Journal of AI and Data Mining
Vol 8, No 4, 2020, 515-523. ***Original Research Paper*** DOI: 10.22044/JADM.2020.9021.2038

A Fuzzy C-means Algorithm for Clustering Fuzzy Data and Its Application in Clustering Incomplete Data

J. Tayyebi[1] and E. Hosseinzadeh[2*]

1. Department of Industrial Engineering, Birjand University of Technology, Birjand, Iran,
2. Department of Mathematics, Kosar University of Bojnord, Bojnord, Iran.

Received 13 October 2019; Revised 01 April 2020; Accepted 07 April 2020
**Corresponding author: e.hosseinzadeh@kub.ac.ir (E. Hosseinzadeh).*

Abstract
The fuzzy c-means clustering algorithm is a useful tool forclustering; but it is convenient only for crisp complete data. Inthis article, an enhancement of the algorithm is proposed, which issuitable for clustering trapezoidal fuzzy data. A linear rankingfunction is used to define a distance for trapezoidal fuzzy data. Then, as anapplication, a method based on the proposed algorithm is presentedto cluster the incomplete fuzzy data. This method substitutes the missing attribute by a trapezoidal fuzzy number to be determined using the corresponding attribute of the q nearest-neighbor. Comparisons and analysis of the experimental results demonstrate the capability of the proposed method.

Keywords: *Intrusion Detection System, Cloud Computing, Classification Algorithm, Anomaly Detection, Dataset Generation, IDS Assessment, Machine Learning.*

1. Introduction

One of the most important tasks in data mining and pattern recognition is data clustering. Cluster analysis groups data objects based on the information found in data objects that describes the objects and their relationships. Clustering has been intensively studied in machine learning and data mining communities [5, 29, 34]. The goal is that the objects within a group be similar or related to one another and different from the objects in the other groups. The greater similarity within a group and the greater difference between the groups, the better or more distinct the clustering [26].There are various methods and algorithms for data clustering. The fuzzy c-means (FCM) algorithm proposed by Bezdek [2] is a popular method for data clustering, which partitions a real t-dimensional data set$X = \{x_1, x_2, \dots, x_n\} \subseteq \mathbb{R}^t$ into several clusters that are represented by prototypes and degrees of membership of each instance to each cluster [28].

In practical applications, many data sets suffer from incompleteness. Some objects of these data sets have attributes with missing values. It is not unusual for an object to be missing one or more attribute values. In some cases, the information is not collected. In other cases, some attributes are not applicable to all objects. Regardless, the missing values should be taken into account during the data analysis.

In the past four decades, various approaches have been introduced to deal with incomplete data by using supervised tasks [19, 24, 25]. In the past four decades, various approaches have been introduced to deal with incomplete data by using supervised tasks [19, 24, 25]. The expectation-maximization (EM) algorithm [3] was a useful approach for modelling and estimation of the missing attributes, and was used in probabilistic clustering [18]. Subsequently, several methods were proposed for handling the missing values in FCM [20]. One basic strategy, called imputation strategy, replaces the missing values with weighted average of the corresponding attributes [7]. Another approach, ignoring, discards the missing values and calculates the distances from the remaining coordinates [9].

In [9], Hathaway and Bezdek have proposed four strategies to cluster data set suffering from

incompleteness, in which the whole data strategy (WDS) and the partial distance strategy (PDS) are discarding/ignoring methods, and the optimal completion strategy (OCS) and the nearest prototype strategy (NPS) belong to the imputation methods. In WDS, instances that include the missing values must be removed from data set, but this strategy is not desirable because the elimination bring a loss of data. PDS uses the concept of partial distance to be defined for incomplete data by ignoring the missing attributes of incomplete data [4]. OCS views the missing values as an optimization problem and imputes missing values in each iteration to find better estimates. NPS replaces the missing values with the corresponding attributes of the nearest prototype.Li et. al. [14] have proposed a clustering method to cope with the incomplete data. Their method, first, estimates the missing values in the form of intervals using the nearest-neighbor method, which utilizes information about the distribution of data and transforms an incomplete data set into an interval-valued one. Then a kernel method is introduced to increase the separability between data by implicitly mapping them into a higher dimensional feature space.

In [21], Owhadi et al. introduced an Entropy-based Consensus on Cluster Centers for clustering in distributed systems with a consideration for confidentiality of data; i.e.it is the negotiations among local cluster centers that are used in the consensus process, hence no private data are transferred. Yang et al. [33] have constructed a robust learning FCM algorithm, so that it becomes free of the fuzziness index m and initializations without parameter selection, and can also automatically find the best number of clusters. Wu et al. [31] have introduced an advanced FCM clustering algorithm to overcome the weakness of the traditional FCM algorithm, including the instability of random selecting of initial center and the limitation of the data separation or the size of clusters. Li et al. [12] have developed a fuzzy clustering algorithm based on the nearest-neighbor interval (FCM-NNI). In this approach, each one of the attribute values is transformed into an interval based on q nearest-neighbors. If the value of an attribute is not missing, the lower and upper bounds of the interval are equal; otherwise, the lower and upper bounds of the interval will be equal to the minimum and maximum values of the corresponding attribute in the q nearest-neighbors, respectively. This approaches may not be robust when there are outliers in data, because the length of intervals increases and it yields an inaccurate analysis and increases the uncertainty.

A new fluid identification method in carbonate reservoir based on the modified FCM clustering algorithm has been proposed by Liu et. al. [15]. They proposed a modified FCM Clustering algorithm named as CQPSO-FCM Clustering, which combines the Fuzzy C-Means (FCM) Clustering algorithm with Chaotic Quantum Particle Swarm Optimization (CQPSO) algorithm. The modified method can solve the problems of FCM Clustering algorithm's sensitivity to initial values and falling into local convergence. In fact, in their method, clustering is performed on crisp data.

