Fuzzy Neighborhood Grid-Based DBSCAN Using Representative Points

Oguz Altun and Abdallah Mekky
Computer Engineering Department
Yildiz Technical University
Istanbul, Turkey
oaltun@yildiz.edu.tr, Abdallah_makki@outlook.com

ABSTRACT

Clustering process is considered as one of the most important part in data mining, and it passes through many levels of developments. One of the most famous algorithm is Density-Based Spatial Clustering of Application with Noise (DBSCAN) [1,2,4]. It is a density-based clustering algorithm that uses a crisp neighborhood function to calculate the neighbor sets, and basically depends on distance function. In fuzzy clustering [9], which is considered as a soft clustering algorithm, it uses a fuzzy neighborhood function that allow a node in the dataset to have a membership degree in each point in the dataset. In this paper we propose a new algorithm that depends on both bases the speed of DBSCAN and the accuracy of fuzzy clustering. FNGMDBSCAN-UR is a Fuzzy Neighborhood Grid-based Multi-density DBSCAN Using Representative points. That uses grid-based to separate the dataset into small nets and fuzzy neighborhood function to create neighborhood sets.it is noticeable that FNGMDBSCAN-UR is much accurate than crisp DBSCAN with nested shapes and multi-dense datasets as we will see in the result section in this paper.

KEYWORDS

Clustering; Fuzzy neighborhood function; DBSCAN; FN-DBSCAN; GMDBSCAN-UR; Density-based; Grid-based Clustering.

1 INTRODUCTION

Cluster analysis is an unsupervised learning method that constitutes a cornerstone of an intelligent data analysis process. “It is used for the exploration of inter-relationships among a collection of patterns, by organizing them into homogeneous clusters” [1]. The main goal of clustering process is to simplify the statistical analysis by grouping the similar objects together in the same cluster. And when we mention similarity here it means the objects are related in some way to each other in some characteristics. Many methods of clustering are existing and they are belonging to different groups such as hierarchal clustering methods, partitioning/ prototype-based, density or neighborhood-based, grid-based methods and so on, [1,2,3]. In hierarchical clustering, the remoteness of elements is the main point, the closest elements are come into the same cluster as a first step, and then elements that are a little bit far away are put in the same cluster and so on [1]. In partitioning-based methods, algorithms learn cluster directly, since the prototypes which have common features of some certain classes are formed and then the elements are taken in to these classes according to the similarity degree to the prototypes. In other words, the clusters try to discover the clusters iteratively relocating points between subsets, or try to identify clusters as areas highly populated with data. In such a situation, not the remoteness of the above-mentioned elements from each other, but their remoteness from the prototypes is considered. Some example these methods are K-means [1,3], K-medoids [1,13]and Fuzzy c-means (FCM) [4,8]. Density-based algorithms typically classify clusters as dense region of objects in the data space that are separated by region of low density. The main idea of density-based approach is to find regions of high density and low density, with high-density regions being separated from low-density region. This approach can make it easy to discover arbitrary clusters. Some example of density-based

algorithms, DENsity-based CLUstEring (DENCLUE) and Ordering Points to Identify Clustering Structure (OPTICS) [8,12,15]. Density-based spatial clustering of applications with noise (DBSCAN) is one of the important algorithms, but it can be handled as hierarchical and density-based algorithms [2]. For DBSCAN to determine the core points of the clusters or noise points, a classical neighborhood density analysis is performed, so that a point called a core point if and only if the number of points in the assigned radius is equal or larger than a threshold, thus the point is just assigned to one cluster, which we call it a hard clustering method [2, 3], while on the other hand for example Fuzzy Joint Points (FJP) [14] method uses fuzzy neighborhood cardinality in order to determine core points and each point in the data set space can belong to many cluster depending on its neighborhood participation degree. For the grid-based methods the main concept is to divide the data space into equivalent cells and then perform the required operations on the quantized space. Grids that contains points more than a specific threshold are considered as dense cells, and they connect to each other to create a cluster [1].

DBSCAN algorithm pass through many levels of development since it is the algorithm that handle with data set and discover clusters of arbitrary shape as well as to distinguish noise, algorithms such as multi-density DBSCAN (MDBSCAN), grid-based multi-density DBSCAN (GMDBSCAN) and GMDBSCAN using representative points are proposed recently. Moreover, a new methodology that depends on fuzzy set theory that allows the object to have varying grades of membership in a set has been applied to the clustering algorithms like k-mean and DBSCAN as well. This can be applied to the case of a data point having a grade of membership in multiple clusters [4]. In this paper we will present a new clustering algorithm that depends on fuzzy set method and GMDBSCAN-UR which we will explain in the third section.

The rest of this paper is organized as follows. We will introduce a quick overview of clustering methods. Where we concentrate on crisp DBSCAN, GMDBSCAN-UR and FNDBSCAN clustering methods.

Section 3 describes our new algorithm FNGMDBSCAN-UR clustering method in detail. Experimental setup and results in the fourth section. The paper conclusion in section 5.

