A Dynamic Method for Discovering Density Varied Clusters

Mohammed T. H. Elbatta and Wesam M. Ashour

Faculty of Computer Engineer, Islamic University of Gaza
mohtb@hotmail.com, washour@iugaza.edu.ps

Abstract

Density-based spatial clustering of applications with noise (DBSCAN) is a base algorithm for density based clustering. It can find out the clusters of different shapes and sizes from a large amount of data, which is containing noise and outliers. However, it fails to handle the local density variation that exists within the cluster. Thus, a good clustering method should allow a significant density variation within the cluster because, if we go for homogeneous clustering, a large number of smaller unimportant clusters may be generated. In this paper an enhancement of DBSCAN algorithm is proposed, which detects the clusters of different shapes, sizes that differ in local density. We introduce new algorithm Dynamic Method DBSCAN (DMDBSCAN). It selects several values of the radius of a number of objects (Eps) for different densities according to a k-dist plot. For each value of Eps, DBSCAN algorithm is adopted in order to make sure that all the clusters with respect to the corresponding density are clustered. For the next process, the points that have been clustered are ignored, which avoids marking both denser areas and sparser ones as one cluster.

Experimental results are obtained from artificial data sets and UCI real data sets. The final results show that our algorithm get a good results with respect to the original DBSCAN and DVBSCAN algorithms.

Keywords: *Density Different Cluster, Variance Density, DBSCAN, K-dist*

1. Introduction

Unsupervised clustering techniques are an important data analysis task that tries to organize the data set into separated groups with respect to a distance or, equivalently, a similarity measure [1]. Clustering has been applied to many applications in pattern recognition [2], imaging processing [3], machine learning [4], and bioinformatics [5].

Clustering methods can be categorized into two main types: fuzzy clustering and hard clustering. In fuzzy clustering, data points can belong to more than one cluster with probabilities [6]. In hard clustering, data points are divided into distinct clusters, where each data point can belong to one and only one cluster. These data points can be grouped with many different techniques. Such as Partitioning, Hierarchical, Density based, Grid based, and Model based.

Partitioning algorithms minimize a given clustering criterion by iteratively relocating data points between clusters until a (locally) optimal partition is attained. The most popular partition-based clustering algorithms are the k-means [7] and the k-mediod [8]. The advantage of the partition-based algorithms is the use an iterative way to create the clusters, but the limitation is that the number of clusters has to be determined by user and only spherical shapes can be determined as clusters.

Hierarchical algorithms provide a hierarchical grouping of the objects. These algorithms can be divided into two approaches, the bottom-up or agglomerative and the top-down or divisive approach. In case of agglomerative approach, at the start of the algorithm, each object

represents a different cluster and at the end, all objects belong to the same cluster. In divisive method at the start of the algorithm all objects belong to the same cluster, which is split, until each object constitute a different cluster. Hierarchal algorithms create nested relation-ships of clusters, which can be represented as a tree structure called dendrogram [9]. The resulting clusters are determined by cutting the dendrogram by a certain level. Hierarchal algorithms use distance measurements between the objects and between the clusters. Many definitions can be used to measure distance between the objects, for example Euclidean, *City-block (Manhattan)*, Minkowski etc.

Between the clusters one can determine the distance as the distance of the two nearest objects in the two clusters (single linkage clustering) [10], or as the two furthest (complete linkage clustering) [11], or as the distance between the mediods of the clusters. The disadvantage of the hierarchical algorithm is that after an object is assigned to a given cluster it cannot be modified later. Also only spherical clusters can be obtained. The advantage of the hierarchical algorithms is that the validation indices (correlation, inconsistency measure), which can be defined on the clusters, can be used for determining the number of the clusters. The popular hierarchical clustering methods are CHAMELEON [12], BIRCH [13] and CURE [14].

Density-based algorithms like DBSCAN [15] and OPTICS [16] find the core objects at first and they are growing the clusters based on these cores and by searching for objects that are in a neighborhood within a radius epsilon of a given object. The advantage of these types of algorithms is that they can detect arbitrary form of clusters and it can filter out the noise.

Grid-based algorithms quantize the object space into a finite number of cells (hyper-rectangles) and then perform the required operations on the quantized space. The advantage of this approach is the fast processing time that is in general independent of the number of data objects. The popular Grid-based algorithms are STING [17], CLIQUE [18], and WaweCluster [19].

Model-based algorithms find good approximations of model parameters that best fit the data. They can be either partitional or hierarchical, depending on the structure or model they hypothesize about the data set and the way they refine this model to identify partitionings. They are closer to density-based algorithms, in that they grow particular clusters so that the preconceived model is improved. However, they sometimes start with a fixed number of clusters and they do not use the same concept of density. Most popular model-based clustering methods are EM [20].

Fuzzy algorithms suppose that no hard clusters exist on the set of objects, but one object can be assigned to more than one cluster. The best known fuzzy clustering algorithm is FCM (Fuzzy C-MEANS) [21].

Categorical data algorithms are specifically developed for data where Euclidean, or other numerical-oriented, distance measures cannot be applied.

Rest of the paper is organized as follows. Section 2 provides related work on density based clustering. Section 3 presents DBSCAN clustering algorithm. In Section 4 the proposed algorithm. In Section 5, simulation and results are presented and discussed. Finally, Section 6 presents conclusion and future works.

