Chapter 7. Cluster Analysis

Dong-Kyu Chae

PI of the Data Intelligence Lab @HYU
Department of Computer Science & Data Science
Hanyang University

Data
Intelligence
LAB

Contents

1. What is Cluster Analysis?
2. Categories & Basic Concepts of Clustering
3. Partitioning Methods
4. Hierarchical Methods
5. Integration of Hierarchical & Distance-based Clustering
6. Density-Based Methods
7. Summary

CHAMELEON

- **Main idea**
  - Two clusters can be merged only if the interconnectivity and closeness (proximity) between two clusters are high
    - **Relative** to the internal interconnectivity of the clusters and internal closeness of items within the clusters

1. Draw a *k*-nearest neighbor graph (KNN graph) first

- **Node**: object, **edge**: k-nearest neighbor's link, **weight**: similarity

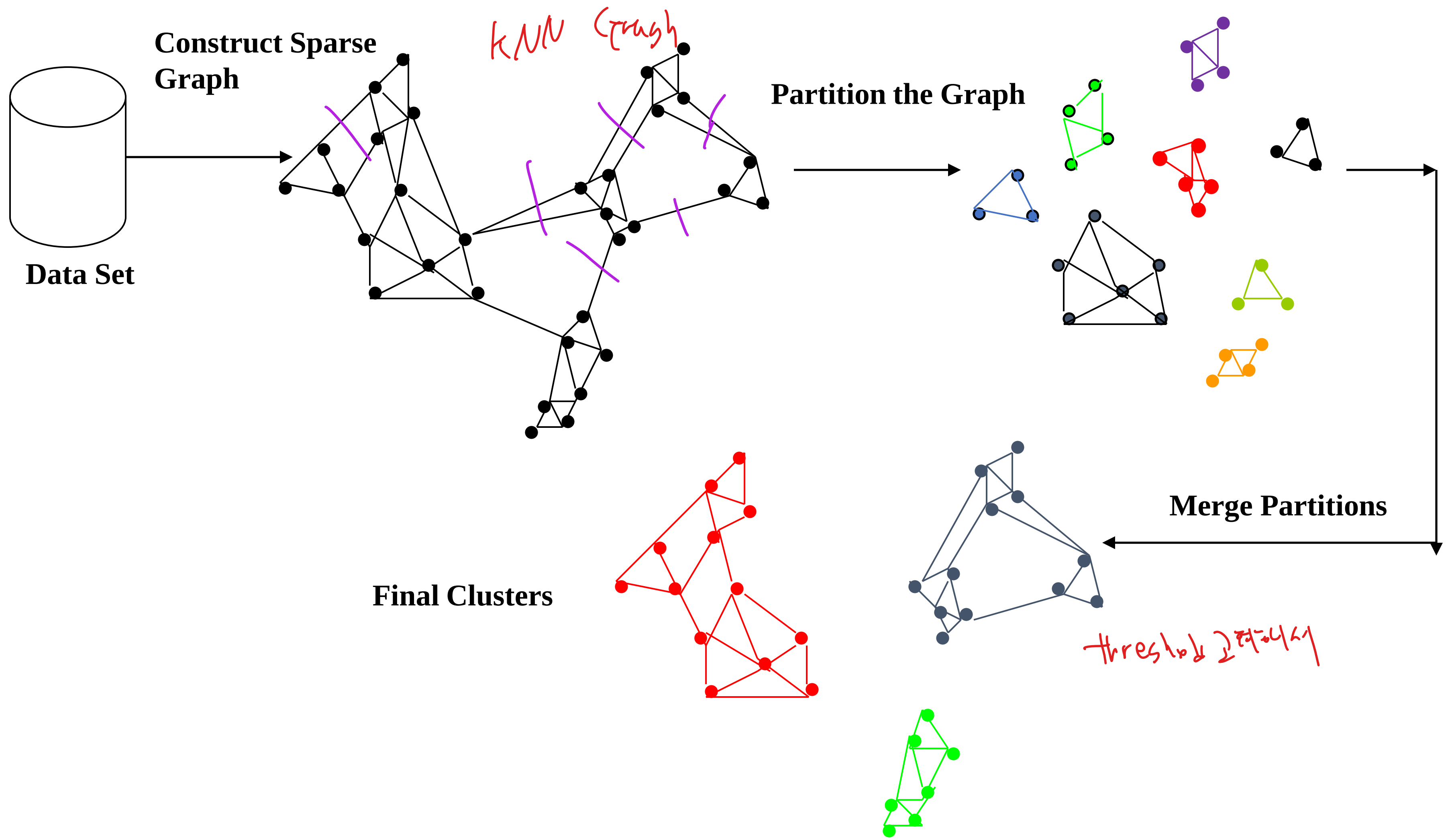
2. Partition: Use a graph partitioning algorithm

