{N}\sum_{i=1}^{C}\left(au_{ik}^{m}+bt_{ik}^{p}\right)d_{ik}^{2}+\sum_{i=1}^{C}r_i\sum_{k=1}^{N}(1-t_{ik})^{p} \tag{2}$$

subject to the constrains $\sum_{i=1}^{C}u_{ik}=1\ (k=1,2,\ldots,N)$ and $t_{ik}\le 1$. Here t_{ik} is a typicality component. $m\in(1,+\infty)$ and $p\in(1,+\infty)$ increases fuzziness and overlap among the clusters. $a>0$ and $b>0$ are relative importance of membership grades u_{ik} and typicalities t_{ik}. r_i is a penalty factor, which assumes the following form:

$$r_i=K\frac{\sum_{k=1}^{N}u_{ik}^{m}d_{ik}^{2}}{\sum_{k=1}^{N}u_{ik}^{m}}. \tag{3}$$

r_i is computed using FCM algorithm and the general value of K is 1.

The PFCM algorithm is more robust than the FCM, but it still has the problem of being sensitive to the initialization of clustering centers. The calculation of r_i depends on the FCM algorithm, so its operations are not simple enough.

The PCA algorithm is an improvement over the PCM2 algorithm [24]. As PCA combines the validity indices PE and PC to extend the objective function of the FCM algorithm. It can directly calculate the typicality value without running the FCM algorithm to get the value of r_i. The PCA objective function is:

$$J_{PCA}=\sum_{i=1}^{C}\sum_{k=1}^{N}u_{ik}^{m}d_{ik}^{2}+\frac{\beta}{m^{2}C}\sum_{i=1}^{C}\sum_{k=1}^{N}\left(u_{ik}^{m}\log u_{ik}^{m}-u_{ij}^{m}\right) \tag{4}$$

where m corresponds with the fuzzifier m in FCM. β is the sample co-variance that measures the degree of separation of the data set. In other words, we have

$$\beta=\frac{\sum_{k=1}^{N}\|x_k-\bar{x}\|^{2}}{N}\ \text{with}\ \bar{x}=\frac{\sum_{k=1}^{N}x_k}{N}. \tag{5}$$

PCA solves the problem of dependency of PCM2 on FCM, but it is still sensitive to the initialization of cluster centroids. So PCA sometimes generates coincident clusters.

The GEPFCM algorithm is an improved algorithm based on the ECM (Entropy C-Means) algorithm [25,26] and the PFCM algorithm. It enhances the accuracy of determining clustering centers of noisy data by generalizing ECM combined with PFCM. And the new distance function is used to replace the commonly used Euclidean distance to weaken the effect of noise points on the positioning of clustering centers. Its extended objective function is:

$$\begin{aligned}J_{GEPFCM}=&\sum_{k=1}^{N}\sum_{i=1}^{C}\left(au_{ik}^{m}f_{i,FCM}\left(\|x_k-v_i\|^{2}\right)+bt_{ik}^{p}f_{i,PCM}\left(\|x_k-v_i\|^{2}\right)\right.\\&\left.+cs_{ik}f_{i,E}\left(\|x_k-v_i\|^{2}\right)\right)+\sum_{k=1}^{N}\sum_{i=1}^{C}s_{ij}\ln(s_{ij})\\&+\sum_{i=1}^{C}r_i\sum_{k=1}^{N}(1-t_{ik})^{p}\end{aligned} \tag{6}$$

where a, b, c are coefficients to balance memberships grades u_{ik}, typicalities t_{ik} and entropies s_{ik}. They actually take $a=b=c=1$. Here we also require that $\sum_{i=1}^{C}u_{ik}=1(k=1,\ldots,N)$. $f_{i,FCM}(\|x_k-v_i\|^{2})$, $f_{i,PCM}(\|x_k-v_i\|^{2})$ and $f_{i,E}(\|x_k-v_i\|^{2})$ are distance functions defined in [5], which are calculated as follows:

$$f_{i,E}\left(\|x_k-v_i\|^{2}\right)=c\|x_k-v_i\|^{2}, \tag{7}$$

$$f_{i,FCM}\left(\|x_k-v_i\|^{2}\right)=f_{i,PCM}\left(\|x_k-v_i\|^{2}\right)=1-\exp\left(-\frac{\rho\|x_k-v_i\|^{2}}{R_i^{2}}\right), \tag{8}$$

$$R_i^2 = \frac{\sum_{k=1}^{N} t_{ik}^p \|x_k - v_i\|^2}{\sum_{k=1}^{N} t_{ik}^p}. \quad (9)$$

Here, $\rho = \sum_{k=1}^{N}\sum_{i=1}^{C} s_{ik}\|x_k - v_i\|^2$.

Note that the penalty factor r_i in the objective function (6) depends on PFCM. Its formula is expressed as (3). The initialization of u_{ik}, t_{ik} and v_i are the results of PFCM. So the GEPFCM algorithm not only is sensitive to the initialization of the cluster centroids but also relies on the FCM and PFCM algorithms. Its running process is complicated.

The V-FCM algorithm introduced the concept of "viewpoints" based on the FCM algorithm. A viewpoint is a domain knowledge provided by a user and it involves navigation for a personalized search of the structure. Viewpoints can be typical situations in data, such as average, maximum, minimum, etc. So the V-FCM algorithm makes the clustering process easier with the help of the viewpoints, and the clustering results are more reasonable.

The viewpoints are defined by the two matrixes, denoted as B and F:

$$b_{ij} = \begin{cases} 1, & \text{if the } j\text{th feature of the } i\text{th row of } B \text{ is from the viewpoint} \\ 0, & \text{otherwise.} \end{cases},$$

$$f_{ij} = \begin{cases} y, & b_{ij} = 1 \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

where y is the specific value of the viewpoint.

The V-FCM algorithm explores data structure by minimizing the following objective function:

$$J_{V-FCM} = \sum_{k=1}^{N}\sum_{i=1}^{C}\sum_{j=1}^{l} u_{ik}^m \left\|x_{kj} - g_{ij}\right\|^2 \text{ with } g_{ij} = \begin{cases} v_{ij}, & b_{ij} = 0 \\ f_{ij}, & b_{ij} = 1 \end{cases}. \quad (11)$$

Compared with the FCM algorithm, V-FCM can generate more reasonable clustering results, but there exist problems for noise points and clustering center initialization. Moreover, the types of viewpoints also affect the clustering.

The C-PCM algorithm locates the data points by obtaining the class core of each cluster center from β-cutset, which solves the problem of cluster-center consistency of the PCM1 algorithm. Its objective function is:

$$J_{CPCM} = \sum_{i=1}^{C}\sum_{k=1}^{N} t_{ik}^m d_{ik}^2 + \eta \sum_{i=1}^{C}\sum_{k=1}^{N} (1 - t_{ik})^m, \quad (12)$$

$$t_{ik} = \frac{1}{1 + \left(d_{ik}^2/\eta\right)^{1/(m-1)}}, \quad (13)$$

The (12) is similar to the one of the PCM1 however η assumes a certain predetermined fixed value. It differs from the PCM1 algorithm in that it has further operations on t_{ij} during the iterative update process. After calculating with (13), the typicalities t_{ij} are updated by the following way:

If $t_{qk} = \max_{1\le i\le C} t_{ik} > \beta_i$, then

$$\begin{cases} t_{ik} = 0, & i = 1, \ldots, C \ \& \ i \neq q \\ t_{ik} = t_{ik}, & i = q \end{cases}. \quad (14)$$

If $t_{qk} = \max_{1\le i\le C} t_{ik} < \beta_i$, then

$$t_{ik} = t_{ik}, i = 1, \ldots, C, \quad (15)$$

where $\beta_i = K \cdot \left(1 + \left(\frac{\sum_{k=1}^{N} t_{ik}^m d_{ik}}{\eta^{1/2}\sum_{k=1}^{N} t_{ik}^m}\right)^{-2/(m-1)}\right)$, and K usually takes 0.85.

The RLM algorithm is a fast clustering method based on density of data. It does not iteratively update clustering centers, but directly obtains those according to the density. Then it calculates the distances from each point to each clustering center. The data point belongs to a cluster if it is closer to this cluster. This algorithm is based on the assumption that the clustering center is surrounded by less dense neighborhoods and there is a larger distance from any points with a higher local density. Therefore, two values need to be calculated for any data points, i.e., its local density ρ_k and its minimum distance δ_k to other points with higher local densities.

The local density of a data point is calculated in the form:

$$\rho_k = \sum f(d_{kj} - r), \quad (16)$$

$$f(x) = \begin{cases} 1, & x = d_{kj} - r < 0 \\ 0, & \text{otherwise} \end{cases}. \quad (17)$$

where d_{kj} is the distance between data point x_k and x_j. r is the density radius. The local density ρ_k represents the number of data objects in the radius r range of data point x_k.

If the clustering centers are determined by the local density values, then it is possible to consider noise points as clustering centers. Because with the density radius, noise points also have higher local density values. There is a requirement to calculate another parameter δ_k ($k \in \{1, \ldots, N\}$). Its formula is expressed as follows:

$$\delta_k = \min\{d_{kj} | \rho_j > \rho_k, \ j \in \{1, 2, \ldots, N\}\}. \quad (18)$$

For the point with the largest local density, its $\delta_k = \max\{d_{kj} | k \neq j, \ j \in \{1, \ldots, N\}\}(k \in \{1, \ldots, N\})$. The parameter δ_k describes the minimum distance between point k and other points with higher densities. It can be seen that the true clustering centers have larger values of ρ_k and δ_k. But noise points have small δ because they are isolated and it is often impossible to get close to other higher-density points.

In the RLM algorithm, the value of density radius r is given to calculate the density ρ_k and the minimum distance δ_k to points with higher density values. Then according to the $\rho - \delta$ distribution map, the data points with high density and farther to other points with higher density are selected to be clustering centers. There are some problems in its optimization process:

- Density radius r is just a pre-estimated value, and it affects the values of the density and distance, which will eventually affect the clustering results.
- This method cannot automatically define the clustering centers. It is necessary to observe the $\rho - \delta$ map to determine clustering centers. This result is somewhat subjective and there are often human errors.

2.2. Problems with related algorithms

According to the above introduction to the six algorithms, we can know that FCM, PFCM, PCA, GEPFCM and V-FCM all have the problem of sensitivity to clustering centers initialization. Unsuitable initial centroid values may cause the result to converge to a local optimal value or lead to a slowly clustering process, which has a great negative impact on the clustering results. In addition, the FCM algorithm is sensitive to noise points. The PFCM algorithm alleviates this problem by introducing typicality, but it also brings the problem that the value of the parameter r_i depends on the results of the FCM algorithm. The PCA algorithm eliminates this dependency, but sometimes there is a trouble of consistency of clustering centers [27]. This is a serious problem in fuzzy clustering algorithms [28]. The GEPFCM algorithm has a dependency on the FCM algorithm and the PFCM algorithm. Because the parameter in this algorithm based on the results in the FCM algorithm and the PFCM algorithm to calculate its value. This constraint makes the algorithm running time longer. The V-FCM algorithm simplifies the clustering process by introducing viewpoints. However, there are still problems such as insensitivity to noise points and initialization

Table 1
Comparison of the advantages and disadvantages of each algorithm.

