Int. J. Fuzzy Syst. (2017) 19(5):1585–1602
DOI 10.1007/s40815-016-0260-3

Tune Up Fuzzy C-Means for Big Data: Some Novel Hybrid Clustering Algorithms Based on Initial Selection and Incremental Clustering

Le Hoang Son[1,2] · **Nguyen Dang Tien**[3,4]

Received: 7 January 2016 / Revised: 27 July 2016 / Accepted: 19 September 2016 / Published online: 3 October 2016

Abstract Data are getting larger, and most of them are necessary for our businesses. Rapid explosion of data brings us a number of challenges relating to its complexity and how the most important knowledge can be captured in reasonable time. Fuzzy C-means (FCM)—one of the most efficient clustering algorithms which have been widely used in pattern recognition, data compression, image segmentation, computer vision and many other fields—also faces the problem of processing large datasets. In this paper, we propose some novel hybrid clustering algorithms based on incremental clustering and initial selection to tune up FCM for the Big Data problem. The first algorithm determines meshes of rectangle covering data points as the representatives, while the second one considers data points that have high influence to others as the representatives. The representatives are then clustered by FCM, and the new centers are selected as initial ones for clustering of the dataset. Theoretical analyses of the new algorithms including comparison of quality of solutions when clustering the representatives set versus the entire set are examined. The experimental results on both simulated and real datasets show that total computational time of the new methods including time of finding representatives and clustering is faster than those of other relevant algorithms. The validation on clustering quality is also examined. The findings of this paper have great impact and significance to researches in the fields of soft computing and Big Data processing. It is obvious that computing methodologies nowadays are facing with huge amount of diverse and complex data structures. Speed of processing is the main priority when considering effectiveness of a specific method. The findings demonstrated practical algorithms and investigated their characteristics that could be referenced by other researchers in similar applications. The usefulness and significance of this research are clearly demonstrated within the extent of real-life applications.

Keywords Big data · Density-based clustering · Fuzzy C-means · Grid-based clustering · Incremental clustering · Initial selection

1 Introduction

Fuzzy C-means (FCM) [4] is considered as a strong aid of rule extraction and data mining from a set of data in which fuzzy factors are common [7, 13]. It has been used for pattern recognition, data compression, image segmentation, computer vision and many other fields [21, 23, 29–39, 41, 43, 44, 46, 51]. Given a dataset of N attributes: $X = \{x_1, \ldots, x_N\}$, with $x_k \in R^p$ and $V = (v_1, \ldots, v_C)$ being the centers of all groups. The FCM algorithm aims to minimize:

$$J = \sum_{k=1}^{N} \sum_{j=1}^{C} u_{kj}^m \left\| x_k - v_j \right\|^2 \to \min, \tag{1}$$

where u_{kj} is the membership value showing the degree to which element x_k belongs to cluster c_j and m is the fuzzifier

✉ Le Hoang Son
lehoangson@tdt.edu.vn

Nguyen Dang Tien
dangtient36@gmail.com

[1] Division of Data Science, Ton Duc Thang University, Ho Chi Minh City, Vietnam

[2] Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Vietnam

[3] People's Police University of Technology and Logistics, Bac Ninh, Vietnam

[4] VNU University of Science, Vietnam National University, Hanoi, Vietnam

which determines the level of cluster fuzziness. The constraints for the optimization problem are:

$$\begin{cases} u_{kj} \in [0,1] \\ \sum_{j=1}^{C} u_{kj} = 1, \forall k = \overline{1,N} \end{cases}. \tag{2}$$

Problem (1–2) is solved by an iterative schema that computes the cluster centers and the membership matrix until the difference between two consecutive iteration is smaller than a given threshold. This algorithm was proved to converge to the saddle points by Bezdek et al. [4].

Nevertheless, data tend to be huge and diverse as they can come from everywhere: sensors used to gather climate information, posts to social media sites, digital pictures and videos, purchase transaction records, and cell phone GPS signals to name a few. Some of them are even in different organizations and structures. Rapid explosion of data brings a number of challenges regarding how the most important knowledge can be captured in reasonable time. Jain [17] stated that clustering is the key to the Big Data problem since it provides efficient browsing, search, recommendation and organization of data without prior knowledge of the number and nature of groups in data. However, the FCM algorithm still faces the problem of processing large datasets. A simple and straightforward method to accelerate FCM for the Big Data problem is hence required.

Scanning the literature, we recognize that there are numbers of solutions dealing with this matter [2, 3, 10, 25, 28] including data reduction [6, 14, 15, 18, 22, 27, 49, 50], initial selection [12, 16, 24, 26, 42], suppression [5, 11, 40] and incremental clustering [1, 9, 47, 48]. Motivated by developing a simple and straightforward method to accelerate FCM for the Big Data problem, the idea of hybrid algorithms between incremental clustering and initial selection is utilized. Incremental clustering and initial selection do not require complex preprocessing procedures like other groups. Incremental clustering works based on the assumption that small representatives of the dataset could represent for the whole dataset; therefore, clustering in the representative sets obtains both fast computational speed and reasonable quality in comparison with the clustering in the entire dataset. Initial selection seeks out appropriate initial centers and makes the clustering process faster due to the closeness of the final centers with the initial ones. However, an important question regarding these approaches is how to determine such the representatives and initial centers accordingly.

In this paper, we aim to develop some novel hybrid algorithms based on incremental clustering and initial selection to tune up FCM for the Big Data problem. Specifically, *the first algorithm*, named as GB-FCM, determines meshes of a rectangle covering data points as the representatives whose active ones that cover at least a data point are classified into groups by the FCM algorithm. The new centers, which reflect more accurately the nature of dataset, are selected as initial centers of FCM to classify the entire dataset. *The second algorithm,* named as DB-FCM, considers data points that have high influence to others as the representatives. The term ‘influence’ is equivalent to the number of neighbors in a sphere whose center is a given data point. High-influence data points can be the best reduction of entire dataset. Again, these representatives are clustered by the FCM algorithm and the new centers are selected as initial ones for clustering of the dataset.

The main difference of these algorithms with the relevant ones (e.g. the standalone incremental clustering and initial selection) is how to determine the representatives and initial centers. Unlike incremental clustering that considers centers/medoids, the new algorithms choose either rectangular meshes or data points as representatives on the basis of distribution of data points. Unlike initial selection that often finds out initial centers by metaheuristic optimization methods, the new algorithms utilize the representatives to clarify centers. These modifications do make the proposed algorithm faster and more precise than existing clustering techniques. Lastly, theoretical analyses of the new algorithms are also examined.

Rest of this paper is organized as follows. Sections 2 and 3 describe two novel methods. Section 4 experimentally validates these algorithms in comparison with the relevant ones on both simulated and real datasets. Finally, Section 5 draws the conclusions and delineates future works.

2 Grid-Based Approximation for Fuzzy Clustering

2.1 The Algorithm

The objective function and constraints of the new algorithm, named as *Grid-Based Approximation for Fuzzy Clustering algorithm* (GB-FCM), are expressed as follows.

$$\tilde{J} = \sum_{h=1}^{l} \sum_{j=1}^{C} u_{hj}^{m} \left\| o_h - v_j \right\|^2 \to \min, \tag{3}$$

$$\begin{cases} u_{hj} \in [0,1] \\ \sum_{j=1}^{C} u_{hj} = 1, \forall h = \overline{1,l} \end{cases}, \tag{4}$$

where o_h is the representative of the set $\{x_k\}$, w_h is the number of data points that o_h represents and l is the number of cells.

$$N = \sum_{h=1}^{l} w_h. \tag{5}$$

The GB-FCM algorithm is demonstrated in Table 1. Note that in Step 2 of this algorithm, if an interval in a given axis has no or a data point inside, recursively merge it with its right interval until the number of data points is larger than or equal to 2. By this modification, each interval is assumed to contain at least 2 data points, which are dense enough for clustering.

Example 1 Consider the following dataset with the bound of data points being set up as in Step 1 of the algorithm. Herein, $\varepsilon_X = 0.5$ and $\varepsilon_Y = 1$ (Fig. 1).

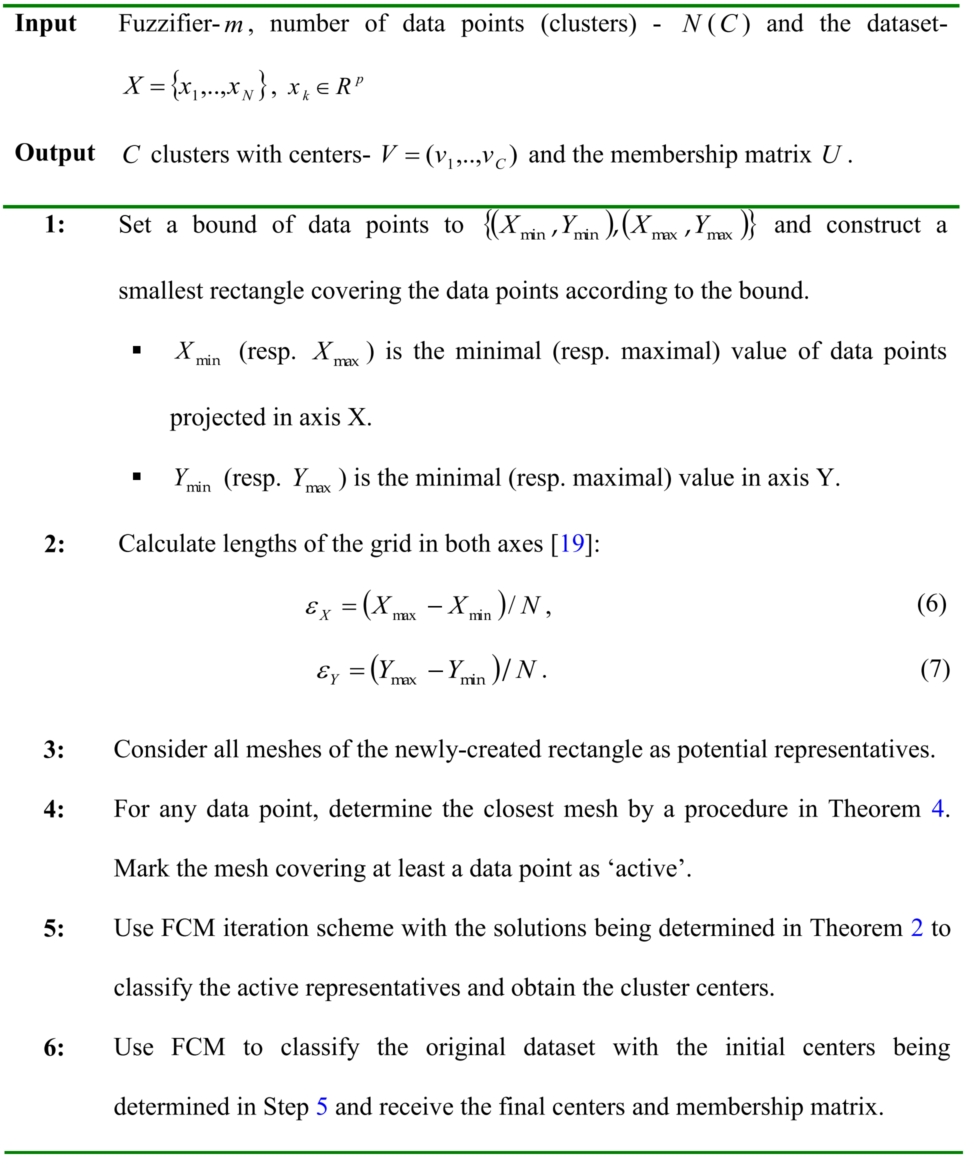
According to the Step 3, we consider all meshes of the newly created rectangle as potential representatives (Fig. 2).