2 RELATED WORK

Over many years ago data mining became one of the most important section in machine leaning and pattern recognition as well, clustering analysis is the answer of the question that comes to our mind when we have a data objects and the distance function between them, are they create groups (clusters)? What that groups look like?

Clustering algorithms are the algorithms that answer our wonders. In this section we are going to discuss three clustering algorithms that lead to our new clustering algorithm. The first sub-section will be about crisp DBSCAN algorithm which is the core of the density-based clustering algorithms and its limitations.

Section 2.2 we will discuss GMDBSCAN using representative points which is one of the developed algorithms from DBSCAN and comes over many of DBSCAN limitations, then in section 2.3 we will go over one of the core clustering algorithms that mixed between fuzzy clustering and density-based clustering algorithms FN-DBSCAN and its advantages and disadvantages.

2.1 K-Means Algorithm

K-Means algorithm is considered as one of the simplest and first algorithms in clustering field. It depends on the distance between nodes, and uses the mean between the nodes and the centroids. The idea behind K-means clustering algorithm is; assigning a group of points that called “Centroids” and for each point in the dataset must be assigned to the nearest cluster center. After that centroids start update itself by calculate the mean of the cluster and make it as the cluster center. The algorithm stop working when there is no new update on its instances.

2.2 Crisp DBSCAN Algorithm

DBSCAN (Density-Based Spatial Clustering of Application with Noise) is density-based clustering algorithm, which grows regions with sufficiently high density into clusters and discovers clusters of arbitrary shape in spatial databases with noise [3,7,8]. The main idea of DBSCAN is to classify the dataset into two types of regions, the high dense region that will be used as a part of cluster and the low dense region that will be considered as noise [1,3,8,13]. The data points that are in the same or in the neighborhood dense area are considered in the same cluster. The algorithms depend on two main parameters the radius (eps) of objects and the minimum number of points in that radius (MinPts) [1,3,8,10]. If the number of MinPts in the radius are equal or greater than the threshold, then the area is dense and the object will be considered as a core point [1,3]. In DBSCAN data points has three classifications:

- Core point: is the point that contains neighbors more than or equal to the MinPts in its radius.
- Directly density-reachable point: is the point that in the neighborhood of a core point and can be reached directly from core point
- Density reachable point: is the point that can be connected with a core point through a chain of points that connected with that core point

Algorithm 1 below illustrates the way that DBSCAN works [8].

Algorithm 1: Basic DBSCAN Algorithm

```
1: Begin
2: randomly select a point p
3: Retrieve all points density-
   reachable from p Eps and MinPts
4: If p is a core point a cluster is
   formed
5: If p is a border point, no points
   are density-reachable from p and
   DBSCAN visits the next point of the
   database
6: Repeat step 4,5 until all of the
   points have been processed
7: End
```

DBSCAN algorithm has many limitations. DBSCAN is considered as a very sensitive algorithm to its parameters (MinPts, eps) [1,3,8], so that any change in those parameters will cause an error in the number and the shape of clusters. Also calculating eps is considered as time consuming process that makes the algorithm slow [3], that's for eps should be entered by the user, moreover using a global MinPts is not sufficient for multi-density datasets, which makes the result of clustering process inaccurate.

2.3 GMDBSCAN-UR Algorithm

Grid-based Multi-Density DBSCAN Using Representative points algorithm that comes to handle with problems of DBSCAN, the algorithm works by applying seven steps to come out with a much better accurate results. The first step for GMDBSCAN-UR is to divide the dataspace to equivalent grids [3], each grid contains a number of points, and the algorithm deals with each grid as a separated data space that has its own MinPts and radius [1,3]. Grids will be either a dense grid that has a number of points that equal or greater than MinPts, or a noise grid that has number of points less than MinPts. Then for each non-empty grid the algorithm takes a number of points in the grid as some keywords, then moves through dimensions until reach d+1 dimension and in each dimension looks for the node in the corresponding layer of SP-Tree [3]. If the corresponding number is existing in SP-Tree then it goes to the next dimension and construct a new leaf node to the grid if the corresponding number is not existing in SP-Tree then it creates the nodes from this layer, as illustrated in figure 2.1 that shows the SP-Tree constructing step [3]. Third step is bitmap formatting. As we mentioned before the GMDBSCAN-UR algorithm deals with each grid as a local data space so that it applies the DBSCAN method to calculate the distances between nodes in the same grid and store the information about neighbors in a bitmap. The fourth step, is dealing with each grid before staring a clustering process as local-MinPts according to the region that the grid located in [1,3]. if the grid volume (VGrid) [3] is not equal to the Neighborhood volume (VEps)

[3], then we use a correction factor which is: VEps/ VGrid [3,16]. And the relation between the Grid Density (GD) [3] and MinPts is: factor = MinPts/ GD [3,16]. The fifth step is to choose representative points that represent the data set. Representative points must represent the dataset's shape very well and dense of the dataset as well, after this step the algorithm use crisp DBSCAN to start clustering process and construct the cluster and core points, also in each iteration the algorithm will start merging the clusters and after finishing clustering we start label the points that weren't in the representative set to each cluster and turn the merging process again to reach to the final number of clusters [3]. The last step is dealing with noise and border processing the algorithm sets parameters according to the size of the dataset. When the number of points in the cluster is less than these parameters then all the cluster considered as noise [2, 3].

