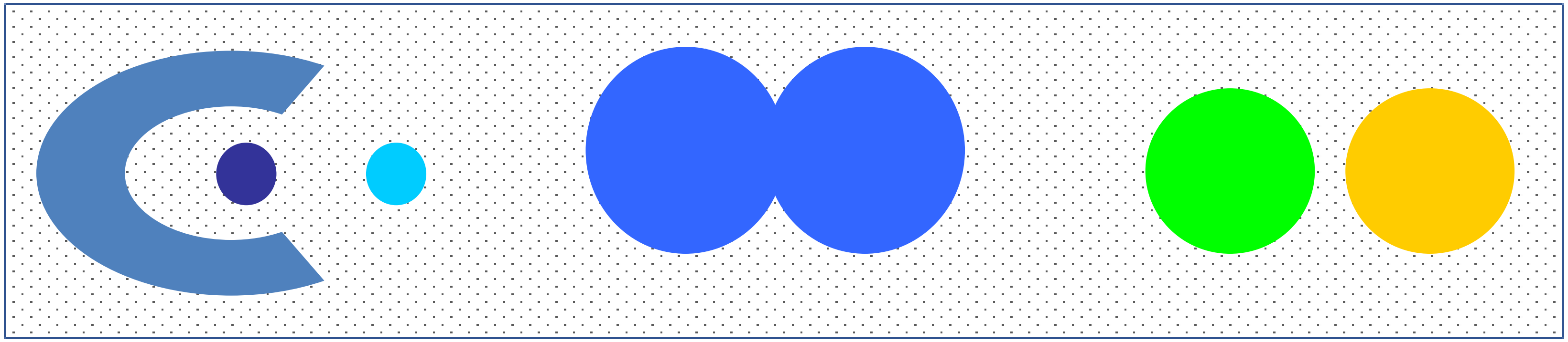
Density-Based clustering: DBSCAN

Lecture 11

by Marina Barsky

Types of Clusters: Density-Based

- Clusters are defined as dense regions of objects in the data space that are separated by regions of low density (representing noise)
- To discover such clusters we need special algorithms



6 density-based clusters

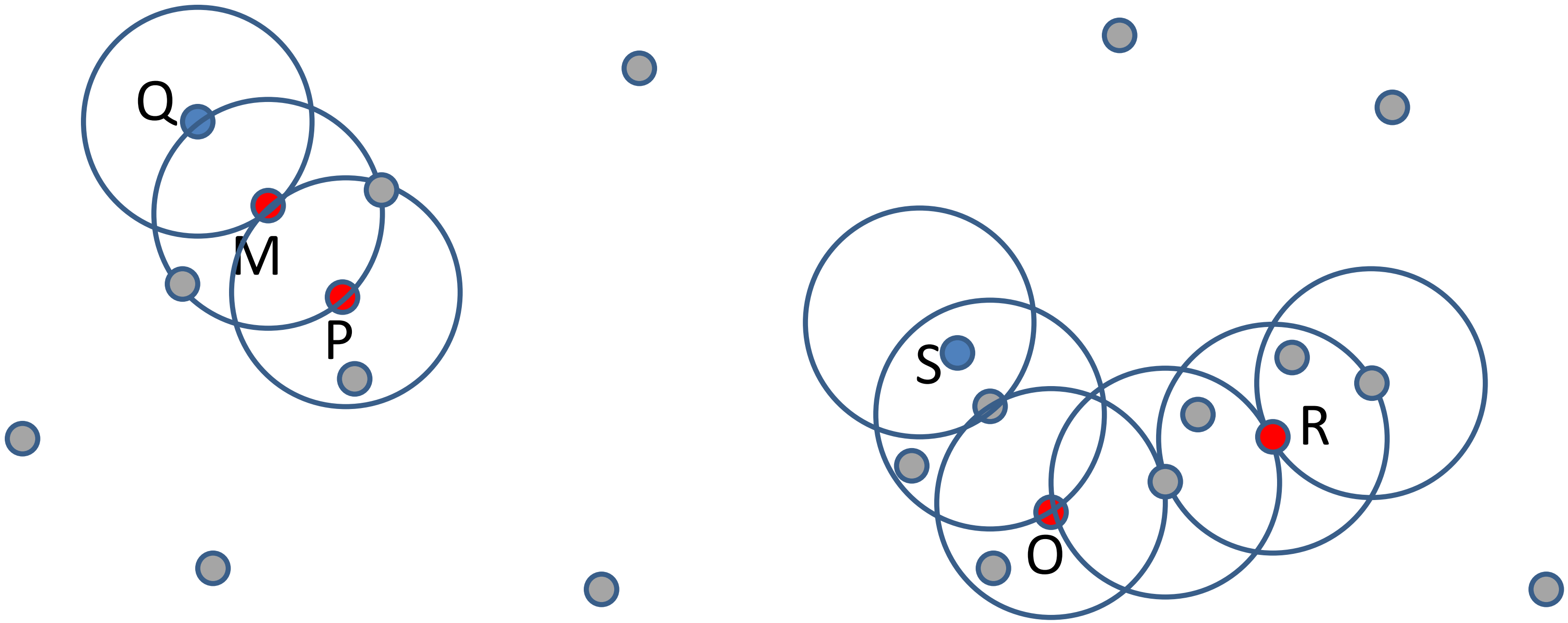
DBSCAN - Density-Based Spatial Clustering of Applications with Noise

