Journal of Systems Engineering and Electronics
Vol. 31, No. 6, December 2020, pp.1245 – 1253

Automatic fuzzy-DBSCAN algorithm for morphological and overlapping datasets

YELGHI Aref[1,*], KÖSE Cemal[2], YELGHI Asef[3], and SHAHKAR Amir[4]

1. Department of Computer Engineering, Avrasya University, Trabzon 61250, Turkey; 2. Department of Computer Engineering, Karadeniz Technical University, Trabzon 61080, Turkey; 3. Department of Business Administration, Gazi University, Ankara 06560, Turkey; 4. Civil Engineering Department, Karadeniz Technical University, Trabzon 61080, Turkey

Abstract: Clustering is one of the unsupervised learning problems. It is a procedure which partitions data objects into groups. Many algorithms could not overcome the problems of morphology, overlapping and the large number of clusters at the same time. Many scientific communities have used the clustering algorithm from the perspective of density, which is one of the best methods in clustering. This study proposes a density-based spatial clustering of applications with noise (DBSCAN) algorithm based on the selected high-density areas by automatic fuzzy-DBSCAN (AFD) which works with the initialization of two parameters. AFD, by using fuzzy and DBSCAN features, is modeled by the selection of high-density areas and generates two parameters for merging and separating automatically. The two generated parameters provide a state of sub-cluster rules in the Cartesian coordinate system for the dataset. The model overcomes the problems of clustering such as morphology, overlapping, and the number of clusters in a dataset simultaneously. In the experiments, all algorithms are performed on eight data sets with 30 times of running. Three of them are related to overlapping real datasets and the rest are morphologic and synthetic datasets. It is demonstrated that the AFD algorithm outperforms other recently developed clustering algorithms.

Keywords: clustering, density-based spatial clustering of applications with noise (DBSCAN), fuzzy, overlapping, data mining.

DOI: 10.23919/JSEE.2020.000095

1. Introduction

Clustering is one of the techniques in data mining. The main aim of clustering is to divide a given data (point) into similar groups while dissimilar groups contain the dissimilar data. Clustering is useful in data mining, document retrieval, image segmentation, pattern classification and statistic [1,2]. In the field of knowledge discovery in databases, clustering is defined as an unsupervised learning technique, because there is no prior knowledge about the dataset for analyzing. Scientific community focuses on the popular techniques and develops them by its own methods with effectiveness and efficiency in data mining. Using fuzzy c-means clustering and neural network, a new multiple model adaptive control method was proposed, which led to the stability of the control switching system [3]. Pulse description words of detected signals are performed by the density-based spatial clustering of applications with noise (DBSCAN) algorithm which shows the identification time of key target signals. It can adapt to the complex signal environment with noise intervention and overlapping signals, and is not susceptible of the loss of local pulse parameters [4]. Local-DBSCAN (LDBSCAN) was proposed to distinguish the false targets (FTs) from the physical targets (PTs) after compensating the FTs time delays, while PTs possess small distribution [5]. Different from the other work which focused on estimating the density of each sample using different kinds of density estimators, a clustering algorithm named adapative DBSCAN was developed based on inherentpropertiesofthenearestneighborgraph[6].Anewalgorithm NRDD-DBSCAN based on the DBSCAN algorithm was presented using resilient-distributed datasets (RDDs) to explore the outliers which influence the data quality of IoT [7].

Many clustering algorithms have been presented with the different points of view and subject intersection in a related field, while each algorithm has its own advantage and disadvantage, and all of them have not been able to solve the complex, overlapping, heterogeneous outlier datasets simultaneously. In [8], a multi-stage model for anomaly detection was proposed to remove the problem of DBSCAN. The other work [9] proposed a method of non-triangle inequality (non-TI) clustering in the context of social network, which used the distance function in quantum logic-based query language.

Manuscript received January 28, 2020.
*Corresponding author.

Generally, the clustering algorithms are divided into eight categories which include partition, hierarchy, fuzzy theory, distribution, density, graph theory, grid and fractal theory model [10]. The K-means algorithm and its framework, with the concept of partition for the developing of other algorithms, is one of the popular algorithms, which is frequently used in the data mining field. K-means algorithm failed, when it had faced with the arbitrary shape in spatial data and it could only minimize an amount of cost functions which lead to convergence in the local minimum [11]. K-medoids [12], partitioning around medoids (Pam) [13] and clustering large applications (CLARA) [14] are also based on partitioning and reduce the sensitivity in terms of noise, but the algorithms fail to address arbitrarily shaped clusters in their strategy.

Some of the algorithms based on hierarchical clusterings such as balanced iterative reducing and clustering using hierarchies (BIRCH) algorithm [15], clustering using representatives (CURE) [16], receiver operating characteristic (ROC) curve [17] and Chameleon [18] from the perspective of tree perform arbitrarily shaped clusters. They show better results than others. Nonetheless, they are not able to overcome the time complexity.

To overcome the complexity structure, some algorithms based on the hierarchical clustering algorithm were proposed. One of them presented an automatic version based on hierarchical clustering [19]. The leaders-subleaders algorithm [20] assures the potentiality and significance of hierarchical clustering in terms of time series in DataStream. The considering of the Euclidean distance discloses the existing of the density problem in a large dataset. In order to tackle them, shared nearest neighbor (SNN) was proposed in [21] which was based on the graph model. The algorithm finds the nearest neighbors of each data point and then redefines the similarity between pairs of points in connection with a number of nearest neighbors. DBSCAN [22] is based on density data with two parameters, one of which is the number of neighbors and the other is the radius of the node. It will be discussed in detail in the next section.

