

Research Article

A Vibration Method for Discovering Density Varied Clusters

Mohammad T. Elbatta, Raed M. Bolbol, and Wesam M. Ashour

Department of Computer Engineering, Islamic University of Gaza, Palestine

Correspondence should be addressed to Wesam M. Ashour, washour@iugaza.edu.ps

Received 4 August 2011; Accepted 28 August 2011

Academic Editors: Z. He and J. A. Hernandez

Copyright © 2012 Mohammad T. Elbatta et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

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 is fail 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 and sizes that differ in local density. Our proposed method VMDBSCAN first finds out the “core” of each cluster—clusters generated after applying DBSCAN. Then, it “vibrates” points toward the cluster that has the maximum influence on these points. Therefore, our proposed method can find the correct number of clusters.

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 of 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 constitutes a different cluster. Hierarchical algorithms create nested relationships 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. Hierarchical 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 and so on.

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 and 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 they 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 WaveCluster [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 is presented. In Section 4, the proposed algorithm. In Section 5, simulation and results are presented and discussed. Finally, Section 6 presents conclusion and future work.

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, that is, 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 struc-

ture. 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 Bhat-tacharyya [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 et al. [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 of 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 suitable 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 satisfies 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 and 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 nonoverlapped spatial regions with reasonable homogeneous density variations within them.

In VDBSCAN [29] (Varied Density-Based Spatial Clustering of Applications with Noise), the authors have also tried to improve the results using DBSCAN algorithm. The method computes k -distance for each object and sort them in ascending order, then plotted using the sorted values. The sharp change at the value of k -distance corresponds to a suitable value.

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 variation. DBSCAN [15] is one of the most popular algorithms due to its high quality of noiseless output clusters. However, also failing to detect the density varied clusters, there are many researches existing 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 sizes. The algorithm typically regards clusters as dense regions of objects in the data spaces 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 have 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 of σ 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 $p_1, p_2, p_3, \dots, p_n, p_1 = q$ 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

- (i) DBSCAN [31] requires two parameters: radius epsilon (Eps) and minimum points (MinPts). It starts with an arbitrary starting point that has not been vis-

ited. It then finds all the neighbor points within distance Eps of the starting point.

- (ii) 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.
- (iii) If the number of neighbors is less than MinPts, the point is marked as noise.
- (iv) 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 data-set.

4. The Proposed Algorithm

One of the problems with DBSCAN is that it has wide density variation within a cluster.

To overcome this problem, new algorithm VMDBSCAN based on DBSCAN algorithm is proposed in this section. It first clusters the data objects using DBSCAN. Then, it finds the density functions for all data objects within each cluster. The data object that has the minimum density function will be the core for that cluster. After that, it computes the density variation of a data object with respect to the density of core object of its cluster against all densities of other core's clusters. According to the density variance, we do the movement for data objects toward the new core. New core is one of other core's clusters, which has the maximum influence on the tested data object.

We intuitively present some definitions.

Definition 6. Suppose that x and y two data objects in a d -dimensional feature space, D . The influence function of data object y on x is a function $f_B^y: D \rightarrow R_0^+$ and can be defined as basic influence function f_B :

$$f_B^y(x) = f_B(x, y). \quad (1)$$

The influence function we will choose will be function that can determine the distance between two data objects, as Euclidean distance function.

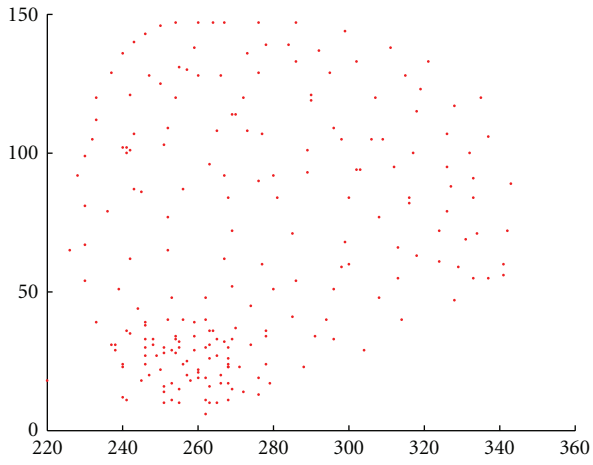
$$f_{\text{Euclidean}}(x, y) = \sqrt{(x - y)^2}. \quad (2)$$

