# DBSCAN – Density-Based Spatial Clustering of Applications with Noise

Lecture 13-1



## DBSCAN

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)

- DBSCAN is a density-based algorithm.
  - 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

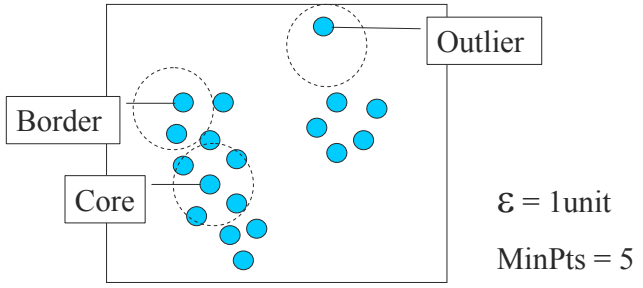


## DBSCAN

- 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

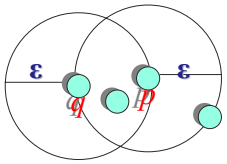


## Border & Core



# Concepts: ε-Neighborhood

- **ε-Neighborhood** - Objects within a radius of ε from an object. (epsilon-neighborhood)
- **Core objects** - ε-Neighborhood of an object contains at least **MinPts** of objects

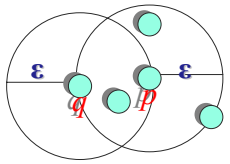



ε-Neighborhood of *p*
ε-Neighborhood of *q*
*p* is a core object (MinPts = 4)
*q* is not a core object



# Concepts: Reachability

- **Directly density-reachable**
  - An object q is directly density-reachable from object p if q is within the ε-Neighborhood of p and p is a core object.

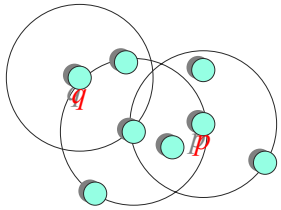



  - q is directly density-reachable from p
  - p is not directly density-reachable from q?



# Concepts: Reachability

- **Density-reachable:**
  - An object *p* is density-reachable from *q* w.r.t ε and *MinPts* if there is a chain of objects $p_1,\ldots,p_n$, with $p_1=q$, $p_n=p$ such that $p_{i+1}$is directly density-reachable from $p_i$ w.r.t ε and *MinPts* for all 1 <= *i* <= *n*

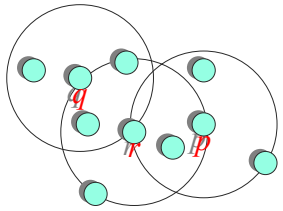



  - *q* is density-reachable from *p*
  - *p* is not density- reachable from *q*?
  - Transitive closure of direct density-Reachability, asymmetric



# Concepts: Connectivity

- **Density-connectivity**
  - Object *p* is density-connected to object *q* w.r.t ε and *MinPts* if there is an object *o* such that both *p* and *q* are density-reachable from *o* w.r.t ε and *MinPts*



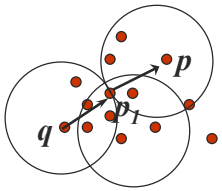

  - *P* and *q* are density-connected to each other by *r*
  - Density-connectivity is symmetric

# Concepts: cluster & noise

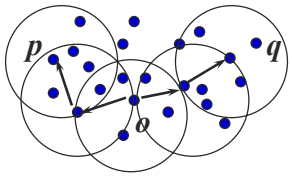
- **Cluster**: a cluster ***C*** in a set of objects ***D*** w.r.t ε and *MinPts* is a non empty subset of ***D*** satisfying
  - Maximality: For all *p*, *q* if *p* ∈ ***C*** and if *q* is density-reachable from *p* w.r.t ε and *MinPts*, then also *q* ∈ ***C.***
  - Connectivity: for all *p*, *q* ∈ ***C***, *p* is density-connected to *q* w.r.t ε and *MinPts* in ***D.***
  - ***Note:*** cluster contains *core objects* as well as *border objects*
- **Noise:** objects which are not directly density-reachable from at least one core object.



# (Indirectly) Density-reachable:




Density-connected





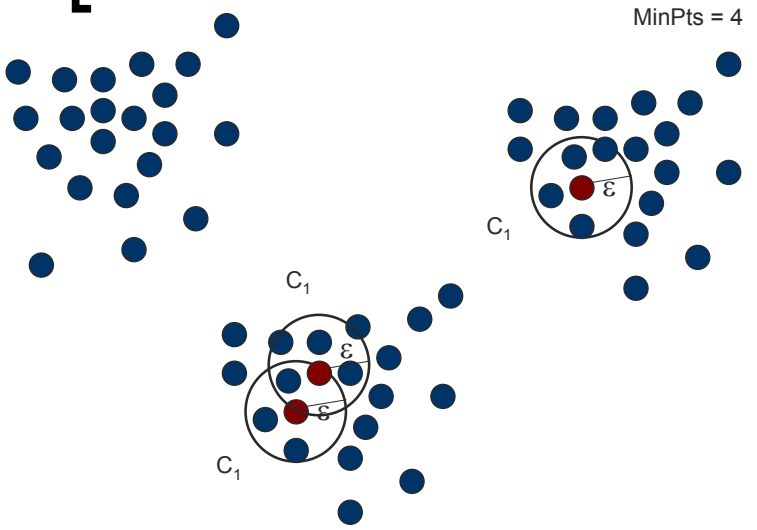

# DBSCAN: The Algorithm

- select a point ***p***
- Retrieve all points density-reachable from ***p*** wrt ε and ***MinPts***.
- If ***p*** is a core point, a cluster is formed.
- If ***p*** is a border point, no points are density-reachable from ***p*** and DBSCAN visits the next point of the database.
- Continue the process until all of the points have been processed.

Result is independent of the order of processing the points

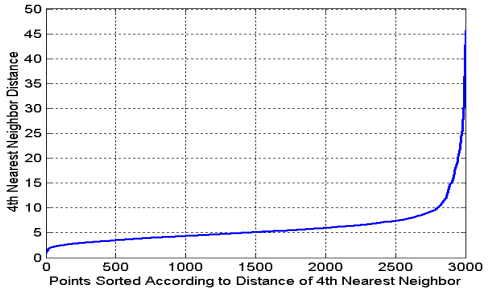


# An Example



# DBSCAN: Determining EPS and MinPts

- Idea is that for points in a cluster, their $k^{th}$ nearest neighbors are at roughly the same distance
- Noise points have the $k^{th}$ nearest neighbor at farther distance
- So, plot sorted distance of every point to its $k^{th}$ nearest neighbor






# DBSCAN: Determining EPS and MinPts

- Distance from a point to its *k^th^* nearest neighbor=>k-dist
- For points that belong to some clusters, the value of k-dist will be small if k is not larger than cluster size
- For points that are not in a cluster such as noise points, the k-dist will be relatively large
- Compute k-dist for all points for some k
- Sort them in increasing order and plot sorted values
- A sharp change at the value of k-dist that corresponds to suitable value of eps and the value of k as MinPts



# DBSCAN: Determining EPS and MinPts

- A sharp change at the value of k-dist that corresponds to suitable value of eps and the value of k as MinPts
  - Points for which k-dist is less than eps will be labeled as core points while other points will be labeled as noise or border points.
- If k is too large=> small clusters (of size less than k) are likely to be labeled as noise
- If k is too small=> Even a small number of closely spaced that are noise or outliers will be incorrectly labeled as clusters