DBSCAN

~S S Roy

DB SCAN is a powerful **density-based clustering algorithm** that groups together points that are closely packed, while marking points that lie alone in low-density regions as **outliers** or **noise**. The images you've provided show a step-by-step example of how to apply the DBSCAN algorithm to a set of 12 data points.

The Problem

We have a set of 12 data points, labeled P1 through P12, each with its own (X, Y) coordinates. Our goal is to cluster these points using the DBSCAN algorithm.

Step 1: Define Density Parameters

Before we begin, we must define the two key parameters for the DBSCAN algorithm. These parameters dictate what we consider "dense" and are essential for the algorithm to work.

- **Epsilon (ϵ)**: This is the maximum distance between two points for them to be considered **neighbors**. The images show ϵ=1.9.
- **Minimum Points (MinPts)**: This is the minimum number of points required in a neighborhood to form a **dense region** or a **core point**. The images don't explicitly state the value, but based on the calculations, we can infer that MinPts = 4.

Step 2: Calculate the Distance Matrix

.

The first step is to calculate the **Euclidean distance** between every pair of points. This forms a distance matrix where the value at row i and column j is the distance between point P_i and point P_j.

$$Distance(P_i, P_j) = \sqrt{(X_j - X_i)^2 + (Y_j - Y_i)^2}$$

For example, the distance between P1 (4.5, 8) and P2 (5, 7) is:

$$Distance(P_1, P_2) = \sqrt{(5 - 4.5)^2 + (7 - 8)^2} = \sqrt{(0.5)^2 + (-1)^2} = \sqrt{0.25 + 1} = \sqrt{1.25} \approx 1.12$$

Your images provide the complete distance matrix. We can use this to find all the neighbors for each point, which are the points within a distance of $\epsilon = 1.9$.

Step 3: Classify Each Point

Now, we use the distance matrix and our defined parameters (ϵ=1.9 and MinPts=4) to classify each point as a **core**, **border**, or **noise** point.

1. **Core Point**: A point is a **core point** if its neighborhood (including itself) contains at least MinPts (4) points.
2. **Border Point**: A point is a **border point** if its neighborhood contains fewer than MinPts, but it's a neighbor of a core point.
3. **Noise Point**: A point is a **noise point** if it's not a core point and it's not a border point.

Point	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
P1	0.00	1.12	2.12	3.91	6.02	5.59	5.70	5.41	4.03	1.58	2.06	3.16
P2	1.12	0.00	1.12	2.83	5.00	4.47	4.61	4.47	3.16	2.06	1.41	2.50
P3	2.12	1.12	0.00	1.80	3.91	3.64	3.61	3.35	3.20	3.16	2.06	2.92
P4	3.91	2.83	1.80	0.00	2.24	2.00	1.80	2.00	3.16	4.72	3.16	3.50
P5	6.02	5.00	3.91	2.24	0.00	2.24	1.12	1.00	5.00	6.95	5.39	5.59
P6	5.59	4.47	3.64	2.00	2.24	0.00	1.12	2.83	3.16	6.02	4.24	4.03
P7	5.70	4.61	3.61	1.80	1.12	1.12	0.00	1.80	4.03	6.40	4.72	4.74
P8	5.41	4.47	3.35	2.00	1.00	2.83	1.80	0.00	5.10	6.50	5.10	5.50
P9	4.03	3.16	3.20	3.16	5.00	3.16	4.03	5.10	0.00	3.64	2.00	1.12
P10	1.58	2.06	3.16	4.72	6.95	6.02	6.40	6.50	3.64	0.00	1.80	2.55
P11	2.06	1.41	2.06	3.16	5.39	4.24	4.72	5.10	2.00	1.80	0.00	1.12
P12	3.16	2.50	2.92	3.50	5.59	4.03	4.74	5.50	1.12	2.55	1.12	0.00

Here's the classification for each point based on the data provided:

- **P1**: The neighbors of P1 (points within 1.9) are P2 and P10. The total count is 3. Since the count is less than MinPts (4), P1 is not a core point.
- **P2**: The neighbors of P2 are P1, P3, and P11. The total count is 4 (including P2 itself). Since the count is equal to MinPts (4), **P2 is a core point** and starts a new cluster.
- **P3**: The neighbors of P3 are P2 and P4. The total count is 3. P3 is not a core point.
- **P4**: The neighbors of P4 are P3 and P7. The total count is 3. P4 is not a core point.

