E-DBSCAN: An evidential version of the DBSCAN method

Malek Bessrour, Zied Elouedi, Eric Lefevre

E-DBSCAN: An evidential version of the DBSCAN method

Malek Bessrour
LARODEC, Université de Tunis,
Institut Supérieur de Gestion,
2000 Le Bardo, Tunisia
bessrour.malek@gmail.com

Zied Elouedi
LARODEC, Université de Tunis,
Institut Supérieur de Gestion,
2000 Le Bardo, Tunisia
zied.elouedi@gmx.fr

Eric Lefèvre
Univ. Artois, ER 3926,
LGI2A
62400 Béthune, France
eric.lefevre@univ-artois.fr

***Abstract*—In later years, data have grown enormously and dealing with them to extract information has become a necessity. Data mining is a subfield of both computer science and statistics that aim to extract useful information in a comprehensive structure. The importance of clustering techniques in data mining has lead to the development of many methods in order to deal with data. Among these methods, we name density-based techniques, such as DBSCAN, that partitions data into heterogeneous shapes according to their local densities. DBSCAN can be suitable when handling big data that have noises and outliers. However, the classic DBSCAN method fails in identifying clusters with a variable density distribution and overlapping borders which is accurate in real-world data. In this paper, we propose an unsupervised learning technique in an uncertain context, that combines the DBSCAN method and the framework of the belief function theory, in order to generate clusters having overlapping borders. The proposed evidential clustering method, that we called E-DBSCAN, has the ability to handle cluster membership degree uncertainty of objects by using the belief function theory.**

***Index Terms*—Unsupervised learning, Density-based clustering, DBSCAN, Uncertainty, Belief function theory, Data mining.**

I. Introduction

Clustering techniques have proven their importance in machine learning and data mining, also named data analysis. Data mining is a process that extracts previously unknown patterns from a large quantity of data such as dependencies and groups of data that share close similarities. The idea of clustering is to regroup data into groups or clusters of objects. Objects in one group are similar to each other but dissimilar to the other objects in other groups. The main objective of clustering techniques is to maximize similarities in each cluster (the intra−cluster) and to minimize similarities between clusters (the inter−cluster). Different measures of distances are used to compare between objects such as the Manhattan distance, the Euclidean distance, and the Minkowski distance. These clustering techniques are widely used in many applications as for example marketing [1], finance [2], medicine [3], and image processing [4]. However, among these applications, some of them require the use of the density-based clustering methods in order to partition data into different shapes and homogeneous local density regions, and to identify noises or outliers.

One of the most widely used density-based techniques is the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [5], and that goes back to the fact that this algorithm does not require the number of clusters as input.

However, in many data mining clustering applications, sometimes there exist some better partitions of the data that can be as good as the best one found. Let's have data where some points are located in the middle of clusters such as the distance between each point and the other cluster's centers is equal. In this case, many possible solutions exist regarding the membership of those points to the nearest cluster. Therefore, dealing with uncertainty in clustering allows one to properly choose the most interpretable partition from an equivalent set of partitions. Nevertheless, crisp density-based clustering algorithms, such as DBSCAN, fail in detecting and dealing with uncertainty. Therefore, in order to cope with unwanted crisp boundaries, soft density-based clustering techniques have been defined. The fuzzy extensions of DBSCAN generate fuzzy overlapping boundaries clusters by affecting objects to two clusters or more with different membership degrees [6]–[8]. A survey regarding the main crisp and fuzzy density-based methods was reported by [9]. Among these fuzzy extensions, we cite Soft DBSCAN [10], Fuzzy core DBSCAN [11], Scalable fuzzy neighborhood DBSCAN [12], and the Fuzzy extensions of the DBSCAN proposed by [13].

The evidential clustering, also called credal clustering, is a soft clustering paradigm that extends fuzzy, possibilistic, and rough set clustering. Moreover, these later are sometimes seen as special cases of the evidential clustering [14]. It describes the uncertainty regarding the assignment of objects to clusters using the framework of belief functions [15]. Furthermore, the evidential clustering affects objects to each possible subset of classes with a membership degree defined by a mass function that is between 0 and 1 for each set of clusters. One of the evidential clustering approaches that was proposed is Evclus in [16]. There are also other methods that have been developed within the theory of belief functions. For instance, we can recall the Evidential C-Means (ECM) method that has been developed to deal with vectorial data [17]. In ECM, each object is represented by a mass of belief that is dependent on

the distance between that object and the space of prototypes. Unlike the classic K-means and the fuzzy K-means algorithms, in ECM prototypes are not only associated to clusters, but also to groups or sets of clusters. The objective of the ECM is to minimize a cost function so that each object has a high mass that is assigned to the cluster that corresponds to the closest prototype to that object. To deal with dissimilarity data, a relational version of the ECM method, called Relational Evidential C-Means (RECM), has been proposed in [18]. Specifically, in [18], a new notion regarding the Euclidean dissimilarity matrix has been proposed in order to deal with proximity data. Another evidential clustering method called Ek−NNclus was later proposed in [19]. In this method, objects are reassigned iteratively to clusters by using the Evidential k−Nearest Neighbors (Ek−NN) rule [20], and this until a final stable partition is obtained. Ek−NN rule is a classification procedure, based on the nearest neighbors method, that affects a label to an object by taking into account its distances to its k neighbors.