New definitions

- The neighborhood within a radius ε of a given object is called the *ε-neighborhood* of the object

- If the ε-neighborhood of an object contains at least a minimum number *MinPts* of objects, then such an object is called **a core point**

Core points example: MinPts=3



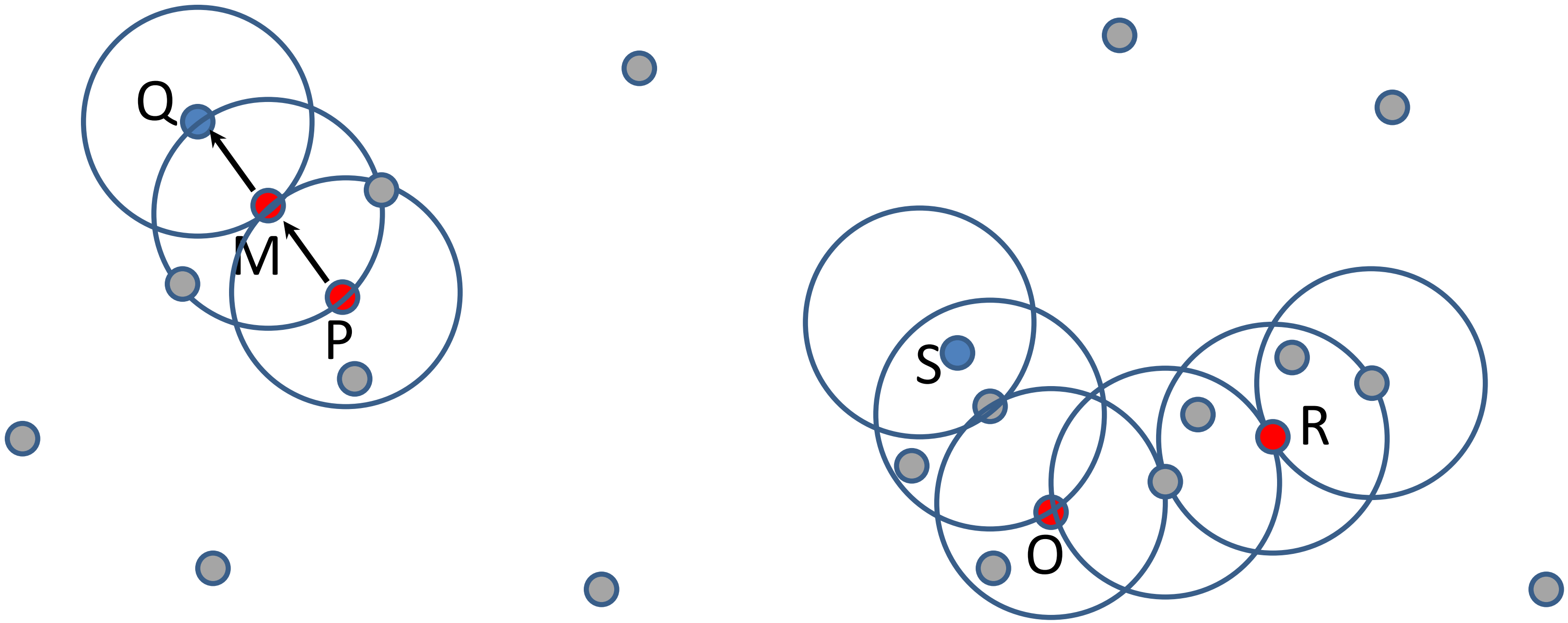
M, P, O and R are core points, since each contains at least 3 points in its ε-neighborhood