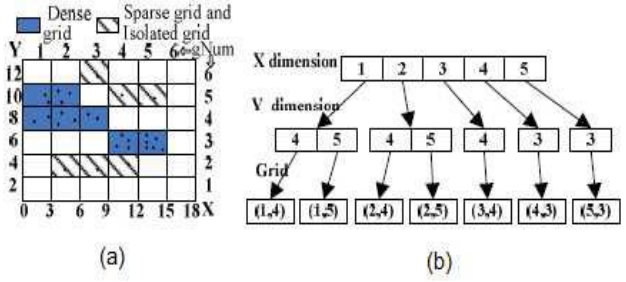

Figure 1. Framework of Constructing a SP-Tree.

GMDBSCAN is time consuming to perform well on large datasets. Moreover, as a hard clustering algorithm it is not considered as a very accurate clustering algorithm in spitting of the good results that gives comparing with crisp DBSCAN [3].

2.4 FN-DBSCAN Algorithm

Fuzzy neighborhood DBSCAN algorithm is one of the algorithms that combined between hard clustering (DBSCAN) and soft clustering (fuzzy clustering) [5,8], the main concept of the Fuzzy sets theory is that for each point in the dataset it has a neighborhood membership degree, which means the data object can belong to many cluster depends on the neighborhood membership degree [1,9]. Fuzzy set approaches have been integrated successfully with many clustering algorithms such as K-means and DBSCAN algorithms [2].

FN-DBSCAN algorithm is similar to DBSCAN clustering algorithm with one main different that, instead of using the distance-based function to find the neighbors of the point, it uses the fuzzy neighborhood function to find the neighbors set and core points. For better understanding of the difference between distance-based function and fuzzy neighborhood function let's investigate about points x_1 and x_2 in figure 2.2. Both nodes x_1 and x_2 have the same number of neighbors within the radius Eps $\leq d^{max}$ radius. According to crisp neighborhood relation that used in DBSCAN both points are the same, but in fuzzy neighborhood function point x_1 will have a higher membership degree of being a core point than that of point x_2. The used fuzzy neighborhood function that used here is:

$$N_x(y) = \begin{cases} 1 - \dfrac{d(x,y)}{d^{max}} & if\ d(x,y) \leq \varepsilon \\ 0 & otherwise \end{cases} \quad (2.1)$$

Where: $N_x(y)$ is the neighborhood membership degree for the node, d^{max} is the max $d(x,y) \leq 1$, and $d(x,y)$ is the distance between x and y points.

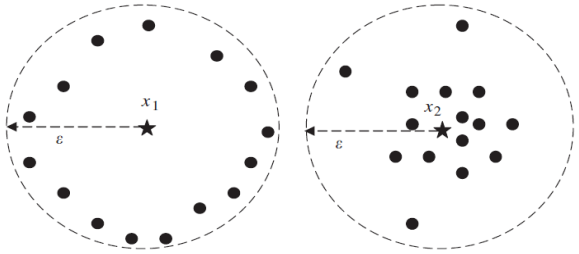

Figure 2. Points $x_1 and\ x_2$ are similar according to crisp neighborhood cardinality, but dissimilar according to fuzzy neighborhood cardinality.

In fuzzy neighborhood function the definition of core point and the neighborhood set are different than distance-based function.as shown in the following definitions:

- Fuzzy core point: is the point that its summation of neighborhood degree is greater than the minimal threshold that given in the specific radius.

- Fuzzy neighborhood set for point: is for each point in the radius of point x must has a fuzzy neighborhood degree that greater than the threshold.

Algorithm 2 is FN-DBSCAN algorithm's steps and it shows the sequential of processes that algorithm applies on the data set.

Algorithm 2: FN-DBSCAN Algorithm

```
1:  Specify parameters ε1 and ε2
2:  Mark all the points in the dataset
    as unclassified. Set t=1.
3:  Find an unclassified fuzzy core
    point with parameters ε1 and ε2
4:  Mark p to be classified. Start a
    new cluster Ct and assign p to the
    cluster Ct.
5:  Create an empty set of seeds S.
    Find all the unclassified points
    in the set N(p, ε1) and put all
    these points into the set S
6:  Get a point q in the set S, mark
    q to be classified, assign q to
    the cluster Ct, and remove q from
    the set S.
7:  Check if q is a fuzzy core point
    with parameters ε1 and ε2, if so,
    add all the unclassified points in
    the set N(q, ε1) to the set S.
8:  Repeat steps 6 and 7 until the set
    of seeds is empty
9:  Find a new fuzzy core point p
    with parameters ε1 and ε2 and
    repeat steps 4-7
10: Mark all the points, which do not
    belong to any cluster as noise.
11: End
```

FN-DBSCAN algorithm always gives better results than DBSCAN algorithm does by adjusting appropriate neighborhood membership functions. The main advantage of transformation of DBSCAN algorithm to the FN-DBSCAN algorithm and using fuzzy sets theory is that various neighborhood member-ship functions that regularize different neighbor-hood sensitivities can be utilized [2].