All these methods mentioned above are partition-based methods, however, in literature, there has been no density-based method proposed within the theory of belief functions. To this extent, we propose a soft version of the density-based clustering method DBSCAN within the belief theory framework in order to handle the uncertainty of the cluster membership degrees of data.

The rest of this paper is organized as follows. Sections 2 and 3 give an overview of the classic DBSCAN and the belief function theory respectively. Section 4 is dedicated to the description of our method. Section 5 compares our method to existing solutions through the experiments conducted on various datasets. Finally, Section 6 concludes and points to some research opportunities.

II. Classic DBSCAN method

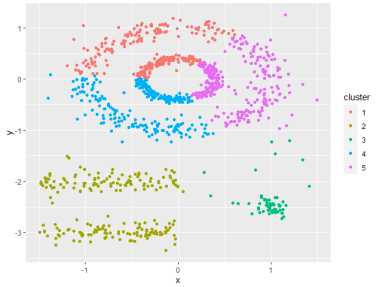
Density-based spatial clustering of applications with noise (DBSCAN), proposed in [5], is considered as one of the well-known density-based clustering methods. Unlike hierarchical clustering and partition-based techniques, DBSCAN can be very efficient when dealing with arbitrary shaped clusters such as seen in Figure 1.

DBSCAN is based on the concept of reachability that is within a radius, how many neighbors does each point have [5]. As a result, it is easier to model clusters with arbitrary shapes. Specifically, DBSCAN assigns each point of the feature space to the clusters that have many points that are close to that point, otherwise it labels them as noises or outliers if their local densities are low, that is to say, their neighbors are below the input radius.

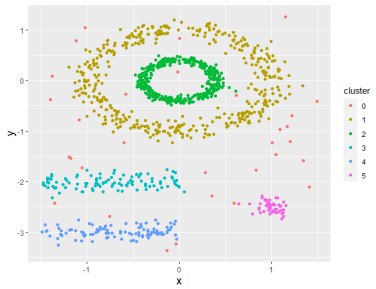
A. DBSCAN parameters

Although, DBSCAN does not require a priori parameter k as number of clusters, it requires two other parameters [5]:

- ϵ**:** radius, specifies how close two points should be one to another in order to be considered neighbors and belong to the same cluster.



(a) K-means



(b) DBSCAN

Fig. 1: Difference between results of K-means (a) and DBSCAN (b)

- **MinPts:** minimum number of neighbors, specifies the minimum number of points within the ϵ radius in order to form a dense region.

B. Parameter estimation

Parameter estimation had become a big problem in every data mining task. To choose good parameters, one needs to understand their use and to have a basic knowledge regarding the data set that is going to be used. Several heuristic studies have been developed to determine DBSCAN parameters. In the following, we recall different methods in literature for MinPts and ϵ estimation.

- **MinPts:** One heuristic approach is to use the natural logarithm function $ln(N)$ [5], where N is the size of the data set of points to be clustered. Another heuristic approach is to derive the parameter from the number of dimensions (D) in the data set such as MinPts is higher or equal to $(D+1)$.
- ϵ**:** Generally, it is preferable to choose small ϵ values. One heuristic approach [5] is to choose the ϵ parameter based on the distances of the dataset. Thus, the k-distance graph is used to find it. Correspondingly, we calculate the mean of the distances of each point to its k nearest neighbors, where k corresponds to MinPts. Then, plot these k-distances in an ascending order and observe a threshold point called a “knee” or a “valley”. The value of that knee point corresponds to the optimal ϵ parameter.

C. Types of points

Based on these parameters, points can be either classified as core point, as border point, or as outlier [5]:

- **Core point:** is a point that has at least MinPts neighbors, including itself, within its neighborhood with an ϵ radius.
- **Border point:** is a point that is reachable from another point that is a core point, and within its ϵ-neighborhood, there exists less than MinPts neighbors.
- **Noise:** is a point that is neither a core point nor a border point.

Figure 2 is an illustration of the different types of points where the MinPts is equal to 4. C is a core point surrounded by 4 neighbors represented in green. Blue points like B are border points located within the ϵ radius of the green points. Point N does not exist in any ϵ radius, thus it represents a noise.

