Clustering

Clustering is an unsupervised machine learning task. You might also hear this referred to as cluster analysis because of the way this method works.

DBSCAN is a density-based algorithm.

DBSCAN stands for Density-Based Spatial Clustering of Applications with Noise.

Density-based Clustering locates regions of high density that are separated from one another by regions of low density.

Density = number of points within a specified radius (Eps)

Tree type of DBSCAN:

■ A point is a **core point** if it has more than a specified number of points (MinPts) within Eps

➤These are points that are at the interior of a cluster

■ A **border point** has fewer than MinPts within Eps, but is in the neighborhood of a core point

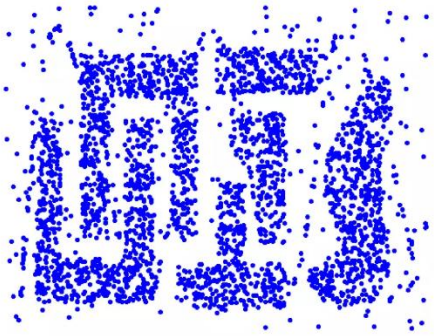
■ A **noise point** is any point that is not a core point or a border point

➤Any two core points are close enough within a distance Eps of one another - are put in the same cluster.

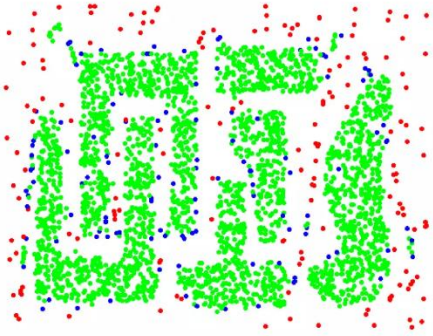
➤Any border point that is close enough to a core point is put in the same cluster as the core point.

➤Noise points are discarded.

Core, Border, Noise points representation



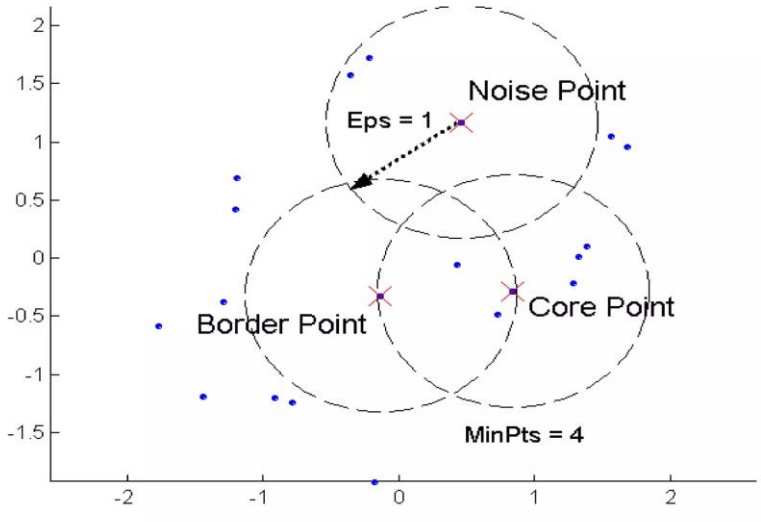
Original Points



Point types: core, border and noise

Eps = 10, MinPts = 4

Concepts: Core, Border, Noise



Example :

• Apply the DBSCAN algorithm to the given data points and Create the clusters with:

• minPts = 4 **desired minimum cluster size**

• epsilon (ε) = 1.9. **physical distance(radius)**

Data Points:

P1: (3, 7) **P2: (4, 6)** **P3: (5, 5)** **P4: (6, 4)** **P5: (7, 3)** **P6: (6, 2)**

P7: (7, 2) **P8: (8, 4)** **P9: (3, 3)** **P10: (2, 6)** **P11: (3, 5)** **P12: (2, 4)**

• Use Eucladian distance and calculate the distance between each points.