Optics [23] is an algorithm proposed in order to overcome the problem of DBSCAN. For solving the problem of detecting meaningful clusters in various densities of data, the algorithm is considered within the scope of two concepts: one is the maximum distance for each core point and the other is the density of each core point. Spatial-temperal DBSCAN (ST-DBSCAN) [24] was constructed by modifying the DBSCAN algorithm, by using the density factor, which tried to find noise data when clusters have dissimilar densities in the spatial-temporal dataset. Enhanced DBSCAN [25] is also based on DBSCAN proposing local epsilon instead of the global epsilon. The Gaussian mixture model (GMM) [26], based on kernel density, estimates the density region with a small number of components and allows for arbitrary clustering. Another algorithm Gaussian density distance (GDD) [27] presents a new clustering method without prior information and parameters by combining the Gaussian kernel and Euclidian distance. By using the rough set theory, rough-DBSCAN [28] overcomes the runtime scanning dataset with the perspective of density.

In order to take the density based on arbitrary shape clusters and overcome the time consuming problem, the algorithm uses the core leader for the selection of the high-density data. Another kind of clustering is fuzzy c-means (FCM) clustering [29], which is basic for developing fuzzy clustering. Fuzzy neighborhood (FN-DBSCAN) [30] based on the DBSCAN algorithm was introduced by using the fuzzy neighborhood relation concept instead of the crisp neighborhood relation. The FN-DBSCAN algorithm gives more robust results than DBSCAN.

The spectral clustering (SC) algorithm is one of the density-based clustering algorithms, and clusters data with the graph method. The algorithm has demonstrated the clustering on the unstructured dataset [31]. Identifying density-based local outliers factor (LOF) converts binary concept to degree concept (degree formula) for each data in the dataset. Then it uses degree formula for distinguishing the near data (normal data in the cluster) and the far data (outlier data) [32]. Breunig et al. [33] used the fuzzy proximity relations between data points in order to gain the dissimilar dense clusters without any priori knowledge of a dataset. The aim of this study is to consider the advantage and disadvantage of the clustering algorithm, which have been mentioned above, and then propose a algorithm which overcomes the limits. This paper conducts discovering with the gaps in clustering by considering the accuracy clustering on three problems (overlapping, morphology and the number of clusters). The proposed algorithm automatic fuzzy-DBSCAN (AFD) has tried to overcome morphology, overlapping and the number of clusters at the same time. It is initialized with two parameters and generates two parameters Eps2 (second epsilon) and Pos (epsilon position for merging and separating) during running which are defined by the state of sub cluster rules. Although many algorithms have been proposed without parameters, they ignore the mentioned problems.

In the experiment, we show and compare the performance of our algorithm regarding accuracy by external indices [34, 35] such as rand index (RI), adjusted rand index (ARI) and f-measure (F) [36]. Our experiment is performed on the three real and five synthetic datasets. We

will also describe the DBSCAN and FN-DBSCAN algorithms (our algorithm is inspired from them) in Section 2. We propose an algorithm with the generated state of sub cluster rules during running. It is written by symbolic set in Section 3. The performance of AFD with regard to accuracy and representation is also demonstrated by comparison of well-known algorithms in Section 4 and finally, this research's conclusions are described in Section 5.

2. Related work

2.1 DBSCAN

DBSCAN is able to distinguish the noise data and classify the arbitrary shape dataset. It includes two parameters: epsilon (Eps1) and the minimum number of points/data (MinPoints), which are based on a user defined neighbor radius and the existing number of points related to the radius. DBSCAN can be conceptually described as follows. The neighborhood is specified by different types of distance functions. For two points p and q and their distance dist(p,q), the epsilon of point p is defined by $\{q \in D|\text{dist}(p,q) \leqslant \text{Eps}\}$ which indicates the radius of it. A core object denotes that a point which is its epsilon contains at least a minimum number of points (MinPoints) (see Fig. 1).

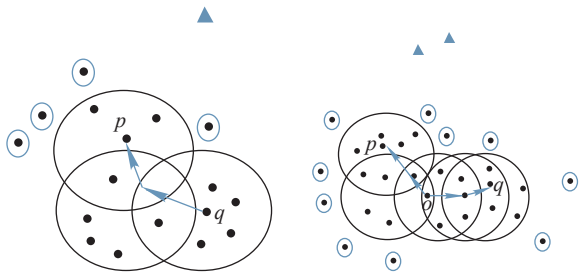


Fig. 1 Basic notion in density clustering

Let $C = \{C_1, \cdots, C_k\}$ be the clusters with respect to Eps1 and MinPoints in the dataset D. We define outlier objects as a set of the objects far from the core object in dataset D not belonging to any cluster C_i, i.e., outlier $= \{p \in D|\forall i : p \notin D\}$. If it is not a core object or outlier object, it would be the border which is density-reachable from a core object (see Fig. 1). An object p is directly density-reachable from the object q, when p is within the epsilon-neighborhood of q and q is also a core object. $p \in N_{Eps}(q)$ where $N_{Eps}(q)$ is the subset of the dataset which includes Eps-neighborhood of q, and $|N_{Eps}(q)| > MinPts$ (core object condition).

Data p is density-reachable from the data q in connection with Eps1 and MinPoints if there is a sequence object $P = \{p_1, p_2, \cdots, p_n\}$, $p_1 = q$ or $p_n = p$ such that $p_i + 1$ is directly density-reachable from p_i with respect to Eps1 and MinPoints, for $1 \leqslant i \leqslant n, p_i \in D$ (see Fig. 1(a)). An object p is density-connected to the object q with respect to Eps1 and MinPoints, if there is an object $o \in D$ then both p and q are density-reachable from o (see Fig. 1(b)). Density connection is a symmetric relation. A cluster C is a non-empty subset of D satisfying the following requirements [22]:

(i) $\forall p,q$: if $q \in C$ and p is density-reachable from q with respect to Eps1 and MinPoints, then $p \in C$. (maximality)

(ii) $\forall p,q \in C$: p is density-connected to q with respect to Eps1 and MinPoints. (connectivity) [22].