Algorithms	Year	Advantages	Disadvantages
FCM	1981	Classic algorithm, automatically discovering clustering centers	Sensitive to clustering centers initialization, poor noise points immunity
PFCM	2005	High noise immunity, automatically discovering clustering centers.	Sensitive to clustering centers initialization; the parameter r_i depends on the results of FCM.
PCA	2006	High noise immunity, automatically discovering clustering centers, solving the PFCM parameter dependency problem.	Sensitive to clustering centers initialization, clustering centers uniformity problem.
V-FCM	2010	Simplifying the clustering process, faster convergence.	Sensitive to clustering centers initialization, poor noise point immunity.
RLM	2014	Quickly identifying clustering centers.	Density radius r is difficult to determine; it is not easy to obtain clustering centers; there exist human errors.
GEPFCM	2017	High noise immunity, finding clustering centers more accurately.	The parameter r_i depends on the results of PFCM; the algorithm is more complex to run.
C-PCM	2018	Overcoming the consistency of clustering centers	Sensitive to clustering centers initialization

of clustering centers. And the types of viewpoints are related to the accuracy of clustering results. The RLM algorithm is the fastest among these six algorithms, but its density radius is difficult to estimate. The clustering centers need to be obtained by observing the density-distance map, and there is a human error. Each of the six algorithms has advantages and disadvantages. The comparison is summarized in Table 1.

In general, the initialization of clustering centers and the sensitivity of noise points are the main problems of many fuzzy clustering algorithms. Therefore, we focus on solving these two types of problems in this study.

3. The DVPFCM algorithm

In this section, we show the idea of our proposed algorithm (i.e., the DVPFCM algorithm) step by step.

3.1. The hypersphere density-based clustering center initialization (HDCCI) algorithm

As mentioned above, many fuzzy clustering algorithms are sensitive to the initialization of clustering centers, such as FCM, PFCM and so on. For these algorithms, unsuitable initial values may cause the results to converge to an undesirable minimum or make the clustering process converge slowly. In the process of clustering optimization, there is an attraction domain around each local minimum point of the objective function. If the selected center is located in the attraction domain close to the attractor, the optimization process will quickly converge to the pole. Otherwise, the convergence speed will be slow. If the initial center falls outside the attraction domain, the optimization process may converge to other local minimum points. Based on this, it is generally believed that a more suitable clustering center is where the samples are denser.

Therefore, we adopt the improved density function to obtain the higher density points in the samples as the initial clustering centers, which aims to solve the problem that the fuzzy clustering algorithm is sensitive to the initialization of clustering centers.

We can know that the first initial clustering center is in the center of the high density area of the sample. It is the point with highest density in the data set that we are looking for. We regard it as a "viewpoint".

Here we propose the Hypersphere Density-based Clustering Center Initialization (HDCCI) method, and show its details.

At first, we calculate the local density ρ_k of data point x_k and its minimum distance δ_k to other points with higher local densities. These two parameters are calculated as (9) and (11), respectively. Where the density radius r is not predetermined, it is calculated by the following new formula we propose:

$$r = \max\left(d_{kj}\right)/(2C) \tag{19}$$

where d_{kj} is the distance from x_k to x_j and C is the cluster number. We can think of the distribution of entire dataset as a hypersphere (noting that a hypersphere contains all data points). The hypersphere diameter is the maximum distance between points (i.e., $\max(d_{kj})$). Each cluster is a small sphere that contains all the points belonging to this cluster, and is approximately regarded as being composed of C small spheres. So the distance between points in each cluster does not exceed r.

Next, the parameter τ_k $(k = 1, \ldots, N)$ is used to help us easily find the true clustering centers, which is expressed as:

$$\tau_k = \rho_k \times \delta_k. \tag{20}$$

τ_k represents the product of density and distance. Because the clustering center has a larger density and distance, and the noise point has a larger density but a small distance. The larger value of τ_k, the more likely it is that the x_k is a clustering center. In our approach, we use the parameter τ and the density radius r to select clustering centers.

Finally, we calculate the parameter τ_k $(k = 1, \ldots, N)$ and sort it. Then we consider that the distance between the selected clustering center(s) and the clustering center to be selected is greater than r to avoid the selected clustering centers in the same area. So that we can easily obtain high density and distributed clustering centers.

Algorithm 1 Hypersphere density-based clustering centers initialization (HDCCI).

Input: a set of N data points $X = \{x_k\}_{k=1}^{N}$, the desired number of clusters C.
Output: a clustering center matrix $V = \{v_i\}_{i=1}^{C}$ and the viewpoint x_d.
procedure HDCCI (Data X, Number C)
  $V = [\,]$;
  Calculate hypersphere radius r using (19)
  Calculate ρ_k using (9);
  Calculate δ_k using (16);
  Calculate τ_k using (20);
  Sort $\tau = \{\tau_k\}_{k=1}^{N}$ in descending order to get a set τ', and get the data set X' corresponding to the set τ';
  Select the point x'_1 corresponding to τ_1 as the first clustering center v_1 so that $V = V \cup v_1$, and $x_d = v_1$;
  The number of selected clustering centers $tmp = 1$, the next considerable data index $k = 2$;
  repeat
    while $\|x'_k - V\| < r$
      $k = k + 1$;
    $V = V \cup x_k$;
    $tmp = tmp + 1$;
  until $tmp = C$
  return V, x_d;
end procedure

Algorithm 1 shows the whole process of the HDCCI method.

As for the RLM algorithm, its density radius r is predetermined and it is difficult to determine what value to set. In addition, the selection of clustering centers in the RLM algorithm requires drawing density – distance map at first. And then the corresponding points in the data set are observed as the selected clustering centers. This process is troublesome and subjective. It is easy to cause human error. However, in the proposed HDCCI method, we build a new formula (19) to calculate the density radius r, which considers the distribution characteristics of the data set. In this way, different data sets can get a suitable density radius automatically. We do not need human operations, which can avoid the human error.

3.2. The mechanism of the DVPFCM algorithm

Based on the HDCCI method, we can get the densest data point, and use it as the viewpoint. In detail, we select the first clustering center v_1 through HDCCI method, which has a high density and is far from other high density points. The selected one is not likely to be a noise point because the noise point also has high density, but it is isolated so that it is close to the other high density points. The first clustering center v_1 meets the features of the true clustering center. So we can regard it as an integral part of the data structure, which help us discover other parts (other clustering centers) as a viewpoint.

Assume that the first data points selected by HDCCI method is x_d (i.e., v_1). The viewpoint in our algorithm can be denoted as x_d. In [22], the position of the viewpoint in the clustering center matrix is definite, but it is varied in our approach. The row position of the viewpoint in the clustering center matrix is $q = \arg(\min(d_{qd}))$ with $d_{qd} = \|v_q - x_d\|$. We replace the clustering center closest to the viewpoint as the viewpoint, because the closer the distance is, the more similar the two points are, which is in line with the actual experience.

Our goal is to introduce the domain knowledge of the high-density viewpoint to optimize both the FCM and PFCM. As a result, the objective function in the proposed algorithm DVPFCM consists of the three parts: (1) the minimization of the distances between the data and the prototypes; (2) the distribution of data and the viewpoint; (3) the control of t_{ij} to avoid the trivial solution. The objective function of DVPFCM can be expressed as follows:

$$\begin{aligned} J_{DVPFCM} = & \sum_{k=1}^{N}\sum_{i=1,i\neq q}^{C}\left(au_{ik}^{m}+bt_{ik}^{p}\right)\|x_k-v_i\|^2 \\ & + \sum_{k=1}^{N}\left(au_{qk}^{m}+bt_{qk}^{p}\right)\|x_k-x_d\|^2 \\ & + \frac{\sigma^2}{m^2C}\sum_{i=1}^{C}\sum_{k=1}^{N}\left(\varphi_k - t_{ik}\right)^{p} \end{aligned} \tag{21}$$

It is noted that the second term of the objective function is about the degree of disassociation of data points and the viewpoint. So we can introduce matrix $H=[h_i]_{i=1}^{C}$ to rewrite (21) to make it more concise:

$$h_i = \begin{cases} v_i, & i \neq q \\ x_d, & i = q \end{cases}. \tag{22}$$

So (21) can be expressed as:

$$\begin{aligned} J_{DVPFCM} = & \sum_{k=1}^{N}\sum_{i=1}^{C}\left(au_{ik}^{m}+bt_{ik}^{p}\right)\|x_k-h_i\|^2 \\ & + \frac{\sigma^2}{m^2C}\sum_{i=1}^{C}\sum_{k=1}^{N}\left(\varphi_k - t_{ik}\right)^{p}. \end{aligned} \tag{23}$$

Algorithm 2 Density viewpoint-induced possibilistic fuzzy C-means (**DVPFCM**).

Input: A set of N data points $X = \{x_k\}_{k=1}^{N}$, the desired number of clusters C.
Output: A membership matrix $U = \{u_{ik}\}_{i,k=1}^{C,N}$, a clustering center matrix $G = \{g_i\}_{i=1}^{C}$ and typicality matrix $T = \{t_{ik}\}_{i,k=1}^{C,N}$.
procedure DVPFCM(Data X, Number C)
  Run Algorithm 1 to get $(H^{(0)}, x_d) = \text{HDCCI}(X, C)$;
  repeat
    $iter = iter + 1$;
    Update $U^{(iter)} = [u_{ik}]$ by calculating memberships u_{ik} using (35);
    Update $T^{(iter)} = [t_{ik}]$ by calculating typicalities t_{ik} using (37);
    Update $H^{(iter)} = [h_i]$ by calculating centers h_i using (36);
  until $\|H^{(iter)} - H^{(iter-1)}\| < \varepsilon$ **or** $iter > iterMax$;
  return $U^{(iter)}$, $H^{(iter)}$, $T^{(iter)}$;
end procedure

where

$$\sigma^2 = \frac{\sum_{k=1}^{N}\|x_k-\bar{x}\|^2}{N} \text{ with } \bar{x} = \frac{\sum_{k=1}^{N}x_k}{N}, \tag{24}$$

$$\varphi_k = \frac{\sum_{j=1}^{N}\left\|x_j-x_k\right\|^2}{N}\ (k=1,\ldots,N). \tag{25}$$

The parameters m and p have the same meaning as in the PFCM algorithm. The parameter σ^2 is a covariance matrix used to measure the compactness of the data set. The degree of compactness and separation of a data set can be used to make clustering more effective [29, 30]. The φ_k represents the weight of each point, which is its contribution to clustering and measures the impact of each point on the clustering results.

