# DDBSCAN: Different Densities-Based Spatial Clustering of Applications with Noise

Mohammad F. Hassanin
Computer Science Dept.,
Faculty of Computers and Information,
Fayoum University, Egypt
mff00@fayoum.edu.eg

Mohamed Hassan
Information Systems Dept.,
Faculty of Computers and Information,
Fayoum University, Egypt
mhi11@fayoum.edu.eg

Abdalla Shoeb
Computer Science Dept.
Faculty of Computers and Information
Fayoum University, Egypt

***Abstract*— Recent advances in using computer with different fields of sciences produced huge amounts of data. These data represent as an analysis tool and key to overcome many problems. Clustering is a primary process to analyze the data as well as, it's a preprocessing step before other techniques like classification. Density-Based clustering algorithms have advantages like clustering any arbitrary shapes and defining number of clusters according to database. DBSCAN (Density Based Spatial Clustering of Application with Noise) [1] is the basic density-based algorithm. But it fails to discover different densities clusters, adjacent clusters and finally some noise points among different densities clusters. This paper addresses DBSCAN problems and tries to solve these problems by developing DDBSCAN. The basic idea is to compute the density of a cluster with respect to radius value Eps and minimum number of points MinPts. Then provide density threshold which is the responsible for joining a point to a certain cluster or not. Experiments show that DDBSCAN outperforms DBSCAN in different densities and adjacent clusters datasets.**

***Keywords: Density-based Clustering; DBSCAN; different density datasets***

## I. INTRODUCTION

Data mining has become a basic process before dealing with data and a preprocessing step in data mining techniques. Clustering tries to partition data to clusters based on similarity metrics while it maximizes inter-relations and minimizes intra-relations among dataset objects. Clustering Techniques lie among Partitioning, Hierarchical, Density based, Grid, and Model methods.

Density-based ones are very important to discover clusters of arbitrary shapes. DBSCAN is the main algorithm in density based techniques and it is efficient in discovering arbitrary shapes as well as it does not require predefined number of clusters as a parameter. It discovers clusters with respect to MinPts and Eps, but it still has problems when discovering multi-density clusters, adjacent clusters or noise points amongst adjacent clusters. Other traditional techniques such as OPTICS and DENCLUE have troubles to recover datasets with varying densities and adjacent clusters also. Many extensions to DBSCAN were developed to overcome above mentioned shortcomings. VDBSCAN is developed to solve the problem of varying densities but did not expose to adjacent clusters or adjacent clusters noise point's problems.

In this paper, a new improved density based algorithm is introduced. DDBSCAN (Different Densities-Based Spatial Clustering of Applications with Noise) tries to address all mentioned DBSCAN problems. The main idea is to define a density factor to the cluster and the object then defines a threshold parameter as decision criterion to determine whether joining this object or not. On this basis, any cluster will contain similar density nodes only.

Rest of paper is organized as follows; section 2 presents related works of DBSCAN. Section 3 has two sections, the first shows basic concepts of DBSCAN and the second presents DDBSCAN concepts. Section 4 shows the proposed algorithm. Section 5 discusses the results of testing DDBSCAN over different datasets. Finally, section 6 is regarded to conclusions.

## II. RELATED WORKS

Because DDBSCAN is an improvement in DBSCAN, we will introduce past studies related to density based algorithms.

DBSCAN is the pioneer in density-based algorithms. It requires two input parameters Eps and MinPts. It starts with point p and gets all Eps-neighbors with respect to. Eps. If p is core point, the cluster is formed. Repeat this procedure until visiting all density-reachable points of p. it visits another point in dataset and so on. It can identify arbitrary shapes and does not require a pre-defined number of clusters. But it does not behave well with different densities datasets.

OPTICS [2] is another density-based algorithm. It computes the ordering of points based on reachability distance so that it produces a structure of clusters not explicit clusters. This structure can be used to produce clusters, basic information about datasets. However, OPTICS has an issue related with explicit clusters of datasets and it needs another algorithm beside it to produce explicit clusters.

ST-DBSCAN [3] is an extension of DBSCAN to handle spatial-temporal datasets. It redefines border point to discover adjacent clusters and noise points among adjacent clusters. ST-DBSCAN does not handle varied densities well.

VDBSCAN [4] is developed to discover clusters with varied densities. It depends on generate several Eps parameters using k-dist plot. It proves a good manipulation of varied densities clusters but it has a trouble when discovering adjacent clusters and its noise points.