2.2 FN-DBSCAN

In this algorithm, four formulas are defined for the extensional DBSCAN by FN-DBSCAN. Here, the neighborhood set of points $N_x(y)$ is shown as follows:

$$F(x;\varepsilon_1) = \{y \in D|N_x(y) \geqslant \varepsilon_1\}, \quad x \in D. \tag{1}$$

$N_x : D \to [0,1]$ is any membership function that determines the neighborhood relation between data (Cartesian plane) that is done by (2). ε_1 is different from the definition of the radius in DBSCAN. Instead of distanced-based DBSCAN, the new level-based set with the fuzzy neighborhood set is used and shown below [30]. Here, k is the value coefficient and $k > 0$ affects the neighborhood radius. Let $Y = \{y_1, \cdots, y_n\}$ and the exponential neighborhood relation and formula are shown in Fig. 2 and (2) respectively.

$$N_x(Y) = \exp\left(-\left(k\frac{d(x,y)}{d_{\max}}\right)^2\right) \tag{2}$$

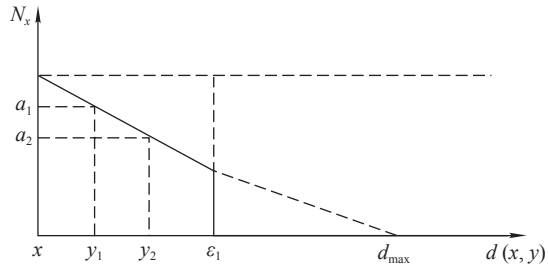


Fig. 2 Exponential neighborhood relation [30]

Each core point x is defined with the neighborhood membership function which is shown in the following equation. $d(x,y)$ is the Euclidian distance between two points x and y. $d_{\max}$ is the maximum distance between two points in the coordinate system [30]. For simplicity, in (3) a point/data is defined with the neighbor degree to all points in the dataset, and (4) shows the concept of fuzzy cardinality [30].

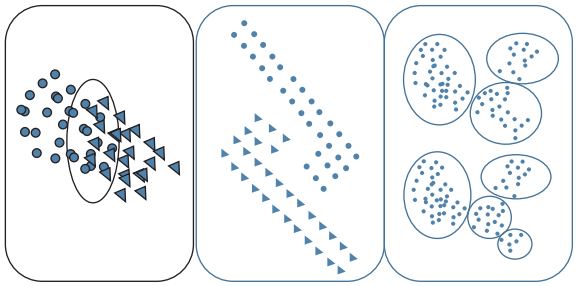
$$N_x=\begin{cases}1-\dfrac{d(x,y)}{d_{\max}}, & d(x,y)\leqslant\varepsilon_1\\ 0, & \text{otherwise}\end{cases} \tag{3}$$

$$\text{Card FN}(x;\varepsilon_1):\sum_{y\in N(x;\varepsilon_1)}N_x(y)\geqslant\varepsilon_2 \tag{4}$$

The algorithm using $\varepsilon_1,\varepsilon_2$ and MinPoints, has tried to convert the distance-based DBSCAN to level-based DBSCAN. However, those features are density-based clustering.

3. AFD algorithm

In this paper, we propose clustering using the features of DBSCAN and fuzzy for solving the number of clusters, overlapping and morphological problems, as shown in Fig. 3. The upside of the subfigures presents three clusters and the downside of them presents four clusters.



(a) Overlapping clusters (b) Morpological clusters (c) Number of clusters in a dataset

Fig. 3 Three problems

Firstly, our algorithm selects the high-density region with (2) (fuzzy value) and then all data are scanned from the left to the right coordinate system by DBSCAN with predefined parameters. Scanning gains the core nodes from the dataset using Eps1 and MinPoints, which leads to categorization of the sub clusters. Secondly, the sub clusters are sent to the A_rule function in order to decide the merging and separating of them. The function generates two parameters such as second epsilon (Eps2) and epsilon position (Pos) automatically and pass to the B_rule function. The generated parameters are done by using the arrangement of the location of sub clusters in a dataset space (Cartesian coordinate system).

All the state locations with the center of sub-clusters are defined in the coordinate system. The states are Minimum, Maximum, Dmean and Diff (see Definition1). Merging and separating are done based on the four state locations. Our algorithm by using the concept of fuzzy cardinality and fuzzy neighborhood, used in (1) and (2) respectively and in the concept of the DBSCAN algorithm, proposes a new point of view for the problem of clustering, which is mentioned above. Data, in order to trim the scaling, are converted to normal data [30] by

$$X_{ij}=\frac{X_{ij}-X_j^{\min}}{(X_j^{\max}-X_j^{\min})},\quad j=1,\cdots,m \tag{5}$$

where

$$X_j^{\min}=\min_{i=\overline{1,n}}x_{i,j}, \tag{6}$$

$$X_j^{\max}=\max_{i=\overline{1,n}}x_{i,j},\quad j=1,\cdots,m. \tag{7}$$

The Iris dataset has 150 instances with four attributes such as sepal length, sepal width, petal length and petal width which are defined in three classes.

Definition 1 State locations are defined as follows and Fig. 4 also presents an example of the Iris dataset.

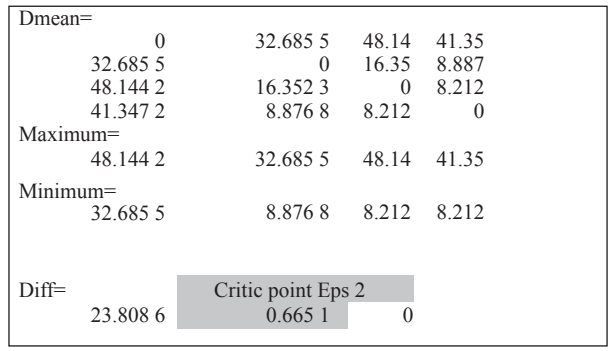


Fig. 4 An example of Iris dataset

Dmean: The distance matric value between the center of sub clusters.

