Computational Statistics (2023) 38:25–51
https://doi.org/10.1007/s00180-022-01215-6

ORIGINAL PAPER

The fuzzy cluster analysis for interval value using genetic algorithm and its application in image recognition

Dinh Phamtoan[1] · Tai Vovan[2]

Received: 11 May 2021 / Accepted: 17 February 2022 / Published online: 26 April 2022

Abstract

This article proposes the genetic algorithm in fuzzy clustering problem for interval value (IGI). In this algorithm, we use the overlap divergence to assess the similarity of the intervals, and take the new index (IDB) as the objective function to build the IGI. The crossover and selection operators in IGI are modified to optimize the results in clustering. The IGI not only determines the suitable number of groups, optimizes the result of clustering but also finds the probability of assigning the elements to the established clusters. The proposed algorithm is also applied in image recognition. The convergence of the IGI is considered and illustrated by the numerical examples. The complex computations of the IGI are performed conveniently and efficiently by the built Matlab program. The experiments on the data-sets having different characteristics and elements show the reasonableness of the IGI, and its advantages overcome other algorithms.

Keywords Fuzzy clustering · Genetic algorithm · Interval data · Overlap divergence

1 Introduction

Clustering is to divide a set of objects to the groups so that the elements in the same group to have more similarities than others based on a certain characteristic. Cluster analysis is a significant development direction of statistics applied in many various problems (Patel et al. 2011;Arivazhagan et al. 2010;Cheng et al. 2010). Therefore, it is being interested in many statisticians. The object of clustering problem can be

✉ Tai Vovan
vvtai@ctu.edu.vn

Dinh Phamtoan
dinh.pt@vlu.edu.vn

[1] Faculty of Mechanical - Electrical and Computer Engineering, School of Engineering and Technology, Van Lang University, Ho Chi Minh City, Vietnam

[2] College of Natural Science, Can Tho University, Can Tho City, Vietnam

discrete elements, probability density functions, and intervals. Cluster analysis for discrete elements (CAD) has studied the first with a lot of announced results both theory and application (Cabanes et al. 2013; Chen and Hung 2015;Goh and Vidal 2008;Hajjar and Hamdan 2011; Tai and Thao 2018a; Tai and Thao 2018b). With large and complex data such as images, each object needs to be considered to be a distribution, clustering for the probability density functions (CDF) is proposed. Because CDF has more advantages than CAD in some cases of real application, it quickly has been interested in many statisticians. The important results in the recent years for this subject are studied by Pham-Gia et al. (2008); Montanari and Calò (2013); Chen and Hung (2015); Tai et al. (2017); Tai and Thao (2018a). In both CAD and CDF, fuzzy clustering (FC), and non-fuzzy cluster (NFC) have been considered. Besides determining the elements in each cluster as NFC, FC provides the probability of each object to belong to the established clusters.

In many cases of reality, we also found that some types of data are recorded and stored by interval value. For example, data of air pollution, temperature and rainfall amounts, etc. As a result, cluster analysis for intervals (CAI) was proposed. Because CAI has been more and more important applications, it has been distinguished interest in recent years with many works being published. De Souza et al. (2004a) has proposed an algorithm for CAI using the squared Euclidean distance as the criterion to evaluate the difference for two or more intervals. A similar goal was pursued by Masson and Denœux (2004) who proposed a clustering method for CAI based on the dissimilarities of the intervals in the Dempster Shafer theory. Peng and Li (2006) studied the subject of clustering interval data using the extended interval data dissimilarity measures. They discussed various approaches to measure the dissimilarities between intervals, and proposed a clustering model for CAI using the relations between the common and the adjusting distances. Meanwhile, De Carvalho et al. (2007) built clusters for CAI by finding the prototype element for each cluster, and determining the similarity between the elements to this prototype element. This method was developed by Sato-Ilic (2011) for fuzzy CAI with an adaptable varied selection. Hung et al. (2016) built an algorithm for intervals by Hausdorff distance that might give the number of clusters and the probability for assigning the clusters of the intervals. Using the overlap ratio, Kabir et al. (2017) introduced a non-fuzzy algorithm for CAI. Excepting the approach of Hung et al. (2016), the above algorithms did not give the suitable number of groups, and level for the relationship of each interval to clusters. Furthermore, for intervals having high overlap, they often met a lot of limits. One of the methods proposed to solve this disadvantage is the genetic algorithm (GA). The main principle of GA is to build a objective function for the researched problem, and to find the best result for this objective function (Holland 1973). For the clustering, it has shown the outstanding advantage (Tai et al. 2020). Some results of the GA for CDE and CDF have also proposed by Kamel and Selim (1994); Falkenauer (1998); Bandyopadhyay and Maulik (2001); Lai (2005); Liu et al. (2011); Tai et al. (2017); Vovan (2017); Thao and Tai (2017). However, we do not find the research for CAI.

There are four major problems in fuzzy clustering: (i) To find a measure to assess the similarity of the elements, and the difference of clusters, (ii) to determine the appropriate number of groups, (iii) to establish the steps of the algorithm, and (iv) to compute the probability for assigning each element to the established groups. For (i), concerning the CDE and CDF, there were many distances proposed as the criterion to build cluster (Webb 2003;Vovan 2017;Pham-Gia et al. 2008). For CAI, City- Block distance (d_C), Hausdorff distance (d_H), overlap divergence (d_{OID}), and Euclidean distance (d_E) were popularly used. (De Souza et al. 2004a;Ren et al. 2009; Hajjar and Hamdan 2011;Cabanes et al. 2013;Hajjar and Hamdan 2013). Considering the distances as d_C, d_E and d_H, we see that they evaluate the difference of two intervals based on the centers. The overlap area of two intervals does not consider. In many cases, they can not discriminate the resemblance of intervals whereas d_{OID} can do it (Tai et al. 2020, Dinh and Tai 2020). Therefore, in this study, we utilize the overlap divergence (d_{OID}) for CAI. Furthermore, this divergence is also used to create the new index called IDB (see the original DB in Davies and Bouldin (1979)) to make the objective function. For (ii), there were a lot of approaches for CDE (Tai and Thao 2018a) and CDF (Montanari and Calò 2013;Tai and Thao 2018a). However, it did not much consideration for CID. Based on the Euclidean distance, Hung et al. (2016) proposed the approach to find the number of groups. This algorithm met many limitations in real application because it had parameters that were difficult to determine. For (iii), although there were lots of popular algorithms for CDE, CDF, and CAI, GA is not considered much. Tai et al. (2020) has the proposed for CAI, however, the parameters in this algorithm have not been optimized yet. For (iv), the probability for assigning each element to the established clusters has been studied for CDE and CDF (Tai et al. 2017;Tai and Thao 2018b). However, it is limited for CID (Sato-Ilic 2011;Tai et al. 2017). Based on the overlap divergence, we proposed the steps to determine these probabilities.

In this article, combining the improvement from (i), (ii), (iii), and (iv), we propose the Genetic algorithm in fuzzy clustering problem for Interval data (IGI). The IGI can determine the number of clusters, the elements in each cluster, and the probability of assigning the elements into the established clusters at the same time. The convergence of IGI is considered in theory and illustrated by the numerical examples. The algorithm is done quickly and efficiently by the established Matlab procedure. An important contribution of this research is the application in image recognition, an interesting and challenging problem at present. The experiments perform with many kinds of data sets shows reasonableness and outstanding advantages in comparison to others.

The remaining parts of this article are structured as follows. Section 2 presents the related problems to the proposed algorithm and application. The proposed algorithm and its convergence are given in Sect. 3. The illustrative experiments are given in Sect. 4. The application of the developed algorithm to image recognition is investigated in Sect. 5, followed by the conclusion section.

2 The related problems

2.1 The overlap divergence

Definition 1 Given $a = \left[a, \hat{a}\right]$ and $b = \left[b, \hat{b}\right]$ be two univariate intervals, and $c_a = \frac{a+\hat{a}}{2}, r_a = \frac{\hat{a}-a}{2}$ vá $c_b = \frac{b+\hat{b}}{2}, r_b = \frac{\hat{b}-b}{2}$. The overlap divergence between a and b is defined as follows:

Table 1 The distances between a_i and $b = [0; 4]$

Distance	$a_1 = [3; 5]$	$a_2 = [3; 6]$	$a_3 = [5; 6]$	$a_4 = [2; 7]$	$a_5 = [3; 7]$	$a_6 = [2; 6]$	$a_7 = [5; 8]$
d_C	4.00	5.00	7.00	5.00	6.00	4.00	9.00
d_H	3.00	3.00	5.00	3.00	3.00	2.00	5.00
d_E	3.16	3.61	5.39	3.61	4.24	2.83	2.83
d_{OID}	0.67	1.50	3.00	2.00	2.40	1.20	5.00

$$d_O(a,b) = \begin{cases} 0 & \text{if } |c_a - c_b| \le r_b - r_a, \\ (|c_a - c_b| + r_a - r_b)\left(1 - \frac{2r_b}{2r_a+1}\right) & \text{if } |c_a - c_b| \le r_a - r_b, \\ |c_a - c_b| & \text{if} r_a = r_b = 0, \\ (|c_a - c_b| + r_a - r_b)\left(1 - \frac{r_a + r_b - |c_a - c_b|}{2r_a+1}\right) & \text{if } |r_a - r_b| < |c_a - c_b| < r_a + r_b, \\ (|c_a - c_b| + r_a - r_b)\left(1 + \frac{|c_a - c_b| - (r_a + r_b)}{2r_a+1}\right) & \text{if } |c_a - c_b| \ge r_a + r_b, \end{cases} \tag{1}$$

Definition 2 Given two p-dimensional intervals,

$$L = \left(l^1, l^2, \ldots, l^p\right) = \left(\left[l_1, \widehat{l_1}\right], \left[l_2, \widehat{l_2}\right], \ldots, \left[l_p, \widehat{l_p}\right]\right),$$
$$M = \left(m^1, m^2, \ldots, m^p\right) = \left(\left[m_1, \widehat{m_1}\right], \left[m_2, \widehat{m_2}\right], \ldots, \left[m_p, \widehat{m_p}\right]\right), p > 1,$$

In the general case, the overlap divergence between L and M is measured as follows:

$$d_{OID}(L, M) = \sum_{i=1}^{p} \max\left\{d_O\left(l^i, m^i\right), d_O\left(m^i, l^i\right)\right\}.$$

Based on experiments, it is the discovery that the overlap divergence has a lot of benefit more than other distances (d_E, d_C, and d_H). For example, given eight intervals $a_i, i = 1, 7$ and b, then the distances for above intervals are shown in Table 1.