3 METHODOLOGY

3.1 Overview

The general overview of the proposed approach that, we will use the crisp DBSCAN algorithm for clustering, but before that the dataset will pass through many steps that will make the results more accurate. The algorithm will be divide the data space to grids, and each grid will be handled as an independent data space in calculating the fuzzy core points and the fuzzy neighborhood set as well. Instead of using the distance-based function, we will use fuzzy neighborhood function, then we will take representative points that represent the shape and the dense of the dataset. In the later step we will use DBSCAN algorithm in the clustering process, and merging the clusters according to DBSCAN rules, after that the rest of the dataset that didn't involve in clustering process will be labeled to clusters and remerge the clusters again to reach the final results of cluster. The following subsections discuss each step in FNGMDBSCAN-UR algorithm.

3.2 Dataset Input and Standardization

The aim of Data Standardization is to make the data dimensionless [3,8,16]. So it helps in defining some standard indices. The location and the scale of the data that originally the algorithm takes will not stay the same and most probably it will be lost, but it is necessary to standardize variables because we are going to calculate the similarity and dissimilarity for the dataset. The Z-Score method [3,16] of standardization is one of the famous technique that transform the normal variants to standard score form, and it is used in our algorithm.

$$x_{ij} = Z_1\left(x_{ij}^*\right) = \frac{x_{ij}^* - \bar{x_{ij}^*}}{\sigma_j^*} \qquad (3.1)$$

Where $\bar{x_{ij}^*}$ and σ_j^* are the sample mean and standard deviation of the jth attribute respectively. The transformed variable will have a mean of 0 and a variance of 1. And the location and the original data's information has been lost [16].

3.3 Divide Data Space into Grids

In this step grid-based technique is used to divide the data space to small grids, each grid contains number of points that are not matching to all the other cells. Moreover, cells are different in their density depends on the number of objects in each of them [3]. Figure 3 shows an example of a data space that partitioned into small grids. The point from partitioning the data set is to deal with each grid as a separate unit in order to make a local clustering on it.

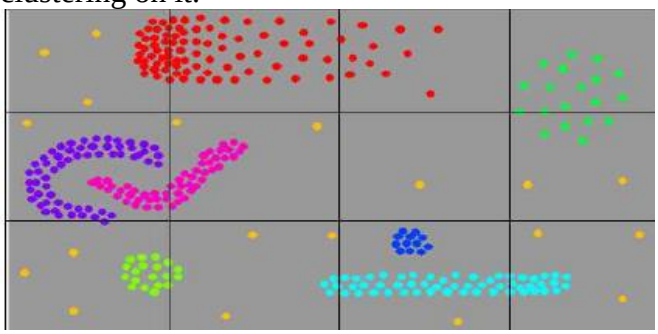

Figure 3. Divided dataset into small grids

3.4 Choose Representative points

After dividing the data space to grids, we choose number of points to represent the data in the clustering process, the data points that will be chosen as representative must satisfy an important condition, which is the scattered points in the cell must capture the shape and extent of the cell as shown in Figure 4 [3], the black points in the figure 4 below are the representative points, and it is obvious that it is less than the original number of points in the dataset. The benefits of this step is to give the proposed algorithm" FN-GMDBSCAN-UR" a good improvement in time consumption.

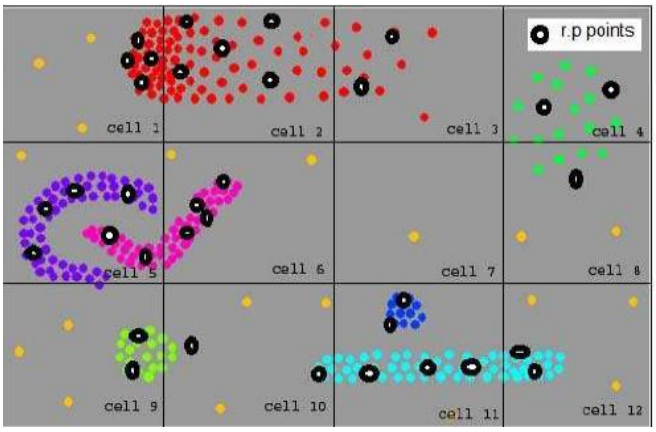

Figure 4. Black points are the representative points in each cell

In our algorithm we go through each level in the SP-Tree and choose percentage number of points to be our representative points that going to enter the clustering process firstly, and the rest of the points will be stored as a different dataset that will be used later in labeling step.

