Amina Dik*, Khalid Jebari and Aziz Ettouhami

An Improved Robust Fuzzy Algorithm for Unsupervised Learning

https://doi.org/10.1515/jisys-2018-0030
Received January 15, 2018; previously published online October 25, 2018.

Abstract: This paper presents a robust, dynamic, and unsupervised fuzzy learning algorithm (RDUFL) that aims to cluster a set of data samples with the ability to detect outliers and assign the numbers of clusters automatically. It consists of three main stages. The first (1) stage is a pre-processing method in which possible outliers are determined and quarantined using a concept of proximity degree. The second (2) stage is a learning method, which consists in auto-detecting the number of classes with their prototypes for a dynamic threshold. This threshold is automatically determined based on the similarity among the detected prototypes that are updated at the exploration of a new data. The last (3) stage treats quarantined samples detected from the first stage to determine whether they belong to some class defined in the second phase. The effectiveness of this method is assessed on eight real medical benchmark datasets in comparison to known unsupervised learning methods, namely, the fuzzy c-means (FCM), possibilistic c-means (PCM), and noise clustering (NC). The obtained accuracy of our scheme is very promising for unsupervised learning problems.

Keywords: Similarity measure, outlier detection, clustering, proximity degree.

1 Introduction

Clustering is one of the most relevant data-mining tasks [42]. It is the process of organizing objects into a set of classes. The classes provided by the classical methods are considered hard, and each object is assigned to a single and unique class. This assumes that the boundaries between classes are well defined, whereas, in fact, class boundaries are often fuzzy and uncertain. This uncertainty is shown by the fact that an object possesses features that make it more likely to belong to more than a single class. Thus, in the fuzzy classification, an object does not belong exclusively to a single class but possesses a degree of membership to all existing classes [36]. The degree of membership is in the interval [0, 1], and the obtained classes are not necessarily disjoint. Clustering becomes very difficult in unsupervised contexts where no prior information on the experimental objects is provided. This difficulty increases when these objects contain outliers [39]. An outlier refers to a value that appears to be suspicious because it is significantly inconsistent with the rest of that set of data. According to Han and Kamber [27], outliers are the set of objects that are considerably dissimilar from the remainder of the data.

In clustering, giving an outlier the same importance that is given to other objects destabilizes the analysis and distorts the results [3]. Hence, outlier detection is important [44, 47]. However, these outliers are not necessarily erroneous and may contain a meaningful indication [45], such as the case during fraud detection [9] or computer network intrusion [35]. Therefore, outliers ought not to be systematically rejected [5].

Handling clustering and outlier detection at the same time is a highly desirable task [33, 38, 40]. Many strong methods that have emerged in this direction take all the data into account, but minimize the influence of outliers [20, 21]. The best-known algorithm is the noise clustering (NC), also called the robust-fuzzy

***Corresponding author: Amina Dik,** Conception and Systems Laboratory, Faculty of Sciences, Mohammed V University in Rabat, Rabat 10001, Morocco, e-mail: a.dik70@yahoo.fr. https://orcid.org/0000-0001-6953-0817
Khalid Jebari: Conception and Systems Laboratory, Faculty of Sciences, Mohammed V University in Rabat, Rabat, Morocco; and Sciences and Technologies Faculty Tangier, Abdelmalek Essaâdi University, Tetuan, Morocco
Aziz Ettouhami: Conception and Systems Laboratory, Faculty of Sciences, Mohammed V University in Rabat, Rabat, Morocco

c-means (FCM) [19]. In this algorithm, the notion of noise class is introduced. The class of these outliers is characterized by a fictitious prototype that has a constant distance δ to other objects. Hence, it is important to determine the distance δ, which is a critical parameter of the algorithm [17].

In this paper, we propose a robust approach, which allows clustering data by auto-detecting the classes they form and providing the existing outliers without giving any parameter. The proposed approach consists of three stages:

- A pre-processing stage using similarity to detect objects likely to be outliers and which will be considered as possible outliers. These objects are quarantined and excluded from the second stage.
- A second stage in which classes are determined based on a dynamic threshold. This threshold is based on the minimum similarity among the detected prototypes, which are updated at the exploration of any new object. This minimum similarity is considered as the condition of adding a new cluster.
- A final stage, which is a processing of possible outliers in order to determine whether they belong to one of these classes detected in the second phase. To this end, each possible outlier is compared to its neighbors to confirm that it is really an outlier.

A more formal description of this method is presented in Section 3 following a brief related work in Section 2. The results obtained from experiments on real and artificial data are presented in Section 4. Section 5 presents the main conclusions of this paper.

2 Related Work

Clustering is commonly used in real-world problems encountered in a variety of applications [13–15, 51, 54, 58]. It is an exploratory data analysis tool, which aims to find structure in a dataset according to the measured characteristics or similarities [12, 48, 52]. It consists in grouping a set of n data points into homogeneous groups, called clusters, without any prior information on the structure or the nature of the clusters.

Clustering can be classified as hard or fuzzy. Hard clustering assigns each data point to a unique cluster with a degree of membership equal to one. Conversely, fuzzy clustering assigns each data point to every cluster with different membership degrees.

In mathematical terms, partitioning a learning base $X = \{x_1, x_2, \ldots, x_n\} \subset \Re^p$ into c clusters can be represented by a $(c \times n)$ partition matrix $U = [u_{ik}]$, which satisfies the following conditions:

$$u_{ik} \in \{0, 1\}, \quad 1 \leq k \leq c, \quad 1 \leq i \leq n \tag{1}$$

$$\sum_{k=1}^{c} u_{ik} = 1, \quad 1 \leq i \leq n \tag{2}$$

$$0 < \sum_{i=1}^{n} u_{ik} < n \quad 1 \leq k \leq c \tag{3}$$

The space of hard partitions is, thus, defined by Bezdek [6]:

$$M_{hc} = \left\{ U \in \Re^{cxn} / u_{ik} \in \{0, 1\} \ \forall \ i, k; \sum_{k=1}^{c} u_{ik} = 1 \ \forall \ i; 0 < \sum_{i=1}^{n} u_{ik} < n, \ \forall \ k \right\}$$

Hard clustering assumes that clusters are disjointed, and their boundaries are well defined. However, the boundaries between clusters are not always definite in real-world datasets. Fuzzy clustering was proposed to deal with overlapping clusters.