Incremental DBSCAN [5] algorithm is discovering clusters from dataset with incremental approach. If any object added later to dataset, it will add to existing clusters. It working normally like DBSCAN but if new points are added, it clusters and merges with existing clusters.

DBCLUM [6] algorithm is an extension of DBSCAN. It defines clustering in two main steps, Clustering and then merging. It clusters dataset individually with Eps and Minpts, then merges joined and similar clusters together according to given threshold. In contrast to DBSCAN, it eliminates concept of density-reachable. DBCLUM handles varied densities well, but it has problem with measuring density factor which it uses to merge clusters.

KDDClus [7] algorithm is another enhancement of DBSCAN. It used nearest neighbor algorithms to find Eps-neighborhood. It uses K D –tree data structure to compute Eps-neighborhood. Then store average KNN distances for different patterns. Then form clusters. It discovers clusters with multiple densities well. But using KD tree increases the time and processing the datasets. In addition to, it suffers when discovering adjacent clusters and amongst noise points.

Chameleon [8] discovers clusters of dataset by two-phase algorithm. Firstly, it generates k-nearest neighbor graph. Finally, it merges similar sub-clusters together.

As mentioned above, if algorithm solved problem like varying densities, it would fall into adjacent clusters and its noise points and vice versa. DDBSCAN is designated to overcome all these problems.

## III. BASIC CONCEPTS

*Definition 1(Eps-Neighborhood):* gets all neighbor nodes within range Eps. There are two types of points, core point and border point. Any point is called core point if its Eps-neighborhood exceeds MinPts. A point is called border point if its Eps-neighborhood less than MinPts.

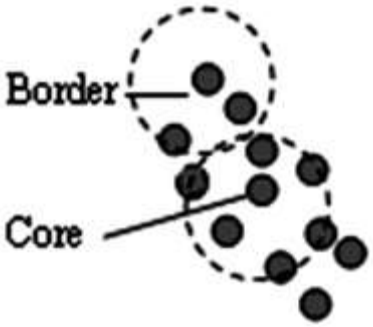


Fig. 1: Core points and border points.

*Definition 2(Density-reachable):* an object p is density-reachable with q, if there is a chain of objects x1, x2,…, xn connected between them such that p=x1 and q=xn

*Definition 3(Density-connected):* an object p is density-connected with q, if there is an objects x such that p is density-reachable from x and also q is density-reachable with x with respect to MinPts and Eps.

*Definition 4 (Cluster):* A cluster C is a subset of D has the following requirements

1. maximality : ∀ p, q: if p ε C and q is density-reachable from p, then q ε C.
2. Connectivity: ∀ p, q: if p ε C and q is density-connected from p, then q ε C.

*Definition 5 (Noise):* noise points refer to those points which are not joined to any cluster.

## IV. DDBSCAN CONCEPTS

The above definitions are defined by DBSCAN algorithms, but still have problems with different densities datasets as well as adjacent clusters. Concepts of DDBSCAN will be presented now

*Definition 5 (local eps):* is equal to Eps of algorithm.

*Definition 6 (global eps):* represents the real radius of cluster. Because of density-reachable points may join cluster, the real radius of cluster is changeable. Initially, global eps equals local eps. If an object will join the cluster, update global eps as shown in Figure 2.

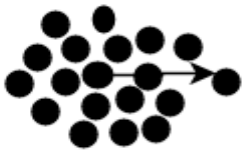

Fig. 2: global Eps.

*Definition 7 (local cluster):* A cluster C is a subset of D with connectivity condition.

∀ p, q: if p ε C and q is density-connected from p, then q ε C.

*Definition 8 (global cluster):* referred to in definition 4.

*Definition 9 (local density):* is the number of objects in local cluster divided by local eps.

*Definition 10(global density):* is the number of objects in global cluster divided by global eps.

*Definition 11 (MinMax):* MinMax(O1,O2) = $\frac{O1-O2}{\max(O1,O2)}$.