2. Related Work

The DBSCAN (Density Based Spatial Clustering of - Applications with Noise) [15] is a pioneer algorithm of density based clustering. It requires user predefined two input parameters, which are radius and minimum objects within that radius. The density of an object is the number of objects in its ε-neighborhood of that object. DBSCAN does not specify upper limit of a core object *i.e.*, how much objects may present in its –neighborhood.

So due to this, the output clusters are having wide variation in local density. So that**,** a large number of smaller unimportant clusters may be generated.

OPTICS [16] algorithm is an improvement of DBSCAN to deal with variance density clusters. OPTICS does not assign cluster memberships but this algorithm computes an ordering of the objects based on their reachability distance for representing the intrinsic hierarchical clustering structure. Pei, *et al.*, [22] proposed a nearest-neighbor cluster method, in which the threshold of density (equivalent to Eps of DBSCAN) is computed via the Expectation-Maximization (EM) [20] algorithm and the optimum value of k (equivalent to MinPts of DBSCAN) can be decided by the lifetime individual k. As a result, the clustered points and noise were separated according to the threshold of density and the optimum value of k.

In order to adapt DBSCAN to data consisting of multiple processes, an improvement should be made to find the difference in the m^{th} nearest distances of processes. Roy and Bhattacharyya [23] developed new DBSCAN algorithm, which may help to find different density clusters that overlap. However, the parameters in this method are still defined by users. Lin and Chang [24] introduced new approach called GADAC, which may produce more precise classification results than DBSCAN does. Nevertheless, in GADAC, the estimation of the radius is dependent upon the density threshold δ, which can only be determined in an interactive way.

Pascual, *et al.*, [25] developed density-based cluster method to deal with clusters of different sizes, shapes, and densities. However, the parameters neighborhood radius R, which is used to estimate the density of each point, have to be defined using prior knowledge and finding Gaussian-shaped clusters and is not always suit for clusters with arbitrary shapes.

Another enhancement of the DBSCAN algorithm is DENCLUE [25], based on an influence function that describes the impact of an object upon its neighborhood. The result of density function gives the local density maxima value and this local density value is used to form the clusters. It produces good clustering results even when a large amount of noise is present.

EDBSCAN (An Enhanced Density Based Spatial Clustering of Application with Noise) [26] algorithm is another extension of DBSCAN; it keeps tracks of density variation which exists within the cluster. It calculates the density variance of a core object with respect to its ε-neighborhood. If density variance of a core object is less than or equal to a threshold value and also satisfying the homogeneity index with respect to its neighborhood then it will allow the core object for expansion. But it calculates the density variance and homogeneity index locally in the ε-neighborhood of a core object.

DD_DBSCAN [27] algorithm is another enhancement of DBSCAN, which finds the clusters of different shapes, sizes which differ in local density. but, the algorithm is unable to handle the density variation within the cluster. DDSC [28] (A Density Differentiated Spatial Clustering Technique) is proposed, which is again an extension of the DBSCAN algorithm. It detects clusters, which are having non-overlapped spatial regions with reasonable homogeneous density variations within them.

In contrast to DBSCAN, DVBSCAN [29] algorithm handles local density variation within the cluster. The input parameters used in this algorithm are minimum objects (μ), radius, threshold values (α, λ). It calculates the growing cluster density mean and then the cluster density variance for any core object, which is supposed to be expanded further by considering density of its E-neighborhood with respect to cluster density mean. If cluster density variance for a core object is less than or equal to a threshold value and is also satisfying the cluster similarity index, then it will allow the core object for expansion.

CHAMELEON [12] finds the clusters in a data set by two-phase algorithm. In first phase, it generates a k-nearest neighbor graph. In the second phase, it uses an agglomerative hierarchical clustering algorithm to find the cluster by combining the sub clusters.

Most of the algorithms are not robust to noise and outlier, Density based algorithms are more important in this case. However, most of the density based clustering algorithms, are not able to handle the local density variations. DBSCAN [15] is one of the most popular algorithms due to its high quality of noiseless output clusters. However, it fails to detect the density-varied clusters, and there are many researches exist as an enhancement of DBSCAN for handling the density variation within the cluster.

3. DBSCAN Algorithm

The DBSCAN [30] is density fundamental cluster formation. Its advantage is that it can discover clusters with arbitrary shapes and size. The algorithm typically regards clusters as dense regions of objects in the data space that are separated by regions of low-density objects. The algorithm has two input parameters, radius ε and MinPts. For understanding the process of the algorithm some concepts and definitions has to be introduced. The definition of dense objects is as follows.

Definition 1. The neighborhood within a radius ε of a given object is called the ε -neighborhood of the object.

Definition 2. If the ε -neighborhood of an object contains at least a minimum number σ of objects, then the object is called an σ -core object.

Definition 3. Given a set of data objects, D, we say that an object p is directly density-reachable from object q if p is within the ε -neighborhood of q and q is a σ -core object.

