Available online at www.sciencedirect.com
ScienceDirect
Fuzzy Sets and Systems 160 (2009) 3601 – 3615
FUZZY sets and systems
www.elsevier.com/locate/fss

Robustness of density-based clustering methods with various neighborhood relations

Efendi N. Nasibov[a,*], Gözde Ulutagay[b]

[a] *Department of Statistics, Faculty of Science and Arts, Dokuz Eylul University, Tinaztepe Campus, 35160 Buca, Izmir, Turkey*
[b] *Department of Computer Engineering, Izmir University, Gursel Aksel Blv. No.14, 35350 Uckuyular, Izmir, Turkey*

Received 5 August 2008; received in revised form 26 February 2009; accepted 17 June 2009
Available online 4 July 2009

Abstract

Cluster analysis is one of the most crucial techniques in statistical data analysis. Among the clustering methods, density-based methods have great importance due to their ability to recognize clusters with arbitrary shape. In this paper, robustness of the clustering methods is handled. These methods use distance-based neighborhood relations between points. In particular, DBSCAN (density-based spatial clustering of applications with noise) algorithm and FN-DBSCAN (fuzzy neighborhood DBSCAN) algorithm are analyzed. FN-DBSCAN algorithm uses fuzzy neighborhood relation whereas DBSCAN uses crisp neighborhood relation. The main characteristic of the FN-DBSCAN algorithm is that it combines the speed of the DBSCAN and robustness of the NRFJP (noise robust fuzzy joint points) algorithms. It is observed that the FN-DBSCAN algorithm is more robust than the DBSCAN algorithm to datasets with various shapes and densities.

Keywords: Clustering; Fuzzy neighborhood; FJP; DBSCAN; FN-DBSCAN

1. Introduction

Cluster analysis is a fundamental tool in statistical data analysis which is widely applied in a variety of scientific areas such as data mining, pattern recognition, geographic information systems, information retrieval, microbiology analysis and so forth [1,5,12,13,23,24]. The main objective of clustering is to facilitate the analysis process by constructing similar objects in a cluster. Clustering methods can be divided into different groups such as hierarchical, partitioning/prototype-based, density/neighborhood-based, etc. [12]. In hierarchical clustering, the remoteness of elements is the cornerstone. First of all, closer elements are put into the same cluster, and in the next step, elements a little bit far away from the previous ones are put into the same cluster and so forth. In prototype-based methods, however, prototypes which have common features of some certain classes are formed and then the elements are taken into these classes with respect to the similarity degrees to the prototypes. Namely, in such a situation, not the remoteness of the above-mentioned elements from each other, but their remoteness from the prototypes is considered. Some examples of these methods are k-means, k-medoids and fuzzy c-means (FCM) [8,11,14,21,22]. Likewise, single-linkage (SLINK), complete-linkage

* Corresponding author. Tel.: +90 536 509 7969; fax: +90 232 453 4265.
E-mail address: efendi_nasibov@yahoo.com (E.N. Nasibov).

0165-0114/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.fss.2009.06.012

(CLINK), density-based spatial clustering of applications with noise (DBSCAN), ordering points to identify clustering structure (OPTICS), fuzzy graph connectedness (FHC), fuzzy joint points (FJP) and noise resistant fuzzy joint points (NRFJP) are some examples of hierarchical methods [2,3,6,9,10,15–20,23–25]. Some of these methods can be handled as both hierarchical and density-based such as DBSCAN, FJP, NRFJP. In methods like DBSCAN, in order to determine the core points of clusters or noise points, classical neighborhood density analysis is performed. Thus, a point is called a core point if the number of points in a certain radius is larger than a specified threshold. However, FJP-like methods use fuzzy neighborhood cardinality in order to determine core points [16,20]. Papers [15,23] are examples of studies in density-based methods that consist fuzzy objects and relations. In study [15], distance between fuzzy objects is defined, and the FDBSCAN algorithm which is integrated directly into the clustering algorithm is proposed. It is also shown that the computation speed of this algorithm is more advantageous than those of EXPDBSCAN, UNION, INTERSECTION [14]. In study [23], fuzzy neighborhood relation-based on the intersection of the properties is proposed. This relation is used in some classification algorithms such as NN, KNN, fuzzy KNN and in some agglomerative hierarchical clustering algorithms. In study [6], dataset is partitioned into sub-clusters and fuzzy graph connectedness measure is used among them. Then, hierarchical clustering is applied based on the level sets of the graph.

FCM-like clustering algorithms are successful in specifying datasets with spherical or ellipsoidal shape and number of clusters and initial cluster centers must be pre-determined. Unless these information are true, clustering results could be far from the ideal [21,26–28]. On the other hand, in DBSCAN-like algorithms, it is not necessary to specify neither the number of clusters nor the initial cluster centers. Such kinds of algorithms are able to detect clusters in any shape. However, in such algorithms, adjusting the parameters that specify neighborhood radius and neighborhood density according to the density of clusters, cause some problems. In datasets with various densed clusters, if the parameters are set up for low densed clusters, high densed clusters could be merged. Conversely, if the parameters are set up for clusters with high density, then the clusters with low density could be perceived as noise. In this sense, algorithms which are able to run correctly in a wide range of change interval could be more advantageous. Thus, algorithm's robustness through parameters provide the datasets with different densities to be classified accurately. Local-density-based LDBSCAN algorithm is proposed to handle this problem [7]. Another approach is FJP-like NRFJP algorithm [17,18]. This algorithm is robust since it uses fuzzy relation in neighborhood analysis. Also it is easier to fine tune on the parameters. However, DBSCAN algorithm's computation speed is faster than that of NRFJP algorithm. In this study, the fuzzy neighborhood—DBSCAN (FN-DBSCAN) algorithm that combines DBSCAN algorithm's speed and NRFJP algorithm's robustness is handled. It is shown by experiments that the FN-DBSCAN algorithm is more robust than the DBSCAN algorithm in datasets with different shapes and densities.

The rest of the paper is organized as follows. In Section 2, the DBSCAN algorithm is mentioned and some concepts about this algorithm are defined. In Section 3, fuzzy neighborhood relation is analyzed and some basic concepts of the FN-DBSCAN algorithm are defined. Then FN-DBSCAN algorithm is explained. In Section 4, FN-DBSCAN and DBSCAN algorithms are compared by using 22 different kinds of datasets. The conclusion is stated in the final section.

2. DBSCAN algorithm and parameters used in neighborhood analysis

Let us consider a dataset $X = \{x_1, x_2, \ldots, x_n\}$. Each object x_i, $i = \overline{1,n}$ has m properties. Thus, each datum x_i could be handled as a point of m-dimensional space, i.e. $x_i = (x_{i1}, x_{i2}, \ldots, x_{im})$. In this sense, the Euclidean distance $d(x_i, x_j)$ between any points $x_i, x_j \in X$ can be determined as follows:

$$d(x_i, x_j) = \left(\sum_{k=1}^{m} (x_{ik} - x_{jk})^2 \right)^{1/2}.$$

First of all, let us define some concepts used in the DBSCAN algorithm.

The neighborhood set of point $x \in X$ detected by using any of the membership function is determined as follows.

Definition 1. The neighborhood set of point $x \in X$ with parameter ε (ε-neighborhood set) is as follows:

$$N(x; \varepsilon) = \{y \in X | d(x, y) \leq \varepsilon\}. \tag{1}$$

It is obvious that the neighborhood set determined by formula (1) will be composed of the points in ε-radius of the point x. This set can be written as

$$N_x(y) = \begin{cases} 1 & \text{if } d(x, y) \leq \varepsilon, \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

where $N_x(y)$ is the membership degree of the point y to the neighborhood set of the point x.

Definition 2. $x \in X$ is called a core point with parameters ε and *MinPts* if

$$|N(x; \varepsilon)| \geq MinPts$$

is satisfied where $|N(x; \varepsilon)|$ is the cardinality of the set $N(x; \varepsilon)$.

