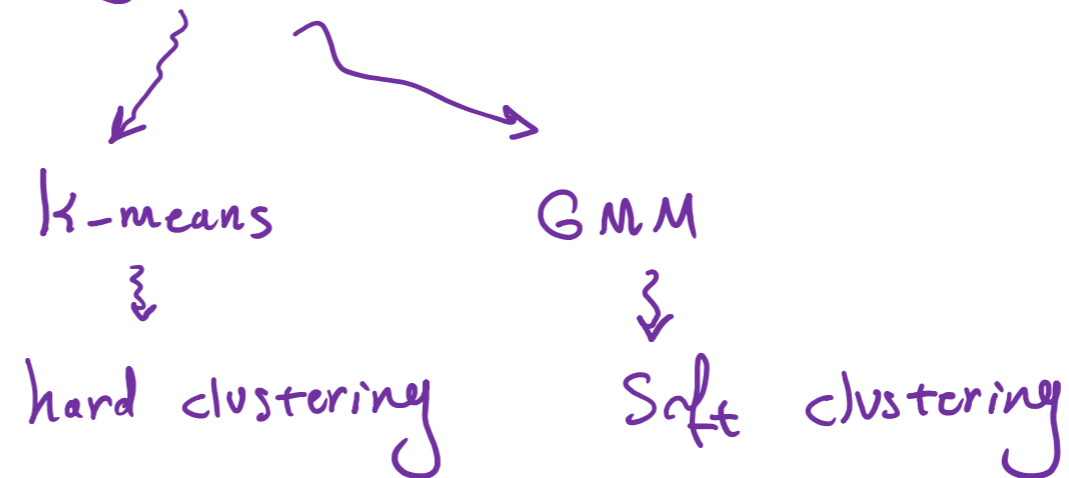


Pre-decision making clustering \leadsto We need to know the # clusters



Post-decision making clustering \leadsto Hierarchical clustering \leadsto dendrogram

\downarrow
"hard clustering"

They suffer from knowing # clusters


Density-Based Clustering

Mahdi Roozbahani
Georgia Tech

when you're a constant
and you see d/dx



Outline

- Overview 
- Basic Concepts
- The DBSCAN Algorithm
- Analysis of DBSCAN

Density-Based Clustering

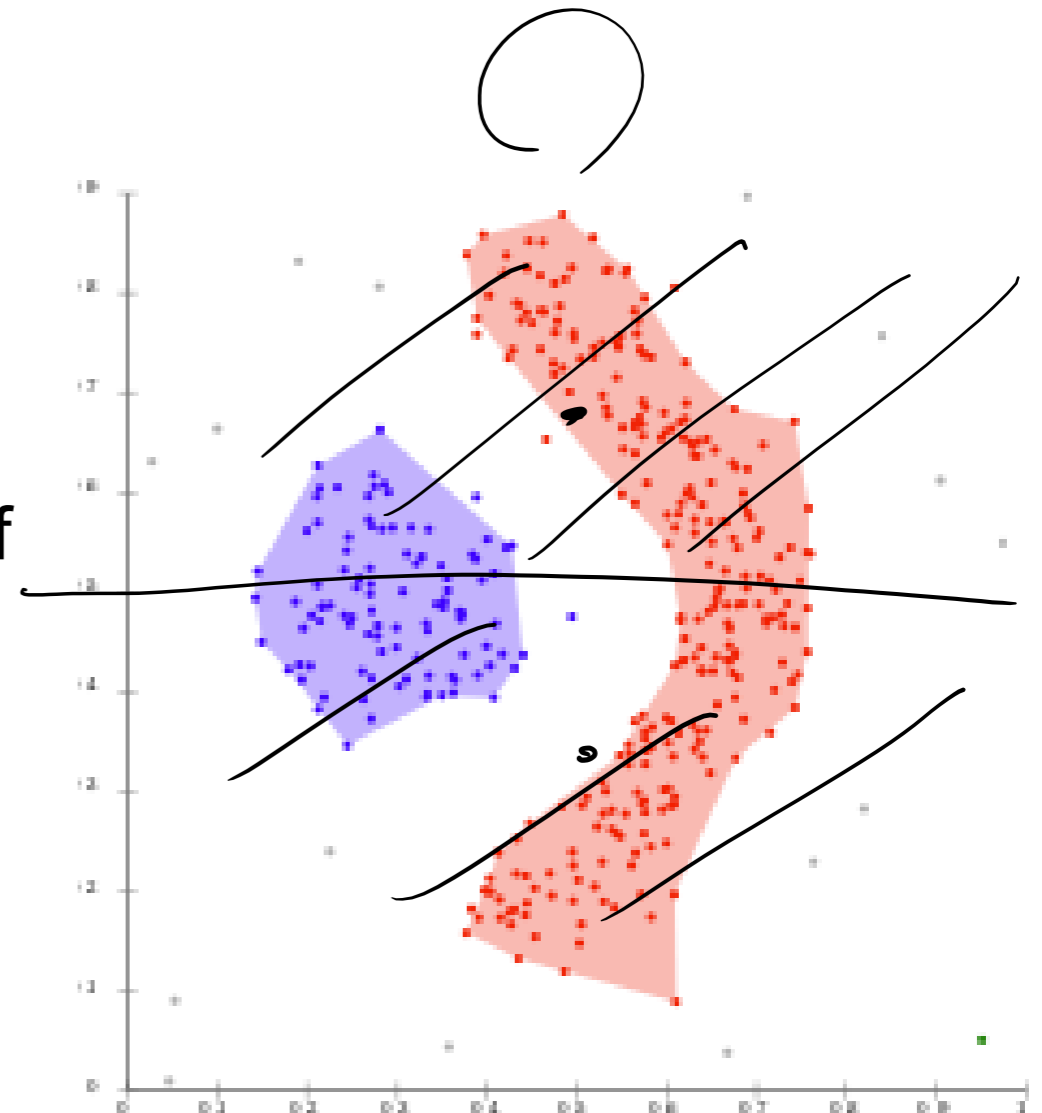


- Basic Idea


- Clusters are dense regions in the data space, separated by regions of lower density
- A cluster is defined as a maximal set of density-connected points
- Detect arbitrarily shaped clusters

- Method

- DBSCAN (Density-Based Spatial Clustering of Applications with Noise)



Outline

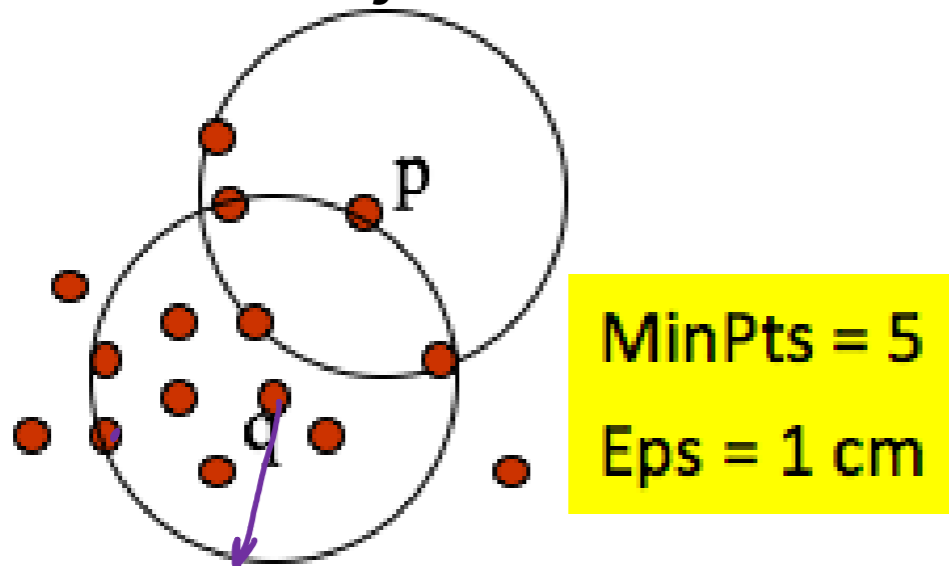
- Overview
- Basic Concepts 
- The DBSCAN Algorithm
- Analysis of DBSCAN