Table 1 expresses that d_{OID} determines clearly level overlap between b and a_i, while d_C, d_E and d_H can not capture the differences between $(a_2; b)$, $(a_4; b)$.

2.2 The indexes to evaluate the built clusters

Definition 3 If N p-dimensional intervals is grouped to c clusters $G_i, i = 1, 2, \ldots, c$, then the Improved Davies and Bouldin (IDB) is defined as follows:

$$IDB = \frac{1}{c}\sum_{i=1}^{c} \max_{i \ne j}\left\{\frac{\frac{1}{|G_i|}\sum\limits_{x \in G_i} d_{OID}(x, \bar{x}_i) + \frac{1}{|G_j|}\sum\limits_{y \in G_j} d_{OID}\left(y, \bar{y}_j\right)}{d_M(\bar{x}_i, \bar{y}_j)}\right\}, \tag{2}$$

where

x and y are the intervals in the clusters G_i and G_j, respectively,
$|G_i|$ and $|G_j|$ are the number of intervals in clusters G_i and G_j, respectively,
$\bar{x}_i$ and $\bar{y}_j$ are the centroid of clusters G_i and G_j, respectively.
d_M is the Minkowski distance.

The IDB is an upgrade of the DB index (Davies and Bouldin 1979), which calculates based on the scattering of intervals within the clusters and separation of the center intervals. The more separated the clusters are, the clustering result more significant is.

Definition 4 The partition coefficient and entropy are used to evaluate the quality of fuzzy clusters. They are given as follows:

$$PC = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{k}\mu_{ji}^{2}, \tag{3}$$

$$PE = -\frac{1}{N}\sum_{i=1}^{N}\sum_{h=1}^{k}\mu_{ji}\log(\mu_{ji}), \tag{4}$$

where k and N are the number of clusters and objects, respectively. The fuzzy matrix of μ_{ji} is calculated by equation (10). The PC index has the value in $[1/k, 1]$. The closer to unity the index the ”crisper” the clustering is.

2.3 Extracting the feature of image

In data processing, images are specially considered by their visibility and application. For example, in agriculture, they are used for automatically classifying fruit (Arivazhagan et al. 2010) or detecting fruit on the tree (Patel et al. 2011). Besides, in the environment, they are used for detecting oil spills (Bora and Gupta 2014). In medicine, they are used for detecting breast cancer in the woman (Cheng et al. 2010). In internet security, they are used for purifying sexy images consisting of the virus. Therefore, in this research, we purely give attention to the image object. Regularly, an image is characterized by three main features: colour, texture and co-occurrence matrix. Among them, using the co-occurrence matrix is widely used for image segmentation or image classification problems (Jain and Vayada 2017). For each data, we apply the extracted data from the grayscale image. The Grey Level Co-occurrence matrix (GLCM) provides information in the spatial arrangement of colours or intensities in an image, characterized by the spatial distribution of intensity levels in a neighbourhood at fixed distance d and orientation θ. If we have an image with size $N_x \times N_y$ (N_x pixels in X-axis and N_y pixels in Y-axis) and G is the domain of grey level, then GLCM is a matrix P size of $G \times G$. Each entry of the $P(i, j)$ holds the count of the number of times that pair of intensities appears in the image with the defined spatial relationship. The formula to compute $P(i, j)$ is presented as follows:

$$p_{d,\theta}(i, j) = \#\left\{(x, y), \left(x', y'\right) \in N_x \times N_y | d = \| (x, y), \left(x', y'\right) \| \right.$$
$$\left. \theta = \Theta\left((x, y), \left(x', y'\right), f(x, y) = i, f\left(x', y'\right)\right) = j\right\}.$$

The representative interval for image is computed as follows.

$$[\mu_x - r_1/2, \mu_x + r_1/2], [\mu_y - r_2/2, \mu_y + r_2/2], \tag{5}$$

where r_1 and r_2 are random numbers uniformly distributed between 0 and 1, respectively, and

$$\mu_x = \frac{1}{N_y}\sum_{j}^{N_y}\left(\frac{1}{N_x}\sum_{i}^{N_x}(i)p_{d\theta}(i,j)\right); \mu_y = \frac{1}{N_y}\sum_{i}^{N_x}\left(\frac{1}{N_x}\sum_{j}^{N_y}(j)p_{d\theta}(i,j)\right).$$

3 The proposed algorithm

Let $X = \{a_1, a_2, \ldots, a_N\}$ be the set of N p-dimensional intervals, and $V^{(t)} = \left\{v_1^{(t)}, v_2^{(t)}, \ldots, v_N^{(t)}\right\}$ be the set of N centroid clusters at iteration t. Then, the fuzzy genetic algorithm in clustering for interval data (IGI) is proposed as follows:

Step 1: Initialize the vector $V^{(0)}$ at time $t = 0$ as follows:

$$V^{(0)} = \left\{v_1^{(0)}, v_2^{(0)}, \ldots, v_N^{(0)}\right\} = \{a_1, a_2, \ldots, a_N\}.$$

Step 2: Update the centroid of intervals by (6):

$$v_i^{(t+1)} = \frac{\sum_{j=1}^{N} f\left(v_i^{(t)}, v_j^{(t)}\right) v_j^{(t)}}{\sum_{j=1}^{N} f\left(v_i^{(t)}, v_j^{(t)}\right)}, i = 1, 2, \ldots, N, \tag{6}$$

where

$$f\left(v_i^{(t)}, v_j^{(t)}\right) = \begin{cases} \exp\left[-\frac{d_{OID}\left(v_i^{(t)}, v_j^{(t)}\right)}{\lambda}\right] & \text{if } d_{OID}\left(v_i^{(t)}, v_j^{(t)}\right) \leq \mu\alpha_{ij}(t), \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

with

* $\alpha_{ij}(t) = \alpha_{ij}(t-1) / \left[1 + \alpha_{ij}(t-1) f\left(v_i^{(t-1)}, v_j^{(t-1)}\right)\right]$ is the factor pattern matrix, $\alpha_{ij}(0) = 1$,

* $\mu = \sum_{i<j} d_{OID}\left(v_i^{(0)}, v_j^{(0)}\right) / \binom{N}{2}$ is the average of $d_{OID}\left(v_i^{(0)}, v_j^{(0)}\right)$,

* $\lambda = \sigma/r, \sigma = \sqrt{\sum_{i<j}\left[d_{OID}\left(v_i^{(0)}, v_j^{(0)}\right) - \mu\right]^2 / \binom{N}{2}}$ is the standard deviation, and r is a constant.

Step 3: Repeat Step 2 until $\max_i\{d_{OID}\left(v_i^{(t+1)}, v_i^{(t)}\right)\} < \varepsilon$. After ending Step 3, there are c elements $v_i^{(t+1)}$.

$v_i^{(t+1)}$ calculated by (6) is expansion or narrowing of $v^{(t)}$ so that the intervals of X will be changed to become centroid intervals. When Step 3 finishes, the intervals in the same cluster will converge to the representative interval. If we have c representative intervals then we also have c clusters.

Step 4: Coding the chromosome by the size of $c.2p$ genes representing for $p-$ dimensional prototypes of c clusters. The genes are assigned to the non-integer values representing the lower/upper bounds of the intervals.

For instance, if we have the minimum and maximum interval of the dataset are $[0, 1]$ and $[1, 3]$, then the value of the chromosome can be coded as $[0.5, 2]$. Expanding for the multi-dimensional case, we have the encryption as Figure 1.

Step 5: Initialize $c.2p$ chromosomes and evaluate their IDB index using Formula (2).

Step 6: Utilize the selection, crossover, and mutation operators:

- **Crossover**: The crossover points can only lie in between two clusters center or the cluster center is considered to be indivisible due to the purpose of a crossover.