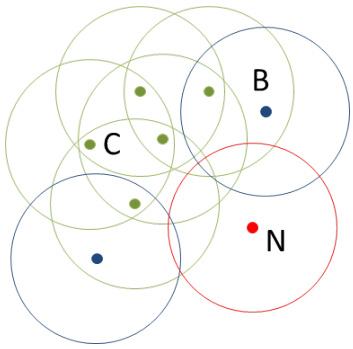


Fig. 2: Type of points in DBSCAN method

By exploring the concepts of density connectivity and density reachability, these parameters and these types of points can be well understood.

D. Reachability

In terms of density, reachability considers a point as being directly reachable from another one if it exists within an ϵ distance from it [5].

- **Directly density reachable:** An object A is directly density reachable from another object B if A is exists in the ϵ-Neighborhood of B and B is a core object.
- **Density reachable:** An object A is density-reachable from another object B if there exists a chain of objects $o_1, o_2 \ldots, o_n$, with $o_1 = A$ and $o_n = B$ such that o_{i+1} is directly density-reachable from o_i for all $1 <= i <= n$.

E. Connectivity

Connectivity is based on a chaining-approach to decide whether a point is located in a particular cluster [5]. For instance, object A is density-connected to another object B if there exists an object C such that both A and B are density-reachable from C.

F. Algorithm

The DBSCAN algorithm works iteratively where for each point o selected randomly and not yet visited, if the number of its neighbors is below MinPts, it marks it as a noise, otherwise it affects it to a new cluster and expends the cluster with its neighbors. For each neighbor, DBSCAN checks its ϵ-neighborhood and if there exists at least MinPts objects, it expands the cluster with these objects. DBSCAN approach is presented in Algorithms 1 and 2.

Algorithm 1: *DBSCAN(D,ϵ,MinPts)*

Input: ϵ: radius around a point to form a neighborhood area
Input: MinPts: minimum number of neighbors to be be considered as a core point
Data: D: Dataset

```
C=0
Clusters=∅
forall o ∈ D s.t. o is unvisited do
    mark o as visited
    neighborsPts = regionQuery(o,ε)
    if sizeof(neighborsPts) <= MinPts then
        Mark o as Noise
    else
        C = next cluster
        Clusters = Clusters ∪ expandCluster(o,
        neighborsPts, C, ε, MinPts)
return Clusters
```

Algorithm 2: *expandCluster(o, neighborsPts, C, ϵ, MinPts)*

Input: o: the point that was just marked as visited
Input: neighborsPts: the neighborhood of point o
Input: C: the current cluster
Input: ϵ: radius around a point to form a neighborhood area
Input: MinPts: minimum number of neighbors to be be considered as a core point

```
add o to cluster C
forall o′ ∈ neighborsPts do
    if o′ is not visited then
        Mark o′ as visited
        neighborsPts′ = regionQuery(o′,ε)
        if sizeof(neighborsPts′) > MinPts then
            neighborsPts = neighborsPts ∪
            neighborsPts′
    if o′ is not yet member of any cluster then
        add o′ to cluster C
return C
```

III. Belief function theory

Belief function theory (also referred to as evidence theory or Dempster-Shafer theory) is a theoretical framework, like the possibility or the probability theories, for reasoning under uncertainty. It was first introduced by Dempster (1967) in the statistical inference's context [21]. After that, it was formalized as a theory of evidence by Shafer (1976) [15]. Then, it was developed as the Transferable Belief Model (TBM) by Smets in the 1980's and 1990's [22].
In this section, we will recall the main concepts of this theory.

A. Frame of discernment

Within the framework of the evidence theory, the frame of discernment regroups all its subsets and is denoted by 2^{Ω} where each element is called an event or a proposition. The frame of discernment is defined as:

$$2^{\Omega} = \{A, A \subseteq \Omega\} = \{\emptyset, \{\omega_1\}, \{\omega_2\}, ..., \{\omega_1, \omega_2\}, ..., \Omega\} \tag{1}$$

Where the empty set $\emptyset$ represents the impossible proposition, and the set Ω represents the certain proposition.

B. Basic belief assignment

The basic belief assignment (bba), also called mass function represents the effect of an uncertain evidence on the frame of discernment's all subsets. This function is defined as $2^{\Omega} \rightarrow [0, 1]$ such that:

$$m(\emptyset) = 0 \quad and \quad \sum_{A \subseteq \Omega} m(A) = 1 \tag{2}$$

where the value $m(A)$, that is named a basic belief mass (bbm), interprets the fraction of evidence that supports exactly the assertion that the actual event ω belongs to A ($\omega \in A$) and nothing more specific.