Maximum: The maximum value of each column in Dmean.

Minimum: The minimum value of each column in Dmean.

Diff: The difference value between each column in Minimum.

Definition 2 Find peaks (Pos): return a vector local maximum and minimum value as peaks. A local peak is a data sample that is either higher than its two neighboring samples (here, one sub cluster with its two neighboring sub clusters) or lower than them. See Fig. 5(e) findpeak = 3 and Fig. 5(f) findpeak = 2.

Definition 3 Binary separation: return the binary code from each state of sub clusters to each other such as Minimum, Maximum and Average of both. The binary code for them will be done by left sub clusters to right sub clusters (scanning dataset from left to right) if the left sub cluster is less than the right one, then it takes zero, otherwise one for the station. See Fig. 5(d), Fig. 5(e) and Fig. 5(f) and results of them as follows:

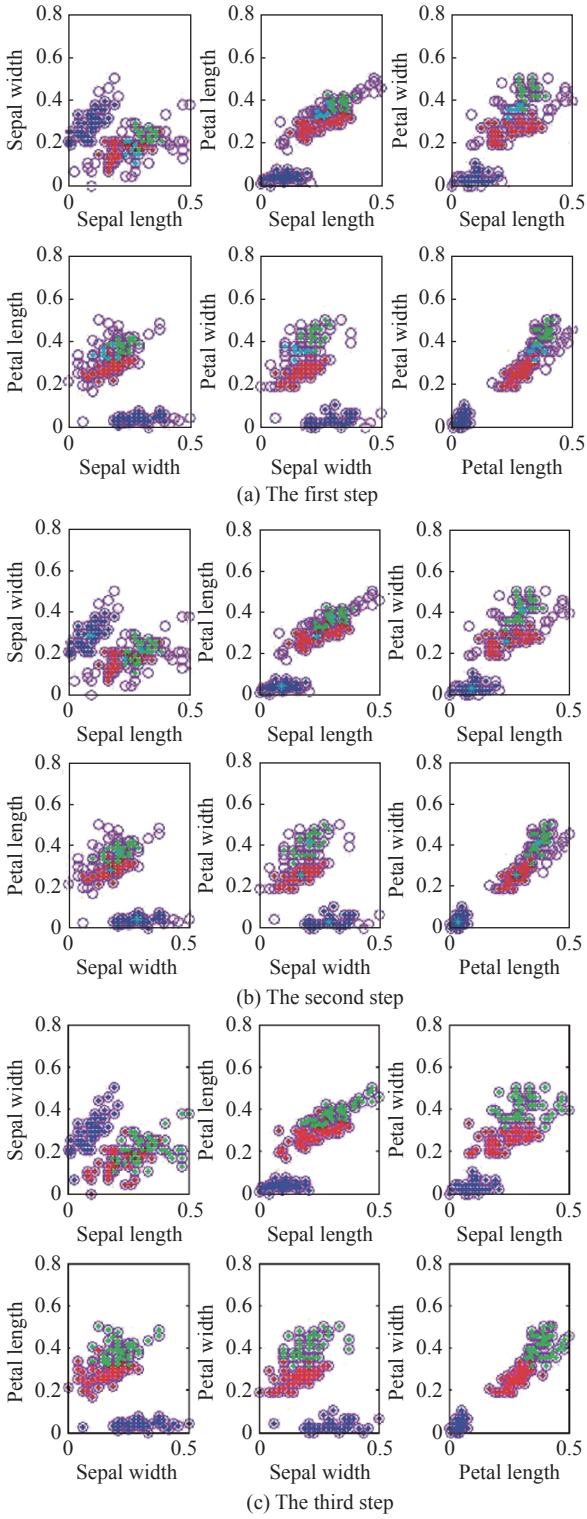
(i) Fig. 5(d) by taking the maximum peak: BMA = Minimum = 101;

(ii) Fig. 5(e) by taking the maximum peak: BME =

Means = 101;

(iii) Fig. 5(f) by taking the minimum peak: BMI = Maximum = 111;

The metric measures are $F = 0.960\,0$, RandIndx = 0.949 5, AdjRandIndx = 0.885 7 on the IRIS dataset. $*$ is used for better presentating the peak.