- **P5**: The neighbors of P5 are P6 and P8. The total count is 3. P5 is not a core point.
- **P6**: The neighbors of P6 are P5, P7, and P8. The total count is 4. Since the count is equal to MinPts (4), **P6 is a core point** and starts a new cluster.
- **P7**: The neighbors of P7 are P4, P5, P6, P8, P12. The total count is 6. Since the count is greater than MinPts (4), **P7 is a core point**.
- **P8**: The neighbors of P8 are P5, P7. The total count is 3. P8 is not a core point.
- **P9**: The neighbors of P9 are P12. The total count is 2. P9 is not a core point.
- **P10**: The neighbors of P10 are P1 and P11. The total count is 3. P10 is not a core point.
- **P11**: The neighbors of P11 are P2, P10, and P12. The total count is 4. Since the count is equal to MinPts (4), **P11 is a core point**.
- **P12**: The neighbors of P12 are P9, P11. The total count is 3. P12 is not a core point.

Step 4: Final Clustering

Now we form the final clusters by connecting the core points and their associated border points.

1. **Cluster 1**: P2, P11, and P7 are all core points. We can see from the distance matrix that they are linked:
    - P2 is a neighbor of P11.
    - P11 is a neighbor of P12.
    - P12 is a neighbor of P9.
    - P7 is a neighbor of P4.
    - P4 is a neighbor of P3.

This creates one large cluster consisting of P1, P2, P3, P4, P10, P11, P12, P7, P9.

2. **Cluster 2**: The other core point is P6. It is a neighbor of P5 and P8. P5 and P8 are neighbors of each other and of P6. This forms a smaller cluster. The total number of points in this cluster is 3.

Let's re-examine the classifications from the last image to find any corrections or insights. The final image (CLUSTERING-5.PNG) shows the final classification after applying the DBSCAN logic:

- **P1**: Classified as **Noise** and then **Border**. This happens because it's not a core point, but it's a neighbor of the core point **P2**.
- **P2**: Classified as a **Cluster** point, specifically a **Core Point**.
- **P3**: Classified as **Noise** and then **Border**. It's a neighbor of the core point **P2**.

- **P4**: Classified as **Noise** and then **Border**. It's a neighbor of the core point **P7**.
- **P5**: Classified as **Noise** and then **Border**. It's a neighbor of the core point **P6**.
- **P6**: Classified as a **Cluster** point, specifically a **Core Point**.
- **P7**: Classified as a **Cluster** point, specifically a **Core Point**.
- **P8**: Classified as **Noise** and then **Border**. It's a neighbor of the core point **P7** and **P6**.
- **P9**: Classified as a **Noise** point. It has only one neighbor **P12**, which is not a core point, making P9 a true outlier or noise point.
- **P10**: Classified as **Noise** and then **Border**. It's a neighbor of the core point **P2** and **P11**.
- **P11**: Classified as a **Cluster** point, specifically a **Core Point**.
- **P12**: Classified as **Noise** and then **Border**. It's a neighbor of the core point **P11**.

Therefore, the final output would be:

- **Cluster 1**: P1, P2, P3, P4, P5, P6, P7, P8, P10, P11, P12.
- **Noise**: P9.

The video's final classification is a bit different, suggesting some of these points (P1, P3, P4, P5, P6, P8, P10, P12) are border points but are still part of the clusters. The point P9 is correctly identified as Noise.

The key takeaway is that DBSCAN effectively identifies dense regions and groups them into clusters while automatically detecting and separating outliers.