Definition 4. An object p is density-reachable from object q with respect to ε and σ in a given set of data objects, D, if there is a chain of objects $p_1, p_2, p_3, \ldots, p_n, p_1 = q$ and $p_n = p$ such that p_{n+1} is directly density-reachable from p_i with respect to ε and σ, for $1 \leq i \leq$n, $p_i \in D$.

Definition 5. An object p is density-connected from object q with respect to ε and σ in a given set of data objects, D, if there is an object $o \in D$ such that both p and q are density-reachable from o with respect to ε and σ.

According to the above definitions, it only needs to find out all the maximal density-connected spaces to cluster the data objects in an attribute space. And these density-connected spaces are the clusters. Every object not contained in any clusters is considered noise and can be ignored.

Explanation of DBSCAN Steps

DBSCAN [31] requires two parameters: radius epsilon (Eps) and minimum points (MinPts). It starts with an arbitrary starting point that has not been visited. It then finds all the neighbor points within distance Eps of the starting point.

If the number of neighbors is greater than or equal to MinPts, a cluster is formed. The starting point and its neighbors are added to this cluster and the starting point is marked as visited. The algorithm then repeats the evaluation process for all the neighbors' recursively.

If the number of neighbors is less than MinPts, the point is marked as noise.

If a cluster is fully expanded (all points within reach are visited) then the algorithm proceeds to iterate through the remaining unvisited points in the dataset.

4. The Proposed Algorithm DMDBSCAN

One of the main problems of DBSCAN is that it has wide density variation within a cluster. To overcome this problem, new algorithm DMDBSCAN based on DBSCAN algorithm is proposed in this section. We will present new method to solve the problem of using one global value of parameter Eps for all densities in the data set, instead DMDBSCAN will use dynamic method to find suitable value of Eps for each density level of the data set.

One of data mining primary method is clustering analysis. Clustering analysis has many methods such as density clustering. This method has advantages as:

1. Its clusters are easy to understand.
2. It does not limit itself to shapes of clusters.

But existing density-based algorithms have trouble in finding out all the meaningful clusters for data sets with varied densities. In this section we will introduce a new algorithm called DMDBSCAN for the purpose of varied-density data sets analysis. The basic idea of DMDBSCAN is that we need some methods to find suitable values of parameter Eps for different levels of densities according to k-dist plot, then we can use traditional DBSCAN algorithm to find clusters. For each value of Eps, DBSCAN algorithm is adopted to find all the clusters with respect to the corresponding density level. Then, in the next step, all points which clustered are ignored. The final result will avoids marking both denser areas and sparser ones as one cluster.

To determine the parameters Eps and MinPts we need to look at the behavior of the distance from point to its kth nearest neighbor, which is called k-dist. This k-dists are computed for all data points for some (k), then the plot sorted values in ascending order, after that, we expect to see the sharp change in the plotted graph. This sharp change at the value of k-dist corresponds with a suitable value of Eps for each density level of data set. For example the Line (A) in Figure 1 shows a simple k-dist line for the value of k = 3. We notice that the value of Eps determined in this way depends on (k), but doesn't change dramatically as (k) changes.

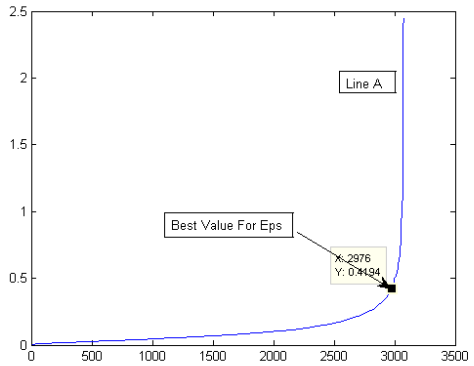


Figure 1. Points Sorted By Distance to the 3rd Nearest Neighbor

The strength point of DBSCAN it can find many clusters which could not be found using other clustering algorithms, like k-means, because DBSCAN uses a density-based definition of a cluster, which result in less relatively resistant to noise and can handle clusters of different shapes and sizes. However, the main weakness of DBSCAN is that it has trouble when the clusters have greatly varied densities.

In order to more description of DMDBSCAN, 2-dimension data is chosen. Figure 2 shows the data points. Obviously, there are two regions with respect to different densities levels in the data set. And data points of each region are uniformly distributed. The data set provides a clustering standard to estimate the accuracy of the result, for it has strong regularity and obvious clusters. In addition, as it has been already acknowledged that density-based clustering algorithms can find out clusters with any shape.

Suppose that the noise around the denser cluster C1 has the same density as the other cluster C2. If the Eps threshold is low enough that DBSCAN finds C2 as cluster, then C1 and the points surrounding it will become a single cluster. If the Eps threshold is high enough that DBSCAN finds C1 as a separate cluster, and the points surrounding are marked as noise, then C2 and the points surrounding it will also be marked as noise. DBSCAN also has trouble with high-dimensional data because density is more difficult to define for such data.

