32 Density-Based Clustering: From Density Peaks to Natural Clusters

32.1 When Density Defines a Cluster

32.1.1 From Points to Populations: The Intuition of Density-Based Clustering

Density-based clustering groups data points according to the density of their local neighborhoods, identifying dense regions as clusters and sparse regions as boundaries. Among density-based algorithms, DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is one of the most widely used and will be the focus of this chapter.

DBSCAN forms clusters by aggregating points that are closely packed together while treating points in low-density regions as noise or outliers. It relies on two key parameters: the neighborhood radius (ε) and the minimum number of points required to form a dense region (min_samples).

Other density-based methods include OPTICS, which improves clustering by building a reachability distance map, and DENCLUE, which models data density with Gaussian kernels and identifies density peaks using gradient-based methods. A major advantage of density-based clustering is its flexibility: it does not require clusters to have specific shapes and is robust to noise and outliers.

32.1.2 Inside the Neighborhood: Connecting Core, Border, and Noise Points

The core idea of DBSCAN can be visualized through a social network analogy. As shown in Figure 1, within a limited distance (represented by the circular neighborhoods), users with a sufficient number of followers are considered core points—they are the ones wearing crowns.

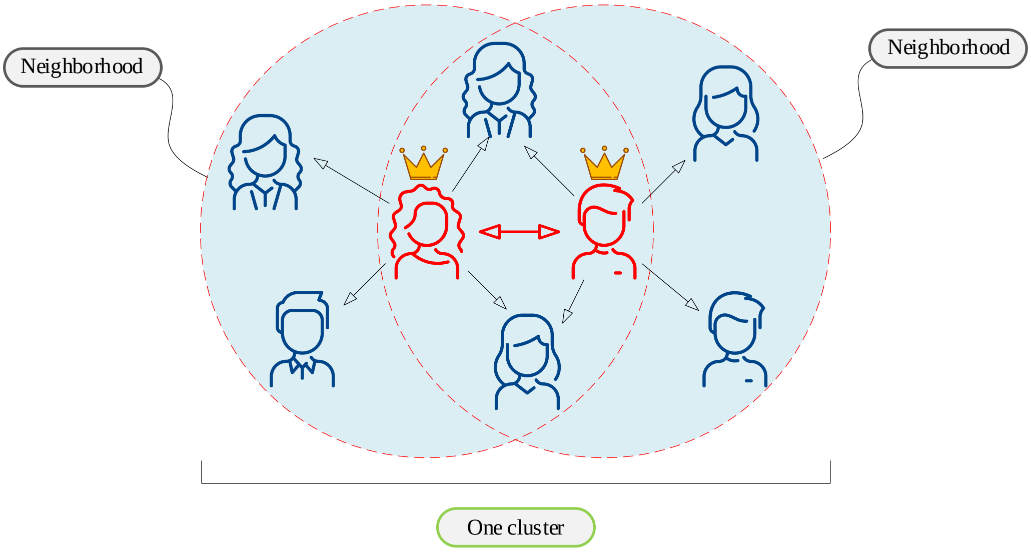


Figure 1. DBSCAN Principle: Core Points and Neighborhoods

Users within a core point's circle but without enough followers themselves are border points. DBSCAN's key principle is that if two crowned users are connected with each other—meaning each lies within the other's neighborhood—the two core points and all their followers form a single cluster.

In this way, connected communities naturally emerge as clusters without needing to predefine their number or shape.

In a simple example as shown in Figure 2, 8 points are scanned: only one point meets the core point criteria, several become border points, and the rest are noise. Expanding to more points, DBSCAN can identify multiple clusters, with core points acting as the centers that link nearby neighborhoods into larger connected clusters.