The element $u_{ik} \in [0, 1]$ in the membership matrix U also needs to satisfy the membership normalization condition that the sum of the memberships of one point attributed to each class is 1:

$$\sum_{i=1}^{C}u_{ik} = 1 \quad (k=1,\ldots,N). \tag{26}$$

Taking this constraint into account, we consider the optimization that invokes the Lagrange multipliers so that we subsequently arrived at the unconstrained minimization of J':

$$J' = J_{DVPFCM} + \lambda\left(1-\sum_{i=1}^{C}u_{ik}\right). \tag{27}$$

The necessary conditions that lead to the minimum of (27) are expressed as follows:

$$\frac{\partial J'}{\partial u_{ik}} = 0,\ \frac{\partial J'}{\partial h_i} = 0,\ \frac{\partial J'}{\partial t_{ik}} = 0,\ i = 1,\ldots,C,\ k = 1,\ldots,N. \tag{28}$$

By setting the gradient of J' to zero with respect to u_{ik}, h_i and t_{ik}, we obtain

$$\frac{\partial J'}{\partial u_{ik}} = amu_{ik}^{m-1}\|x_k-h_i\|^2 - \lambda = 0, \tag{29}$$

$$\frac{\partial J'}{\partial h_i} = 2\sum_{k=1}^{N}\left(au_{ik}^{m}+bt_{ik}^{p}\right)(x_k-h_i) = 0, \tag{30}$$

$$\frac{\partial J'}{\partial t_{ik}} = pb\|x_k-h_i\|^2 t_{ik}^{p-1} - \frac{p\sigma^2}{m^2C}(\varphi_k - t_{ik})^{p-1} = 0. \tag{31}$$

From (29), we obtain

$$u_{ik} = \left(\frac{\lambda}{am}\right)^{\frac{1}{m-1}}\left(\frac{1}{\|x_k-h_i\|^{\frac{2}{m-1}}}\right). \tag{32}$$

Substituting (32) into (26), we have

$$\sum_{i=1}^{C} u_{ik} = \sum_{i=1}^{C} \left(\frac{\lambda}{am}\right)^{\frac{1}{m-1}} \left(\frac{1}{\|x_k - h_i\|^{\frac{2}{m-1}}}\right) = 1. \tag{33}$$

It follows that

$$\left(\frac{\lambda}{am}\right)^{\frac{1}{m-1}} = \frac{1}{\sum_{i=1}^{C} \frac{1}{\|x_k - h_i\|^{\frac{2}{m-1}}}}. \tag{34}$$

Substituting (34) into (32), we obtain

$$u_{ik} = \frac{\|x_k - h_i\|^{-\frac{2}{m-1}}}{\sum_{j=1}^{C} \|x_k - h_j\|^{-\frac{2}{m-1}}}. \tag{35}$$

From (30) and (22), we have

$$h_i = \begin{cases} x_d, & i = q \\ \frac{\sum_{k=1}^{N} (au_{ik}^m + bt_{ik}^p) x_k}{\sum_{k=1}^{N} (au_{ik}^m + bt_{ik}^p)}, & i \neq q \end{cases}. \tag{36}$$

This combines the idea of the viewpoint where $q = \arg(\min(d_{qd}))$.

From (31), we obtain

$$t_{ik} = \frac{\varphi_k}{1 + \left(\frac{bCm^2 \|x_k - h_i\|^2}{\sigma^2}\right)^{\frac{1}{p-1}}}. \tag{37}$$

At this point, the derivation processes of the objective function, clustering centers, membership matrix and typicality matrix have been explained.

To sum up, we get the whole mechanism of the DVPFCM algorithm, which is a novel fuzzy clustering algorithm driven by both data and knowledge. Its execution framework is shown in Algorithm 2.

4. Experimental studies

In this section, we demonstrate the performance of the proposed algorithm (DVPFCM) through a series of experiments, and compare the clustering results with seven algorithms, which were introduced in Section 2 (including FCM, PFCM, PCA, RLM, GEPFCM, V-FCM and C-PCM). The method in [23] focuses on the determination of the optimal number of clusters, which is different from the purpose of the algorithms in our study, so it is not compared with our proposed DVPFCM.

The experiment objects are two synthetic datasets, eight UCI machine learning datasets and the Olivetti face database. The synthetic datasets DATA1 and DATA2 are made up of the function make_blob() in python scikit-learn library to generate points with a Gaussian distribution. These two datasets have overlapped classes and are not easily partitioned. The dataset size of DATA2 is larger than that of DATA1, and the number of clusters is larger (DATA1 contains 3 clusters while DATA2 has 7 clusters). The distribution of DATA2 is more complicated. Tested UCI dataset [31] include Iris, Optical Recognition of Handwritten Digits (0,6,8,9), Letter Recognition (A,B), Breast Cancer Wisconsin, SPECT heart data, Zoo, Balance scale and Image segmentation data. These datasets are popular and famous datasets in machine learning field. The Olivetti face database consists of 400 faces of 40 individuals (10 face images per person). We select 200 from 20 people with 1024 attributes, which are shown as Fig. 6.

Table 2 summarizes the basic information of these 10 datasets, including total number of samples, feature number and number of clusters. For all experiments, the parameters in algorithms employ default values. The specific settings are as follows: $m = 2$, $p = 2$, $a = b = 1$, ε=10^{-5}, $iterMax$=150 and $c = 1$ in GEPFCM. For convenience, the r in RLM is calculated by (19).

4.1. Performance indexes

For comparative analyses, we use two major categories of performance indexes, i.e., hard clustering indexes and fuzzy clustering indexes. In the following, we use the superscript "(+)" to indicate that the clustering validity indexes are extremely large indexes, that is, the larger the index values, the better the clustering performance. In contrast, the superscript "(−)" means the index is an extremely small one.

Fuzzy clustering algorithms divides each sample into its class corresponding to the maximum membership according to the membership matrix, so the hard clustering indexes can also be applied to the fuzzy clustering algorithms. However, Campello pointed out in [32] that the casting of fuzzy clustering to hard clustering often fails to faithfully reflect the performance of fuzzy clustering algorithms. Because it ignores the memberships in fuzzy clustering algorithms so that hard clustering indexes are unable to discriminate between overlapped and not overlapped clusters. As such, the hard clustering indexes might not be appropriate for assessment of fuzzy clustering algorithms. To get around these drawbacks, we mainly use fuzzy clustering indexes to evaluate the performances of the fuzzy clustering algorithms. Furthermore, the three hard clustering indexes and two fuzzy clustering indexes are briefly introduced as follows.

Hard clustering indexes employed here are as follows:

(1) Classification rate

The classification rate (CR) is a common measure used to determine how well clustering algorithms perform on the given dataset with a known cluster structure [33]. It is the percentage of patterns that are correctly classified. The closer to 1, the better the performance of the clustering algorithm. Its calculation formula is as follows:

$$CR^{(+)} = \frac{\sum_{i=1}^{C} d_i}{N}. \tag{38}$$

Here d_i is the number of objects correctly identified in the ith cluster, and N is the number of all objects in the dataset.

(2) Normalized mutual information

The normalized mutual information (NMI) [34] is a symmetric measure to quantify the statistical information shared between two cluster distributions. The larger its value, the more similar the two class distributions are and the better the clustering performance. Its formula is:

$$NMI^{(+)}(R, Q) = \frac{\sum_{i=1}^{I} \sum_{j=1}^{J} p(i,j) \log \frac{p(i,j)}{p(i)p(j)}}{\sqrt{H(R)H(Q)}}. \tag{39}$$

Here R, Q are two partitions of the dataset. Assuming R and Q have I and J clusters, respectively. $P(i)$ is the probability that a randomly selected sample from the dataset falls into cluster R_i in partition R. Its formal is $P(i) = \frac{|R_i|}{N}$ where $|R_i|$ is the number of samples in cluster R_i. $p(i, j)$ denotes the probability that an object belongs to cluster R_i in R and cluster Q_j in Q, which is calculated by $p(i, j) = \frac{|R_i \cap Q_j|}{N}$. $H(R)$ is the entropy associated with all probabilities $p(i)$ in partition R. Its calculation is $H(R) = -\sum_{i=1}^{I} P(i) \log P(i)$. The definition of $H(Q)$ is similar to this.

(3) Calinski–Harabasz index.

Table 2
Summary of the datasets used in the experiments.

ID	Name	Instances	Attributes	Classes
D1	DATA1	150	2	3
D2	DATA2	1000	2	7
D3	Iris	150	4	3
D4	Optical Recognition of Handwritten Digits (0,6,8,9)	713	64	4
D5	Breast Cancer Wisconsin	569	30	2
D6	SPECT heart data	267	22	2
D7	Zoo	101	17	7
D8	Balance scale	625	4	3
D9	Letter Recognition (A,B)	1555	16	2
D10	Image segmentation data	2100	19	7

The Calinski–Harabasz (CH) index [35] based on the intra-class distance of the samples and the dispersion matrix between classes is:

$$CH^{(+)} = \frac{Tr(S_B)}{C-1}\Big/\frac{Tr(S_w)}{N-C},\ \text{with}\ Tr(S_B) = \sum_{i=1}^{C} n_i \times d(v_i,\bar{v}),\ Tr(S_w) = \sum_{i=1}^{C}\sum_{k=1}^{N} d(x_k,v_i). \tag{40}$$

Here n_i is the number of samples belonging to ith cluster in the dataset. Grand mean $\bar{v} = \sum_{i=1}^{C} v_i/C$. The larger the CH index, the smaller the intra-class distance and the greater the distance between classes, the better the clustering performance.

Two fuzzy clustering indexes used here are as follows.

(4) The extension index of ARI

The extension index of ARI (EARI) is a fuzzy extension of the adjusted Rand index (ARI) [32,36,37]. This extended index is obtained by first rewriting the formulation of the ARI in a fully equivalent form using basic concepts from set theory. The EARI index can be used to measure the degree of similarity between two clustering results, so the higher the value, the more similar the data structure of the two clusters, the better the clustering performance.

Assuming that R and Q are two hard partitions, there are some definition in ARI index:

a is the number of pairs of data points belonging to the same class in R and to the same cluster in Q;

b is the number of pairs of data points belonging to the same class in R and to different clusters in Q;

c is the number of pairs of data points belonging to different classes in R and to the same cluster in Q;

d is the number of pairs of data points belonging to different classes in R and to different clusters in Q.

According to the basic set theory, given two membership matrices (U_1 and U_2), the quantities a, b, c, and d are redefined as follows when r and q are two soft partitions:

$$a = |V \cap Y| = \sum_{j_2=2}^{N}\sum_{j_1=1}^{j_2-1} t(V(j_1,j_2), Y(j_1,j_2)), \tag{41}$$

$$b = |V \cap Z| = \sum_{j_2=2}^{N}\sum_{j_1=1}^{j_2-1} t(V(j_1,j_2), Z(j_1,j_2)), \tag{42}$$

$$c = |X \cap Y| = \sum_{j_2=2}^{N}\sum_{j_1=1}^{j_2-1} t(X(j_1,j_2), Y(j_1,j_2)), \tag{43}$$

$$d = |X \cap Z| = \sum_{j_2=2}^{N}\sum_{j_1=1}^{j_2-1} t(X(j_1,j_2), Z(j_1,j_2)). \tag{44}$$

Here $V = \{V(j_1,j_2) | V(j_1,j_2) = s_{i=1}^{k} t(r_{ij_1}, r_{ij_2}),\ j_2 = 2,\ldots,N,\ j_1 = 1,\ldots,j_2-1\}$ is the set of pairs of data points belonging to the same class in U_1. $X = \{X(j_1,j_2) | X(j_1,j_2) = s_{i_1,i_2\in[1,k]|i_1\neq i_2} t(r_{i_1j_1}, r_{i_2j_2}),\ j_2 = 2\ldots N,\ j_1 = 1\ldots j_2-1\}$ is the set of pairs of data points belonging to different classes in U_1. $Y = \{Y(j_1,j_2) | Y(j_1,j_2) = s_{l=1}^{v} t(r_{lj_1}, r_{lj_2}),\ j_2 = 2,\ldots,N,\ j_1 = 1,\ldots,j_2-1\}$ is the set of pairs of data points belonging to the same cluster in U_2. Finally $Z = \{Z(j_1,j_2) | Z(j_1,j_2) = s_{l_1,l_2\in[1,v]|l_1\neq l_2} t(r_{l_1j_1}, r_{l_2j_2}),\ j_2 = 2,\ldots,N,\ j_1 = 1,\ldots,j_2-1\}$ is the set of pairs of data points belonging to different clusters in U_2. Here, "t" is a t-norm that is used as a conjunction to implement the connective "and" of the proposition. The authors of [37] and [38] use the "min" operator as a t-norm. Likewise, "s" is a co-norm that is used as a disjunction to implement the connective "or" of proposition. We use the "max" operator as a co-norm.