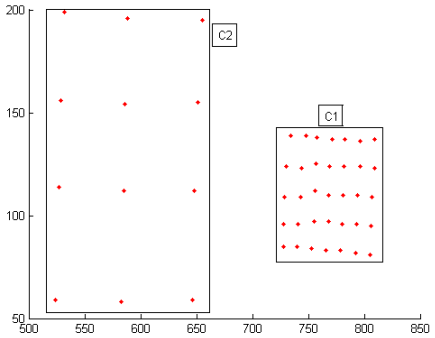


Figure 2. Two Regions with Respect to Different Densities

4.1. Description of Finding Suitable Epsi For Each Density Level

Formally, algorithm can describe our proposed to find suitable Epsi for each density level of data set as follow:

1. Calculates and stores k-dist for each project and partition k-dist plots;
2. The number of densities is given intuitively by k-dist plot;
3. Choose parameters Epsi automatically for each density.

In the first step, K-dist plot is drawn for not only selection of parameters Eps, but also analysis of density levels of the data set. If we have data sets with widely varied density, we notice that there will be some variation, depends on the density of the cluster and the random distribution of points, but the points of the same density level, the range of the variation will not be huge while a sharp changes that expected to see between two density levels. Thus there will be several smooth curves connected by greatly variation ones. For a data set of single-density, if its density does not vary widely, there is only one smooth curve in its k-dist plot.

Figure 1 shows a simple k-dist plot. Line A shows a simple k-dist line of a single-density data set. Figure 3 shows a simple line of a three varied-densities data set. We notice that there are sharp changes in the curves which correspond to noise points connecting two smooth curves which stand for two density levels, as Line b and d which can be called level-turning lines. Line b connects line a and c, and line d connects c and e, while a, c and e stand for different density levels. The outliers are shown with line f are not a level-turning line for it does not connect two smooth lines.

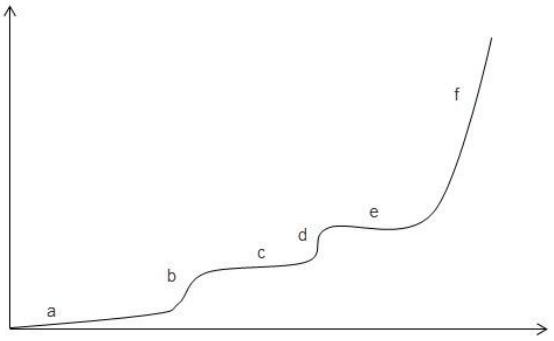


Figure 3. Three Density Levels Data Set

In Figure 3 we have three density levels, the result of that are three suitable values of Eps. Combine line a and b as a sub-k-dist plot to select Eps1, and then take line c and d as a sub-k-dist plot for Eps2, e and f for Eps3.

4.2. DMDBSCAN Algorithm Pseudo-Code

The proposed method of the algorithm to find suitable Epsi for each level of density is shown as pseudo code in Algorithm 1.

Algorithm 1. The pseudo code of the proposed technique DMDBSCAN to find suitable Epsi for each level of density in data set.	
Purpose:	1. *To find suitable values of Eps*
Input:	2. *Data set of size n*
Output:	3. *Eps for each varied density*
Procedure:	4. $For\ i = 1\ to\ n$ 5. $For\ j = 1\ to\ n$ 6. $d(i,j) \leftarrow find\ distance(x_i, x_j)$ 7. $find\ minimum\ values\ of\ distances\ to\ nearest\ 3$ 9. $end\ for$ 10. $end\ for$ 11. $sort\ distances\ ascending\ and\ plot\ to\ find\ each\ value$ 12. $Eps\ corresponds\ to\ critical\ change\ in\ curves$

5. Simulation and Results

We evaluated our proposed algorithm on several artificial and real data sets.

