Happy Friday!

- Project Proposal Due Tonight
- Peer Evaluation Out tonight
  - Due on Apr 20th

Georgia Tech

Kmeans – Hard – K

GMM – Soft – K

Hierarchical Clustering – Hard – X

introduce hyperparameters

Single / Complete / Centroid / Average

L2 Manhattan

DBSCAN – Hard – X

Machine Learning CS 4641-7641

Density-Based Clustering

Nimisha Roy
Lecturer, College of Computing
Director, Online Undergraduate Initiatives

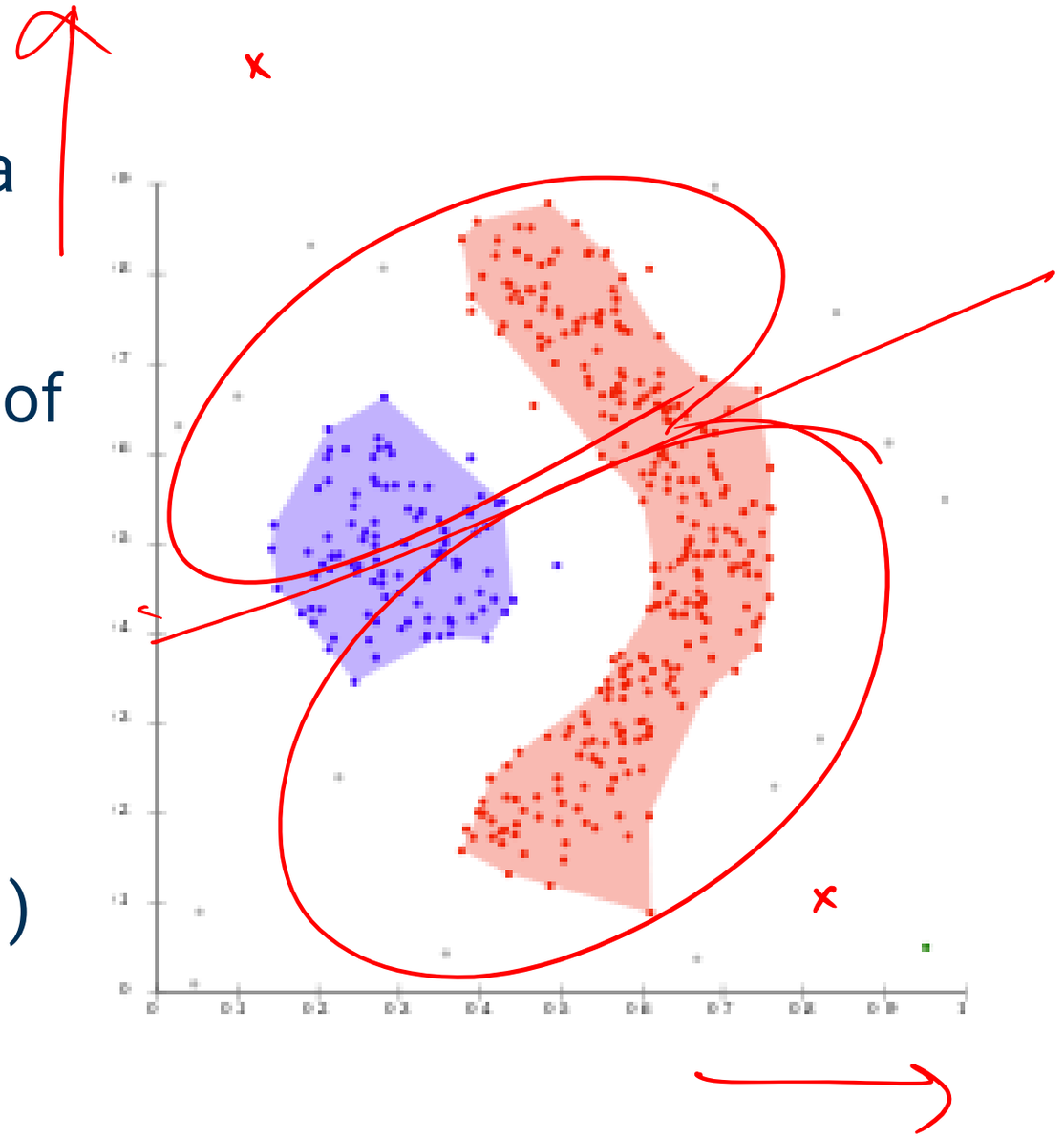
These slides are based on slides from Mahdi Roozbahani