DBSCAN - Density-Based Spatial Clustering of Applications with Noise

More definitions

- We say that object p is directly reachable from object q if p is within ε-neighborhood of q, and q is a **core point**

Directly reachable example: MinPts=3



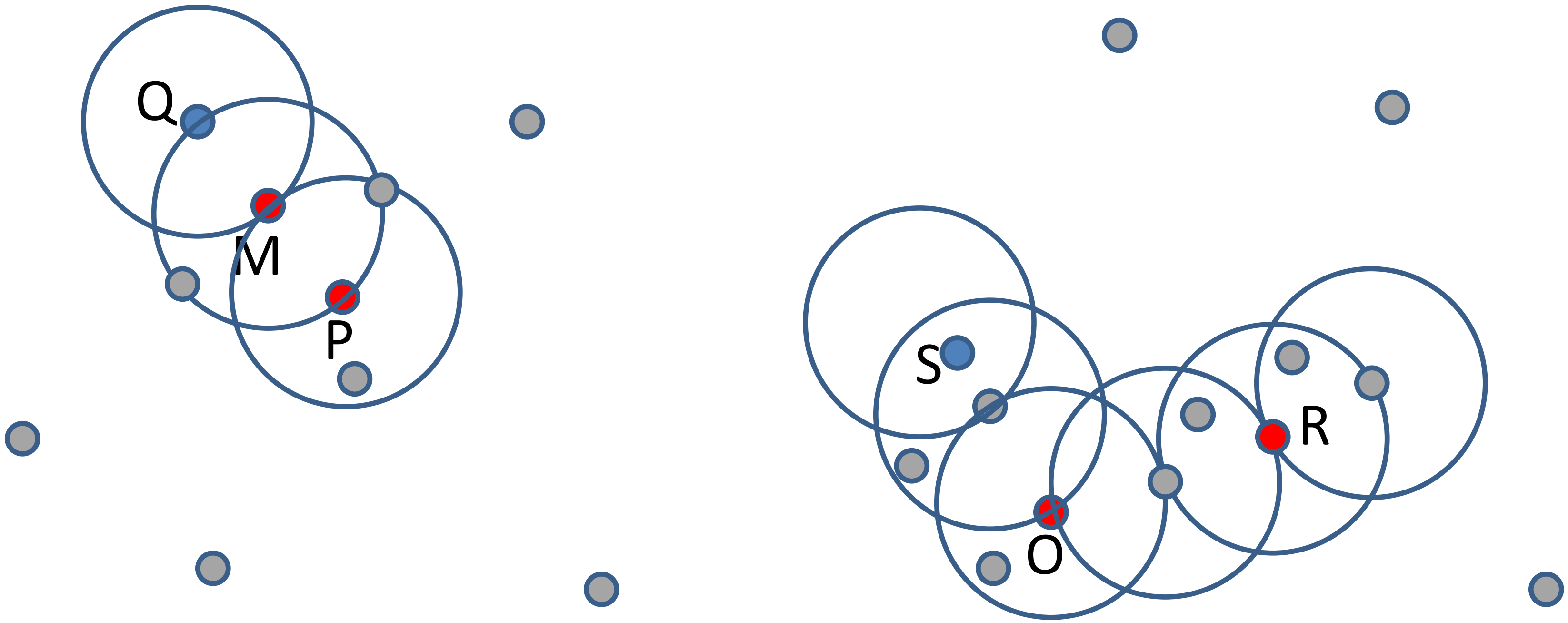
Q is directly density-reachable from M, M is directly density reachable from P, and P is directly density-reachable from M