3.5 Selecting MinPts, Eps and Ɛ2 Parameters

Two approaches are available to apply for selecting the minimum number of points and the radius, the first way is to put the radius as constant and use varying MinPts from cell to cell. And the second method is to make the MinPts as a constant and a varying radius that depends on the point's presentation in the data set [3,16]. For each cell in the dataset we calculate Ɛ2 parameter which is representing the density of the point and if the summation of the Fuzzy cardinality of the point is bigger than this parameter then the point will be considered as a Fuzzy core point. For calculating this parameter, we have to calculate the maximum cardinality of the point in each cell with a certain eps. Equation 3.2 shows the threshold formula.

$$Ɛ2 = \alpha * \frac{MinPts}{w^{max}} \qquad (3.2)$$

Where $w^{max} = max_{i=1,2,\ldots,n} w_i$ and w_i is the summation membership degree of points in the radius eps to the neighborhood set.

α : is a constant double number that used as a correction factor and it changes from dataset to another depending on the dense of the data set. If the dataset is a dense dataset, then the factor will

be big. The value of this factor comes from experiment.

3.6 Bitmap Formatting

In this step most of the statistical results that are needed in the next step will be ready and saved in a bitmap [3]. The fuzzy neighborhood membership calculation between two nodes which are exists in the same or adjacent cells is done and the result will be compared with the radius eps and the information stored in bitmap [3,16]. If the distance between nodes is less than radius and the fuzzy membership degree of the point $N_x(y)$ is equal or bigger than Minimum cardinality then the point will be add to fuzzy neighborhood set and it will be stored in bitmap. Same thing happens to the fuzzy core points where the fuzzy core point is the point that its fuzzy membership degree is bigger than or equal to Ɛ2 for each cell. Where:

$$N_x(y) = \begin{cases} 1 - \frac{d(x,y)}{d^{max}} & if\ d(x,y) \le Eps \\ 0 & otherwise \end{cases} \quad (3.3)$$

One of the important parameter that helping at calculate the minimum number of points MinPts is Cell Density, denoted by CD [16], and it is the amount of data in a cell. Taking CD as local-MinPts approximation of the cell region. If the Cell Volume (VCell) is not equal to the data point's neighborhood volume (VEps), we set a factor to correct it. $factor = \frac{VEps}{VCell}$ [3]. The relationship of CD and MinPts is:

$$\text{Factor} = \text{MinPts}/\text{CD} = VEps/VCell \quad (3.4)$$

$$MinPts = factor * CD \quad (3.5)$$

3.7 Local Clustering and Merging

In this step original DBSCAN algorithm is used locally in each cell using the computed MinPts, eps, Fuzzy neighboring degree and Ɛ2 [2,7]. For each cell, processing clusters with their local-MinPts to form a number of distributed local clusters. This step contains many sub-steps to make a local clusters and merging the similar sub-clusters as follow:

- First is to select the densest cell and hasn't been clustered, and we deal with borders. The cell is considered as dense cell if its density is bigger than or equal to the predefined threshold.
- The cells that are close to the dense cells but not dense cell by themselves are called a sparse cell [3], and this kind of cells has two classifications either a border or noise. Isolated cells are the cells that they are not dense cells and not close to a dense cell so they could be regulated as noise. In DBSCAN, if the border object is in the scope of eps neighborhood of different core object, it is classified into the cluster to sort firstly [7]. In our algorithm "FN-GMDBSCAN-UR" we set such object to the cluster whose fuzzy core object is the nearest to this object. Second we compute MinPts for each data in cell which its equation is 3.5 equation.
- Then cluster with original DBSCAN algorithm and for each unvisited point, P, in the dataset, D, mark P as visited and compute the fuzzy neighborhood degree of the point p, and compare this number with the fuzzy neighborhood threshold Ɛ2 if it is less than the threshold then it is a noise, otherwise it is a fuzzy core point and so on. Then expand the current cluster.
- If the data belong to another sub-cluster, then merge the two clusters, and if not, assign it to the cluster who has the nearest representative point from this point and tag the data as a new cluster.

3.8 Labeling and Post processing

Until this point we have a number of clusters that came as a result of clustering process, but as we mentioned before the clustering process just occurred on the representative dataset, while the rest of the original dataset didn't involve in the clustering process so that this step comes to start labeling each point to its nearest point that belong to representative dataset and already clustered, in this case all the points will be allocated in a cluster, and the post processing method starts, if there are two clusters that have nearly the same density and close to each other the remerging method will

merge both of them. Now in this point we have all points that belong to the dataset are clustered.

3.9 Noise Elimination

Any data set almost always contains outliers. These do not belong to any of the clusters. That is, the neighborhoods of outliers are generally sparse compared to points in clusters, and the distance of an outlier to the nearest cluster is comparatively higher than the distances among points in points in the clusters themselves [16]. Every clustering method needs mechanisms to eliminate outliers. In FNGMDBSCAN-UR, outliers due to their larger distances from other points, tend to merge with other points less and typically grow at a much slower rate than actual clusters [3]. Thus the clusters which are growing very slowly are identified and eliminated as outliers [3,16]. Also, since the number of points in a collection of outliers is typically much less than the number in a cluster and that outliers form very small clusters, we can easily identify such small groups and eliminate them. Consequently, the final step, the outlier elimination, is an important step for good clustering [3].

