

Density-Based Methods ***(DBSCAN Algorithm)***

- ***Density-based*** clustering is an unsupervised learning algorithm that groups similar data points in a dataset based on their ***density***.
- Where the ***density*** refers the ***number of data points*** within a ***specified radius*** (**epsilon ϵ**) of a given data point.
- The most popular density-based clustering algorithm is **DBSCAN** (**D**ensity-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise).

Why Density-based clustering ?

- We have seen in the partitioning clustering (eg: K-means) there are a predetermined number of clusters, and hierarchical clustering forms the clusters along with all data points.(i.e. including noise / outliers).
- To solve the above challenges, we can opt for the Density-based clustering algorithm because, in this algorithm, we ***do not need*** to have knowledge about the ***predefined number*** of clusters and It ignores the ***noise or outlier*** data points.

DBSCAN *Algorithm*

(**D**ensity-**B**ased Spatial **C**lustering of **A**pplications
with **N**oise).

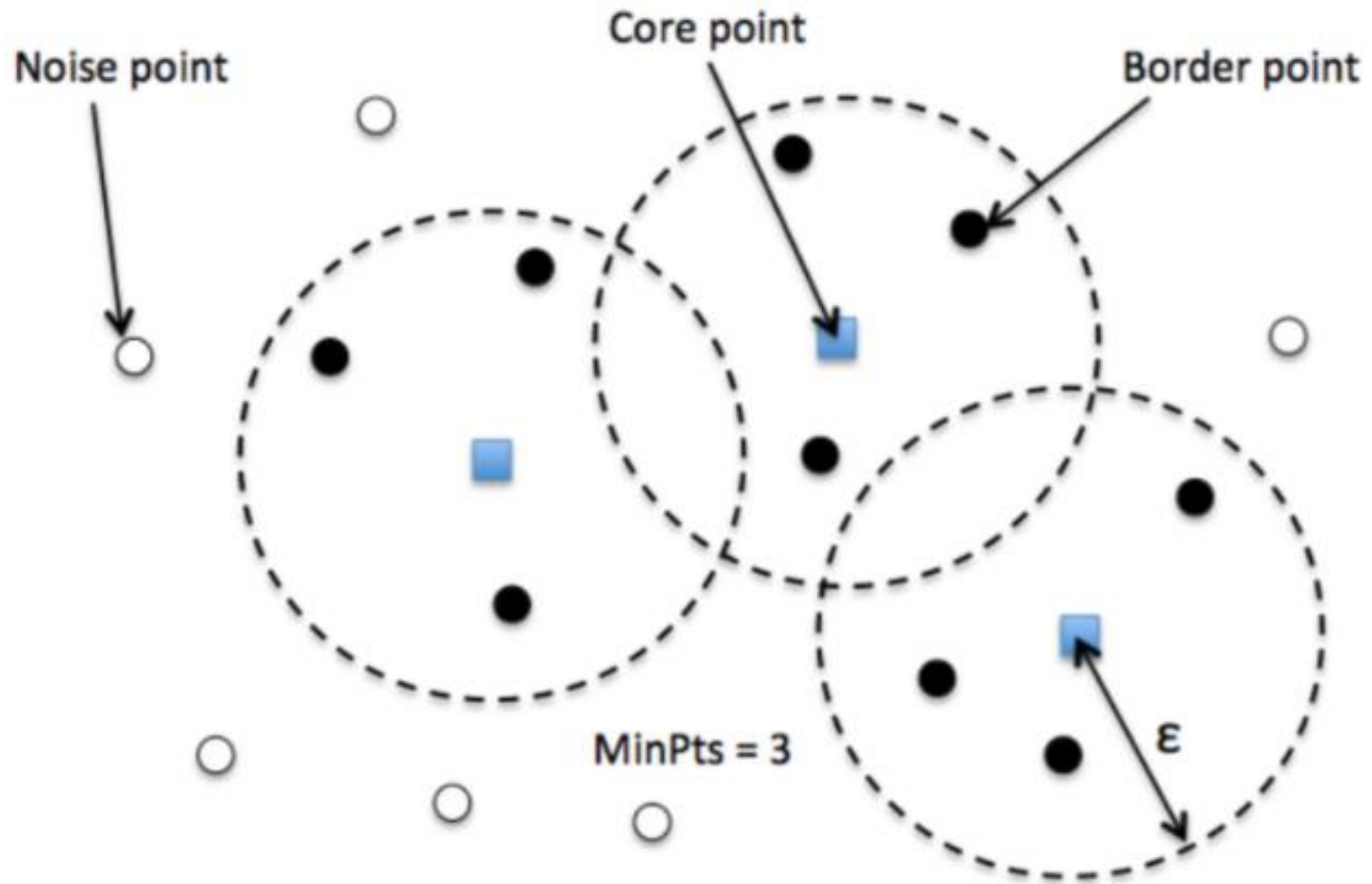
- One commonly used density-based clustering algorithm is **DBSCAN** (Density-Based Spatial Clustering of Applications with Noise).
- DBSCAN groups the data points that are close to each other based on a ***distance measure*** and a ***density threshold***.
- It can discover clusters of **different *shapes*** and ***sizes*** from a large amount of data, which is containing ***noise*** and ***outliers***.