- Divide the KNN graph into a large number of relatively small sub-clusters

3. Merge: Use an agglomerative hierarchical clustering algorithm

- Iteratively find clusters by repeatedly combining these sub-clusters

Overall Framework of CHAMELEON



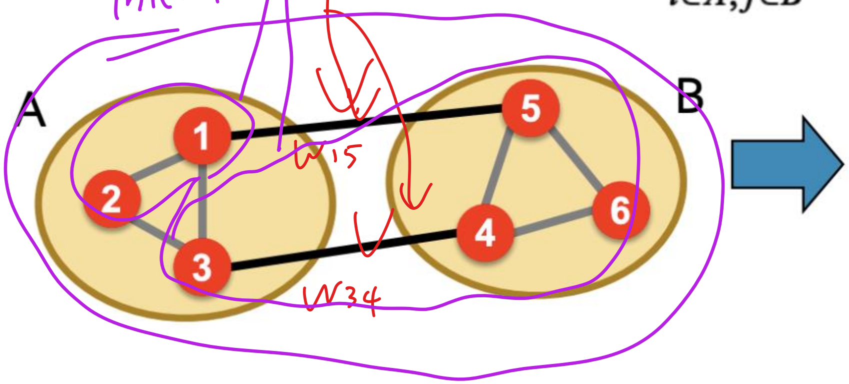
CHAMELEON: Partitioning

- **Partition the KNN graph such that the edge cut is minimized.**
  - The edge-cut of a partition is **the sum of the weights** of edges whose **vertices lie in different partitions**.

이경: cut(A,B) = 3

$$cut(A,B) = \sum_{i \in A, j \in B} w_{ij}$$

initial



$cut(A,B) = 2$

- hMeTiS library (**METIS**) is used
  - Tries to split a graph into two subgraphs of nearly equal sizes

not only edge cut is minimized
but also balance number of data points inside partition

CHAMELEON: Merging

- **Merging the partitions**
  - This step computes the cluster similarity based on the **relative inter-connectivity** and **relative closeness** of the clusters.

- **Relative inter-connectivity**

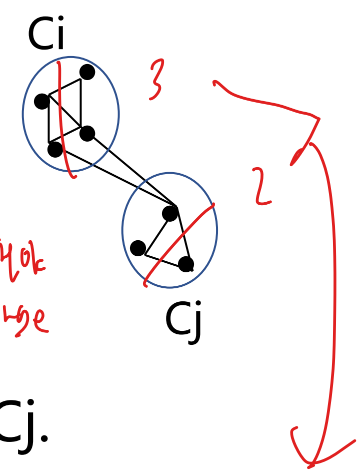
criteria

set of edges

$$RI(C_i, C_j) = \frac{|EC_{\{C_i, C_j\}}|}{\frac{1}{2}(|EC_{C_i}| + |EC_{C_j}|)},$$

standard

두 값이 비슷해야 merge



  - $EC_{\{Ci,\ Cj\}}$ = edges that connect Ci and Cj.
  - EC_{Ci} = edges that partition the cluster into roughly equal parts.