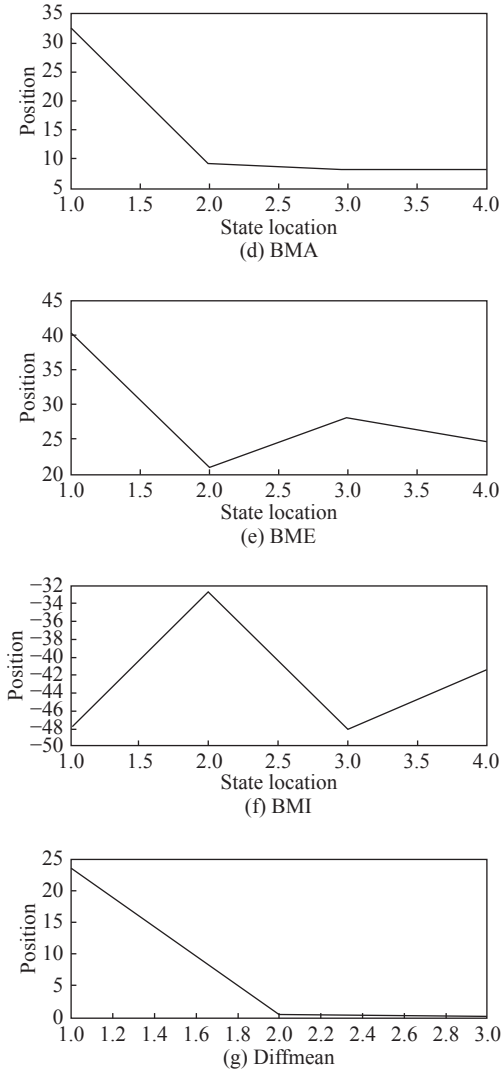


Fig. 5 Dataset performed by the proposed algorithm

The proposed algorithm is presented below and written in nine lines, which could be divided into three steps (see Algorithm 1). Lines 1–6 are written for the first step, as shown in Fig. 5(a). Lines 7 and 8 refer to the second step as shown in Fig. 5(b). For a better understanding of Lines 7 and 8, two rule are described (see Algorithm 2 and Algorithm 3). In these lines based on the gained Eps2, values have been tried to merge and separate the sub clusters and line 9 is the deciding data, which does not belong to any clusters, as shown in the third step and Fig. 5(c). Rule A is divided into five parts: The first part is defined as the sets and the rest of them are sub rules, which happens during the running algorithm. Sub rule 0 (line 1) checks the completely separable clusters. Other sub rules (lines 3–5) check different conditions and gain Eps2. Rule B with eight lines based on Eps and Diff is assigned to carry out the separation or merging of sub clusters.

Algorithm 1 AFD

BEGIN

1. Require: d-dimensional data set (D), Number of data

points (N), Eps1, and MinPoints parameters.

2. Normalizing the data points with (5).

3. Gaining neighborhood degree of normalized data points with (2).

4. Providing sub clusters with DBSCAN (by using Eps1 and MinPoints parameters).

5. For all sub clusters gain the mean of sub clusters.

6. Calculating the distance between per mean of sub clusters (called as Dmean).

7. Finding the Eps2 and Pos with Dmean (Call the rule_A function).

8. Merging or separating sub clusters with Eps2 (call the rule_B function).

9. Obtaining the mean of sub clusters and clustered remaining of data points (not belong to any cluster).

END

Algorithm 2 Rule_A function

BEGIN

1. Use Definition 1 and Definition 2 for finding peak of them, then create set for them.

2. Use the binary check of sets and calculate intersect with them.

3. Obtaining the peak of the Minimum set intersected with the maximum set.

Eps2 = Diff(position = peak of the minimum set)

4. Obtaining the peak of the minimum set and no any peak in the maximum set.

Eps2 = Diff(position = peak of the minimum set)

5. Obtaining the peak of the minimum set intersected with the minimum set.

Eps2 = Diff(position = intersect the peak of minimum and minimum sets)

Return Eps2, Pos

END

Algorithm 3 Rule_B function

BEGIN

1. Based on the Eps2 set

2. For I each of Eps2 set

3. For J each of Diff set

4. If Eps(I) $\geqslant$ Diff(J)

5. Create new cluster

6. End

7. End

8. End

Return Set of sub cluster

END

For better representing the dataset, we show three graphics (three classes in 2-dimentional based on the two attributes, as shown in Fig. 5(a), Fig. 5(b) and Fig. 5(c)). In running, with initialization of the parameters Eps1 = 0.53, MinPts = 5, we gain states of sub clusters, which are guided to decide the merging and separating of sub clusters. Fig. 5(d), Fig. 5(e), Fig. 5(f) and Fig. 5(g) are generated by Rule_A with Eps2 = 0.664 9 and Pos = 2, see Fig. 5(g) and Fig. 4. Here, Eps2 = 0.664 9 is the same as Eps2 = 0.665 1 in Fig. 4. For separation, we need the amount less than one (e.g., Eps2 = Eps2 – 0.000 2). Then Rule_B is used for separating and merging sub clusters which means if Eps2 = 0.664 9 < Eps2 = 0.665 1 (critic point), it leads to generating two new sub clusters. See Figs. 6–8 for observing clustering steps on the Iris dataset.