The DBSCAN algorithm uses *two* parameters:

- ***eps* (ϵ):** The epsilon parameter defines the radius of the neighbors around a datapoint ***x***. (or) It is a distance measure that will be used to locate the points in the neighborhood of any point.
- ***MinPts*:** The ***MinPts*** parameter is the minimum number of neighbors (threshold) within the “***eps***” radius.

In density-based clustering, the ***number of clusters*** does not need to be predetermined, but the values of ***MinPts*** and ***eps*** do.

- In the DBSCAN algorithm, there are three types of points: *core points*, *border points*, and *noise points*.



- **Core point:** A core point is a point that has at least ***MinPts*** within its ***epsilon*** neighbourhood. In other words, a core point has enough nearby neighbours to be considered as part of a cluster.
- **Border point:** A border point is a point that is ***not a core point***, but it is within the epsilon neighbourhood of a core point. Border points can be considered as ***part of a cluster***, but they are not as strongly connected to the cluster as core points.
- **Noise point:** A noise point is a point that ***does not have any core points*** within its epsilon neighbourhood. Noise points are often considered ***outliers*** and are not part of any cluster.

DBSCAN algorithm:

Here are the steps involved in the DBSCAN algorithm:

Step -1: First, define ***epsilon*** and identify ***MinPts***.

Step -2: Starting with a ***core point***, the algorithm finds all the points within its ***epsilon*** neighbourhood(radius) and assigns them to the same cluster. It then repeats this process for each core point and continues to add points to the same cluster as long as they are within epsilon distance of another point in the cluster.

Step -3: Identify border points: A border point is a point that has fewer than *MinPts* within its epsilon neighbourhood, but is *reachable* from a core point. These points are often considered part of a cluster but can also be considered noise.

Step-4: Output clusters: The algorithm outputs the clusters that have been identified.

- The *advantages* of DBSCAN algorithm:
- Density-based clustering algorithms can *handle noisy data* since they can identify noise points as outliers that do not belong to any cluster.
- Unlike some other clustering algorithms, density-based clustering can identify clusters of *different shapes* and *sizes*, including irregularly shaped clusters.
- They *don't require prior knowledge* of the number of clusters, making them more flexible and versatile.
- They can *efficiently process large datasets* and handle high-dimensional data.

- The *disadvantages* of DBSCAN algorithm:
- The performance of density-based clustering algorithms is *highly dependent* on the *choice of parameters*, such as **epsilon** ϵ and *MinPts*, which can be challenging to tune.
- They can be *computationally expensive* and time-consuming, especially for large datasets with complex structures.

Example: Apply the ***DBSCAN*** algorithm to given data points and create the clusters with

MinPts = 3 and epsilon = 1.9.

Data points:

P1: (3,7)

P2: (4,6)

P3: (5,5)

P4: (6,3)

P5: (7,3)

P6: (6,2)

P7: (3,3)

Solution:

The given data

MinPts = 3

Epsilon radius = 1.9

- In this we use ***Euclidian distance*** to calculate the distance between the points.

Euclidian distance (A (x1, y1), B (x2, y2))

$$= \sqrt{((x1 - x2)^2 + (y1 - y2)^2)}$$

Now, Calculate the *distance(Euclidian)* from each point to all the data points.

