ECE 365 (Fall 2023) **Instructor:** S. Bose

Quiz 6

Assigned: Oct 4, Deadline: Oct 6 Total points = 20

Name and UIN: Abhay Narayanan

Instructions. You are allowed to collaborate with your peers for this quiz. Write the names of your collaborators. You are allowed to use any resource (including the internet). Appropriately cite any reference you use.

Question 1/1. [20 points]

Consider the set of points $\mathcal{D}$ shown in the picture below. These are generated as two different clusters, one subset denoted via blue triangles and the other subset denoted via orange circles.

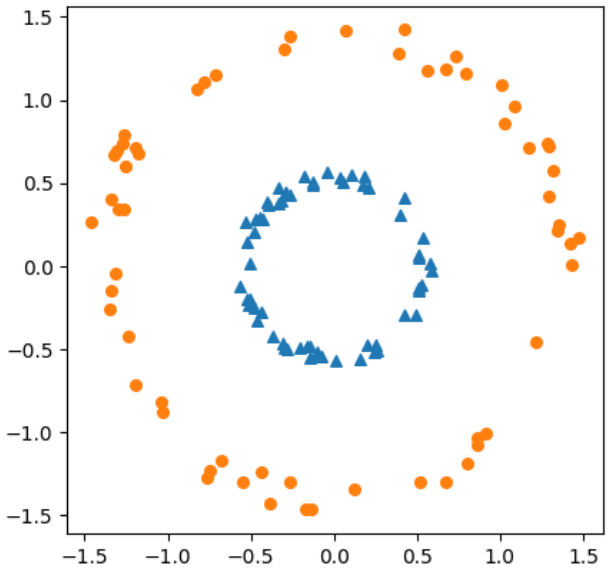


Figure 1: Scatter plot of two sets of points on a plane.

Describe what you expect as the output of 2-means clustering. If 2-means clustering fails to work, can you think about a method that produces the clusters based on "nearness" of neighbors, rather than creating a center for each cluster?

Propose a method in a document that will allow a fellow ECE 365 student to code your method. Provide some intuition behind your method. There are a variety of established techniques that can be used to solved this problem. If you happen to use a resource, cite it. You are welcome to not use any resource and think creatively to come up with a technique. In either case, describe

clearly the intuition and the mechanics of the algorithm.

Citation:

KDnuggets, April 4, 2022: DBSCAN Clustering Algorithm in Machine Learning
https://www.kdnuggets.com/2020/04/dbscan-clustering-algorithm-machine-learning.html

Please proceed to the next page for the solution to this quiz.

Thanks :)

When attempting to classify such a dataset into two distinct classes using the 2-means clustering method, we'd expect to get an output like the following:

In the above image, we obtain the two different classes as a result of applying 2-means clustering. This is because the algorithm selects eventually computes two centers and performs classification on a point by assigning it to a class if the distance from the point to a given center is shortest compared to other distance(s) of the point to centers. As such, the only possible outcome this could give for 2 classes is one where the classes are clearly separated by a plane. Therefore, this would fail in this dataset because the clusters are nested (one is within the other).

Intuition: As such, this can be rectified by a model that thinks in a different manner, without assigning centroids and computing minimum distance. It is especially important to avoid assigning a fixed number of centers and finding other points around them (as k-means fails), but rather one that could use a center and then determine a class around it, and further from those points and so on, recursively. This is because the **density** of points is what is needed to correctly classify a dataset of concentric circles, as given. One such example is the *DBSCAN* (Density-Based Spatial Clustering of Applications with Noise) algorithm.

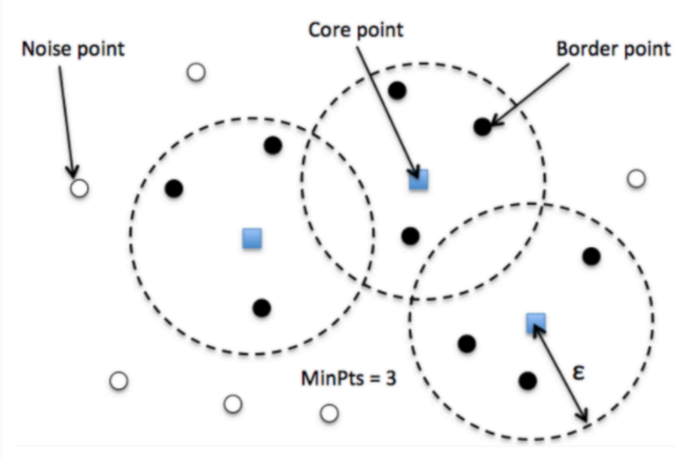
Please note that there are two parameters, *minPoints* and ϵ.

minPoints: The minimum number of points clustered together for them to be considered a class

ϵ: The maximum distance from a core point for a point to be considered the same class as the core point.

It works in the following manner:

- The algorithm first picks a point (a core point) in the dataset.
- The algorithm then checks if there are at least *minPoints* points within a radius ϵ from it. Once all points within this region have been exhausted, these points are all considered one class.
- These classes are then expanded through recursive repetitions of the previous computation to determine a similar neighborhood calculation for each neighboring point (a point within the neighborhood of a core point).
- The algorithm continues until all points in the dataset have been visited. Any points that do not manage to form a class due to there being an insufficient number of points in its vicinity are simply "noise", and can be classified as any class.



An example of the implementation of the DBSCAN algorithm.

Notes:

- The distance measure should be some reasonably chosen metric, as this could have a significant impact on classification.

- The value of ϵ can be chosen by plotting the distance to the k = *minPoints* - 1 nearest neighbor ordered from the largest to the smallest value, and choosing ϵ as the "elbow" of the function.

- Large values for *minPoints* may be necessary for noisy or very large data sets. The same is true for datasets with many duplicates. As a rule of thumb, however, *minPoints* = 2*dim (where dim is the number of dimensions in the dataset) will be acceptable.