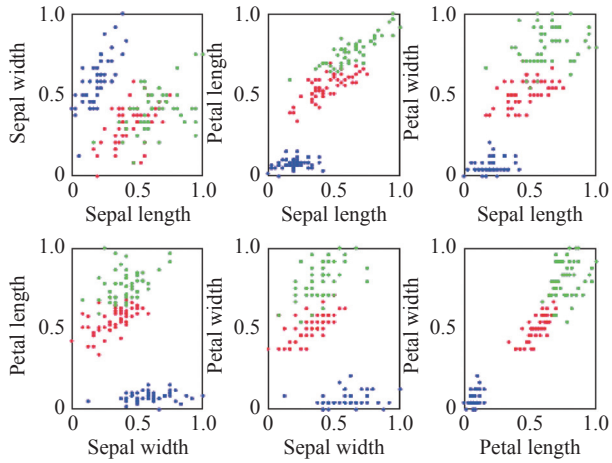


Fig. 6 Original Iris dataset

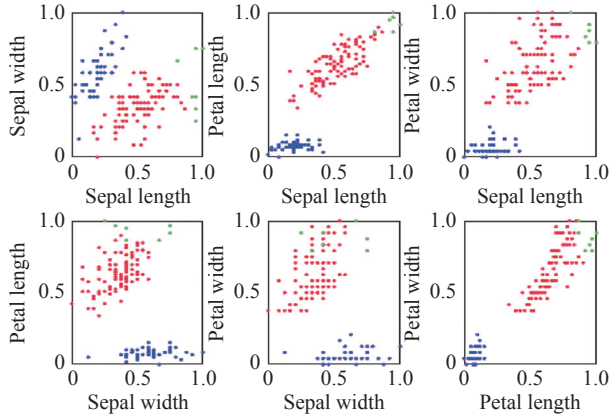


Fig. 7 Spactral clustering on Iris dataset with sigma = 2

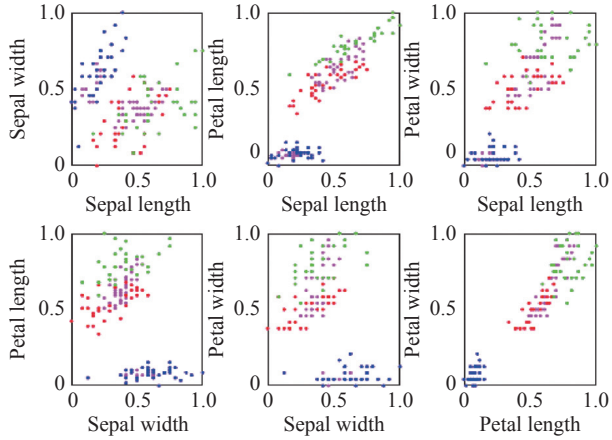


Fig. 8 LOF clustering on Iris dataset with Minpts = 10 and Eps = 1

4. Experimental results

The experiment is performed on three real and five synthetic datasets (see Fig. 9). The real datasets are obtained from the University of California, Irvine (UCI) Machine Learning Repository such as Wine, Glass and Iris datasets that are suitable for testing overlapping datasets and for testing the morphology of datasets. We use synthetic datasets. To compare algorithms, features based on partial, fuzzy, density clustering are considered.

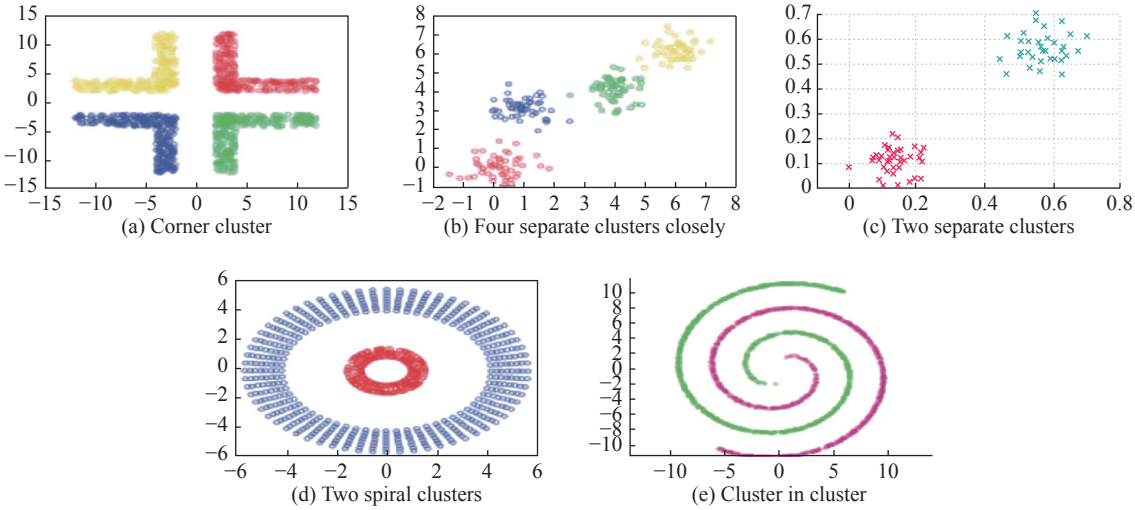


Fig. 9 Synthetic datasets

They are also popular in the clustering field, but the disadvantages of them are not overcome to one of the mentioned problems. In this paper, our algorithm is compared with well-known DBSCAN, fuzzy means and K-means with the average of 30 times running. For measuring the performance of the algorithm regarding accuracy on datasets, well-known measures such as rand, adjusted rand and F-measure have been used. In Table 1, the maximum values of indices are bolded and Table 2 shows their parameters.

Table 1 All algorithms with rand, adjusted rand and F-measure

Name	Index	Wine	Glass	Iris	Cluster in cluster	Corner separate	Four separate cluster	Two separate cluster	Two spiral
ADF	RandIndx	**9.28E-01**	7.96E-01	**9.42E-01**	**1**	**1**	**1**	**1**	**1**
	AdjRandIndx	**8.39E-01**	**4.87E-01**	**8.68E-01**	**1**	**1**	**1**	**1**	**1**
	F-measure	9.48E-01	4.66E-01	**9.53E-01**	**1**	**1**	**1**	**1**	**1**
DBSCAN	RandIndx	5.81E-01	6.55E-01	8.64E-01	**1**	**1**	5.87E-01	**1**	**1**
	AdjRandIndx	2.43E-01	3.03E-01	7.16E-01	**1**	**1**	3.36E-01	**1**	**1**
	F-measure	4.47E-01	**5.70E-01**	6.72E-01	**1**	**1**	2.58E-01	**1**	**1**
Fuzzy means	RandIndx	8.69E-01	**7.97E-01**	8.80E-01	5.00E-01	**1**	**1**	**1**	5.34E-01
	AdjRandIndx	7.09E-01	4.49E-01	7.29E-01	–9.15E-04	**1**	**1**	**1**	6.73E-02
	F-measure	8.85E-01	4.70E-01	8.93E-01	5.04E-01	**1**	**1**	**1**	6.30E-01
K-means	RandIndx	9.42E-01	7.65E-01	8.74E-01	5.00E-01	8.31E-01	8.84E-01	**1**	5.26E-01
	AdjRandIndx	8.69E-01	3.37E-01	7.16E-01	–5.28E-04	8.25E-01	7.22E-01	**1**	5.24E-02
	F-measure	**0.954 9**	4.50E-01	8.85E-01	5.10E-01	7.47E-01	7.16E-01	**1**	6.06E-01

We notice the AFD algorithm has more robust results than the others except on the Glass dataset with adjusted rand and F-measure, and on the Wine dataset with F-measure. The other advantage of the AFD algorithm is that it achieves all datasets with correct number except for Glass. The proposed algorithm and some recent density-based algorithms such as LOF and spectral clustering which are performed on the Iris dataset are shown in Figs. 6–8. It can be seen that spectral clustering and LOF are unable to cluster the Iris dataset. Each color indicates a cluster except pink.