CHAMELEON: Merging

- **Relative closeness**

0.6

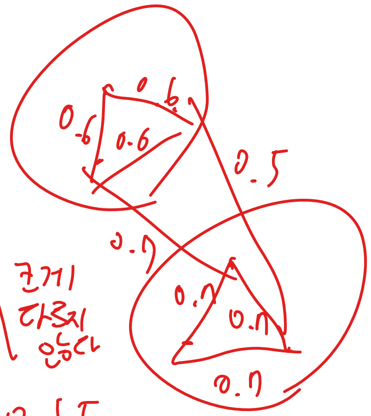
$$RC(C_i, C_j) = \frac{\bar{S}_{EC_{\{C_i, C_j\}}}}{\frac{|C_i|}{|C_i|+|C_j|}\bar{S}_{EC_{C_i}} + \frac{|C_j|}{|C_i|+|C_j|}\bar{S}_{EC_{C_j}}},$$

number of data points

$\frac{3}{3\ 3}$ 0.6 $\frac{3}{3\ 3}$ 0.7

크기 다르지 않다

= 0.65



weight

- $\bar{S}_{EC_{\{Ci, Cj\}}}$ = average weight of the edges from Ci to Cj
- $\bar{S}_{EC_{Ci}}$ = average of the weights of the edges in the cluster.

- **Merging**
  - So far, we have got **Relative Inter-Connectivity** and **Relative Closeness**
  - Using them:

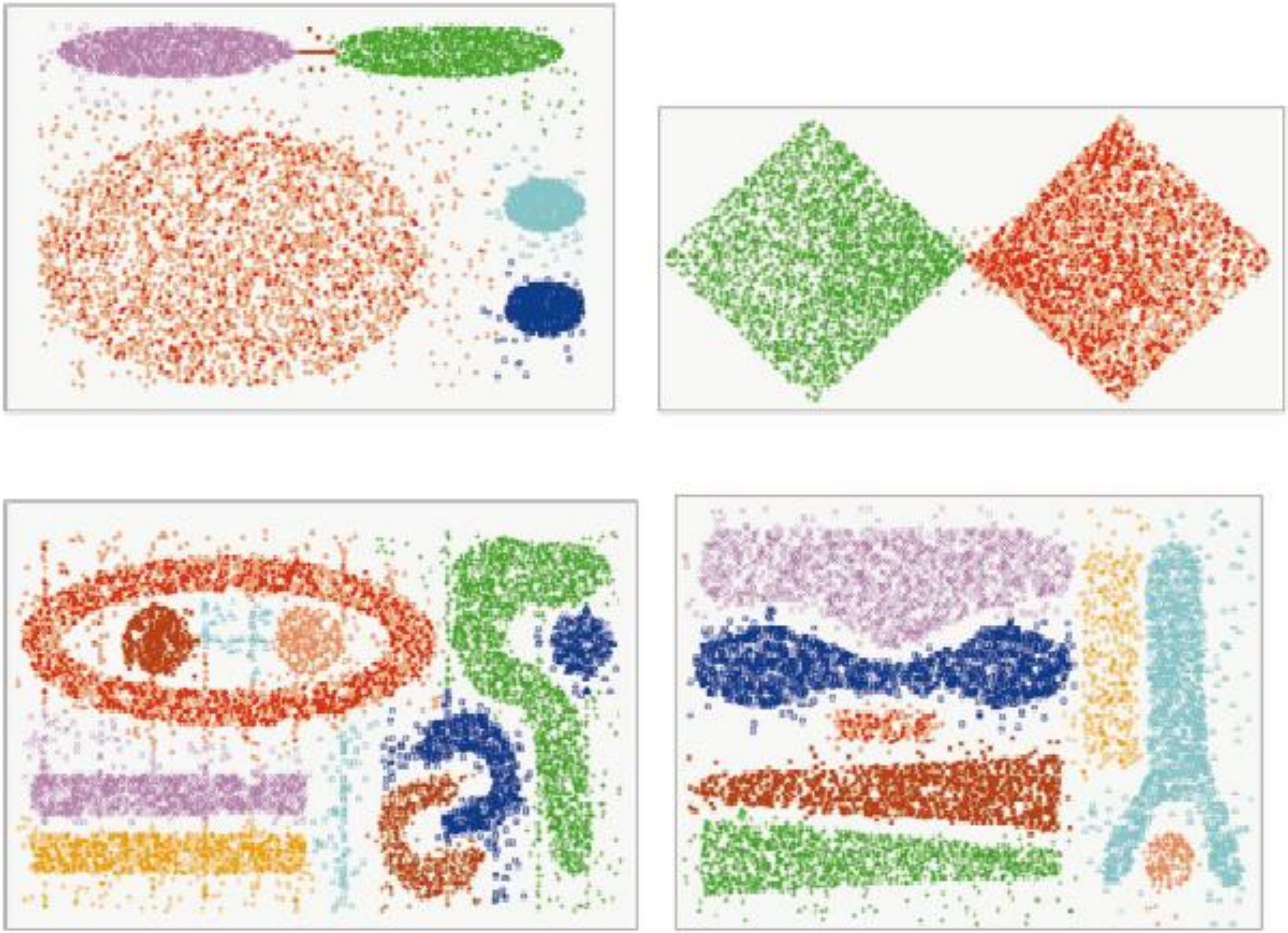
$$RI(C_i, C_j) * RC(C_i, C_j)^{\alpha}$$