DBSCAN - Density-Based Spatial Clustering of Applications with Noise

More definitions

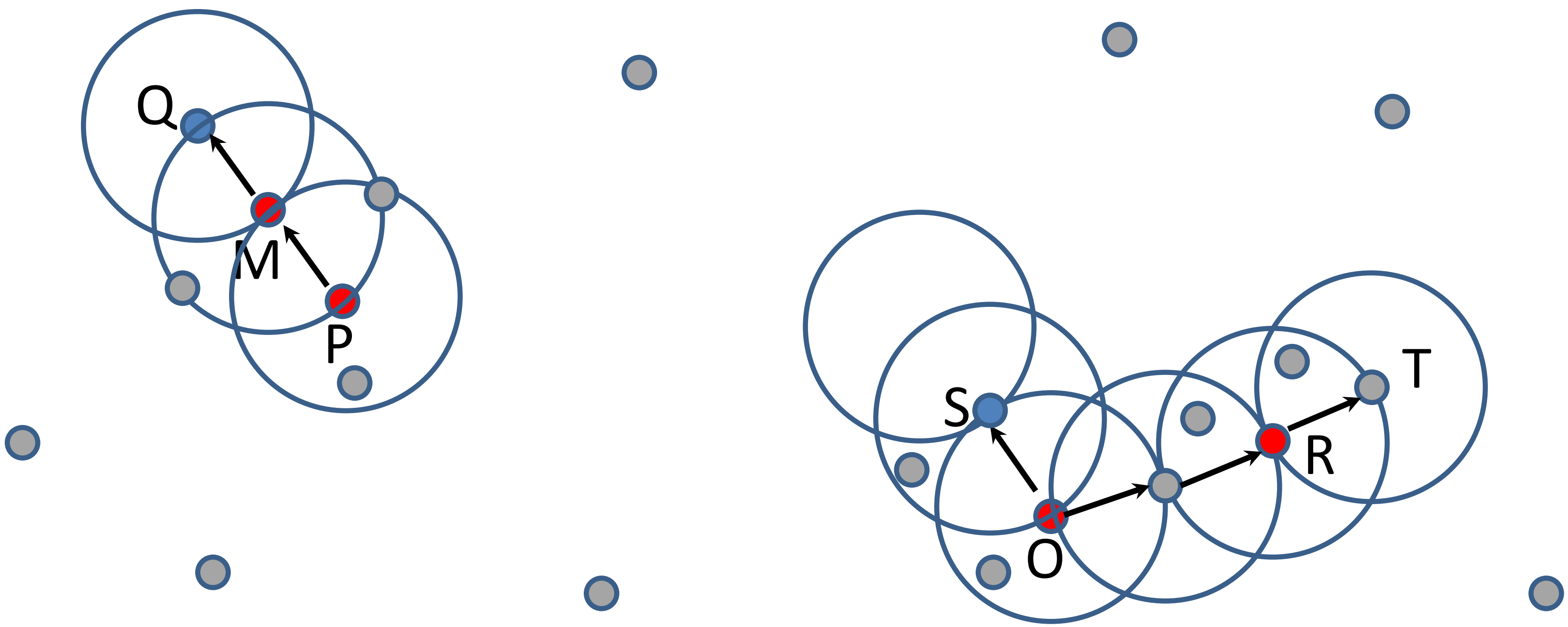
- A **border point** has fewer than *MinPts* objects in its ε-neighborhood , but is **directly reachable from some core point**
- A **noise point** is any point that is neither a core point nor a border point.

Definitions: example: MinPts=3



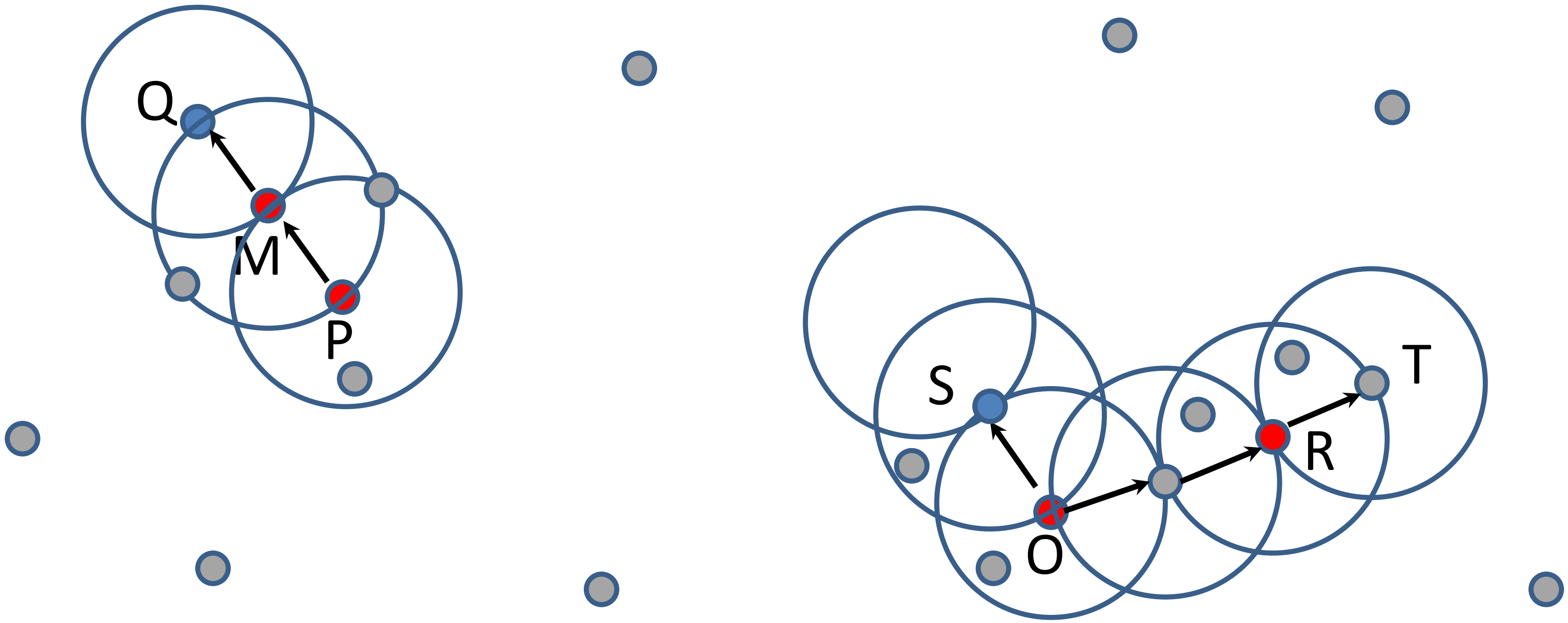
M, P, O and R are core points, since each contains at least 3 points in its ε-neighborhood