## V. DDBSCAN ALGORITHM

Initially, DDBSCAN accepts two parameters Eps and Threshold. Eps is used to get all objects within this region and Threshold is used as a decision criterion to determine whether an object will join the cluster or not. It starts with computing density for each object. This density is used to determine which object will be the first to expand. Then the best object which has the highest density value is retrieved to pass it to expand method.

```
DDBSCAN (SetOfObjects, Eps, Threshold)
// SetOfObjects is UNCLASSIFIED
  FOR i FROM 1 TO SetOfObjects.size DO
     Objects := SetOfObjects.regionQuery(currentO,Eps);
     currentO.density = Objects.size
  END FOR
  ClusterId := nextId(NOISE);
  FOR i FROM 1 TO SetOfObjects.size DO
     Object := SetOfObjects.getHighestDensity();
     IF Object.CluId = UNCLASSIFIED THEN
       IF ExpandCluster(SetOfObjects, Object, CluId,Eps)THEN
                CluId := nextId(CluId)
       END IF
     END IF
  END FOR
END; // DDBSCAN
```

First step in Expand process is to retrieve all objects that lie into Eps distance as stated in Definition 1. The object id marked NOISE (Definition 5) if and only if it is not has cluster id. DDBSCAN changed the criteria of joining or merging objects with clusters. To join object to cluster, the density must be less than threshold as stated. It defines maxEps as in figure 2 that holds the distance between current Point and farthest one in cluster as well as, local eps given parameter. Density now is ready to be computed. Divide seeds list size which is current point region by local density and divide result size which is cluster size by global density. Finally, use MinMax definition 13 to determine whether to merge current point or not.

```
ExpandCluster(SetOfObjects, Object, CluId, Eps) : Boolean;
  seedsList:=SetOfObjects.regionQuery(Object,Eps);
  IF seedsList.size== 0 THEN // no core Object
     SetOfObject.changeCluId(Object,NOISE);
  RETURN False;
  ELSE // all Objects in seedsList are density-reachable from Object
    SetOfObjects.changeCluIds(seedsList,CluId);
    seedsList.delete(Object);
    WHILE seedsList <> Empty DO
       currentP := seedsList.first();
                 maxEps = currentP.distance(Object);
                 IF maxEps > globalEps THEN
                        globalEps = maxEps;
        results := SetOfObjects.regionQuery(currentO,Eps);
       IF results.size >= 0 THEN
         FOR i FROM 1 TO results.size DO
          resultO := results.get(i);
          IF resultO.CluId
              IN {UNCLASSIFIED, NOISE} THEN
           IF resultO.CluId = UNCLASSIFIED THEN
               localDensity = results.size / Eps;
               globalDensity = seedsList.size / globalEps;
               seedsList.append(resultO);
           END IF;
           SetOfObjects.changeCluId(resultO, CluId);
         END IF; // UNCLASSIFIED or NOISE
        END FOR;
       END IF;
       seedsList.delete(currentO);
    END WHILE; // seedsList <> Empty
    RETURN True;
  END IF
END; // ExpandCluster
```

## VI. SIMULATION AND RESULTS

Real datasets and artificial datasets were tested to prove the validity of DDBSCAN. The artificial dataset has two main clusters which are different densities as they are adjacent clusters as in Figure 3. This dataset has 3706 points.

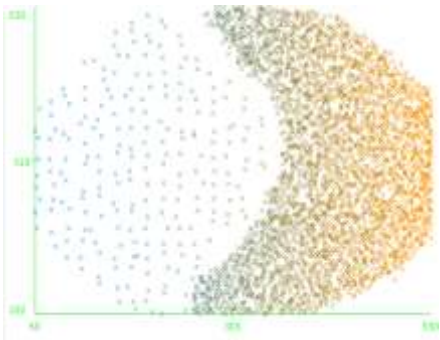

Fig. 3: different densities and adjacent clusters dataset.

Weka[10] is used as the experiment tool. From experiment, DBSCAN fails to detect adjacent clusters and also different densities ones. Figure 4 shows the result of clustering using DBSCAN.

The following figures show the drawbacks of DBSCAN which lie between two cases. In one hand, it fails to cluster the dataset and mark most of them as noise as in figure 4. In the other hand, DBSCAN fails to discriminate the adjacent and different densities clusters by producing single cluster as in figure 5.