In this article, a new fuzzy c-means algorithm for clustering trapezoidal fuzzy data is proposed. This algorithm employs a linear ranking function to define a distance between fuzzy vectors. Since any real or interval data is a special kind of trapezoidal fuzzy numbers (TFNs), it follows that the proposed algorithm can be applied for clustering the data sets consisting of real, interval or trapezoidal fuzzy data.

Using the proposed algorithm, not only we can cluster fuzzy data, but also it has an application in clustering incomplete data. We also proposed an imputation method to cluster incomplete fuzzy data. The method performs a preprocessing on dataset to transform any non-missing attribute into trapezoidal fuzzy attribute and impute TFNs to the missing attributes of incomplete data. Then, it uses the proposed algorithm to cluster the transformed fuzzy dataset.

This article is organized as follows. Section 2 presents some preliminaries and reviews some notions and notations of fuzzy theory. The new algorithm for clustering fuzzy data is introduced in Section 3. In Section 4, a method for clustering incomplete data based on the introduced algorithm is proposed. Section 5 presents the clustering results. Finally, Section 6 gives the concluding remarks.

2. Preliminaries

2.1. Some Notions of the Fuzzy Set Theory

In this section, we review the fundamental notions of fuzzy set theory, initiated by Bellman and Zadeh [1], to be used throughout this article. The following definitions and notations are taken from [30].

Let X be the universal set. A mapping $\tilde{a}: X \to [0,1]$ is a fuzzy set. The value $\tilde{a}(x)$ of $\tilde{a}$ at $x \epsilon X$ stands for the degree of membership of x in $\tilde{a}$. A fuzzy set $\tilde{a}$ is normal if there exists $x_0 \in X$ such that $\tilde{a}(x_0) = 1$. An α-cut of fuzzy number $\tilde{a}$, $\alpha \in [0,1]$, is a crisp set as

$$\tilde{a}_{\alpha} = \{x \in X: \tilde{a}(x) \geq \alpha\}.$$

If a fuzzy set $\tilde{a}$ satisfies that $\tilde{a}_{\alpha}$ is a closed interval for every $\alpha \in [0,1]$, then $\tilde{a}$ is called a fuzzy number. A special type of fuzzy numbers is trapezoidal fuzzy number (TFN) to be defined as:

$$\tilde{a}_{\alpha} = \begin{cases} \dfrac{x - a^1}{a^2 - a^1} & x \in [a^1, a^2], \\ 1 & x \in [a^2, a^3], \\ \dfrac{x - a^4}{a^3 - a^4} & x \in [a^3, a^4], \\ 0 & otherwise. \end{cases}$$

For simplification, we denote the TFN $\tilde{a}$ by (a^1, a^2, a^3, a^4) (see Figure 1(a)). $\tilde{a}$ is called triangular fuzzy number when $a^2 = a^3$. For instance, (1:5, 2, 2, 2.5) and (1.7; 2, 3, 3.4) are trapezoidal fuzzy numbers, which may be used to describe the fuzzy notion of around number 2 and around interval [2,3], respectively. We denote the set of all trapezoidal fuzzy numbers by $\mathcal{F}(\mathbb{R})$. Since any real number c and any interval $[a, b]$ can be written as (c, c, c, c) and (a, a, b, b), respectively, it is obvious that TFNs are an extension of the real numbers and intervals. A trapezoidal fuzzy vector $\tilde{a}$ is a member of the Cartesian product $\mathcal{F}^n(\mathbb{R}) = \mathcal{F}(\mathbb{R}) \times \mathcal{F}(\mathbb{R}) \times \ldots \times \mathcal{F}(\mathbb{R})$. Figure 1(b) illustrates representation of the vector $(\tilde{a}, \tilde{b}) \in \mathcal{F}^2(\mathbb{R})$. The black regions show full membership and the gray regions partial membership.

We next define arithmetic on trapezoidal fuzzy numbers. Let $\tilde{a} = (a^1, a^2, a^3, a^4)$ and $\tilde{b} = (b^1, b^2, b^3, b^4)$ be two trapezoidal fuzzy numbers and *c* be a real number. The scalar production and addition operators are defined as follows:

- $c\tilde{a} = (ca^1, ca^2, ca^3, ca^4), \quad if\ c \geq 0;$
- $c\tilde{a} = (ca^4, ca^3, ca^2, ca^1), \quad if\ c < 0;$
- $\tilde{a} + \tilde{b} = (a^1 + b^1, a^2 + b^2, a^3 + b^3, a^4 + b^4),$
- $\tilde{a} - \tilde{b} = (a^1 - b^4, a^2 - b^3, a^3 - b^2, a^4 - b^1).$

We point out that the arithmetic on trapezoidal fuzzy numbers follows the Extension Principle (for a discussion, see [30]).

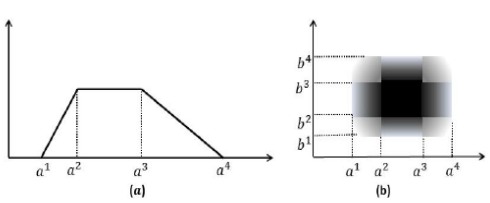

Figure 1. (a) Membership function of TFN $\tilde{a} = (a^1, a^2, a^3, a^4)$, (b) Representation of $(\tilde{a}, \tilde{b}) = ((a^1, a^2, a^3, a^4), (b^1, b^2, b^3, b^4)) \in \mathcal{F}^2(\mathbb{R})$.