Partitioning X into c fuzzy clusters can be defined by c fuzzy sets $E_1, \ldots, E_c$ and a membership function [57] assuming values in the interval [0, 1] such as:

$$E_k = \{\mu_k(x_i) / x_i \in X, \ 1 \leq i \leq n\} \tag{4}$$

$$\forall\ i, k \quad \mu_k : \begin{cases} X \to [0,\ 1] \\ x_i \to\ \mu_k(x_i) = u_{ik} \end{cases} \tag{5}$$

where u_{ik} is interpreted as the membership degree to which the object i belongs to the k^{th} cluster ($1 \leq k \leq c$ and $1 \leq i \leq n$) [6, 19]. Therefore, a ($c \times n$) fuzzy membership matrix $U = [u_{ik}]$ can be used to represent the fuzzy partition of X. The k^{th} row of this matrix contains values of the k^{th} membership function μ_k of the subset E_k. Elements u_{ik} satisfy the following conditions:

$$0 \leq u_{ik} \leq 1;\ 1 \leq k \leq c; 1 \leq i \leq n \tag{6}$$

$$0 < \sum_{i=1}^{n} u_{ik} < n \quad 1 \leq k \leq c \tag{7}$$

Thereby, fuzzy clustering is considered as a generalization of hard clustering that can be used to describe imprecise or fuzzy information [30, 50, 53]. The most widely used clustering algorithm is FCM [15], which is highlighted in what follows:

2.1 The FCM Algorithm

FCM generalizes the hard c-means algorithm [K-means] to allow a point to partially belong to all existing clusters [6, 7]. FCM is an iterative process, which optimizes an objective function J_m defined by:

$$J_m(U, V; X) = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})^m d^2(x_k, v_i) \tag{8}$$

where:
- u_{ik} is the degree to which the element x_k belongs to the i^{th} class ($1 \leq i \leq c, 1 \leq k \leq n$).
- m ($1 < m < \infty$) is an exponent of the weighting used to monitor the relative contribution of each object x_i and the fuzziness degree of the final partition.
- $V = (v_1, v_2, \ldots, v_c)$ represents a c triplet of prototypes, in which each prototype characterizes a class.
- $d(x_k, v_i)$ is the distance between the i^{th} prototype and the k^{th} object.

Bezdek demonstrated that FCM converges to an approximate solution when two conditions are satisfied [10]:

$$u_{ik} = \left[\sum_{j=1}^{c} \left(\frac{d(x_k, v_i)}{d(x_k, v_j)} \right)^{2/m-1} \right]^{-1};\ 1 \leq i \leq c;\ 1 \leq k \leq n \tag{9}$$

$$v_i = \frac{\sum_{k=1}^{n} (u_{ik})^m\ x_k}{\sum_{k=1}^{n} (u_{ik})^m};\ 1 \leq i \leq c \tag{10}$$

The pseudo-code of the FCM algorithm is given in Figure 1.

The idea of clustering data is natural. Indeed, we tend to group a large number of data into a small number of groups in order to facilitate further analysis. The search for these groups is not a simple task when data is affected by outliers. Generally, outliers are far from all the other items without neighbors. Outliers may significantly affect the estimation of the centers of detected clusters, which is the case for the FCM algorithm. Two methods were proposed to handle this problem: the possibilistic C-means algorithm (PCM) [34] and robust-FCM [19]. These are summarized in what follows.

Store unlabeled Dataset $X = \{x_1, x_2, \ldots, x_n\} \subset \Re^p$;

Input parameters

- $1 < c < n$ // number of clusters
- $m > 1$ // weight exponential
- t // iteration number
- t_{max} // iteration maximal
- ε // given tolerance;
- norm for clustering criterion J_m (e.g. Euclidean distance):

$$J_m(U, V; X) = \sum_{k=1}^{n}\sum_{i=1}^{c}(u_{ik})^m d^2(x_k, v_i) + \sum_{i=1}^{c}\eta_i\sum_{k=1}^{n}(1-u_{ik})^m;$$

- norm for termination error $E_t = \|V_t - V_{t-1}\|_{err}$;

Initialize

- prototypes $V_0 = (v_{1,0}, v_{2,0}, \ldots, v_{c,0}) \in \Re^{cxp}$
- t = 0

do {

- t++
- Calculate U_t = membership matrix at the iteration t using V_{t-1} and

$$u_{ik} = \left[\sum_{j=1}^{c}\left(\frac{d(x_k, v_i)}{d(x_k, v_j)}\right)^{2/m-1}\right]^{-1}; 1 \le i \le c; 1 \le k \le n$$

- Calculate V_t = prototypes matrix at the iteration t using U_t and

$$v_i = \frac{\sum_{k=1}^{n}(u_{ik})^m x_k}{\sum_{k=1}^{n}(u_{ik})^m}; 1 \le i \le c$$

} ***while*** (($\|V_t - V_{t-1}\|_{err} > \varepsilon$) and ($t < t_{max}$));

$U^* = U_t$; $V^* = V_t$;

Use U^* and V^*;

Figure 1: FCM Algorithm.

2.2 The PCM Algorithm

The PCM introduces a possibilistic type of membership function to describe the degree of belonging [56] and releases the objective function J_m [Eq. (8)] by dropping the sum to 1 [Eq. (2)]. Hence, membership degrees became independent [41].

The PCM optimizes the objective J_m defined as follows:

$$J_m(U, V; X) = \sum_{k=1}^{n}\sum_{i=1}^{c}(u_{ik})^m d^2(x_k, v_i) + \sum_{i=1}^{c}\eta_i\sum_{k=1}^{n}(1-u_{ik})^m \tag{11}$$

where:

- η_i $(1 < I < c)$ is a parameter defined by:

$$\eta_i = K\frac{\sum_{k=1}^{n}(u_{ik})^m d^2(x_k, v_i)}{\sum_{k=1}^{n}(u_{ik})^m}, \quad K > 0 \tag{12}$$

and u_{ik} is defined by:

$$u_{ik} = \left[1 + \left(\frac{d^2(x_k, v_i)}{\eta_i}\right)^{1/m-1}\right]^{-1}; 1 \le i \le c; 1 \le k \le n \tag{13}$$

The parameter η_i is a positive weight defined for modulating the opposing effects of the two terms in $\boldsymbol{J_m}$. It is set by the user and chosen according to each class. Unfortunately, it is not always available.

2.3 The Robust-FCM Algorithm

Dave [19] proposed a new method known as the «*robust-FCM*». It consists in introducing an additional class of noise that contains all the outliers. The fictive prototype of the noise cluster is set such that it is always at the same distance from the considered data points. This distance is called the noise distance δ. Thus, an object is not an outlier if its distance from one of the prototypes is inferior to δ.