Rectify all meshes in Fig. 2 by Step 4, we get the active representatives in red color (Fig. 3).

Use FCM to classify the active representatives in Fig. 3 into 2 groups and obtain the cluster centers represented by the triangular symbol in Fig. 4.

Herein, we recognize that the achieved cluster centers are 'nearly' approximate to the optimal cluster centers of the original dataset. The final cluster centers are obtained by the same process of FCM with the initial solutions being extracted above (See Step 6, Fig. 5). In fact, when the

Table 1 Algorithm 1: GB-FCM



```
Input   Fuzzifier-$m$, number of data points (clusters) - $N(C)$ and the dataset-
        $X = \{x_1,..,x_N\}$, $x_k \in R^p$
Output  $C$ clusters with centers- $V = (v_1,..,v_C)$ and the membership matrix $U$.

1:  Set a bound of data points to $\{(X_{\min}, Y_{\min}), (X_{\max}, Y_{\max})\}$ and construct a
    smallest rectangle covering the data points according to the bound.
      ▪ $X_{\min}$ (resp. $X_{\max}$) is the minimal (resp. maximal) value of data points
        projected in axis X.
      ▪ $Y_{\min}$ (resp. $Y_{\max}$) is the minimal (resp. maximal) value in axis Y.
2:  Calculate lengths of the grid in both axes [19]:
        $\varepsilon_X = (X_{\max} - X_{\min})/N$,    (6)
        $\varepsilon_Y = (Y_{\max} - Y_{\min})/N$.    (7)
3:  Consider all meshes of the newly-created rectangle as potential representatives.
4:  For any data point, determine the closest mesh by a procedure in Theorem 4.
    Mark the mesh covering at least a data point as 'active'.
5:  Use FCM iteration scheme with the solutions being determined in Theorem 2 to
    classify the active representatives and obtain the cluster centers.
6:  Use FCM to classify the original dataset with the initial centers being
    determined in Step 5 and receive the final centers and membership matrix.
```

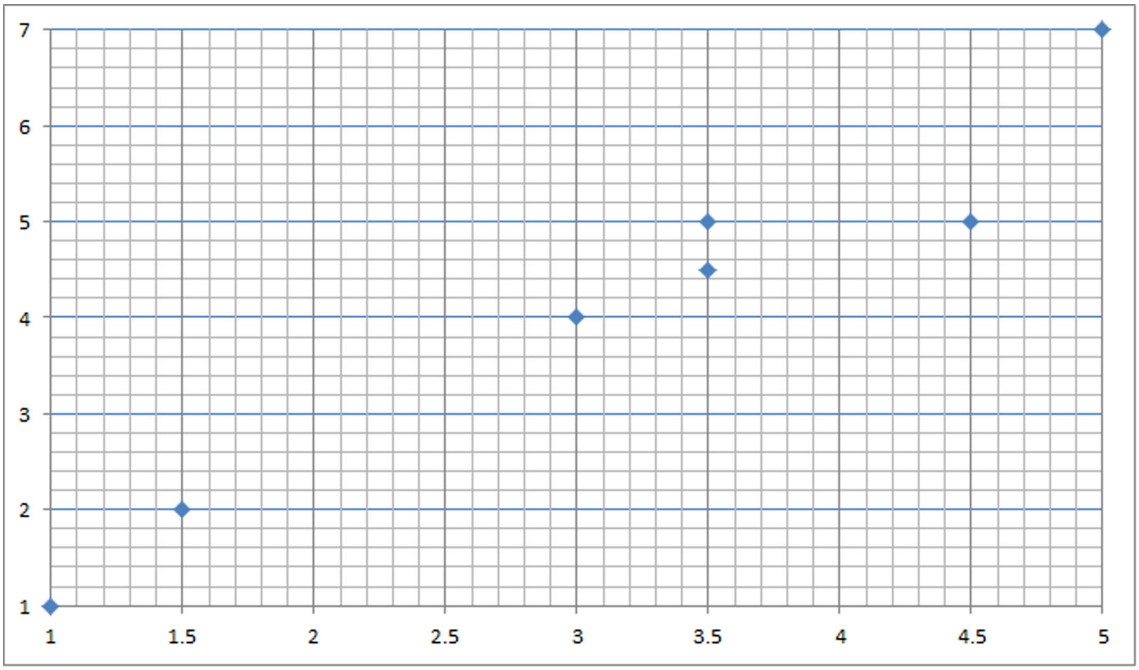


Fig. 1 Small dataset

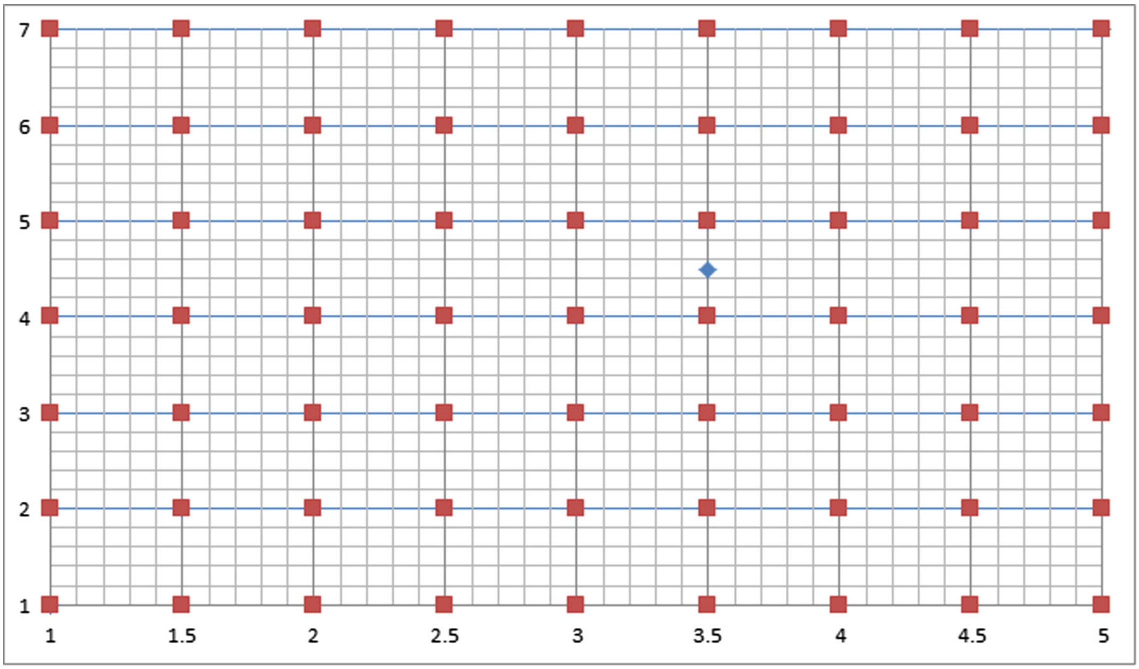


Fig. 2 Possible meshes for the dataset

number of data points is large, this method is quite efficient to quickly determine the optimal centers and results.

2.2 Theoretical Analyses of GB-FCM

Firstly, we prove that the problem in (1, 2) is equivalent to the problem in (3, 4).

Theorem 1 *After replacing* $\{x_k\}$ *by its representative* o_h*, the problem (1, 2) is reduced to minimize the alternative objective function (3) with the partition matrix U satisfying the condition (4).*