2.2. Ranking Function

There are several methods comparing fuzzy numbers which can be seen in Fang and Hu [6], Lai and Hwang [11], Shoacheng [23] and Tanaka and Ichihashi [27]. One of the most convenient of these methods is based on the concept of comparison of fuzzy numbers using ranking functions [8, 17]. In fact, an efficient approach for ordering the elements of $\mathcal{F}(\mathbb{R})$ is to define a ranking function $\Re: \mathcal{F}(\mathbb{R}) \to \mathbb{R}$ that maps each trapezoidal fuzzy number into the real line, where a natural order exists. The concept of ranking function is used to define a distance between trapezoidal fuzzy vectors in the next section. We only restrict our attention to linear ranking functions, i.e. a ranking function $\Re$ such that

$$\Re(\tilde{a} + c\tilde{b}) = \Re(\tilde{a}) + c\Re(\tilde{b}), \qquad (1)$$

for any $\tilde{a}, \tilde{b} \in \mathcal{F}(\mathbb{R})$ and any $c \in \mathbb{R}$. It is obvious that $\Re(\tilde{0}) = 0$, where $\tilde{0} = (0,0,0,0)$.

Lemma 1. *For fixed nonnegative numbers $\alpha, \beta \in \mathbb{R}$, the function $\Re: \mathcal{F}(\mathbb{R}) \to \mathbb{R}$ is defined as:*

$$\Re(\tilde{a}) = \alpha a^1 + \beta a^2 + \beta a^3 + \alpha a^4 \qquad (2)$$

where $\tilde{a} = (a^1, a^2, a^3, a^4) \in \mathcal{F}(\mathbb{R})$, is a linear ranking function.

Proof. Let the ranking function $\Re$ be defined by $\Re(\tilde{a}) = \alpha a^1 + \beta a^2 + \beta' a^3 + \alpha' a^4$, where $\alpha, \alpha', \beta, \beta' \in \mathbb{R}$ and $\tilde{a} = (a^1, a^2, a^3, a^4) \in \mathcal{F}(\mathbb{R})$. It is easy to verify linearity for any real number $c \geq 0$, i.e. $\Re(\tilde{a} + c\tilde{b}) = \Re(\tilde{a}) + c\Re(\tilde{b})$ for each $\tilde{a}, \tilde{b} \in \mathcal{F}(\mathbb{R})$. Since $\Re((0,0,0,0)) = 0$, it follows that:

$$\Re(-\tilde{a}) = -\Re(\tilde{a}),$$

or equivalently,

$$-\alpha a^4 - \beta a^3 - \beta' a^2 - \alpha' a^1 = -(\alpha a^1 + \beta a^2 + \beta a^3 + \alpha a^4), \quad (3)$$

for each $\tilde{a} = (a^1, a^2, a^3, a^4) \in \mathcal{F}(\mathbb{R})$. The relation (3) implies that $\alpha' = \alpha$ and $\beta' = \beta$.

For instance, if $\alpha = \beta = \frac{1}{4}$, then $\Re(\tilde{a}) = \frac{a1+a2+a3+a4}{4}$, that has been proposed by Yager [32].

3. Fuzzy C-means Clustering Algorithm for Fuzzy Data

In this section, a novel FCM algorithm is resented for clustering trapezoidal fuzzy data. This algorithm is an extension of the regular FCM

algorithm presented in [2] since the regular FCM algorithm only clusters real data.

Suppose that $\tilde{X} = [\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n]$ is a data set where $\tilde{x}_k = [\tilde{x}_{1k}, \tilde{x}_{2k}, \dots, \tilde{x}_{tk}]^T$, k=1, 2, …, n and $\tilde{x}_{lk} = \left(x_{lk}^1, x_{lk}^2, x_{lk}^3\right) \in \mathcal{F}(\Re)$ for all k=1, 2, …, n and l= 1, 2, …, t. Thus, $\tilde{X}$ is a member of $\mathcal{F}^{t\times n}(\mathbb{R})$. We want to partition $\tilde{x}_k$'s into c clusters. Let $\tilde{V} = [\tilde{v}_1, \tilde{v}_2, \dots, \tilde{v}_c] \in \mathcal{F}^{t\times c}(\mathbb{R})$ be a trapezoidal fuzzy matrix of the prototypes where $\tilde{v}_i = [\tilde{x}_{1i}, \tilde{x}_{2i}, \dots, \tilde{x}_{ti}]^T$, i=1, 2, …, c and $\tilde{v}_{li} = \left(v_{li}^1, v_{li}^2, v_{li}^3\right)$ for all i=1, 2, …, c and l=1, 2, …, t. Since the data are fuzzy, it is supposed that the prototypes are also TFNs.

In the following, we use the concept of linear ranking function to define a distance between the fuzzy vectors $\tilde{x}_k$'s and $\tilde{v}_i$'s to be required to extend the regular FCM algorithm.

Definition 1. Let $\Re$ be a linear ranking function. The mapping $d_\Re: \mathcal{F}^t(\mathbb{R}) \times \mathcal{F}^t(\mathbb{R}) \to \mathbb{R}$ with

$$d_\Re(\tilde{x}, \tilde{y}) = \sqrt{\sum_{l=1}^{t} \Re^2(\tilde{x}_l - \tilde{y}_l)} = \sqrt{\sum_{l=1}^{t} (\Re(\tilde{x}_l) - \Re(\tilde{y}_l))^2}$$

is called a fuzzy distance with respect to $\Re$ where $\tilde{x} = [\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_t]$, $\tilde{y} = [\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_t] \in \mathcal{F}^t(\mathbb{R})$.