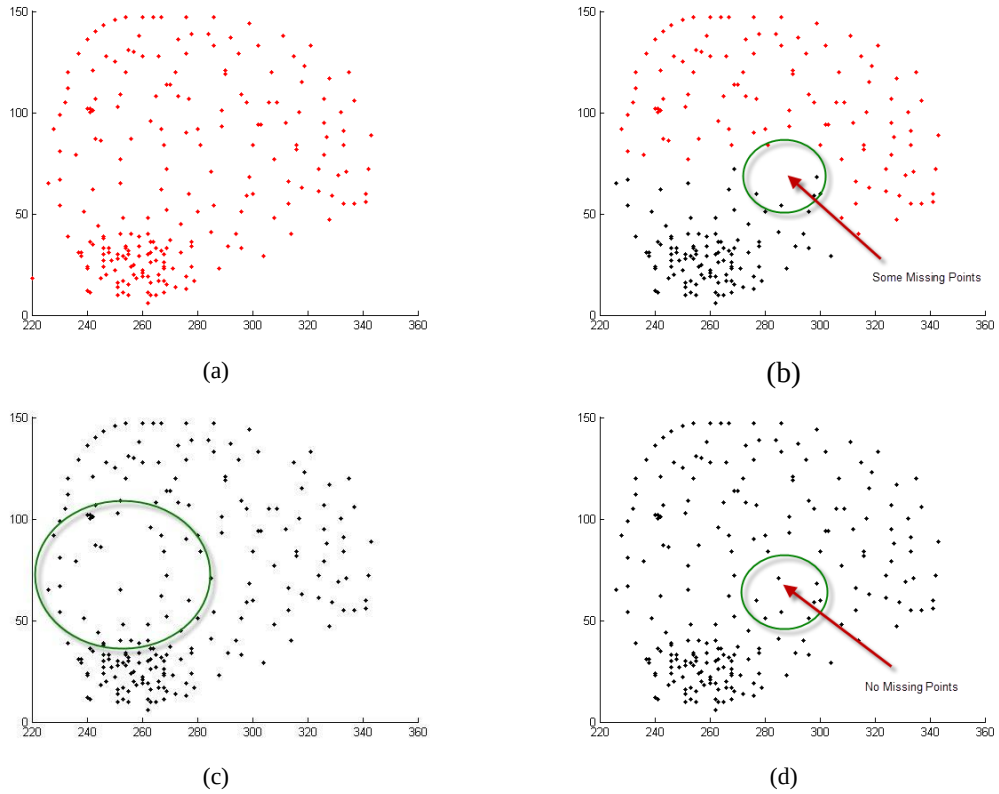
5.1. Artificial Data Sets

We use three artificial two-dimensional data sets, since the results are easily visualized. The first data set is shown in Figure 4 which consists of 226 data points with one cluster.

Figure 4(a) shows the original dataset plotting. In Figure 4(b), after applying the DBSCAN algorithm, with MinPts = 5, Eps = 11.8, we get 2-clusters. In Figure 4(c), after applying the

DVBSCAN algorithm, with $\alpha = 100$, $\lambda = 50$, $\mu = 20$, Eps = 12, we get 1-clusters, but there are some missing points. In Figure 4(d), after applying our proposed algorithm with Eps equal 10.77, 12.17 and 18.25 respectively, we get the correct number of clusters, that is, we have only 1-cluster. And we note that the points that deleted by DBSCAN or DVBSCAN, as DBSCAN and DVBSCAN considered them as noise points, now they are appeared after applying our proposed algorithm.

Figure 5(a) shows the original dataset plotting. Figure 5(b) shows the result of applying DBSCAN on the second dataset, with MinPts = 5, and Eps = 0.2. The rsulted clusters are 3-clusters. In Figure 5(c), after applying the DVBSCAN algorithm, with $\alpha = 100$, $\lambda = 50$, $\mu = 20$, Eps = 0.25, we get 4-clusters. But if we applied our proposed algorithm Figure 5(d) with Eps equal 0.1007, 0.1208 and 0.171 respectively, we get the correct number of clusters, which are 2-clusters.



(a) (b) (c) (d)

Figure 4. (a) 208 data points with one cluster. (b) DBSCAN applied Eps = 11.8, MinPts = 5. (c) DVBSCAN applied for the values, $\alpha = 100$, $\lambda = 50$, $\mu = 20$, Eps = 12. (d) DMDBSCAN algorithm with Eps = 10.77, 12.17 and 18.25 Respectively

Figure 6(a) shows the original dataset plotting. In Figure 6(b), after applying the DBSCAN algorithm, with MinPts = 5, Eps = 8, we get 4-clusters. In this dataset, DBSCAN treats some points as noise and remove them. In Figure 6(c), after applying the DVBSCAN algorithm, with $\alpha = 100$, $\lambda = 50$, $\mu = 20$, Eps = 8.5, we get 4-clusters. In Figure 6(d), after applying our proposed algorithm with Eps equal 8.062, 13.15 and 18.03 respectively, we get the correct number of clusters, that is, we have only 5-clusters.

5.2. Real Data Sets

We use the iris data set from the UCI (http://archive.ics.uci.edu/ml/datasets/Iris) which contains three clusters, 150 data points with 4 dimensions. For measuring the accuracy of our proposed algorithm, we use an average error index in which we count the misclassified samples and divide it by the total number of samples. We apply the DBSCAN algorithm with Eps = 0.35 and MinPts = 5, and obtain an average error index of 45.33%. After applying DVBSCAN algorithm on this data set with $\alpha = 100$, $\lambda = 50$, $\mu = 20$, Eps = 0.4, we obtain an average error index of 17.22%. While when applying the DMDBSCAN algorithm, we have an average error index of 15.00%.

