Comparing Some Case Of Density-based Spatial Clustering Of Applications With Noise (DBSCAN) With Changing Data And Algorithm Parameter(HW3)

AmirHossein Asadi
Computer Engineering Department
Shahid Rajaee Teacher Training University
Tehran, Iran
amirasadi@sru.ac.ir

***Abstract*—In this homework first we review the concepts of clustering specially density based ones and equations and concept behind them. then we show the results of DBSCAN in six different condition. after that discuss about the results.**

***Index Terms*—Clustering, Density Based Clustering, DBSCAN, Clustering Evaluation**

I. Introduction

"Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters)" [1]
We use clustering when have some data with out label or tag for example consider we have a lot of height and weight measurements of cat and dog and wants to group them without knowing the animal type. these groups of algorithm called unsupervised learning in machine learning. in reality there are more situations that you have the data but you don't have the label.
These algorithms have many application such as: bioinformatics, Medicine, Business and marketing, World wide web, Computer science, Social science and etc. for example in medical imaging clustering can help to differentiate between different types of tissue in a three-dimensional image for many different purposes [2]. There are many way you can separate clustering methods one of them depends on nature of algorithms. In thins manner wen can define five different type of clustering algorithms including:

1) Connectivity-based clustering (hierarchical clustering)
2) Centroid-based clustering
3) Distribution-based clustering
4) Density-based clustering
5) Grid-based clustering

In this homework we want to analyses some cases with different data and parameters of Density-based Spatial Clustering Of Applications With Noise - DBSCAN method. as the algorithm name shows this algorithm belongs to forth group, named Density-based clustering.
In density based approaches we define a group of data a cluster if they have enough density and objects in spare areas are considered noise or border points.DBSCAN and OPTICS[1] are most known in density based approaches. we can say OPTICS is generalized form of DBSCAN.
It was first introduced in 1996 by Ester et al [3]. The advantages of this algorithm against other approaches are that in this method data could be in any shape and form while in k-means data must be compact and not in complex forms. another benefit is that there is no need to define value of k the algorithm observe all the data and add cluster if it is needed.

II. Methods

DBSCAN has two main parameters epsiolon(ϵ) and minPoint. ϵ defines the distance to search for object and minPoint defines minimum number of objects to be dense. there are three type of point in DBSCAN:

- Core
- Border
- Noise

Core is a point which has at least p point in distance d. Border is a point that has at lease one core point at distance d, and Noise is a point that there is less than p point in distance d or in other word it is not core nor border. **Reachability** means if a point distance is less than ϵ it is reachable. **Connectivity** means when two points are connected through reachable dense space. you can see an example in figure 1. the method can be summery in algorithm 1.

III. Results

In this section we run six different case for DBSCAN algorithm some changes in input data and other in algorithm it

[1]Ordering points to identify the clustering structure

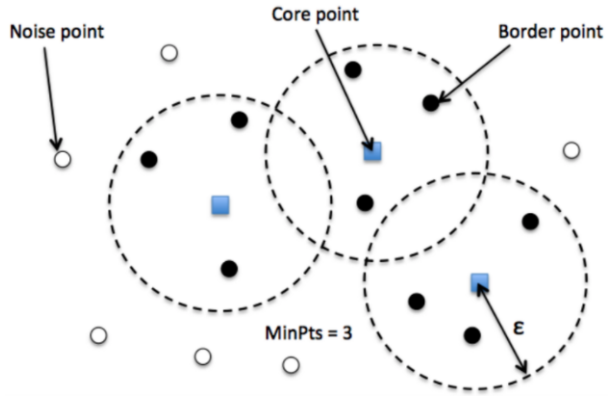


Fig. 1: Example of DBSCAN and its definitions [4].

Algorithm 1: DBSCAN pseudocode

Input: ϵ, MinPoint, DistanceFunction, DB

set $X_{un} = X$
set $m = 0$; // Cluster counter
while $X_{un} \neq \emptyset$ **do**
  Arbitrarily select a $x \in X_{un}$
  if *x is a noncore point* **then**
    Mark x as a noise point
    $X_{un} = X_{un} - \{x\}$
  end
  if *x is a core point* **then**
    $m = m + 1$
    Determine all density-reachable points in X from x
    Assign x and the previous points to the cluster C_m
    The border points that may have been marked as noise are also assigned to C_m
    $X_{un} = X_{un} - C_m$
  end
end

self.all codes and implementation are done with python 3.8.5 and sklearn package [5]. you can see the experiments and their details in table I. We can see the result of clustering for base condition in figure 2 and its confusion matrix in figure 3. the first row of confusion matrix is for noise data. and then we run the algorithm in cases that we just change the input data