Proof From (1), we have

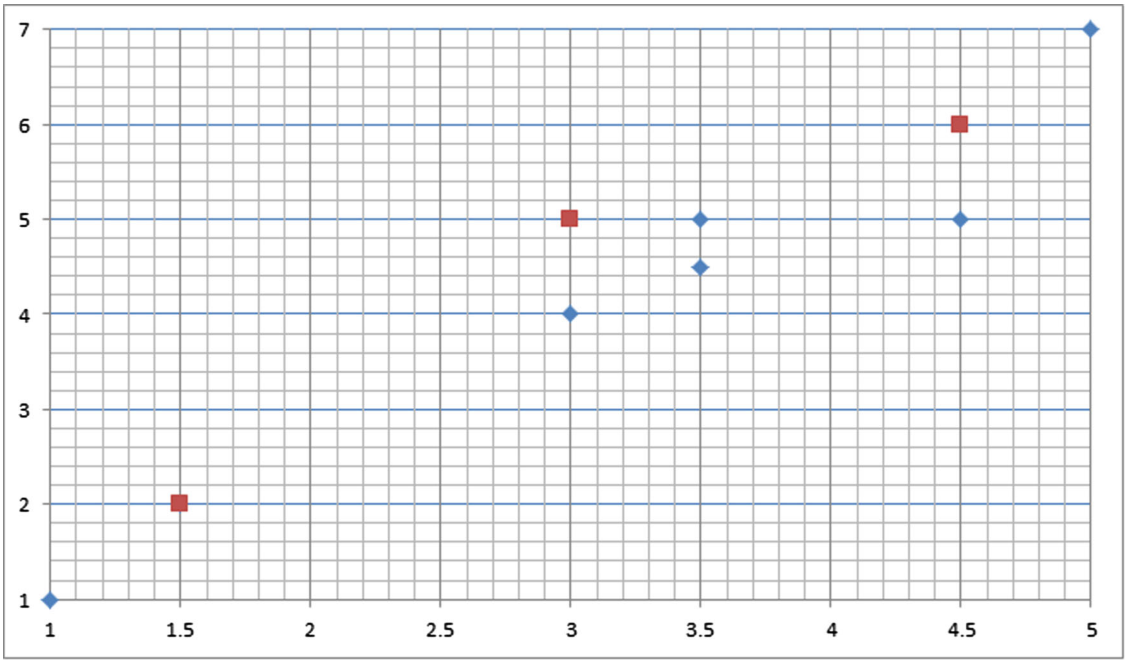


Fig. 3 Active meshes for the dataset

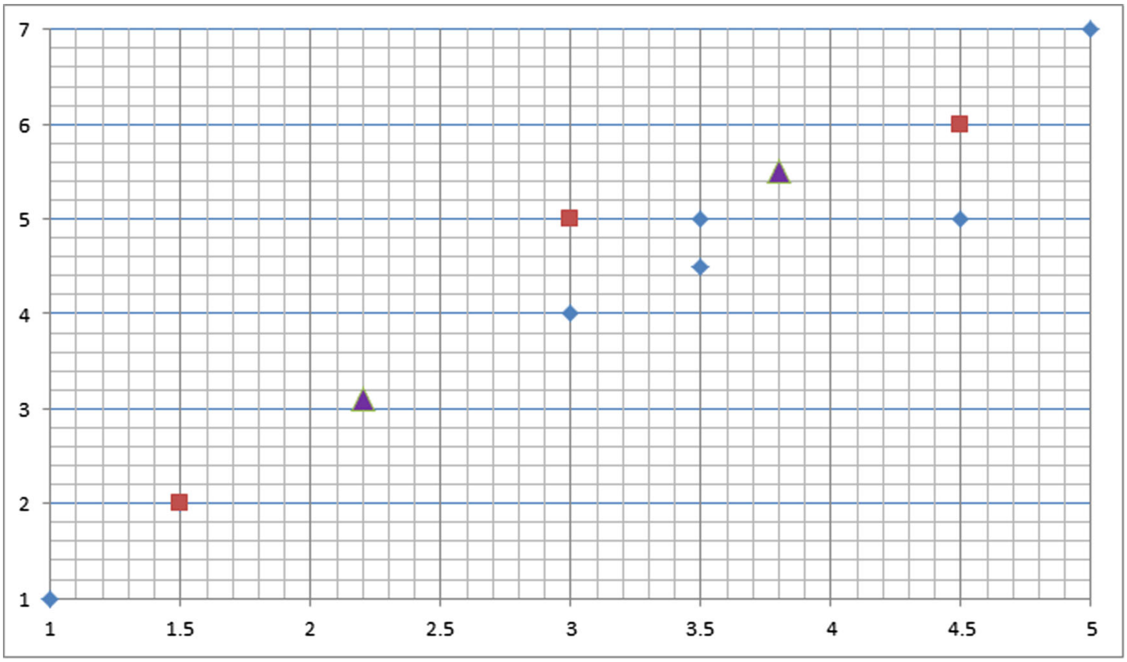


Fig. 4 Use FCM for the representatives

$$J = \sum_{h=1}^{l}\sum_{k=1}^{w_h}\sum_{j=1}^{C} u_{kj}^{m}\left\|o_h - v_j\right\|^2. \tag{8}$$

Apply Jensen's inequality, we get

$$\frac{\sum_{k=1}^{w_h} u_{kj}^{m}}{w_h} \geq \left(\frac{\sum_{k=1}^{w_h} u_{kj}}{w_h}\right)^{m} = u_{hj}^{m}. \tag{9}$$

Therefore, from (8) and (9) we have

$$J \geq \tilde{J} = \sum_{h=1}^{l}\sum_{j=1}^{C} u_{hj}^{m} w_h \left\|o_h - v_j\right\|^2. \tag{10}$$

Because of the property of u_{kj}, it is obvious that,

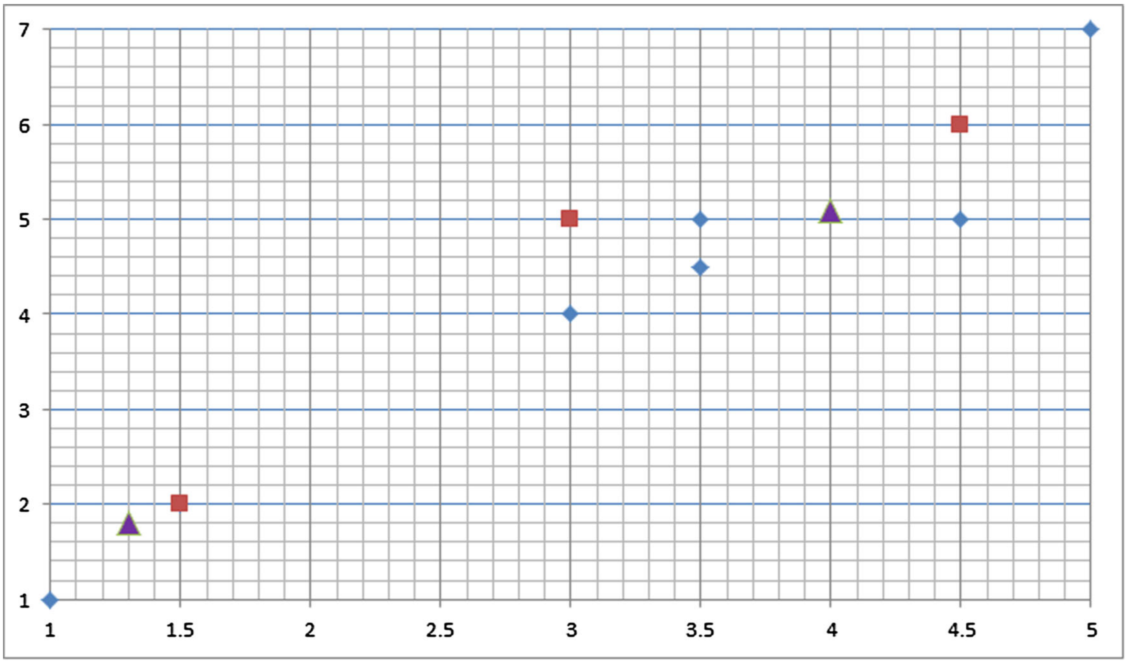


Fig. 5 Final cluster centers

$$u_{kj} \in [0,1] \Rightarrow u_{hj} = \frac{\sum_{k=1}^{w_h} u_{kj}}{w_h} \in [0,1] \tag{11}$$

$$\sum_{j=1}^{C} u_{kj} = 1 \Rightarrow \sum_{j=1}^{C} u_{hj} = \frac{\sum_{k=1}^{w_h} \sum_{j=1}^{c} u_{kj}}{w_h} = 1 \tag{12}$$

□

Theorem 2 *The optimal solutions of the system (3) and (4) are*

$$v_j = \frac{\sum_{h=1}^{l} u_{hj}^m.w_h.o_h}{\sum_{h=1}^{l} u_{hj}^m.w_h}; \quad j = \overline{1,C} \tag{13}$$

$$u_{hj} = \frac{1}{\sum_{i=1}^{C} \left(\frac{\|o_h - v_j\|}{\|o_h - v_i\|} \right)^{\frac{2}{m-1}}} \tag{14}$$

Proof Similar to the proof of Bezdek et al. [4], we use the gradient method for the objective function $\tilde{J}$ in (3),

$$\tilde{J} = \sum_{h=1}^{l} \sum_{j=1}^{C} u_{kj}^m w_h (o_h - v_j)^t (o_h - v_j), \tag{15}$$

$$\Rightarrow \frac{\partial \tilde{J}}{\partial v_j} = \sum_{h=1}^{l} u_{hj}^m w_h \left(-2o_h + 2v_j\right) \tag{16}$$

$$\frac{\partial \tilde{J}}{\partial v_j} = 0 \Rightarrow v_j = \frac{\sum_{h=1}^{l} u_{hj}^m \cdot w_h \cdot o_h}{\sum_{h=1}^{l} u_{hj}^m \cdot w_h}, \quad j = \overline{1,C}. \tag{17}$$

Use the Lagrange multiplier, we get

$$L(u,\lambda) = \sum_{h=1}^{l} \sum_{j=1}^{C} u_{kj}^m w_h \left\|(o_h - v_j)\right\|^2 - \sum_{h=1}^{l} \lambda_h \left(\sum_{j=1}^{C} u_{hj} - 1 \right), \tag{18}$$

$$\frac{\partial L(u,\lambda)}{\partial u_{hj}} = 0 \Rightarrow u_{hj} = \left(\frac{\lambda_h}{m.w_h.\left\|o_h - v_j\right\|^2} \right)^{\frac{1}{m-1}}, \quad h = \overline{1,l};\ j = \overline{1,C} \tag{19}$$

Because of constraint (4), we obtain

$$\lambda_h = m \cdot w_h \sum_{i=1}^{C} \left\|o_h - v_j\right\|^2 \ h = \overline{1,l}. \tag{20}$$

From Eqs. (19) and (20), we have

$$u_{hj} = \left(\frac{\sum_{i=1}^{C} \|o_h - v_i\|^2}{\left\|o_h - v_j\right\|^2} \right)^{\frac{1}{m-1}} = \frac{1}{\sum_{i=1}^{C} \left(\frac{\|o_h - v_j\|}{\|o_h - v_i\|} \right)^{\frac{2}{m-1}}} \quad h = \overline{1,l};\ j = \overline{1,C}. \tag{21}$$

From Eqs. (17, 21), we have the solutions of the system (3) and (4).

Theorem 2 allows us to determine the centers and the partition matrix described in Step 5 of the GB-FCM algorithm.

Theorem 3 The minimal cost of clustering representatives differs from the minimal cost of clustering the original dataset a quantity of ε × *l*.

Proof Denote by $(\overline{u_{kj}}, \overline{v_j})$ the optimal solutions of original problem and $(\overline{\overline{u_{hj}}}, \overline{\overline{v_j}})$ the optimal solutions of representative clustering. From the minimal property of these solutions, we have

$$J'_{\min} \le \sum_{k=1}^{N}\sum_{j=1}^{C} \overline{u_{kj}}^{m} \left\| o_h - \overline{v_j} \right\|^2, \tag{22}$$

$$\le \sum_{k=1}^{N}\sum_{j=1}^{C} \overline{u_{kj}}^{m} \left(\left\| x_k - \overline{v_j} \right\|^2 + \left\| o_h - x_k \right\|^2 \right), \tag{23}$$

$$\le J_{\min} + \sum_{h=1}^{l} \varepsilon_h, \tag{24}$$

$$\le J_{\min} + l \times \varepsilon. \tag{25}$$

where ε_h in (24) is the farthest distance in each set $\{x_k\}$ to its representative o_h.

$$\varepsilon = \max \varepsilon_h, \quad h = \overline{1, l}. \tag{26}$$

From the inequality (25), we recognize that the centers of the problem (1–2) are not too far from those of representative clustering in Eqs. (3–4). □

Next, we assess the quality of choosing representatives in GB-FCM. Let us state some definitions below.