Definitions: example: MinPts=3



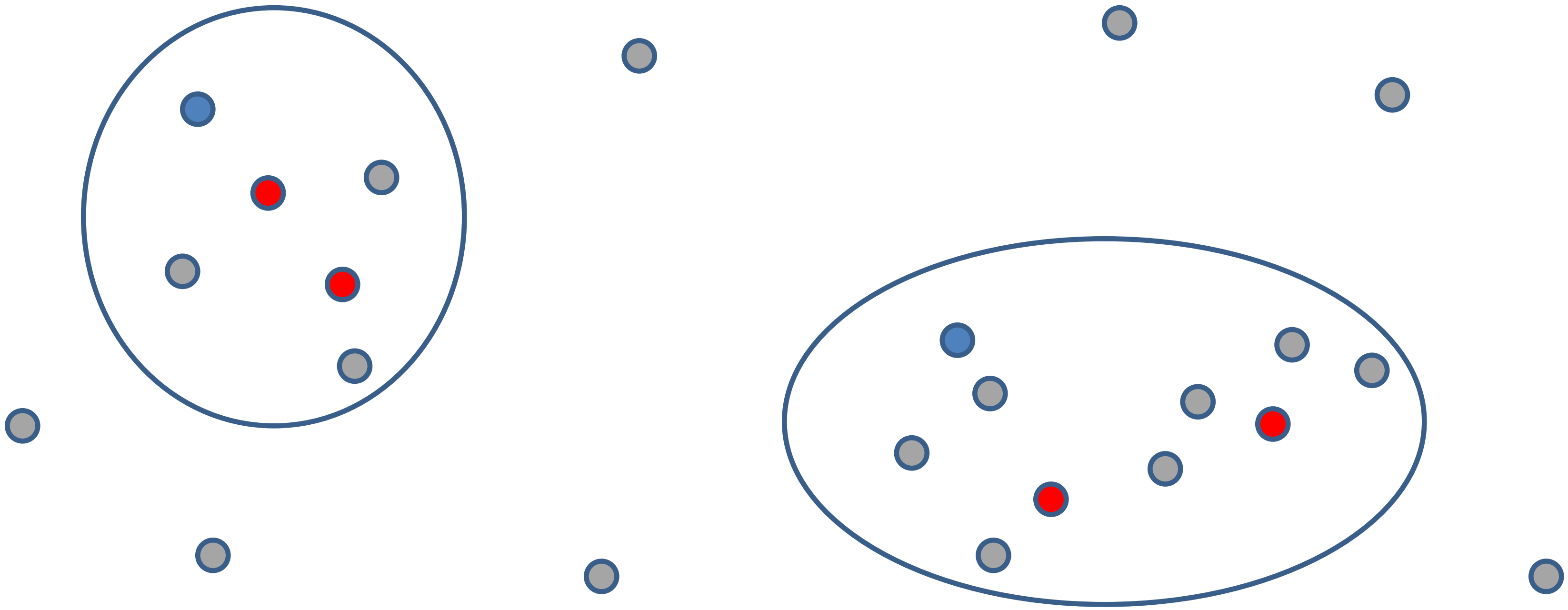
S is directly density-reachable from O, T is indirectly density-reachable from O, and T is directly density-reachable from R