  For example, let L_1 and L_2 be the two parent chromosomes of one-dimensional intervals.

  $$L_1 = \{[1, 2];[3, 4]\}\,; L_2 = \{[5, 6]; [7, 8]\}\,; rand = \{[0.2, 0.5];[0.8, 0.2]\}$$

  ($rand$ is the random vector in [0, 1]). Then, the children chromosome is created as follows:

  $$\begin{aligned} Child &= L_1 + rand * 0.85 * (L_2 - L_1) \\ &= \{[1.68,\ 3.7]; [5.72,\ 4.68]\}\,. \end{aligned}$$

- **Mutation**: The mutation operator realizes with probability $\gamma \in [0, 1]$. Let x be the value at a gene location. After mutation, x becomes x' as follows:

  $$x' = \begin{cases} (1 \pm 2\delta).x & \text{if } x \neq 0, \\ \pm 2\delta & \text{if } x = 0, \end{cases}$$

  where δ is a random number in the range [0, 1] generated with uniform distribution, and the '+' or '-' sign occurs with equal probability. For example, from chromosome L_1 and L_2 given above, we calculate the value of the mutational gene at the second position ($\delta = 0.12$ and $\gamma = 0.01$) as follows:

  $$x' = \{[1.68,\ 2.80]; [5.72,\ 4.68]\}\,.$$

- **Selection**: The method of Roulette wheel is used in this operator.

Step 7: Compute the IDB index of the chromosomes achieved in Step 6.

Step 8: Perform Step 5, Step 6, and Step 7 until the current iteration is greater than maximum iteration or

$$\left| IDB^{(t)} - \overline{IDB}^{(t)} \right| \leq \varepsilon$$

where, $IDB^{(t)}$ is the value of objective function at tth iteration and $\overline{IDB}^{(t)}$ is the IDB mean of 100 chromosomes in population. In this algorithm, we choose the number of maximum iteration which is 1000. After Step 8 ends, chromosome G will be published, and it is coded into $U^{(0)}$ matrix, with $U^{(0)} = [\mu_{ik}^{(0)}]$ as follows:

$$\mu_{ik}^{(0)} = \begin{cases} 1 \text{ if } a_k \in G_i \\ 0 \ \text{ otherwise} \end{cases}, 1 \leq k \leq N, 1 \leq i \leq c. \tag{8}$$

For example, the clustering result of four intervals a_1, a_2, a_3, a_4 are given as follows:

$$\{a_1\} \in G_1; \{a_2,\text{a}_4\} \in G_2; \{a_3\} \in G_3$$

Then,

$$G = [1, 2, 3, 2] \rightarrow \begin{cases} \mu_{1\times 1} = 1, \mu_{1\times 2} = 0, \mu_{1\times 3} = 0, \mu_{1\times 4} = 0 \\ \mu_{2\times 1} = 0, \mu_{2\times 2} = 1, \mu_{2\times 3} = 0, \mu_{2\times 4} = 1 \\ \mu_{3\times 1} = 0, \mu_{3\times 2} = 0, \mu_{3\times 3} = 1, \mu_{1\times 4} = 0 \end{cases}$$
$$\Rightarrow U = [\mu_{ik}] = \begin{bmatrix} 1\,0\,0\,0 \\ 0\,1\,0\,1 \\ 0\,0\,1\,0 \end{bmatrix}.$$

Step 9: Initialize the partition matrix w_i, where $\mu_{ik}^{(0)}$ in the first iteration illustrated by (8). The representative element of cluster is determined by (9).

$$\text{w}_i = \frac{\sum_{k=1}^{N} \left(\mu_{ik}^{(h)}\right)^m a_k}{\sum_{k=1}^{N} \left(\mu_{ik}^{(h)}\right)^m}, \tag{9}$$

where $\mu_{ik}^{(h)} \in U^{(h)}, 1 \leq i, j \leq c, 1 \leq k \leq N$ is probability to belong to the i^{th} cluster of the k^{th} element, and w_i is the interval center of c clusters.

Step 10: Update the new partition matrix $U^{(h)}$, where each element of $U^{(h)}$ is determined by the formula (10):

$$\mu_{ik}^{(h)} = \frac{d_{OID}^2(\text{w}_i, a_k)}{\sum_{j=1}^{c} d_{OID}^2\left(\text{w}_j, a_k\right)}, 1 \leq i \leq c, 1 \leq k \leq N, \tag{10}$$

with d_{OID} (w_i, a_k) is the overlap divergence of cluster central w_i and original data a_k .

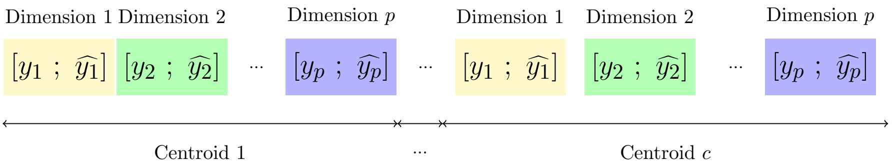

Dimension 1
Dimension 2
Dimension p
Dimension 1
Dimension 2
Dimension p
$[y_1 \; ; \; \widehat{y_1}]$
$[y_2 \; ; \; \widehat{y_2}]$
···
$[y_p \; ; \; \widehat{y_p}]$
···
$[y_1 \; ; \; \widehat{y_1}]$
$[y_2 \; ; \; \widehat{y_2}]$
···
$[y_p \; ; \; \widehat{y_p}]$
Centroid 1
···
Centroid c

Fig. 1 The coding of the chromosomes

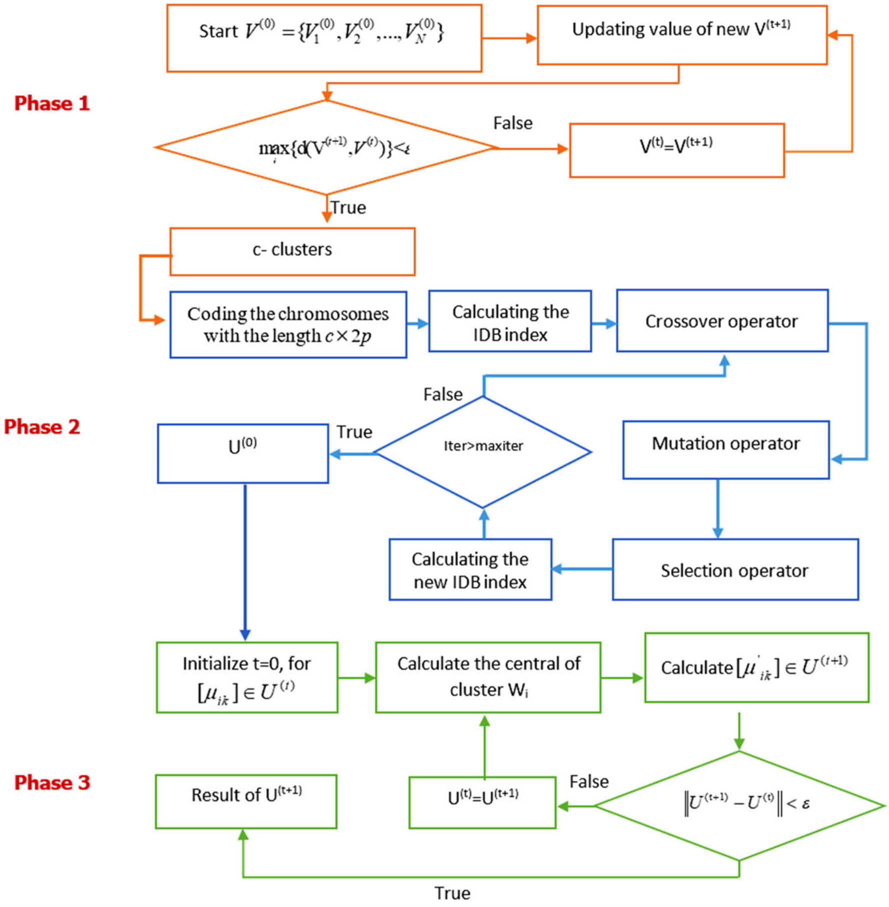

Phase 1
Start $V^{(0)} = \{V_1^{(0)}, V_2^{(0)}, ..., V_N^{(0)}\}$
Updating value of new $V^{(t+1)}$
$\max_i\{d(V^{(t+1)}, V^{(t)})\} < \varepsilon$
False
$V^{(t)}=V^{(t+1)}$
True
c- clusters
Phase 2
Coding the chromosomes with the length $c \times 2p$
Calculating the IDB index
Crossover operator
False
True
$U^{(0)}$
Iter>maxiter
Mutation operator
Calculating the new IDB index
Selection operator
Initialize t=0, for $[\mu_{ik}] \in U^{(t)}$
Calculate the central of cluster W_i
Calculate $[\mu'_{ik}] \in U^{(t+1)}$
Phase 3
Result of $U^{(t+1)}$
$U^{(t)}=U^{(t+1)}$
False
$\|U^{(t+1)} - U^{(t)}\| < \varepsilon$
True

Fig. 2 The flowchart of the IGI algorithm

Step 11: Repeat Step 9 and Step 10 until $\left\|U^{(h)} - U^{(h-1)}\right\| < \varepsilon$.
When Step 11 ends, we obtained the matrix $U^{(h)} = [\mu_{ij}]_{c \times N}$, where μ_{ij} is the probability for signing the i^{th} element to the j^{th} cluster.

The IGI algorithm is given by Fig.2.

In the proposed algorithm, we need to take note of the following problems:

i) In Step 2, r is considered as the variance parameter of the truncated Gauss kernel $f(.)$ given by (10). The larger r is, the larger the standard deviation of each established cluster is taken. Then, the number of clusters for the universal set is otherwise. When $r \to 0$, the data has n clusters, and when $r \to \infty$ the data has only one cluster. Performing with many data sets, we see that $r = 16$ is the most suitable. Therefore, in this study, we choose $r = 16$ in numerical examples.

ii) The value of m in (9) is the fuzziness degree. When $m = 1$, the fuzzy clustering becomes the non-fuzzy clustering. When $m \to \infty$, the partition becomes completely fuzzy with $\mu_{ik} = 1/c$. Generally, it is difficult to determine the optimal m. Although Cannon et al. (1986); Pal and Bezdek (1995); Bora and Gupta (2014) had proposed the rules to determine the supreme of m for the clustering problem, the best value of m has not still determined. In this article, $m = 2$ is chosen.

iii) The value of ϵ in Step 3 and Step 11 are a very small number chosen arbitrarily. The smaller is, the more iterations and computer time are taken. In this article, we choose $\varepsilon = 0.0001$ for all examples.