The presence of the noise cluster modifies the objective function defined by Eq. (8) as follows:

$$J_m(U, V; X) = \sum_{k=1}^{n}\sum_{i=1}^{c}(u_{ik})^m d^2(x_k, v_i) + \sum_{k=1}^{n} u_{\star k}^m.\delta^2 \tag{14}$$

where $u_{\star k}$ is the membership degree of the object x_k to the noise cluster, and δ is the distance of noise defined, respectively, by:

$$u_{\star k} = 1 - \sum_{i=1}^{c} u_{ik} \tag{15}$$

$$\delta^2 = \lambda \frac{\sum_{k=1}^{n}\sum_{i=1}^{c} d(x_k, v_i)}{n\,(c-1)} \tag{16}$$

By minimizing the objective function defined by Eq. (14), we obtain:

$$u_{ik} = \frac{1}{\sum_{j=1}^{c}\left[\frac{d(x_k,v_i)}{d(x_k,v_j)}\right]^{\frac{2}{m-1}} + \left[\frac{d(x_k,v_i)}{\delta}\right]^{\frac{2}{m-1}}} \tag{17}$$

The variable λ is a multiplying factor for obtaining δ. Thus, the choice of δ depends on λ. Dave proposed a heuristic to select this parameter. However, this choice does not always give satisfaction.

Recently, some approaches were proposed on outlier detection [33, 43], and the outliers, themselves, become the "focus" in outlier-mining tasks [16].

3 Proposed Approach (RDUFL)

Handling simultaneously clustering data and detecting outliers, as mentioned earlier, is a highly desirable task. The intuitive approach consists in applying a clustering algorithm and considering objects that are distant from their nearest prototype as outliers. However, this algorithm may, itself, be extremely sensitive to the outliers that may have a disproportionate impact on prototypes [26]. Hence, detecting outliers is important in clustering tasks.

RDUFL allows to cluster the considered data and to detect eventual outliers. RDUFL consists of the following three phases:

3.1 Detection of Possible Outliers

The first phase of our approach consists in detecting objects that are likely to be outliers and which we will refer to as «*possible*» outliers. It originates from the fact that a normal object has more neighbors with which it shares similar characteristics [11, 27]. It is based on the proximity degree of an object in relation to

other objects. This concept consists in calculating the sum of similarities of an object to all other objects [22] and not just to its neighbors [1]:

$$D(x_i) = \left(\sum_{\substack{j=0 \\ j \neq i}}^{n} sim\,(x_i\,, x_j) \right) \tag{18}$$

with:

$$Sim(x_i,\ x_k) = 1 - \frac{\|x_i - x_k\|_A}{\sqrt{p}} \tag{19}$$

where:
- $Sim\,(x_i, x_k)$ is the similarity between the two objects x_i and x_k [25].
- p is the dimension of the objects space: $x_\mathrm{i} = (x_{i1}, x_{i2}, \ldots, x_{ip}) \in \Re^p$.
- A is the $p \times p$ matrix defined by Bouroumi [10]:

$$A_{jt} = \begin{cases} (r_j)^{-2}, j = t \\ 0,\ \text{otherwise} \end{cases} \tag{20}$$

The factor r_j stands for the difference between the maximum and minimal values of an attribute. It is defined by:

$$r_j = \max_{1 \leq i \leq n} \{x_{ij}\} - \min_{1 \leq i \leq n} \{x_{ij}\},\ 1 \leq j \leq p \tag{21}$$

The choice of this measure of similarity is motivated by the following properties [10, 25, 26]:

(i) $Sim\,(x_i, x_k) \in [0, 1]$; $\forall\, x_i, x_k \in \Re^p$ since $(x_{ij} - x_{kj}) \leq r_j\ \forall\ 1 \leq j \leq p$ and $\|x_i - x_k\|_A = \sqrt{\sum_{j=1}^{p} \frac{(x_{ij} - x_{kj})^2}{r_j^2}} \leq \sqrt{p}$

(ii) $Sim\,(x_i, x_k) = 1$ for $x_i = x_k$

(iii) $Sim\,(x_i, x_k) \longrightarrow 0$ for $(x_{ij} - x_{kj}) \longrightarrow r_j\ \forall\, 1 \leq j \leq p$, which means that objects present a maximum of difference of each of their p components.

Therefore, an object has a high degree of proximity when its neighbors are several, and the object with a low value «*D*» is more likely to be an outlier. It is considered as «possible» outlier.

This phase does not require any notion of clusters or expected number of outliers [29]. It allows determining the top objects within the small proximity degree. We note *M* as their number. Once these objects are detected, they are quarantined, and we proceed to the learning of the set *X* without taking into account these *M* possible outliers.

3.2 Learning Phase

Assuming that object vectors, which form the training database *X*, belong at least to two distinct classes and given an inter-point similarity measure, the learning algorithm of this phase starts by the creation of two classes around two first objects x_1 and x_2 [4].

RDUFL sequentially explores all the «n-2-M» objects of the training base X and analyzes their resemblances by utilizing the measure of similarity given by Eq. (19). A dynamic threshold ξ is utilized to detect when a new object is dissimilar to all existing prototypes. This threshold represents the minimum similarity that each object must have with its nearest prototype [4, 10, 23]. When this threshold is not attained, a new class is created, and its prototype is initialized with the current object.

In this paper, ξ is dynamic and depends on the current object. It is automatically calculated at each iteration as follows:

If x_i is the current object and v_k its nearest prototype, ξ is defined by:

$$\xi = \underset{\substack{1\le j\le c\\ j\neq k}}{Min}\left[\frac{Sim\,(v_j, v_k)}{2}\right] \tag{22}$$

The algorithm utilizes the similarity measure and its associated threshold in order to build classes. At each iteration, the similarity of the current object to the existing prototypes is calculated. Following the maximum of this similarity, two decisions will be conceivable [10]:

(a) A new class is created when:

$$\underset{1\le k\le c}{Max}\,(Sim\,(x_i, x_k)) < \xi \tag{23}$$

This means that the current item x_i is not sufficiently similar to the prototypes of the previously detected classes. It is supposed to come from a class that has not been detected yet and must, therefore, represent a new class [10]. Thus, we put:

$$c = c + 1 \text{ and } v_c = x_i.$$

(b) The prototypes are updated when:

$$\underset{1\le k\le c}{Max}\,(Sim\,(x_i, x_k)) \geq \xi \tag{24}$$

In this case, x_i is considered to have the minimal similarity required with the prototypes of the previously detected classes. Therefore, we must not create any new classes.