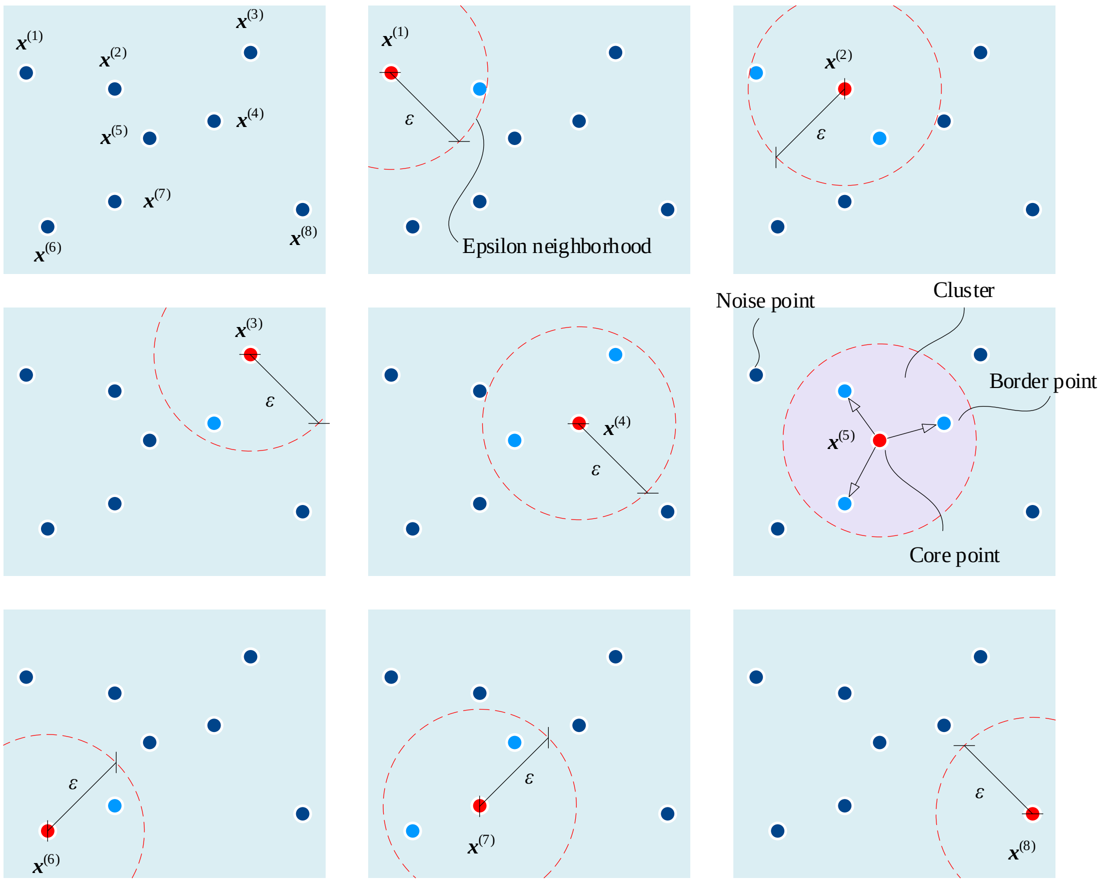


Figure 2. Scanning Sample Points to Identify Core, Border, and Noise Points

Figure 3 shows how DBSCAN identifies clusters based on data density. Each red circle represents a core point, surrounded by its neighborhood defined by the radius ε. Blue points are border points, located within the neighborhood of a core point but not dense enough to be core points themselves. Dark blue points scattered outside are noise points, too isolated to belong to any cluster.

Clusters form naturally when neighborhoods of core points overlap — as seen here, three groups emerge: C_1, C_2, and C_3. Each cluster grows outward from its core points, connecting nearby border points, while noise remains unassigned.