Definition 1 The quality of a representative method is defined by a minimal number a_{method} satisfying condition,

$$s_{\text{method}} \le a_{\text{method}} \times s_{\min}, \tag{27}$$

where $s_{\min}$ is the optimal number of spheres with radii ε, s_{method} is the number of spheres.

Definition 2 Method 1 is said to be better than method 2 if

$$a_{\text{method1}} < a_{\text{method2}}. \tag{28}$$

If we consider all meshes in a grid as potential representatives, then a possible determination of a data point to any mesh is specified as follows. If for any x_k in the dataset, there exists a coordinate,

$$x_{ki}^{j} = t \cdot \varepsilon + \frac{\varepsilon}{2}, \quad t \in Z, \tag{29}$$

then we assign x_k to representatives having coordinates $o_h^j < x_k^j$.

Theorem 4 *The quality of the grid-based method is:* $a_{gridmethod} = 3^d$*, for which d is the number of dimensions of feature space. In the other words,*

$$s_{\text{gridmethod}} \le 3^d \times s_{\min}. \tag{30}$$

Proof

- If $d = 1$: Assume that o_h, o_{h+1}, o_{h+2}, o_{h+3} are the consecutive meshes (Fig. 6). For any data point having the coordinate in this dimension is y, there are three possible cases.

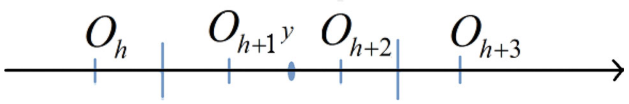


Fig. 6 Determine a suitable mesh for representing data points having coordinate y

  - *Case 1*: If $y < (o_{k+1} + o_{k+2})/2$, all data points that y represents would be represented by o_h, o_{h+1}, o_{h+2}.
  - *Case 2*: If $y > (o_{k+1} + o_{k+2})/2$, all data points that y represents would be represented by o_{h+1}, o_{h+2}, o_{h+3}.
  - *Case 3*: Ify $= (o_{k+1} + o_{k+2})/2$, all data points that $|y - x_k| < \varepsilon$ would be represented by o_h, o_{h+1}, o_{h+2}. The left endpoint certainly belongs to o_h. The right endpoint is assigned to o_{h+2} because it is less than $(o_{h+2} + o_{h+3})/2$.
    In this dimension, we need three representatives for an optimal representative y.

- If $d > 1$: It is due to the fact that in every dimension, we need the maximum three rows of representatives.

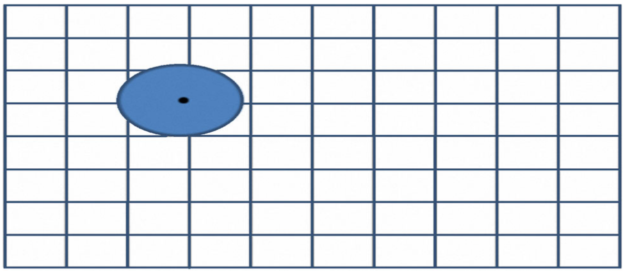


Fig. 7 Determine number of representatives when the dimension is greater than 1

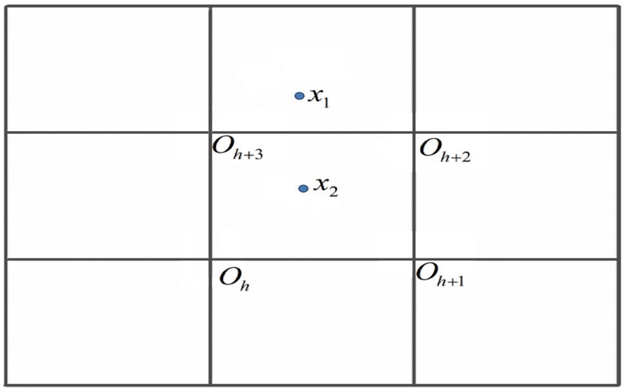


Fig. 8 Example of mesh determination and representatives for $\{x_1, x_2\}$

Thus, in d dimensions, we totally need 3^d representatives (Fig. 7) □

Example 2 In Fig. 8, the point x_1 is assigned to o_{h+3}, x_2 is assigned to o_h.

3 Density-Based Approximation for Fuzzy Clustering

3.1 The Algorithm

In this section, we introduce another algorithm, named as *Density-Based Approximation for Fuzzy Clustering Method* (DB-FCM). The descriptions of this algorithm are shown in Table 2. Note that in Step 5, new representative of merged sphere is calculated by the average value of old representatives. In cases that there is only one sphere for the whole dataset after merging, return to Step 1 with a different initial data point. If the results of two consecutive initializations are identical, stop the algorithm.

Example 3 Consider the dataset in Example 1 (Fig. 1). Herein, $\varepsilon_X = 0.5$ and $\varepsilon_Y = 1$ so $\varepsilon = 1.118$. Steps 1–4 create some representatives of the sphere with radius ε (Fig. 9). The red points are the centers of the spheres.

Use FCM to classify the centers into 2 groups, and finally perform FCM again to classify the original dataset with the initial centers of the previous steps, we get the results in Fig. 10. Notice that these steps are analogous to those in GB-FCM.

Table 2 Algorithm 2: DB-FCM

```
Input   Fuzzifier-$m$, number of data points (clusters) - $N(C)$ and the dataset-
        $X = \{x_1,..,x_N\}$, $x_k \in R^p$
Output  $C$ clusters with centers- $V = (v_1,..,v_C)$ and the membership matrix $U$.

1:  Choose a point $x_k$ from $X$ that has not been scanned yet and let it be a pivot
    element.
2:  Assign other points to the sphere with center $x_k$ and radius $\varepsilon$ with $\varepsilon_X$ and $\varepsilon_Y$
    being calculated in equations (6-7).
        $\varepsilon = \sqrt{\varepsilon_X^2 + \varepsilon_Y^2}$,                    (31)
3:  Calculate a new representative of the sphere:
        $\hat{x} = \frac{x_k + \sum x_i}{Number\_of\_elements}$.                  (32)
4:  Repeat Step 1-3 until all data points are assigned.
5:  If the distance between two representatives is smaller than $\varepsilon$ then merge two
    spheres.
6:  Use FCM iteration scheme with the solutions being determined in Theorem 2 to
    classify the active representatives and obtain the cluster centers.
7:  Use FCM to classify the original dataset with the initial centers being
    determined in Step 6 and receive the final centers and membership matrix.
```

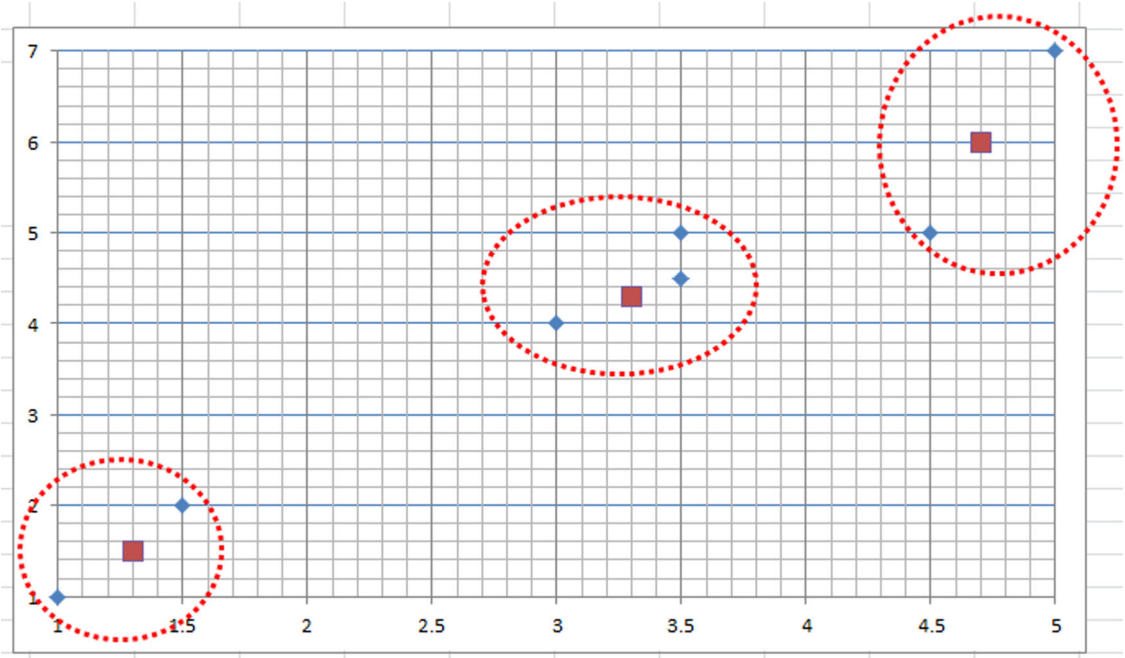


Fig. 9 Forming spheres

3.2 Theoretical Analyses of DB-FCM

Firstly, from Definition 1, we would like to know whether or not the quality of DB-FCM is better than that of GB-FCM. The following theorem helps us answer this question.

Theorem 5 *DB-FCM is better than GB-FCM. In other words,*

$$a_{\text{densitybased}} < a_{\text{gridbased}}. \tag{33}$$

Proof We recognize that $a_{\text{densitybased}}$ is the maximal number of spheres whose centers lie in a given radius sphere and their distances are greater than ε. It comes from the fact that data points belong to one sphere representative only, and the next representative must not belong to any previous sphere representative (Fig. 11). By the condition centers in the given sphere, the problem can be changed to finding the maximal number of spheres with radii ε/2 that can be put into a sphere with radius 3ε/2 (Fig. 12). Besides, all spheres with radii ε/2 are not allowed to touch together. Obviously, this property is equivalent to the fact that distances of centers are greater than ε. When all centers are in the boundary of the given sphere of the original problem, the spheres with radii ε/2 touch the boundary of the sphere with radius 3ε/2.