The prototypes of the previously created classes are then updated according to the following learning rule:

$$v_k(i) = v_k(i-1) + \frac{Sim\,(x_i, v_k)}{n_k(i)}\left[x_i - v_k(i-1)\right] \quad 1 \le k \le c, \quad c \ge 2 \tag{25}$$

where:

$v_k(i)$, $v_k(i-1)$ are, respectively, the prototypes of the class k before and after the addition of x_i.

$n_i(k)$ designates the fuzzy cardinal of the class k after the addition of x_i, defined by:

$$n_i(k) = \sum_{j=1}^{k} Sim\,(x_i, v_k) \qquad 1 \le k \le c, \quad i \le n \tag{26}$$

Failure to consider the possible outliers during this phase allows the stabilization of prototype calculation and the no distortion of the automatic detection of the number of classes.

3.3 Treatment of Possible Outliers

In this phase, we deal with the M possible outliers $O_i (1 \le i \le M)$ that have been detected and quarantined during the first phase. For each possible outlier O_i, we look for its nearest prototype v_k and its corresponding class marked C_k. For this class C_k, the farthest element x_1 is determined:

$$x_1 = \underset{x_j\in C_k}{Max}\left(d(x_j, v_k)\right) \tag{27}$$

If the point x_1 is closer to the outlier O_i than to its nearest prototype v_k, the point O_i has neighbors that are neighbors of x_1. This point O_i is not, therefore, an outlier. Hence, two cases are possible:

- If $Sim\,(O_i, x_l) < Sim\,(x_l, v_k)$: O_i is a true outlier.
- If $Sim\,(O_i, x_l) \geq Sim\,(x_l, v_k)$: O_i is not an outlier.

The pseudo-code of RDUFL is presented in Figure 2.

Let the dataset $X = \{x_1, x_2, \ldots, x_n\} \subset \Re^p$;

Step 1: Determine the possible M outliers O_i.

1 – *For* $i = 1$ to n calculate $D(x_i)$ defined by:

$$D(x_i) = \left(\sum_{\substack{j=0 \\ j \neq i}}^{n} sim\,(x_i, x_j) \right)$$

2 – Search for the minimum values of $D(x_i)$;

3 – If the variance is bigger, then the corresponding point is an Outlier.

Step 2: Partition the set X after removing temporarily the points O_i.

1 – Choose a similarity measure $Sim\,(x_i, x_k) = 1 - \frac{\|x_i - x_k\|_A}{\sqrt{p}}$;

2 – Initialise $c = 2$, prototypes $v_1 = x_1$, $v_2 = x_2$

3 – *For* $i = 2$ to n

{ Determine the nearest prototype v_k to the object x_i

If $\left(\underset{1 \leq j \leq c}{\text{Max}}\,(Sim\,(x_i, x_j)) < \underset{\substack{1 \leq j \leq c \\ j \neq k}}{\text{Min}} \left[\frac{Sim\,(v_j, v_k)}{2} \right] \right) \{c{+}{+}; v_i = x_i\}$

Otherwise {update the prototypes v_j, $1 \leq j \leq c$ (Eq. 28) }}

Step 3: *For* $i = 1$ to M {

1 – Determine the nearest prototype v_k to the objet O_i (C_k is its corresponding class)

2 – Determine the point x_l of the class C_k that is the farthest of the prototype v_k

3 – If ($Sim\,(O_i, x_l) < Sim\,(x_l, v_k)$) Then O_i is an outlier.
Otherwise reassign O_i to the class C_k

}

Figure 2: Pseudo-Code of RDUFL.

4 Results and Discussion

To assess the performance of our approach, some experiments were conducted on an artificial dataset X_1, and on eight real-world databases that are available in UCI [8]: Lymphography, Diabetes, Indian, Haberman's Survival, BCW, Post-operative Patient, Parkinsons, and EEG Eyes State.

A first comparison is based on the recognition rate defined by:

$$\text{Recognition rate} = 100 * \frac{\text{Number of correctly identified objects}}{\text{Total number of objects}} \tag{28}$$

A second comparison is based on:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{29}$$

where:

- TP (true positive) is the number of objects correctly identified
- FP (false positive) is the number of objects incorrectly identified
- TN (true negative) is the number of objects correctly rejected
- FN (false negative) is the number of objects incorrectly rejected.

4.1 Artificial Dataset

To illustrate the usefulness of the proposed algorithm, we consider the dataset X_1, which is a two-dimensional artificial example derived from Ref. [10]. It is divided into three classes with 58 points in the plan and seven

outliers (Figure 3). This two-dimensional dataset is important due to the possibility it presents in terms of visualization.

For the dataset X_1, RDUFL detected seven possible outliers. The learning and treatment phases of the outliers demonstrated that these are true outliers. The number of detected classes is three, and the recognition rate is 100% (Figure 4).

On the other hand, we applied on X_1 the following algorithms: FCM, PCM, and robust-FCM.

First, the FCM algorithm failed to detect the existing outliers for $c = 3$ (Figure 5A). This algorithm detected two outliers for $c = 4$ (Figure 5B) and three outliers for $c = 5$ (Figure 5C). It is only for $c = 6$ (Figure 5D) that the FCM detected all the seven outliers by considering them as points that belong to two clusters with a weak cardinal.

As for the robust-FCM algorithm intended to detect the outliers, it only detects two outliers (Figure 6). In addition, the obtained recognition rate for the robust-FCM equals 92.31%, whereas our approach was able to recognize all the objects.

For the tests we carried out, the value of λ ranged between 0.01 and 0.9, and the best result is obtained for $\lambda = 0.7$.

Table 1 presents the recognition rate of learning through the considered algorithms compared to RDUFL.

These results show the sensitivity of these algorithms toward the outliers and their difficulties in correctly extracting the classes [31, 32].

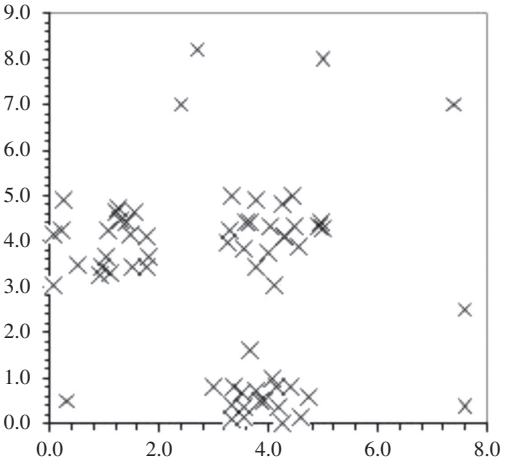

Figure 3: Representation of X_1 (65 Points).