It is obvious that the definition of $d_\Re$ is a direct extension of the formal Euclidean distance. Based on Lemma 1, the ranking mapping $\Re$ to be defined by (2) is linear. Thus, for this ranking function $\Re$, the fuzzy distance can rewritten as follows:

$$d_\Re^2([\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_t], [\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_t]) = \sum_{l=1}^{t} [\alpha\left(x_l^1 + x_l^4 - y_l^1 - y_l^4\right) + \beta(x_l^2 + x_l^3 - y_l^2 - y_l^3)]^2 \quad (4)$$

Where $\tilde{x}_l = (x_l^1, x_l^2, x_l^3, x_l^4)$ and $\tilde{y}_l = (y_l^3, y_l^3, y_l^3, y_l^3)$ for l=1, 2, …, t. The last relation determines the value $d_\Re(\tilde{x}, \tilde{y})$ explicitly; but we will use the compact form of Definition 1.

The proposed FCM clustering algorithm solves

$$\min_{U\in\mathbb{R}^{c\times n}, \tilde{V}\in\mathcal{F}^{t\times c}(\mathbb{R})} J(U, \tilde{V}) = \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^m d_\Re^2(\tilde{x}_k, \tilde{v}_i), \quad (5)$$

Subject to

$$\sum_{i=1}^{c} u_{ik} = 1, \quad k = 1, 2, \dots, n, \quad (6)$$

where $m > 1$ is a nonnegative integer and called fuzzification parameter. Instead of solving (5) subject to (6), the constraints (6) is adjoined to $J(U, \tilde{V})$ by means of Lagrange multipliers method [16] as follows:

$$J(U, \tilde{V}, \lambda) = \sum_{i=1}^{c}\sum_{k-1}^{n} u_{ik}^m d_\Re^2(\tilde{x}_k, \tilde{v}_i) - \sum_{k=1}^{n} \lambda_k \left(\sum_{i=1}^{c} u_{ik} - 1\right), \quad (7)$$

Where $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_n]^T \in \mathbb{R}^n$ is the Lagrange multipliers vector. By setting the gradients of $J(U, \tilde{V}, \lambda)$ with respect to U, $R(\tilde{V})$ and λ to zero, the stationary points of the objective function (7) can be found. Thus the following relations are obtained:

$$\Re(\tilde{v}_i) = \frac{\sum_{k=1}^{n} u_{ik}^m \Re(\tilde{x}_k)}{\sum_{k=1}^{n} u_{ik}^m}, \quad i = 1, \dots, c \quad (8)$$

and

$$u_{ik} = \left[\sum_{t=1}^{c} \left(\frac{d_\Re^2(\tilde{x}_k, \tilde{v}_t)}{d_\Re^2(\tilde{x}_k, \tilde{v}_i)}\right)^{\frac{1}{m-1}}\right]^{-1}, \quad i = 1, \dots, c \quad k = 1, \dots, n. \quad (9)$$

If the matrix $\tilde{V}$ is given the relation (9) explicitly determines the partition matrix U. If we can also obtain $\tilde{V}$ for a the given partition matrix U, then we can repeatedly calculate U and $\tilde{V}$ with respect to another.

Since the ranking function *R* is linear, the relation (8) can be re-written as

$$\Re(\tilde{v}_i) = \Re\left(\frac{\sum_{k=1}^{n} u_{ik}^m \tilde{x}_k}{\sum_{k=1}^{n} u_{ik}^m}\right), \quad i = 1, \dots, c. \quad (10)$$

For the prototype $\tilde{v}_i$ to be defined as

$$\tilde{v}_i = \frac{\sum_{k=1}^{n} u_{ik}^m \tilde{x}_k}{\sum_{k=1}^{n} u_{ik}^m}, \quad i = 1, 2, \dots, c \quad (11)$$

the relation (8) is satisfied. Due to the closeness of the set $\mathcal{F}(\mathbb{R})$ with respect to fuzzy addition and scaler production, the imputation (11) is well-defined. Thus, we can say that the obtained prototypes are linear combinations of fuzzy data.

Now, we describe our proposed algorithm for clustering trapezoidal fuzzy data.

Because real numbers and intervals are the special kinds of TFNs, Algorithm 1 can also be applied

for clustering real (or interval) data. Since any real number c can be written as (c, c, c, c), Algorithm 1 is converted to the regular FCM algorithm for real data. It is a privilege for our algorithm that if the data is a special kind (such as real numbers, intervals or TFNs), then the prototypes are the same type; because prototypes are linear combinations of data (see (11)).

Algorithm 1 Clustering complete fuzzy data.

Input: The trapezoidal fuzzy matrix $\tilde{X} = [\tilde{x}_1, \tilde{x}_2, \ldots \tilde{x}_n]$.
Initialization: Choose the numbers c, m and $\varepsilon > 0$. Initialize the partition matrix $U^{(0)}$ and set $itn := 1$;
Step 1: Calculate the matrix of cluster prototypes $\tilde{V}^{(itn)}$ using (11) and $U^{(itn-1)}$.
Step 2: Compute the partition matrix $U^{(itn)}$ using (9) and $\tilde{V}^{(itn)}$.
Step 3: if $\max_{i,k}\left|u_{i,k}^{(itn)} - u_{i,k}^{(itn-1)}\right| \leq \varepsilon$, then stop; otherwise set $itn := itn + 1$ and return to step 1.
Output: The partition matrix $U^{(itn)}$ and the matrix $\tilde{V}^{(itn)}$

Despite the efficiency of Algorithm 1 for fuzzy data, it is not suitable for incomplete data. In the next section, we introduce a modification of the algorithm to be used for clustering fuzzy incomplete data.