Definitions: example: MinPts=3



S, O, R, T are density-connected

Density-based cluster



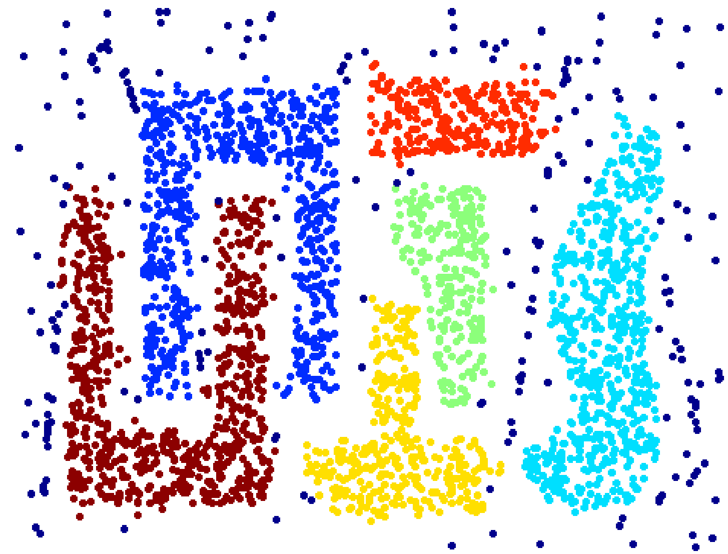
- A density-based cluster is a set of density-connected objects that is **maximal** with respects to density-reachability

DBSCAN algorithm

1. Check ε-neighborhood of each point and label each point as core, border, or noise point
2. Eliminate noise points
3. Combine all core points which are density-reachable into a single cluster
4. Assign each border point to one of the clusters of its associated core points