The proposed algorithm has 3 phases: Phase 1 (Step 1 to Step 3), Phase 2 (Step 4 to Step 8), and Phase 3 (Step 9 to Step 11). Phase 1 determines the suitable number of clusters. At the end of Phase 1, if we obtain c intervals then data is divided into c clusters. The number of clusters c continues to be used in Phase 2. Phase 2 finds the specific intervals for each cluster. When Phase 2 finishes, the intervals in the same cluster will be convergence to a cluster. Phase 3 determines the probability of each element into the established clusters from Phase 2.

The convergence of the IGI occurs when all phases are stopped. The convergence of Phase 1 was proved by Tai et al. (2020). Phase 2 stops when the number of iterations is maximum ($maxiter = 1000$). The convergence of Phase 3 is similar to the fuzzy cluster analysis algorithm for the discrete elements (FCM) that has been proven in many documents as Kamel and Selim (1994).

4 Numerical examples

In this section, we realize two examples to illustrate the IGI algorithm, and to compare it with the others. Example 1 builds the clusters for one-dimension intervals. In this example, the IGI is described step by step. Example 2 establishes the clusters for two-dimensional intervals simulated from the normal distribution. The average of Corrected Rand (CR) index (Hubert and Arabie 1985) is utilized to compare the effectiveness of algorithms.

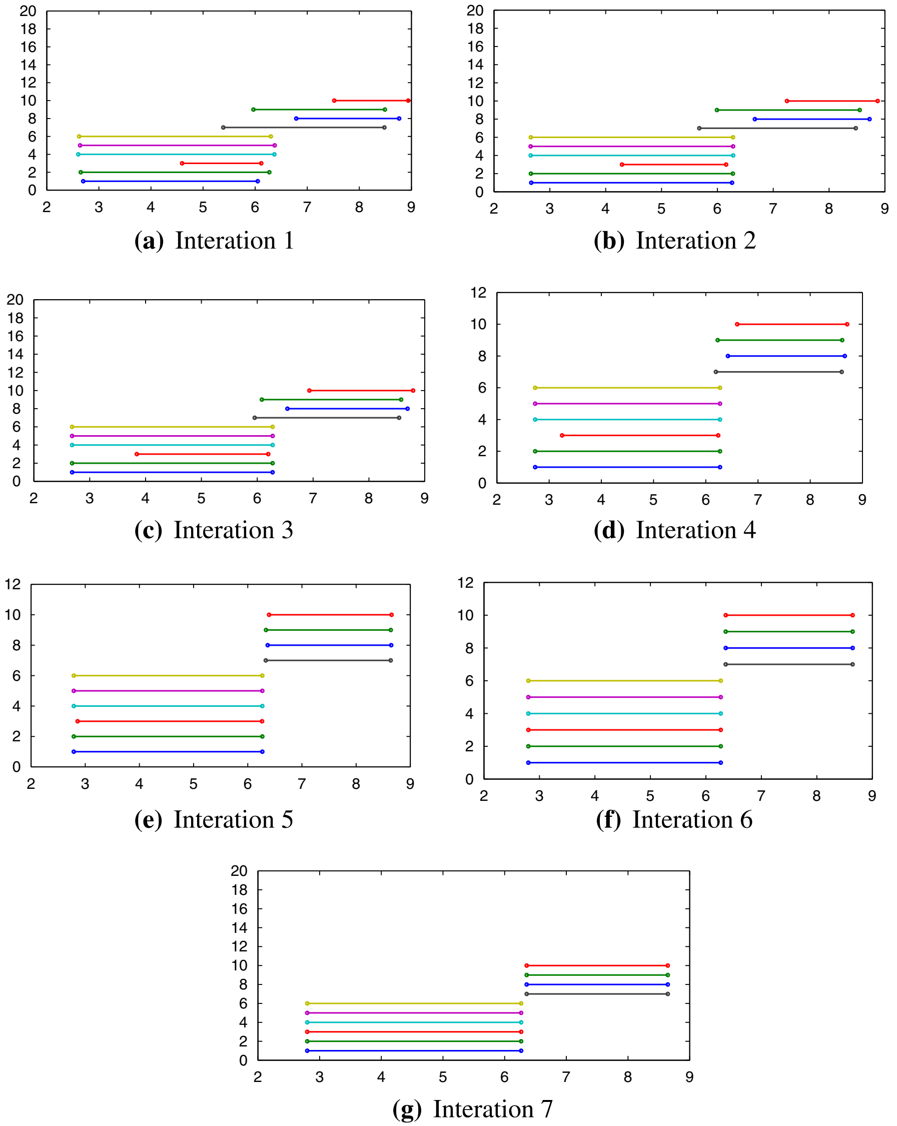
Example 1 Given ten intervals which are signed as follows:

$$a_1 = [3; 5], a_2 = [3; 6], a_3 = [5; 6], a_4 = [2; 7], a_5 = [3; 7],$$
$$a_6 = [2; 6], a_7 = [5; 9], a_8 = [7; 9], a_9 = [6; 8], a_{10} = [8; 9].$$

First of all, we will find the suitable number of clusters for these intervals based on the first phase of the proposed algorithm. After seven iterations, we have the result in Table 2, and illustrated by Fig.3.

Table 2 The number of iterations of 10 intervals in phase 1

Data	iter1	iter2	iter3	iter4	iter5	iter6	iter7
a_1	(2.70;6.05)	(2.67;6.26)	(2.69;6.28)	(2.73;6.27)	(2.79;6.27)	(2.80;6.27)	(2.80;6.27)
a_2	(2.65;6.27)	(2.66;6.28)	(2.68;6.28)	(2.73;6.27)	(2.79;6.27)	(2.80;6.27)	(2.80;6.27)
a_3	(4.60;6.12)	(4.29;6.16)	(3.84;6.20)	(3.25;6.24)	(2.86;6.26)	(2.80;6.27)	(2.80;6.27)
a_4	(2.61;6.37)	(2.66;(6.28)	(2.68;6.28)	(2.73;6.27)	(2.79;6.27)	(2.80;6.27)	(2.80;6.27)
a_5	(2.64;6.38)	(2.66;6.28)	(2.68;6.28)	(2.73;6.27)	(2.79;6.27)	(2.80;6.27)	(2.80;6.27)
a_6	(2.62;6.30)	(2.66;6.28)	(2.68;6.28)	(2.73;6.27)	(2.79;6.27)	(2.80;6.27)	(2.80;6.27)
a_7	(5.39;8.48)	(5.68;8.48)	(5.96;8.54)	(6.19;8.60)	(6.33;8.64)	(6.36;8.64)	(6.36;8.65)
a_8	(6.79;8.76)	(6.67;8.73)	(6.54;8.69)	(6.42;8.66)	(6.37;8.65)	(6.36;8.65)	(6.36;8.65)
a_9	(5.97;8.49)	(5.99;8.55)	(6.08;8.58)	(6.23;8.61)	(6.34;8.64)	(6.36;8.64)	(6.36;8.65)
a_{10}	(7.52;8.94)	(7.25;8.87)	(6.93;8.79)	(6.60;8.71)	(6.39;8.65)	(6.36;8.65)	(6.36;8.65)



(a) Interation 1

(b) Interation 2

(c) Interation 3

(d) Interation 4

(e) Interation 5

(f) Interation 6

(g) Interation 7

Fig. 3 The convergence of 10 intervals to two clusters

Table 2 and Fig.3 show that the adequate quantity of clusters is two.

Performing Phase 2, the value of IDB index is given by Fig.4.

When Phase 2 stops, we have the two clusters: $G_1 = \{a_1, \ldots, a_6\}; G_2 = \{a_7, \ldots, a_{10}\}$. Phase 3 gives the probability to assign elements into cluster G_1 (μ_1) and cluster G_2 (μ_2). The result of this phase when using the proposed distance (IGI), Hausdorff distance (IGI-H), Euclidean distance (IGI-E), and City-block distance (IGI-C) are presented in Table 3.