High Density v.s. Low Density

- Two parameters

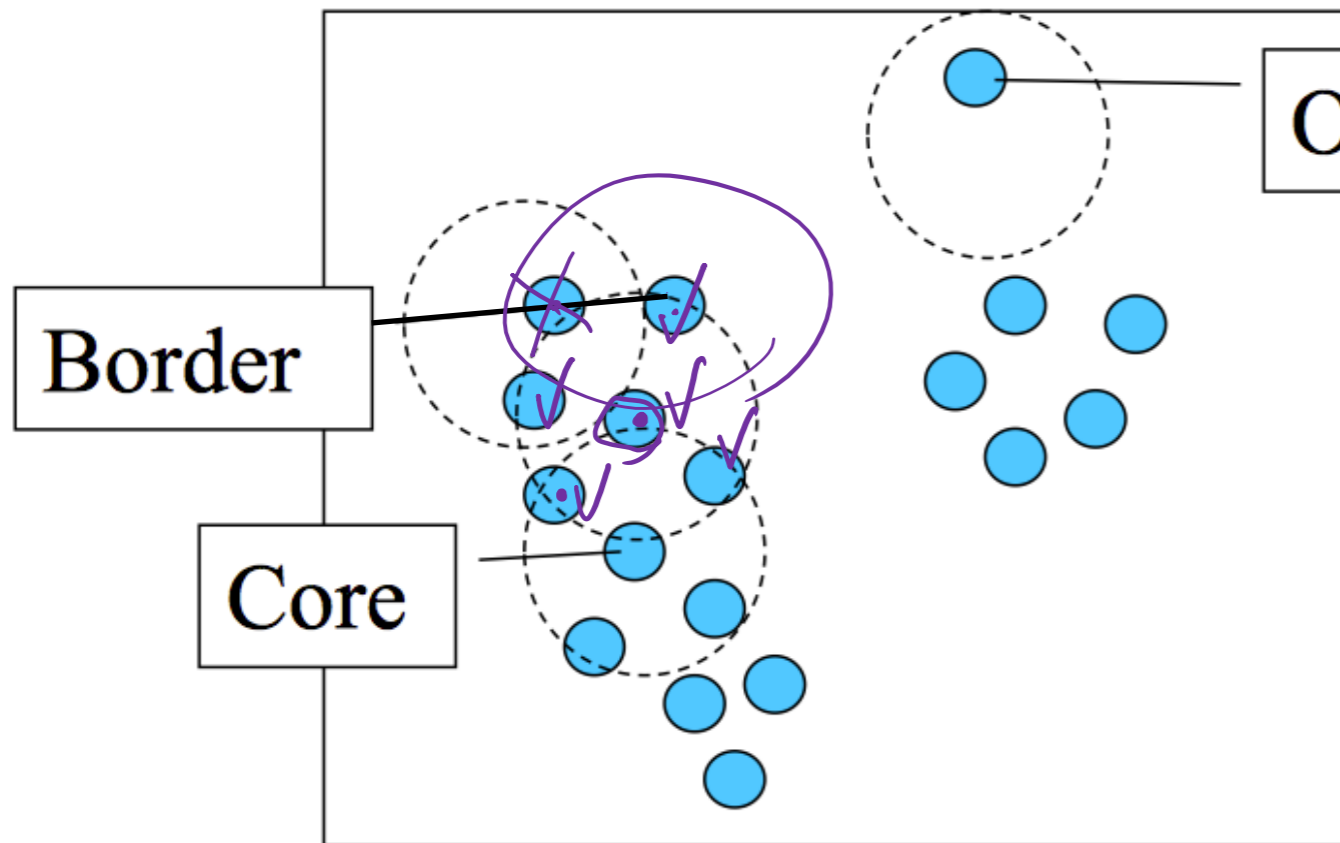
- **Eps (ϵ)**: Maximum radius of the neighborhood
- **MinPts**: Minimum number of points in the Eps-neighborhood of a point

- High density: ϵ -Neighborhood of an object contains at least MinPts of objects



Density of p is low
Density of q is high

Core Points, Border Points, and Outliers



$\epsilon = 1 \text{ unit}, \text{MinPts} = 5$

Outlier

Given ϵ and *MinPts*, categorize the objects into three exclusive groups.

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.

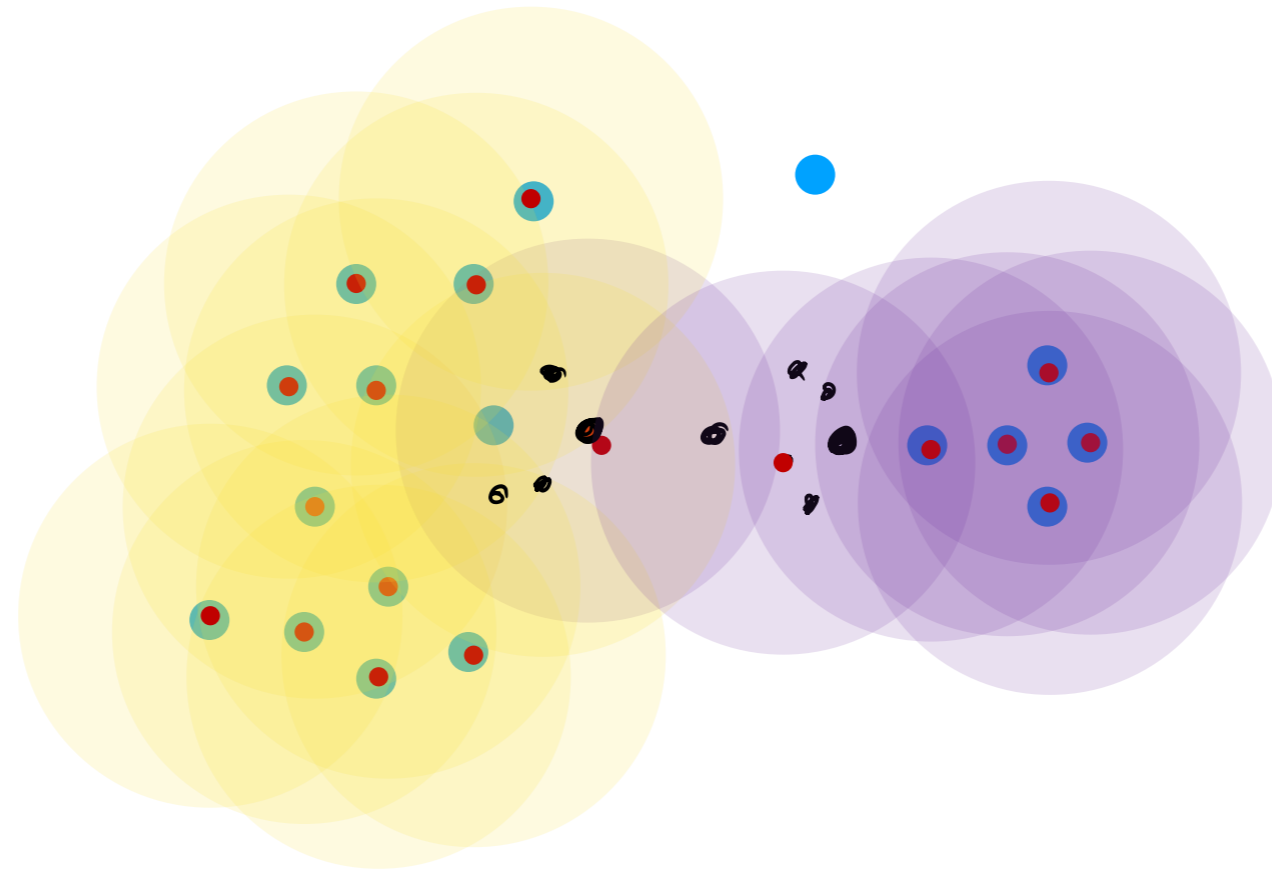
A **border point** has fewer than MinPts within Eps, but is in the neighborhood of a core point.

A **noise point** is any point that is not a core point nor a border point.

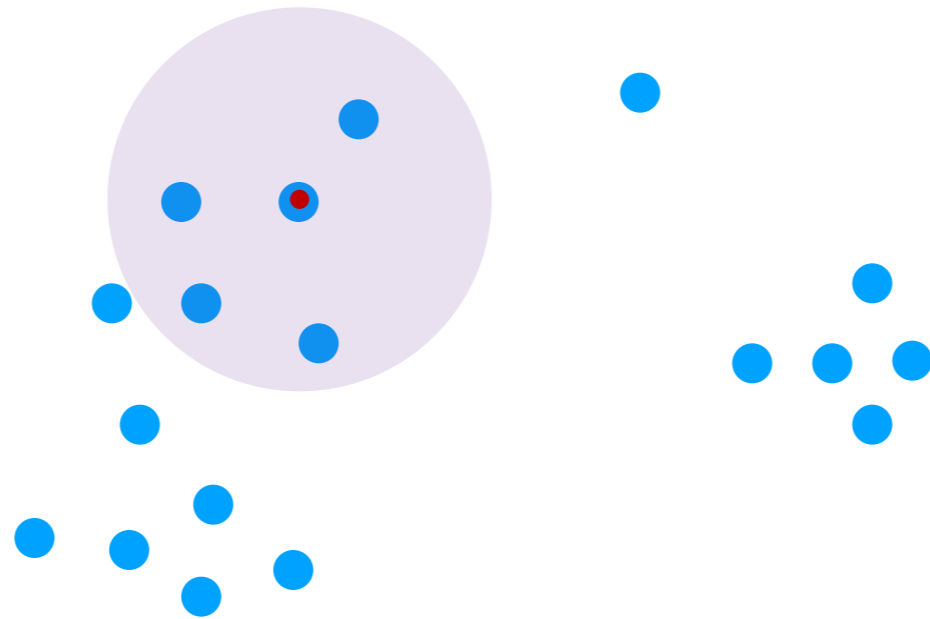
Practice:

Almost deterministic

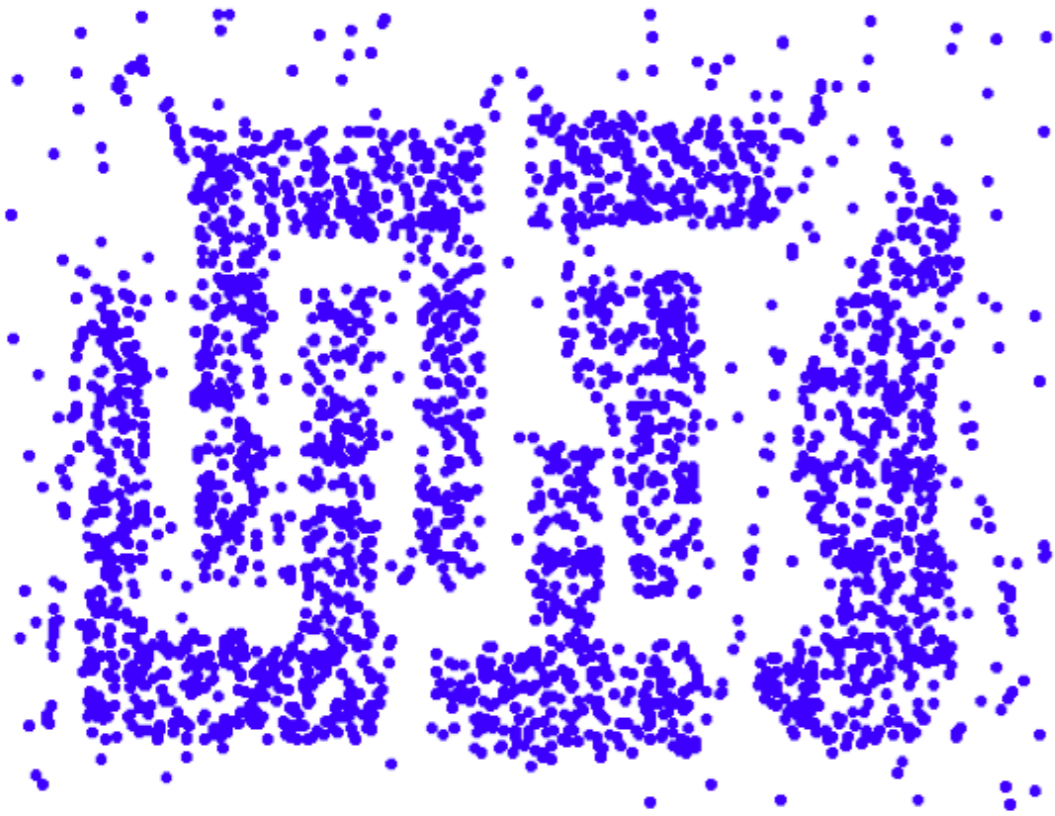
$\epsilon = 1$ unit
MinPts = 5



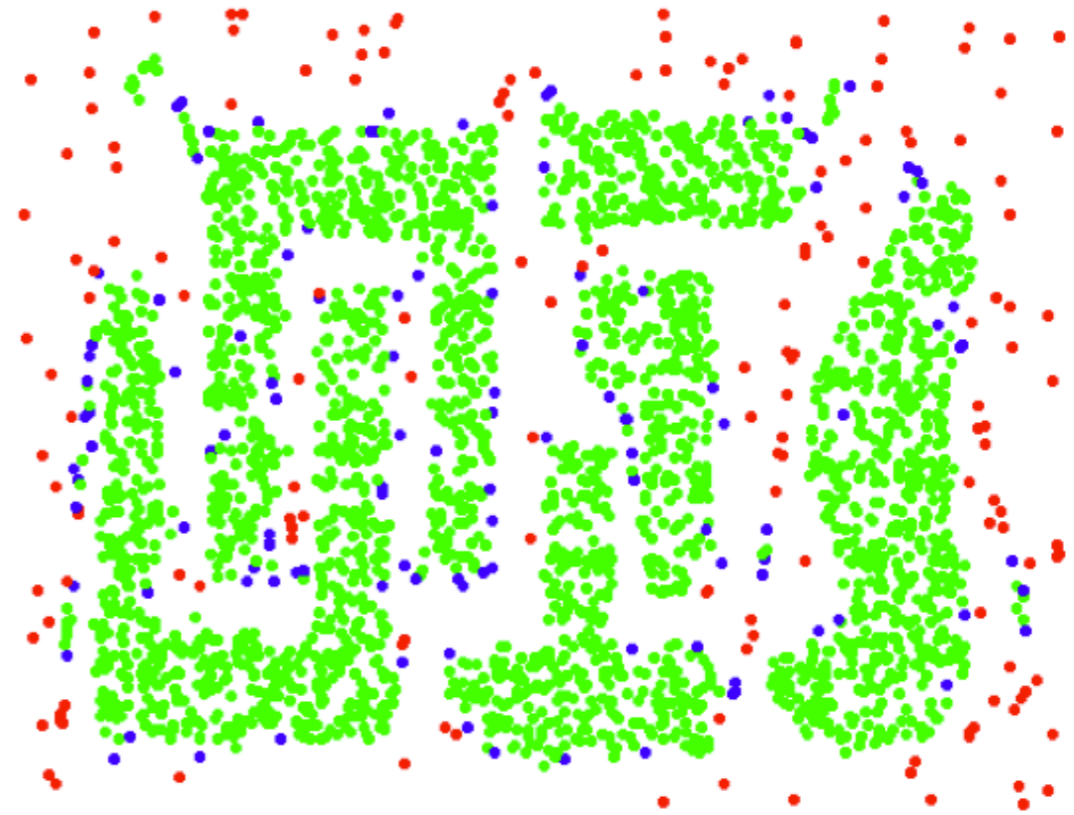
$\epsilon = 1$ unit
MinPts = 5



Examples



Original Points

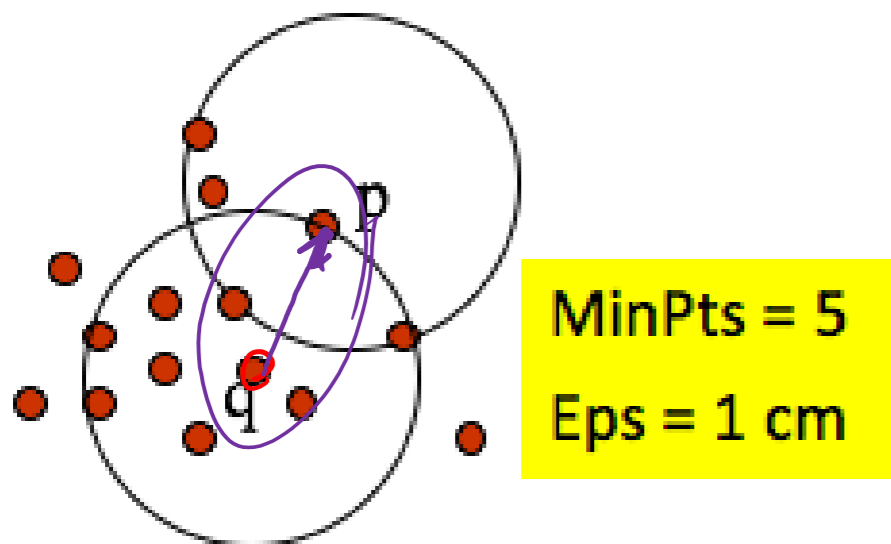


Point types: **core**,
border and **outliers**

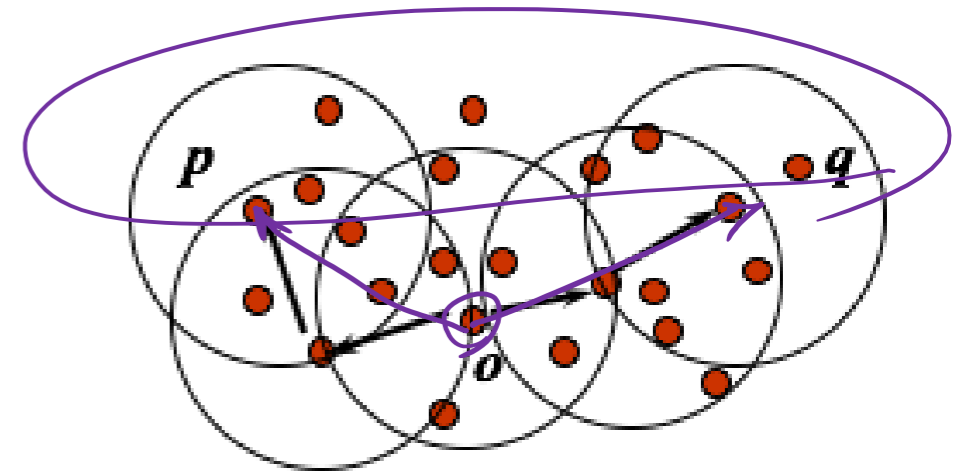
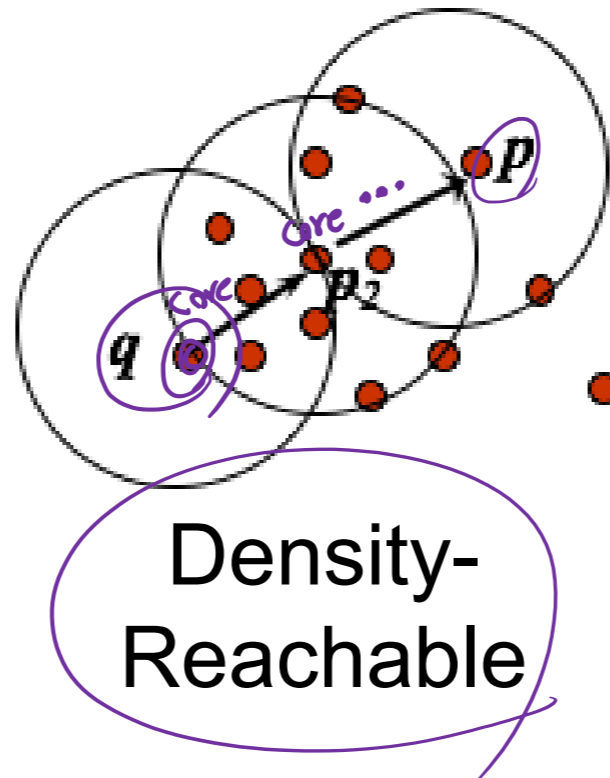
$\epsilon = 10$, MinPts = 4