When DBSCAN Works Well

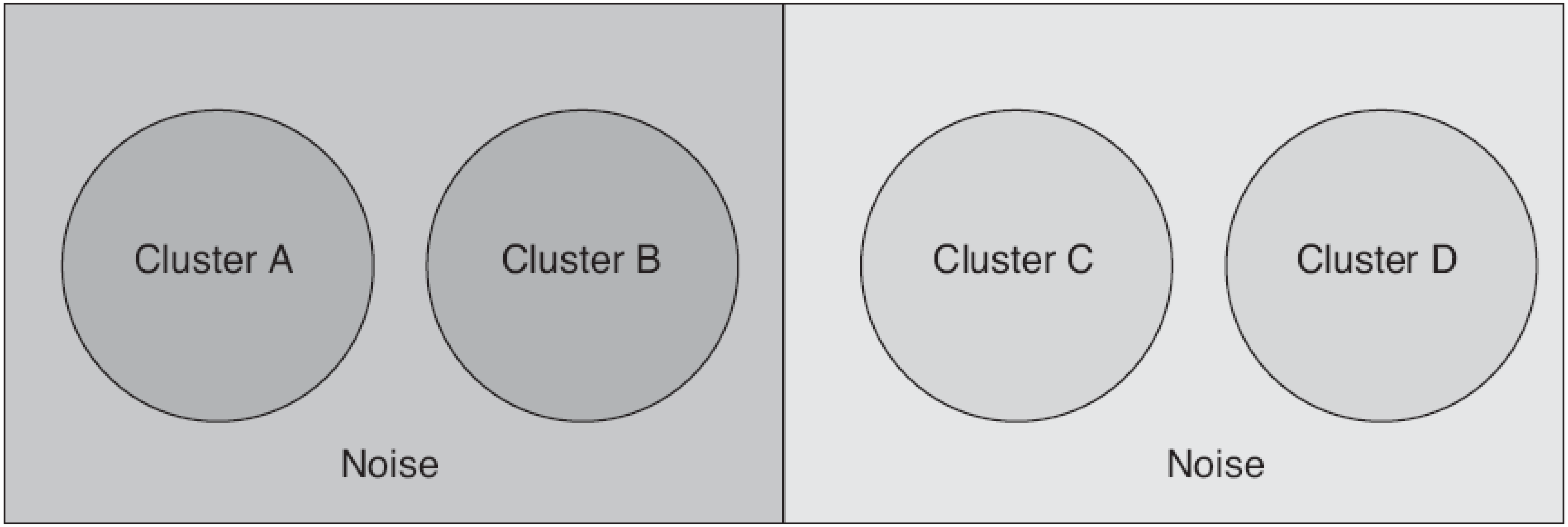
Original Points



Clusters

- **Resistant to Noise**
- **Can handle clusters of different shapes and sizes**

When DBSCAN Does NOT Work Well

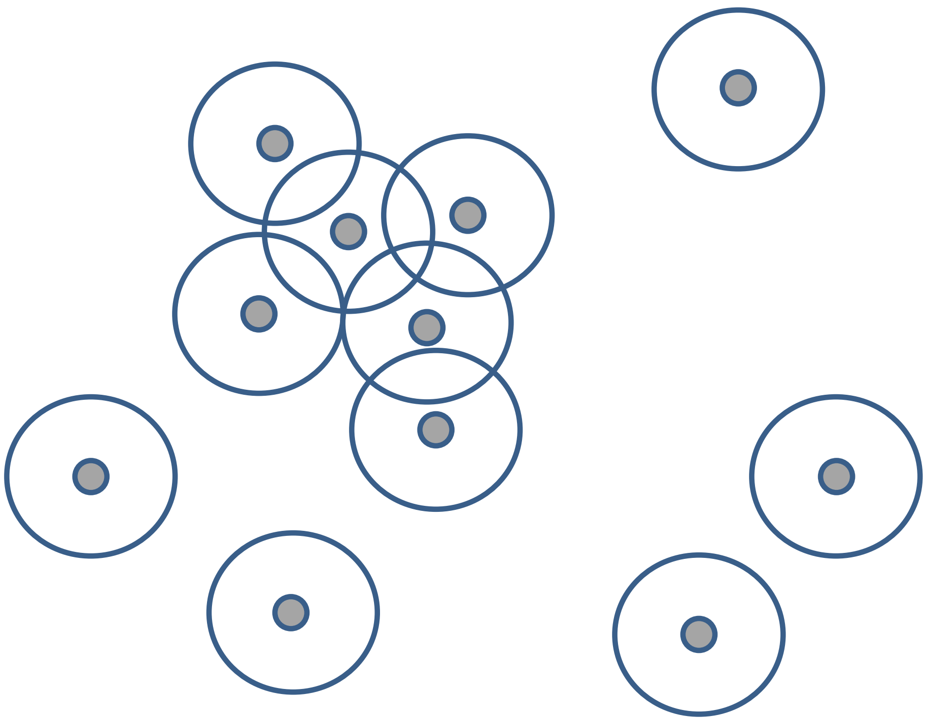


Why DBSCAN doesn’t work well here?

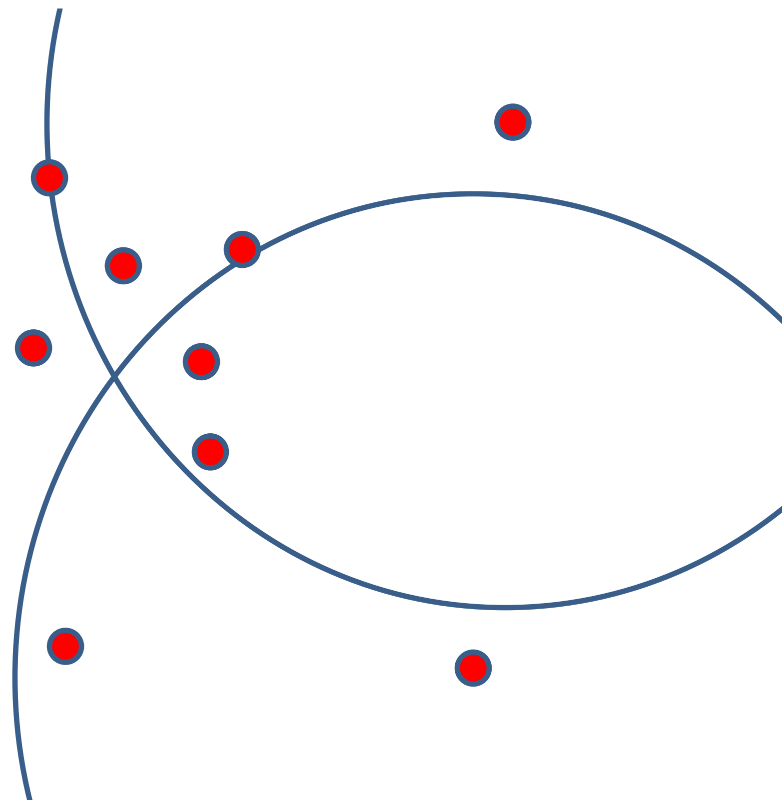
Selecting ε and MinPts

- If the radius is too small, then all points are noise points
- If the radius is too large, than all points are core points