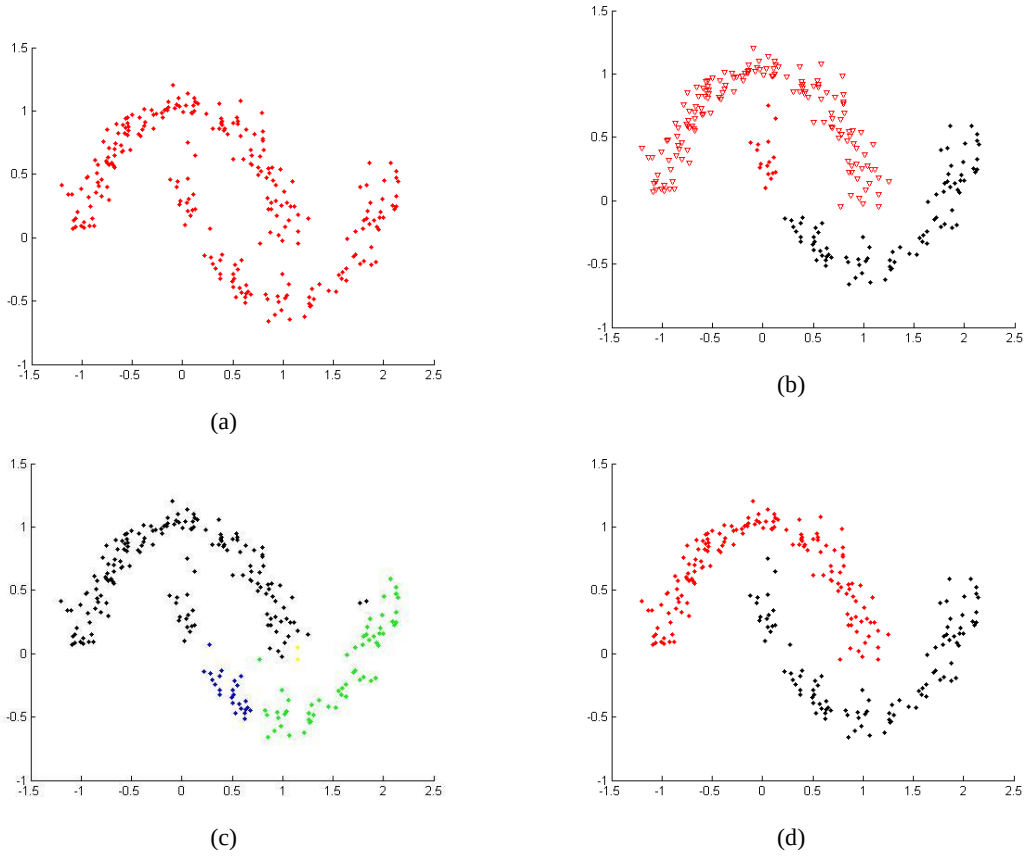


Figure 5. (a) 256 data points with tow cluster. (b) DBSCAN applied Eps = 0.2, MinPts = 5. (c) DVBSCAN applied for the values, $\alpha = 100$, $\lambda = 50$, $\mu = 20$, Eps = 0.25. (d) DMDBSCAN algorithm with Eps equal 0.1007, 0.1208 and 0.171 Respectively

We apply another data set, which is Haberman data set from UCI (http://archive.ics.uci.edu/ml/datasets/Haberman's+Survival) to show the efficiency of our proposed algorithm. The Haberman data set contains tow clusters, 306 data points with 3 dimensions. The obtained results are shown in Table 1. We get an average error index of 33.33% when we apply DBSCAN algorithm with Eps = 4.3 and MinPts = 5. After applying DVBSCAN on Haberman data set, we get an average error index of 32.65% with $\alpha = 100$, $\lambda = 50$, $\mu = 20$, Eps = 4.5. While when applying the DMDBSCAN algorithm, we have an average

error index of 20.33%. We apply another data set, which is Glass data set from UCI (http://archive.ics.uci.edu/ml/ datasets/Glass+Identification). The Glass data set contains six clusters, 214 data points with 9 dimensions. The obtained results are shown in Table 1. We get an average error index of 66.82% when we apply DBSCAN algorithm with Eps = 0.85 and MinPts = 5. After applying DVBSCAN we get an average error index of 41.23% with α = 100, λ = 50, μ = 20, Eps = 0.9. While when applying the DMDBSCAN algorithm, we have an average error index of 50.34%. We notice in this dataset that the error rate resulted by using DBSCAN, DVBSCAN and DMDBSCAN is large. This is due to the fact that as the number of dimensions increase, the clustering algorithms fail to find the correct number of clusters.