To sum up, the EARI can be defined as follows:

$$EARI^{(+)} = \frac{a - \frac{(a+b)(a+c)}{a+b+c+d}}{\frac{(a+b)+(a+c)}{2} - \frac{(a+b)(a+c)}{a+b+c+d}}. \tag{45}$$

(5) Xie–Beni index.

The Xie–Beni (XB) index [38] is a popular index in measuring fuzzy clustering performance. It expresses intracluster compactness by the sum of the distances from each sample to the clustering centers, and intercluster separation by the minimum of the distance between all clustering centers. The XB index can be formulated as follows:

$$XB^{(-)} = \frac{\sum_{i=1}^{C}\sum_{k=1}^{N} u_{ik}^{m} d(x_k,v_i)}{N \times \min_{i\neq j} d(v_i,v_j)}. \tag{46}$$

4.2. Testing synthetic datasets

Table 2 shows the basic information of all the used datasets. We first discuss the synthetic datasets DATA1 and DATA2. Fig. 2 shows the point distribution of DATA1 and DATA2. Fig. 3 shows the $\rho-\delta$ distribution and γ distribution of DATA1.

Here we first explain the difference between the algorithm we proposed and the RLM algorithm from the angle of clustering center initialization. Fig. 2(a) shows the spatial distribution characteristics of the DATA1 dataset. The "x" marks are the "bridge points" or noise points that affect the clustering results. From Fig. 3, it can be noticed that the "bridge points" or noise points (in their corresponding $\rho-\delta$ distributions and γ values) are submerged in the $\rho-\delta$ distribution and γ distribution of other data. And in particular, they are completely absent from the γ distribution. So noise or "bridge points" cannot be selected as the clustering centers. The positive "△" indicates the initial clustering center applied to the DVPFCM algorithm obtained by the clustering center initialization method HDCCI. That is, the data point corresponding to the high γ

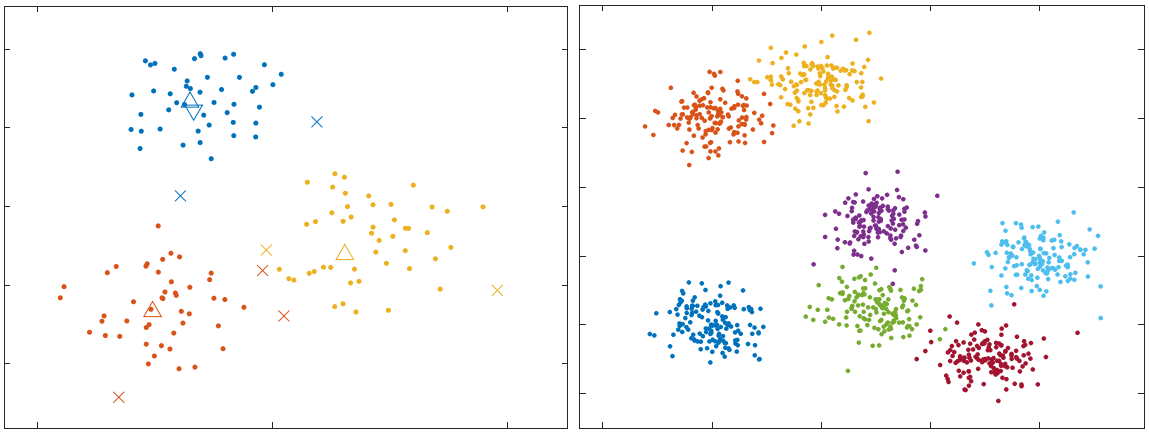


Fig. 2. Synthetic datasets. (a) DATA1. (b) DATA2.

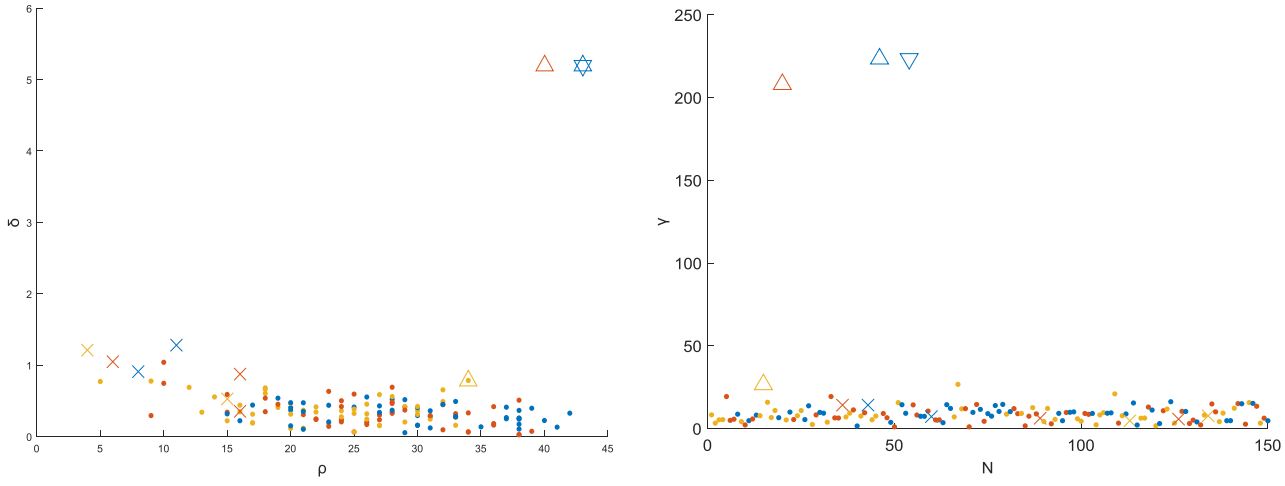


Fig. 3. Results of the algorithms on DATA1. (a) ρ–δ distribution of RLM. (b) γ distribution of HDCCI.

value is selected as the initial clustering center. In Fig. 3(a), there exists the case that a "▽" marked data point coincides with a "△" marked data point. That is to say the densities of these two data points are almost the same.

According to the RLM algorithm, these two points both should be selected as the clustering centers. However, we can know in Fig. 2(a) that the position of these two data are not far apart and that they belong to the same cluster, so we cannot choose both as clustering centers. Then it can be seen in Fig. 3(b) that the γ values of these two points are also equal. Assuming that one point has been selected as the clustering center, the clustering center initialization method HDCCI we proposed does not choose another as the clustering center because it is within r from the selected one. In this way, the initial clustering centers that we ultimately choose are all distributed in high-density areas and are far apart (as shown in Fig. 2(a)), which are in line with the characteristics of the ideal clustering centers and is therefore superior to the RLM algorithm.

For the DATA1 data, we further compare the initial clustering centers obtained by the HDCCI method and the final clustering centers got by the DVPFCM algorithm. The actual clustering centers of DATA1 are [[−6.5,−3.5], [−7.5,−9], [−2.5,−7]]. The initialized clustering centers from HDCCI are [[−6.76,−3.34], [−7.56,−8.68], [−3.45,−7.22]] (identified by "△" in Fig. 2(a)), and the average distance between them and the actual clustering centers is 0.5353. From Fig. 2(a), it can also be seen that the selected clustering center [−3.45,−7.22] is not in the center of the structure, and the clustering center [−6.76,−3.34] is not only at the high density area but also closest to one of the actual clustering centers. So it is appropriate to adopt this point as the viewpoint. After running the DVPFCM algorithm, we get the final clustering centers, which are [−6.76,−3.34], [−7.56,−8.94], [−2.30,−6.79]]. And the average distance from the actual clustering center is 0.2267. It is closer to the actual clustering centers than the value of 0.5353. Consequently it is evident that the DVPFCM algorithm obtains great improvement and better results than the HDCCI method.

For the synthetic dataset DATA2, the distribution characteristics of its data points can be seen from Fig. 2(b). The points labeled with different shapes represent different classes, and the points labeled with the same shape indicate that they belong to the same cluster. The reference clustering center coordinates of the DATA2 are [[8,4],[2,8],[2,2],[4,9],[5,5],[5,2.5],[7,1]]. Bridge points exist in adjacent areas between clusters, which add difficulty to distance-based clustering algorithms.

Table 3
The distance between the clustering centers obtained by FCM and the actual clustering centers.

Algorithm	Centers	Distance	Average distance	Algorithm	Centers	Distance	Average distance
FCM	[8.0,3.9]	0.1000	0.8363	PFCM	[7.7,3.9]	0.3162	0.2407
	[5.0,5.3]	4.0361			[2.2,7.9]	0.2236	
	[2.0,2.0]	0.0000			[2.3,2.2]	0.3606	
	[3.0,8.5]	1.1180			[3.9,8.8]	0.2236	
	[5.0,4.5]	0.5000			[5.0,4.9]	0.1000	
	[5.0,2.5]	0.0000			[5.0,2.6]	0.1000	
	[7.0,1.1]	0.1000			[6.8,1.3]	0.3606	
PCA	[7.9,3.8]	0.2236	0.1982	RLM	[8.0,3.9]	0.2236	0.1898
	[2.2,8.2]	0.2828			[2.1,8.1]	0.1414	
	[2.1,2.0]	0.1000			[2.0,2.0]	0.2236	
	[3.7,8.9]	0.3162			[3.0,8.5]	0.3162	
	[5.0,4.9]	0.1000			[5.0,4.5]	0.2236	
	[5.1,2.6]	0.1414			[5.0,5.3]	0.1000	
	[6.9,1.2]	0.2236			[7.1,1.0]	0.1000	
GEPFCM	[7.9,3.8]	0.2236	0.1765	V-FCM	[8.0,3.9]	0.1000	0.0980
	[2.0,7.9]	0.1000			[2.1,8.1]	0.1414	
	[2.2,2.1]	0.2236			[2.1,2.0]	0.1000	
	[3.8,8.9]	0.2236			[3.9,9.0]	0.1000	
	[5.0,4.9]	0.1000			[5.0,5.0]	0.0000	
	[5.1,2.6]	0.1414			[5.1,2.6]	0.1414	
	[6.9,1.2]	0.2236			[7.1,1.0]	0.1000	
C-PCM	[7.9,3.9]	0.1414	0.0892	DVPFCM	[8.0,3.9]	0.1000	**0.0770**
	[3.0,9.0]	0.1414			[2.0,8.1]	0.1000	
	[2.0,2.0]	0.0000			[1.9,1.9]	0.1414	
	[3.9,9.1]	0.1000			[3.9,9.0]	0.1000	
	[5.0,5.0]	0.0000			[5.0,5.0]	**0.0000**	
	[4.9,2.5]	0.1000			[5.0,2.4]	0.1000	
	[7.0,1.1]	0.1000			[7.1,1.0]	0.1000	

Table 3 lists the clustering centers coordinates obtained by clustering algorithms, the distances and the average distances from the actual clustering centers. By comparing the average distances between the real clustering centers and the clustering centers obtained by each algorithm, we can find that the results obtained by our proposed DVPFCM algorithm are closest to the true value and its error is only 0.077. Meanwhile the second closest is the V-FCM algorithm. However, the distance between the results of FCM and the real results is 0.8363, which is 10 times that of the DVPFCM and V-FCM algorithm. This shows that the "viewpoint" can guide the clustering algorithm to find the real data structure. Look at the clustering centers obtained by the RLM algorithm and there is a certain gap between the actual clustering centers, which shows that (1) the real clustering center is not necessarily a point in the dataset; (2) the RLM algorithm can find the approximate location of the real clustering center. Therefore, the clustering center initialization method based on this algorithm is reliable. The V-FCM, GEPFCM, PCA, PFCM, and FCM algorithms are affected by noise points or the initialization of clustering centers, so the resulting clustering centers are far away from the true clustering centers. Summarizing above, the results obtained by the DVPFCM algorithm are closest to the true values, and the viewpoint plays a key guiding role.