C. Combination rule

In belief function theory, if different information sources of evidence are available we aggregate them using the Dempster rule of combination [15]:

$$m_1 \oplus m_2(A) = \begin{cases} \frac{\sum_{B \cap C = A} m_1(B) m_2(C)}{1 - \sum_{B \cap C = \emptyset} m_1(B) m_2(C)} & for\ A \neq \emptyset\ and\ A \subseteq \Omega. \\ 0 & for A = \emptyset. \end{cases} \tag{3}$$

D. Decision making

In some cases, it is necessary to make a decision based the available evidence that is modelled in bba forms. In this context, pignistic transformation [22] has been introduced to transform belief functions into probability measures, denoted $BetP$ and defined as:

$$BetP(A) = \sum_{B \subseteq \Omega} \frac{|A \cap B|}{|B|} m(B),\ \forall A \subseteq \Omega. \tag{4}$$

E. Evidential $k-$Nearest Neighbors rule

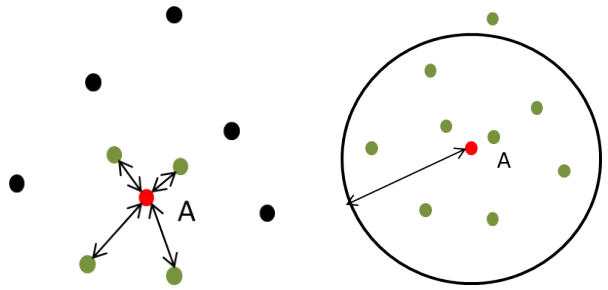
Let $\Omega = \{\omega_1, ..., \omega_c\}$ be the set of classes or groups, and let d_{ij} be the distance that separates the object o_i to be labeled and the object o_j that belongs to the class or group $\omega_{k(j)}$ with $k(j) \in \{1, \ldots, c\}$. The knowledge about the label of object o_j and the distance d_{ij} from o_i to o_j is taken as a piece of evidence and can be represented, thus, by the mass function in the following:

$$m_{ij}(\{\omega_{k(j)}\}) = \alpha_{ij} \tag{5}$$

$$m_{ij}(\Omega) = 1 - \alpha_{ij} \tag{6}$$

with

$$\alpha_{ij} = \varphi(d_{ij}) \tag{7}$$



(a) Combination of k neighbors masses

(b) Combination of all neighbors masses within radius ϵ

Fig. 3: Comparison between the use of the two parameters k and ϵ in the calculation of the mass function of an object A

where φ is set to be a non-increasing mapping function such that:

$$\lim_{d \rightarrow +\infty} \varphi(d) = 0 \tag{8}$$

In [20], the author proposed to choose φ as:

$$\varphi(d_{ij}) = \alpha_0\ exp(-\gamma_{k(j)} d_{ij}^{\beta}) \tag{9}$$

where α_0 and β are constants such that $0 < \alpha_0 < 1$ and $\beta \in \{1, 2, ...\}$. The distances from object o_i to the k objects are, then, considered as k pieces of evidence, thus, mass functions m_i are combined by using Dempster's rule in Equation 3 in order to obtain the mass function of the object o_i.

In [23], it was proposed to replace the k parameter in the E$k-$NN rule, that refers to the number of neighbors to be considered in the combination rule, with another parameter that refers to a radius that will represent the area to be considered in the combination rule. Let's have a point o_i that we would like to calculate its mass function and a given radius parameter ϵ, all the points that exist within the ϵ area of point o_i are taken as pieces of evidence in order to calculate its mass function. Figure 3 represents the difference between the two constraints where in Figure 3 (a), the mass function of point A is calculated by combining the masses of its k neighbors, while in Figure 3 (b), the mass function of point A is calculated by combining the masses of all the points that exist within the ϵ radius.

IV. Evidential DBSCAN

In this section, we propose our evidential version of DBSCAN which we call E-DBSCAN. We define it by specifying an approximate value of the radius instead of a crisp numeric parameter ϵ. The proposed method consists basically of three steps. Firstly, we determine core points and their neighbors that are within the ϵ_{min}, and within the belief function framework, this is the state of total certainty. Thus, we assign these points to their clusters with the equation:

$$\exists \omega_i \in \Omega, m(\{\omega_i\}) = 1 \tag{10}$$

Secondly, for each point that exists within the ϵ_{max} radius of core points, we assign its membership degree based on its

neighbors by following the principle of the Ek-NN rule [20]. These points are called soft border points. Finally, unlabelled points or outliers that does not exist within ϵ_{max} radius of any core point will be assigned to total ignorance class.

The proposed method is described in Algorithms 3 and 4.