Definition 7. Given a d -Dimensional feature space, D , the density function at a data object $x \in D$ is defined as the sum of all the influence to x from the rest of data objects in D .

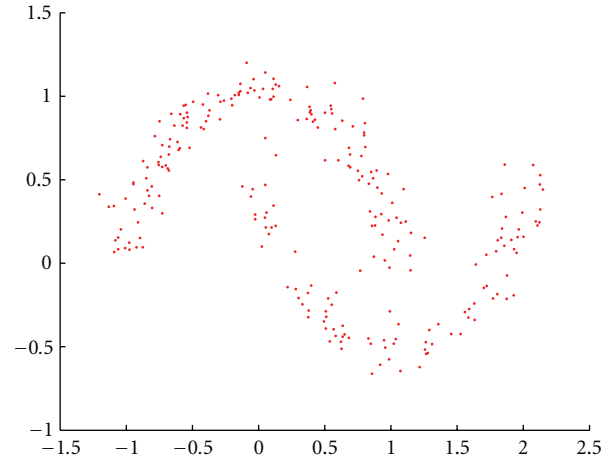
$$f_B^D(x) = \sum_{i=1}^n f_B^{x_i}(x), \quad 1 \leq i \leq n. \quad (3)$$

According to Definitions 6 and 7, we can calculate the density function for each data point in the space.

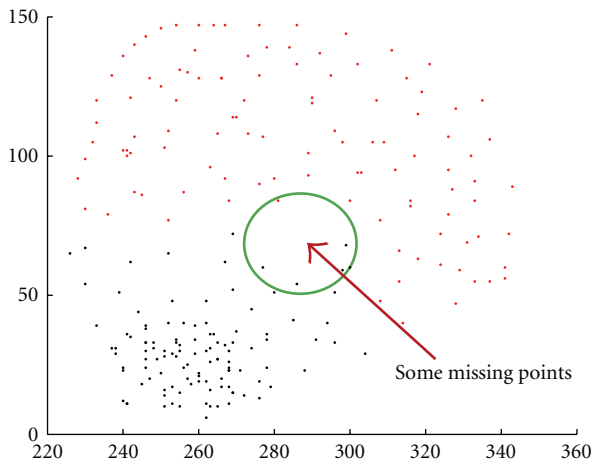
Definition 8. Core, the core object for each cluster, is the object that has the minimum density function value according to Definition 7. That is, we can calculate the density



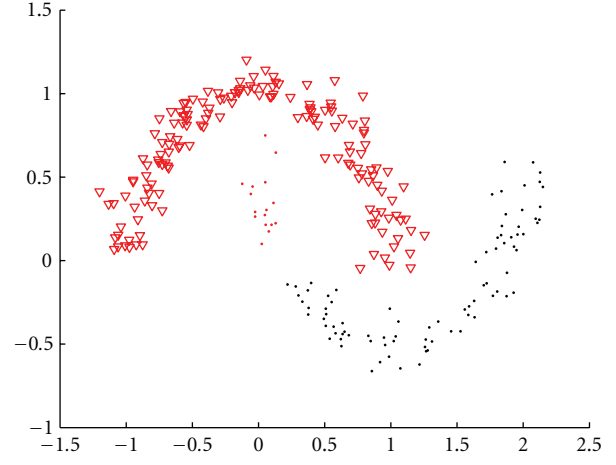
(a)