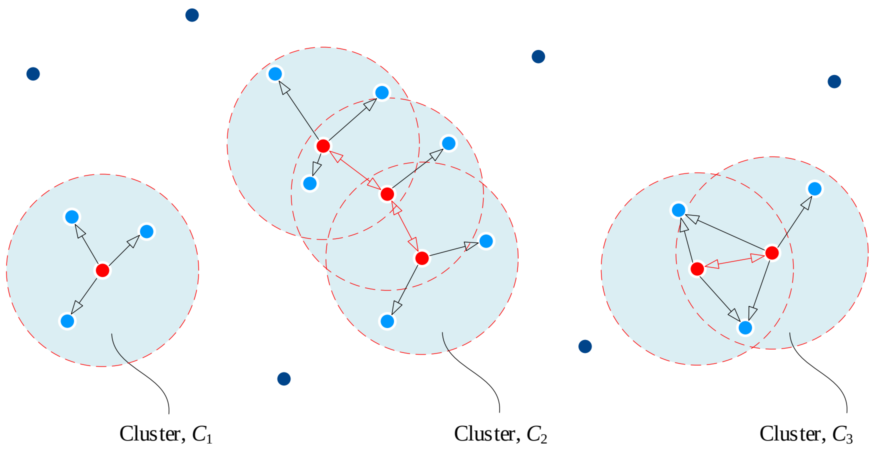


Figure 3. DBSCAN Clustering Results: Three Clusters Identified

32.2 Tuning the Parameters: EPS and min_samples in Action

32.2.1 Finding the Right Neighborhood Radius (EPS)

In DBSCAN, the quality of clustering depends heavily on two key parameters: the neighborhood radius (ε, or EPS) and the minimum number of points required to form a dense region (min_samples).

Understanding how to adjust these parameters is essential for achieving meaningful clusters. The neighborhood radius EPS defines how far around a point we search for neighboring points. If EPS is set too small, many points may fail to meet the density requirement, leading to excessive noise points and fragmented clusters.

Conversely, if EPS is too large, distant points may be grouped together, merging distinct clusters into a single one. For example, in a ring-shaped dataset, an EPS of 0.1 marks most points as noise, while increasing EPS to 0.2 reduces noise and begins forming clusters.

At EPS = 0.4, the algorithm correctly identifies two clusters, but at EPS = 0.6, all points are grouped into a single cluster. It is also worth noting that the neighborhood distance does not have to be Euclidean; other distance metrics can be used depending on the data.