TABLE I: Summery of designed experiments

Experiment NO	Details
base	Bolbs dataset, ϵ = 3, MinPoint = 3
1	No Structure
2	Noisy Moon dataset
3	Noisy Circles dataset
4	Manhattan distance
5	$\epsilon = 0.5$
6	MinPoint = 5

TABLE II: Evaluation of base case accuracy was 0.97

Label	precision	recall	f1-score
0	1.00	0.96	0.98
1	1.00	0.97	0.98
2	1.00	0.96	0.98

but with the same algorithm parameter in base condition. the result is show in figure 4.

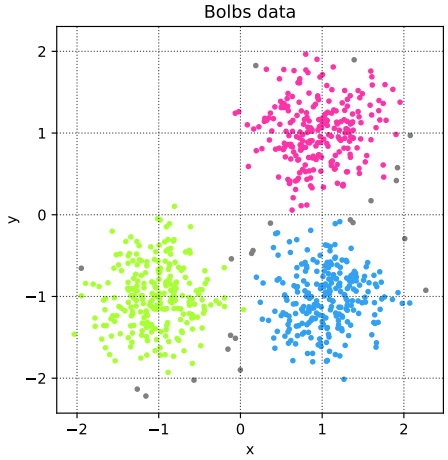


Fig. 2: DBSCAN in base experiment, gray points are considered as noise

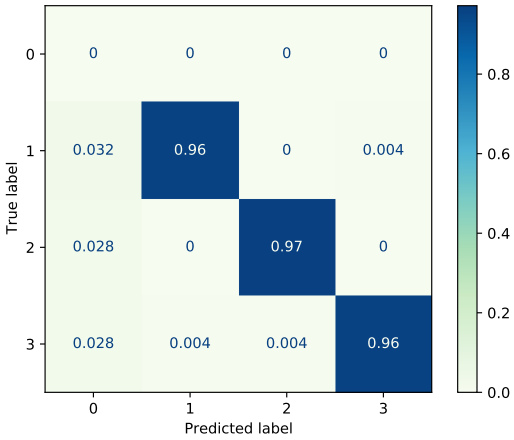


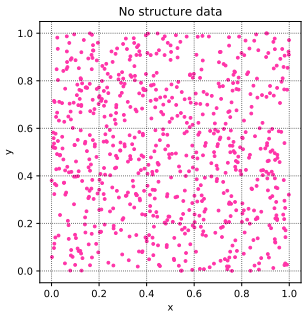
Fig. 3: Normalized confusion matrix

In figure 5 you can see the results when we change the parameters of algorithm such as metric function, ϵ and MinPoint.

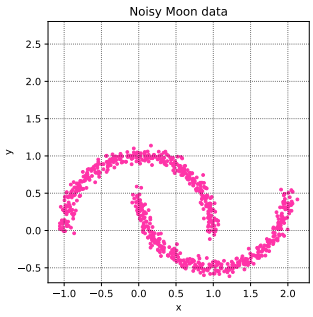
in table tables II to VIII you can see the result of evaluating the clustering methods.

IV. Discussion

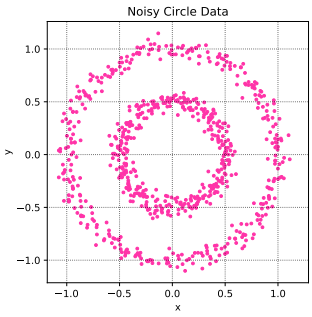
In section III we see the results of our clustering in base and six different case and after that their evaluation with five