Definition 3. Let $p, q \in X$. A point p is directly density-reachable from a point q with respect to the ε and *MinPts* if q is a core point and $p \in N(q; \varepsilon)$.

Note that other points can only be directly density-reachable from core points.

Definition 4. Let $p_i \in X,\ i = 1, \ldots, n$. A point p is density reachable from a point q w.r.t. ε and *MinPts* if there is a chain of points $p_1, \ldots, p_n$, $p_1 = q$, $p_n = p$, such that p_{i+1} is directly density-reachable from p_i.

Definition 5. Let $p, q, o \in X$. A point p is density connected to a point q w.r.t. ε and *MinPts* if there is a core point o such that both p and q are density-reachable from o w.r.t. ε and *MinPts*.

Definition 6. Let D be a database of points. A cluster C w.r.t. ε and *MinPts* is a non-empty subset of D satisfying the following conditions:

(a) Maximality: $\forall p, q$: if $p \in C$ and q is density-reachable from p w.r.t. ε and *MinPts*, then $q \in C$.
(b) Connectivity: $\forall p, q \in C$: p is density-connected to q w.r.t. ε and *MinPts*.

Definition 7. Let $C_1, \ldots, C_k$ be the clusters of the database D w.r.t. parameters ε and *MinPts*. Then we define noise as the set of points in the database D not belonging to any cluster C_i, i.e. $noise = \{p \in D | \forall i : p \notin C_i\}$.

The main idea of DBSCAN algorithm is that each core point must have a certain minimum number of neighbors (*MinPts*) in a certain ε radius. The running principle of the algorithm is as follows: starting from each core point, every core point and points in its neighborhood which are directly density reachable from it (so-called seed points) form a set of seeds. Then, the process continues by starting from another core point and a new set of seeds is formed until each core point is handled in this sense. DBSCAN algorithm can be formulated as follows:

DBSCAN algorithm.

Step 1. Specify ε and *MinPts*.
Step 2. Mark all the points in the dataset as unclassified and set $t = 1$.
Step 3. Find an unclassified core-point p with parameters ε and *MinPts*. Mark the point p to be classified. Start a new empty cluster C_t and assign p to this cluster.
Step 4. Find all the unclassified points in the ε-neighborhood of p and call them a set of seeds.
Step 5. Get a point q in the set of seeds, mark q to be classified, assign q to the cluster C_t, and remove q from the set of seeds.
Step 6. Check if q is a core-point with parameters ε and *MinPts*, if so, add all the unclassified points in the ε-neighborhood of q to the set of seeds.
Step 7. Repeat steps 5 and 6 until the set of seeds is empty.
Step 8. Set $t = t + 1$ and repeat steps 3–7 until no more core points can be found.
Step 9. Output all the clusters found so far; and mark all the points which do not belong to any cluster as noise.
End.

3. Fuzzy neighborhood relation and FN-DBSCAN algorithm

As we mentioned in the previous section, the neighborhood radius (i.e. ε), and density threshold (i.e. *MinPts*) parameters are used in classical DBSCAN algorithm. However, since ε represents the direct value of the neighborhood radius, it takes values from different intervals corresponding to the scale of data. Such a case causes some problems in adjusting the ε parameter. In order to eliminate this problem, we can normalize data and get an ε value between [0, 1] by using the following transformation:

$$x'_{ik} = \frac{x_{ik} - x_k^{\min}}{(x_k^{\max} - x_k^{\min})\sqrt{m}}, \quad k = 1, \ldots, m,$$

where x'_{ik} are normalized new data and $x_k^{\min} = \min_{i=\overline{1,n}} x_{ik}$ and $x_k^{\max} = \max_{i=\overline{1,n}} x_{ik}$, $k = 1, \ldots, m$. The multiplier $1/\sqrt{m}$ implies that the maximum distance between the normalized points belongs to the interval [0, 1], i.e. the whole dataset belongs to the sphere with diameter 1. Hence,

$$d^{\max} = \max_{x'_i, x'_j \in X} d(x'_i, x'_j) = \max \sqrt{\sum_{k=1}^{m} (x'_{ik} - x'_{jk})^2}$$
$$= \max \sqrt{\sum_{k=1}^{m} \left(\frac{(x_{ik} - x_{jk})}{(x_k^{\max} - x_k^{\min})} \frac{1}{\sqrt{m}} \right)^2} \leq \sqrt{\sum_{k=1}^{m} \left(\frac{1}{\sqrt{m}} \right)^2} = 1$$

i.e.

$$d^{\max} = \max_{x'_i, x'_j \in X} d(x'_i, x'_j) \leq 1,$$

where $d(x'_i, x'_j)$ is the distance between (normalized) x'_i and x'_j. Hence, the condition $d_{\max} \leq 1$ and consequently $0 \leq \varepsilon \leq 1$ will be satisfied for normalized data. For simplicity, we will use the notation x_{ik} again for x'_{ik} from now on.

On the other hand, in order to invert the value of the parameter *MinPts* to the parameter ξ from the interval [0, 1] we can use the formula given below:

$$\xi = \frac{MinPts}{w^{\max}}, \tag{3}$$

where $w^{\max} = \max_{i=1,\ldots,n} w_i$, and w_i, is the cardinality of the point x_i within a certain ε radius. In general words regarding fuzzy situation, w_i is the sum of the membership degrees of points in the ε radius to the neighborhood set. Thus,

$$w_i = |N(x_i; \varepsilon)|,$$

where $|N(x_i; \varepsilon)|$ is the cardinality of the set $N(x_i; \varepsilon)$ determined with respect to Definition 1.

We can expand the neighborhood set determined in formula (2) to the fuzzy neighborhood case, i.e. N can be formed as any neighborhood membership function. In the following part of the paper, several cases of this neighborhood membership function will be discussed.

Since the neighborhood membership degrees of the points with different distances from core point also alter, utilizing fuzzy neighborhood function provides an advantage. Thus, there is an obvious difference between these points. But, in classical case there is no difference with respect to membership degrees between points within the same neighborhood radius of core point (Fig. 1). Because of that it could be more advantageous to use fuzzy neighborhood function instead of crisp neighborhood function used in DBSCAN algorithm.

Now let us investigate points x_1 and x_2 which have the same number of neighbors within $\varepsilon \leq d^{\max}$ radius (Fig. 2).

It is obvious that the points x_1 and x_2 in Fig. 2 are the same according to the crisp neighborhood relation used in DBSCAN. On the other hand, if fuzzy neighborhood function is used, point x_1 will have a higher membership degree

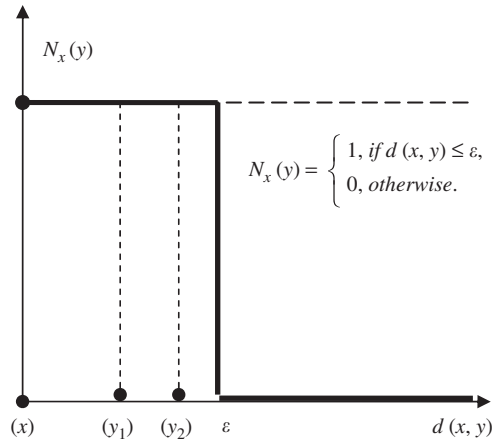

Fig. 1. Crisp neighborhood relation used in DBSCAN method.