Density-based related points

- Direct density reachability:
 - An object p is directly density-reachable from object q if **(1) q is a core object**; and **(2) p is in q 's ϵ -neighborhood**

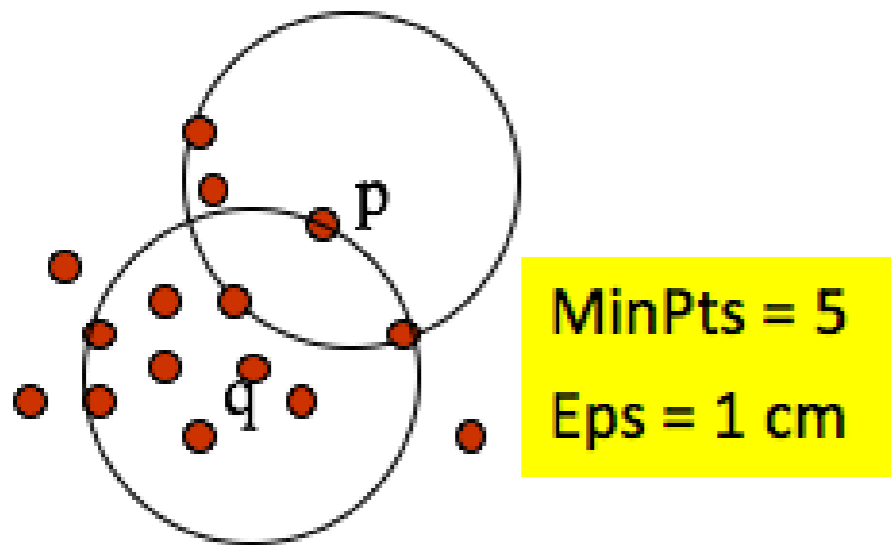


Directly Density-Reachable

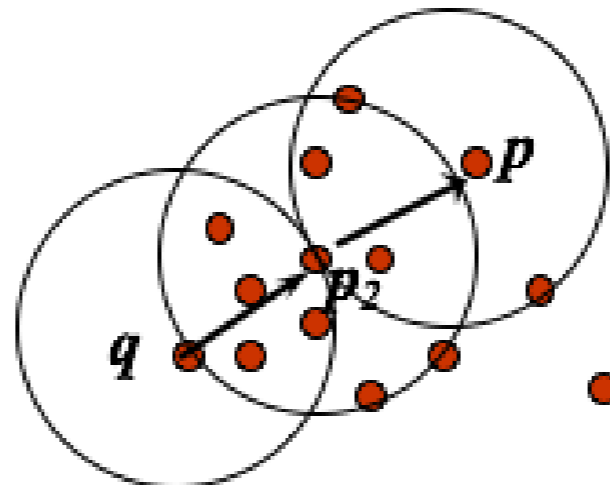


Density-based related points

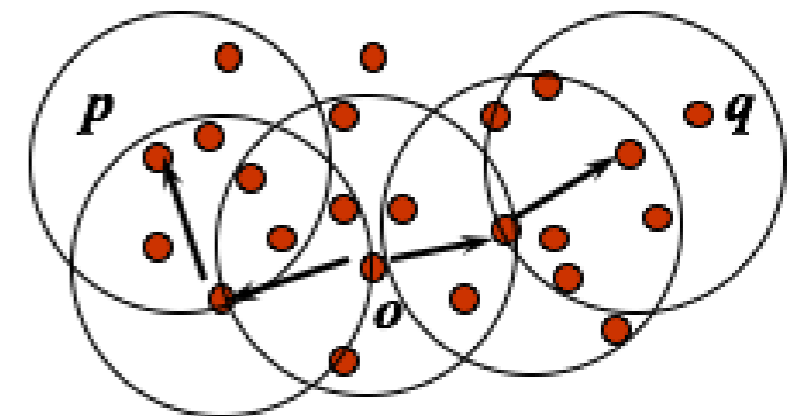
- Density reachability:
 - A point p is density-reachable from a point q if there is a chain of points $p_1, \dots, p_n, p_1 = q, p_n = p$ such that p_{i+1} is directly density-reachable from p_i
 - $p_1 = q \rightarrow p_2 \rightarrow \dots \rightarrow p_n = p$



Directly Density-Reachable



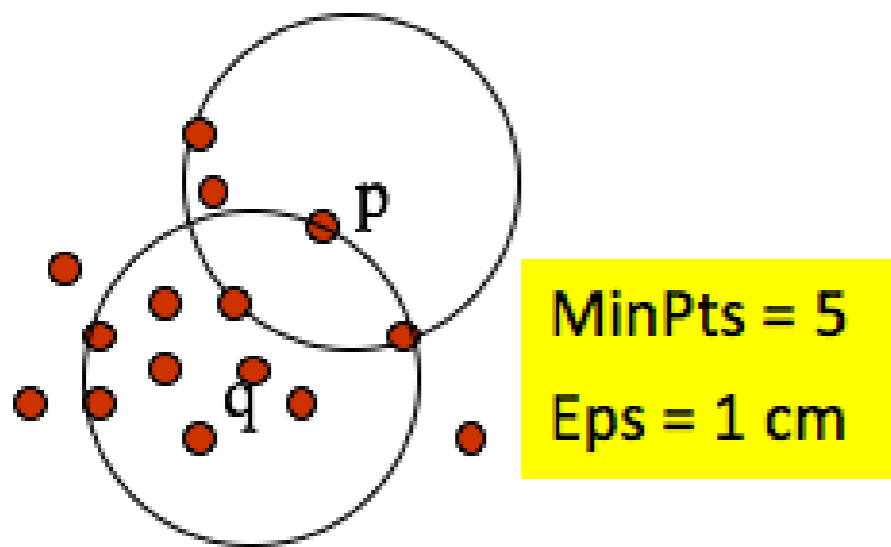
Density-Reachable



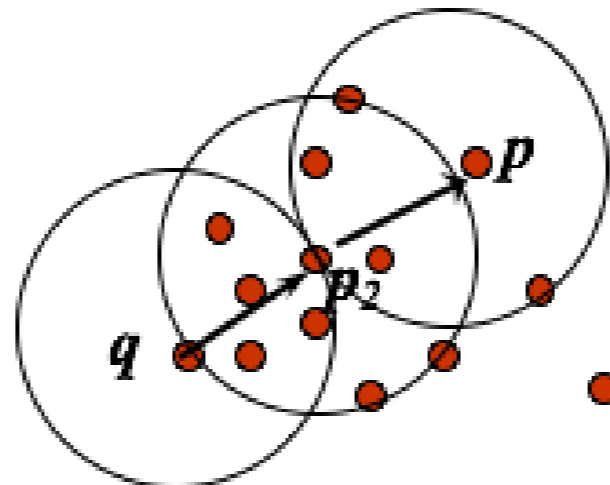
Density-Connected

Density-based related points

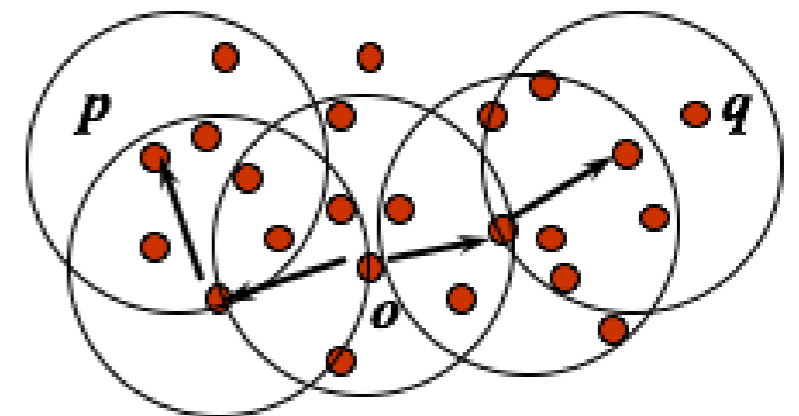
- Density connectivity:
 - A point p is density-connected to a point q if there is a point o such that both p and q are density-reachable from o