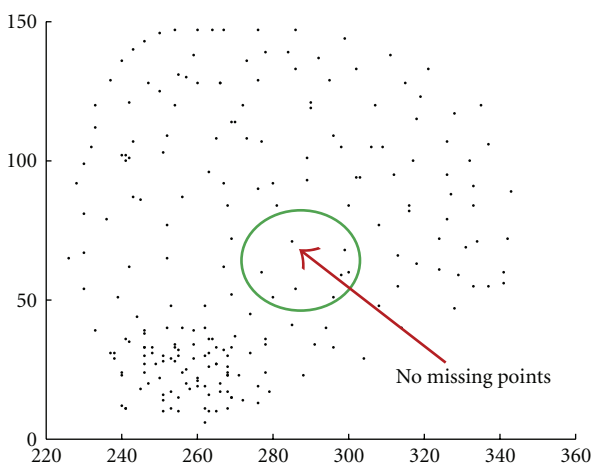
(a)



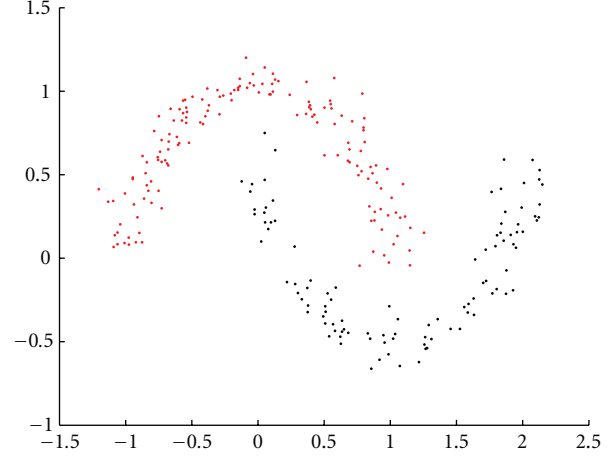
(b)



(b)



(c)



(c)

FIGURE 1: (a) 208 data points with one cluster. (b) DBSCAN applied $Eps = 11.8$, $MinPts = 5$. (c) VMDBSCAN algorithm with $\eta = 0.0005$.

FIGURE 2: (a) 256 data points with two clusters. (b) DBSCAN applied $Eps = 0.2$, $MinPts = 5$. (c) VMDBSCAN algorithm with $\eta = 0.005$.

function for each object in the cluster, which is given initially by DBSCAN, and the object which has the minimum connection to all other objects will be the core for that cluster.

Definition 9. Total Density Function E represents the difference among the data objects, which is based on the core. That is, the Total Density Function E for data object x_i

$$E_i = d(x_i, C_j), \quad 1 \leq i \leq n, \quad 1 \leq j \leq k, \quad (4)$$

i number of points, k number of cores

is the difference between the data object x_i and the core of its cluster.

In addition, according to our initial clusters which are given by the density-based clustering methods, we can take over the influence function (Definition 6) and density function (Definition 7) to calculate the Total Density Function E of the data objects by subtracting the value of their density function to the value of the core's:

$$E_i = \left| f_B^D(x_i) - f_B^D(C) \right|. \quad (5)$$

4.1. Vibration Process. Our main idea is the vibration of data objects according to the density of the data object with respect to core (Definition 8), the core that represents each cluster, and measure of the Total Density Function E of each data object as in (5). Then, if its Total Density Function E_i with respect to its core is greater than Total Density Function E_i for some other cores, vibrate all points in that cluster toward the core object which has the maximum influence on that object point, according to:

$$x(i+1) = x(i) + \eta * (x(c) - x(i)) * \left(e^{-1/2\sigma^2} \right), \quad (6)$$

where $\sigma = e^{-i/T}$, $x(i)$ is the current tested point, $x(c)$ is the current tested core, η is the learning rate, and T : is the control of reduction in sigma.