A. Allocation of soft border points

Soft border points that are between the ϵ_{min} radius and the ϵ_{max} radius of core points can be assigned to the existing clusters based on their neighbors and the labelled points in the neighbourhood can be seen as a source of evidence. We replace the k parameter of the Ek-NN rule with the ϵ_{max} radius following the approach proposed by [20]. Suppose point o_i is a soft border point. Following the principle of label determination processing based on the Ek-NN rule and the radius approach of [23], [24], for each neighbor o_j of o_i within the ϵ_{max} radius, a mass function m_{ij} representing the membership of o_j can be assigned following Equations 5 and 6. In our method, α_{ij} can be determined by the dissimilarity between the soft border point o_i and its neighbor o_j, that is to say, α is high (respectively low) when d_{ij} is small (respectively big). Thus, α can be set as a decreasing function of d_{ij}:

$$\alpha_{ij} = exp\left(-\gamma_{k(j)} d_{ij}^2\right) \tag{11}$$

where α_0 from Equation 9 is set to 1 as default, β is set to 2 as default, and $\gamma_{k(j)}$ can be set to the inverse of the mean squared distance between points belonging to class $\omega_{k(j)}$ heuristically.

Using the Dempster rule of combination, we can induce border points memberships to clusters by combining the bbas of their neighbors within ϵ_{max} radius. Suppose that point o_i is a soft border point in overlapping regions, the evidence provided by its k neighbors are in the form of bbas $m_{i1}, ..., m_{ik}$ and thus the bba for point o_i's cluster membership can be obtained by combining the k pieces of evidence from neighbors.

B. Allocation of the remaining points

The remaining unlabelled points which do not belong to any neiborhood of core points are considered as outliers or noises and will be assigned to total ignorance class.

Figure 4 is an illustration of the E-DBSCAN method where points G and F are core points, B, C, D, E are border points, A is a soft border point, and red points are noisy points. Following the Ek-NN rule, the membership degree of point A is calculated by combining the bbas of its labelled neighbors B, C, D, E, F and G.

V. Experimental study

Within this section, we evaluate our proposed approach in order to prove its effectiveness and feasibility. In subsection A, we present the experimental framework we used and the parameter setup of each data sat. Then, in subsection B, we describe the results as well as a comparison between the proposed method and other existing methods.

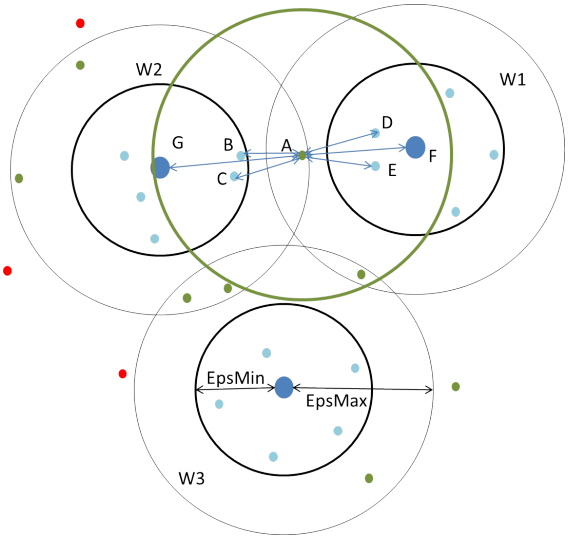


Fig. 4: E-DBSCAN illustration

A. Experimental setup

We first developed our proposed method E-DBSCAN using Python 3.7, then we tested it on classical datasets from the UCI Machine Learning Repository [25] and the "pdfCluster" package [26], also a synthetic dataset from the "EVCLUST" package [27]. The characteristics of these datasets are summarized in Table I. After that, once the results of the evaluation are established, we compared them with the following crisp and soft methods:

- K-means which is one of the simplest crisp clustering methods and require the number of clusters as input [28].
- DBSCAN which is a nonparametric crisp density-based clustering method [5].
- ECM which is a soft clustering method based on the belief function theory and require the number of clusters as input [17].
- E$k-$NNclus which is a soft clustering method based on the belief function theory, and requires the number k of neighbors and a scale parameter [19].

We evaluate the results using the Adjusted Rand Index (ARI) [29] defined by:

$$ARI = \frac{RI - Expected\_RI}{max(RI) - Expected\_RI} \tag{12}$$

where

$$RI = \frac{TP + TN}{TP + FP + FN + TN} \tag{13}$$

That measures the similarity between partitions. When two partitions are equal this index equals 1, and when the partitions are completely different this index equals 0. We note that this index only compares hard partitions, i.e., we refer to the pignistic transformation to compare the results with the hard clustering methods.

Algorithm 3: *DBSCAN(D,ϵ,MinPts)*

Input: ϵ_{Min}, ϵ_{Max}: soft constraint on the radius around a point to define neighbors and border points
Input: MinPts: minimum number of neighbors to be be considered as a core point
Data: D: Dataset