Directly Density-Reachable

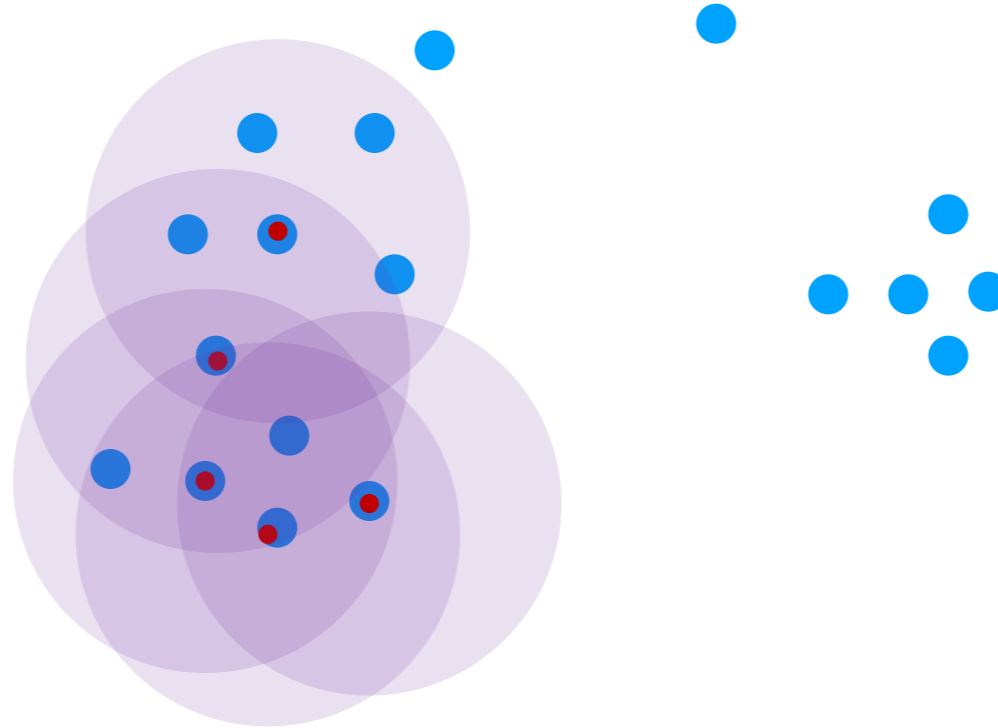


Density-Reachable




Density-Connected

$\epsilon = 1$ unit
MinPts = 5



Outline

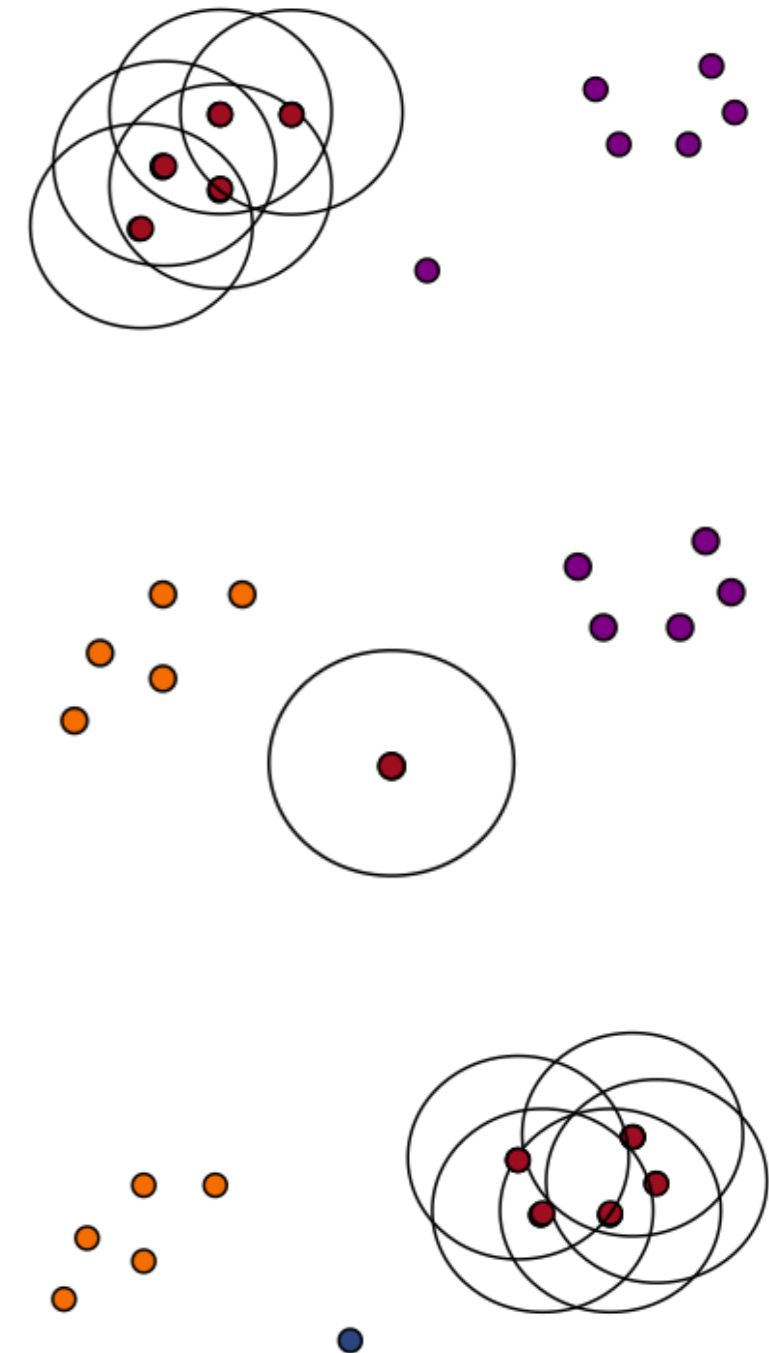
- Overview
- Basic Concepts
- The DBSCAN Algorithm ← 
- Analysis of DBSCAN

The DBSCAN Algorithm


```
DBSCAN(X, eps, MinPts)
C = 0
for each unvisited point P in dataset X
    mark P as visited
    NeighborPts = regionQuery(P, eps)
    if sizeof(NeighborPts) < MinPts
        mark P as NOISE
    else
        expandCluster(P, NeighborPts, C, eps, MinPts)
        C = next cluster

expandCluster(P, NeighborPts, C, eps, MinPts)
    add P to cluster C
    for each point P' in NeighborPts
        if P' is not visited
            mark P' as visited
            NeighborPts' = regionQuery(P', eps)
            if sizeof(NeighborPts') >= MinPts
                NeighborPts = NeighborPts joined with NeighborPts'
        if P' is not yet member of any cluster
            add P' to cluster C
```

regionQuery(P, eps) return all points within P's eps-neighborhood (including P)



Outline

- Overview
- Basic Concepts
- The DBSCAN Algorithm
- Analysis of DBSCAN ← 

DBSCAN is Sensitive to Parameters

Figure 8. DBScan results for DS1 with MinPts at 4 and Eps at (a) 0.5 and (b) 0.4.