1이면 같은 비중 1보다 크면 RC가 더 큰 비중

← threshold 보다 크면 merge

  - where alpha controls the importance of RC

CHAMELEON (Clustering Complex Objects)

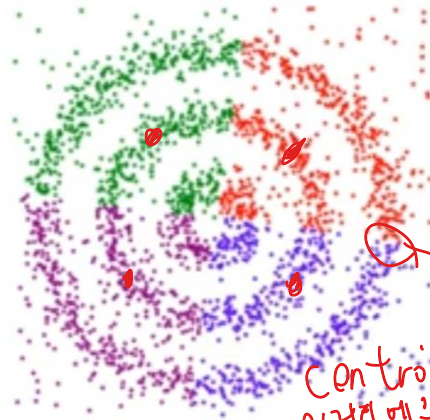


Contents

1. What is Cluster Analysis?
2. Categories & Basic Concepts of Clustering
3. Partitioning Methods
4. Hierarchical Methods
5. Integration of Hierarchical & Distance-based Clustering
6. Density-Based Methods
7. Summary

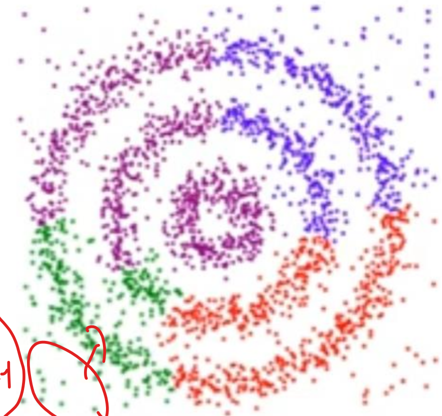
Why Density-based Clustering?

DATASET

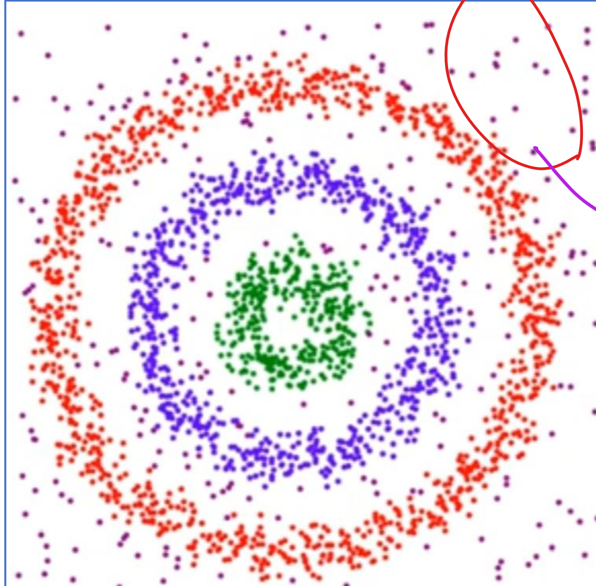


K-MEANS

centroid와의 거리에 의해서
가까워도
다른 cluster에 있다



Hierarchical
Clustering



DBSCAN

handle noise

→ not the member of cluster

does not depend on distance

different shapes 가능

Density-Based Clustering Methods

- **Clustering based on density (local cluster criterion), such as density-connected points, rather than just a distance**
- **Major features:**
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan, thus being efficient
  - Need density parameters as termination condition