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



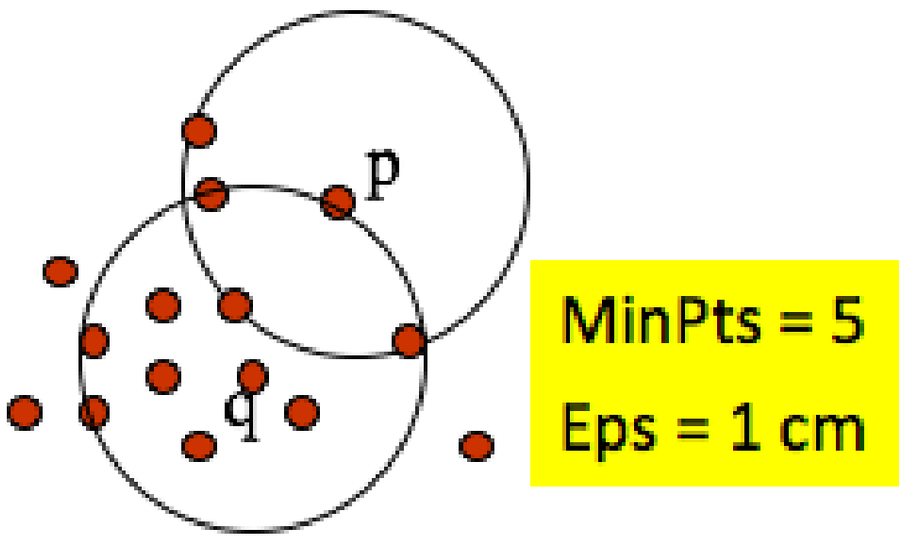
- Method
  - DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

Outline

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

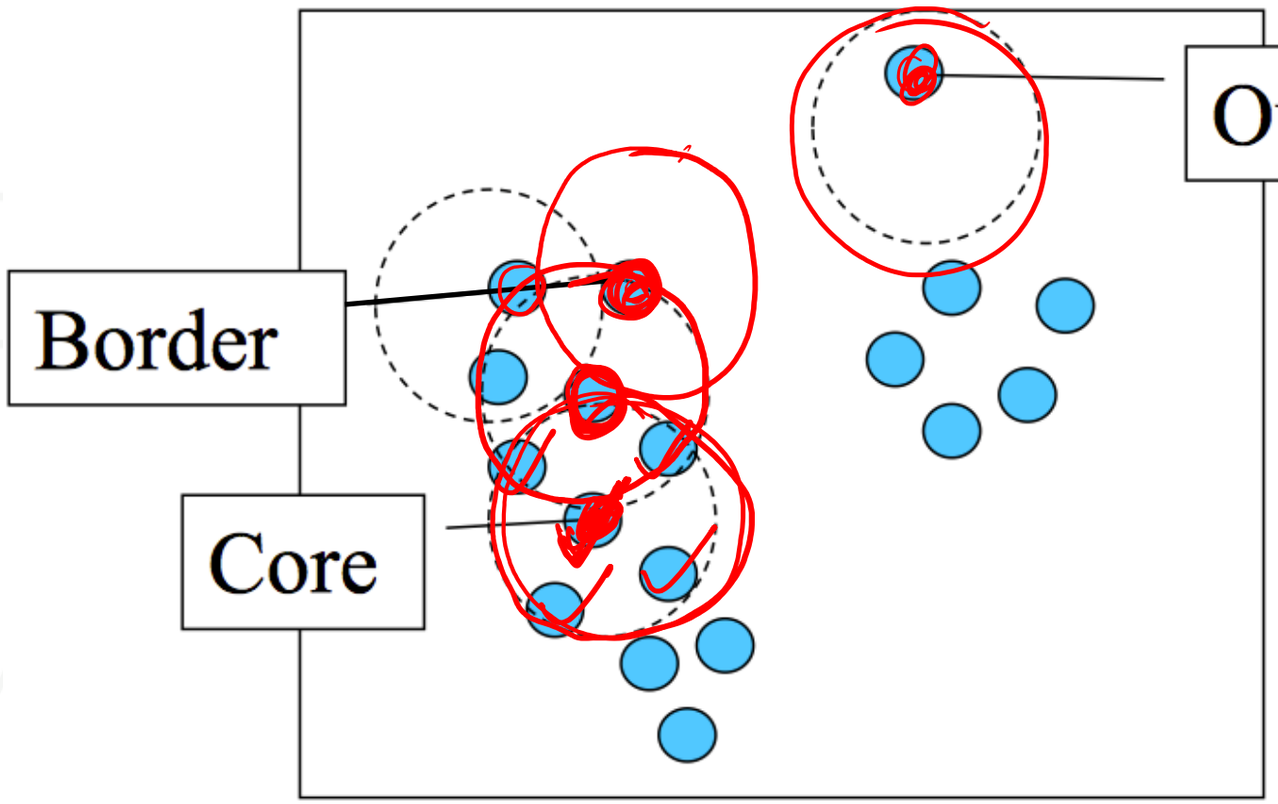
High Density v.s. Low Density

- 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



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

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

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.