Table 2 Parameters settings

Name	Wine	Glass	Iris	Cluster in cluster	Corner separate	Four separate cluster	Two separate cluster	Two spiral
AFD	Eps1 = 0.19 MinPoint = 4	Eps1 = 0.60 MinPoint = 3	Eps1 = 0.60 MinPoint = 3	Eps1 = 0.7 MinPoint = 10	Eps1 = 0.8 MinPoint = 31	Eps1 = 0.7 MinPoint = 10	Eps1 = 0.2 MinPoint = 5	Eps1 = 0.7 MinPoint = 10
DBSCAN	MinPoint = 4, Eps = 0.5	MinPoint = 5, Eps = 0.5,	MinPoint = 5, Eps = 0.895 0	MinPoint = 1, Eps = 0.001	MinPoint = 4, Eps = 0.895 0	MinPoint = 4, Eps = 0.9	MinPoint = 3, Eps = 0.5	MinPoint = 3, Eps = 0.5
Fuzzy means	M = 2, Cluster number = 3, M = 3, Cluster number = 3, M = 4, Cluster number = 3	M = 2, Cluster number = 6, M = 3, Cluster number = 6, M = 4, Cluster number = 6	M = 2, Cluster number = 3, M = 3, Cluster number = 3, M = 4, Cluster number = 3	M = 2, Cluster number = 2, M = 3, Cluster number = 2, M = 4, Cluster number = 2	M = 2, Cluster number = 4, M = 3, Cluster number = 4, M = 4, Cluster number = 4	M = 2, Cluster number = 4, M = 3, Cluster number = 4, M = 4, Cluster number = 4	M = 2, Cluster number = 2, M = 3, Cluster number = 2, M = 4, Cluster number = 2	M = 2, Cluster number = 2, M = 3, Cluster number = 2, M = 4, Cluster number = 2
K-means	Cluster number = 3	Cluster number = 6	Cluster number = 3	Cluster number = 2	Cluster number = 4	Cluster number = 4	Cluster number = 2	Cluster number = 2

5. Conclusions and future work

The proposed algorithm focusing on a variant shape dataset in terms of efficiency has been tested on the real dataset and synthetic dataset. It generates two parameters called Eps2 and Pos in the inner dataset, which are presented in order to make a decision between separating and merging data. The results of our experiments demonstrate that the AFD algorithm shows better results than the other algorithms, which is applied to the real dataset of UCI benchmark (Wine, Glass and Iris) datasets and synthetic datasets (the rest of the datasets). AFD is able to run in low dimensions, but it is more time consuming than others. The challenge here is Eps1 and MinPoints parameters, which should be initialized carefully. The different initializations lead to generating different clusters with Eps2 and Pos, which will be a challenge in this study. Generally, initialization parameters are performed by observing the high density. We need to estimate the amount of them in initialization. In the future, we plan to extend this algorithm considering the mentioned challenge by metaheuristic methods. Metaheuristic methods with their own power search on functional space lead to the best solution to initialization parameters.

References

[1] JAIN A K, MURTY M N, FLYNN P J. Data clustering: a review. ACM Computing Surveys, 1999, 31(3): 264–323.
[2] XU H X, TIAN Z. An optimal spectral clustering approach based on Cauchy-Schwarz divergence. Chinese Journal of Electronics, 2009, 18(1): 105–108.
[3] TANG W Q, LONG W K, SUN L J, et al. Multiple model adaptive control of nonlinear systems based on clustering method and neural network. Systems Engineering and Electronics, 2019, 41(9): 2100–2106. (in Chinese)
[4] ZHANG Y X, GUO W P, KANG K, et al. Key radar signal fast recognition method based on clustering and time-series correlation. Systems Engineering and Electronics, 2020, 42(3): 597–602. (in Chinese)
[5] ABDALLA A, AHMED M G S, ZHAO Y, et al. Deceptive jamming suppression in multistatic radar based on coherent clustering. Journal of Systems Engineering and Electronics, 2018, 29(2): 269–277.
[6] LI H, LIU X J, LI T, et al. A novel density-based clustering algorithm using nearest neighbor graph. Pattern Recognition, 2020, 102: 107206.
[7] GHALLAB H, FAHMY H, NASR M. Detection outliers on internet of things using big data technology. Egyptian Informatics Journal. DOI: 10.1016/j.eij.2019.12.001.
[8] GARG S, KAUR K, BATRA S, et al. A multi-stage anomaly detection scheme for augmenting the security in IoT-enabled applications. Future Generation Computer Systems, 2019, 104: 105–118.
[9] SAHA S K, SCHMITT I. Non-TI clustering in the context of social networks. Procedia Computer Science, 2020, 170: 1186–1191.
[10] XU D, TIAN Y. A comprehensive survey of clustering algorithms. Annals of Data Science, 2015, 2: 165–193.
[11] MACQUEEN J B. Some methods for classification and analysis of multivariate observations. Proc. of the 5th Berkeley Symposium on Mathematical Statistics and Probability, 1967: 281–297.
[12] PARK H S, JUN C H. A simple and fast algorithm for Kmedoids clustering. Expert Systems with Applications, 2009, 36(2): 3336–3341.
[13] KAUFMAN L, ROUSSEEUW P J. Partitioning around medoids (program PAM). Finding groups in data: an introduction to cluster analysis. New York: Wiley, 1990.
[14] KAUFMAN L, ROUSSEEUW P J. Finding groups in data: an introduction to cluster analysis. New York: Wiley, 1990.
[15] ZHANG T, RAMAKRISHNAN R, LIVNY M. BIRCH: an efficient data clustering method for very large databases. ACM SIGMOD Record, 1999, 25(2): 103–114.
[16] GUHA S, RASTOGI R, SHIM K. CURE: an efficient clustering algorithm for large databases. Information Systems, 2001, 26(1): 35–58.
[17] GUHA S, RASTOGI R, SHIM K. Rock: a robust clustering algorithm for categorical attributes. Information Systems,