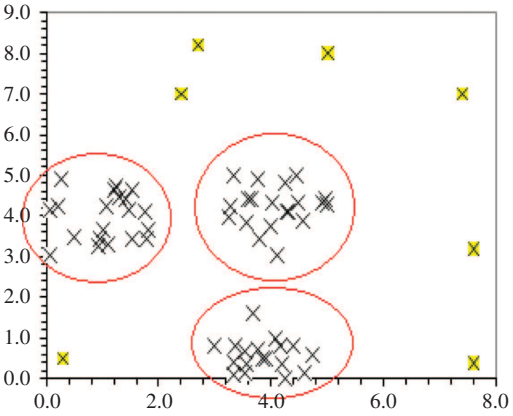

Figure 4: Results of RDUFL on the Dataset X_1.
The number of detected classes is $c = 3$; the outliers are indicated in yellow.

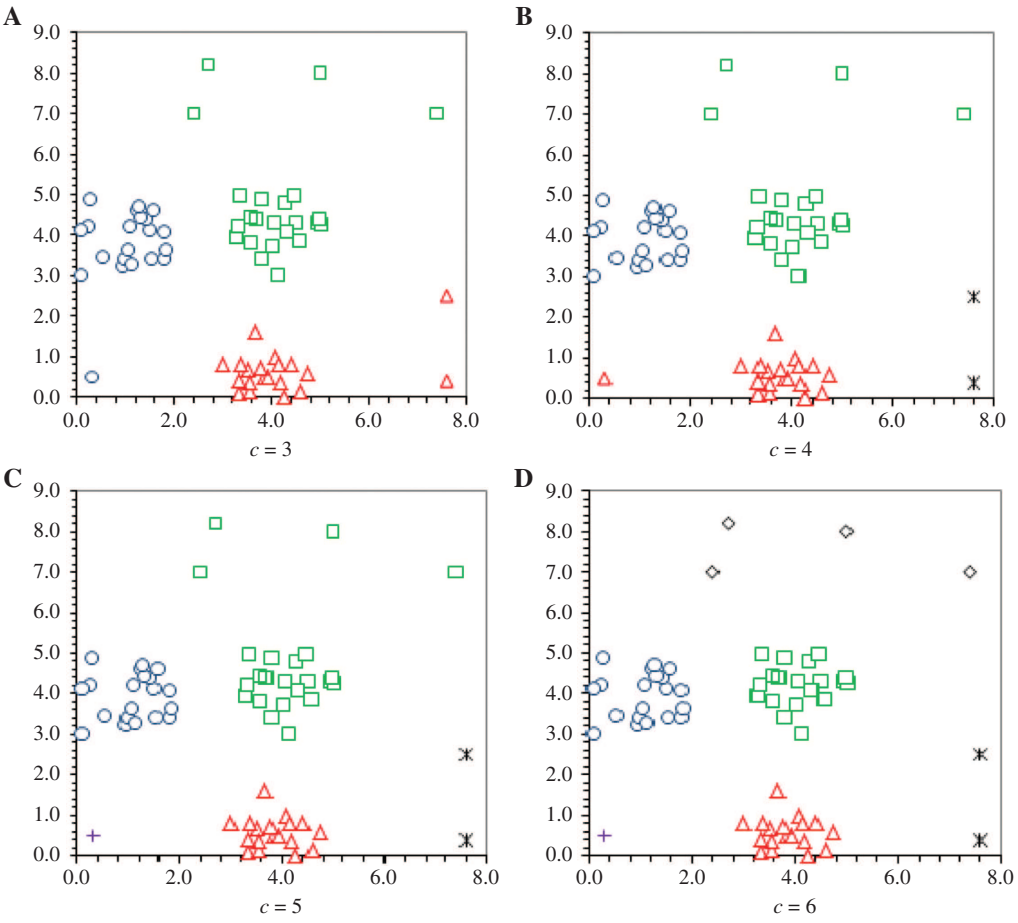

Figure 5: Clustering of X_1 by FCM and Different Values of c.
(A) $c = 3$, Outliers are not detected and belong to existing classes. (B) $c = 4$, FCM detects only two outliers, represented by a star. (C) $c = 5$, FCM detects three outliers, two represented by stars and one by the sign $+$. (D) $c = 6$, all outliers are detected, they are represented as follows: One by the sign $+$, two by stars and four by lozenge.

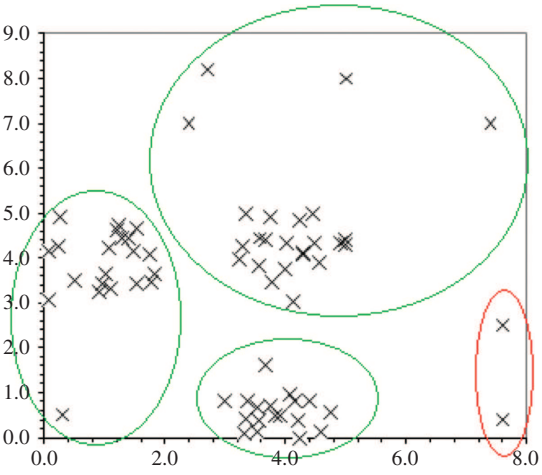

Figure 6: Results of Robust-FCM on the Dataset X_1.
The number of detected classes is 3; the class of outliers is in red.

Table 1: Recognition Rate Obtained by the Four Considered Algorithms on Synthetic Dataset X_1.

FCM	PCM	Robust-FCM	RDUFL
89.23%	40%	92.31%	100%

RDUFL has the highest rate whereas PCM has the lowest.

4.2 Real-World Dataset

The first considered real dataset is «*Lymphography*», which has 148 objects with 18 attributes. These are observations, which were made on patients with cancer in the lymphatic of the immune system. It comprises four classes: normal (two objects), metastases (81 objects), malignant lymphs (61 objects), and fibrosis (four objects).

The second dataset is «Diabetes», and it is composed of 768 objects with four attributes. The data fall into two classes: the class 0 with 500 instances and the class 1 interpreted as "tested positive for diabetes" with 268 instances.