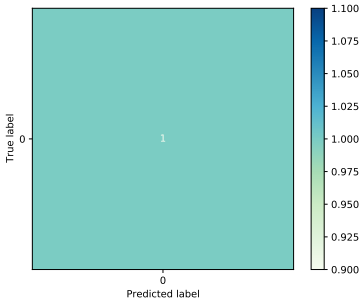
(a) DBSCAN in first experiment



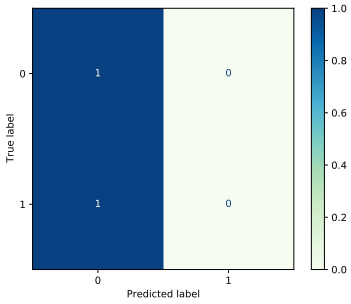
(b) DBSCAN in second experiment



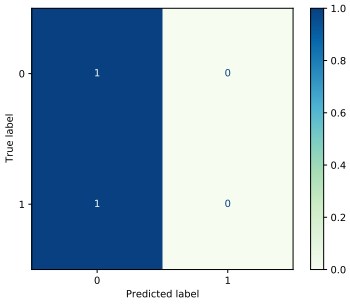
(c) DBSCAN in third experiment



(d) Confusion matrix of no structure data



(e) Confusion matrix of noisy moon data



(f) Confusion matrix of noisy circle data

Fig. 4: Result of first a three experiment as you can see in figs. 4a and 4b DBSCAN just found one cluster while there is two, and this is because of setting its parameters not precise.

TABLE III: Evaluation of first case accuracy was 1.00

Label	precision	recall	f1-score
0	1.00	1.00	1.00

TABLE IV: Evaluation of second case accuracy was 0.50

Label	precision	recall	f1-score
0	0.50	1.00	0.67
1	0.00	0.00	0.00

TABLE V: Evaluation of third case accuracy was 0.50

Label	precision	recall	f1-score
0	0.50	1.00	0.67
1	0.00	0.00	0.00

TABLE VI: Evaluation of forth case accuracy was 0.93

Label	precision	recall	f1-score
-1	0.00	0.00	0.00
0	1.00	0.93	0.96
1	1.00	0.94	0.97
2	1.00	0.92	0.96

TABLE VII: Evaluation of fifth case accuracy was 0.66

Label	precision	recall	f1-score
-1	0.00	0.00	0.00
0	1.00	0.98	0.99
1	0.49	0.98	0.66
2	0.00	0.00	0.00
3	0.00	0.00	0.00

TABLE VIII: Evaluation of sixth case accuracy was 0.65

Label	precision	recall	f1-score
-1	0.00	0.00	0.00
0	1.00	0.98	0.99
1	0.49	0.98	0.66
2	0.00	0.00	0.00
3	0.00	0.00	0.00

different measure including: precision, recall, f1-score and confusion matrix.

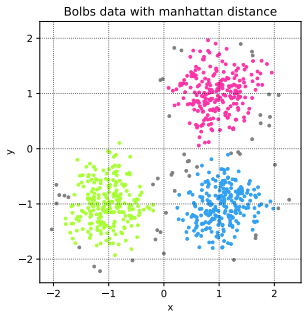
From the results we understand DBSCAN has three main parameter

1) ϵ
2) MinPoint
3) Distance Function

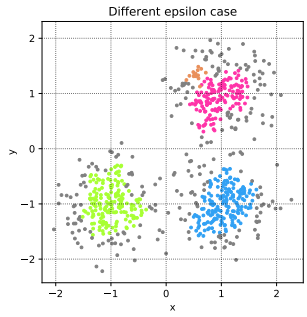
These six experiment show if we make ϵ smaller it will make more data as noise and on the other side if we choose larger value we have less noise data but if it is large enough it may merge two cluster.

The other parameter is MinPoint results shows that if we choose smaller value it will make more clusters which maybe they are not actually cluster figure 5c is example of that on the other hand if we choose large value for MinPoint as it requires lots of point to be dense then we may have more border points and noise point.

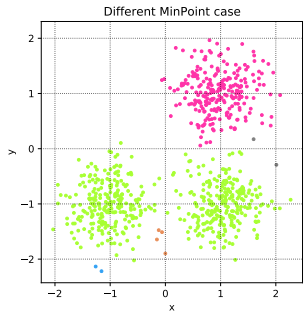
Testing many different distance function such as: cityblock, cosine, euclidean, l1, l2, manhattan' 'braycurtis, canberra, chebyshev, correlation, dice, hamming, jaccard, kulsinski, ma-