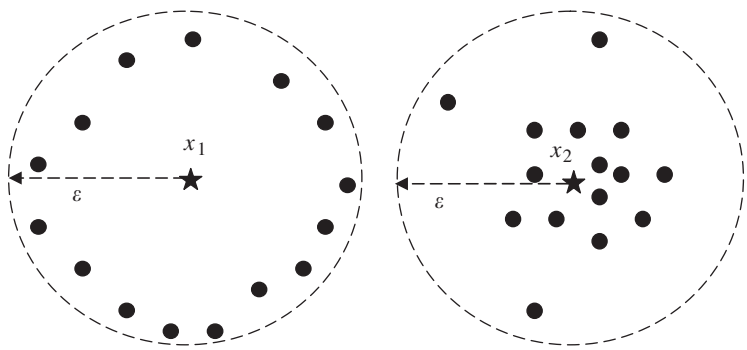

Fig. 2. Points x_1 and x_2 are similar according to crisp neighborhood cardinality, but dissimilar according to fuzzy neighborhood cardinality.

of being a core point than that of point x_2. Such a neighborhood membership function used in NRFJP algorithm is as follows:

$$N_x(y) = \left\{ \begin{array}{ll} 1 - \dfrac{d(x,y)}{d^{\max}} & \text{if } d(x,y) \le \varepsilon \\ 0 & \text{otherwise} \end{array} \right\}. \tag{4}$$

In neighborhood relation determined by the above formula, neighborhood degrees of points with varying distances to the core point will be different from each other (Fig. 3).

As it is seen from Fig. 3, points y_1 and y_2 have different neighborhood membership degrees to the point x. Hence, the membership degree of y_1, i.e. α_1, is higher than the membership degree of y_2, i.e. α_2. If we continue such a point of view, we can handle other neighborhood membership functions which can take neighborhood relation into consideration more sensitively. For example, in the following neighborhood membership function, sensitivity of the points with varying distances to neighbor points can be set up based on the parameter k.

$$N_x(y) = \max\left\{1 - k\frac{d(x,y)}{d^{\max}}, 0\right\}. \tag{5}$$

In such a case, the selection of the parameter $k > 0$ might determine the neighborhood radius. The parameter k could be adjusted with respect to the ε as follows (Fig. 4):

$$1 - k\frac{\varepsilon}{d^{\max}} = 0 \Rightarrow k = \frac{d^{\max}}{\varepsilon}.$$

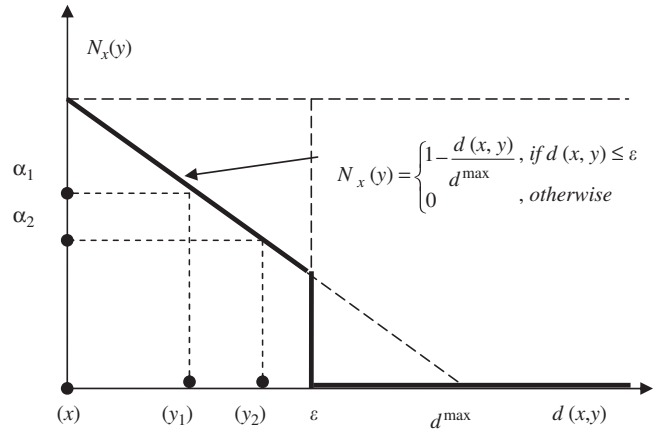

Fig. 3. Fuzzy neighborhood relation used in FJP method.

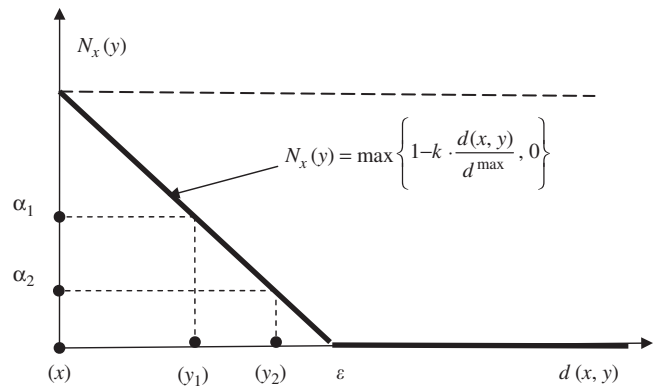

Fig. 4. Linear neighborhood relation within an ε radius.

Another important neighborhood membership function is given below:

$$N_x(y) = \exp\left(-\left(k\frac{d(x,y)}{d^{\max}}\right)^2\right). \tag{6}$$

In Eq. (6), the selection of parameter $k > 0$ affects the neighborhood radius. If the neighborhood set within ε radius is demanded to be suitable to a certain $\alpha \in (0, 1]$-level set, it should be $\exp(-(k\varepsilon/d_{\max})^2) = \alpha$ and from here $k = (d^{\max}/\varepsilon)\sqrt{-\ln\alpha}$ (Fig. 5).

To sum up what we explained from so far, one of the important concepts in fuzzy neighborhood analysis is the membership function that determines fuzzy neighborhood set. Clustering results may differ according to the shape of this function. Some suggestions for selection of the parameter k, used in neighborhood membership functions, according to the structure of datasets are given in Section 4.

In order to explain the FN-DBSCAN algorithm the definitions given above will be redefined for fuzzy logic approach. The main advantage of transformation of the DBSCAN algorithm to the FN-DBSCAN algorithm and using fuzzy sets theory is that various neighborhood membership functions that regularize different neighborhood sensitivities can be utilized. So the FN-DBSCAN method could be more robust to the scale and density variations of the datasets.

Definition 8. The fuzzy neighborhood set of point $x \in X$ with ε_1 parameter is as follows:

$$FN(x; \varepsilon_1) = \{\langle y, N_x(y)\rangle | y \in X, N_x(y) \ge \varepsilon_1\}. \tag{7}$$

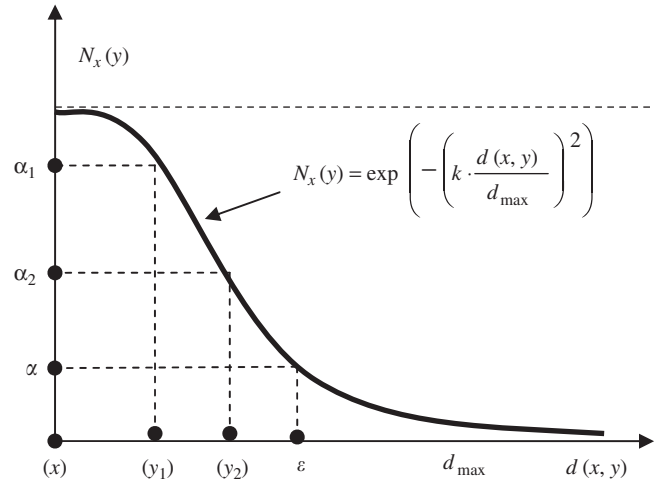

Fig. 5. Exponential neighborhood relation.

In definition 8, $N_x : X \to [0, 1]$ is any membership function that determines neighborhood relation between points. For instance, it can be any function defined in Eqs. (4)–(6) and given in Fig. 3–5. Note that, ε_1 parameter used in Eq. (7) determines the minimal threshold of the neighborhood membership degree where ε parameter used in Eq. (1) determines the maximal threshold of distance between points.

Definition 9. A point x is called a fuzzy core point with parameters ε_1 and ε_2 if

$$card\, FN(x; \varepsilon_1, \varepsilon_2) \equiv \sum_{y \in N(x;\varepsilon_1)} N_x(y) \geq \varepsilon_2$$

holds for any point $x \in X$, where

$$N(x; \varepsilon_1) = \{y \in X | N_x(y) \geq \varepsilon_1\}$$

determines the ε_1-level set of the fuzzy neighborhood set of the point x.

By taking into account the formula (3), we can relate the parameter ε_2 with parameter *MinPts* of DBSCAN algorithm as follows:

$$\varepsilon_2 = \xi = \frac{MinPts}{w^{\max}}.$$

Definitions 8 and 9 in FN-DBSCAN algorithm are used instead of Definitions 1 and 2 in DBSCAN algorithm, respectively. Definition 9 differs from Definition 2 in such a way that it uses a level-based neighborhood set instead of a distance-based neighborhood set and it uses the concept of fuzzy cardinality instead of classical cardinality in the determination of a core point.