The third dataset is the «Indian» dataset, which comprises 583 objects with 10 attributes. There are two classes: the first with 416 objects and the second with 167 objects.

The fourth dataset is Haberman's Survival dataset that is the result of a measure of 306 cases on the survival of patients who had undergone surgery for breast cancer. It is a three-dimensional pattern classification problem from two classes.

The Breast Cancer dataset is a nine-dimensional pattern classification problem with 699 samples from malignant (cancerous) class and benign (non-cancerous) class. The two classes contain, respectively, 458 and 241 points.

The Parkinsons Disease dataset is composed of a range of biomedical voice measurements. There are 22 attributes and 195 samples from the two classes corresponding to healthy people and those with Parkinson's disease. The two classes contain, respectively, 48 and 147 points.

The Postoperative Patient dataset aims to determine where to send patients in a postoperative recovery area. The number of instances is 90 distributed over three classes: class I (patient sent to Intensive Care Unit) with two items, class S (patient prepared to go home) with 24 items, and class A (patient sent to general hospital floor) with 64 items.

The EEG Eyes dataset consists of EEG values and a value indicating the eye state. This eye state was detected via a camera during the EEG measurement and added later manually to the file. It is a 14-dimensional pattern classification problem with 14,980 samples. The two classes contain, respectively, 8257 and 6723 points.

Table 2 describes the data and provides information on the attributes, size, and number of classes.

We initially checked if there are possible outliers in the considered datasets. To this end, we calculated the proximity degree for the objects and looked for their small values.

Once possible outliers were determined and isolated, we applied RDUFL to the other objects by assessing an object at each iteration. At the end of the learning phase, we obtained the detected classes with their prototypes.

Table 2: Description of Real Datasets.

Data	No. of samples	No. of attributes	No. of classes
Lymphography	148	18	4
Diabetes	768	8	2
Indian	583	10	2
Haberman's Survival	306	3	2
BCW	699	9	2
Post-operative Patient	90	8	3
Parkinsons	197	23	2
EEG Eyes State	14,980	14	2

Table 3: Number of Detected Classes by RDUFL, Possible and True Outliers.

Dataset	Number of detected classes	Number of possible outliers	Number of true outliers
Lymphography	2	15	11
Diabetes	2	6	3
Indian	2	32	14
Haberman's Survival	2	12	6
BCW	2	18	0
Post-operative Patient	2	2	2
Parkinsons	2	9	3
EEG Eyes State	2	4	2

The treatment phase of the possible outliers $\boldsymbol{O_i}$ allowed to obtain the results described in Table 3.

The first finding is that RDUFL allows detecting the exact number of classes for all the examples of the considered data based on the defined dynamic threshold and the learning rule.

For the Lymphography dataset, there really exist four clusters in which two classes are considered rare (classes 1 and 4) given their small size [28, 29]. Class 1 contains two items, and class 4 contains four. The RDUFL detects these six outliers, whereas the robust-FCM detects five outliers in which only two items belong to the rare classes. The PCM does not recognize any rare classes of this set.

For the BCW dataset, the concept of proximity degree allows detecting 18 possible outliers, which were isolated and not considered in the learning phase. The algorithm clusters the dataset into two clusters. The comparison of the 18 possible outliers with the prototypes of detected clusters demonstrates that these items had enough characteristics in common with these detected prototypes. Therefore, they are not true outliers.

We recreate a very unbalanced distribution of the breast cancer data by choosing one in every six malignant records [29]. The result dataset has 39 malignant records (8%) and 444 benign records (92%). For testing the RDUFL, different values of possible outliers are considered instead of excluding all possible outliers. To

Table 4: Sensitivity and Specificity Percentages for Modified BCW Data Using RDUFL.

Possible outliers	Number of malignant	% of malignant	Sensitivity	Specificity
5	39	100	100%	80%
10	39	100	100%	79%
15	39	100	100%	78%
20	39	100	100%	77%
25	39	100	100%	76%
30	39	100	100%	72%
35	39	100	100%	70%
39	39	100	100%	67%

FCM and RDUFL have the same accuracy Percentage.

Table 5: Recognition Rates Obtained for the Considered Data.

Data set	FCM	PCM	Robust-FCM	RDUFL
Lymphography	50.05%	64.87%	58.11%	72.30%
Diabetes	66.02%	37.11%	58.47%	67. 89%
Indian	30.37%	66.03%	46.14%	70.65%
Haberman's Survival	49.02%	50.33%	42.81%	60.13%
BCW	65.96%	61.52%	41.35%	96.57%
Post-operative Patient	61.11%	*	72.22%	73.33%
Parkinsons	74.36%	54.87%	69.04%	75.90%
EEG Eyes State	55.28%	49.71%	50.82%	57.61%

RDUL has the highest recognition rate of 96.5M% on BCW, while if we consider the other algorithms, FCM achieves 74.36% on Parkinsons data, and Robust-FCM achieves 72.22% on Post-operative Patient dataset. *Indicates that it is not possible to learn Post-Operative Patient dataset by using the PCM algorithm.

Table 6: Accuracy Percentage Obtained for the Considered Data.

Data set	FCM	PCM	Robust-FCM	RDUFL
Lymphography	*	*	*	70%
Diabetes	66%	44%	61%	66%
Indian	69%	66%	45%	76%
Haberman's Survival	48%	49%	42%	59%
BCW	95%	65%	76%	95%
Post-operative Patient	*	*	*	70%
Parkinsons	74%	55%	73%	75%
EEG Eyes State	55%	50%	51%	58%

*Indicates that it is not possible to learn Post-Operative Patient dataset by using the PCM algorithm.

evaluate cluster solutions, we measured the sensitivity and the specificity defined by:

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{30}$$

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{31}$$

where:
- Sensitivity is the ability to correctly identify ill patients.
- Specificity is the ability to correctly identify patients without the disease.

The RDUFL detects anomalies for this dataset and identifies outliers (small class) for each considered value of possible outliers. We report the results in Table 4.

Moreover, the RDUFL can improve considerably the performance of clustering and allows an increase of 30.61% on the BCW dataset. Those results show that the adopted approach can lead to an increase in accuracy for class discovering even in the absence of outliers. Indeed, the RDFUFL improves learning and yields a good recognition rate as depicted in Table 5. The recognition rates of each of the studied learning algorithms are also reported, for each dataset, in Table 5.

The accuracy percentage is also determined for each algorithm, based on the values of the equation [Eq. (29)]. The results of the considered algorithms are shown in Table 6. According to this table, we say that RDUFL has the highest accuracy percentage.