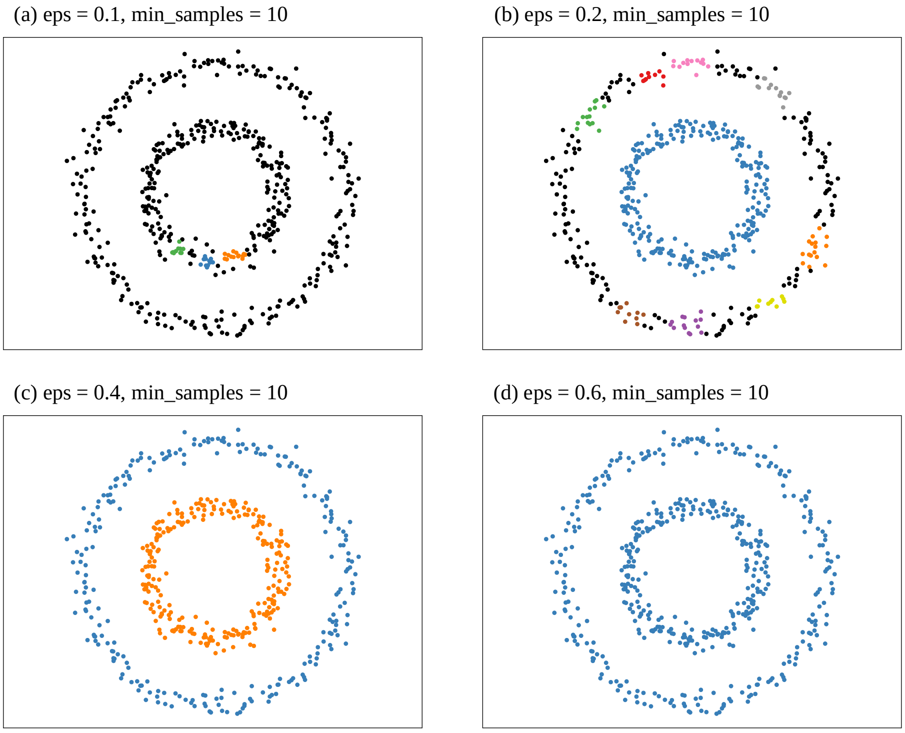


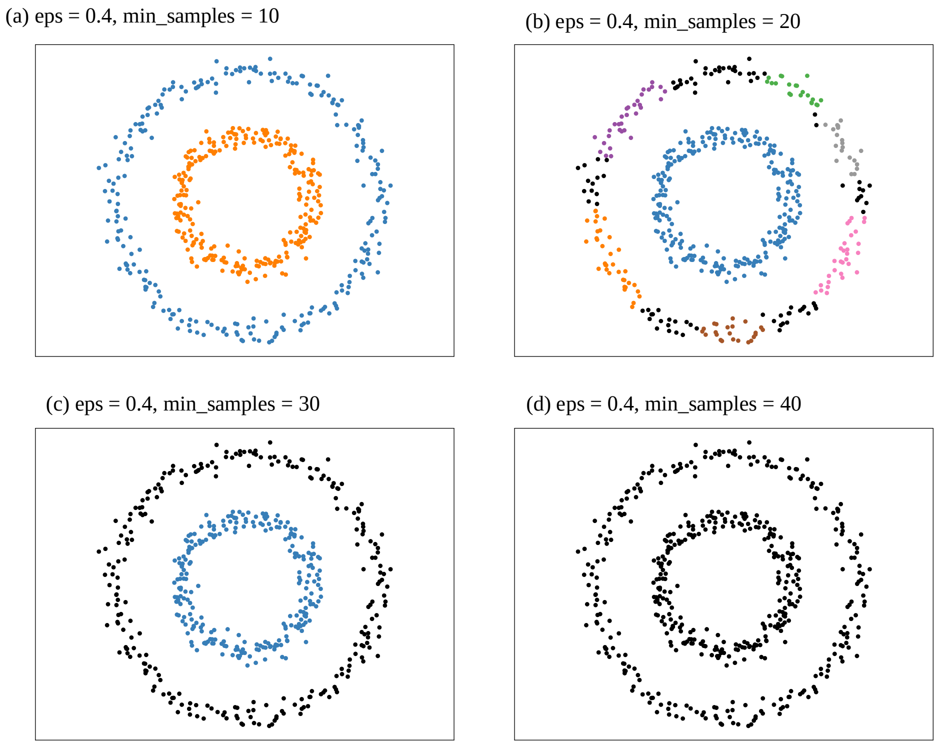
Figure 4. Effect of EPS on DBSCAN Clustering. Figure generated by Ch32_01_DBSCAN.ipynb.

32.2.2 How Many Friends Make a Crowd? (min_samples)

The parameter min_samples controls the minimum number of points needed to define a dense region. Raising min_samples increases the algorithm's tolerance to noise, which is useful when the dataset contains many outliers.

Unlike K-means or Gaussian Mixture Models, DBSCAN does not require specifying the number of clusters in advance. It can adapt to arbitrary cluster shapes and naturally identify outliers.

Because DBSCAN is sensitive to both EPS and min_samples, careful coordination of these parameters is crucial for accurate clustering. Adjusting them in tandem ensures clusters are well-defined while noise points are correctly identified.



(a) eps = 0.4, min_samples = 10

(b) eps = 0.4, min_samples = 20

(c) eps = 0.4, min_samples = 30

(d) eps = 0.4, min_samples = 40

Figure 5. Effect of min_samples on DBSCAN Clustering

32.3 Conclusion

Density-based clustering groups data points by identifying regions of high density while treating sparse areas as boundaries or noise. DBSCAN, or Density-Based Spatial Clustering of Applications with Noise, is a popular algorithm in this category. It defines clusters by connecting points that meet a density requirement, called core points, along with nearby points that do not meet the threshold, called border points. Points that are neither core nor border are classified as noise.

Clusters form naturally by linking reachable core points and their neighbors, without needing to specify the number of clusters in advance. The success of DBSCAN depends on careful selection of two key parameters, the neighborhood radius and the minimum number of points required to form a dense region. If the neighborhood is too small, clusters may fragment and produce many noise points. If it is too large, distinct clusters may merge. DBSCAN is robust to noise, flexible with cluster shapes, and well suited for complex datasets where traditional algorithms may struggle.