We use τ in the vibration equation to control the winner of the current cluster, and we can adapt it to get the best clustering result. T is used in our formula to control the reduction in sigma, that is, as the time increased, the movement (vibrate) of the point toward the new core is reduced.

Formally, we can describe our proposed algorithm as follows

- (1) Calculate the Density Function for all the data objects.
- (2) Do clustering for the data objects using traditional DBSCAN algorithm.
- (3) Calculate the Density Function for all the data objects again, and then find out the core of each generated cluster.
- (4) For each data object, if its Total Density Function with respect to its core is greater than with respect to other cores, then vibrate the data objects in that cluster.

```

VMDBSCAN()
(1) Begin initialize  $\eta$ 
(2)   For  $i = 1$  to  $n$ 
(3)      $d_i \leftarrow \text{density}(x_i)$ 
(4)   end_For
(5)   Class  $\leftarrow$  DBSCAN()
(6)   For  $j = 1$  to  $c$ 
(7)      $c_j \leftarrow \text{core}(x_j)$ 
(8)   end_For
(9)   For  $i = 1$  to  $n$ 
(10)     $E_i = c_i - \text{density}(x_i)$ 
(11)    For  $j = 1$  to  $c$ 
(12)       $E_i c = c_j - \text{density}(x_i)$ 
(13)      if  $E_i c < E_i$ 
(14)        vibrate the point
(15)      else no vibrate
(16)    end_If
(17)  end_For
(18) end_For

```

ALGORITHM 1: VMDBSCAN algorithm.

The proposed method of the algorithm is described as pseudo code in Algorithm 1.

The first step initializes the value of learning rate η it can take small values from $[0, 1]$; n is the number of data points in the D data set. For each data point in the data set, we compute the Density Function of this data point according to (3), and then store results in an array list of Point Density (d). Line 5 of the algorithm calls the DBSCAN algorithm to make initial clustering. From lines 6–8, we find the core object for each cluster resulting from DBSCAN. Line 10 calculates the Total Density Function E for each point x_i with respect to its core object. Line 12 calculates the Total Density Function E for that point x_i with respect to all other core objects. From line 13 to line 16 we check the effect of core objects on the data object x_i ; if the effect of its core object is less than other core objects c_j then vibrate the whole points which data object belongs to toward the core c_j .

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 1. which consists of 226 data points with one cluster.

Figure 1(a) shows the original dataset plotting. In Figure 1(b), after applying the DBSCAN algorithm, with $\text{MinPts} = 5$, $\text{Eps} = 11.8$, we get 2 clusters. In Figure 1(c), after applying our proposed algorithm with $\eta = 0.0005$, we get the correct number of clusters, that is, we have only 1 cluster. And we note that the points deleted by DBSCAN, as DBSCAN considered it then noise points, now appeared after applying our proposed algorithm.