5 Conclusion

In this paper, we introduced an adapted approach in order to partition a dataset and detect outliers. This approach consists of three stages. The first stage is a method of pre-treatment, which identifies objects that are likely to be considered possible outliers by utilizing the concept of proximity degree. The second stage is an unsupervised fuzzy learning algorithm, which detects existing classes formed by the data without possible outliers. In this stage, the algorithm equally provides the prototypes of these detected classes and the membership degrees of each object to these classes. The creation of classes is carried out according to a dynamic threshold, which is recalculated at each iteration of the algorithm. This threshold is based on the similarity among the prototypes updated at the exploration of a new object. As for the last stage, it consists in comparing the similarity of each possible outlier to the farthest object belonging to the class that corresponds to its nearest prototype. The experimental results demonstrated the effectiveness of the proposed approach especially that it does not require any user-specified parameter.

Future work will introduce the notion of granular computing [18, 24, 46, 55] to quantify the imprecision and the tolerance of uncertainty in the given large attribute dataset [2, 37, 49]. Indeed, granularity allows simplification, clarity, low cost, and tolerance of uncertainty [36]. It "underlies the remarkable human ability

to make rational decisions in an environment of imprecision" [36]. Thereby, a new robust fuzzy algorithm for unsupervised learning using granular computing techniques will be developed.

Bibliography

[1] F. Angiulli, S. Basta and C. Pizzuti, Distance-based detection and prediction of outliers, *IEEE Trans. Knowl. Data Eng*. **18** (2006), 145–160.

[2] M. Antonelli, P. Ducange, B. Lazzerini, and F. Marcelloni, Multi-objective evolutionary multiplicative aggregation in group decision making design of granular rule-based classifiers, *Granul. Comput.* **1** (2016), 37–58.

[3] S. Ben-David and N. Haghtalab, Clustering in the presence of background noise, in: *Proceedings of the 31st International Conference on Machine Learning*, vol. 32, pp. 280–288, Bejing, China, 2014.

[4] M. Benrabh, A. Bouroumi and A. Hamdoun, A fuzzy validity-guided procedure for cluster detection, *Malays. J. Comput. Sci.* **18** (2005), 31–39.

[5] M. Berthold, Fuzzy models and potential outliers, in: *Proceedings 18th International Conference of the North American Fuzzy Information Processing Society, NAFIPS*, pp. 532–535, IEEE Press, New York, USA, 1999.

[6] J. C. Bezdek, *Pattern recognition with fuzzy objective function algorithms*, Plenum Press, New York, 1981.

[7] J. C. Bezdek, FCM: the fuzzy c-means clustering algorithm, *Comput. Geosci*. **10** (1984), 191–203.

[8] C. L. Blake and C. J. Merz, *UCI repository of machine learning databases*, University of California, Irvine, Department of Information and Computer Sciences (1998). http://www.ics.uci.edu/mlearn/MLRepository.html.

[9] R. J. Bolton and D. J. Hand, Statistical fraud detection: a review, *Stat. Sci.* **17** (2002), 235–255.

[10] A. Bouroumi, M. Limouri and A. Essaïd, Unsupervised fuzzy learning and cluster seeking, *Intell. Data Anal*. **4** (2000), 241–253.

[11] M. M. Breunig, H. P. Kriegel, R. T. Ng and J. Sander, LOF: identifying density-based local outliers, in: *Proceedings of the International Conference on Management of Data*, pp. 93–104, Dallas, TX, USA, May 15–18, 2000.

[12] S. M. Chen and J. H. Chen, Fuzzy risk analysis based on similarity measures between interval-valued fuzzy numbers and interval-valued fuzzy number arithmetic operators, *Exp. Syst. Appl*. **36** (2009), 6309–6317.

[13] S. M. Chen and C. Y. Chien, Parallelized genetic ant colony systems for solving the traveling salesman problem, *Exp. Sys. Appl*. **38** (2011), 3873–3883.

[14] S. M. Chen and P. Y. Kao, TAIEX forecasting based on fuzzy time series, particle swarm optimization techniques and support vector machines. *Inf. Sci*. **247** (2013), 62–71.

[15] S. M. Chen, N. Y. Wang and J. S. Pan, Forecasting enrollments using automatic clustering techniques and fuzzy logical relationships, *Exp. Syst. Appl*. **36** (2009), 11070–11076.

[16] T. Chenglong, Clustering of steel strip sectional profiles based on robust adaptive fuzzy clustering algorithm, *Comput. Inf*. **30** (2011), 357–380.

[17] M. G. C. A. Cimino, G. Frosini, B. Lazzerini and F. Marcelloni, On the noise distance in robust fuzzy c-means, *Int. J. Comput. Inf. Syst. Control Eng*. **1** (2007), 217–220.

[18] D. Ciucci, Orthopairs and granular computing, *Granul. Comput.* **1** (2016), 159–170.

[19] R. N. Dave, Characterization and detection of noise in clustering. *Pattern Recognit. Lett*. **12** (1991), 657–664.

[20] R. N. Dave and R. Krishnapuram, Robust clustering methods: a unified view, *IEEE Trans. Fuzzy Syst.* **5** (1997), 270–293.

[21] R. N. Dave and S. Sen, Noise clustering algorithm revisited, in: *Proceedings of the Annual Meeting of the North American Fuzzy Information Processing Society*, pp. 199–204, Syracuse, NY, USA, September 21–24, 1997.

[22] A. Dik, K. Jebari, A. Bouroumi and A. Ettouhami, A new fuzzy clustering by outliers, *J. Eng. Appl. Sci.* **9** (2014), 372–377.

[23] A. Dik, A. El Moujahid, K. Jebari and A. Ettouhami, A new dynamic algorithm for unsupervised learning, *Int. J. Innov. Comput. Inf. Control* **11** (2015), 1325–1339.

[24] D. Dubois and H. Prade, Bridging gaps between several forms of granular computing, *Granul. Comput.* **1** (2016), 115–126.

[25] A. El Imrani, A. Bouroumi, M. Limouri and A. Essaid, A coevolutionary genetic algorithm using fuzzy clustering, *Int. J. Intell. Data Anal*. **4** (2000), 183–193.

[26] A. Gosaina and S. Dahiya, Performance analysis of various fuzzy clustering algorithms: a review, *Procedia Comput. Sci*. **79** (2016), 100–111.