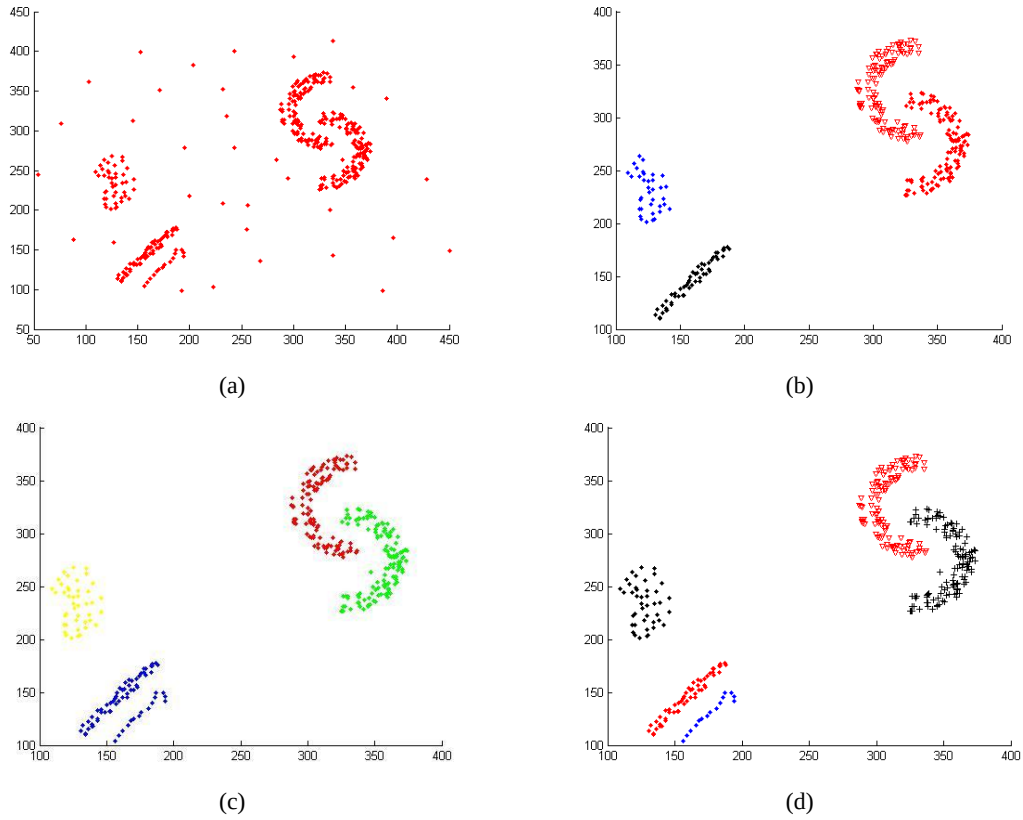


Figure 6. (a) 5743 data points with five clusters. (b) DBSCAN applied Eps = 8, MinPts = 5. (c) DVBSCAN applied for the values, α = 100, λ = 50, μ = 20, Eps = 8.5. (d) DMDBSCAN algorithm with Eps equal 8.062, 13.15 and 18.03 Respectively

Table 1. Comparison Of Average Error Index Between The Results Of DBSCAN, DVBSCAN And Our Proposed Algorithm DMDBSCAN On Real Data Sets

Dataset	True Clusters	Determined Clusters DBSCAN	Determined Clusters DVBSCAN	Determined Clusters DMDBSCAN	DBSCAN Error %	DVBSCAN Error %	DMDBSCAN Error %
IRIS	3	2	3	3	45.33	17.22	15.00
Haberman	2	1	2	2	33.33	32.65	20.33
Glass	6	3	5	5	66.82	41.23	50.34

6. Conclusions

We have proposed an enhancement algorithm based on DBSCAN to cope the problems of one of the most used clustering algorithm. Our proposed algorithm DMDBSCAN gives far more stable estimates of the number of clusters than existing DBSCAN or DVBSCAN over many different types of data of different shapes and sizes.Several opportunities for future research, how to select all the parameters automatically is one of the interesting challenges as parameter k has to be chosen subjectively in DMDBSCAN algorithm. The future work can be focused on reducing the time complexity of DMDBSCAN algorithm.

References

[1] A. K. Jain and R. C. Dubes, "Algorithm for Clustering Data", Printice Hall Englewood cliffs NJ, **(1998)**.
[2] B. BahmaniFirouzi, T. Niknam and M. Nayeripour, "A New Evolutionary Algorithm for Cluster Analysis", Proceeding of world Academy of Science, Engineering and Technology, vol. 36, (2008), December.
[3] M. Celebi, "Effecitive Initialization of K-means for Color Quantization", Proceeding of the IEEE International Conference on Image Processing, **(2009)**, pp. 1649-1652.
[4] M. Al- Zoubi, A. Hudaib, A. Huneiti and B. Hammo, "New Efficient Strategy to Accelerate k-Means Clustering Algorithm", American Journal of Applied Science, vol. 5, no. 9, **(2008)**, pp. 1247-1250.
[5] M. Borodovsky and J. McIninch, "Recognition of genes in DNA sequence with ambiguities", Biosystems, vol. 30, Issues 1-3, **(1993)**, pp. 161-171.
[6] J. Bezdek and N. Pal, "Fuzzy Models for Pattern Recognition", IEEE press, New York, NY, USA, **(1992)**.
[7] L. Kaufman and P. Rousseeuw, "Finding Groups in Data: An Introduction to Cluster Analysis", John Wiley & Sons, **(1990)**.
[8] L. Kaufman and Rousseeuw, "Clustering by means of medoids", StatisticalData Analysis based on the L1 Norm, Elsevier, **(1987)**, pp. 405–416.
[9] G. Gan, Ch. Ma and J. Wu, "Data Clustering: Theory, Algorithms, and Applications", ASA-SIAM series on Statistics and Applied Probability, SIAM, **(2007)**.
[10] D. Defays, "An Efficeint Algorithm for A Complete Link Method", The Computer Journal, vol. 20, **(1977)**, pp. 364-366.
[11] R. Sibson, "SLINK: an Optimally Efficent Algorithm for the Single Link Cluster Method", The Computer Journal, vol. 16, no. 1, **(1973)**, pp. 30-34.
[12] G. Karypis, E. H. Han and V. Kumar, "CHAMELEON: Ahierarchical clustering algorithm using dynamic modeling", Computer, vol. 32, no. 8, **(1999)**, pp. 68–75.
[13] T. Zhang, R. Ramakrishnan and M. Livny, "BIRCH: an efficient data clustering method for very large databases", SIGMOD '96 Proceedings of the 1996 ACM SIGMOD international conference on Management of data, ACM New York, NY, USA **(1996)**, pp. 103-114.
[14] S. Guha, R. Rastogi and K. Shim, "Cure: An efficient clustering algorithm for large databases", in SIGMOD 1998, Proceedings ACM SIGMOD International Conference on Management of Data, **(1998)** June 2-4, Seattle,Washington, USA (L. M. Haas and A. Tiwary, eds.), pp. 73-84, ACM Press.
[15] M. Ester, 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 (KDD'96), Portland: Oregon, **(1996)**, pp. 226-231.
[16] M. Ankerst, M. Breunig, H. P. Kriegel and J. Sander, "OPTICS: Ordering Objects to Identify the Clustering Structure", Proc. ACM SIGMOD, in International Conference on Management of Data, **(1999)**, pp. 49–60.
[17] W. Wang, J. Yang and R. Muntz, "Sting: A statistical information grid approach to spatial data mining", In Proceeding, VLDB '97 Proceedings of the 23rd International Conference on Very Large Data Bases, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, **(1997)**, pp. 186-195.
[18] R. Agrawal, J. Gehrke, D. Gunopulos and P. Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications", in Proceeding, SIGMOD '98 Proceedings of the 1998 ACM SIGMOD international conference on Management of data, ACM New York, NY, USA **(1998)**, pp. 94– 105.
[19] G. Sheikholeslami, S. Chatterjee and A. Zhang, "WaveCluster: A multi-resolution clustering approach for very large spatial databases", in Proc. 24th Int. Conf. Very Large Data Bases, VLDB, **(1998)**, pp. 428– 439.
[20] R. M. Neal and G. E. Hinton, "A new view of the EMalgorithm that justifies incremental, sparse and other variants", In M. I. Jordan, ed., Learning in Graphical Models", Kluwer Academic Publishers, **(1998)**, pp. 355–3681.
[21] J. C. Bezdeck, R. Ehrlich and W. Full, "Fcm: Fuzzy c-means algorithm", Computers and Geoscience, vol. 10,