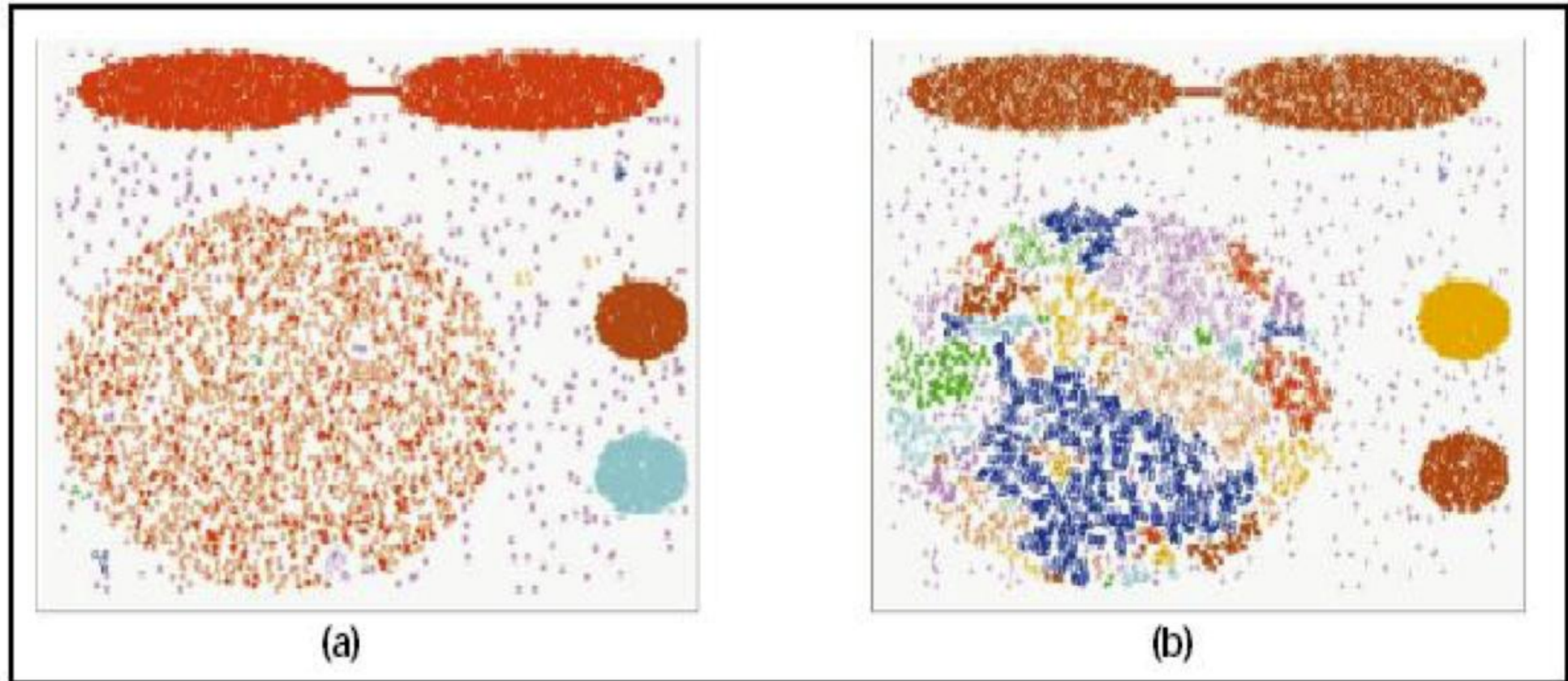
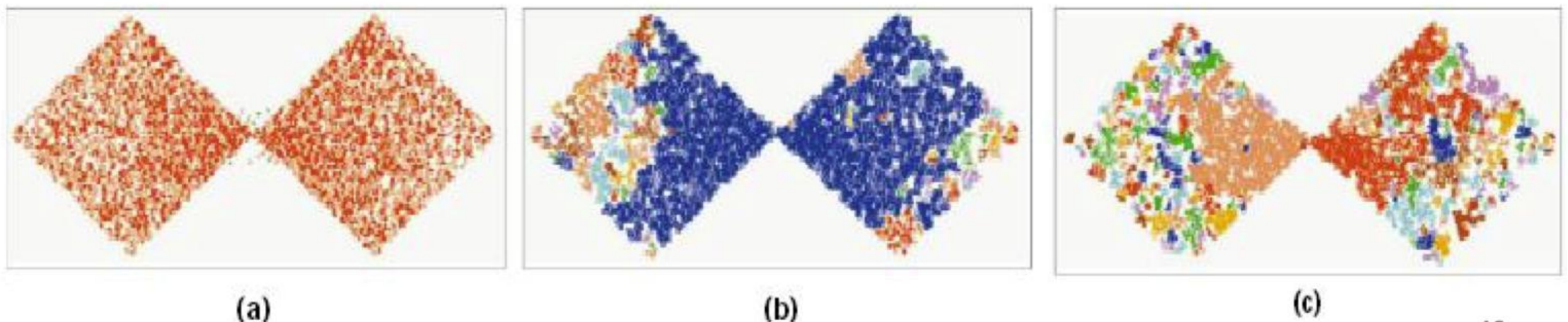
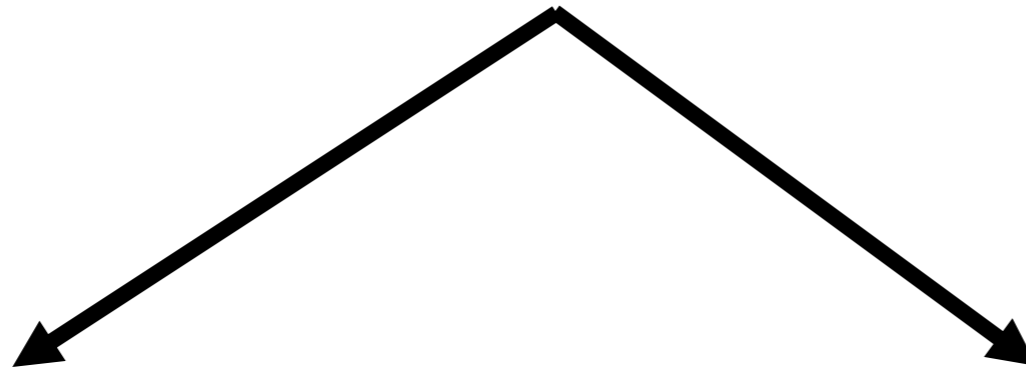


Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.



ϵ



High value (what will happen?)

Low value (what will happen?)

Clusters will merge and the majority of data points will be in the same cluster

A large part of data won't be clustered and considered as outliers. Because, they won't satisfy the number of points to create a dense region

Do we need to define the number of clusters in DBSCAN?

Nope

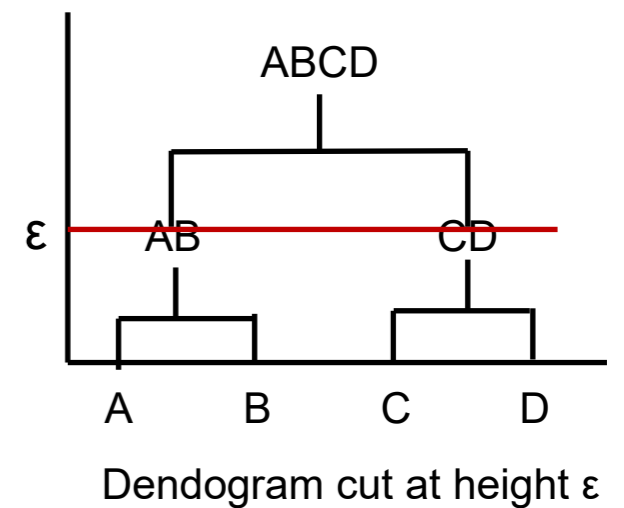
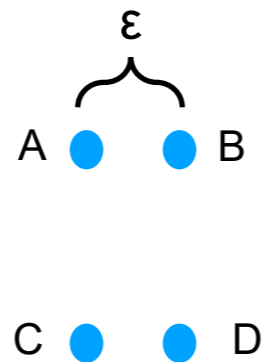
Minimum number of Points (**MinPts**)

Every point will be a cluster on its own, Why?

MinPts = 1?

Don't forget, in DBSCAN, a core point is counted as the number of neighboring points

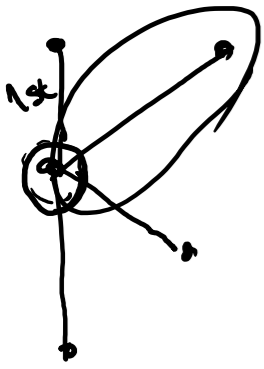
MinPts = 2?



So, MinPts should be at least 3

Rule of thumb, MinPts \geq $D+1$;
For noisy data \Rightarrow MinPts = $2 \cdot D$ (yield more significant clusters)

feature



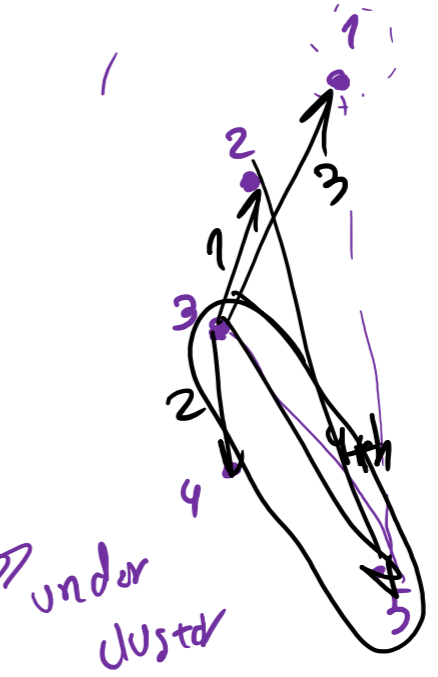
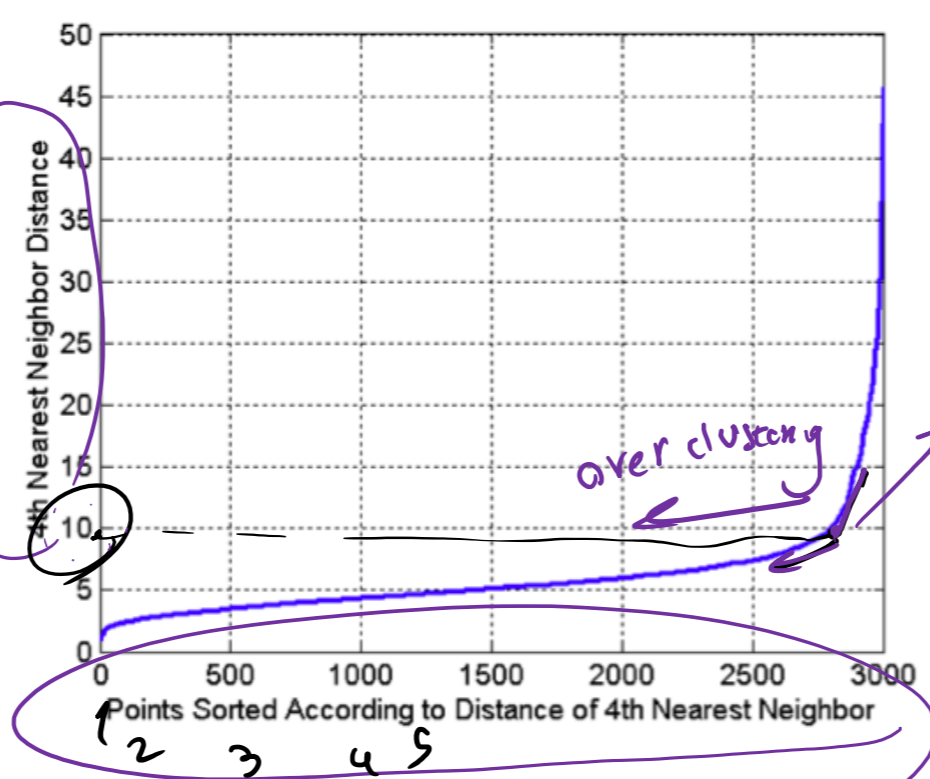
How about Eps? (Elbow effect)