[27] J. Han and M. Kamber, *Data mining: concepts and techniques*, 2nd ed., Morgan Kaufmann Publishers, San Francisco, 2006.

[28] Z. He, X. Xu and S. Deng, Discovering cluster-based local outliers, *Pattern Recognit. Lett.* **24** (2003), 1641–1650.

[29] Z. He, S. Deng and X. Xu, An optimization model for outlier detection in categorical data, in: *Advances in Intelligent Computing., ICIC 2005, Lecture Notes in Computer Science*, vol. 3644, D. S. Huang, X. P. Zhang and G. B. Huang (Eds.), Springer, Berlin, Heidelberg, 2005.

[30] Y. J. Horng, S. M. Chen, Y. C. Chang and C. H. Lee, A new method for fuzzy information retrieval based on fuzzy hierarchical clustering and fuzzy inference techniques, *IEEE Trans. Fuzzy Syst*. **13** (2005) 216–228.

[31] A. K. Jain, Data clustering: 50 years beyond k-means, *Pattern Recognit. Lett*. **31** (2010), 651–666.

[32] J.-M. Jolion and A. Rosenfeld, Cluster detection in background noise, *Pattern Recognit*. **22** (1989), 603–607.
[33] E. M. Knorr and R. T. Ng, Algorithms for mining distance-based outliers in large dataset, in: *Proceedings of the 24rd International Conference on Very Large Data Bases*, pp. 392–403, San Francisco, CA, USA, August 24–27, 1998.
[34] R. Krishnapuram and J. Keller, A possibilistic approach to clustering, *IEEE Trans. Fuzzy Syst*. **1** (1993), 98–110.
[35] T. Lane and C. E. Brodley, Temporal sequence learning and data reduction for anomaly detection, *ACM Trans. Inform. Syst. Secur*. **2** (1999), 295–331.
[36] P. Lingras, F. Haider and M. Triff, Granular meta-clustering based on hierarchical, network, and temporal connections, *Granul. Comput.* **1** (2016), 71–92.
[37] L. Livi and A. Sadeghian, Granular computing, computational intelligence, and the analysis of non-geometric input spaces, *Granul. Comput*. **1** (2016), 13–20.
[38] A. Loureiro, L. Torgo and C. Soares, Outlier detection using clustering methods: a data cleaning application, in: *Proceedings of KDNet Symposium on Knowledge-Based Systems for the Public Sector*, Bonn, Germany, June 3–4, 2004.
[39] F. Morsier, D. Tuia, M. Borgeaud, V. Gass and J. P. Thiran, Cluster validity measure and merging system for hierarchical clustering considering outliers, *Pattern Recognit.* **48** (2015), 1478–1489.
[40] L. Ott, L. Pang, F. Ramos and S. Chawla, On integrated clustering and outlier detection, *Adv. Neural Inf. Process. Syst.* **27** (2014), 1359–1367.
[41] N. R. Pal, K. Pal, J. M. Keller and J. C. Bezdek, A possibilistic fuzzy c-means clustering algorithm, *IEEE Trans. Fuzzy Syst*. **13** (2005), 517–530.
[42] G. Peters and R. Weber, DCC: a framework for dynamic granular clustering, *Granul. Comput.* **1** (2016), 1–11.
[43] S. Ramaswamy, R. Rastogi and K. Shim, Efficient algorithms for mining outliers from large data sets, in: *Proceedings of SIGMOD'00*, pp. 93–104, Dallas, Texas, 2000.
[44] S. Ramaswamy, R. Rastogi and K. Shim, Efficient algorithms for mining outliers from large data sets, in: *Proceedings of the International Conference on Management of Data*, pp. 427–438, Dallas, TX, USA, May 15–18, 2000.
[45] F. Rehm, F. Klawonn and R. Kruse, A novel approach to noise clustering for outlier detection, *Soft Comput*. **11** (2007), 489–494.
[46] A. Skowron, A. Jankowski and S. Dutta, Interactive granular computing, *Granul. Comput*. **1** (2016), 95–113.
[47] C. Tang, S. Wang and Y. Chen, Clustering of steel strip sectional profiles based on robust adaptive fuzzy clustering algorithm, *Comput. Inform*. **30** (2012), 357–380.
[48] P. W. Tsai, J. S. Pan, S. M. Chen, B. Y. Liao and S. P. Hao, Parallel cat swarm optimization, in: *Proceedings of the seventh International Conference on Machine Learning and Cybernetics*, vol. 6, pp. 3328–3333, Kunming, China, 2008.
[49] G. Wang, J. Yang and J. Xu, Granular computing: from granularity optimization to multi-granularity joint problem solving, *Granul. Comput.* **2** (2017), 105–120.
[50] Y. J. Xu, L. Chen, R. M. Rodríguez, F. Herrera and H. M. Wang, Deriving the priority weights from incomplete hesitant fuzzy preference relations in group decision making, *Knowl. Based Syst*. **99** (2016), 71–78.
[51] Y. J. Xu, J. F. Cabrerizo and E. Herrera-Viedma, A consensus model for hesitant fuzzy preference relations and its application in water allocation management, *Appl. Soft Comput.* **58** (2017), 265–284.
[52] Y. J. Xu, X. Liu and H. M. Wang, The additive consistency measure of fuzzy reciprocal preference relations, *Int. J. Mach. Learn. Cybern*. **9** (2017), 1141–1152.
[53] Y. J. Xu, C. Y. Li and X. W. Wen, Missing values estimation and consensus building for incomplete hesitant fuzzy preference relations with multiplicative consistency, *Int. J. Comput. Intell. Syst*. **11** (2018), 101–119.
[54] Y. J. Xu, X. W. Wen and W. C. Zhang, A two-stage consensus method for large-scale multi-attribute group decision making with an application to earthquake shelter selection, *Comput. Ind. Eng*. **116** (2018), 113–129.
[55] Y. Yao, A triarchic theory of granular computing, *Granul. Comput.* **1** (2016), 145–157.
[56] J. Yu, S. H. Lee and M. Jeon, An adaptive ACO-based fuzzy clustering algorithm for noisy image segmentation, *Int. J. Innov. Comput. Inf. Control* **8** (2012), 3907–3918.
[57] L. A. Zadeh, Fuzzy sets, *Inf. Control* **8** (1965), 338–353.
[58] W. C. Zhang, Y. J. Xu and H. M. Wang, A consensus reaching model for 2-tuple linguistic multiple attribute group decision making with incomplete weight information, *Int. J. Syst. Sci*. **47** (2016), 389–405.