Algorithm 3: FNGMDBSCAN-UR Algorithm

```
1:  Divide data space to equal grids
2:  Take representative points that
    represent the dense and shape of
    the original dataset
3:  Specify parameters ε1 and ε2 and
    MinPts
4:  Mark all the points in the
    dataset as unclassified. Set t=1.
5:  Find an unclassified fuzzy core
    point with parameters ε1 and ε2
6:  Mark p to be classified. Start a
    new cluster Ct and assign p to the
    cluster Ct.
7:  Create an empty set of seeds S.
    Find all the unclassified points
    in the set N(p, ε1) and put all
    these points into the set S
8:  Get a point q in the set S, mark
    q to be classified, assign q to
    the cluster Ct, and remove q from
    the set S.
9:  Check if q is a fuzzy core point
    with parameters ε1 and ε2, if so,
    add all the unclassified points
    in the set N(q, ε1) to the set S.
10: Repeat steps 6 and 7 until the
    set of seeds is empty
11: Find a new fuzzy core point p
    with parameters ε1 and ε2 and
    repeat steps 4-7
12: Label all the non-representative
    points to its nearest
    representative points
13: Start post processing and
    remerging clusters
14:  Mark all the points, which do
    not belong to any cluster as
    noise.
15: End
```

4 EXPERIMENTS

In this section we will show the results after we applied three clustering algorithms: crisp DBSCAN, GMDBSCAN-UR and our proposed algorithm FNGMDBSCAN-UR on the datasets. The following sub-sections will contain many figures that illustrate the results. In order to test our algorithm FNGMDBSCAN-UR which is based on fuzzy neighborhood analysis with GMDBSCAN-UR and the main GMDBSCAN also the Crisp DBSCAN, we use mainly in this study two datasets one is the artificial chameleon dataset and the real iris dataset as illustrated in the next section. The algorithms were programmed in java and the experiments were done on Intel core i5, 2.50GHZ, 6GB RAM hp. Brand computer.

4.1 Data Sets

In this paper we use two testing datasets, the first one is Iris dataset that is one of the most famous datasets that used in data mining field, it contains 150 data object and 5 main attributes, the first 4 attributes are numerical and the fifth one is a string that shows the class that object belongs to. In our research we use the first and the third attributes, which contain the values of sepal length and petal length. Figure 5 shows the spread of the objects on the data space.

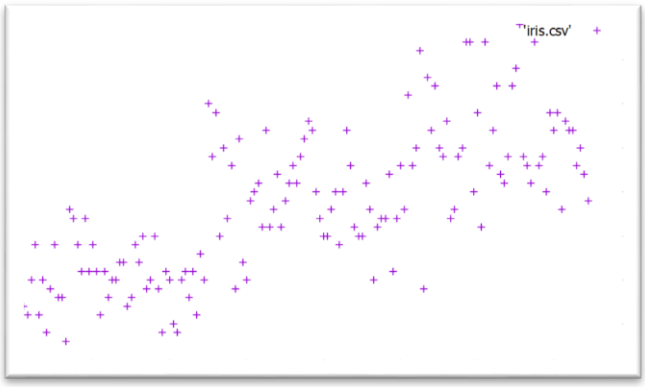

Figure 5. Iris dataset

Also, we use chameleon dataset It is one of the important datasets in machine learning and data mining field, it is an artificial dataset contains 8000 data element, in our research we use the first and the second double attributes. This dataset considered as a not simple, nested shapes dataset that contains a nested multi-dense clusters, each cluster has an arbitrary shape and a lot of noise. Figure 6 shows the dataset objects over the dataset space [3].

Figure 6. Chameleon Dataset

4.2 IRIS Dataset's Result

Applying crisp DBSCAN algorithm on a small dataset, that is not complex and doesn't have many clusters with a nested shape like IRIS will give a result in fast way, but not the "fastest" and the accuracy is so low since the result will have an unignorably error. We can see in figure 7 the algorithm gives 4 clusters and some noise. GMDBSCAN-UR solves the problems of crisp DBSCAN and also many algorithm problems', since the algorithm divides the data space into grids and start the local clustering, then start merging between the sub-clusters. The algorithm shows very good results for chameleon dataset, since it can catch the noise in a much better way, and that's obvious in figure 8. When applying our proposed algorithm on the same dataset iris and chameleon we will see a very good results that approve how FNGMDBSCAN-UR solved the problems of other clustering algorithms that we reviewed in this paper. FNGMDBSCAN-UR can recognize the noise and the outliers much better than the other clusters moreover, it shows the exact arbitrary shape of each cluster in the dataset. we notice in GMDBSCAN-UR algorithm that uses the hard clustering method the shape of the clusters is not so arbitrary and there is some overlap between clusters and the recognition of noise and outliers' detection process is not so accurate.