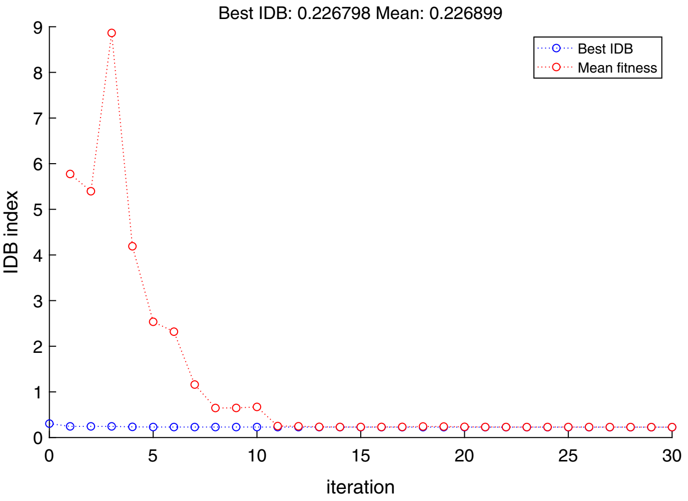


Fig. 4 The value of IDB index for 30 iterations

Table 3 The probability of each element to assign into clusters using the different distances

Data	IGI		IGI-C		IGI-E		IGI-H	
	μ_1	μ_2	μ_1	μ_2	μ_1	μ_2	μ_1	μ_2
a_1	0.99	0.01	0.97	0.03	0.95	0.05	0.89	0.11
a_2	1.00	0.00	1.00	0.00	1.00	0.00	0.99	0.01
a_3	0.80	0.20	0.78	0.22	0.66	0.34	0.57	0.43
a_4	0.99	0.01	0.93	0.07	0.95	0.05	0.97	0.03
a_5	0.99	0.01	0.97	0.03	0.96	0.04	0.96	0.04
a_6	1.00	0.00	0.98	0.02	0.97	0.03	0.97	0.03
a_7	0.15	0.85	0.12	0.88	0.16	0.84	0.24	0.76
a_8	0.00	1.00	0.01	0.99	0.01	0.99	0.01	0.99
a_9	0.02	0.98	0.06	0.94	0.05	0.95	0.03	0.97
a_{10}	0.02	0.98	0.05	0.95	0.06	0.94	0.07	0.93

CR index of IGI-H is 0.8, CR index of IGI-C, IGI-E, and IGI are the same (CR =1). Moreover, Table 3 shows that the partition of elements into G_1 and G_2 of the IGI algorithm is clearer than the ones. For instance, μ_1 of IGI, IGI-C, IGI-E, and IGI-H are 0.99, 0.97, 0.95, and 0.89, respectively. CR index of the proposed algorithm and the others are presented in Table 4.

It can be observed from Table 4 that the algorithm of De Souza et al. (2004b) and De Souza et al. (2004a) are unsuitable in this case when providing the lowest CR value, whereas all of the remaining methods perform well, in which the IGI provides the best result with $CR = 1$. For studies of Hung et al. (2016), De Souza et al. (2004a) and De Souza et al. (2004b), they only performed the stage of clustering for interval data, so they do not have the outcomes of PC and PE. Moreover, based on the PC and PE indexes, the fuzzy clustering of Dinh and Tai (2020) obtains the second-best

Table 4 CR index of the IGI and others for ten intervals

Method	No.cluster	CR index	PC	PE
Hung et al. (2016)	2	0.950	–	–
De Souza et al. (2004a)	3	0.890	–	–
De Souza et al. (2004b)	3	0.890	–	–
FCM	2	1.000	0.869	0.233
Carvalho et al. (2017)	2	1.000	0.869	0.233
Rodriguez et al. (2019)	2	1.000	0.890	0.205
Dinh and Tai (2020)	2	1.000	0.915	0.147
IGI-C	2	1.000	0.893	0.200
IGI-E	2	1.000	0.873	0.226
IGI-H	2	1.000	0.855	0.251
Proposed	2	1.000	0.929	0.129

with 0.915 of PC and 0.147 of PE. The other methods also have quite good result with the value of PC from 0.855 to 0.893, and the value of PE from 0.147 to 0.251. The developed algorithm has the best result for PC and PE because PE gives the largest value, and PE obtains the smallest value.

Example 2 This example examines the interval data given in Bustince et al. (2016). The dataset consist three clusters with 100 intervals for each group. The intervals are generated from the two-dimensional normal distribution with the following parameters:

$$\text{Cluster 1}: \mu = [0; 3]; \sigma = \begin{bmatrix} 2 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}; \text{Cluster 2}: \mu = [3; 0]; \sigma = \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix};$$
$$\text{Cluster 3}: \mu = [-3; 0]; \sigma = \begin{bmatrix} 2 & -0.5 \\ -0.5 & 0.5 \end{bmatrix}.$$

Based on the above distributions, the rectangular intervals (Fig.5) are derived from the following rules:

$$[x_1 - r_1/2, x_1 + r_1/2], [x_2 - r_2/2, x_2 + r_2/2],$$

where x_1, x_2 are generated based on the cluster distributions, and r_1, r_2 are random numbers with a uniform distribution in [1;4].

Performing the first phase of the developed algorithm, after five iterations, all intervals converge to three cluster centers highlighted by 3 black rectangles in Fig.5.

Next, Phase 2 of the IGI algorithm continues to be run with $c = 3$. Its result is shown in Fig.6.

When Phase 2 finishes, we obtain the best value of the objective function with $IDB = 0.307$ and three clusters as follows:

$$C_1 = \{a_1, a_2, \ldots, a_{100}\}; G_2 = \{a_{101}, a_{102}, \ldots, a_{150}\}; G_3 = \{a_{151}, a_{152}, \ldots, a_{300}\}.$$

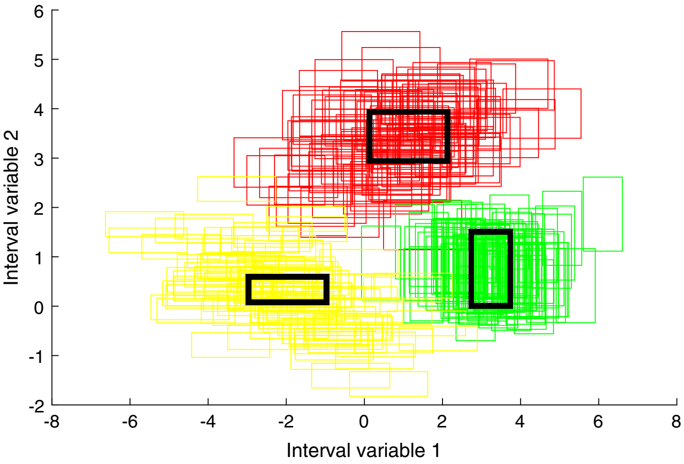


Fig. 5 300 interval data and convergence to three clusters

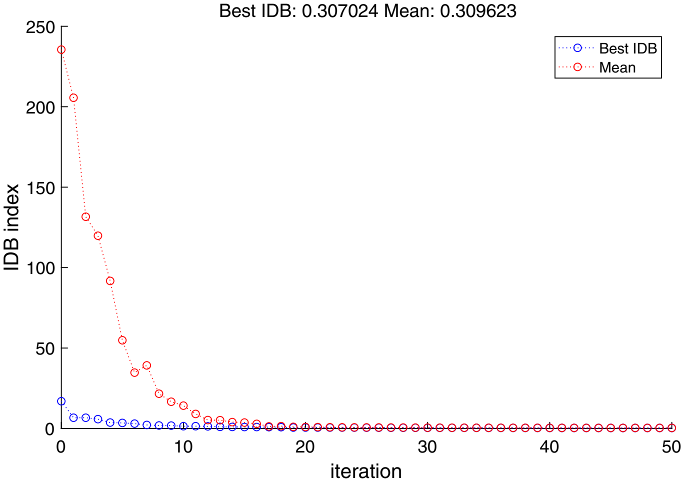


Fig. 6 The convergence of IDB index after fifty iterations

Fig.7 gives the probability belong to three clusters of the elements when Phase 3 ends.

Comparing the CR index of IGI algorithm and others, we have the Table 5.

Table 5 shows that all results of the CR index are very high. The CR index obtained by Dinh and Tai (2020) and Hung et al. (2016) are higher than that obtained by other methods. The CR of IGI is the highest. Furthermore, PC and PE of IGI are the best. In our opinion, using the new objective function and measure are the reasons for the good result of the proposed algorithm. In the next section, we will apply the proposed algorithm for the image recognition problem.

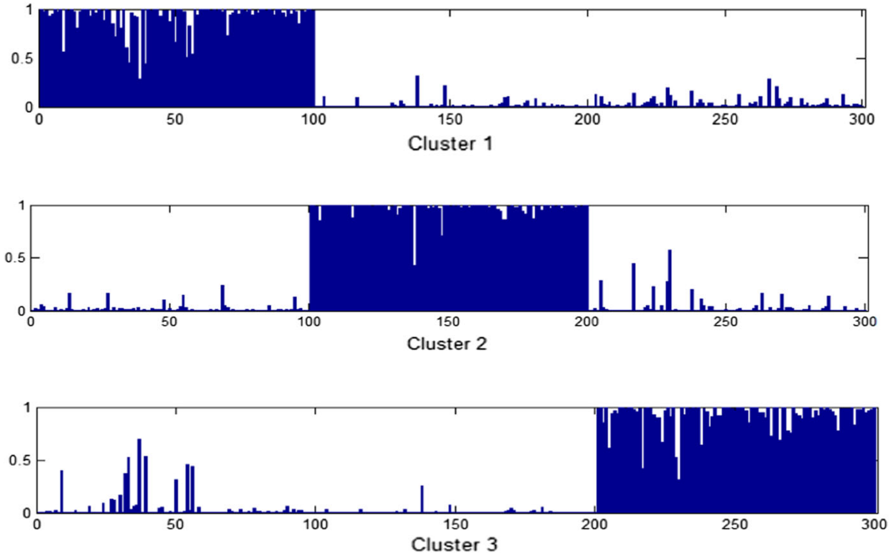


Fig. 7 The probability of 300 intervals belonging to three clusters

Table 5 The CR index of the algorithms in clustering for 300 intervals

Method	No. cluster	CR index	PC	PE
Hung et al. (2016)	3	0.983	–	–
De Souza et al. (2004b)	3	0.949	–	–
De Souza et al. (2004a)	3	0.949	–	–
FCM-O	3	0.949	0.783	0412
Carvalho et al. (2017)	3	0.951	0.821	0.348
Rodriguez et al. (2019)	3	0.953	0.879	0.316
Dinh and Tai (2020)	3	0.995	0.945	0.132
IGI-C	3	0.949	0.787	0.405
IGI-E	3	0.949	0.783	0.412
IGI-H	3	0.953	0.821	0.348
IGI (Propsoed)	3	1.000	0.989	0.012

5 Applying in image classification

5.1 Benchmark dataset 1: cat and Apricot blossom images

Data 1 has ten images detailed in Appendix. The images are divided into two groups. They are Cat and Apricot blossom with five images for each group. Some sample images of the two groups are given in Fig.8

Firstly, we extract the images to become the two-dimension intervals based on (5). The extracted intervals are presented in Table 6.