no. 2-3, **(1984)**, pp. 191-203.
[22] T. Pei, A. X. Zhu, C. H. Zhou, B. L. Li and C. Z. A. Qin, "new approach to the nearest-neighbour method to discover cluster features in overlaid spatial point processes", Int. J GeogrInfSci., **(2006)**, pp. 153–168.
[23] S. Roy and D. K. Bhattacharyya, "An approach to find embedded clusters using density based techniques", Lect Notes Comput., vol. 3816, **(2005)**, pp. 523–535.
[24] C. Y. Lin, C. C. Chang, "A new density-based scheme for clustering based on genetic algorithm", Fundam. Inform., vol. 68, **(2005)**, pp. 315–331.
[25] D. Pascual, F. Pla and J. S. Sanchez, "Non parametric local density-based clustering for multimoda overlapping distributions", In: Proceedings of intelligent data engineering and automated learning, (IDEAL2006), Spain, Burgos, **(2006)**, pp. 671–678.
[26] A. Ram, A. Sharma, A. S. Jalal, A. Agrawal and R. Singh, "An Enhanced Density Based Spatial Clustering of Applications with Noise", Advance Computing Conference, IACC 2009, IEEE International, **(2009)** March 6-7, pp. 1475-1478.
[27] B. Borach and D. K. Bhattacharya, "A ClusteringTechnique using Density Difference", In proceedings of International Conference on Signal Processing, Communications and Networking, (2007), pp. 585–588.
[28] B. Borah and D. K. Bhattacharyya, "DDSC, "A Density Differentiated Spatial Clustering Technique", Journal Of Computers, vol. 3, no. 2, **(2008)** February.
[29] A. Ram, S. Jalal, A. S. Jalal and M. Kumar, "DVBSCAN: A Density based Algorithm for Discovering Density Varied Clusters in Large Spatial Databases", International Journal of Computer Applications (0975 – 8887), vol. 3, no. 6, **(2010)** June.
[30] D. Hsu and S. Johnson, "A Vibrating Method Based Cluster Reducing Strategy", Fifth International Conference on Fuzzy Systems and Knowledge Discovery, vol. 2, **(2008)**, pp. 376-379.
[31] J. H. Peter and A. Antonysamy, "Heterogeneous Density Based Spatial Clustering of Application with Noise", International Journal of Computer Science and Network Security, vol. 10, no. 8, **(2010)**, pp. 210-214.

Authors

Mohammed T. H. Elbatta is with the Department of Computer Engineering, The Islamic University of Gaza, and IUG. He received his master degree in computer engineering from Islamic University of Gaza in 2012. His research interests include data mining in large databases, data warehousing. E-mail: mohtb@hotmail.com.

Wesam M. Ashour is with the Department of Computer Engineering, The Islamic University of Gaza, and IUG. He has graduated in 2000 with B.Sc. in Electrical and Computer Engineering from Islamic University of Gaza. He has worked at IUG for 3 years as a teaching assistant before getting a studentship and traveling to UK for M.Sc. Dr. Ashour has finished his M.Sc. in Multimedia with Distinction in 2004 from the University of Birmingham, UK. During his M.Sc. study, he was one of the top two students in the class and he was awarded a prize for the best project 2003/2004. Dr. Ashour is a researcher in Applied Computational Intelligence Research Unit in the University of the West of Scotland, UK since October, 2005. Dr. Ashour has been the head of the Computer Engineeing Department 2009-2010. Currently, Dr Ashour is visiting the UWS, UK, as an academic visitor for the period August 2012- August 2013. E-mail: washour@iugaza.edu.ps.