$$\text{Distance}(A(X1,Y1), B(X2,Y2)) = \sqrt{(x2-x1)^2 + (y2 - Y1)2}$$

minPts = 4 and epsilon (e) = 1.9

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
P1	0											
P2	1.41	0										
P3	2.83	1.41	0									
P4	4.24	2.83	1.41	0								
P5	5.66	4.24	2.83	1.41	0							
P6	5.83	4.47	3.16	2.00	1.41	0						
P7	6.40	5.00	3.61	2.24	1.00	1.00	0					
P8	5.83	4.47	3.16	2.00	1.41	2.83	2.24	0				
P9	4.00	3.16	2.83	3.16	4.00	3.16	4.12	5.10	0			
P10	1.41	2.00	3.16	4.47	5.83	5.66	6.40	6.32	3.16	0		
P11	2.00	1.41	2.00	3.16	4.47	4.24	5.00	5.10	2.00	1.41	0	
P12	3.16	2.83	3.16	4.00	5.10	4.47	5.39	6.00	1.41	2.00	1.41	0

P1: P2, P10

P2: P1, P3, P11

P3: P2, P4

P4: P3, P5

P5: P4, P6, P7, P8

P6: P5, P7

P7: P5, P6

P8: P5

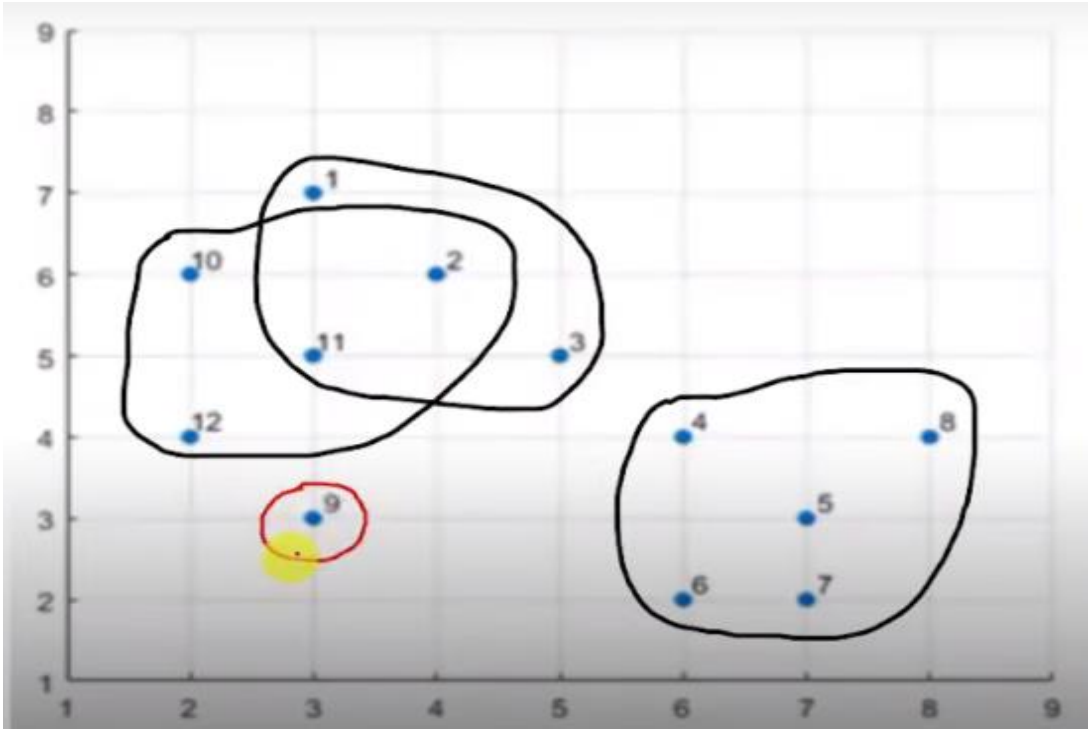
P9: P12

P10: P1, P11

P11: P2, P10, P12

P12: P9, P11

point	status	
P1	Noise	Border
P2	Core	
P3	Noise	Border
P4	Noise	Border
P5	Core	
P6	Noise	Border
P7	Noise	Border
P8	Noise	Border
P9	Noise	
P10	Noise	Border
P11	Core	
P12	Noise	Border



DBSCAN Algorithm

Input: N objects to be clustered and global parameters Eps, MinPts.

Output: Clusters of objects.

DBSCAN: Advantages

- No need to specify the number of clusters in advance.
- Can find clusters of arbitrary shapes.
- Robust to noise and outliers.
- Requires only two parameters.

DBSCAN: Disadvantages

- Difficulty in accurately determining epsilon values.
- Inefficiency for high-dimensional data.