4. A New Method for Clustering Incomplete Data

In this section, the main strategy for dealing with incomplete data is proposed. This strategy, which is called Fuzzy Nearest Neighborhood Mean (FCM-FNNM), consists of some preprocessing that should be done before applying Algorithm 1. Suppose that we have an incomplete data set
where some (but not all) of its attribute values are missing. In the preprocessing, we replace the missing values with TFNs to be determined using the corresponding attributes of the q nearest-neighbors, where q is a fixed nonnegative integer. Since we deal with fuzzy incomplete
data, we cannot apply the fuzzy distance defined in Definition 3 directly and it is required to introduce the concept of partial distance [4]. For two fuzzy incomplete data $\tilde{x}_p = [\tilde{x}_{1p}, \tilde{x}_{2p}, \ldots, \tilde{x}_{tp}], \tilde{x}_q = [\tilde{x}_{1q}, \tilde{x}_{2q}, \ldots, \tilde{x}_{tq}] \in \mathcal{F}^t(\mathbb{R})$, the partial distance is defined as:

$$d_{par}(\tilde{x}_p, \tilde{x}_q) = \frac{t}{\sum_{j=1}^{t} I_j} \sum_{j=1}^{t} (\Re(\tilde{x}_{jp} - \tilde{x}_{jq})))^2 I_j, \quad (12)$$

where $\Re$ is a linear ranking function and

$$I_j = \begin{cases} 1, & if\ both\ x_{jp}\ and\ x_{jq}\ are\ nonmissing; \\ 0, & otherwise. \end{cases}$$

The introduced partial distance can be applied for fuzzy incomplete data.
By calculating the partial distance, we can find the q nearest-neighbors to an incomplete data. In [12] (FCM-NNI), the authors have used the concept of partial distance to search for the maximum and minimum values of a missing attribute in the q nearest-neighbors. They recommended an appropriate technique and formed these two values as an interval of the missing attribute. In their approach the lower and upper bounds of the interval are equal with the minimum and maximum values of the corresponding attribute in the q nearest-neighbors, respectively [12]. This approach is not suitable any more when there are outliers in data, because the length of intervals increases and it yields an inaccurate analysis and also the uncertainty increases.
Similar to FCM-NNI, our proposed method uses the partial distance to find the q nearest-neighbors to an incomplete object, but it can also be used for fuzzy data sets. The method replaces any missing value with a TFN. Suppose that the value of jth attribute in object $\tilde{x}_p \in \mathcal{F}^t(\mathbb{R})$ is missing and Q denotes the index set of its q nearest neighbors whose jth attribute is not missing. The missing attribute can be rewritten into the fuzzy form as $\tilde{x}_{jp} = (x_{jp}^1, x_{jp}^2, x_{jp}^3, x_{jp}^4)$ where

$$x_{jp}^1 = \min\{x_{jr}^1 : r \in Q\},$$

$$x_{jp}^2 = avg\{x_{jr}^2 : r \in Q\},$$

(13)

$$x_{jp}^3 = avg\{x_{jr}^3 : r \in Q\},$$

$$x_{jp}^4 = \max\{x_{jr}^4 : r \in Q\},$$

and *avg* denotes the mean of a set.
The proposed approach makes full use of attribute information of both complete and incomplete data, although, determination of q (number of nearest neighbors) is important. It is obvious that determination strategy of q is related to the number of missing attributes. If the number of nearest neighbors is too large, the performance and accuracy of the analysis will be affected. The determination strategy is to randomly consider one non-missing attribute x_{jp} as missing and find its TFN $\tilde{x}_{jp}$ by assuming q = 1,2, … , and then

compute the degree of membership $\tilde{x}_{jp}$ at x_{jp}. This process should be done repeatedly to estimate the mean of the degrees of membership, denoted by m_q, for each q. Let q_0 be the least value of the set $\{q = 1, 2, \ldots : m_q \geq \mu_0\}$, where $\mu_0 \in [0,1]$ is the expectation degree of membership for missing data (e.g., $\mu_0 = 0.5$). When $q < q_0$, the expectation degree of membership is not achieved; and when $q > q_0$, the computational time is large. Thus, $q = q_0$ can be selected as the number of nearest-neighbors for the incomplete data set.

Now, we are ready to explain our proposed method for clustering incomplete data. The method (FCM-FNNM) overcomes the problem of clustering incomplete data in two phases. The first phase transforms the original data set to a fuzzy complete one to be done as follows. Suppose that $\tilde{x}_{jp}$ is the jth attribute of $\tilde{x}_p$

– If $\tilde{x}_{jp}$ is a real number c, then we reset $\tilde{x}_{jp} = (c, c, c, c)$.
– If $\tilde{x}_{jp}$ is an interval $[a, b]$, then we reset $\tilde{x}_{jp} = (a, a, b, b)$.
– If $\tilde{x}_{jp}$ is missing, then we reset $\tilde{x}_{jp} = \left(x_{jp}^1, x_{jp}^2, x_{jp}^3, x_{jp}^4\right)$, where x_{jp}^1, x_{jp}^2, x_{jp}^3 and x_{jp}^4 are defined in (13).

Then, the second phase applies Algorithm 1 for clustering the transformed data set.

Remark 1. When the method is applied to cluster incomplete real data set, the missing attributes are estimated by triangular fuzzy numbers because $x_{jp}^2 = x_{jp}^3$ in this case.

5. Experiments

In this section, the capabilities of the proposed method are evaluated using numerical experiments. The proposed method is evaluated using several data sets. As will be discussed, the proposed method is evaluated with different portions of the missing attributes.