2000, 25(5): 345–366.
[18] KARYPIS G, HAN E H, KUMAR V. Chameleon: hierarchical clustering using dynamic modeling. Computer , 1999, 32(8): 68–75.
[19] BOUDAILLIER E, HEBRAIL G. Interactive interpretation of hierarchical clustering. Intelligent Data Analysis, 1998, 2: 229–244.
[20] RODRIGUES P P, GAMA J, PEDROSO J. Hierarchical clustering of time-series data streams. IEEE Trans. on Knowledge and Data Engineering, 2008, 20(5): 615–627.
[21] ERTOZ L, STEINBACH M, KUMAR V. Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. Proc. of the 3rd SIAM International Conference on Data Mining, 2003: 47–58.
[22] ESTER M, KRIEGEL H P, SANDER J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. Proc. of the 2nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1996: 226 – 231.
[23] ANKERST M, BREUNIG M M, KRIEGEL H P, et al. OPTICS: ordering points to identify the clustering structure. ACM SIGMOD Record, 1999, 28(2): 49–60.
[24] BIRANT D, KUT A. ST-DBSCAN: an algorithm for clustering spatial-temporal data. Data & Knowledge Engineering, 2007, 60(1): 208–221.
[25] FAHIM A M, SAAKE G, SALEM A B M, et al. An enhanced density based spatial clustering of applications with noise. Proc. of the International Conference on Data Mining, 2009: 517–523.
[26] RASMUSSEN C E. The infinite Gaussian mixture model. Advances in Neural Information Processing Systems, 2000, 12: 554–560.
[27] GUNGOR E, OZMEN A. Distance and density based clustering algorithm using Gaussian kernel. Expert Systems with Applications, 2017, 69: 10–20.
[28] VISWANATH P, BABU V. Rough-DBSCAN: a fast hybrid density based clustering method for large data sets. Pattern Recognition Letters, 2009, 30(16): 1477–1488.
[29] BEZDEK J C, EHRLICH R, FULL W. FCM: The fuzzy c-means clustering algorithm. Computers & Geosciences, 1984, 10(2/3): 191–203.
[30] NASIBOV E N, ULUTAGAY G. Robustness of density-based clustering methods with various neighborhood relations. Fuzzy Sets and Systems, 2009, 160(24): 3601–3615.
[31] HIMMELSPACH L, CONRAD S. Density-based clustering using fuzzy proximity relations. Proc. of the Annual Meeting of the North American Fuzzy Information Processing Society, 2011. DOI: 10.1109/NAFIPS.2011.5751999.
[32] CHEN W Y, SONG Y Q, BAI H J, et al. Parallel spectral clustering in distributed systems. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2011, 33(3): 568–586.
[33] BREUNIG M M, KRIEGEL H P, NG R T, et al. LOF: identifying density-based local outliers. ACM SIGMOD Record, 2020, 29(2): 93–104.
[34] ARBELAITZ O, GURRUTXAGA I, MUGUERZA J, et al. An extensive comparative study of cluster validity indices. Pattern Recognition, 2013, 46(1): 243–256.
[35] ZHAO Q, XU M, FRAONTI P. Epanding external validity measures for determining the number of clusters. Proc. of the 11th International Conference on Intelligent Systems Design and Applications, 2011: 931–936.
[36] POWERS D M W. Evaluation: from precision, recall and f-measure to ROC, informedness, markedness & correlation. Journal of Machine Learning Technologies, 2011, 2(1): 37–63.

Biographies

YELGHI Aref was born in 1986. He received his B.S. degree and M.S. degree in computer science from the Azad University Sari branch in 2008 and 2012 respectively. He obtained his Ph.D. degree in computer engineering from Karadeniz Technical University in 2018. His research interests include optimization, neural network and data mining.

E-mail: aref.yelghi@avrasya.edu.tr

KÖSE Cemal was born in 1964. He is a professor in the Department of Computer Engineering at Karadeniz Technical University. He received his Ph.D. degree from Faculty of Engineering, and Department of Computer Science, University of Bristol, in 1997. His research interests include signal and image processing, pattern recognition, and human computer interaction.

E-mail: ckose@ktu.edu.tr

YELGHI Asef was born in 1986. He received his B.S. degree in accounting, M.S. degree in capital markets and stock exchange from University of Marmara of Turkey in 2008 and 2014 respectively. He obtained his Ph.D. degree in bussines administration in University of Gazi of Turkey in 2020. Also since 2014, he has been pursuing his Ph.D. degree in banking at University of Marmara of Turkey. His research interests include exchange rate, bond marketing and data mining.

E-mail: asefyelghi@gmail.com

SHAHKAR Amir was born in 1989. He received his B.S. degree in civil engineering in 2011 from Azad University of Tabriz. He received his M.S. degree in transportation engineering from Karadeniz Technical University, Trabzon, Turkey, in 2015. He started his Ph.D. learning in traffic engineering in 2016. Currently, he is a Ph.D. student in the Civil Engineering Department, Karadeniz Technical University, Trabzon. His research interests include analyzing traffic accidents through geographic information system (GIS), traffic simulation by Verkehr in Stadten simulations model (VISSIM), and optimizing traffic lights with adaptive neuro fuzzy inference system (ANFIS)-based metaheuristic algorithms.

E-mail: amirshahkartrabzon@gmail.com