In FN-DBSCAN algorithm, other points could be directly density-reachable only from a fuzzy core point as in the DBSCAN algorithm.

Definitions 3–7 in DBSCAN algorithm are also used directly in FN-DBSCAN algorithm. By the guidance of these definitions, FN-DBSCAN algorithm on the basis of fuzzy neighborhood relation is given below:

FN-DBSCAN algorithm.

Step 1. Specify parameters ε_1 and ε_2.
Step 2. Mark all the points in the dataset as unclassified. Set $t = 1$.
Step 3. Find an unclassified fuzzy core-point with parameters ε_1 and ε_2.
Step 4. Mark p to be classified. Start a new cluster C_t and assign p to the cluster C_t.

Step 5. Create an empty set of seeds S. Find all the unclassified points in the set $N(p;\varepsilon_1)$ and put all these points into the set S.
Step 6. Get a point q in the set S, mark q to be classified, assign q to the cluster C_t, and remove q from the set S.
Step 7. Check if q is a fuzzy core-point with parameters ε_1 and ε_2; if so, add all the unclassified points in the set $N(q;\varepsilon_1)$ to the set S.
Step 8. Repeat steps 6 and 7 until the set of seeds is empty.
Step 9. Find a new fuzzy core point p with parameters ε_1 and ε_2 and repeat steps 4–7.
Step 10. Mark all the points, which do not belong to any cluster, as noise.
End.

Note that, in FN-DBSCAN algorithm the same results of DBSCAN algorithm could be found by choosing the neighborhood membership function as in Eq. (2). So FN-DBSCAN algorithm always gives better results than does the DBSCAN algorithm by adjusting appropriate neighborhood membership functions.

4. Experimental results

In order to compare FN-DBSCAN algorithm which is based on fuzzy neighborhood analysis with DBSCAN algorithm which is based on classical neighborhood analysis, we use 22 datasets with various shapes and densities. The datasets were obtained from the papers [4,6,16–20] and some of them were simulated. Some of the datasets used in experiments are shown in Fig. 6.

The algorithms were programmed in C++ and the experiments were computed in Pentium(R)-D, 2.80 GHz, 2 GB RAM computers.

Correctness of the clustering results is validated by the expert visually. To evaluate the performances of the algorithms, we use the indicators given below, as "correct number percent (CNP)'' that indicates the percentage of the ratio of number of correct classified datasets to all number of datasets and "correct range percent (CRP)'' that indicates the percentage of correct result range of ε_1 parameter to the whole [0, 1] interval. To formulate the CNP and CRP criteria we use the following notations:

N	number of all datasets;
$n(\varepsilon_1,\varepsilon_2)$	number of correct clustered datasets with fixed parameters ε_1 and ε_2 of the algorithm;
$[\varepsilon_{1i}^L(\varepsilon_2), \varepsilon_{1i}^U(\varepsilon_2)]$	the "correct range (CR)'', i.e. the widest continuous interval of the ε_1 parameter for fixed ε_2, in which algorithm gives correct results for the dataset i, $i = 1,\ldots,N$, where $\varepsilon_{1i}^L(\varepsilon_2)$ is the lower bound, and $\varepsilon_{1i}^U(\varepsilon_2)$ is the upper bound of the interval;
$\varepsilon_1^{opt}(\varepsilon_2)$	the value of the ε_1 parameter in which the number of correct classified datasets is maximum for fixed ε_2.

So the CNP and CRP indicators are calculated as follows:

$$CNP(\varepsilon_1,\varepsilon_2) = \frac{n(\varepsilon_1,\varepsilon_2)}{N}100\%,$$

$$CRP_i(\varepsilon_2) = |\varepsilon_{1i}^U(\varepsilon_2) - \varepsilon_{1i}^L(\varepsilon_2)|100\%,$$

$$CRP(\varepsilon_2) = \frac{\sum_{i=1}^N CRP_i(\varepsilon_2)}{N}.$$

The experimental results for 22 datasets are given in Table 1.

As it is seen in Table 1, $k = 1$ for linear case has approximately same results with the crisp case. However, the results vary when different neighborhood membership functions are used. The results of the algorithm ameliorate when we change the value of the parameter k from 1 to 20, and then they deteriorate. The best results are found in linear case for $k = 15$.

We can say that results generally get better when exponential neighborhood function is used. The best results in exponential case are found for $k = 20$. If we compare this result with the best results of linear case and the best results

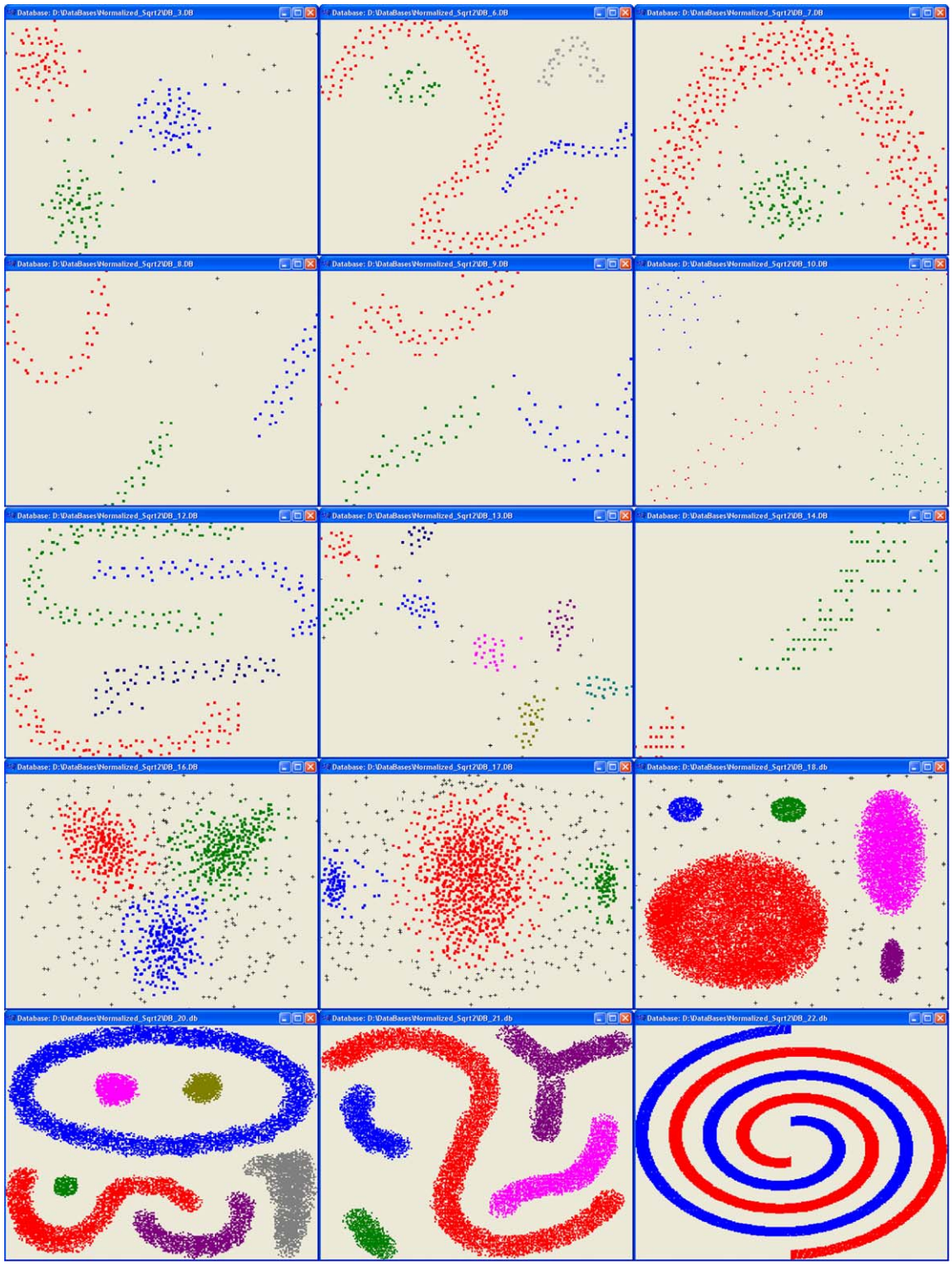

Fig. 6. Some of the datasets with various shapes and densities used in experiments.