5.1. Data Sets

In order to evaluate the proposed method, it is applied to several sets including an artificial data sets and two UCI data sets.

The artificial data set consists of two Gaussian multivariate distributions representing two clusters. The mean vector and covariance matrices of the Gaussians are chosen $\mu_1 = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$, $\Sigma_1 = \begin{pmatrix} 0.5 & 0.1 \\ 0.1 & 0.5 \end{pmatrix}$ and $\mu_2 = \begin{pmatrix} -2 \\ -1 \end{pmatrix}$, $\Sigma_2 = \begin{pmatrix} 0.7 & 0 \\ 0 & 0.7 \end{pmatrix}$.

The two clusters are shown in figure 2. These two clusters are separable in only one dimension and the missing of value may occur in that dimension. In this case the object may belong to any of the two clusters depending on the value of the missing attribute. The proposed can provide a good degree of uncertainty, which is a solution to this problem, and as the experiments show the proposed method is robust to this issue.

The real world data sets used in this work are available at the UCI machine learning repository [10]. The characteristics of these data sets are summarized in table 1.

None of these data sets include the missing objects. In order to evaluate the clustering performance and its robustness, we artificially made the data sets missing with different missing percentages.

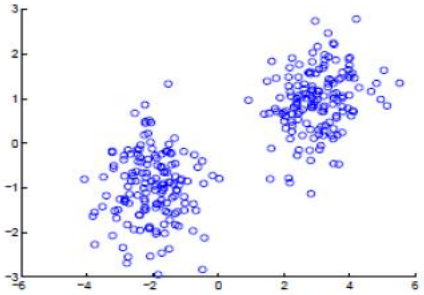

Figure 2. Two Gaussian distributions representing two clusters.

Table 1. Characteristics of UCI data sets.

Data sets	Number of instances	Number of features
Iris	345	150
BUPA	5	4

5.2. FCM Parameters

Determination of q is the next step. As mentioned in Section 4, the determination strategy is based on distribution of incomplete data. The selected values for q, based on different missing percentages are shown in table 2.

In this experiment, we set the fuzzification parameter m = 2, the convergence threshold $\varepsilon = 10^{-6}$, and as discussed in Section 2, we set $\alpha = \beta = \frac{1}{4}$.

In order to compare the performance of different approaches, we need to apply them to the data sets with different missing percentages and then repeat our experiment several times in order to reduce the variation from trail to trial. The results of this experiment are presented in the next section.

Table 2. Selected values for q.

Data sets	%Missing			
	5%	10%	15%	20%
Artificial	3	3	5	7
Iris	5	6	7	7
BUPA	5	5	5	5

5.3. Evaluation Criterions

In our experiments, numbers of misclassification and mean prototype error are considered as evaluation criterions. The cluster prototype for the artificial data set is obtained using ordinary FCM on the complete data set, which is as follows:

$$\boldsymbol{V}^{*}_{Artificial} = \begin{pmatrix} 3.0325 & -2.0278 \\ 0.9684 & -0.9821 \end{pmatrix}$$

In practical clustering problems, optimal cluster prototypes are unavailable but Hathaway and Bezdek have presented the actual cluster prototypes for the Iris data set [9]

$$\boldsymbol{V}^{*}_{Iris} = \begin{pmatrix} 5.00 & 5.93 & 6.58 \\ 3.42 & 2.77 & 2.97 \\ 1.46 & 4.26 & 5.55 \\ 0.24 & 1.32 & 2.02 \end{pmatrix}$$

Let $\widetilde{\boldsymbol{V}}$be the obtained prototype matrix using the proposedmethod. The prototype error can be defined as follows:

$$\left\|\widetilde{\boldsymbol{V}} - \boldsymbol{V}^{*}\right\|_2^2 = \sum_{j=1}^{t}\sum_{i=1}^{c}\left(\boldsymbol{R}\left(\widetilde{\boldsymbol{v}}_{ij}\right) - \boldsymbol{v}^{*}_{ij}\right)^2$$

where $R(.)$is a linear ranking function.

5.4. Numerical Results

In this section, the proposed method is evaluated based on the aforementioned criteria. Firstly, the mean prototype error on artificial data set, for different portions of the missing values is presented. These results are summarized in table 3, which are the averaged results of 30 trials.

After the artificial data set, in order to have a more realistic evaluation, the proposed method is applied to the aforementioned UCI data sets. Since the optimal cluster prototypes are only available for Iris data set, the means prototype error is calculated only for this data set. As another evaluation criterion, the number of misclassification is calculated for both data sets. The results are obtained by repeating the experiments 30 times.

In the rest of this section, the proposed method is compared with the competing methods. For 0% missing, all of the approaches are reduced to the regular FCM. In this case, the numbers of misclassification for all of the approaches are equal. There is no doubt that the information of all objects is important for clustering; despite this, the WDS approach, ignores the incomplete objects and this loss of information can have undesirable effects on the clustering results.

Table 3. Mean prototype error on the artificial data set.

%Missing	5	10	15	20
	Mean number of misclassification			
WDS	1.2586	1.5382	1.7420	1.9476
PDS	0.8701	1.0360	1.2017	1.4360
OCS	0.0492	0.6390	0.2396	0.9450
NPS	0.0109	0.0704	0.1074	0.7341
NNI	0.0128	0.0620	0.1038	0.5702
FNNM	**0.0094**	**0.0371**	**0.0755**	**0.1420**

Table 4. Mean number of misclassification on Iris.