		P1	P2	P3	P4	P5	P6	P7
P1: (3,7)	P1	0						
P2: (4,6)	P2	1.41	0					
P3: (5,5)	P3	2.83	1.41	0				
P4: (6,3)	P4	5.00	3.60	2.23	0			
P5: (7,3)	P5	5.66	4.24	2.83	1.00	0		
P6: (6,2)	P6	5.83	4.47	3.16	2.00	1.00	0	
P7: (3,3)	P7	4.00	3.16	2.83	3.00	4.00	3.16	0

Next, find the *nearest data points* for each data point are:

minPts = 3 and *epsilon* = **1.9**

	P1	P2	P3	P4	P5	P6	P7	Nearest Datapoints
P1	0							P1: P2
P2	1.41	0						P2: P1,P3
P3	2.83	1.41	0					P3: P2
P4	5.00	3.60	2.23	0				P4:P5
P5	5.66	4.24	2.83	1.00	0			P5:P4,P6
P6	5.83	4.47	3.16	2.00	1.00	0		P6:P5
P7	4.00	3.16	2.83	3.00	4.00	3.16	0	P7

Once the calculate *nearest data points*, we need to find the *Core*, *Border* and *Noise* data points.

minPts = 3 and epsilon = 1.9

P1: P2
P2: P1,P3
P3: P2
P4:P5
P5:P4,P6
P6:P5
P7

Point	Status	
P1	Noise	
P2	Core	
P3	Noise	
P4	Noise	
P5	Core	
P6	Noise	
P7	Noise	

Once the calculate *nearest data points*, we need to find the *Core*, *Border* and *Noise* data points.

minPts = 3 and epsilon = 1.9

P1: P2
P2: P1,P3
P3: P2
P4:P5
P5:P4,P6
P6:P5
P7

Point	Status	
P1	Noise	Border
P2	Core	
P3	Noise	Border
P4	Noise	Border
P5	Core	
P6	Noise	Border
P7	Noise	

From the above,

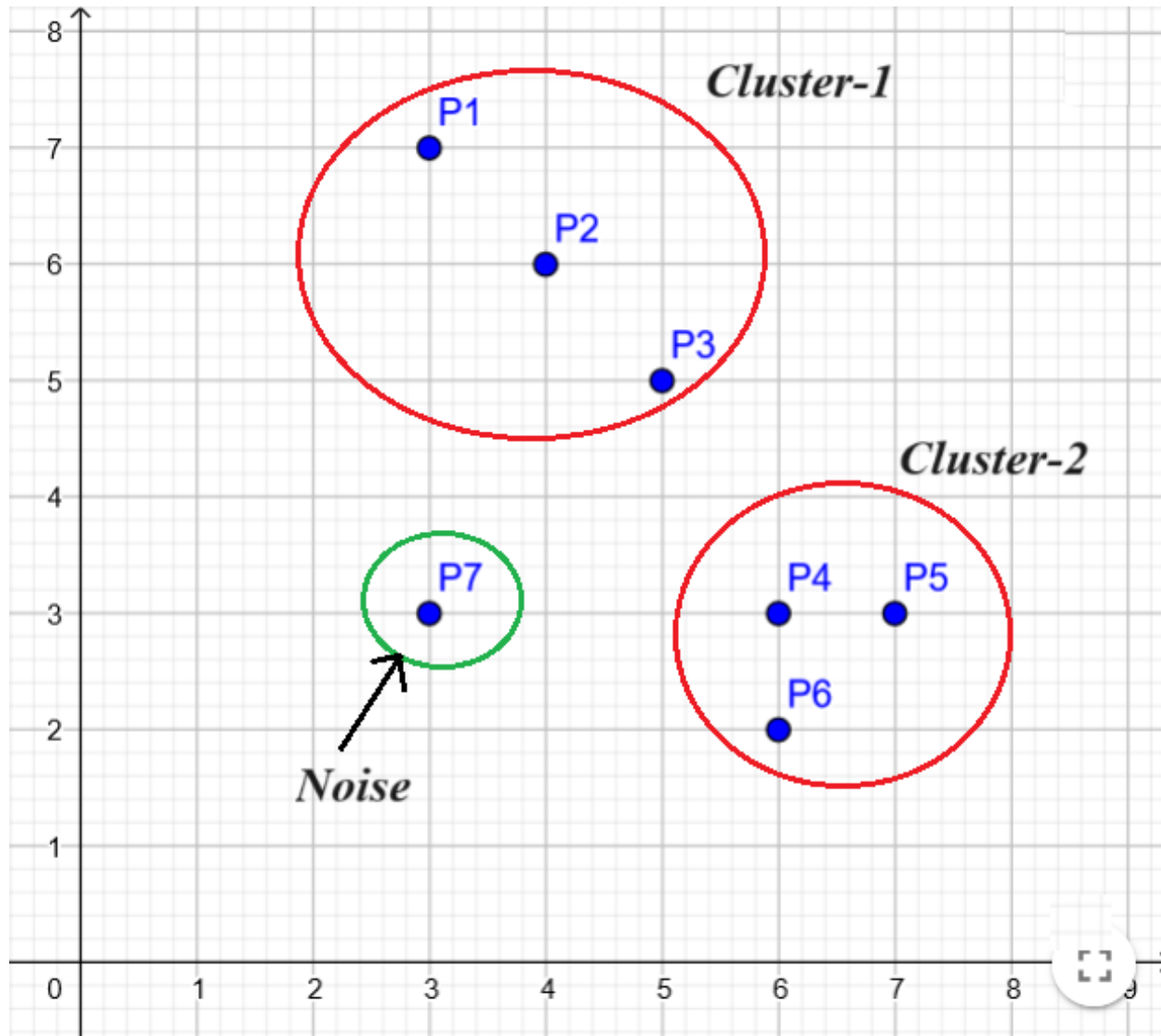
- **P2** and **P5** are *Core data points*
- **P1**, **P3**, **P4** and **P6** are *Border data points*
- **P7** is *Noise* (or) *Outlier* data point.