$a_{\text{densitybased}}$ is the maximal number of spheres with radii ε/2 that can be put into a sphere with radius 3ε/2. It is less than or equal to the maximal number of spheres if we dismiss the condition that spheres do not touch together. We know the volume,

$$V_{\frac{3\varepsilon}{2}} = A_d.\left(\frac{3\varepsilon}{2}\right)^d, \tag{34}$$

$$V_{\frac{\varepsilon}{2}} = A_d \cdot \left(\frac{\varepsilon}{2}\right)^d, \tag{35}$$

where A_d is a constant in d dimensional space. Because spheres with radii ε/2 lie in the sphere with radius 3ε/2, the maximal number of spheres is less than or equal to,

$$\frac{V_{\frac{3\varepsilon}{2}}}{V_{\frac{\varepsilon}{2}}} = 3^d. \tag{36}$$

However, this number cannot be 3^d because in that situation, data points in the sphere with radius 3ε/2 are covered by spheres with radius ε/2. In fact, the number of intersections between spheres with radii ε/2 and the sphere with radius 3ε/2 is limited. Indeed, a limited number of points in the boundary of the sphere with radius 3ε/2 are covered by spheres with radii ε/2. Therefore, the quality of density-based method is better than the one of the grid-based and satisfies the inequality,

$$a_{\text{scanning}} < 3^d - 1. \tag{37}$$

Theorem 6 (*A better representative method in the dimension that is smaller than three—half-sphere representative method*).

In fact, the half-sphere representative method is also iterative. Initially, we fix a dimension, then iterate.

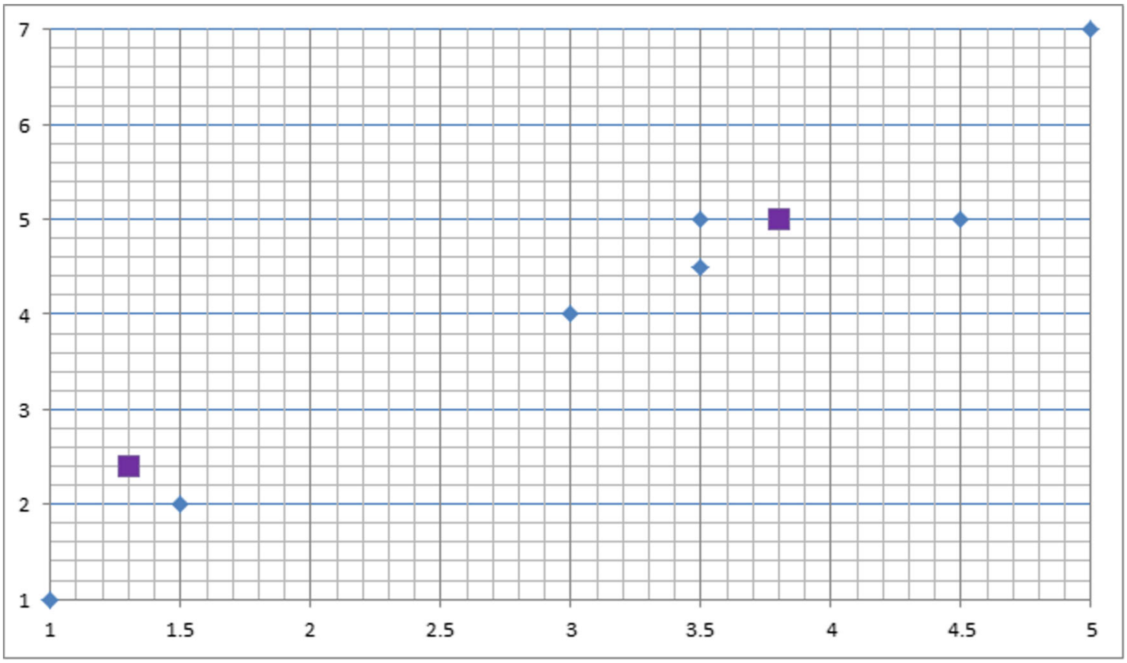


Fig. 10 Final cluster centers of DB-FCM

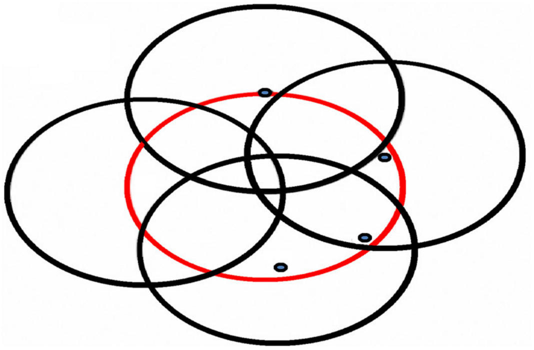


Fig. 11 Data points belong to the red sphere representative only

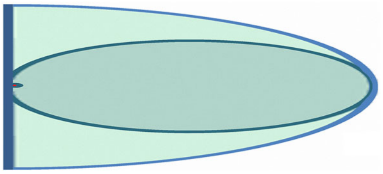


Fig. 13 Representative is the red point which has smallest value among all

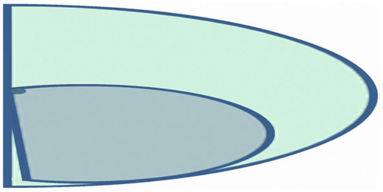


Fig. 14 Representative is the lowest point in the remaining part of representative sphere

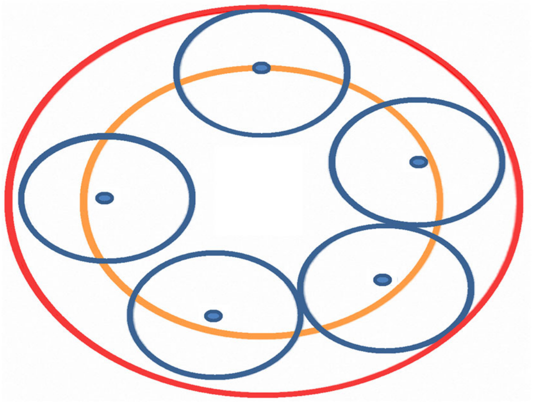


Fig. 12 Alternative problem: finding maximal number of spheres in a red sphere

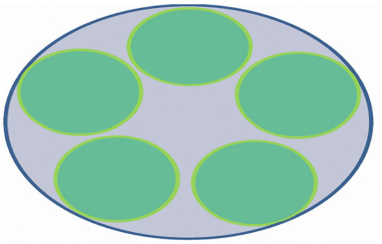


Fig. 15 Five spheres with radii ε/2 can be arranged in a sphere with radius 3ε/2

- Choose the point having the smallest value on that dimension. Scan all points that lie in the half-sphere with radius 2ε.
- Replace the half-sphere by a set of spheres whose radii cover all its points.

At the first iteration, the half-sphere representative is the pole in a fixed dimension (Fig. 13). In next steps of iteration, the representative is the lowest point in the remaining part of some optimal representative sphere (Fig. 14).

In one-dimensional space, it is just the interval width of ε. The $a_{\text{halfsphere}}$ in this case is equal to one. This means that the number of representatives chosen by the half-sphere method is minimal. In addition, it is easy to check that $a_{\text{densitybased}} = 2$. In two-dimensional space, $a_{\text{densitybased}} \geq 5$, because we can arrange five spheres with radii $\varepsilon/2$ in a sphere with radius $3\varepsilon/2$ (Fig. 15). Thus, we just need four spheres with radii ε to cover a sphere with radius $2.\varepsilon$. The set of four spheres are constructed by putting a sphere with the center is the midpoint of the cut line of the half-sphere, then drawing a half of hexagon and putting three left spheres' centers in the midpoints of edges of this half hexagon. Thus, in two-dimensional space, $a_{\text{halfsphere}} \leq 4$. It follows that the half-sphere representative method in this case is better than the density-based method. In fact, $a_{\text{halfsphere}}$ is the minimal number of spheres with radii ε to cover a sphere with radius $2.\varepsilon$.

4 Results

4.1 Experimental Environment

- *Experimental tools*: We have implemented the proposed algorithms (GB-FCM and DB-FCM) in addition to FCM [4] and the stand-alone methods of initial selection—psFCM [16], suppression—neural network [5] and incremental clustering—FHCA [9] in C programming language and executed them on a Linux Cluster 1350 with eight computing nodes of 51.2GFlops. Each node contains two Intel Xeon dual core 3.2 GHz, 2 GB Ram.
- *Experimental datasets*: We use six simulated (D_1–D_6) and four real (T_1–T_4) datasets described in Table 3.
    - *Simulated datasets*: Datasets from D_1 to D_3 are generated from a Gaussian distribution by using the Marsaglia [20] method. The standard deviation of these data points in each cluster is one. The dimension of these datasets is two. The datasets from D_4 to D_6 are also generated from a Gaussian distribution, but in the five-dimensional space;
    - *Real* datasets: they are taken from UCI KDD Archive Website [45]. T_1 is Forest CoverType including 581012 instances in 54 dimensions showing the actual forest cover type for a given observation (30 × 30 m cell) determined from US Forest Service (USFS) Region 2 Resource Information System (RIS). T_2 is a text dataset namely Enron Emails in Bag of Words Data Set including 39,861 numbers of documents, 28,102 numbers of words in the vocabulary and 3,710,420 number of words in total. T_3 is a multivariate and text dataset, namely KEGG Metabolic Relation Network (Directed) including 53,414 instances in 24 dimensions. T_4 namely NYSK (New York v. Strauss-Kahn) is a collection of English news articles about the case relating to allegations of sexual assault against the former IMF director Dominique Strauss-Kahn (May 2011) including 10,421 instances in 7 dimensions.
- *Cluster validity measurement*: Davies-Bouldin (DB) index [8] in Eqs. (38–40) to evaluate the clustering qualities. In these equations, T_i is the size of cluster ith, S_i is a measure of scatter within the cluster and M_{ij} is a measure of separation between cluster ith and jth. The minimum value indicates the better performance for DB index.

Table 3 Descriptions of the datasets

Dataset	Size (MB)	Dimensions	No. of Points	No. of Clusters (C)
D_1	4.27	2	200,000	16
D_2	10.6	2	500,000	12
D_3	21.3	2	1,000,000	12
D_4	2.33	5	50,000	12
D_5	3.66	5	80,000	8
D_6	4.58	5	100,000	8
T_1	11.2	54	581,012	8
T_2	47.4	2	3,710,420	10
T_3	7.0	24	53,414	12
T_4	18	7	10,421	16

$$DB = \frac{1}{C}\sum_{i=1}^{C}\left(\max_{j:j\neq i}\left\{\frac{S_i + S_j}{M_{ij}}\right\}\right) \tag{38}$$

$$S_i = \sqrt{\frac{1}{T_i}\sum_{j=1}^{T_i}\left|X_j - V_i\right|^2}, \quad (i = \overline{1,C}) \tag{39}$$

$$M_{ij} = \left\|V_i - V_j\right\|, \ (i = \overline{1,C},\ j = \overline{1,C},\ i \neq j), \tag{40}$$

- *Objective*:
  - To compare the total computational time of algorithms;
  - To compare the computational time of algorithms for finding representatives;
  - To measure the clustering qualities of algorithms.