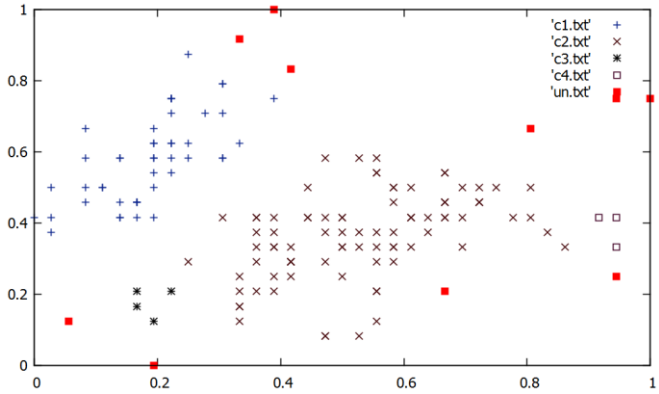

Figure 7. Iris dataset clustering result using crisp DBSCAN algorithm

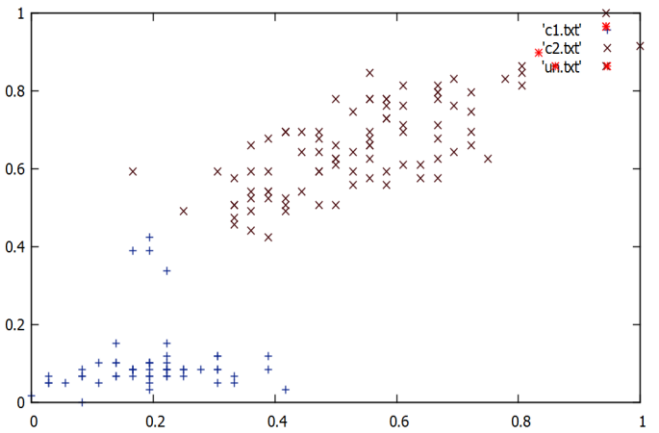

Figure 8. IRIS dataset clustering result using GMDBSCAN-UR algorithm

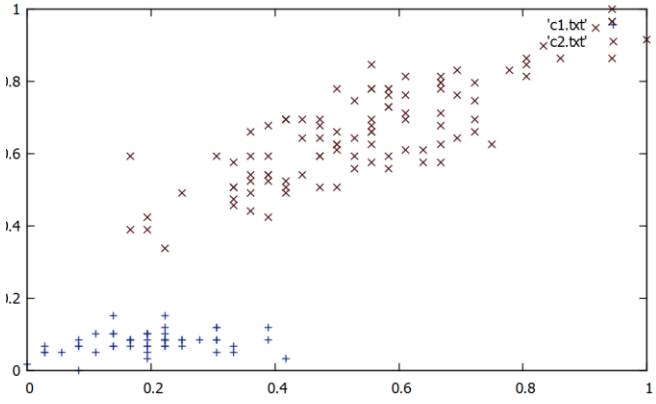

Figure 9. IRSI dataset clustering result using FNGMDBSCAN-UR algorithm

4.3 Chameleon Dataset's Result

Using DBSCAN algorithm on multi-dense dataset as Chameleon dataset will not come with a good result as shown in figure 10, the algorithm fails totally since it gives just one cluster for all the dataset. DBSCAN merges between different clusters whereas it is impossible to merge between them, and can't recognize the shapes of clusters.

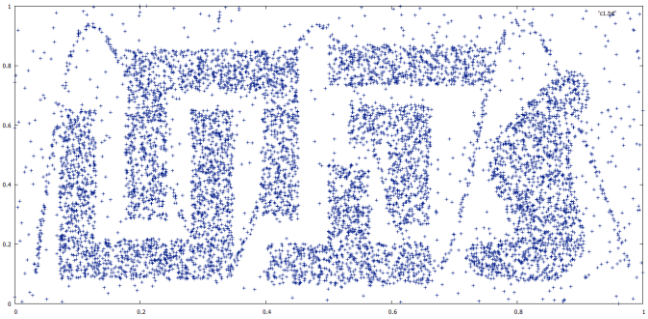

Figure 10. Chameleon dataset clustering result using DBSCAN algorithm

GMDBSCAN-UR is much better in dealing with nested shapes of clusters however, the overlapping between clusters still happens as shown in figure 11. The noise elimination process is not accurate, since the noise detection is not working very well.

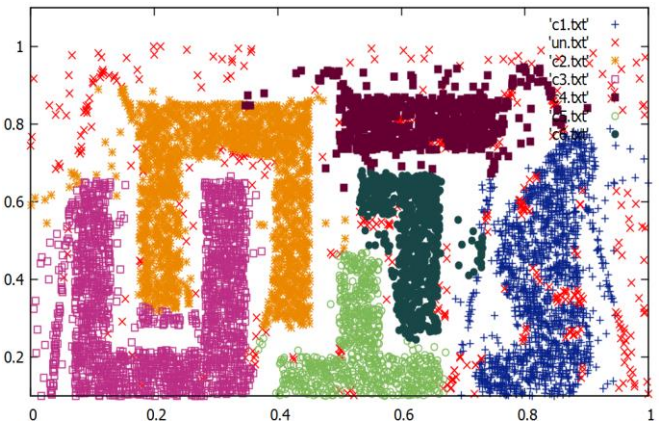

Figure 11. Chameleon dataset clustering result using GMDBSCAN-UR algorithm

For the same dataset using our proposed algorithm will give result much better, by solving the overlapping problem between clusters and reduce the noise, also, it catches the outlier points much better and more accurate shapes of clusters appears, as shown in figure 12.