(a) Cat

(b) Apricot blossom

Fig. 8 The image samples of Cat and Apricot blossom

Table 6 The extracted data of 10 images

Image	Interval variable 1	Interval variable 2
1	[5.493, 9.493]	[6.493, 8.493]
2	[6.111, 8.111]	[5.111, 9.111]
3	[4.596, 8.596]	[5.596, 7.596]
4	[5.917, 7.917]	[6.417, 7.417]
5	[5.310, 8.310]	[6.310, 7.310]
6	[5.446, 6.446]	[4.448, 7.448]
7	[4.546, 5.546]	[3.048, 7.048]
8	[3.521, 5.521]	[4.021, 5.021]
9	[5.023, 6.023]	[4.528, 6.528]
10	[4.348, 5.348]	[3.348, 6.348]

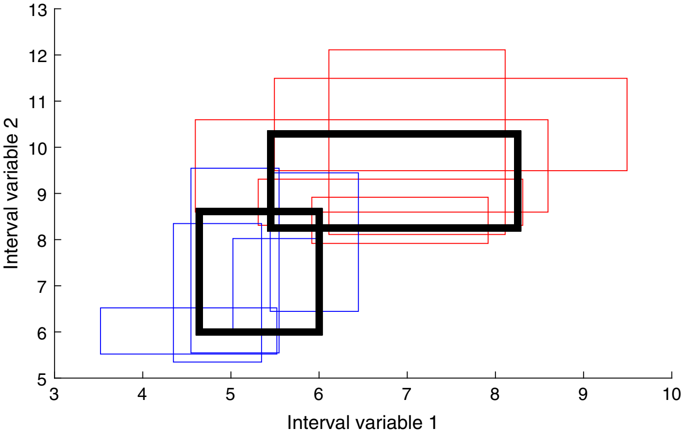


Fig. 9 Intervals extracted from 10 images and convergence of the IGI algorithm in Phase 1

Secondly, we determine the number of clusters for intervals. After seven interactions of Phase 1, we obtain $c = 2$, and the cluster centres are highlighted by the black rectangle shown in Fig.9.

After determining the suitable number of groups is 2 ($c = 2$), the IGI algorithm in Phase 2 starts with $t = 0$ and until the number of iterations obtained equal 1000. As a

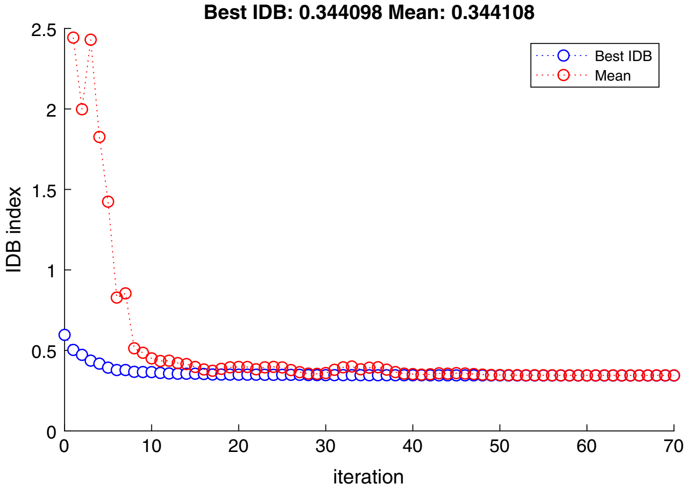


Fig. 10 The convergence of IDB after 70 iterations

Table 7 The probability of 10 elements belonging to three clusters

P	μ_1	μ_2
I_1	0.011	0.989
I_2	0.046	0.954
I_3	0.024	0.976
I_4	0.019	0.981
I_5	0.012	0.988
I_6	0.812	0.188
I_7	0.990	0.010
I_8	0.937	0.063
I_9	0.969	0.031
I_{10}	0.993	0.007

result, we have the best value of the objective function with $IDB = 0.3441$, and two clusters (See Fig.10).

$$G_1 = \{a_1, \ldots, a_5\}; G_2 = \{a_6, \ldots, a_{10}\}.$$

The result of the IGI algorithm in Phase 3 is presented in Table 7 and Fig.11: where, μ_1 vá μ_2 is the probability of each element which belongs to G_1 and G_2 .

Comparing the CR, PC and PE indices of the IGI and other algorithms, we obtain Table 8.

Table 8 shows the CR index of Hung et al. (2016) and the IGI algorithm are better than others as De Souza et al. (2004b) and De Souza et al. (2004a).

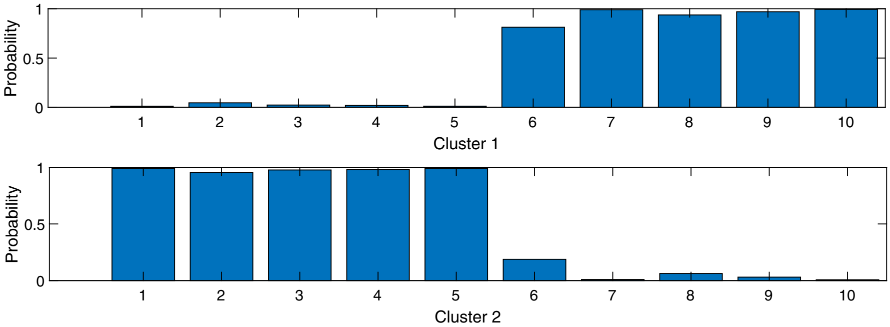


Fig. 11 The bar graph of probability belongs to 3 clusters of 10 images

Table 8 The CR, PC and PE indices of the algorithms in fuzzy clustering of data 1

Method	No. cluster	CR index	PC	PE
Hung et al. (2016)	2	1.000	–	–
De Souza et al. (2004b)	2	0.601	–	–
De Souza et al. (2004a)	2	0.601	–	–
FCM	2	0.950	0.844	0.280
Carvalho et al. (2017)	2	0.960	0.857	0.260
Rodriguez et al. (2019)	2	0.980	0.853	0.262
Dinh and Tai (2020)	2	1.000	0.926	0.143
IGI-C	2	1.000	0.852	0.257
IGI-E	2	1.000	0.828	0.291
IGI-H	2	1.000	0.900	0.192
IGI (Propsoed)	2	1.000	1.000	0.000

5.2 Benchmark dataset 2: sensitive and insensitive images

The considered images are divided into two groups: Sensitive and Insensitive with 49 and 50, respectively Table (9). This data is provided from https://drive.google.com/drive/folders/1Hp89pVYVUlhAVxRc8A3ExQVgZRsdys9n?usp=sharing

Some sample images of the three groups are shown in Fig.12.

After six iterations of (see Fig. 13), two clusters are established (see Fig.14).

Continue to perform Phase 2 of IGI with 50 iterations (see Fig. 15), we obtain the results as follows:

- The best value of objective function: $IDB = 0.378$.
- The optimal partition:

$$G_1 = \{a_1, \ldots, a_{49}\}; G_2 = \{a_{50}, \ldots, a_{99}\}.$$

The CR index of the algorithms are given in Table 10.