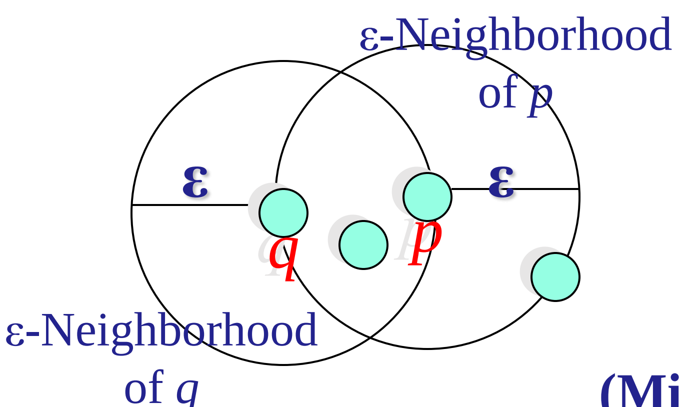
    programmer가 optimal한 hyper parameter 정해야 함
- **Several interesting studies:**
  - **DBSCAN**
  - **OPTICS**
  - CLIQUE

Density-Based Clustering: Hyper-Parameters

- **Two parameters:**
  - ε: radius for the neighborhood of any point p:

    $N_\varepsilon(p)$:= {any q in dataset D | $dist(p, q) \leq \varepsilon$}

    - ε-Neighborhood – Objects within a radius of ε from an object.
  - *MinPts* : minimum number of points in the given neighborhood
    - "High density" : ε-Neighborhood of an object contains at least *MinPts* of objects



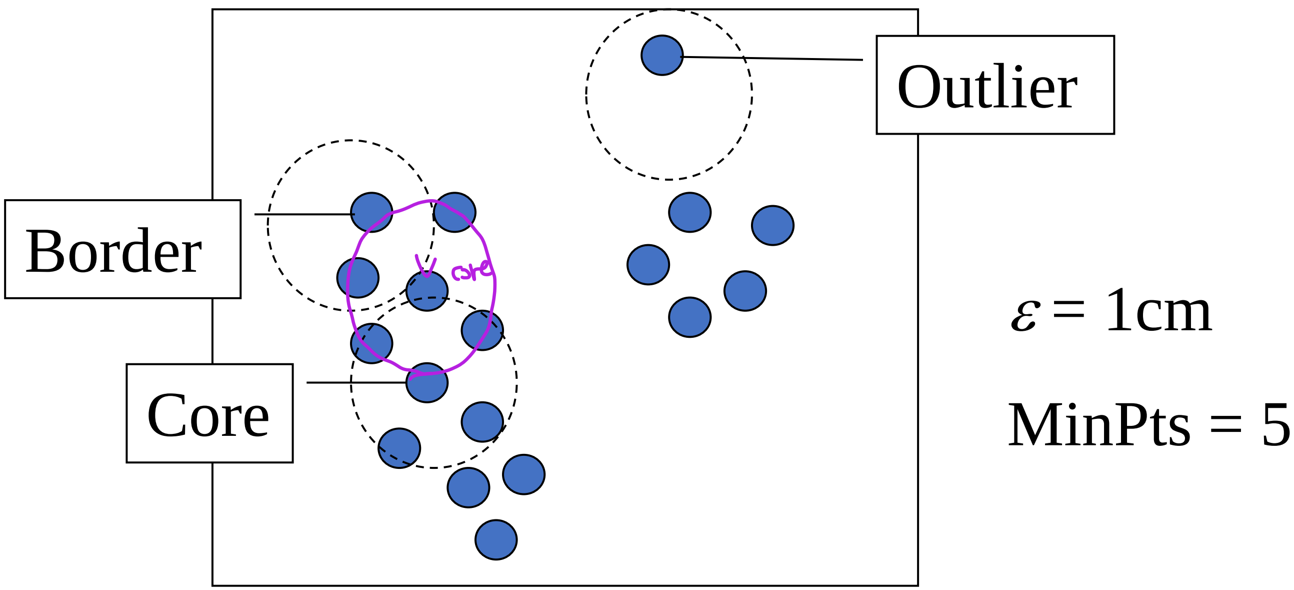
Density of p is "high"