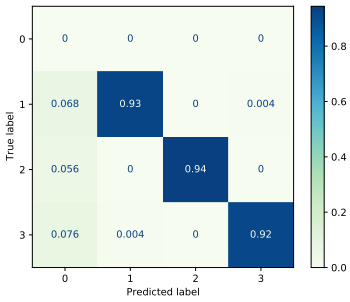
(a) DBSCAN result in forth experiment



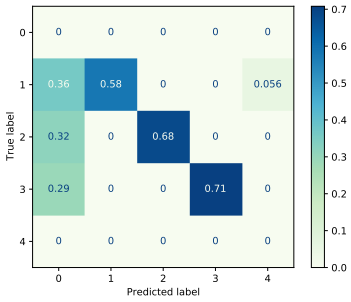
(b) DBSCAN result in fifth experiment



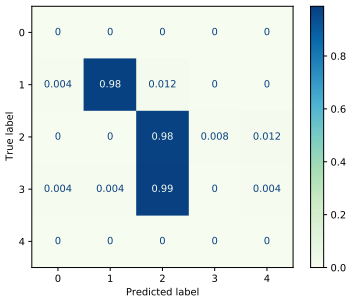
(c) DBSCAN result in sixth experiment



(d) Confusion matrix of manhattan distance



(e) Confusion matrix of $\epsilon = 0.15$



(f) Confusion matrix of MinPoint = 2

Fig. 5: Result of second three experiment, here we change parameters of DBSCAN such as metric function, ϵ and MinPoint.

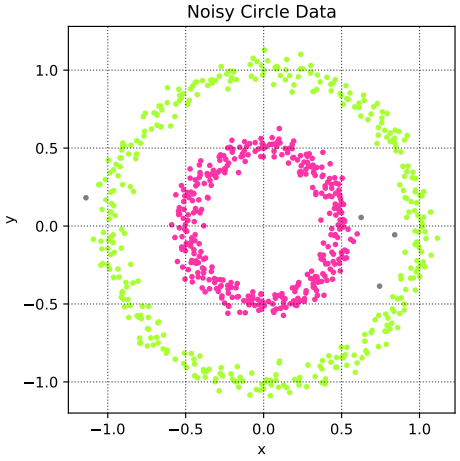


Fig. 6: Noisy Cricle with $\epsilon = 0.1$ and MinPoint=3

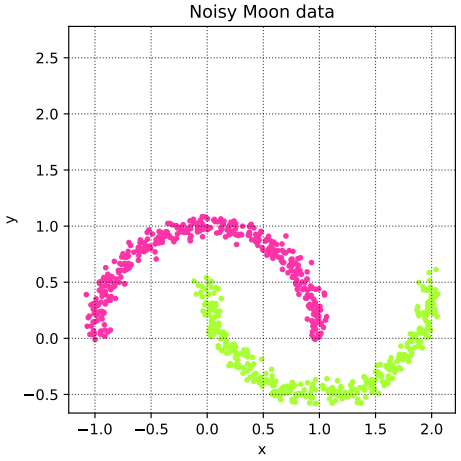


Fig. 7: Noisy Moon with $\epsilon = 0.3$ and MinPoint = 10

halanobis, minkowski, rogerstanimoto, russellrao, seuclidean, sokalmichener, sokalsneath, sqeuclidean and yule show that the default one(euclidean) has the best results and after that manhattan distance while most of others function fails badly and could find just one cluster in three circle cluster.

V. Conclusion

In this homework we use synthesis some data to analyses DBSCAN algorithm with different parameters and to measure the quality of our results we use five different evaluation measurement. we design six different case which three of them has different input data and others has different algorithm parameters and one base case to compare them.

Results shows that if parameters of this algorithm tuned well it can do clustering even data in complex forms while other common and well known method such as k-means fail. on the other hand there is no need to know the number of cluster before the algorithm. also running time of this algorithm

relatively is better than others. the implementation is available here [6].

REFERENCES

[1] Wikipedia contributors, "Cluster analysis — Wikipedia, the free encyclopedia." https://en.wikipedia.org/w/index.php?title=Cluster_analysis&oldid=1005655155, 2021. [Online; accessed 12-February-2021].

[2] R. Filipovych, S. M. Resnick, and C. Davatzikos, "Semi-supervised cluster analysis of imaging data," *NeuroImage*, vol. 54, no. 3, pp. 2185–2197, 2011.

[3] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise.," in *Kdd*, vol. 96, pp. 226–231, 1996.

[4] N. S. Chauhan, "Dbscan clustering algorithm in machine learning." https://www.kdnuggets.com/2020/04/dbscan-clustering-algorithm-machine-learning.html, 2020.

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[6] Amirhosein Asadi, "implementation of hw4." https://github.com/amirasaadi/PR399_DBSCAN, 2021. [Online; accessed 13-February-2021].