Table 1
Indicators of ε_1 parameter for various k and ε_2 values in crisp, linear and exponential membership function cases.

	$\varepsilon_2 = 0.1$			$\varepsilon_2 = 0.2$			$\varepsilon_2 = 0.3$			$\varepsilon_2 = 0.4$		
	$\varepsilon_1^{opt}(\varepsilon_2)$	CNP $(\varepsilon_1^{opt}, \varepsilon_2)$	CRP (ε_2)	$\varepsilon_1^{opt}(\varepsilon_2)$	CNP $(\varepsilon_1^{opt}, \varepsilon_2)$	CRP (ε_2)	$\varepsilon_1^{opt}(\varepsilon_2)$	CNP $(\varepsilon_1^{opt}, \varepsilon_2)$	CRP (ε_2)	$\varepsilon_1^{opt}(\varepsilon_2)$	CNP $(\varepsilon_1^{opt}, \varepsilon_2)$	CRP (ε_2)
Linear												
***k*=1**	0.96	73	3.5	0.96	59	4.1	0.95	59	4.5	0.95	55	4.6
***k*=5**	0.71	68	16.5	0.78	50	19.6	0.82	45	22.0	0.77	45	22.1
***k*=10**	0.55	64	**24.0**	0.56	45	29.7	0.57	36	37.0	0.46	45	36.2
***k*=15**	0.31	68	22.4	0.33	50	**32.6**	0.31	36	**38.6**	0.18	36	**40.0**
***k*=20**	0.08	73	17.1	0.15	5	27.3	0.08	41	34.5	0.18	45	34.2
Exponential												
***k*=1**	0.99	**91**	0.4	0.99	**91**	0.4	0.99	**91**	0.4	0.99	**91**	0.4
***k*=5**	0.96	64	9.8	0.96	50	10.8	0.93	5	11.6	0.96	50	11.2
***k*=10**	0.87	64	21.0	0.82	50	25.3	0.76	50	27.9	0.81	45	30.1
***k*=15**	0.62	64	**26.6**	0.62	45	32.8	0.65	36	39.5	0.46	50	37.3
***k*=20**	0.53	64	25.6	0.45	45	**35.1**	0.44	32	**44.3**	0.26	41	**42.2**
Crisp	0.93	**73**	3.6	0.96	59	4.0	0.95	64	4.3	0.95	59	**4.5**

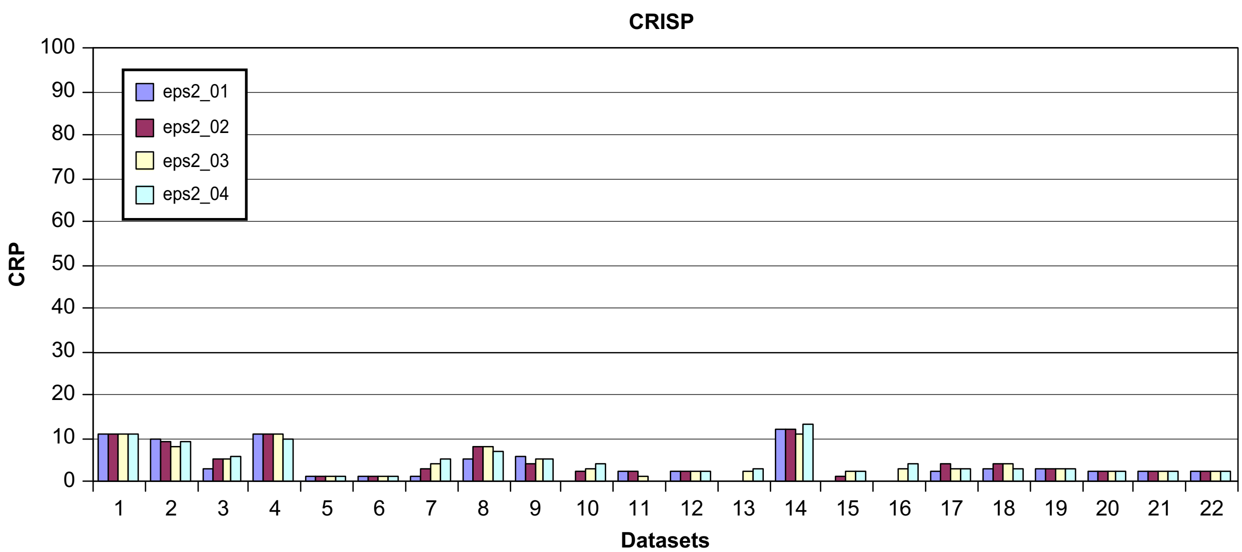

Fig. 7. $CRP(\varepsilon_2)$ for $\varepsilon_2 = 0.1, 0.2, \ldots, 0.4$ in crisp membership case.

of the crisp case, we can conclude that exponential membership function is very effective. Good results are marked in bold in Table 1. In order to show these results visually, comparisons are given as histograms for various values of the parameters (Figs. 7–10).

As it is seen from the histogram, crisp neighborhood membership function results in correct partitions in a narrow range of ε_1 parameter. However, in fuzzy neighborhood membership function case, the results of FN-DBSCAN algorithm are stable in a wider range of ε_1 parameter. It is obvious that the best results are found by using exponential membership function given in Eq. (6). Otherwise, the worst results are observed by using crisp membership function given in Eq. (2). Approximately the same results are obtained by using Eqs. (4) and (5) for neighborhood membership functions. In exponential case, CRP indicator for ε_1 parameter is about 25–45% while this index is only 3–5% in crisp case.

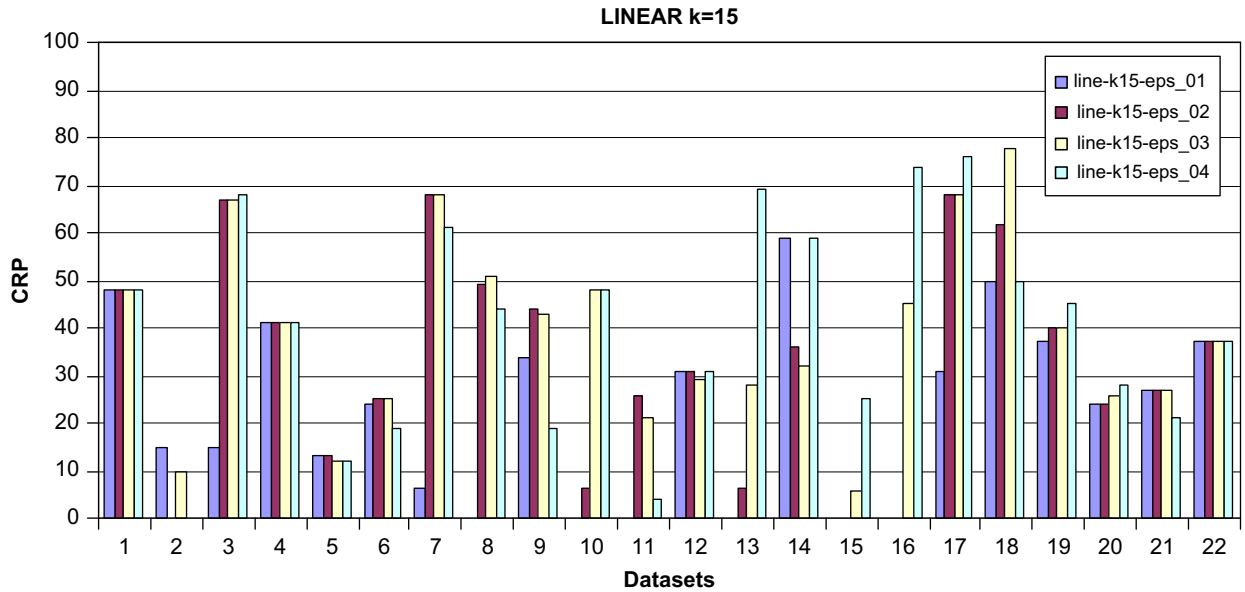

Fig. 8. $CRP(\varepsilon_2)$ for $\varepsilon_2 = 0.1, 0.2, \ldots, 0.4$ and $k = 15$ in linear membership case.