%Missing	0	5	10	15	20
	Mean number of misclassification				
WDS	16	16.58	16.85	16.50	16.65
PDS	16	16.96	16.93	17.93	16.59
OCS	16	17.05	16.68	17.11	16.58
NPS	16	16.81	16.75	16.70	16.41
NNI	16	16.57	16.40	16.23	16.30
FNNM	16	**16.03**	**15.86**	**15.73**	**15.80**

Table 5. Mean prototype error on Iris.

%Missing	0	5	10	15	20
	Mean prototype error				
WDS	0.068	0.069	0.078	0.131	0.150
PDS	0.068	0.053	0.057	0.064	0.067
OCS	0.068	0.051	0.056	0.063	0.064
NPS	0.068	0.052	0.058	0.063	0.065
NNI	0.068	0.046	0.043	0.042	0.044
FNNM	0.068	**0.040**	**0.039**	**0.038**	**0.041**

Table 6. Mean number of misclassification on BUPA.

%Missing	0	5	10	15	20
	Mean number of misclassification				
WDS	181	181.50	182.46	182.65	183.43
PDS	181	181.60	181.36	182.96	183.56
OCS	181	181.63	181.30	182.46	183.30
NPS	181	181.60	181.36	182.26	183.60
NNI	181	181.20	181.40	178.73	180.36
FNNM	181	**181.16**	**181.26**	**179.26**	**179.66**

The performance of WDS is correlated with the size of data set and the number of incomplete samples. In general, for the data sets with small size and small number of missing objects, WDS has a good performance [9]. The PDS method is also based on an ignorance scheme. On the other hand, the imputation approaches could suffer from the outlier's issues. The outliers may cause a biased imputation. Since the proposed method employs TFNs for imputation rather than intervals, it is more robust to outriders and at the same time provide a reasonable degree of uncertainty.

The incomplete data of elements is converted into a fuzzy complete data based on the information of their neighbours. Therefore, it is clear that they

will inherit the corresponding attributes of the dominant cluster among their neighbours. Sometimes completing the incomplete data improves the clustering data, because deleting outlier of some attributes and imputing them with the corresponding attributes of nearest-neighbours cause those elements to be clustered correctly.
The obtained results for FNNM in comparison with the other competing methods show the robustness and capabilities of the proposed method. FNNM makes full use of attribute information, even the information of missing objects. Using the proposed scheme, all of the instances are taken into account in order to find prototypes. As the results shows, as the number of samples with missing attributes increases, FNNM has the best performance and robustness for all data sets.

6. Conclusions

In this article, we have presented an algorithm for clustering fuzzy databased on the fuzzy c-means algorithm, and we used it for presenting a method forclustering incomplete data sets. It is notable that, the proposed method can beapplied for clustering incomplete data sets with fuzzy, interval or real data.The proposed method makes full use of attribute information, and in comparisonwith the competing approaches, it is simpler and less susceptible to both outliersand increase in the number of missing data. Experiments using two famous UCIdata sets show the performance and capabilities of the proposed method. The resultsobtained show that the proposed algorithm is superior to the competingmethod, and it is an effective solution to the problem of clustering incompletedata.

References

[1] Bellman, R. E. & Zadeh, L. A. (1970). Decision making in a fuzzy environment, Manag. Sci, vol. 17, pp. 141-164.

[2] Bezdek, J. C. (1981). Pattern recognition with fuzzy objective function algorithms, Plenum, New York.

[3] Dempster, A. P., Laird, N. M. & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society, Series B, vol. 39, pp. 1-38.

[4] Dixon, J. K. (1979). Pattern recognition with partly missing data, IEEE Trans Syst Man Cybern, vol. 9, pp. 617-621.

[5] Dutt, A., Ismail, M. A., & Herawan, T. (2017). A systematic review on educational data mining, IEEE Access, vol. 5, pp. 15991-16005.

[6] Fang, S. C., Hu, C. F., Wang, H. F., & Wu, S. Y. (1999). Linear programming with fuzzy coefficients in constraints, Computers &Mathematics with Applications, vol. 37, no. 10, pp. 63-76.

[7] Farhangfar, A., Kurgan, L. A., & Pedrycz, W. (2007). A novel framework for imputation of missing values in databases, IEEETransactions on Systems, Man, and Cybernetics-Part A: System sand Humans, vol. 37, no. 5, pp. 692-709.

[8] Garcia-Aguado, C., & Verdegay, J. L. (1993). On the sensitivity of membership functions for fuzzy linear programming problems, Fuzzy Sets and Systems, vol. 56, no. 1, pp. 47-49.

[9] Hathaway, R. J. &Bezdek, J. C. (2001). Fuzzy c-means clustering of incomplete data, IEEE Transactions on systems, Man, and Cybernetics Part B: Cybernetics, vol. 31, no. 5, pp. 735-744.

[10] Hettich, S., Blake, C. L. & Merz, C. J. (1998). UCI repository of machine learning database, Department of Information and Computer Science, University of California, Irvine, CA. http.

[11] Lai, Y. J. & Hwang, C. L. (1992). Fuzzy Mathematical Programming Methods and Applications, Springer, Berlin.

[12] Li, D., Gu, H., & Zhang, L. (2010). A fuzzy c-means clustering algorithm based on nearest-neighbor intervals for incomplete data, Expert Systems with Applications, vol. 37, no. 10, pp. 6942-6947.

[13] Li, D., Gu, H., & Zhang, L. (2013). A hybrid genetic algorithm fuzzy c-means approach for incomplete data clustering based on nearest-neighbor intervals, Soft Computing, vol. 17, no. 10, pp.1787-1796.

[14] Li, T., Zhang, L., Lu, W., Hou, H., Liu, X., Pedrycz, W. & Zhong,C. (2017). Interval kernel Fuzzy C-Means clustering of incomplete data, Neurocomputing, vol. 237, pp. 316-331.