Fig. 4 displays the classification rates for FCM, PFCM, PCA, RLM, GEPFCM, V-FCM and DVPFCM in DATA1 and DATA2. We can know that the correct rate of the proposed DVPFCM algorithm is the highest and it can even reach 1, which means that all points are classified correctly. The second one is the V-FCM algorithm. So by analyzing the experimental results of each algorithm on the synthetic datasets, we can find that the DVPFCM algorithm combines the advantages of the V-FCM algorithm and the PFCM algorithm. Because the typicality weakens the interference of noises or bridge points, the real data structure of the dataset is found under the guidance of the viewpoint. The clustering center initialization method HDCCI finds the approximate locations of the clustering centers before the further iterations of the DVPFCM algorithm, which avoid the algorithm falling into the local optimal value or iteration divergence.

4.3. Testing UCI machine learning datasets

The adopted UCI machine learning datasets are Iris, zoo, balance scale, Optical Recognition of Handwritten Digits (0,6,8,9), Breast Cancer Wisconsin, SPECT heart data, Letter Recognition (A,B), image segmentation data. The Iris dataset is a standard dataset in experimental cases. Fig. 5 shows its Sammon mapping graph. This dataset include three clusters, but there are two classes are not easy to be clustered (labeled by "X" and "*") because they are overlapped each other. The Zoo dataset has more features and clusters than that of Iris, while the Balance scale dataset has a larger number of instances. The Optical Recognition of Handwritten Digits (0,6,8,9) dataset is the recognition of hand-written numbers of 0, 6, 8, and 9, which has the largest number of attributes in all datasets. The Breast Cancer Wisconsin and SPECT heart dataset are medical data in which the types of features are numerical and binary, respectively. The Image segmentation data has the largest number of instances while the Letter Recognition (A, B) is slightly less than it. But the Letter Recognition (A, B) dataset has less clusters than the Image Segmentation dataset.

Table 4 shows the results of the selected machine learning datasets by various clustering algorithms. The performance indexes include the above-mentioned three hard clustering measures (CR, CHI and NMI) and two fuzzy clustering measures (EARI and XB). Because the indexes EARI and XB relate to the specific values of the membership matrix, and the RLM algorithm belongs to the hard clustering, which does not have the membership parameter, so the RLM does not have the EARI and XB indexes values.

As we can see from Table 4, by comparing results of the selected hard clustering measures, the performance relationship of the clustering algorithms in most datasets (except SPECT heart data) is as follows:

$$\text{FCM} < \text{PFCM} < \text{PCA} < \text{RLM} < \text{GEPFCM} < \text{V-FCM} < \text{DVPFCM}.$$

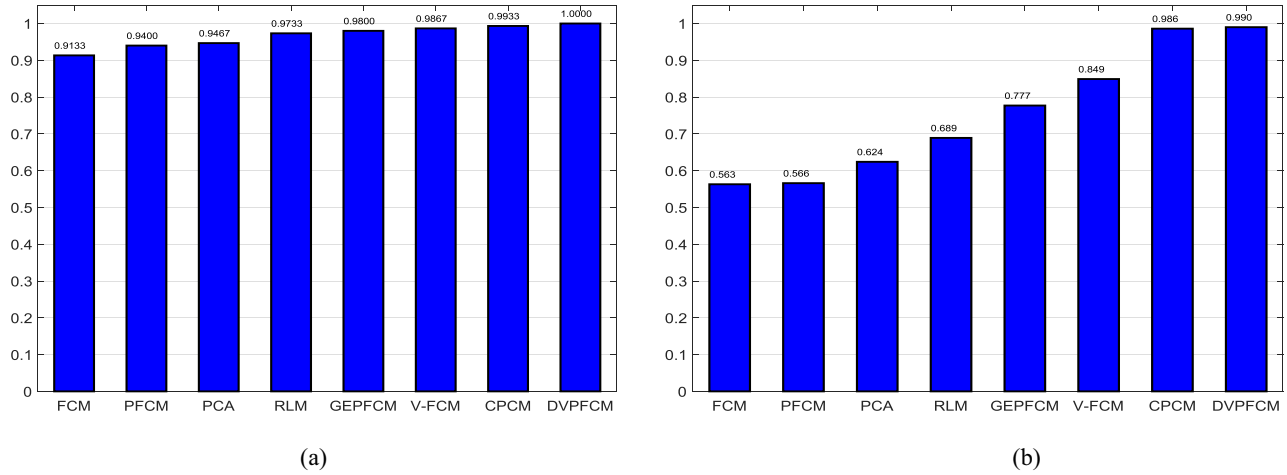


Fig. 4. Comparison of classification rates: (a) DATA1 (b) DATA2.

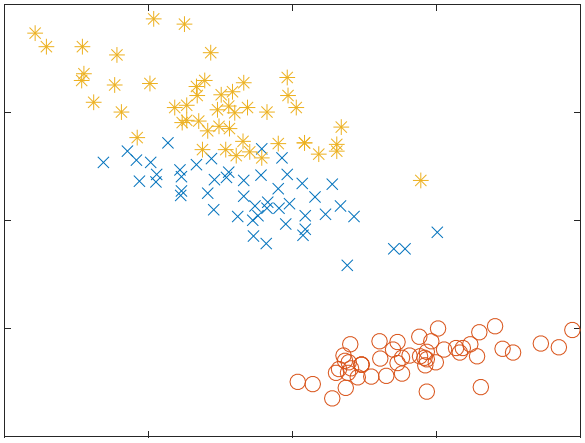


Fig. 5. Iris dataset (Sammon mapping).

Then from the fuzzy clustering validity indexes (EARI and XB), the performance relationship of the clustering algorithms in all dataset is:

FCM < PFCM < PCA < GEPFCM < V-FCM < DVPFCM.

Compared with other algorithms, the proposed DVPFCM algorithm can also guarantee higher classification rate when it gets smaller *XB* values. For the Image dataset, the lowest classification rate is only 0.2719 obtained by other algorithms except DVPFCM, which is much lower than the result of DVPFCM (0.5319). In addition, Table 5 gives the growth ratios of *CR* for the DVPFCM algorithm. Compared with the classical FCM and PFCM algorithm, DVPFCM can improve the correct rate by at least 8.13%, while the maximum is increased by 93.49%. And compared with the C-PCM algorithm whose performance is second only to that of the DVPFCM algorithm on most indexes, the proposed DVPFCM algorithm can improve the accuracy of 7.27% on average, while the maximum can be improved by 38.23%. It can be seen that the clustering results obtained by our proposed algorithm have

Fig. 6. The Olivetti face database.

Table 4
Clustering results on UCI datasets.

Datasets	Algorithms	FCM	PFCM	PCA	RLM	GEPFCM	V-FCM	C-PCM	DVPFCM
Iris	$CR^{(+)}$	0.5867	0.6000	0.6600	0.8067	0.8200	0.8667	**0.8933**	**0.8933**
	$CH^{(+)}$	9.6191	9.6269	9.7422	10.9058	11.9973	12.1314	12.1314	**12.2276**
	$NMI^{(+)}$	0.2585	0.3287	0.4999	0.6821	0.7031	0.7274	0.7431	**0.7688**
	$EARI^{(+)}$	0.7135	0.7371	0.7914	–	0.9042	0.9822	0.9926	**0.9943**
	$XB^{(-)}$	0.7407	0.6801	0.5021	–	0.4533	0.3743	0.3629	**0.2238**
Optical Recognition of Handwritten Digits (0,6,8,9)	$CR^{(+)}$	0.5386	0.5582	0.5750	0.8668	0.8773	0.9158	0.9285	**0.9369**
	$CH^{(+)}$	18.4032	21.2999	23.5065	24.0312	24.2069	25.3173	25.3843	**28.0189**
	$NMI^{(+)}$	0.4335	0.5447	0.6360	0.6905	0.7362	0.7742	0.8174	**0.8309**
	$EARI^{(+)}$	0.5790	0.7202	0.8490	–	0.8637	0.9350	0.9500	**0.9659**
	$XB^{(-)}$	1.7151	1.7057	1.5858	–	1.5094	0.9486	0.7749	**0.2643**
Breast Cancer Wisconsin	$CR^{(+)}$	0.7856	0.7873	0.8366	0.8541	0.8893	0.9051	0.9051	**0.9192**
	$CH^{(+)}$	40.89951	53.6177	63.263	66.2016	81.966	83.6849	84.254	**92.7073**
	$NMI^{(+)}$	0.2908	0.3067	0.3328	0.4672	0.5286	0.5336	0.5490	**0.5814**
	$EARI^{(+)}$	0.8351	0.8957	0.9316	–	0.9794	0.9823	0.9962	**0.9999**
	$XB^{(-)}$	0.9728	0.8796	0.7203	–	0.4723	0.3949	0.3746	**0.3713**
SPECT heart data	$CR^{(+)}$	0.5655	0.6742	0.7528	0.3558	0.7940	0.8165	0.7940	**0.8502**
	$CH^{(+)}$	16.5447	19.8554	25.3664	18.8236	29.2436	32.9398	30.0071	**33.3627**
	$NMI^{(+)}$	0.1128	0.1380	0.1818	0.0266	0.1907	0.2119	0.1748	**0.2142**
	$EARI^{(+)}$	0.2636	0.3394	0.5175	–	0.5760	0.6498	0.5522	**0.8935**
	$XB^{(-)}$	6.5483	4.0402	2.678	–	2.2130	0.7602	0.7500	**0.7592**
Zoo	$CR^{(+)}$	0.4257	0.5050	0.5248	0.5545	0.5743	0.6040	0.6238	**0.6634**
	$CH^{(+)}$	1.0072	1.1213	1.1353	1.1512	1.1878	1.2508	1.2423	**1.264**
	$NMI^{(+)}$	0.4172	0.4466	0.5205	0.5748	0.5929	0.7247	0.7429	**0.7695**
	$EARI^{(+)}$	0.8633	0.8984	0.9327	–	0.9656	0.9700	**0.9772**	**0.9772**
	$XB^{(-)}$	0.7063	0.6517	0.5005	–	0.4265	0.3642	0.2294	**0.2119**
Balance scale	$CR^{(+)}$	0.5376	0.5520	0.6176	0.7360	0.7456	0.7504	0.7712	**0.8000**
	$CH^{(+)}$	24.6063	25.1068	25.9596	28.1003	32.9543	34.1714	34.7239	**36.7628**
	$NMI^{(+)}$	0.1548	0.1747	0.2252	0.2492	0.3469	0.3595	0.3827	**0.4773**
	$EARI^{(+)}$	0.2819	0.3465	0.6541	–	0.7295	0.7316	0.8577	**0.8577**
	$XB^{(-)}$	0.9973	0.8461	0.7000	–	0.4354	0.2827	0.1101	**0.1101**
Letter Recognition (A,B)	$CR^{(+)}$	0.8476	0.8695	0.8965	0.9029	0.9273	0.9370	0.9370	**0.9402**
	$CH^{(+)}$	128.5351	142.8131	180.0137	218.1078	224.3567	242.0209	250.0544	**252.3733**
	$NMI^{(+)}$	0.3842	0.4424	0.5229	0.5282	0.6544	0.7109	0.7109	**0.7236**
	$EARI^{(+)}$	0.8187	0.8601	0.8823	–	0.9229	0.9445	0.9600	**0.9781**
	$XB^{(-)}$	1.5822	0.9043	0.793	–	0.5906	0.5259	0.5042	**0.4831**
Image segmentation data	$CR^{(+)}$	0.2749	0.3105	0.3338	0.3538	0.4200	0.4816	0.3848	**0.5319**
	$CH^{(+)}$	21.3247	22.5122	23.5307	24.4764	26.6791	28.8336	28.2917	**30.7711**
	$NMI^{(+)}$	0.3175	0.4339	0.4994	0.6256	0.6273	0.6395	0.5077	**0.6911**
	$EARI^{(+)}$	0.5373	0.6391	0.6689	–	0.6706	0.7329	0.8280	**0.8617**
	$XB^{(-)}$	1.7168	0.9718	0.8905	–	0.5971	0.5944	0.4979	**0.3151**

Table 5
The CR growth ratios of the DVPFCM on UCI datasets (%).