4.2 The Comparison of the Total Computational Time

In this section, we compare the total computational time of all algorithms illustrated in Table 4. It is obvious that the computational time of the proposed methods (GB-FCM and DB-FCM) is smaller than those of other algorithms. Thus, the first remark from the experiments is that the proposed algorithms are faster than the relevant ones.

In what follows, we measure the computational time of all algorithms per data point. The results in Table 5 show that GB-FCM is the most effective algorithm because it takes smallest computational time to process a data point. The average results also demonstrate that FHCA is the worst algorithm for this matter.

Analogously, the comparison of computational time of all algorithms per the number of clusters in Table 6 also shows that GB-FCM is the most effective algorithms. This is the second remark from the experiments.

Next, we find out which algorithm is the most effective in all types of data (simulated and real). The results from Tables 4, 5 and 6 clearly show that GB-FCM is the fastest algorithm among all. Just to give an example: the computational time of GB-FCM on D_4 is 4.31, 11.49, 7.1, 10.8 and 140 times faster than those of DB-FCM, FCM, psFCM, neural network and FHCA, respectively. This is the third remark from the experiments.

Table 4 Computational time of all algorithms (s)

	GB-FCM	DB-FCM	FCM	psFCM	Neural network	FHCA
D_1	34.14	8.93	118.19	131.1	598.85	1388.25
D_2	36.13	22.29	420.18	286.88	1180	2781.61
D_3	38.83	45.03	1037.1	319.14	2342.7	6129.8
D_4	11.63	50.19	133.69	82.26	126.03	1633.7
D_5	30.06	76.33	141.21	148.59	144.2	1706.17
D_6	46.78	140.77	251.5	156.98	179.9	2589.16
T_1	586.52	626.15	11,016	4552.5	27,579	80,725
T_2	426.9	815.3	12,546	6587	33,242	93,684
T_3	134.1	245.6	487.2	1876	12,387	29,147
T_4	26.3	11.7	86.3	78.6	318.6	962.9

Table 5 Computational time of all algorithms per data point (sec)

	GB-FCM	DB-FCM	FCM	psFCM	Neural network	FHCA
D_1	0.000171	4.465E−05	0.000591	0.000656	0.0029943	0.0069413
D_2	7.23E−05	4.458E−05	0.0008404	0.000574	0.00236	0.0055632
D_3	3.88E−05	4.503E−05	0.0010371	0.000319	0.0023427	0.0061298
D_4	0.000233	0.0010038	0.0026738	0.001645	0.0025206	0.032674
D_5	0.000376	0.0009541	0.0017651	0.001857	0.0018025	0.0213271
D_6	0.000468	0.0014077	0.002515	0.00157	0.001799	0.0258916
T_1	0.001009	0.0010777	0.01896	0.007835	0.0474672	0.1389386
T_2	0.000115	0.0002197	0.0033813	0.001775	0.0089591	0.0252489
T_3	0.002511	0.004598	0.0091212	0.035122	0.2319055	0.5456809
T_4	0.002524	0.0011227	0.0082814	0.007542	0.0305729	0.0924
Average	0.00075168	0.0010518	0.0049166	0.00588959	0.0332724	0.0900795

Table 6 Computational time of all algorithms per cluster (sec)

	GB-FCM	DB-FCM	FCM
D_1	2.13375	0.558125	7.3868
D_2	3.01083	1.8575	35.015
D_3	3.23583	3.7525	86.425
D_4	0.96916	4.1825	11.1408
D_5	3.7575	9.54125	17.6512
D_6	5.8475	17.5962	31.4375
T_1	73.315	78.2687	1377
T_2	42.69	81.53	1254.6
T_3	11.175	20.4666	40.6
T_4	1.64375	0.73125	5.39375
Average	14.77	21.84	286.66
	psFCM	**Neural network**	**FHCA**
D_1	8.19375	37.428	86.765
D_2	23.9066	98.333	231.800
D_3	26.595	195.225	510.816
D_4	6.855	10.502	136.141
D_5	18.57375	18.025	213.271
D_6	19.6225	22.487	323.645
T_1	569.0625	3447.375	10,090.625
T_2	658.7	3324.2	9368.4
T_3	156.33	1032.25	2428.916
T_4	4.9125	19.912	60.181
Average	149.27	820.57	2345.0564

4.3 The Comparison of the Computational Time for Finding Representatives

In Table 7, we describe the number of representatives and the related computational time of algorithms. The results show that GB-FCM finds representatives faster than DB-FCM and other algorithms. However, the number of representatives produced by GB-FCM is larger than that by DB-FCM.

Now, we illustrate the dataset D_1 (Fig. 16), the representatives and their centers produced by GB-FCM (Fig. 17) and DB-FCM (Fig. 18).

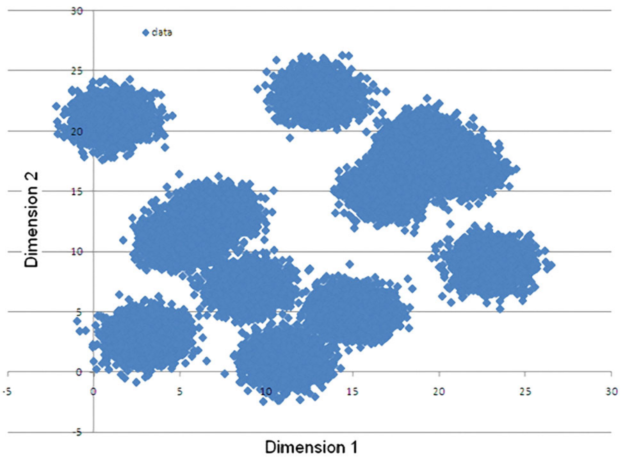


Fig. 16 Dataset D_1

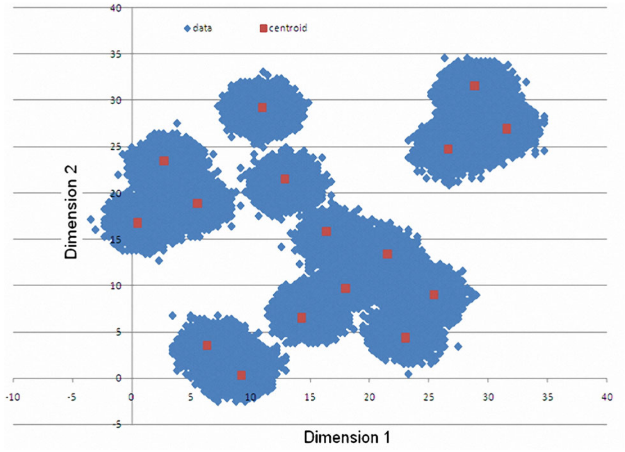


Fig. 17 Clustering results of GB-FCM with centers marked in red

Table 7 Number of representatives (computational time in seconds) of algorithms

Dataset	GB-FCM	DB-FCM	psFCM	FHCA
D_1	3632 (1.75)	516 (2.1)	6480 (29.4)	6613 (437)
D_2	4845 (5.08)	651 (4.36)	8300 (38.1)	9239 (971)
D_3	4923 (9.1)	684 (9.5)	10,700 (42.7)	11,106 (2034)
D_4	5582 (2.54)	1938 (10.2)	7500 (27.61)	7180 (512)
D_5	6695 (4.02)	2043 (16.4)	7848 (32.9)	9100 (702)
D_6	7482 (15.7)	2481 (48.27)	8165 (35.23)	9176 (858)
T_1	86,945 (172.9)	32,586 (346.8)	124,636 (675.7)	157,389 (11,083)
T_2	984,561 (175.3)	288,786 (278)	1,549,722 (1100)	1,999,230 (28,493)
T_3	5121 (41.2)	2002 (89.6)	7611 (828)	7594 (11,175)
T_4	1354 (7.3)	896 (4.5)	2736 (21.7)	2915 (358)

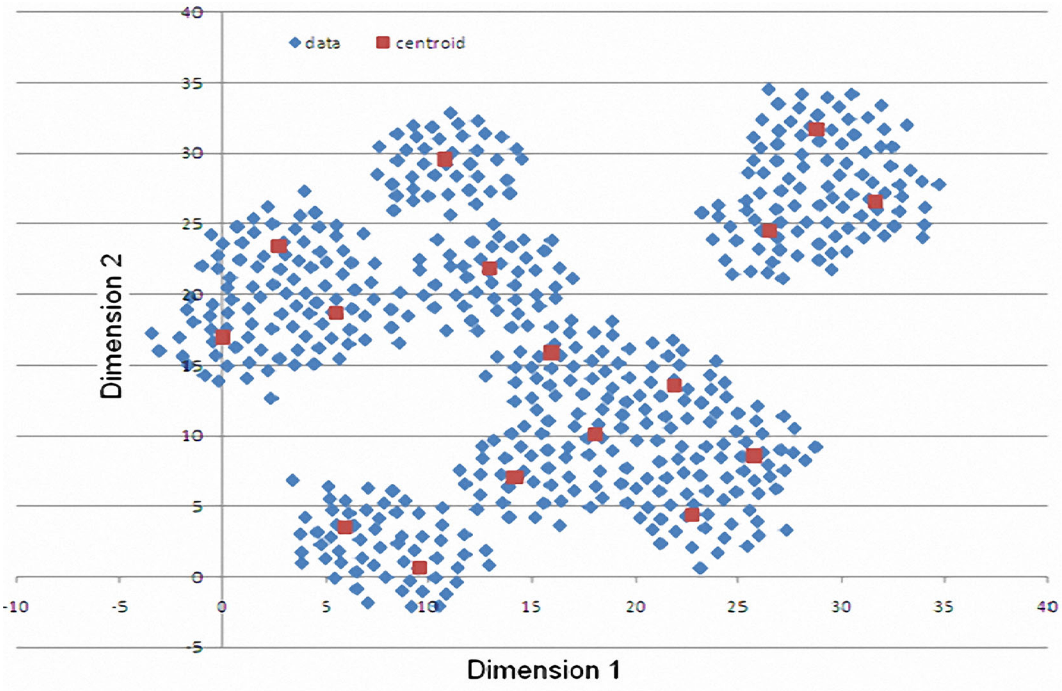


Fig. 18 Clustering results of DB-FCM with centers marked in red

4.4 The Comparison of Clustering Quality

Table 8 measures the DB values of algorithms. The results reveal that the DB values of the proposed algorithms are approximate to that of FCM and mostly smaller than those of the relevant works. Moreover, the statement in Theorem 5 affirming that the clustering quality of DB-FCM is better than that of GB-FCM has been verified.