- 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



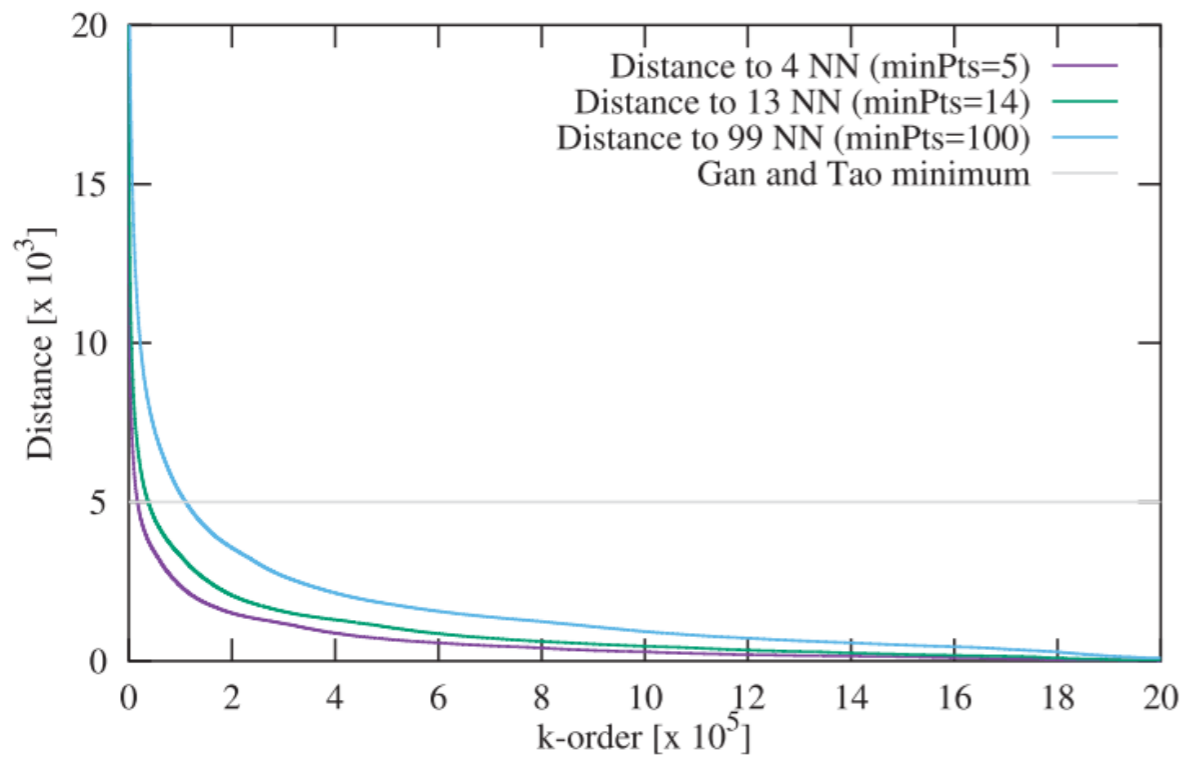
4th nearest neighbor

datapoint index	distance value
1	5.1
2	3.2
3	7.1

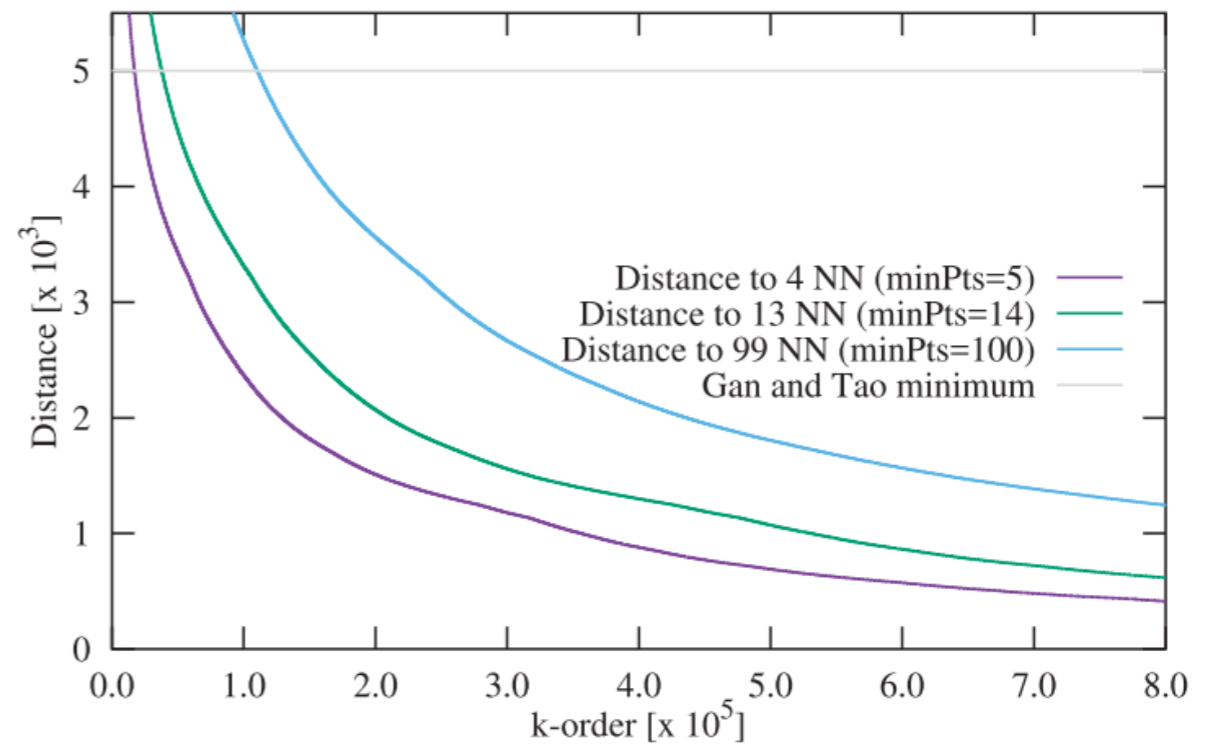


Here we have 3000 points and x-axis shows just a point index. Point indices are sorted in ascending order based on their 4th nearest neighbor distance

Elbow effect another example



(a) k -distance plots



(b) k -distance plots (magnified region)

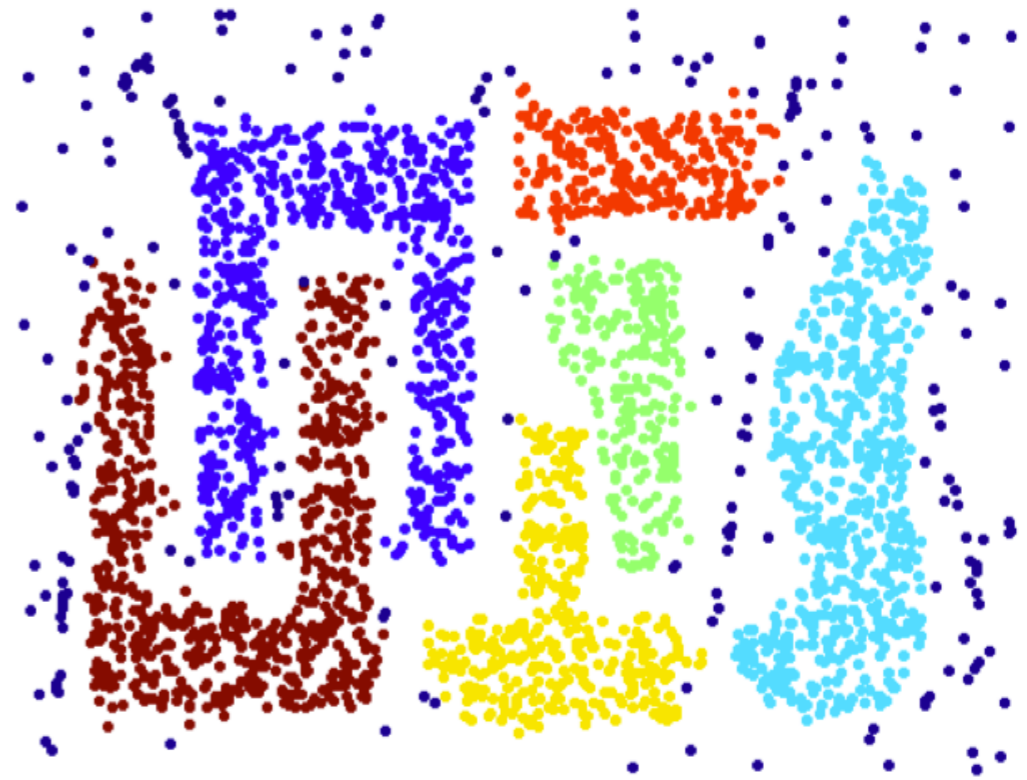
minPts often does not have a significant impact on the clustering results