	FCM	PFCM	PCA	RLM	GEPFCM	V-FCM	C-PCM
D3	52.26	48.88	35.34	10.74	8.94	3.07	0
D4	73.95	67.84	**62.94**	8.09	6.79	2.30	0.91
D5	17.01	16.75	9.87	7.62	3.36	1.56	1.56
D6	50.34	26.11	12.94	**138.95**	7.08	4.13	7.08
D7	55.84	31.16	26.41	19.64	15.51	9.83	6.35
D8	48.81	44.93	29.53	8.70	7.30	6.61	3.73
D9	10.93	8.13	4.87	4.13	1.39	0.34	0.34
D10	**93.49**	**71.30**	59.35	50.34	**26.64**	**10.44**	**38.23**
Average	50.33	39.41	30.16	31.03	9.63	4.79	7.27

higher reliability. In terms of other evaluation indexes, the proposed DVPFCM algorithm seems always slightly better, which indicates that DVPFCM may be more suitable for the actual machine learning datasets and outperforms other techniques when the datasets have larger numbers or more features.

Furthermore, one thing to note is that for the SPECT heart data, the results obtained by the RLM algorithm and the C-PCM algorithm are not ideal, so RLM and C-PCM may not be suitable for binary dataset. For the image dataset, C-PCM does not perform better than GEPFCM and V-FCM in hard cluster validity indexes (CR, CHI, NMI). But for other datasets, CPCM is superior to other algorithms except our proposed DVPFCM in all cluster validity indexes. It may show the dataset with a large dimension and a large number of features is a great challenge to the CPCM algorithm.

Overall, DVPFCM is the most outstanding in terms of obtaining initial clustering centers, the distances between clustering results and reference clustering centers, and various performance indexes.

4.4. Testing the Olivetti face database

The 400 different images of the Olivetti face database were taken at different times, under different lighting conditions, under different expressions of the same person (open or closed eyes, smile or not smile). Our experiment uses photos of the first 20 persons. We implement DVPFCM for the 200 face images and gets its CR of 0.8950, while C-PCM, V-FCM, GEPFCM, RLM, PCA, PFCM and FCM have 0.7200, 0.8450, 0.6750, 0.6350, 0.5000, 0.4450, 0.5000 of CR, respectively. Those experimental results show that our DVPFCM algorithm has better performance in face recognition.

Moreover, all the experiments are based on knowing the clusters number C of dataset. But in reality, this priori knowledge is often unknown. Therefore, with the help of some clustering validity indexes, we test the accuracy of DVPFCM, C-PCM, V-FCM, GEPFCM, PFCM and FCM in obtaining the optimal number of clusters. Those indexes include: normalized partition entropy (NPE) [39], DB index proposed in [40], XB index [38] and Calinski–Harabasz index (CH) [35], which are all commonly used clustering validity indexes. Now, assuming that we do not know the C of the Olivetti face database, we take the number of clusters corresponding to the extreme val-

Table 6
Optimal cluster numbers got from validity indexes for UCI datasets under different clustering algorithms.

dataset	True C	Index	FCM	PFCM	PCA	GEPFCM	VFCM	CPCM	DVPFCM
Iris	3	CH	2	2	2	2	2	2	2
		DB	2	2	2	2	2	2	2
		NPE	2	2	2	2	2	2	2
		XB	2	2	2	2	2	2	2
Optical Recognition of Handwritten Digits (0,6,8,9)	4	CH	3	3	3	**4**	**4**	3	3
		DB	3	11	18	3	**4**	4	3
		NPE	5	3	2	2	3	**4**	3
		XB	**4**	3	**4**	**4**	3	2	**4**
Breast Cancer Wisconsin	2	CH	**2**	**2**	**2**	4	**2**	**2**	**2**
		DB	**2**	**2**	**2**	*8*	**2**	8	**2**
		NPE	**2**	**2**	**2**	**2**	**2**	**2**	**2**
		XB	**2**	**2**	**2**	**2**	**2**	**2**	**2**
SPECT heart data	2	CH	**2**	3	**2**	**2**	**2**	**2**	**2**
		DB	7	**2**	**2**	**2**	**2**	7	**2**
		NPE	**2**	**2**	**2**	**2**	**2**	**2**	**2**
		XB	7	**2**	**2**	**2**	**2**	3	**2**
Zoo	7	CH	**7**	8	**7**	**7**	**7**	**7**	**7**
		DB	**7**	**7**	**7**	8	**7**	8	8
		NPE	8	8	9	**7**	**7**	**7**	**7**
		XB	**7**	**7**	6	**7**	5	6	**7**
Balance scale	3	CH	4	**3**	**3**	3	**3**	**3**	**3**
		DB	**3**	**3**	11	2	**3**	6	**3**
		NPE	4	2	2	2	2	**3**	**3**
		XB	**3**	**3**	2	2	2	4	**3**
Letter Recognition (A,B)	2	CH	**2**	**2**	**2**	**2**	**2**	**2**	**2**
		DB	3	6	**2**	**2**	3	**2**	**2**
		NPE	**2**	**2**	**2**	3	**2**	**2**	**2**
		XBI	**2**	**2**	**2**	**2**	**2**	**2**	**2**
Image segmentation data	7	CH	4	4	4	4	**7**	**7**	**7**
		DB	4	4	**7**	4	4	**7**	4
		NPE	**7**	4	**7**	4	4	4	4
		XB	4	4	**7**	**7**	4	4	**7**

ues of these cluster validity indexes as the ideal number of clusters. Fig. 7 shows the optimal number of clusters obtained by our algorithm DVPFCM based on these indexes values. Note that in order to more clearly represent the magnitude of the change in the value of the cluster validity indexes under different clusters, the NPE takes its log value and the XB is inverted. We observe that the results of CH, DB and NPE indexes show that the ideal cluster number of the Olivetti face database is 20 while the result got by XB^{-1} is 19. So the DVPFCM algorithm can recognize how many faces are in the Olivetti face database. In addition, Table 6 gives the optimal cluster numbers estimated by these cluster validity indexes for the UCI datasets. From Table 6, we can see that compared to other fuzzy clustering algorithms (FCM, PFCM, PCA, GEPFCM, V-FCM, CPCM), the DVPFCM algorithm seems to be able to better discover the true cluster number C.

4.5. Computational complexity

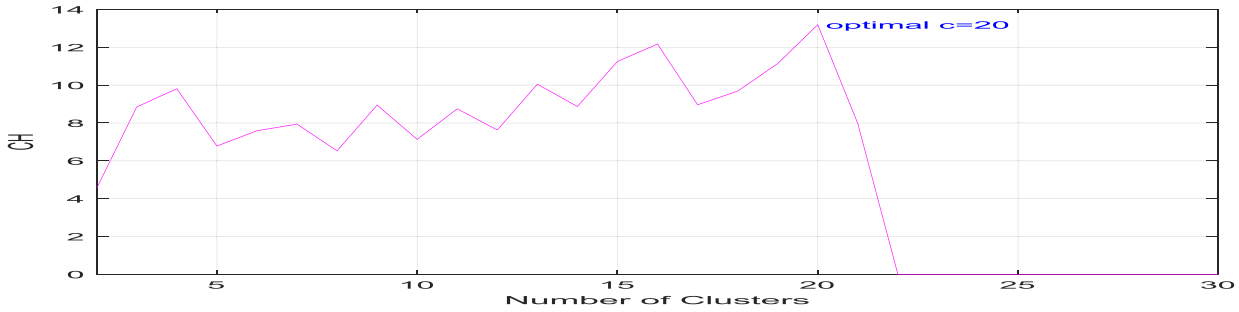
Next, we analyze the computational complexities for the DVPFCM, C-PCM, V-FCM, RLM, GEPFCM, and FCM algorithms. The DVPFCM algorithm can be divided into four parts: (1) we run HDCCI algorithm to obtain initial cluster centers and the high-density viewpoint, which needs $O(Nl)$; (2) we iteratively update membership degree matrix U, which needs $O(iter(NC^2l))$; (3) we iteratively update typicality matrix T, which needs $O(iter(NCl))$; (4) we iteratively update cluster center matrix V, which needs $O(iter(NC))$. Since the big O notation only describes the upper bound on the growth rate of the function, the total computational complexity of the DVPFCM algorithm is $O(iter(NC^2l))$. The computational complexities of other seven comparison algorithms are shown in the Table 7. We can find that the DVPFCM algorithm adds the domain knowledge of "high-density viewpoint", but it does not increase the time complexity of the algorithm, which is equal to the computational complexity of FCM. Compared with the two possibilistic fuzzy clustering algorithms, PFCM and GEPFCM, the computational complexity of our DVPFCM algorithm is greatly reduced. The reason is that we optimize the parameter γ_i of the two algorithms to parameter σ^2. γ_i is calculated by substituting the cluster centers and membership matrix got by FCM and need to be updated iteratively, while the σ^2 considers the data points distribution characteristics so that it can be calculated before iteration, which greatly saves the running time of the DVPFCM algorithm.

Table 7
Computational complexities of various clustering algorithms.

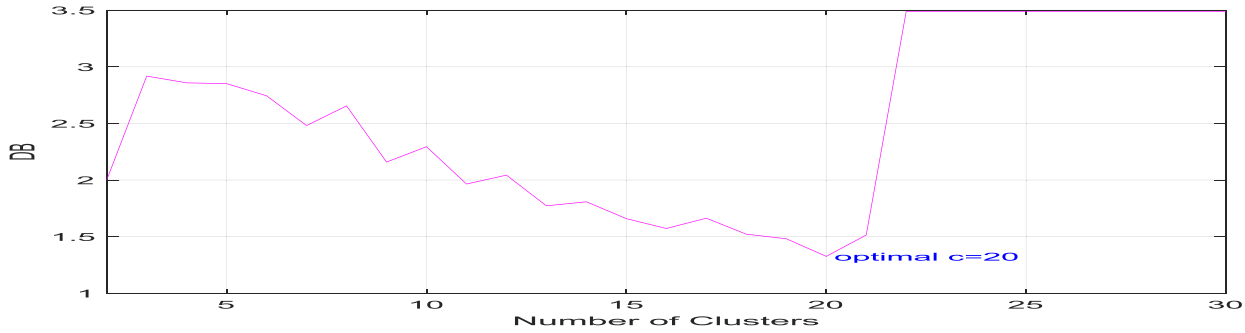
Algorithm	Computational complexity
FCM	$O(iterNC^2l)$
PFCM	$O(iter(N^2Cl + NC^2l))$
PCA	$O(iterNCl)$
GEPFCM	$O(iter(N^2Cl + NC^2l))$
RLM	$O(Nl)$
V-FCM	$O(iterNC^2l)$
CPCM	$O(iterNCl)$
DVPFCM	$O(iterNC^2l)$

Tables 8 and 9 display the average number of iterations and the average calculation time for each run of various algorithms. We performed 50 runs for every algorithm except for the DVPFCM algorithm. The results of the DVPFCM algorithm are certain because the initial clustering centers of each run are the same obtained by the HDCCI method. The number in parentheses indicates the ranking of the algorithm and *Arank* denotes the average ranking of the algorithm. Note that the RLM algorithm does not result in iterations, so statistics for it are not demanded here.