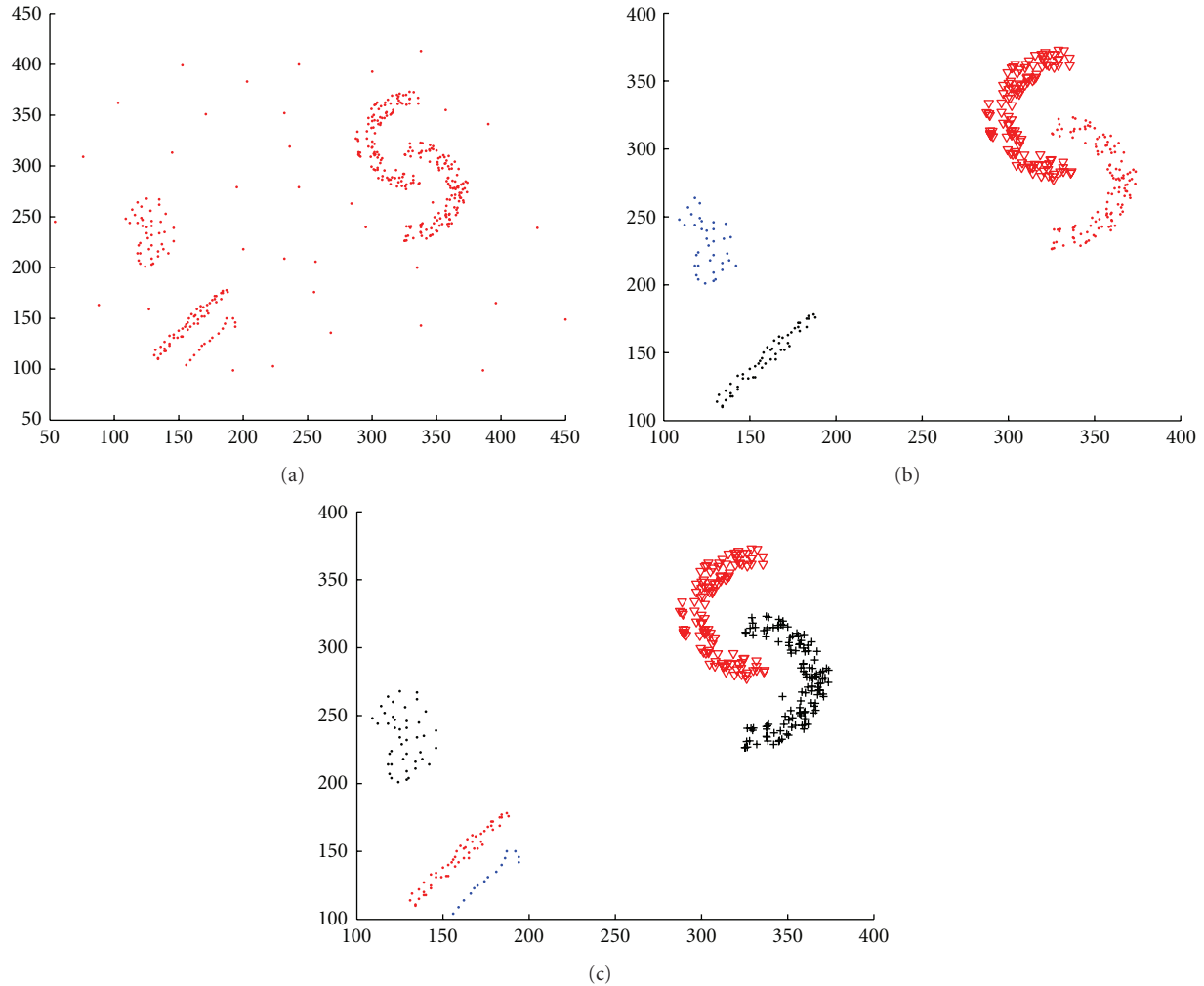


FIGURE 3: (a) 5743 data points with five clusters. (b) DBSCAN applied Eps = 8, MinPts = 5. (c) VMDBSCAN algorithm with $\eta = 0.0005$.

Figure 2(a) shows the original dataset plotting. Figure 2(b) shows the result of applying DBSCAN on the second dataset, with MinPts = 5, and Eps = 0.2. The resulting clusters are 3 clusters. But, if we applied our proposed algorithm Figure 2(c) with $\eta = 0.005$, we get the correct number of clusters, which are 2 clusters.

Figure 3(a) shows the original dataset plotting. In Figure 3(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 removes them. In Figure 3(c), after applying our proposed algorithm with $\eta = 0.0005$, 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%, while, when ap-

plying the VMDBSCAN algorithm with $\eta = 0.00005$, we have an average error index of 20.00%.

We apply another data set, which is Haberman data set from UCI ([http://archive.ics.uci.edu/ml/datasets/Haberman's +Survival](http://archive.ics.uci.edu/ml/datasets/Haberman%20Survival)) to show the efficiency of our proposed algorithm. The Haberman data set contains two 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, while, when applying the VMDBSCAN algorithm with $\eta = 0.0005$, we have an average error index of 27.78%.