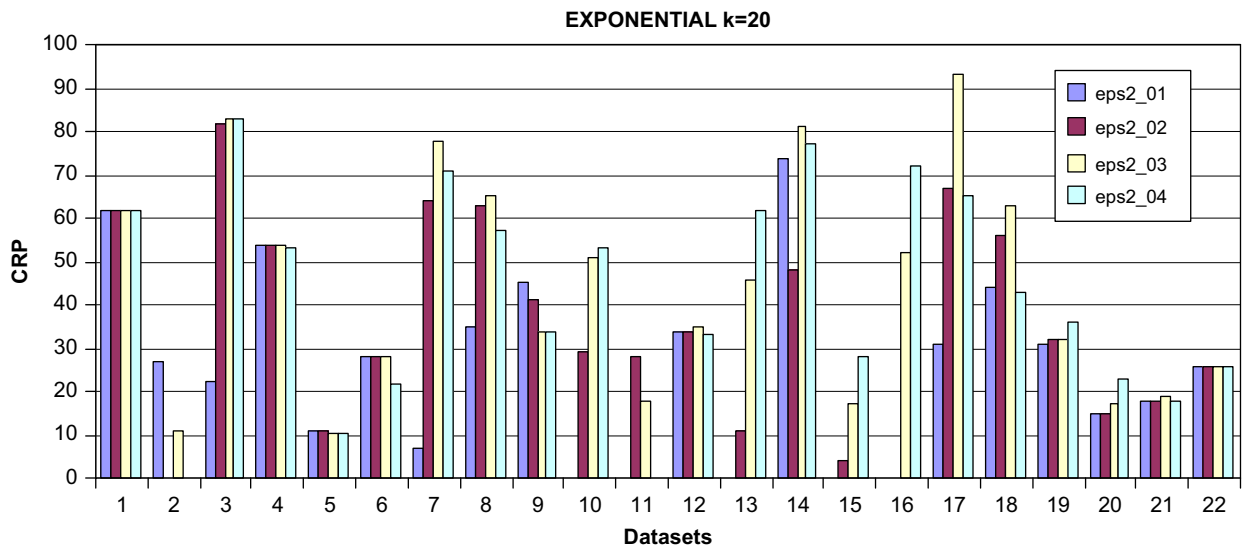

Fig. 9. $CRP(\varepsilon_2)$ for $\varepsilon_2 = 0.1, 0.2, \ldots, 0.4$ and $k = 20$ in exponential membership case.

In the second indicator the least squares method of regression is used. Hence, we look such an optimal value of the ε_1 parameter that its distance defined as follows from its CR will be minimum (Fig. 11):

$$d(\varepsilon_1, [\varepsilon_{1i}^L, \varepsilon_{1i}^U]) = \begin{cases} 1 & \text{if } \varepsilon_1 \notin [\varepsilon_{1i}^L, \varepsilon_{1i}^U], \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

The optimal value of the ε_1 parameter at fixed value of the parameter ε_2 is obtained by the following optimization problem:

$$f(\varepsilon_1; \varepsilon_2) = \sum_{i=1}^{N} d(\varepsilon_1, [\varepsilon_{1i}^L, \varepsilon_{1i}^U])^2 \to \min. \tag{9}$$

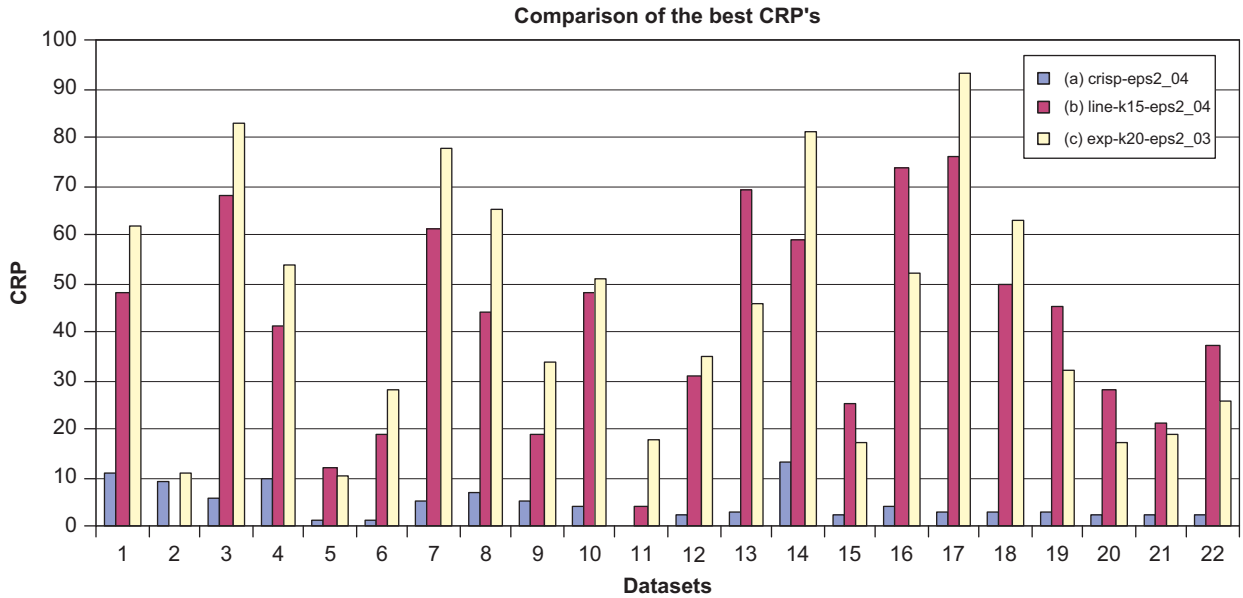

Fig. 10. Comparison of the best results of the $CRP(\varepsilon_2)$, (a) for $\varepsilon_2 = 0.4$ in crisp membership case, (b) for $k = 15$, $\varepsilon_2 = 0.4$ in linear membership case, (c) for $k = 20$, $\varepsilon_2 = 0.3$ in exponential membership case.

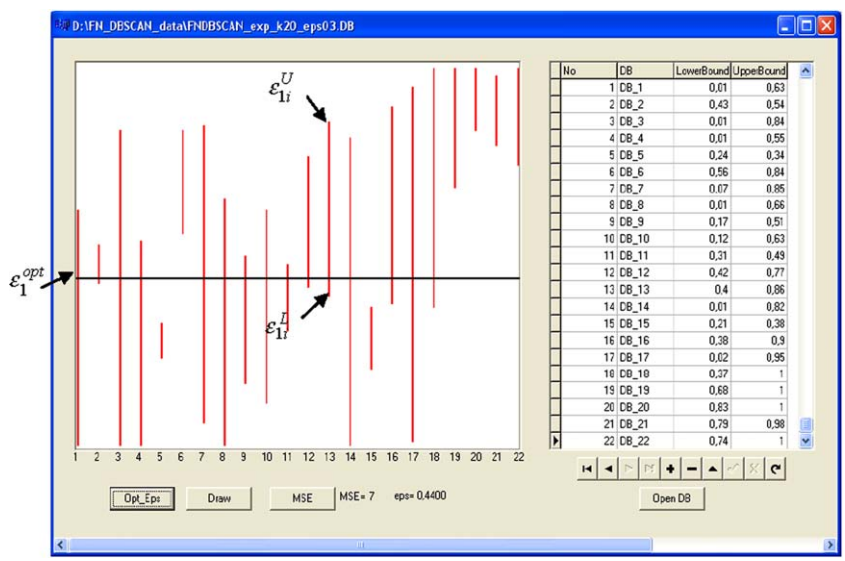

Fig. 11. $[\varepsilon_{1i}^L, \varepsilon_{1i}^U]$ intervals and the graphics of ε_1^{opt} value for $k = 20$, $\varepsilon_2 = 0.3$ in exponential membership case.