4.5 Summary of the Findings

It has been observed from the experimental results in Sections 4.2–4.4 that *clustering qualities* of the proposed works (GB-FCM and DB-FCM) are approximate to that of FCM and mostly better than those of other algorithms. It is understandable because FCM is the original clustering algorithm while others are the approximate methods, which were created to handle the problem of processing large datasets. According to Fig. 19 where the average DB

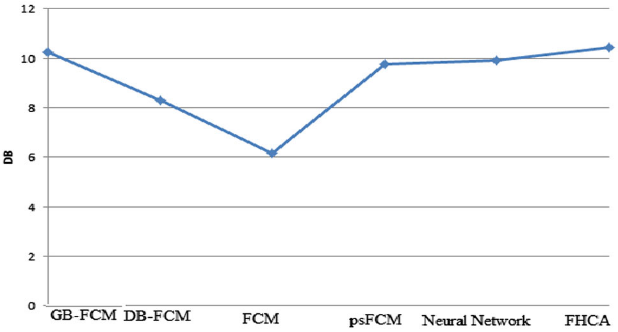


Fig. 19 Average DB values of all algorithms

Table 8 DB values of algorithms

Data	GB-FCM	DB-FCM	FCM	psFCM	Neural network	FHCA
D_1	9.56	8.42	5.65	8.42	10.8	14.1
D_2	9.17	6.74	4.38	7.37	7.22	10.3
D_3	9.83	7.28	4.63	5.26	9.45	8.18
D_4	19.6	16.1	12.3	18.4	15.8	18.7
D_5	9.03	8.77	7.22	12.5	13.2	11.4
D_6	5.82	4.16	3.75	7.34	7.38	8.29
T_1	6.28	6.21	3.26	6.88	6.28	5.72
T_2	9.32	8.96	7.89	9.46	9.26	9.35
T_3	15.7	11.1	9.22	14.3	13.1	12.9
T_4	8.36	5.27	3.48	7.63	6.48	5.28

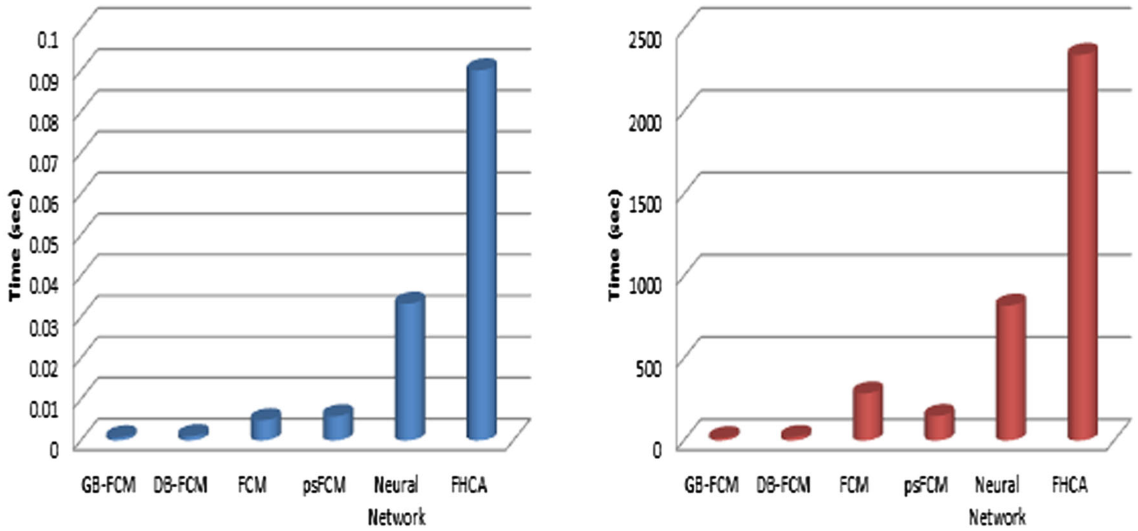


Fig. 20 Average computational time of all algorithms per (a) data points; (b) clusters

values of all algorithms are illustrated, we realize that the clustering quality of DB-FCM is better than psFCM, neural network and FHCA. GB-FCM is approximate to these algorithms in terms of clustering quality. The clustering quality of DB-FCM is better than those of other algorithms because it creates good representations through the process of forming and merging 'weak' clusters into 'strong' ones. Therefore, cluster centers are nearly identical to the optimal results. In GB-FCM, the representatives are fixed into meshes of the grid, which somehow do not reflect the nature of dataset, thus making the limitation of GB-FCM regarding clustering quality as compared with DB-FCM. Thus, when selecting an approximation algorithm that has good clustering quality, DB-FCM is the first choice.

In terms of computational time (which is the main issue of approximation algorithms), GB-FCM shows the superiority versus other algorithms: It is faster than the relevant ones by various types of data. As illustrated in Fig. 20 where the average computational time of all algorithms per (a) data points and (b) clusters is described, two proposed algorithms are faster than others due to the idea of hybrid mechanism between incremental clustering and initial selection mentioned in the Introduction. That is to say, we have appropriate methods to determine the representatives and initial centers in GB-FCM and DB-FCM. Therefore, clustering in the representative sets obtains both fast computational speed and reasonable quality in comparison with the clustering in the entire dataset. Moreover, since the closeness of the final centers with the initial ones, those algorithms would converge to the final results faster than other methods. The results in Tables 5, 6 and Fig. 20 have clearly demonstrated this fact. It should be noted that when selecting an approximation algorithm that has good computing speed, GB-FCM is the first choice.

Last but not least, the verification on the representatives showed that GB-FCM finds representatives faster than other algorithms and the number of representatives of DB-FCM is smaller than those of other algorithms. This remark should be noted when finding an algorithm that has both fast computing speed and low memory space of data storage. In some cases when the memory of storage is limited, DB-FCM is a good choice to use.

We have analyzed the reason why the proposed algorithms perform better than the others in terms of clustering quality, computation time and the representatives. Also, their advantages and disadvantages have been identified to make them feasible in practical applications.

5 Conclusions

In this paper, we proposed two novel hybrid clustering algorithms namely GB-FCM and DB-FCM based on incremental clustering and initial selection to tune up FCM for the Big Data problem. Details of the algorithms including a series of theorems were described. The theoretical contributions of the new algorithms are: i) the equivalence of clustering in the representative sets to that in the entire set; ii) difference of solutions of the representative sets vs. those of the entire set that demonstrates clustering quality of the new algorithms; iii) the definition of quality; iv) the estimation of clustering quality of the new methods; v) the half-sphere representative. We also proved that the quality of solutions when clustering the representatives set is approximate to that of clustering the original dataset. Such analyses would help explaining the algorithms better. The proposed algorithms were verified experimentally on both simulated and real datasets. The results showed that the new algorithms run faster than other relevant methods. Analyses about the clustering qualities of algorithms and representatives were performed accordingly. Further researches on this theme will extend the half-sphere representative method to the dimension greater than two. Moreover, a more exact bound for the density-based method will be considered.

Acknowledgments The authors are greatly indebted to the editor-in-chief, Prof. Shun-Feng Su and anonymous reviewers for their comments and their valuable suggestions that improved the quality and clarity of paper. A great thank was dedicated to Msc. Nguyen Duc Thien for his discussion and supports in theoretical validation of this paper. We acknowledge the Center for High Performance Computing, VNU for running the codes in the IBM 1350 system.