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, While, when applying the VMDBSCAN algorithm with $\eta = 0.0005$, we have an average error index of 62.15%. We note in this dataset the error rate result by using DBSCAN or VMDBSCAN is large. This is due to the fact that as

TABLE 1: Comparison of average error index between the results of DBSCAN and our proposed VMDBSCAN on real data set.

Dataset	True clusters	Determined clusters DBSCAN	Determined clusters VMDBSCAN	DBSCAN error %	VMDBSCAN error %
IRIS	3	2	3	45.33	20.00
Haberman	2	1	2	33.33	27.78
Glass	6	3	4	68.22	62.61

the number of dimensions increases, the clustering algorithms fail to find the correct number of clusters.

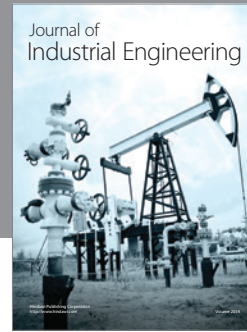
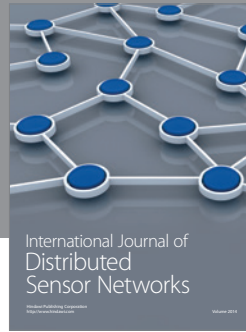
6. Conclusions and Future Works

We have proposed an enhancement algorithm based on DBSCAN to cope with the problems of one of the most used clustering algorithm. Our proposed algorithm VMDBSCAN gives far more stable estimates of the number of clusters than existing DBSCAN over many different types of data of different shapes and sizes. Future work will focus on determining the best value of the parameter η and improve the results for high dimensions data sets.