(a) Sensitive **(b)** Insensitive

Fig. 12 Sample images of two groups : **a** Sensitive **b** Insensitive

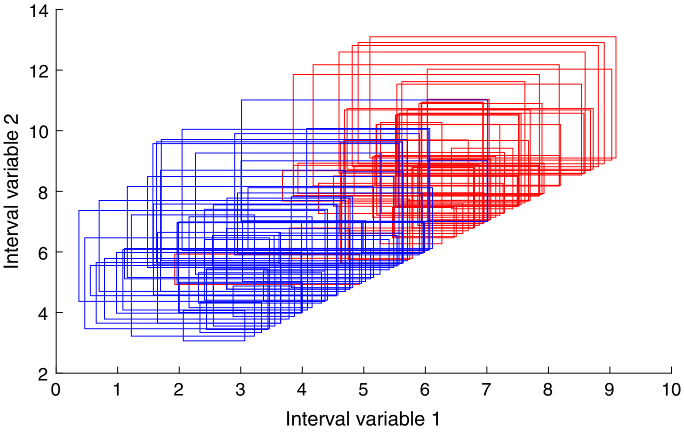


Fig. 13 The scatter plot of 99 intervals are extracted from images

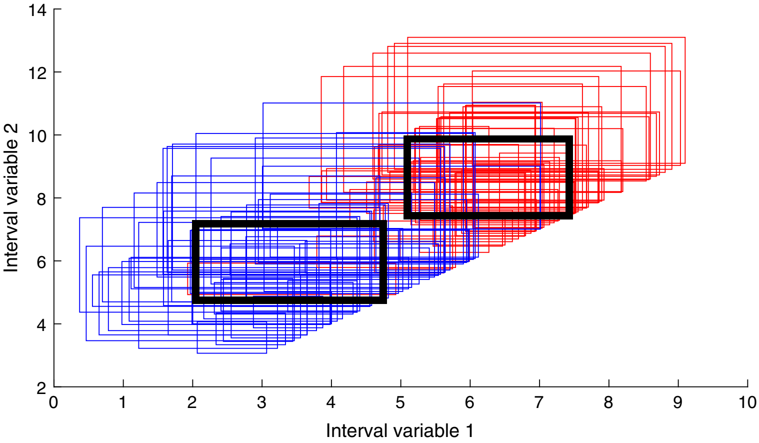


Fig. 14 The convergence of the IGI in phase 1 for 99 images

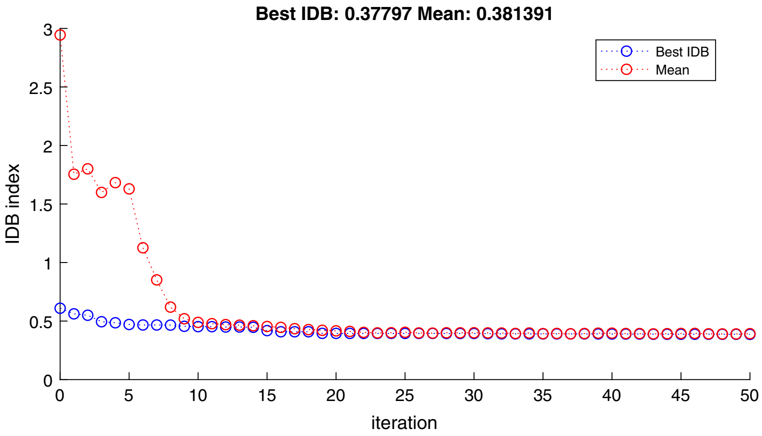


Fig. 15 The convergence of the IGI index after 50 iterations

Table 9 The CR index of the IGI and other methods in clustering for 99 images

Method	No. cluster	CR index	PC	PE
Hung et al. (2016)	2	0.958	–	–
De Souza et al. (2004b)	2	0.958	–	–
De Souza et al. (2004a)	2	0.959	–	–
FCM	2	0.980	0.857	0.255
Carvalho et al. (2017)	2	1.000	0.838	0.238
Rodriguez et al. (2019)	2	0.980	0.883	0.215
Dinh and Tai (2020)	2	1.000	0.921	0.143
IGI-C	2	0.960	0.876	0.226
IGI-E	2	0.960	0.857	0.255
IGI-H	2	1.000	0.838	0.283
IGI	2	1.000	1.000	0.000

Table 9 again demonstrates the superiority of the IGI over other algorithms. The probability belonging to clusters is presented by Fig. 16.

5.3 Benchmark dataset 3: lotus, Gazania and passion images

The dataset is considered in this experiment to have 519 images divided into three groups with 192 Lotus, 76 Gazania, and 251 Passion flower images. It is provided from https://drive.google.com/drive/folders/1tJGVpGebzrI4t5OiPfAm9dVO_beWqymP?usp=sharing

The sample images is illustrated in Fig. 17.

Fig. 18 shows the extracted intervals from images.

Running Step 1 to Step 4, we have the Fig. 19.

Performing Phase 2, after 70 iterations, we have Fig. 20.

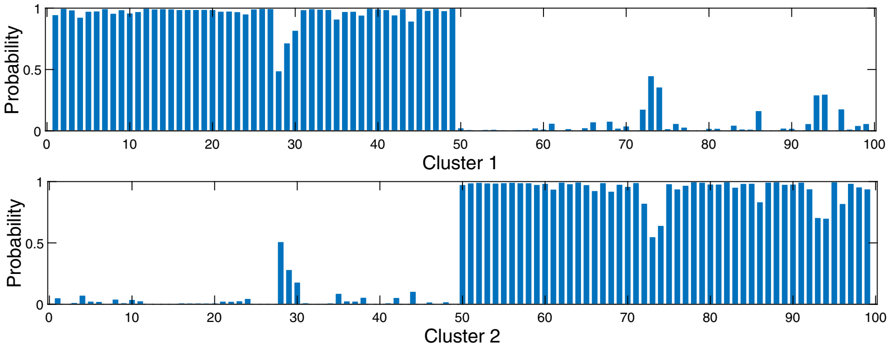
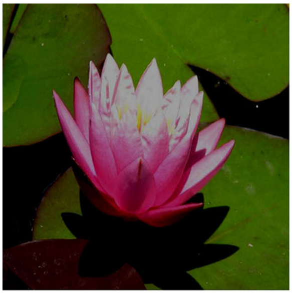
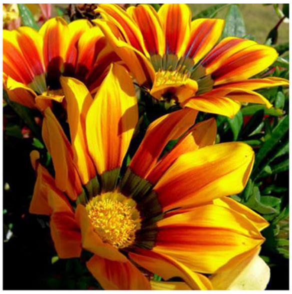


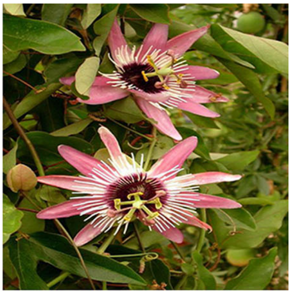
Fig. 16 The probability of 99 images belongs to two clusters



(a) Lotus flower



(b) Gazania flower



(c) Passion flower

Fig. 17 Sample images of three groups

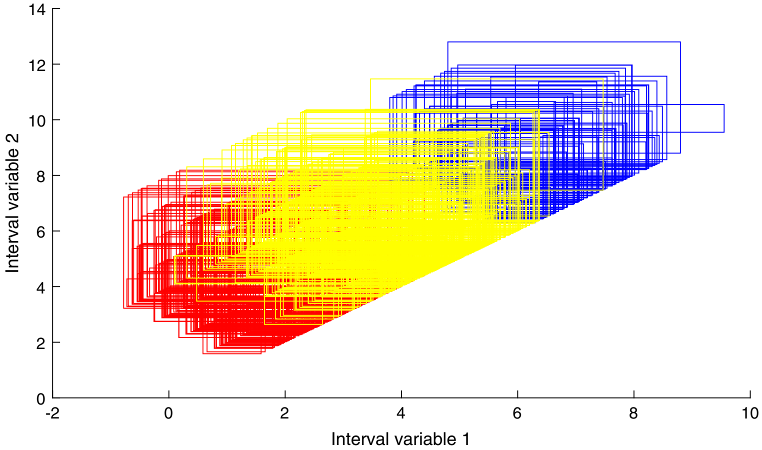


Fig. 18 The scatter of 519 extracted intervals

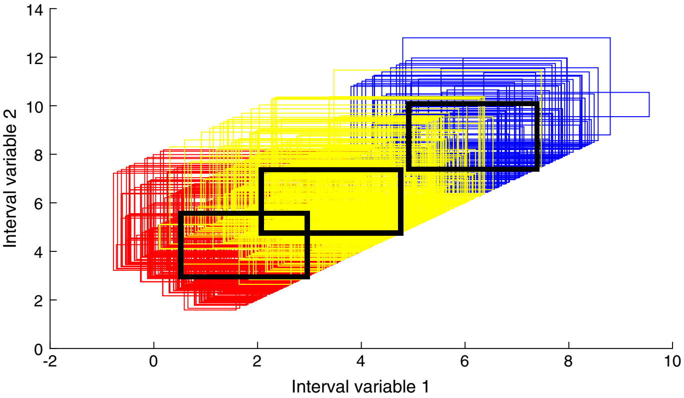


Fig. 19 The convergence of 519 intervals into three cluster's centroid

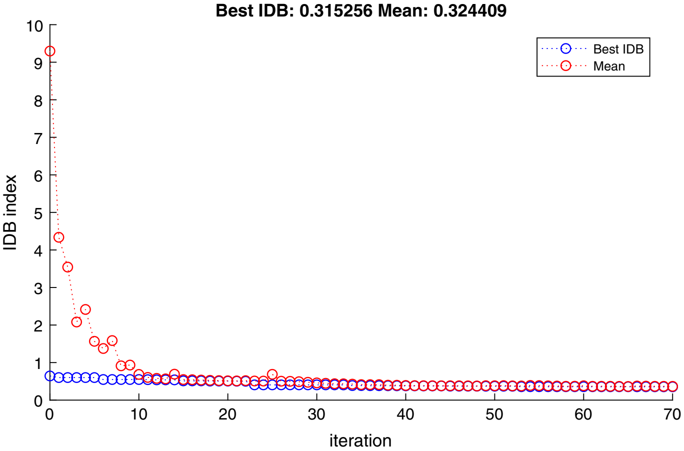


Fig. 20 The convergence of IDB index after 70 iterations

At that time, we have
* The best value of the objective function: $IDB = 0.378$.
* The clustering result:

$$G_1 = \{a_1, \ldots, a_{192}\}; G_2 = \{a_{193}, \ldots, a_{268}\}; G_3 = \{a_{269}, \ldots, a_{519}\}.$$

The final phase gives the probability to belong to clusters of each element in Fig. 21.

The CR index of the algorithms are given in Table 10.