References

1. Aaron, B., Tamir, D., Rishe, N., Kandel, A.: Dynamic incremental fuzzy C-means clustering. In: 6th International Conferences on Pervasive Patterns and Applications (PATTERNS 2014), pp. 28–37 (2014)
2. Anderson, D.T., Luke, R.H., Keller, J.M.: Speedup of fuzzy clustering through stream processing on graphics processing units. IEEE Trans. Fuzzy Syst. **16**(4), 1101–1106 (2008)
3. Arora S., Chana, I.: A survey of clustering techniques for big data analysis. In: 2014 5th IEEE International Conference on the Next Generation Information Technology Summit (Confluence), pp. 59–65 (2014)
4. Bezdek, J.C., Ehrlich, R., Full, W.: FCM: the fuzzy c-means clustering algorithm. Comput. Geosci. **10**(2), 191–203 (1984)
5. Borgelt, C., Kruse R.: Speeding up fuzzy clustering with neural network techniques. In: Proceeding of the 12th IEEE International Conference on Fuzzy Systems (FUZZ '03), St. Louis, Missouri, USA, Vol. 2, pp. 852–856 (2003)
6. Cheng, T.W., Goldgof, D.B., Hall, L.O.: Fast fuzzy clustering. Fuzzy Sets Syst. **93**(1), 49–56 (1998)
7. Cuong, B.C., Son, L.H., Chau, H.T.M.: Some context fuzzy clustering methods for classification problems. In: Proceedings of the 2010 Symposium on Information and Communication Technology, Hanoi, Vietnam, pp. 34–40 (2010)
8. Davies, D.L., Bouldin, D.W.: A cluster separation measure. IEEE Trans. Patt Anal. Mach. Intell. **2**, 224–227 (1979)
9. Dong, Y., Zhuang, Y.: Fuzzy Hierarchical clustering algorithm facing large databases. In: Proceeding of the 5th IEEE World Congress on Intelligent Control and Automation, Hangzhou, China, Vol. 5, pp. 4282–4286 (2004)
10. Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A.Y., Foufou, S., Bouras, A.: A survey of clustering algorithms for big data: taxonomy & empirical analysis. IEEE Trans. Emerg. Top. Comput. **2**(3), 267–279 (2014)
11. Fan, J., Li, J.: A fixed suppressed rate selection method for suppressed fuzzy c-means clustering algorithm. Appl. Math. **5**, 1275–1283 (2014)
12. Feng, X.B., Yao, F., Li, Z.G., Yang, X.J.: Improved fuzzy C-means based on the optimal number of clusters. Appl. Mech. Mater. **392**, 803–807 (2013)
13. Gobi, A.F., Pedrycz, W.: The potential of fuzzy neural networks in the realization of approximation reasoning engines. Fuzzy Sets Syst. **157**(22), 2954–2973 (2006)
14. Hall, L.O.: Exploring big data with scalable soft clustering. In: Synergies of Soft Computing and Statistics for Intelligent Data Analysis, pp. 11–15. Springer, Berlin (2013)
15. Hu, Y., Qu, F., Wen, C.: An unsupervised possibilistic c-means clustering algorithm with data reduction. In: 10th IEEE International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2013), pp. 29–33 (2013)
16. Hung, M. C., Yang, D.L. An efficient Fuzzy C-means clustering algorithm. In: Proceedings of the IEEE International Conference on Data Mining 2001 (ICDM 2001), San Jose, CA, USA, pp. 225–232 (2001)
17. Jain, A.K.: Data clustering: 50 years beyond K-means. Pattern Recogn. Lett. **31**(8), 651–666 (2010)
18. Kothari, D., Narayanan, S.T., Devi, K.K.: Extended fuzzy C-means with random sampling techniques for clustering large data. Int. J. Innov. Res. Adv. Eng. **1**(1), 1–4 (2014)
19. Levy, R.: Probabilistic models in the study of language, Ms. University of California, San Diego (2010)
20. Marsaglia, G.: Random variables and computers. In: Information Theory Statistical Decision Functions Random Process, pp. 499–510 (1962)
21. Ozturk, C., Hancer, E., Karaboga, D.: Improved clustering criterion for image clustering with artificial bee colony algorithm. Pattern Anal. Appl. **18**(3), 587–599 (2015)
22. Parker, J.K., Hall, L.O.: Accelerating fuzzy-c means using an estimated subsample size. IEEE Trans. Fuzzy Syst. **22**(5), 1229–1244 (2014)
23. Parvin, H., Minaei-Bidgoli, B.: A clustering ensemble framework based on selection of fuzzy weighted clusters in a locally adaptive clustering algorithm. Pattern Anal. Appl. **18**(1), 87–112 (2015)
24. Qu, F., Hu, Y., Xue, Y., Yang, Y.: A modified possibilistic fuzzy c-means clustering algorithm. In: 2013 IEEE 9th International Conference on Natural Computation (ICNC 2013), pp. 858–862 (2013)
25. Rahimi S., Zargham M., Thakre A., Chhillar D.: A parallel Fuzzy C-Mean algorithm for image segmentation. In: Proceeding of the IEEE Annual Meeting of the Fuzzy Information Processing Society (NAFIPS '04), Vol. 1, pp. 234–237 (2004)
26. Ramathilagam, S., Devi, R., Kannan, S.R.: Extended fuzzy c-means: an analyzing data clustering problems. Cluster Comput. **16**(3), 389–406 (2013)
27. Sarma, T.H., Viswanath, P., Reddy, B.E.: Speeding-up the kernel k-means clustering method: a prototype based hybrid approach. Pattern Recogn. Lett. **34**(5), 564–573 (2013)
28. Shirkhorshidi, A.S., Aghabozorgi, S., Wah, T.Y., Herawan, T.: Big data clustering: a review. In: Computational Science and its Applications–ICCSA 2014 (pp. 707–720). Springer International Publishing (2014)
29. Son, L.H., Cuong, B.C., Lanzi, P.L., Thong, N.T.: A novel intuitionistic fuzzy clustering method for geo-demographic analysis. Expert Syst. Appl. **39**(10), 9848–9859 (2012)
30. Son, L.H., Cuong, B.C., Long, H.V.: Spatial interaction—modification model and applications to geo-demographic analysis. Knowl. Based Syst. **49**, 152–170 (2013)
31. Son, L.H., Lanzi, P.L., Cuong, B.C., Hung, H.A.: Data mining in GIS: A novel context-based fuzzy geographically weighted clustering algorithm. Int. J. Mach. Learn. Comput. **2**(3), 235–238 (2012)
32. Son, L.H.: Enhancing clustering quality of geo-demographic analysis using context fuzzy clustering type-2 and particle swarm optimization. Appl. Soft Comput. **22**, 566–584 (2014)
33. Son, L.H.: HU-FCF: a hybrid user-based fuzzy collaborative filtering method in recommender systems. Expert Syst. Appl. **41**(15), 6861–6870 (2014)
34. Son, L.H.: Optimizing municipal solid waste collection using chaotic particle swarm optimization in GIS based environments: a case study at Danang City, Vietnam. Expert Syst. Appl. **41**(18), 8062–8074 (2014)

35. Son, L.H.: DPFCM: A novel distributed picture fuzzy clustering method on picture fuzzy sets. Expert Syst. Appl. **42**(1), 51–66 (2015)
36. Son, L.H.: Dealing with the new user cold-start problem in recommender systems: a comparative review. Inform. Syst. **58**, 87–104 (2015)
37. Son, L.H.: HU-FCF++: a novel hybrid method for the new user cold-start problem in recommender systems. Eng. Appl. Artif. Intell. **41**, 207–222 (2015)
38. Son, L.H., Linh, N.D., Long, H.V.: A lossless DEM compression for fast retrieval method using fuzzy clustering and MANFIS neural network. Eng. Appl. Artif. Intell. **29**, 33–42 (2014)
39. Son, L.H., Thong, N.T.: Intuitionistic fuzzy recommender systems: an effective tool for medical diagnosis. Knowl.-Based Syst. **74**, 133–150 (2015)
40. Szilágyi, L., Szilágyi, S.M.: Generalization rules for the suppressed fuzzy c-means clustering algorithm. Neurocomputing **139**, 298–309 (2014)
41. Szilagyi, L., Denesi, G., Szilagyi, S.M.: Fast color reduction using approximative c-means clustering models. In: 2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 14'), pp. 194–201 (2014)
42. Taherdangkoo, M., Bagheri, M.H.: A powerful hybrid clustering method based on modified stem cells and Fuzzy C-means algorithms. Eng. Appl. Artif. Intell. **26**(5), 1493–1502 (2013)
43. Thong, N.T., Son, L.H.: HIFCF: an effective hybrid model between picture fuzzy clustering and intuitionistic fuzzy recommender systems for medical diagnosis. Expert Syst. Appl. **42**(7), 3682–3701 (2015)
44. Thong, P.H., Son, L.H.: A new approach to multi-variables fuzzy forecasting using picture fuzzy clustering and picture fuzzy rules interpolation method. In: Proceeding of 6th International Conference on Knowledge and Systems Engineering (KSE 2014), Hanoi, Vietnam, pp 679–690 (2014)
45. UCI Machine Learning Repository. (2015). *Datasets*, Available at: https://archive.ics.uci.edu/ml/datasets.html. Accessed: 11/03/2015
46. Wang, J., Chung, F.L., Wang, S., Deng, Z.: Double indices-induced FCM clustering and its integration with fuzzy subspace clustering. Pattern Anal. Appl. **17**(3), 549–566 (2014)
47. Wang, Y., Chen, L., Mei, J.P.: Incremental fuzzy clustering with multiple medoids for large data. IEEE Trans. Fuzzy Syst. **22**(6), 1557–1568 (2014)
48. Zang, X., Vista IV, F.P., Chong, K.T.: Fast global kernel fuzzy c-means clustering algorithm for consonant/vowel segmentation of speech signal. J Zhejiang Univ. Sci. C **15**(7), 551–563 (2014)
49. Zhang, Q., Chen, Z.: A weighted kernel possibilistic c-means algorithm based on cloud computing for clustering big data. Int. J. Commun Syst **27**(9), 1378–1391 (2014)
50. Zhang, Z., Havens, T.C.: Scalable approximation of kernel fuzzy c-means. In: 2013 IEEE International Conference on Big Data, pp. 161–168 (2013)
51. Zhao, Y., Wu, X., Kong, S.G., Zhang, L.: Joint segmentation and pairing of multispectral chromosome images. Pattern Anal. Appl. **16**(4), 497–506 (2013)

Dr. Le Hoang Son obtained the PhD degree on Mathematics—Informatics at VNU University of Science, Vietnam National University (VNU). He has been working as a researcher and now Vice Director of the Center for High Performance Computing, VNU University of Science, Vietnam National University since 2007. His major field includes Soft Computing, Fuzzy Clustering, Recommender Systems, Geographic Information Systems (GIS) and Particle Swarm Optimization. He is a member of International Association of Computer Science and Information Technology (IACSIT), a member of Center for Applied Research in e-Health (eCARE), a member of Vietnam Society for Applications of Mathematics (Vietsam), Editorial Board of Neutrosophic Sets and Systems (NSS), Editorial Board of International Journal of Ambient Computing and Intelligence (IJACI, SCOPUS) and associate editor of the International Journal of Engineering and Technology (IJET). Dr. Son served as a reviewer for various international journals and conferences such as PACIS 2010, ICMET 2011, ICCTD 2011, KSE 2013, BAFI 2014, NICS 2014 & 2015, ACIIDS 2015 & 2016, ICNSC15, GIS-2015, FAIR 2015, International Journal of Computer and Electrical Engineering, Imaging Science Journal, International Journal of Intelligent Systems Technologies and Applications, IEEE Transactions on Fuzzy Systems, Expert Systems with Applications, International Journal of Electrical Power and Energy Systems, Neural Computing and Applications, International Journal of Fuzzy System Applications, Intelligent Data Analysis, Computer Methods and Programs in Biomedicine, World Journal of Modeling and Simulation, Knowledge-Based Systems, Engineering Applications of Artificial Intelligence. He gave a number of invited talks at many conferences such as 2015 National Fundamental and Applied IT Research (FAIR 15′), 2015 National conference of Vietnam Society for Applications of Mathematics (VietSam15′), 2015 Conference on Developing Applications in Virtual Reality, GIS and Mobile technologies, and International Conference on Mathematical Education Vietnam 2015 (ICME Vietnam 2015), 2016 3rd National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS 16′), and 2016 HUST Conference on Applied Mathematics and Informatics (SAMI 16′). Dr. Son has got 66 publications in prestigious journals and conferences including 29 SCI/SCIE papers and undertaken more than 20 major joint international and national research projects. He has published 2 books on mobile and GIS applications. So far, he has awarded '2014 VNU Research Award for Young Scientists', '2015 VNU Annual Research Award' and '2015 Vietnamese Mathematical Award'.

Dr. Nguyen Dang Tien was born in Vietnam, in 1962. He holds the MSc from Electrical Engineering Department at Hanoi University of Science and Technology and the PhD from the Electrical Engineering Department at Le Quy Don Technical University in 2007. Now he's the Rector of People's Police University of Technology and Logistics, Bac Ninh, Vietnam. His interests span the areas of Cryptographic Algorithms, Fuzzy Clustering and Data Structure, where he has published papers in international journal and conferences. He has also co-authored one book.