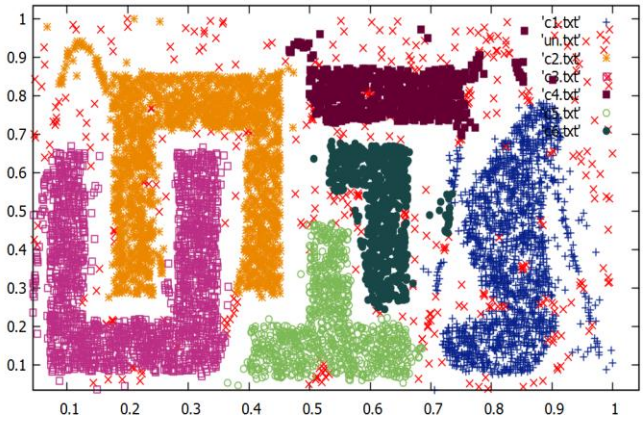

Figure12. Chameleon dataset clustering result using FNGMDBSCAN-UR algorithm

5 CONCLUSIONS AND FUTURE WORK

In this study we propose a new density-based and grid-based FNGMDBSCAN-UR algorithm based on fuzzy neighborhood function [9], and using representative points. The algorithm has been tested over many multi-density datasets. The results that we have showing how is the algorithm is accurate in detecting the noise and outliers as well as the arbitrary shape of the clusters in the datasets, moreover we made a comparison between our algorithm and DBSCAN and it shows that it is more robustness in multi-density datasets. Under the same conditions and datasets, we had another

comparison with GMDBSCAN-UR algorithm that uses a distance-base function to assign core points and neighborhood sets which is different that our algorithm that uses fuzzy neighborhood function to find fuzzy core points and the fuzzy neighborhood sets as well and the results
that we got from FNGMDBSCAN-UR algorithm show high accuracy.
Finally, in the field of data mining and specially clustering algorithms we can't say that, this algorithm is the best for all datasets, but we can say for each dataset and condition there is an algorithm that is much better than the other.

REFERENCES

[1] S.B. KOTSIANTIS, P. E. PINTELAS, "Recent Advances in Clustering: A Brief Survey", University of Patras Educational Software Development Laboratory, Hellas.

[2] Efendi N. Nazimova, GözdeUlutagay, "Robustness of density-based clustering methods with various neighborhood relations", Dokuz Eylul University, Fuzzy Setsand Systems 160 (2009) 3601–3615 4 July 2009

[3] Mohammed A. Alhanjouri, Rwand D. Ahmed, "New Density-Based Clustering Technique: GMDBSCAN-UR", Islamic university of Gaza, International Journal of Advanced Research in Computer Science, No. 1, Jan-Feb 2012

[4] Jonathon K. Parker, Lawrence O. Hall, and Abraham Kandel "Scalable Fuzzy Neighborhood DBSCAN"

[5] Hans-Peter Kriegel, Martin Pfeifle, "Density-Based Clustering of Uncertain Data", University of Munich, Germany.

[6] Efendi N. Nasibova, GözdeUlutagay, "A new unsupervised approach for fuzzy clustering", Dokuz Eylul University, Fuzzy Sets and Systems 158 (2007) 2118 – 2133.

[7] Jonathon Karl Parker, "Accelerated Fuzzy Clustering", University of South Florida.

[8] Abir Smiti and Zied Eloudi, "Soft DBSCAN: Improving DBSCAN Clustering method using fuzzy set theory", LARODEC, Universit´e de Tunis, Tunisia, Sopot, Poland, June 06-08, 2013.

[9] Jonathon K. Parker & Joni A. Downs, "Footprint generation using fuzzy-neighborhood clustering", 6 March 2012

[10] Sait Suer, Sinan Kockara, Mutlu Mete, "An improved border detection in dermoscopy images for density based clustering", Eighth Annual MCBIOS Conference. Computational Biology and Bioinformatics for a New Decade College Station, TX, USA. 1-2 April 2011

[11] David Vernet and Elisabet Golobardes, "An Unsupervised Learning Approach for Case-Based Classifier Systems", Enginyeria i Arquitectura La Salle, Universitat Ramon Llull, Pg. Bonanova 8, 08022 Barcelona, Spain

[12] Rui Xu, "Survey of Clustering Algorithms", IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 16, NO. 3, MAY 2005

[13] G. ULUTAGAY AND E. NASIBOV, " ON FUZZY NEIGHBORHOOD BASED CLUSTERING ALGORITHM WITH LOW COMPLEXITY", Iranian Journal of Fuzzy Systems Vol. 10, No. 3, (2013) pp. 1-20.

[14] Ester, M., H. P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, Portland, OR, AAAI Press, pp. 226-231. 1996

[15] C. Xiaoyun, M. Yufang, Z. Yan and W. Ping, "GMDBSCAN: Multi-Density DBSCAN Cluster Based on Grid," School of Information Science and Engineering, Lanzhou University Lanzhou 730000, PRC China.