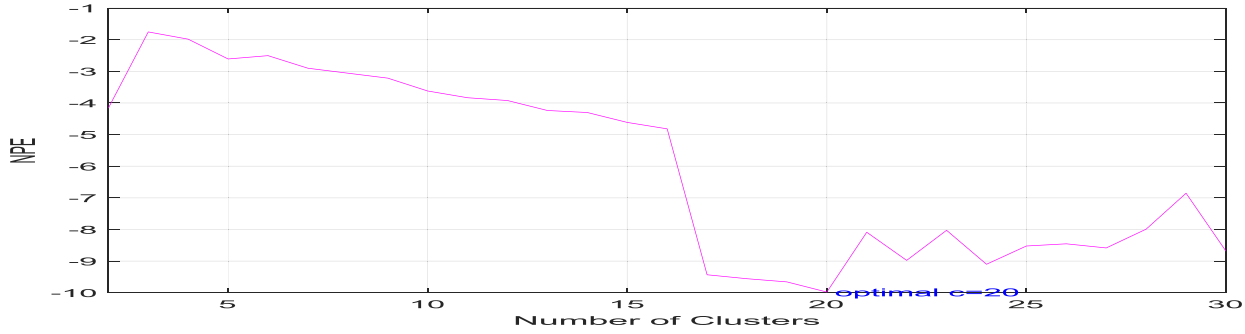
It can be seen from these two tables that the DVPFCM algorithm requires the least iteration times on most datasets except the GEPFCM algorithm and the V-FCM algorithm is second to it. This means that with the help of a viewpoint and the clustering



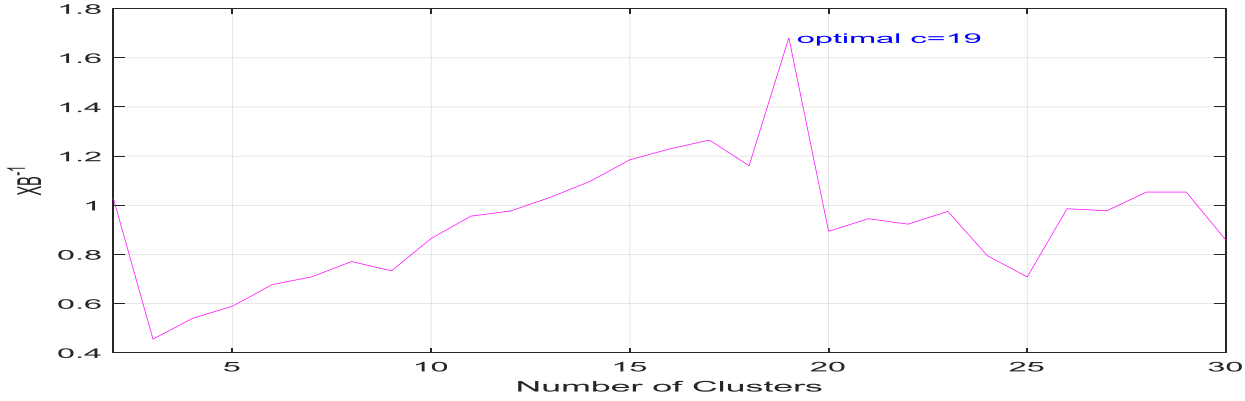
(a) The CH index



(b) The DB index



(c) The NPE index



(d) The XB^{-1} index

Fig. 7. Four cluster validity indexes for the Olivetti face database using DVPFCM.

Table 8
Average number of iterations for tested algorithms.

	FCM	PFCM	PCA	GEPFCM	V-FCM	CPCM	DVPFCM
D1	15.3(6)	17.0(7)	13.9(5)	3.1(1)	12.9(4)	10.2(3)	8.0(2)
D2	24.2(4)	30.3(6)	35.0(7)	4.3(1)	24.2(4)	8.0(3)	4.0(2)
D3	15.9(4)	26.2(6)	43.0(7)	3.0(1)	20.5(5)	15.0(3)	12.0(2)
D4	27.9(6)	6.7(3)	78.0(7)	2.0(1)	8.2(4)	19.3(5)	6.0(2)
D5	19.7(5)	43.8(7)	22.0(6)	2.0(1)	13.0(4)	6.3(2)	10.0(3)
D6	79.9(7)	17.8(5)	24.0(6)	2.0(1)	8.6(2)	9.2(4)	9.0(3)
D7	67.6(7)	15.0(4)	24.0(5)	2.7(1)	31.0(6)	10.6(3)	9.0(2)
D8	99.2(7)	12.6(3)	27.0(5)	2.0(1)	81.0(6)	22.2(4)	11.0(2)
D9	26.4(7)	18.9(5)	23.0(6)	2.0(1)	11.5(3)	12.0(4)	10.0(2)
D10	78.4(6)	16.0(2)	83.0(7)	4.5(1)	48.0(4)	50.5(5)	31.0(3)
Arank	5.9	4.8	6.1	1	4.2	3.6	2.3

Table 9
Average time of iterations for tested algorithms.

	FCM	PFCM	PCA	RLM	GEPFCM	V-FCM	CPCM	DVPFCM
D1	0.0030(2)	0.0310(7)	0.0045(4)	0.0190(5)	0.1314(8)	0.0017(1)	0.0038(3)	0.0250(6)
D2	0.0367(3)	0.0727(4)	0.0132(1)	0.0780(5)	1.1946(8)	0.5900(7)	0.0249(2)	0.1810(6)
D3	0.0034(3)	0.0079(5)	0.0025(2)	0.0170(6)	0.0911(8)	0.0027(1)	0.0078(4)	0.0222(7)
D4	0.1269(4)	0.1330(5)	0.0628(2)	0.1640(6)	0.5402(7)	0.0086(1)	0.0650(3)	0.8170(8)
D5	0.0382(4)	0.0751(6)	0.0066(2)	0.0730(5)	0.2307(7)	0.0043(1)	0.0175(3)	0.5313(8)
D6	0.0642(5)	0.0685(6)	0.0032(2)	0.0240(4)	0.1447(8)	0.0018(1)	0.0095(3)	0.0725(7)
D7	0.0192(5)	0.0229(6)	0.0029(1)	0.0140(3)	0.1573(8)	0.0150(4)	0.0050(2)	0.0472(7)
D8	0.0951(5)	0.1005(6)	0.0048(1)	0.0370(2)	0.2252(8)	0.0400(3)	0.0469(4)	0.1950(7)
D9	0.0796(4)	0.1520(5)	0.0105(2)	0.3060(6)	0.4117(7)	0.0063(1)	0.0593(3)	2.4066(8)
D10	0.4289(4)	0.4782(5)	0.1197(1)	0.5840(6)	1.8114(7)	0.1780(2)	0.4081(3)	14.0320(8)
Arank	3.9	5.5	1.8	4.8	7.6	2.2	3.0	7.2

center initialization method, the DVPFCM algorithm is faster than other algorithms. The GEPFCM algorithm has the lowest iteration times, but on average, its running time cost is the highest, because it depends on the FCM and PFCM algorithms.

In terms of running time, the DVPFC algorithm runs the clustering center initialization method HDCCI to get the initial clustering centers, and then find more accurate data structure through the guidance of the viewpoint, so it also requires more running time (on average, second to the GEPFCM algorithm). However, it can obtain better clustering results than other algorithms and has more outstanding clustering performance. In this respect, these costs are worthwhile.

4.6. Discussion

In this section, we verify the accuracy and convenience of the clustering center initialization method HDCCI. The results show that this novel clustering center initialization algorithm can accurately find high-density data points as the initial clustering centers. Moreover, this method not only calculates the density of the data point, but also compute the point distance to other higher density points, so it can effectively avoid the interference of noise points (because noise point may have higher density but the distance value is definitely small). The results of the selection are obtained by directly calculating the relevant parameters by the program, and without human participation. This process is simple and convenient. This is also superior to RLM.

Then, we tested the performance of our proposed DVPFCM algorithm and other algorithms on UCI datasets and synthetic datasets. By comparison, the DVPFCM algorithm did obtain higher classification rates both in the synthetic datasets and UCI datasets. In other indexes, the DVPFCM algorithm performed better than the FCM, PFCM, PCA, RLM, GEPFCM and V-FCM algorithms.

Finally, we compared the average number of iterations for various algorithms and found that the DVPFCM algorithm requires fewer iterations than the traditional clustering algorithms, which shows that the DVPFCM algorithm converges faster. In general, the DVPFCM algorithm can provide more accurate clustering results and is more robust.

The better performance of the DVPFCM algorithm can be attributed to the following points:

- The proposed HDCCI clustering center initialization method can get the initial clustering centers close to the real data structures. This not only prevents the algorithm from iterating divergence and speeds up the algorithm convergence, but also helps to improve the classification rate of the algorithm.
- As a part of the data structure, the viewpoint helps to guide the clustering algorithm to find the real data structure, so as to obtain better clustering results.
- The objective function of the DVPFCM algorithm introduces typicality in the possibilistic clustering algorithm and its parameter σ^2 considers the degree of compactness and separation of the dataset, which makes the algorithm more robust against noise points.

The proposed DVPFCM algorithm and the V-FCM algorithm have the following distinct differences:

- The purposes of clustering are different. As mentioned above, the V-FCM algorithm focuses on personalized search, and the DVPFCM algorithm is to get more accurate clustering centers with the help of the "viewpoint" to improve the reliability of the algorithm.
- The categories of viewpoint are not the same. The V-FCM algorithm uses the average or boundary value that is of interest to the user to represent the viewpoint, where the viewpoint needs to be specified in advance. The DVPFCM algorithm adopts the point with the highest density as the viewpoint which is based on clearer selection. And there is no need to prior design the viewpoint.
- The noise immunity of the algorithm is different. The V-FCM algorithm integrates the viewpoint into the FCM algorithm. But the FCM algorithm is sensitive to noise points. The DVPFCM algorithm combines the viewpoint with the

PFCM algorithm, and optimizes some parameters related to the tightness of the data distribution. The robustness of the DVPFCM algorithm is significantly higher than that of V-FCM.