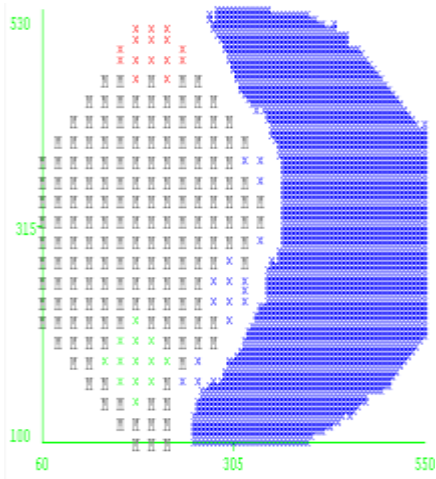

Fig. 4: DBSCAN result if parameters are Eps=.06 and MinPts=6

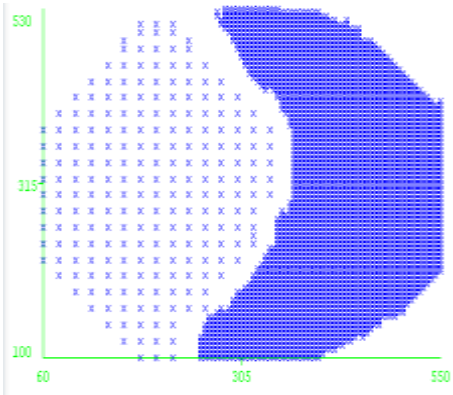

Fig. 5: DBSCAN result if parameters are Eps=.061 and MinPts=6

Clustering using DDBSCAN becomes better than DBSCAN especially adjacent clusters and different densities. The following figure shows the results of clustering using DDBSCAN. The key behind DDBSCAN powerful is threshold that defines which node has to join the cluster. As well as, threshold must define weight of node individually before joining it.

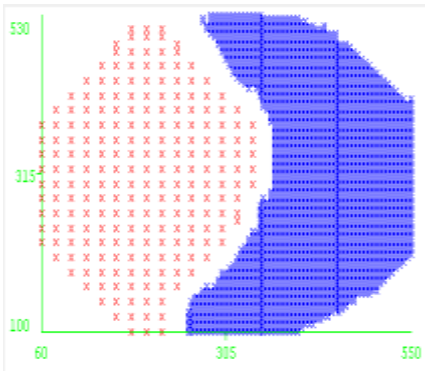

Fig. 5: DDBSCAN results using Eps=29 and Threshold=0.95

Real datasets have been used for testing as well. Iris , Haberman from UCI [9] which are used. In addition to, three artificial datasets have been tested. AdjacentDS and NoisyDS. DDBSCAN has proved its ability and validity for identifying any types of clusters whether it is adjacent, different densities or general cases.

## VII. Conclusion

No profound of density-based clustering algorithms because of its ability to cluster datasets with arbitrary shapes. Many studies had DBSCAN to overcome main problems of it. In this literature, DDBSCAN is proposed to handle mentioned DBSCAN problems. Other algorithms tried to solve these problems before like ST-DBSCAN, but still have concerns with different-densities databases and adjacent clusters as well as, it adds new parameters other Eps and MinPts. DDBSCAN works with two parameters only and proved its ability to overcome DBSCAN problems especially adjacent clusters and different-densities clusters.

## References

[1] Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In Kdd(Vol. 96, No. 34, pp. 226-231).

[2] Ankerst, M., Breunig, M. M., Kriegel, H. P., & Sander, J. (1999, June). OPTICS: ordering points to identify the clustering structure. In ACM Sigmod Record (Vol. 28, No. 2, pp. 49-60). ACM.

[3] Birant, D., & Kut, A. (2007). ST-DBSCAN: An algorithm for clustering spatial–temporal data. Data & Knowledge Engineering, 60(1), 208-221.

[4] Liu, P., Zhou, D., & Wu, N. (2007, June). VDBSCAN: varied density based spatial clustering of applications with noise. In Service Systems and Service Management, 2007 International Conference on (pp. 1-4). IEEE.

[5] Ester, M., Kriegel, H. P., Sander, J., Wimmer, M., & Xu, X. (1998, August). Incremental clustering for mining in a data warehousing environment. In VLDB(Vol. 98, pp. 323-333).

[6] Fawzy, M., Badr, A., Reda, M., & Farag, I. (2013). DBCLUM: Density-based Clustering and Merging Algorithm. International Journal of Computer Applications, 79(14), 1-6.

[7] Mitra, S., & Nandy, J. (2011). KDDClus: A Simple Method for Multi-Density Clustering. SKAD'11–Soft Computing Applications and Knowledge Discovery, 72.

[8] Karypis, G., Han, E. H., & Kumar, V. (1999). Chameleon: Hierarchical clustering using dynamic modeling. Computer, 32(8), 68-75.

[9] (http://archive.ics.uci.edu/ml/).

[10] Witten, I. H., Frank, E., Trigg, L. E., Hall, M. A., Holmes, G., & Cunningham, S. J. (1999). Weka: Practical machine learning tools and techniques with Java implementations.