References

- [1] A. K. Jain and R. C. Dubes, *Algorithm for Clustering Data*, Prentice Hall, Englewood Cliffs, NJ, USA, 1998.
- [2] B. BahmaniFirouzi, T. Niknam, and M. Nayeripour, "A new evolutionary algorithm for cluster analysis," in *Proceedings of the World Academy of Science, Engineering and Technology*, vol. 36, December 2008.
- [3] M. E. Celebi, "Effective initialization of k-means for color quantization," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '09)*, pp. 1649–1652, Cairo, Egypt, November 2009.
- [4] M. B. Al-Zoubi, A. Hudaib, A. Huneiti, and B. Hammo, "New efficient strategy to accelerate k-means clustering algorithm," *American Journal of Applied Sciences*, vol. 5, no. 9, pp. 1247–1250, 2008.
- [5] M. Borodovsky and J. McIninch, "Recognition of genes in DNA sequence with ambiguities," *BioSystems*, vol. 30, no. 1–3, pp. 161–171, 1993.
- [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, New York, NY, USA, 1990.
- [8] L. Kaufman and P. J. Rousseeuw, *Clustering by Means of Medoids. Statistical Data Analysis Based on the L1 Norm*, Elsevier, 1987.
- [9] G. Gan, Ch. Ma, and J. Wu, *Data Clustering: Theory, Algorithms, and Applications*, ASA-SIAM Series on Statistics and Applied Probability, Society for Industrial and Applied Mathematics, 2007.
- [10] D. Defays, "An efficient algorithm for a complete link method," *The Computer Journal*, vol. 20, pp. 364–366, 1977.
- [11] R. Sibson, "SLINK: an optimally efficient algorithm for the single link cluster method," *The Computer Journal*, vol. 16, no. 1, pp. 30–34, 1973.
- [12] G. Karypis, E. H. Han, and V. Kumar, "Chameleon: hierarchical clustering using dynamic modeling," *Computer*, vol. 32, no. 8, pp. 68–75, 1999.
- [13] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: an efficient data clustering method for very large databases," *ACM Special Interest Group on Management of Data*, vol. 25, no. 2, pp. 103–114, 1996.
- [14] S. Guha, R. Rastogi, and K. Shim, "Cure: an efficient clustering algorithm for large databases," in *Proceedings of the ACM International Conference on Management of Data (SIGMOD '98)*, L. M. Haas and A. Tiwary, Eds., pp. 73–84, ACM Press, Seattle, sWash, USA, June 1998.
- [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)*, pp. 226–231, Portland, Ore, USA, 1996.
- [16] M. Ankerst, M. M. Breunig, H. P. Kriegel, and J. Sander, "OPTICS: ordering points to identify the clustering structure," *ACM Special Interest Group on Management of Data*, vol. 28, no. 2, pp. 49–60, 1999.
- [17] W. Wang, J. Yang, and R. Muntz, "Sting: A statistical information grid approach to spatial data mining," in *Proceedings of the 23rd International Conference on Very Large Data Bases*, 1997.
- [18] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications," *ACM Special Interest Group on Management of Data*, vol. 27, no. 2, pp. 94–105, 1998.
- [19] G. Sheikholeslami, S. Chatterjee, and A. Zhang, "WaveCluster: a multi-resolution clustering approach for very large spatial databases," in *Proceedings of the 24th International Conference on Very Large Data Bases (VLDB '98)*, pp. 428–439, 1998.
- [20] R. M. Neal and G. E. Hinton, "A new view of the EM algorithm that justifies incremental, sparse and other variants," in *Learning in Graphical Models*, M. I. Jordan, Ed., pp. 355–368, Kluwer Academic, Boston, Mass, USA, 1998.
- [21] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: the fuzzy c-means clustering algorithm," *Computers and Geosciences*, vol. 10, no. 2-3, pp. 191–203, 1984.
- [22] T. Pei, A. X. Zhu, C. Zhou, B. Li, and C. Qin, "A new approach to the nearest-neighbour method to discover cluster features in overlaid spatial point processes," *International Journal of Geographical Information Science*, vol. 20, no. 2, pp. 153–168, 2006.
- [23] S. Roy and D. K. Bhattacharyya, "An approach to find embedded clusters using density based techniques," *Lecture Notes in Computer Science*, vol. 3816, pp. 523–535, 2005.
- [24] C. Y. Lin, C. C. Chang, and C. C. Lin, "A new density-based scheme for clustering based on genetic algorithm," *Fundamenta Informaticae*, vol. 68, no. 4, pp. 315–331, 2005.
- [25] D. Pascual, F. Pla, and J. S. Sanchez, "Non parametric local density-based clustering for multimodal overlapping distributions," in *Proceedings of the Intelligent Data Engineering and Automated Learning (IDEAL '06)*, pp. 671–678, Burgos, Spain, 2006.
- [26] A. Ram, A. Sharma, A. S. Jalal, R. Singh, and A. Agrawal, "An enhanced density based spatial clustering of applications with

- noise,” in *Proceedings of the International Advance Computing Conference (IACC '09)*, pp. 1475–1478, March 2009.
- [27] B. Borach and D. K. Bhattacharya, “A clustering technique using density difference,” in *Proceedings of the International Conference on Signal Processing, Communications and Networking*, pp. 585–588, 2007.
- [28] B. Borah and D. K. Bhattacharyya, “DDSC: a density differentiated spatial clustering technique,” *Journal of Computers*, vol. 3, no. 2, pp. 72–79, 2008.
- [29] L. Peng, Z. Dong, and W. Najjun, “VDBSCAN: varied density based spatial clustering of applications with noise,” in *Proceedings of the International Conference on Service Systems and Service Management (ICSSSM '07)*, pp. 528–531, Chengdu, China, June 2007.
- [30] D. Hsu and S. Johnson, “A vibrating method based cluster reducing strategy,” in *Proceedings of the 5th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD '08)*, pp. 376–379, Shandong, China, October 2008.
- [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, pp. 210–214, 2010.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