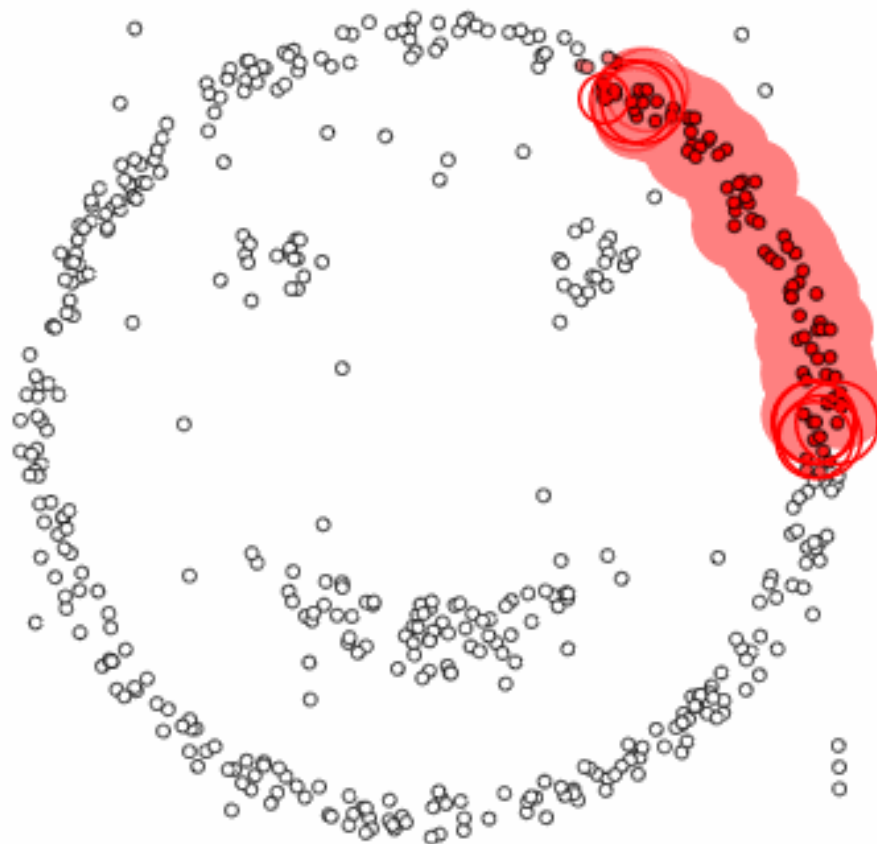
There are *two core* data points that means there are *two clusters*.

Those are

- **Cluster-1: (P2: P1, P3)**
- **Cluster-2: (P5: P4, P6)**

- Now, try to draw **clusters** on 2D space.





epsilon = 1.00
minPoints = 4

Restart



Pause