In fact,

$$f(\varepsilon_1; \varepsilon_2) = N - n(\varepsilon_1, \varepsilon_2). \tag{10}$$

The solution of the problem (9) has been mentioned previously as $\varepsilon_1^{opt}(\varepsilon_2)$. Corresponding optimal value $f(\varepsilon_1^{opt}; \varepsilon_2)$ indicates the number of datasets for which the algorithm gives minimum incorrect results for the fixed value of the ε_2 parameter (see Figs. 11–13).

It is clear from Table 1 that in fuzzy neighborhood function cases, most of the results are better than that of crisp case. Moreover, we get the best results for $k = 15$ and $\varepsilon_2 = 0.4$ in linear case, and $k = 20$ and $\varepsilon_2 = 0.3$ in exponential

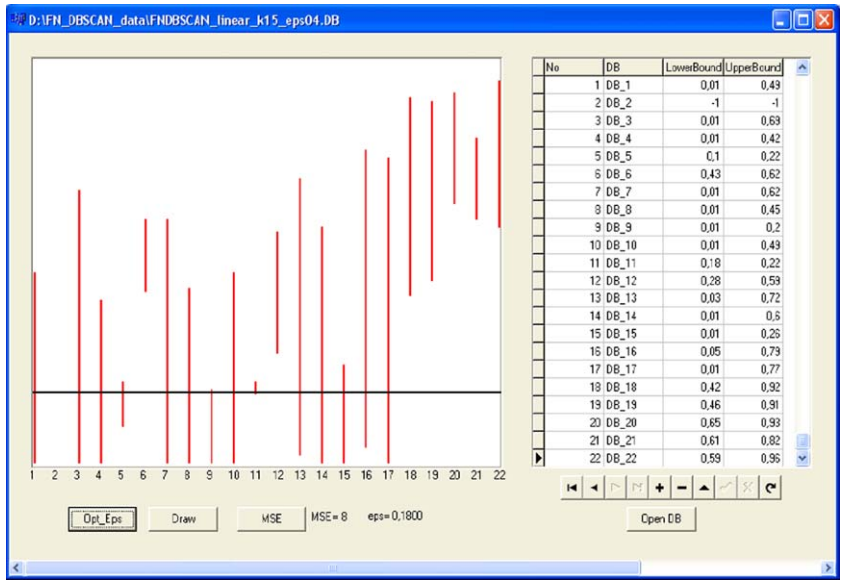

Fig. 12. $[\varepsilon_{1i}^{L}, \varepsilon_{1i}^{U}]$ intervals and the graphics of ε_1^{opt} value for $k = 15$, $\varepsilon_2 = 0.4$ in linear membership case.

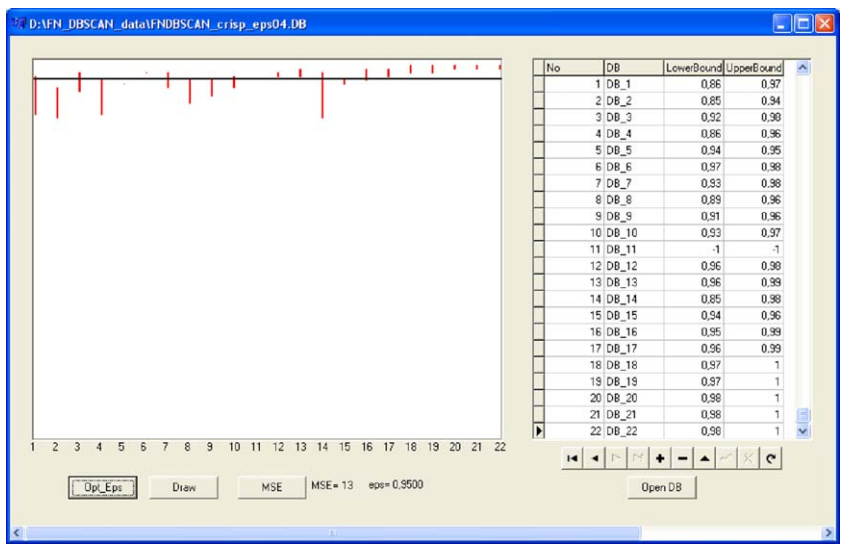

Fig. 13. $[\varepsilon_{1i}^{L}, \varepsilon_{1i}^{U}]$ intervals and the graphics of ε_1^{opt} value for $\varepsilon_2 = 0.4$ in crisp membership case.

case. The accuracy rate for the CNP indicator is also better than that of the crisp case, in other words the indicator is 91% for the fuzzy exponential ($k = 1$) case whereas it is less than 73% in all variants of the crisp case.

To sum up, we can conclude that for datasets with high density, greater values of the parameters k and ε_1 ($k = 15$–20, $\varepsilon_1 = 0.90$–0.99), and in datasets with low density, smaller values of these parameters should be preferred. We can also note that, for datasets with large number of noise points, greater values of the parameter ε_2 ($\varepsilon_2 = 0.3$–0.4) give better results.

The main drawback of the FN-DBSCAN similar to DBSCAN algorithm is in detecting overlapping clusters such as LDBSCAN dataset which has 473 points used in study [7]. As shown in study [7], DBSCAN algorithm is not successful in detecting clusters of LDBSCAN dataset. Hence, when the parameters of the DBSCAN algorithm are set to detect clusters C_1, C_2, C_3, it perceives cluster C, which forms a base for clusters C_1, C_2, C_3, as noise. On the other hand, if its parameters are set to detect cluster C as an individual cluster, then clusters C_1, C_2, C_3 are merged to cluster C. However,

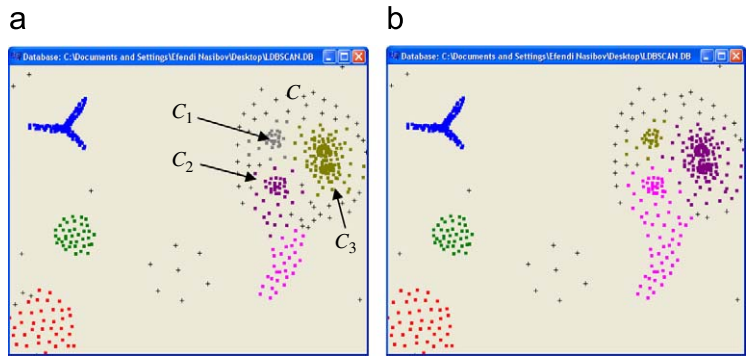

Fig. 14. Results of FN-DBSCAN algorithm for LDBSCAN dataset in exponential membership case for $k = 15$ and (a) $\varepsilon_1 = 0.82$, $\varepsilon_2 = 0.1$; (b) $\varepsilon_1 = 0.79$, $\varepsilon_2 = 0.1$.

as seen in Fig. 14, in various settings of ε_1 and ε_2 parameters, the clusters C_1, C_2, C_3 and a part of the cluster C are detected accurately by FN-DBSCAN algorithm. For instance, in exponential membership case for $k = 15$, $\varepsilon_1 = 0.82$ and $\varepsilon_2 = 0.1$, the clusters C_1, C_2, C_3 are detected accurately, but only a part of the cluster C is recognized (Fig. 14a). For $\varepsilon_1 = 0.79$ and $\varepsilon_2 = 0.1$, the cluster C_3 is merged to the cluster C (Fig. 14b).