5. Conclusions

In this study, we have proposed a new density-based cluster center initialization method-HDCCI at first. The initial cluster centers obtained by this method are located at the center of the sample-intensive region, which can make the initial cluster centers close to the real data structure. Thus we solve the problem that most fuzzy clustering algorithms are sensitive to cluster center initialization. Then, we introduce the "high-density viewpoint" obtained by the HDCCI method to assist the clustering algorithm so as to establish the DVPFCM algorithm. It is directly used as a part of the prototypes to guide the algorithm to find other classes faster. By comparing the iteration times of each algorithm, we find that our DVPFCM algorithm has fewer iterations than FCM, PFCM, PCA, V-FCM and C-PCM algorithms, while the V-FCM algorithm has the least number of iterations compared with the first three algorithms. This result fully demonstrates that the "viewpoint" helps speed up the convergence of the algorithm.

Applying our proposed DVPFCM algorithm and comparison algorithms (FCM, PFCM, PCA, RLM, GEPFCM, V-FCM, C-PCM) to the synthetic datasets and the real UCI datasets, experimental results show that DVPFCM poses better performance in the initialization of cluster centers, the distance between the clustering results and the reference cluster centers, and various evaluation indexes. In summary, the DVPFCM algorithm can provide more ideal clustering results and is more robust. Moreover, when these algorithms are applied to the Olivetti face database, we find that the accuracy of our DVPFCM algorithm in face recognition is much higher than the accuracy offered by other algorithms, which indicates that the proposed DVPFCM algorithm exhibits tangible practical significance.

In future studies, it is worth applying the DVPFCM algorithm to other areas, such as image segmentation, mechanical fault detection, and so on. Moreover, the kernel-based fuzzy clustering algorithms can also be introduced viewpoints to enhance the performance of the clustering algorithm, which is also one of important pursuits in the future.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Nos. 61673156, 61432004, U1613217 and 61672202), the Natural Science Foundation of Anhui Province (Nos.1408085MKL15 and 1508085QF129), and the China Postdoctoral Science Foundation (Nos. 2012M521218 and 2014T70585).

References

[1] C. Otto, D. Wang, A.K. Jain, Clustering millions of faces by identity, IEEE Trans. Pattern Anal. Mach. Intell. 40 (2) (2018) 289–303.
[2] K. Honda, H. Ichihashi, Linear fuzzy clustering techniques with missing values and their application to local principal component analysis, IEEE Trans. Fuzzy Syst. 12 (2) (2004) 183–193.
[3] J.Z. Huang, M.K. Ng, H. Rong, et al., Automated variable weighting in K-means type clustering, IEEE Trans. Pattern Anal. Mach. Intell. 27 (5) (2005) 657–668.
[4] F.L. Huang, X.L. Li, S.C. Zhang, et al., Harmonious genetic clustering, IEEE Trans. Cybern. 48 (2018) 199–214.
[5] W. Pedrycz, A. Amato, V.D. Lecce, et al., Fuzzy clustering with partial supervision in organization and classification of digital images, IEEE Trans. Fuzzy Syst. 16 (4) (2008) 1008–1026.
[6] W. Huang, S.K. Oh, W. Pedrycz, Fuzzy wavelet polynomial neural networks: analysis and design, IEEE Trans. Fuzzy Syst. 25 (5) (2017) 1329–1341.
[7] Y.M. Tang, X.P. Liu, NRDSP: no-reference assessment of SAR image despeckling, Neurocomputing 249 (2) (2017) 225–236.
[8] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (6191) (2014) 1492–1496.
[9] J.C. Bezdek, Pattern Recognition With Fuzzy Objective Function Algorithms, Academic, New York, 1981.
[10] R. Krishnapuram, J. Keller, A possibilistic approach to clustering, IEEE Trans. Fuzzy Syst. 1 (2) (1993) 98–110.
[11] F.C. Rhee, Uncertain fuzzy clustering: insights and recommendations, IEEE Comput. Intell. Mag. 2 (2007) 44–56.
[12] J.S. Zhang, Y.W. Leung, Improved possibilistic C-means clustering algorithms, IEEE Trans. Fuzzy Syst. 12 (2004) 209–217.
[13] M.S. Yang, K.L. Wu, Unsupervised possibilistic clustering, Pattern Recognit. 39 (1) (2006) 5–21.
[14] H. Yu, J. Fan, Cutset-type possibilistic C-means clustering algorithm, Appl. Soft Comput. 64 (2018) 401–422.
[15] N.R. Pal, K. Pal, J.C. Bezdek, A possibilistic fuzzy C-means clustering algorithm, IEEE Trans. Fuzzy Syst. 13 (4) (2005) 517–530.
[16] D.S. Yeung, X.Z. Wang, Improving performance of similarity-based clustering by feature weight learning, IEEE Trans. Pattern Anal. Mach. Intell. 24 (4) (2002) 556–561.
[17] S.C. Chen, D.Q. Zhang, Robust image segmentation using FCM with spatial constrains based on new kernel-induced distance measure, IEEE Trans. Syst. Man Cybern. B 34 (4) (2004) 1907–1916.
[18] L. Chen, C.L.P. Chen, M. Lu, A multiple-kernel fuzzy C-means algorithm for image segmentation, IEEE Trans. Syst. Man Cybern. B 41 (5) (2011) 1263–1274.
[19] Y. Ding, X. Fu, Kernel-based fuzzy C-means clustering algorithm based on genetic algorithm, Neurocomputing 188 (2016) 233–238.
[20] S. Askari, N. Montazerin, M.H.F. Zarandi, et al., Generalized entropy based possibilistic fuzzy C-means for clustering noisy data and its convergence proof, Neurocomputing 219 (2017) 186–201.
[21] X. Zhu, W. Pedrycz, Z. Li, Granular data description: designing ellipsoidal information granules, IEEE Trans. Cybern. 47 (12) (2017) 4475–4484.
[22] W. Pedrycz, V. Loia, S. Senatore, Fuzzy clustering with viewpoints, IEEE Trans. Fuzzy Syst. 18 (2010) 274–284.
[23] P. Fazendeiro, J. Valente De Oliveira, Observer-biased fuzzy clustering, IEEE Trans. Fuzzy Syst. 23 (1) (2015) 85–97.
[24] R. Krishnapuram, J.M. Keller, The possibilistic C-means algorithm: insights and recommendations, IEEE Trans. Fuzzy Syst. 4 (3) (1996) 385–393.
[25] R.P. Li, M. Mukaidono, A maximum-entropy approach to fuzzy clustering, in: Proceedings of the 1995 FUZZ/IEEE-IFES, Yokohama, Japan, March, 1995, pp. 2227–2232.
[26] R.P. Li, M. Mukaidono, Gaussian clustering method based on maximum–fuzzy-entropy interpretation, Fuzzy Sets Syst. 102 (1999) 253–258.
[27] X. Wu, B. Wu, J. Sun, H. Fu, Unsupervised possibilistic fuzzy clustering, J. Inf. Comput. Sci. 5 (2010) 1075–1080.
[28] M. Barni, V. Cappellini, A. Mecocci, Comments on a possibilistic approach to clustering, IEEE Trans. Fuzzy Syst. 4 (3) (1996) 393–396.
[29] S.D. Xenaki, K.D. Koutroumbas, A.A. Rontogiannis, A novel adaptive possibilistic clustering algorithm, IEEE Trans. Fuzzy Syst. 24 (2015) 791–810.
[30] Y. Fukuyama, M. Sugeno, A new method of choosing the number of clusters for the fuzzy C-means method, in: Proceedings of the Fifth Fuzzy Systems Symposium, 1989, pp. 247–250.
[31] A. Asuncion, D.J. Newman, UCI Machine Learning Repository, School of Information and Computer Science, University of California, Irvine, CA, USA, 2007 [online]. Available: http://archive.ics.usi.edu/ml/datasets.html .
[32] R.J.G.B. Campello, A fuzzy extension of the rand index and other related indexes for clustering and classification assessment, Pattern Recognit. Lett. 28 (2007) 833–841.
[33] D. Graves, W. Pedrycz, Kernel-based fuzzy clustering and fuzzy clustering: a comparative experimental study, Fuzzy Sets Syst. 161 (2010) 522–543.
[34] A. Strehl, J. Ghosh, Cluster ensembles – a knowledge reuse framework for combining multiple partitions, J. Mach. Learn. Res. 3 (2003) 583–617.
[35] R.B. Calinski, J. Harabasz, A dendrite method for cluster analysis, Commun. Stat. 3 (1) (1974) 1–27.
[36] H.C. Huang, Y.Y. Chuang, C.S. Chen, Multiple kernel fuzzy clustering, IEEE Trans. Fuzzy Syst. 20 (2012) 358–371.
[37] L. Hubert, P. Arabie, Comparing partitions, J. Classif. 2 (1985) 193–218.
[38] X.L. Xie, G.A. Beni, Validity measure for fuzzy clustering, IEEE Trans. Pattern Anal. Mach. Intell. 13 (8) (1991) 841–847.
[39] M.M Gupta, Fuzzy Automata and Decision Processes, 108, North-Holland, 1977.
[40] D.L. Davies, D.W. Bouldin, A cluster separation measure, IEEE Trans. Pattern Anal. Mach. Intell. PAMI-1 (1979) 224–227.

Yiming Tang received the Ph.D. degree from Hefei University of Technology in 2011. He is now an Associate Professor in Hefei University of Technology, and his research interests include clustering, image processing, fuzzy system, fuzzy logic, and affective computing.

He has published more than 50 papers in many journals and conferences. He serves an Associate Editor of JOURNAL OF MATHEMATICS AND INFORMATICS, and also a senior member of CAAI (Chinese Association for Artificial Intelligence) and a member of IEEE, CCF (China Computer Federation). He serves as a Professional Committee of Multiple-Valued and Fuzzy Logic of CCF, a Professional Committee of Rough Sets and Soft Computing of CAAI, a Professional Corresponding Committee of Machine Learning of CAAI. He is a

reviewer of several journals (including IEEE TRANSACTIONS ON FUZZY SYSTEMS, INFORMATION SCIENCES, ISPRS JOURNAL OF PHOTOGRAMMETRY AND REMOTE SENSING, SIGNAL PROCESSING). He is now a visiting professor at University of Alberta, Canada.

Xianghui Hu received the Bachelor Degree in 2016 from School of Computer and Information, Hefei University of Technology. She is now a Master Degree Candidate. Her research direction lies in fuzzy clustering, and image processing.

Witold Pedrycz received the M.Sc., Ph.D., and D.Sc. degrees from Silesian University of Technology, Gliwice, Poland, in 1977, 1980, and 1984, respectively. He is a Professor and the Canada Research Chair of computational intelligence with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. He is also with the Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland, where he was elected as a foreign member in 2009. He also holds an appointment as Special Professor with the School of Computer Science, University of Nottingham, Nottingham, U.K. His current research interests include computational intelligence, image processing, fuzzy modeling and granular computing, knowledge discovery and data mining, fuzzy control, pattern recognition, knowledge-based neural networks, relational computing, and software engineering. He has authored or coauthored numerous papers. He is also the author of 14 research monographs that cover various aspects of computational intelligence and software engineering.

He is the Editor-in-Chief of INFORMATION SCIENCES. He is also an Associate Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS and APPLIED SOFT COMPUTING. He has edited a number of volumes, including Handbook of Granular Computing (Wiley, 2008). He has been a member of numerous program committees of IEEE conferences in the area of fuzzy sets and neurocomputing. He received the prestigious Norbert Wiener Award from the IEEE Systems, Man, and Cybernetics Council in 2007. He received the IEEE Canada Computer Engineering Medal in 2008 and the Cajastur Prize for Soft Computing in 2009 from the European Centre for Soft Computing for pioneering and multifaceted contributions to granular computing.

Xiaocheng Song is now a Master Degree Candidate. His research direction focuses on fuzzy clustering, and image processing.