According to the results obtained from three experiments with five data sets, they show that the proposed algorithm is more advantageous than the existing ones. With the data sets having different characteristics (number of dimensions and elements), the proposed algorithm always produces the best result. It is also very stable in clustering, especially for complex data such as images. Besides the above considered data sets,

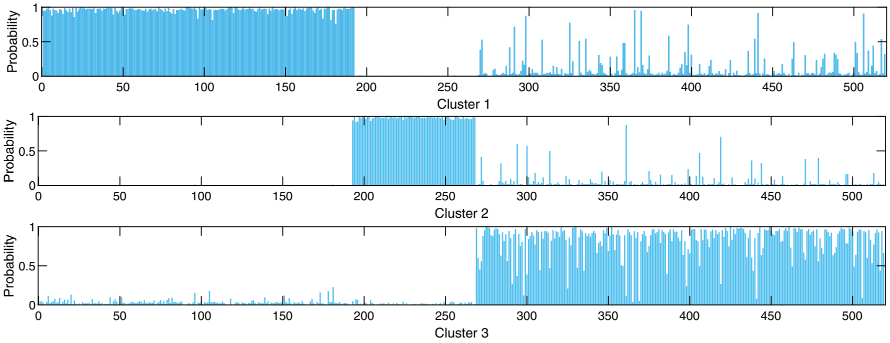


Fig. 21 The probability to belong to three clusters of 519 images

Table 10 The CR index of the IGI and other methods in clustering for 519 images

Method	No.cluster	CR index	PC	PE
Hung et al. (2016)	3	0.969	–	–
De Souza et al. (2004b)	3	0.933	–	–
De Souza et al. (2004a)	3	0.932	–	–
FCM	3	0.933	0.760	0.453
Carvalho et al. (2017)	3	0.939	0.748	0.475
Rodriguez et al. (2019)	3	0.864	0.781	0.417
Dinh and Tai (2020)	3	0.995	0.937	0.124
IGI-C	3	1.000	0.788	0.404
IGI-E	3	1.000	0.600	0.453
IGI-H	3	1.000	0.797	0.342
IGI	3	1.000	0.959	0.102

we have implemented a lot of other data. All of them have given the competitive results in comparison with the existing algorithms.

6 Conclusion

This study has the contribution for both theory and application. Using the overlap divergence to evaluate the similarity of intervals, the new index for the objective function, and the adjusted operations from the original genetic algorithm, this article proposes an intelligent fuzzy genetic algorithm in clustering for interval data. In this algorithm, the appropriate number of groups, the intervals of each cluster, and the probability for each element belonging to the clusters is performed at the same time. We have established the Matlab procedure to run the proposed algorithm quickly and effectively. The proposed approach has shown the outstanding advantages in comparing to others through many complex data sets having the difference in characters and the number of intervals. These are interesting and appealing applications that we will continue to perform in the next time.

Acknowledgements The authors would like to thank Van Lang University, Vietnam for funding this work.

Appendix. The images of Data 1

References

Arivazhagan S, Shebiah RN, Nidhyanandhan SS, Ganesan L (2010) Fruit recognition using color and texture features. J Emerg Trends Comput Inf Sci 1(2):90–94

Bandyopadhyay S, Maulik U (2001) Nonparametric genetic clustering: comparison of validity indices. IEEE Trans Syst Man Cybernet Part C 31(1):120–125

Bora DJ, Gupta AK (2014) Impact of exponent parameter value for the partition matrix on the performance of fuzzy c means algorithm. arXiv preprint. arXiv:1406.4007

Bustince H, Barrenechea E, Pagola M, Fernandez J, Xu Z, Bedregal B, Montero J, Hagras H, Herrera F, De B (2016) A historical account of types of fuzzy sets and their relationships. IEEE Trans Fuzzy Syst 24(1):179–194

Cabanes G, Bennani Y, Destenay R, Hardy A (2013) A new topological clustering algorithm for interval data. Pattern Recognit 46(11):3030–3039

Cannon RL, Dave JV, Bezdek JC (1986) Efficient implementation of the fuzzy c-means clustering algorithms. IEEE Trans Pattern Anal Mach Intell 2:248–255

Chen JH, Hung WL (2015) An automatic clustering algorithm for probability density functions. J Statist Comput Simul 85(15):3047–3063

Cheng HD, Shan J, Ju W, Guo Y, Zhang L (2010) Automated breast cancer detection and classification using ultrasound images: a survey. Pattern Recognit 43(1):299–317

Davies DL, Bouldin DW (1979) A cluster separation measure. IEEE Trans Patt Anal Mach Intell 2:224–227

De Carvalho FDA, Pimentel JT, Bezerra LX (2007) Clustering of symbolic interval data based on a single adaptive L^1 distance. Neural Networks, 2007 International Joint Conference: 224–229. https://doi.org/10.1109/IJCNN.2007.4370959

De Souza RM, De Carvalho FDA (2004) Clustering of interval data based on city-block distances. Pattern Recognit Lett 25(3):353–365

De Souza RM, de Carvalho FDA, Silva FC (2004) Clustering of interval-valued data using adaptive squared euclidean distances. In: International Conference on Neural: 775–780. https://doi.org/10.1007/978-3-540-30499-9_119

De Carvalho FDA, Simões EC (2017) Fuzzy clustering of interval-valued data with city-block and hausdorff distances. Neurocomputing 266:659–673

Dinh PT, Tai VV (2020) Automatic fuzzy genetic algorithm in clustering for images based on the extracted intervals. Multimedia Tools and Applications: 1–23 (2020). https://doi.org/10.1007/s11042-020-09975-3

Falkenauer E (1989) Genetic algorithms and grouping problems. Wiley, New York

Goh A, Vidal R (2008) Clustering and dimensionality reduction on riemannian manifolds. In: IEEE Conference on Computer Vision and Pattern Recognition: 1–7 https://doi.org/10.1109/CVPR.2008.4587422

Hajjar C, Hamdan H (2011) Self-organizing map based on hausdorff distance for interval-valued data. IEEE International Conference on Systems, Man, and Cybernetics: 1747–1752. https://doi.org/10.1109/ICSMC.2011.6083924

Hajjar C, Hamdan H (2013) Interval data clustering using self-organizing maps based on adaptive mahalanobis distances. Neural Netw 46:124–132

Holland JH (1973) Genetic algorithms and the optimal allocation of trials. SIAM J Comput 2(2):88–105

Hubert L, Arabie P (1985) Comparing partitions. J Classif 2(1):193–218

Hung WL, Yang JH, Shen KF (2016) Self-updating clustering algorithm for interval-valued data. Fuzzy Syst 2:1494–1500

Jain M, Vayada MG (2017) Non-cognitive color and texture based image segmentation amalgamation with evidence theory of crop images. Signal Process Security 160–165

Kabir S, Wagner C, Havens TC, Anderson DT, Aickelin U (2017) Novel similarity measure for interval-valued data based on overlapping ratio. Fuzzy Systems, 2017. In: IEEE International Conference, pp.1–6

Kamel MS, Selim SZ (1994) New algorithms for solving the fuzzy clustering problem. Pattern Recognit 27(3):421–428

Lai CC (2005) A novel clustering approach using hierarchical genetic algorithms. Intell Autom Soft Comput 11(3):143–153

Liu Y, Wu X, Shen Y (2011) Automatic clustering using genetic algorithms. Appl Math Comput 218(4):1267–1279

Masson MH, Denœux T (2004) Clustering interval-valued proximity data using belief functions. Pattern Recognit Lett 25(2):163–171

Montanari A, Calò DG (2013) Model-based clustering of probability density functions. Adv Data Anal Classificat 7(3):301–319

Pal NR, Bezdek JC (1995) On cluster validity for the fuzzy c-means model. IEEE Trans Fuzzy syst 3(3):370–379

Patel HN, Jain R, Joshi MV (2011) Fruit detection using improved multiple features based algorithm. Int J Comp Appl 13(2):1–5

Peng W, Li T (2006) Interval data clustering with applications. In: Tools with Artificial Intelligence, 18th IEEE International Conference on IEEE: 355–362. https://doi.org/10.1109/ICTAI.2006.71

Pham-Gia T, Turkkan N, Tai VV (2008) Statistical discrimination analysis using the maximum function. Commun Stat Simul Comput 37(2):320–336

Ren Y, Liu YH, Rong J, Dew R (2009) Clustering interval-valued data using an overlapped interval divergence. Proc Eighth Australasian Data Min Conf 101:35–42

Rodriguez SIR, De Carvalho FDA (2019) A new fuzzy clustering algorithm for interval-valued data based on City-Block distance. In: 2019 IEEE International Conference on Fuzzy Systems, pp. 1–6. https://doi.org/10.1109/FUZZ-IEEE.2019.8859017

Sato-Ilic M (2011) Symbolic clustering with interval-valued data. Proc Comp Sci 6:358–363

Tai VV, Thao NT (2018) Similar coefficient for cluster of probability density functions. Commun Statist Theory Methods 47(8):1792–1811

Tai VV, Thao NT (2018) Similar coefficient of cluster for discrete elements. Sankhya B 80(1):19–36

Tai VV, Trung N, Vo-Duy T, Ho-Huu V, Nguyen-Trang T (2017) Modified genetic algorithm-based clustering for probability density functions. J Statist Comput Simulat 87(10):1964–1979

Tai VV, Dinh PT, Tuan LH, Thao NT (2010) An automatic clustering for interval data using the genetic algorithm. Annals of Operations Research, pp. 1–22. https://doi.org/10.1007/s10479-020-03606-8

Tai VV (2017) L^1-distance and classification problem by Bayesian method. J Appl Statist 44(3):385–401

Thao NT, Tai VV (2017) A new approach for determining the prior probabilities in the classification problem by Bayesian method. Adv Data Anal Classif 11(3):629–643

Webb AR (2003) Statistical Pattern Recognition. John Wiley & Sons

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.