Generally speaking, FN-DBSCAN algorithm has a drawback in overlapping clusters such as LDBSCAN dataset. But by the help of the parameter *pct*, LDBSCAN algorithm successfully partitions the clusters C_1, C_2, C_3 which are placed on cluster C, since it can give lower and upper bounds on cluster density. However, LDBSCAN algorithm will partition a cluster with more density around center and less density far away from the center, into separate clusters. This is why none of the clustering algorithms is suitable for all types of applications.

5. Conclusion

In this paper, a density-based clustering algorithm FN-DBSCAN based on fuzzy neighborhood function is handled and the effects of fuzzy neighborhood relation in density-based clustering is investigated. Besides being a more general algorithm, the FN-DBSCAN algorithm transforms into the well-known DBSCAN algorithm when the crisp neighborhood function is used. Experiments with various shapes and densities show that FN-DBSCAN algorithm gives more robust results than does the DBSCAN algorithm. On the other hand, FN-DBSCAN algorithm runs faster than the fuzzy neighborhood relation-based algorithms FJP and NRFJP. By the way, FN-DBSCAN algorithm combines the speed of DBSCAN and robustness of FJP-like algorithms.

In our study, parameter-based linear and exponential neighborhood functions are also proposed. After experiments with several values of the parameters, we determined the parameters that give better results. However, in this stage of our study, our aim is not to find the optimal values of the parameters, but to prove that one can get more realistic and robust results by using fuzzy neighborhood relation in FN-DBSCAN algorithm instead of using crisp neighborhood relation utilized in DBSCAN algorithm. Adjusting the optimal values of the parameters will be the subject of our future research.

Finally, we think that in areas such as data mining, pattern recognition, image processing, geographic information systems using fuzzy neighborhood relations instead of crisp ones could be more effective for datasets with different scales and densities.

Acknowledgments

This study is supported by the Scientific and Technological Research Council of Turkey (TUBITAK, Project no. 106T312). The authors are grateful to Dr. Lian Duan for providing dataset LDBSCAN.

References

[1] T. Abraham, J.F. Roddick, Survey of spatio-temporal databases, GeoInformatica 3 (1) (1999) 61–99.
[2] M. Ankerst, M.M. Breunig, H.-P. Kriegel, J. Sander, OPTICS: ordering points to identify the clustering structure, in: Proc. ACM SIGMOD Internat. Conf. on Management of Data, Philadelphia, PA, 1999, pp. 49–60.
[3] Z. Aoying, Z. Shuigeng, Approaches for scaling DBSCAN algorithm to large spatial database, Journal of Computer Science and Technology 15 (6) (2000) 509–526.
[4] A.M. Bensaid, L.O. Hall, J.C. Bezdek, L.P. Clarke, M.L. Silbiger, J.A. Arrington, R.F. Murtagh, Validity-guided (re)clustering with applications to image segmentation, IEEE Transactions on Fuzzy Systems 4 (2) (1996) 112–123.
[5] D. Birant, A. Kut, ST-DBSCAN: an algorithm for clustering spatial–temporal data, Data & Knowledge Engineering 60 (2007) 208–221.
[6] Y. Dong, Y. Zhuang, K. Chen, X. Tai, A hierarchical clustering algorithm based on fuzzy graph connectedness, Fuzzy Sets and Systems 157 (2006) 1760–1774.
[7] L. Duan, L. Xu, F. Guo, J. Lee, B. Yan, A local-density based spatial clustering algorithm with noise, Information Systems 32 (2007) 978–986.
[8] J.C. Dunn, A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, Journal of Cybernetics 3 (3) (1973) 32–57.
[9] M. Ester, H.P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proc. Second Internat. Conf. on Knowledge Discovery and Data Mining, 1996, pp. 226–231.
[10] S. Guha, R. Rastogi, K. Shim, CURE: an efficient clustering algorithms for large databases, in: Proc. ACM SIGMOD Internat. Conf. on Management of Data, Seattle, WA, 1998, pp. 73–84.
[11] R.E. Hammah, J.H. Curran, On distance measures for the fuzzy K-means algorithm for joint data, Rock Mechanics and Rock Engineering 32 (1) (1999) 1–27.
[12] J. Han, M. Kamber, Data Mining Concepts and Techniques, Morgan Kaufmann Publishers, San Francisco, CA, 2001, pp. 335–391.
[13] J. Han, M. Kamber, A.K.H. Tung, Spatial clustering methods in data mining: a survey, in: H. Miller, J. Han (Eds.), Geographic Data Mining and Knowledge Discovery, Taylor & Francis, London, 2001.
[14] H.-P. Kriegel, K. Kailing, A. Pryakin, M., Schubert, Clustering multi-represented objects with noise, PAKDD, 2004, pp. 394–403.
[15] H.-P. Kriegel, M. Pfeifle, Density-based clustering of uncertain data, in: Proc. 11th ACM SIGKDD Internat. Conf. on Knowledge Discovery in Data Mining, 2005, pp. 672–677.
[16] E.N. Nasibov, An alternative fuzzy-hierarchical approach to cluster analysis, in: Proc. Seventh Internat. Conf. on Application of Fuzzy Systems and Soft Computing, Siegen, Germany, 2006, pp. 113–123.
[17] E.N. Nasibov, A robust algorithm for fuzzy clustering problem on the base of fuzzy joint points method, Cybernetics and Systems Analysis 44 (1) (2008).
[18] E.N. Nasibov, G. Ulutagay, A new unsupervised approach for fuzzy clustering, Fuzzy Sets and Systems 158 (2007) 2118–2133.
[19] E.N. Nasibov, G. Ulutagay, A new approach to clustering problem using the fuzzy joint points method, Automatic Control and Computer Sciences 39 (6) (2005) 8–17.
[20] E.N. Nasibov, G. Ulutagay, On the fuzzy joint points method for fuzzy clustering problem, Automatic Control and Computer Sciences 40 (5) (2006) 33–44.
[21] N.R. Pal, J.C. Bezdek, On cluster validity for the fuzzy c-means model, IEEE Transactions on Fuzzy Systems 3 (3) (1995) 370–379.
[22] W. Pedrycz, F. Gomide, An Introduction to Fuzzy Sets, Massachusetts Institute, 1998.
[23] M. Sadaaki, Y. Endo, S. Hayakawa, E. Kataoka, Classification and clustering of information objects based on fuzzy neighborhood system, in: IEEE Internat. Conf. on Systems, Man and Cybernetics, Hawaii, 2005, pp. 3210–3215.
[24] H. Samet, The Design and Analysis of Spatial Data Structures, Addison-Wesley, Reading, MA, 1990.
[25] J. Sander, M. Ester, H.P. Kriegel, X. Xu, Density-based clustering in spatial databases: the algorithm GDBSCAN and its applications, Data Mining and Knowledge Discovery 2 (1998) 169–194.
[26] R.P. Velthuizen, L.O. Hall, L.P. Clarke, M.L. Silbiger, An investigation of mountain method clustering for large data sets, Pattern Recognition 30 (7) (1997) 1121–1135.
[27] R.R. Yager, D.P. Filev, Approximate clustering via the mountain method, IEEE Transactions on Systems, Man and Cybernetics 24 (8) (1994) 1279–1284.
[28] N. Zahid, O. Abouelala, M. Limouri, A. Essaid, Fuzzy clustering based on K-nearest-neighbours rule, Fuzzy Sets and Systems 120 (2001) 239–247.