C=0
Clusters=$\emptyset$
forall *o* $\in D$ *s.t. o is unvisited* **do**
  mark o as visited
  neighborsPts = regionQuery(o,ϵ_{Min})
  if *sizeof(neighborsPts)* <= *MinPts* **then**
    Mark o as Noise
  else
    C = next cluster
    soft-borderPts = regionQuery(o, ϵ_{Max}) \ neighborsPts
    Clusters = *Clusters* $\cup$ expandSoftCluster(o, neighborsPts, soft-borderPts, C, ϵ_{Min},ϵ_{Ma}, MinPts)
    Calculate γ of the current cluster
forall *noisy point o* $\in D$ **do**
  $\underline{m}(\Omega) = 1$
forall *classified point o* $\in D$ *to cluster C* **do**
  m(C)=1
forall $o \in$ *soft-borderPts* **do**
  affect a membership degree using Equations 5, 6 and 11
return *Clusters*

Algorithm 4: *expandSoftCluster(o, neighborsPts, soft-borderPts, C, ϵ_{Min}, ϵ_{Max}, MinPts)*

Input: o: the point that was just marked as visited
Input: neighborsPts: the neighborhood of point o
Input: soft-borderPts: the soft border neighborhood of point o
Input: C: the current cluster
Input: ϵ_{Min} ϵ_{Max}: soft constraint on the radius around a point to define neighbors and border points
Input: MinPts: minimum number of neighbors to be be considered as a core point

add o to cluster *C*
forall $o' \in$ *neighborsPts* **do**
  if o' *is not visited* **then**
    Mark o' as visited
    neighborsPts$'$ = regionQuery(o',ϵ_{Min})
    if *sizeof(neighborsPts$'$)* > *MinPts* **then**
      neighborsPts = neighborsPts $\cup$ neighborsPts$'$
      soft-borderPts$'$ = regionQuery(o', ϵ_{Max}) $\setminus neighborsPts'$
      soft-borderPts = soft-borderPts $\cup$ soft-borderPts$'$
  if o' *is not yet member of any cluster* **then**
    add o' to cluster *C*
return *C*

We also used the Normalized Mutual Information (NMI) [30] to estimate clustering quality. It is defined by:

$$NMI(Y,C) = \frac{2 \times I(Y,C)}{H(Y) + H(C)} \tag{14}$$

Where Y is the class label, C is the cluster label, $H(x)$ is the entropy and $I(Y,C)$ is the Mutual Information between Y and C such that $I(Y,C) = H(Y) - H(Y|C)$. This index equals 1 for perfect correlation, and equals 0 for no mutual information.

For each dataset, we define the values of the parameters that we set for each algorithm in table II.

B. Experimental results

Results of the ARI measure for the proposed E-DBSCAN and the other methods on all the datasets are shown in Table III. For each couple of dataset and algorithm, we have given the best result.

We can see that the E-DBSCAN method identified correctly the number of clusters for all cases. The E$k-$NNclus and the DBSCAN method failed in detecting the correct number of clusters in both Iris and Wine datasets. As measured by the ARI criterion, E-DBSCAN outperformed the other method and gave a better partition quality for all the datasets except the Wine dataset. We can see that ECM gives slightly better results for the Wine dataset, however, the algorithm was initially provided with the correct number of partitions.

TABLE I: Characteristics of the real datasets.

Name	Clusters	Attributes	Instances
Iris	3	4	150
Wine	3	13	178
Four classes	4	2	400
Olive Oil	3	8	572
Statlog(Heart)	2	13	270

Best results of the NMI measure for the proposed E-DBSCAN and the other methods on all the datasets are shown in Figure 5. We can see that our proposed method gives better results for the Iris, Olive oil and Statlog (Heart) datasets and very close results compared to ECM for the Four-classes dataset. However, for the Wine dataset the K-means and the ECM gave better results and this can be explained by the fact that the number of clusters was given as an input which is considered as an extra knowledge regarding the correct number of the cluster partitions.

Comparing computing time results for the soft methods, seen in Table III, we can note that our method is significantly faster than the other soft methods for the Iris and the Wine dataset. However, for the Four-classes and the Olive oil datasets, the other methods were faster, and this can be explained by the high overlapping areas between the clusters

TABLE II: Datasets parameters

Dataset	K-means	ECM	EK-NNclus	DBSCAN	E-DBSCAN
Iris	3	3	k=30 q=0.9	$MinPts = \{3, \ldots, 10\}$ $\epsilon = \{0.3, \ldots, 0.5\}$	$MinPts = \{3, \ldots, 10\}$ $\epsilon_{Min} = \{0.3, \ldots, 0.5\}$ $\epsilon_{Max} = \{0.4, \ldots, 0.8\}$
Wine	3	3	$k = \{40, \ldots, 50\}$ $q = \{0.5, \ldots, 0.8\}$	$MinPts = 3$ $\epsilon = \{30, \ldots, 50\}$	$MinPts = 3$ $\epsilon_{Min} = \{30, \ldots, 50\}$ $\epsilon_{Max} = \{45, \ldots, 50\}$
Four-classes	4	4	$k = \{40, \ldots, 60\}$ $q = \{0.7, \ldots, 0.9\}$	$MinPts = 20$ $\epsilon = \{0.7, \ldots, 0.9\}$	$MinPts = 20$ $\epsilon_{Min} = \{0.7, \ldots, 0.9\}$ $\epsilon_{Max} = \{3.5, \ldots, 4\}$
Olive oil	3	3	$k = \{100, \ldots, 150\}$ $q = \{0.7, \ldots, 0.9\}$	$MinPts = 10$ $\epsilon = \{100, \ldots, 130\}$	$MinPts = 10$ $\epsilon_{Min} = \{100, \ldots, 130\}$ $\epsilon_{Max} = \{130, \ldots, 150\}$
Statlog(Heart)	2	2	$k = \{30, \ldots, 50\}$ $q = \{0.6, \ldots, 0.9\}$	$MinPts = \{5, \ldots, 10\}$ $\epsilon = \{15, \ldots, 25\}$	$MinPts = \{5, \ldots, 10\}$ $\epsilon_{Min} = \{15, \ldots, 25\}$ $\epsilon_{Max} = \{25, \ldots, 35\}$