[15] Liu, L., Sun, S. Z., Yu, H., Yue, X. & Zhang, D. (2016). A modified Fuzzy C-Means (FCM) Clustering algorithm and its application on carbonate fluid identification, Journal of Applied Geophysics, vol. 129, pp. 28-35.

[16] Luenberger, D. G. (1984). Linear and Nonlinear Programming, 2nded. Addison-Wesley.

[17] Maleki, H. R. (2002). Ranking functions and their applications to fuzzy linear programming, Far East J. Math. Sci, vol. 4, pp. 283-301.

[18] Mclachlan, G. J. & Basford, K. E. (1988). Mixture models: inference and applications to clustering, Marcel Dekker, New York.

[19] Mesquita, D. P., Gomes, J. P., Junior, A. H. S., &Nobre, J. S.(2017). Euclidean distance estimation in incomplete datasets. Neurocomputing, vol. 248, pp. 11-18.

[20] Miyamoto, S., Takata, O. & Umayahara, K. (1998). Handling missing values in fuzzy c-means. In Proceedings of the third Asian fuzzy systems symposium, Masan, Korea, pp. 139-142.

[21] Owhadi-Kareshki, M. (2019). Entropy-based Consensus for Distributed Data Clustering, Journal of AI and Data Mining, vol. 7, no. 4, pp. 551-561.

[22] Sebestyen, G. S. (1962). Decision-making process in pattern recognition, NY: Macmillan Press.

[23] Shaocheng, T. (1994). Interval number and fuzzy number linear programming, Fuzzy sets and systems, vol. 66, no. 3, pp. 301-306.

[24] Shen, J., Zheng, E., Cheng, Z. & Deng, C. (2017). Assisting attraction classification by harvesting web data, IEEE Access, vol. 5, pp.1600-1608.

[25] Li, J., Struzik, Z., Zhang, L., & Cichocki, A. (2015). Feature learning from incomplete EEG with denoising auto encoder, Neurocomputing, vol. 165, pp. 23-31.

[26] Tan, P. N., Steinbach, M. & Kumar, V. (2005). Introduction to Datamining, Addison- Wesley.

[27] Tanaka, H. &Ichihashi, H. (1984). A formulation of fuzzy linear programming problem based on comparison of fuzzy numbers, Control Cyber, vol. 13, pp. 185-194.

[28] Teodoridis, S. & Koutroumbas, K. (2006). Pattern recognition, Third ed. Academic press, San Diego.

[29] Wang, Z. (2017). Determining the clustering centers by slope difference distribution, IEEE Access, vol. 5, pp. 10995-11002.

[30] Wang, X., Ruan, D. & Kerre, E. E. (2009). Mathematics of Fuzziness ″U Basic Issues, Springer-Verlag Berlin Heidelberg.

[31] Wu, S., Pang, Y., Shao, S. & Jiang, K. (2018). Advanced fuzzy C-means algorithm based on local Density and Distance, Journal of Shanghai Jiaotong university (Science), vol. 23, no. 5, pp. 636-642.

[32] Yager, R.R. (1981). A procedure for ordering fuzzy sets of the unit interval, Information Sciences, vol. 24, pp. 143-161.

[33] Yang, M. S. & Nataliani, Y. (2017). Robust-learning fuzzy c-means clustering algorithm with unknown number of clusters, Pattern Recognition, vol. 71, pp. 45-59.

[34] Zhang, T. T. & Yuan, B. (2018). Density-based multiscale analysis for clustering in strong noise settings with varying densities, IEEE Access, vol. 6, pp. 25861-25873.

نشریه هوش مصنوعی و داده کاوی

الگوریتم C-means فازی برای خوشه‌بندی داده‌های فازی و کاربرد آن در خوشه‌بندی داده‌های ناقص

جواد طیبی[1] و الهام حسین‌زاده[2،*]

[1] گروه مهندسی صنایع، دانشکده مهندسی کامپیوتر و صنایع، دانشگاه صنعتی بیرجند، بیرجند، ایران.

[2] گروه ریاضیات و کاربردها، دانشکده علوم پایه و فنی مهندسی، دانشگاه کوثر بجنورد، بجنورد، ایران.

ارسال ۲۰۱۹/۱۰/۱۳؛ بازنگری ۲۰۲۰/۰۴/۰۱؛ پذیرش ۲۰۲۰/۰۴/۰۷

چکیده:

الگوریتم خوشه‌بندی فازی یک ابزار مفید برای خوشه‌بندی است، که معمولا برای داده‌های کامل دقیق استفاده می‌شود. در این مقاله، پیشرفتی از این الگوریتم ارائه شده است که برای خوشه‌بندی داده‌های فازی ذوزنقه‌ای مناسب است. در اینجا از یک تابع رتبه‌بندی خطی برای تعریف فاصله بین داده های فازی ذوزنقه استفاده می شود. سپس، به عنوان یک کاربرد، روشی مبتنی بر الگوریتم پیشنهادی برای خوشه‌بندی داده‌های ناقص فازی ارائه شده است. این روش ویژگی گمشده را با یک عدد فازی ذوزنقه‌ای جایگزین می‌کند تا با استفاده از ویژگی متناظر با نزدیکترین همسایگی‌اش تعیین شود. مقایسات و تجزیه و تحلیل نتایج تجربی، توانایی روش پیشنهادی را نشان می‌دهد.

کلمات کلیدی: سیستم تشخیص نفوذ، محاسبات ابری، الگوریتم دسته‌بندی، تشخیص ناهنجاری، تولید مجموعه داده، ارزیابی تشخیص نفوذ، یادگیری ماشین.