ε is too small

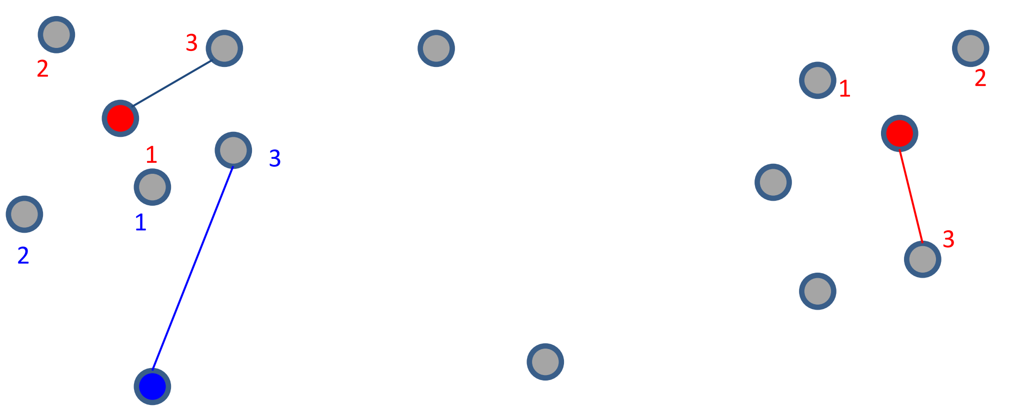


ε is too big



Selecting DBSCAN parameters: 1/2

- Decide how many points you want in a dense region: MinPts. Suppose we want core points to have at least *k* ε-neighbors
- Determine the distance from each point to its *k*-th nearest neighbor, called the *k*dist.
- For points that belong to some cluster, the value of *k*dist will be small [if *k* is not larger than the cluster size].
- However, for points that are not in a cluster, such as noise points, the *k*dist will be relatively large.



Example of k-distance for *k*=3: the third nearest neighbor

What does *k*dist represent?

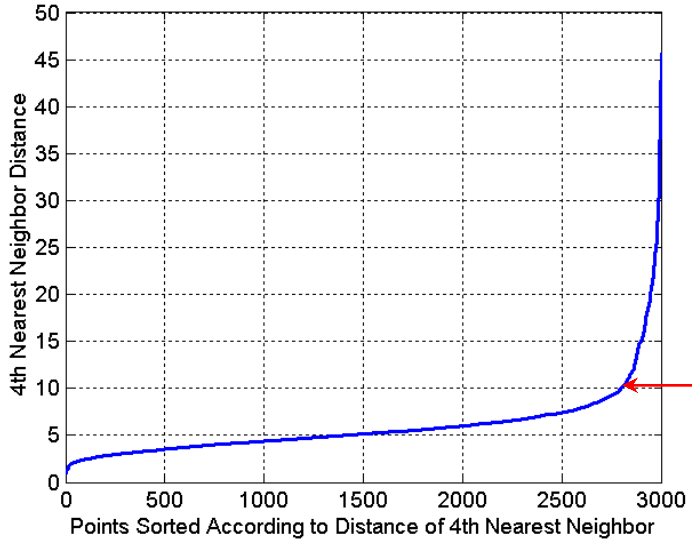
Selecting DBSCAN parameters: 2/2

- So, if we compute the *k*dist for all the data points for some *k*, sort them in increasing order, and then plot the sorted values, we expect to see a **sharp change** at the value of *k*dist that corresponds to a suitable value of ε.

- If we select this dividing distance as the ε parameter and take the value of *k* as the MinPts parameter, then points for which *k*dist is less than ε will be labeled as core points, while other points will be labeled as noise or border points.

- If there is no sharp change in distance then
  - the entire dataset is a noise, or
  - change value of *k*

DBSCAN: Determining ε and MinPts



Use radius 10 to separate clusters from noise

- ε determined in this way depends on *k*, but does not change dramatically as *k* changes.
- If *k* is too small ?

  then even a small number of closely spaced points that are noise or outliers will be incorrectly labeled as clusters.
- If *k* is too large ?

  then small clusters (of size less than *k*) are likely to be labeled as noise.
- Original DBSCAN used *k* = 4, which appears to be a reasonable value for most data sets.