TABLE III: ARI measure for the proposed E-DBSCAN and the other methods on the datasets

Dataset	Result	K-means	DBSCAN	ECM	EK-NNclus	E-DBSCAN
Iris	K	3	2	3	6	3
	ARI	0.730	0.520	0.589	0.188	**0.790**
	time	0.030	0.062	0.573	0.215	0.211
Wine	K	3	6	3	6	3
	ARI	0.371	0.281	**0.429**	0.284	0.364
	time	0.034	0.003	0.687	0.367	0.157
Four-classes	K	4	4	4	4	4
	ARI	0.734	0.465	0.735	0.728	**0.735**
	time	0.046	0.004	1.735	0.574	6.528
Olive oil	K	3	3	3	3	3
	ARI	0.318	0.527	0.328	0.388	**0.546**
	time	0.054	0.011	1.109	2.569	1.928
Statlog(Heart)	K	2	2	2	4	2
	ARI	0.030	0.065	0.011	0.060	**0.065**
	time	0.032	0.006	0.381	0.428	0.027

in these datasets.

VI. Conclusion

In this paper, we developed a new soft clustering method for the Density-Based Spacial Clustering Application with Noise (DBSCAN), that we called Evidential DBSCAN or E-DBSCAN, using the framework of belief function theory. The aim of this method is to model distinct density-based spatial distribution of objects in the feature space. A soft constraint was defined to specify an approximate local density around points in order to handle the cluster membership uncertainty problem and to generate overlapping clusters. Results of the experimental comparison between our proposed method and other state of the art methods over real and synthetic datasets proved a better performance w.r.t the ARI and NMI criteria which highlight the efficiency of our proposal. In future work, a study on the parameter estimation for the E-DBSCAN can be done to enhance, furthermore, its performance.

References

[1] Jih-Jeng Huang, Gwo-Hshiung Tzeng and Chorng-Shyong Ong. Marketing segmentation using support vector clustering. In: Expert Systems with Applications 32, pages 313-317, 2007.

[2] Fan Cai, Nhien-An Le-Khac and Tahar Kechadi. Clustering Approaches for Financial Data Analysis: a Survey, 2016.

[3] Grainne Kerr, Heather J. Ruskin, Martin Crane and Padraig Doolan. Techniques for clustering gene expression data. In: Computers in Biology and Medicine 38, pages 283-293, 2008.

[4] Nameirakpam Dhanachandra and Yambem Jina Chanu. A Survey on Image Segmentation Methods using Clustering Techniques. In: European Journal of Engineering Research and Science 2, pages 15-20, 2017.

[5] Martin Ester, Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining 96(34), pages 226-231, 1996.

[6] Zexuan Ji, Yong Xia, Quansen Sun, Guo Cao. Interval-valued possibilistic fuzzy c-means clustering algorithm. In Fuzzy Sets and Systems, pages 138–156, 2014.

[7] Nikhil Ranjan Pal, Kuhu Pal, James Keller and James Bezdek. A possibilistic fuzzy cmeans clustering algorithm. In IEEE Transactions on Fuzzy Systems 13, pages 517–530, 2005.

[8] Ronald R. Yager and Dimitar Filev. Approximate clustering via the mountain method. In IEEE Transactions on Systems, Man, and Cybernetics 24, pages 1279–1284, 1994.

[9] Gözde Ulutagay and Efendi N. Nasibov. Fuzzy and Crisp Clustering Methods Based on The Neighborhood Concept: A Comprehensive Review. IN Journal of Intelligent and Fuzzy Systems 23, pages 271-281 , 2012.

[10] Abir Smiti and Zied Eloudi. Soft dbscan: improving dbscan clustering method using fuzzy set theory. In International Conference on Human System Interactions, pages 380–385, 2013.

[11] Gloria Bordogna and Dino Ienco. Fuzzy core dbscan clustering algorithm. In International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, pages, 100–109, 2014.