When DBSCAN Works Well

- Robust to noise
- Can detect arbitrarily-shaped clusters



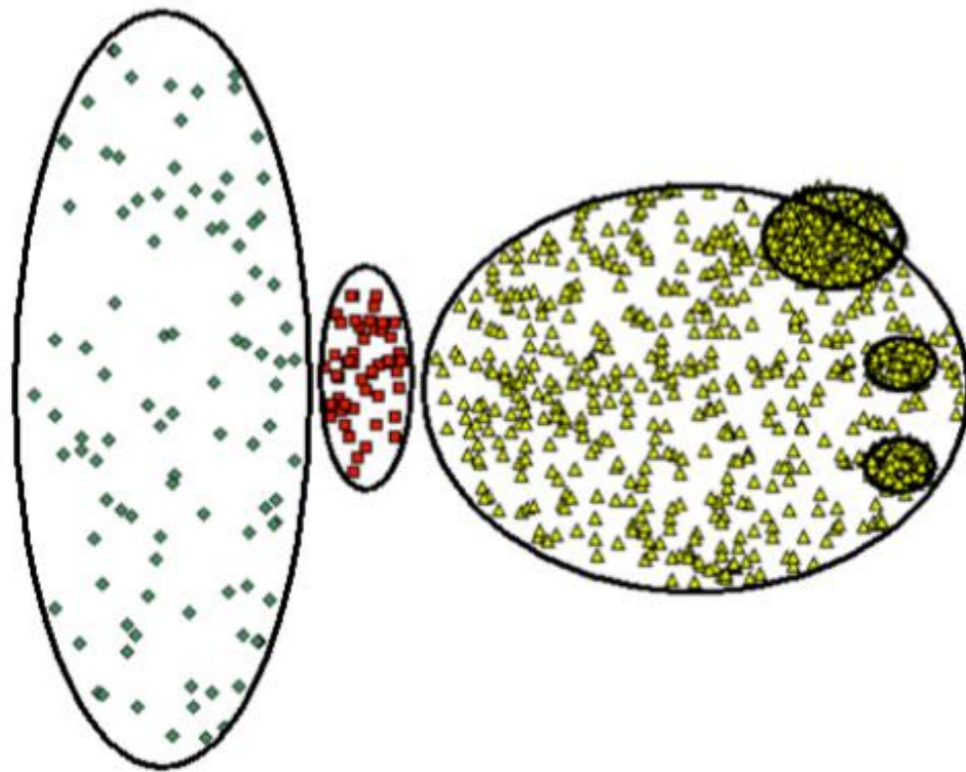
Original Points



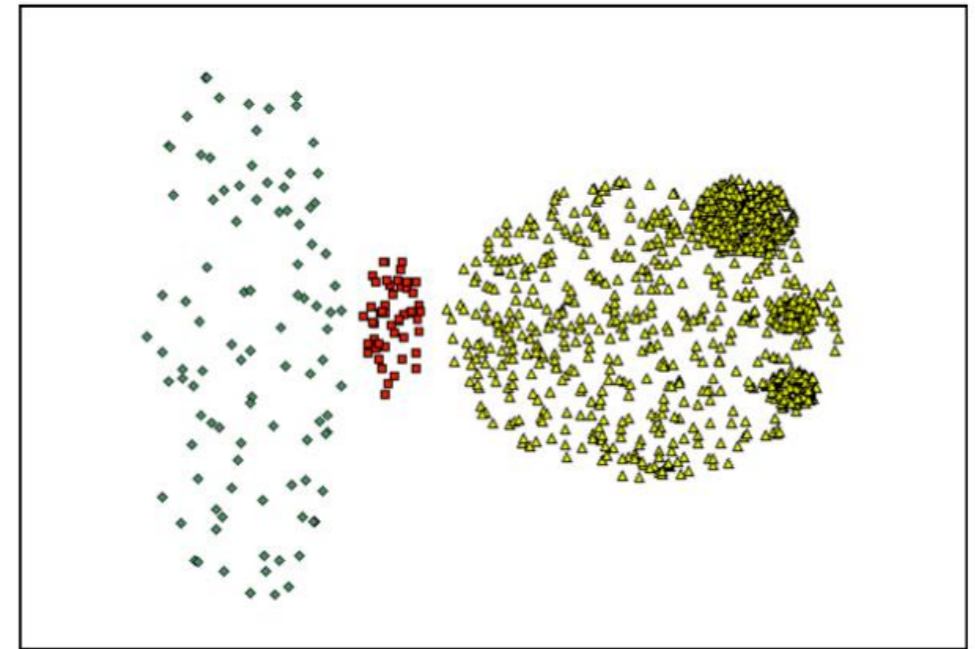
Clusters

When DBSCAN Does NOT Work Well

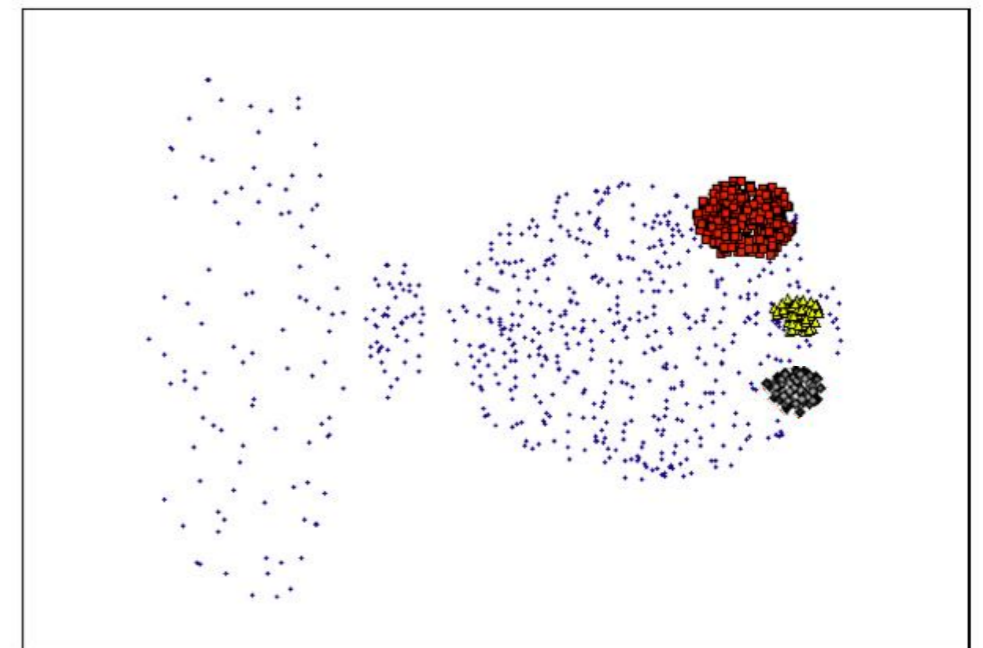
- Cannot handle varying densities
- Sensitive to parameters—hard to determine the best setting of parameters



Original Points



(MinPts=4, Eps=9.92).



(MinPts=4, Eps=9.75)

Take-Home Messages

- The basic idea of density-based clustering
- The two important parameters and the definitions of neighborhood and density in DBSCAN
- Core, border and outlier points
- DBSCAN algorithm
- DBSCAN's pros and cons

Clustering Evaluation

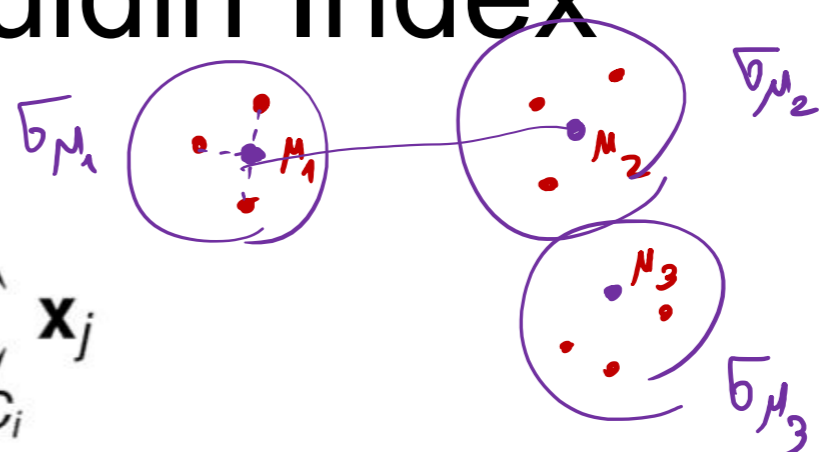
- Internal measures for clustering evaluation
 - Elbow method
 - Silhouette Coefficient
 - Graph-based measures (Beta-CV and Normalized cut)
 - Davies-Bouldin Index *DB Index*

We want intra-cluster datapoints to be as close as possible to each other and inter-clusters to be as far as possible from each other

The Davies-Bouldin Index

Let μ_i denote the cluster mean

$$\mu_i = \frac{1}{n_i} \sum_{\mathbf{x}_j \in C_i} \mathbf{x}_j$$



Let σ_{μ_i} denote the dispersion or spread of the points around the cluster mean

$$\sigma_{\mu_i} = \sqrt{\frac{\sum_{\mathbf{x}_j \in C_i} \delta(\mathbf{x}_j, \mu_i)^2}{n_i}} = \sqrt{\text{var}(C_i)}$$

distance

The Davies-Bouldin measure for a pair of clusters C_i and C_j is defined as the ratio

$D_1, D_2, D_3 \Rightarrow D_1 = \max\{DB_{12}, DB_{13}\}$ $DB_{12} = \frac{\sigma_{\mu_1} + \sigma_{\mu_2}}{\delta(\mu_1, \mu_2)}$

Calculate the DB of i cluster from other clusters

$$DB_{ij} = \frac{\sigma_{\mu_i} + \sigma_{\mu_j}}{\delta(\mu_i, \mu_j)}$$

$$D_i = \max_{i \neq j} DB_{ij}$$

DB_{ij} measures how compact the clusters are compared to the distance between the cluster means. The Davies-Bouldin index is then defined as

$$DB = \frac{1}{k} \sum_{i=1}^k D_i$$

a lower value means that the clustering is better