DBSCAN

Learning Objectives

By the end of this lesson, students should be able to:

- Describe the effect of **`epsilon`** and **`min_points`** on DBSCAN.
- Identify advantages and disadvantages of DBSCAN.
- Implement DBSCAN using **`sklearn`**

k-Means Clustering

In unsupervised learning, one strategy is to cluster observations into groups.

Observations of the same group are more similar than observations in different groups.

We've learned one clustering algorithm so far: k-means.

k-Means Clustering

What are the pros and cons of k-means clustering?

DBSCAN

There's another method of clustering that can sidestep some of the disadvantages of k-means: **DBSCAN**

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

DBSCAN

There's another method of clustering that can sidestep some of the disadvantages of k-means: **DBSCAN**

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

We detect areas of high density and low density:

- We cluster in areas of high density
- We avoid clustering in low-density areas

Hyperparameters

DBSCAN requires you to specify two hyperparameters:

1. **`min_samples`**: the minimum number of points needed to form a cluster
2. **`epsilon`**: the “searching” distance when attempting to build a cluster

How does DBSCAN work?

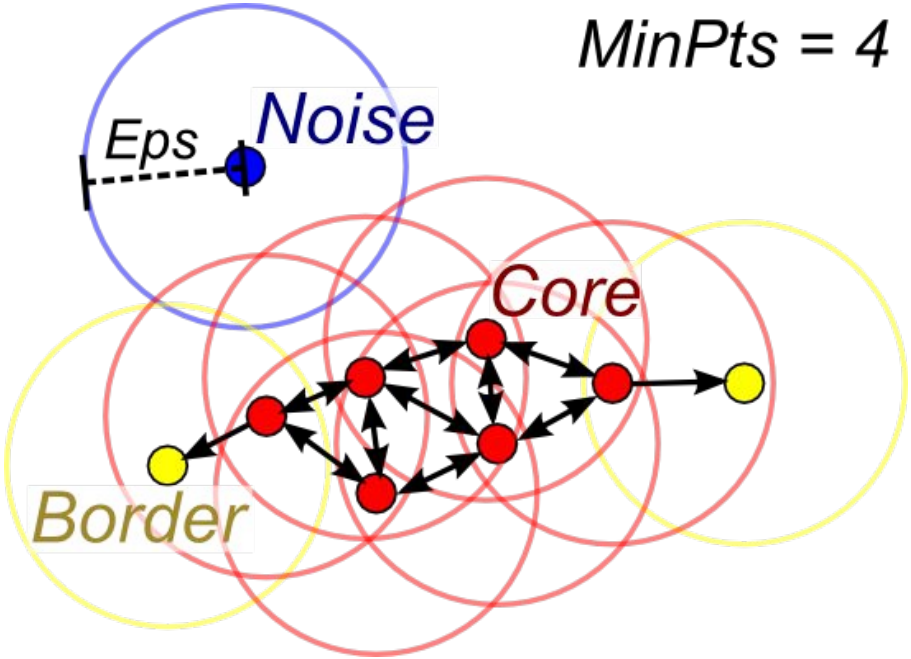
```
DBSCAN(DB, distFunc, eps, minPts)
   C = 0
   for each point P in database DB
      if label(P) ≠ undefined then continue
      Neighbors N = RangeQuery(DB, distFunc, P, eps)
      if |N| < minPts then
         label(P) = Noise
         continue
      C = C + 1
      label(P) = C
      Seed set S = N \ {P}
      for each point Q in S
         if label(Q) = Noise then label(Q) = C
         if label(Q) ≠ undefined then continue
         label(Q) = C
         Neighbors N = RangeQuery(DB, distFunc, Q, eps)
         if |N| ≥ minPts then
            S = S ∪ N
```

Source: https://en.wikipedia.org/wiki/DBSCAN#Algorithm

Visualizing DBSCAN

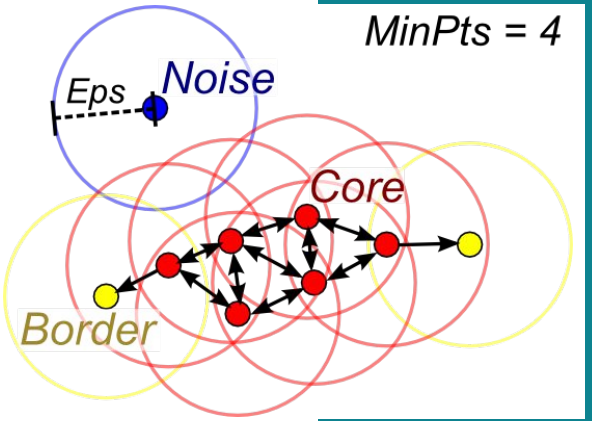
https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/

Visualizing DBSCAN



Visualizing DBSCAN

- **Core points**: Points inside a cluster that have at least **`min_samples`** points within **`epsilon`**.
- **Border points**: Points inside a cluster that do not have at least **`min_samples`** points within **`epsilon`**
- **Noise**: Points that belong to no cluster



Why DBSCAN?

DBSCAN allows us to detect some cluster patterns that k-Means might not be able to detect.

We don’t need to pre-specify the number of clusters; the algorithm will determine how many clusters are appropriate given fixed min_samples and epsilon values. This is particularly valuable when we are clustering data in more than two or three dimensions.

Not every point is clustered! Good for identifying outliers.

Disadvantages of DBSCAN

DBSCAN requires us to tune two parameters.

DBSCAN works well when clusters are of a different density than the overall data, but does not work well when the clusters themselves are of varying density.

Fixed **`epsilon`**.