[12] Jonathon Parker, Lawrence Hall and Abraham Kandel. Scalable fuzzy

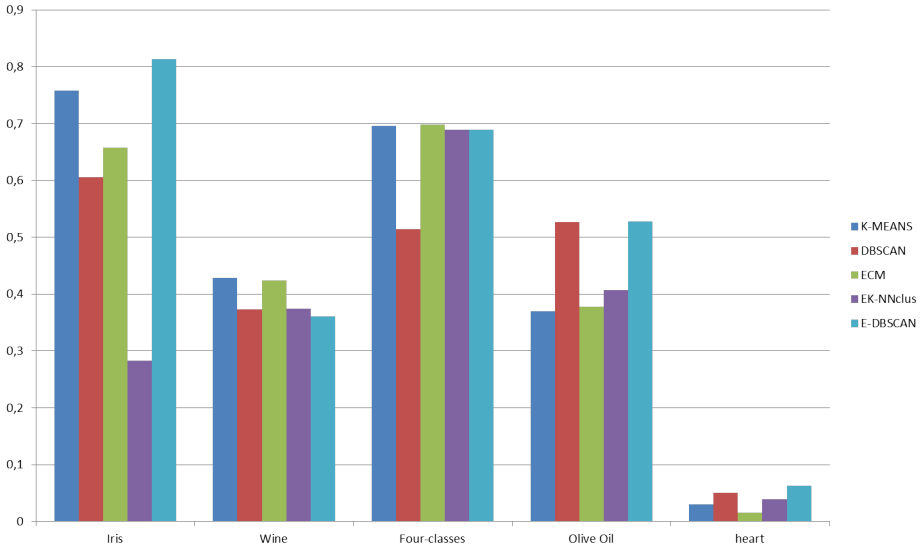


Fig. 5: NMI measure for the proposed E-DBSCAN and the other methods on the datasets

neighborhood DBSCAN. In IEEE International Conference on Fuzzy Systems, pages 1-8, 2010.

[13] Gloria Bordogna and Dino Ienco. Fuzzy extensions of the DBScan clustering algorithm. In Soft Computing 22, pages 1719–1730, 2018.

[14] Thierry Denœux and Orakanya Kanjanatarakul. Evidential Clustering: A Review. In: International Symposium on Integrated Uncertainty in Knowledge Modelling and Decision Making (IUKM), Springer, pages 24-35, 2016.

[15] Glenn Shafer. A mathematical theory of evidence 42. In Princeton university press, 1976.

[16] Thierry Denoeux and Marie-Hélène Masson. EVCLUS: evidential clustering of proximity data. In IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 34, pages 95-109, 2004.

[17] Marie-Hélène Masson and Thierry Denoeux. ECM: An evidential version of the fuzzy c-means algorithm. In Pattern Recognition 41, pages 1384-1397, 2008.

[18] Marie-Hélène Masson and Thierry Denoeux. RECM: relational evidential c-means algorithm. In Pattern Recognition Letters 30, pages 1015–1026, 2009.

[19] Thierry Denoeux, Orakanya Kanjanatarakul nad Songsak Sriboonchitta. EK-NNclus: a clustering procedure based on the evidential k-nearest neighbor rule. In Knowledge-Based Systems 88, pages 57–69, 2015.

[20] Thierry Denoeux. A k-nearest neighbor classification rule based on dempster-shafer theory. IEEE Transactions on Systems, Man, and Cybernetics 25, pages 804-813, 1995.

[21] Arthur P. Dempster. Upper and lower probabilities induced by a multivalued mapping. In The Annals of Mathematical Statistics 38, pages 325–339, 1967.

[22] Philippe Smets. The transferable belief model and other interpretations of dempster-shafer's model. In Uncertain Artificial Intelligence 6, pages 375-384, 1990.

[23] Philippe Smets. Belief functions: The disjunctive rule of combination and the generalized bayesian theorem. In International Journal of Approximate Reasoning 9(1), pages 1-35, 1993.

[24] Thierry Denoeux and Philippe Smets. Classification Using Belief Functions: Relationship Between Case-Based and Model-Based Approaches. In IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 36, pages 1395-1406, 2006.

[25] https://archive.ics.uci.edu/ml

[26] Adelchi Azzalini and Giovanna Menardi. The pdfCluster-package, 2014. R package.

[27] Thierry Denœux. Evclust: Evidential Clustering, 2016. R package version 1.0.2.

[28] Jeffrey MacQueen. Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, pages 281-297, 1967.

[29] Lawrence Hubert and Phipps Arabie. Comparing partitions. In Journal of Classification 2, pages 193-218, 1985.

[30] Alexander Streh and Joydeep Ghosh. Cluster ensembles - A knowledge reuse framework for combining multiple partitions. In Journal of machine learning research 3, pages 583-617, 2002.