Density of q is "low"

(MinPts = 4)

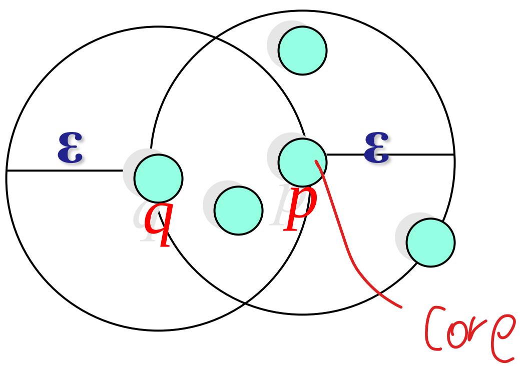
Density-Based Clustering: Types of Points

- **A point is a core point** if it has points more than *MinPts* within ε
- **A border point** has fewer than MinPts within ε , but is in the neighborhood of a core point
- **Outlier** is any point that is not a core point nor a border point. It is thus a noise, or an outlier.



Directly Density-Reachable

- **Directly density-reachable: A point *q* is directly density-reachable from a point *p* if *p* is a core object and *q* is in *p*'s ε-neighborhood.**

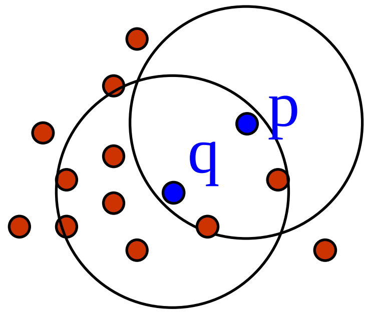


MinPts = 4

- *q* is directly density-reachable from *p*
- *p* is **NOT** directly density-reachable from *q* q가 core X
- Density-reachability is **asymmetric**. 비대칭

Directly Density-Reachable

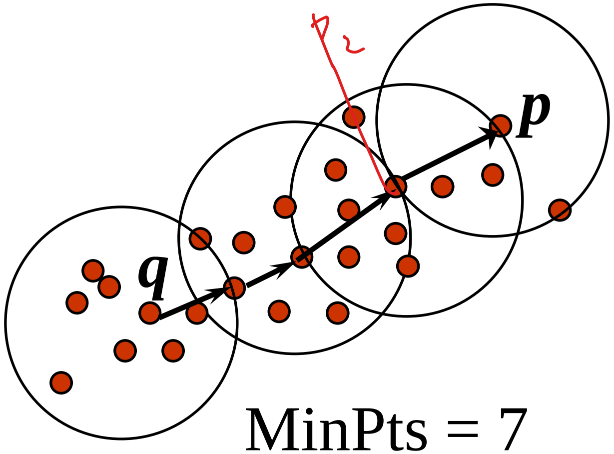
- **Directly density-reachable: A point *q* is directly density-reachable from a point *p* if *p* is a core object and *q* is in *p*'s ε-neighborhood.**
  - *p* belongs to $N_\varepsilon(q)$
  - *p* is directly density-reachable from *q*
  - *q* is NOT directly density-reachable from *p*



MinPts = 5

Density-Reachability

- **Density-Reachable (directly and indirectly):**
  - A point p is directly density-reachable from p2;
  - p2 is directly density-reachable from p1;
  - p1 is directly density-reachable from q;
  - p←p2←p1←q form a chain.
  - Then, *p* is (**indirectly**) **density-reachable** from *q*



- A point p is density-reachable from a point q if **there is a chain of points** $p_1, \ldots, p_n$, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i
- Is q ~~not~~ density-reachable from p?
  => NO

p is not core

Thank You

Data
Intelligence
Lab