DBSCAN algorithm

Goal: Find clusters of arbitrary shapes based on density while marking noisy points as outliers.

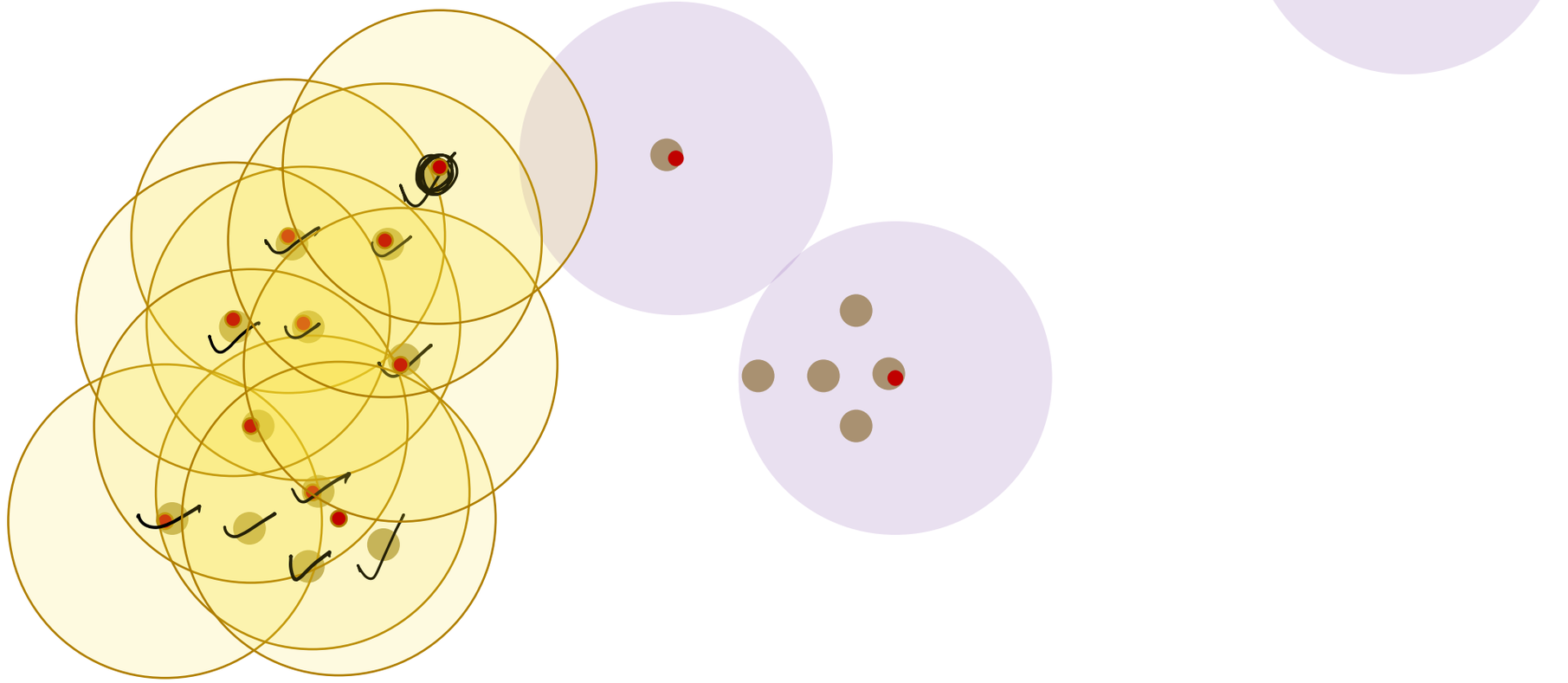
Steps:

- **Pick any unvisited point.**
  - Mark it as *visited* so we don't process it twice.
  - If it has enough neighbors within distance ϵ ($\geq$ *minPts*), it's a **core point** $\rightarrow$ start a new cluster.
  - If not, temporarily mark it as **noise** (it might later become a border point if another cluster grows into it).
- **Expand the cluster.**
  - Add all neighbors of the core point into a **queue of unvisited points**.
  - Each time we pull a point from the queue, we mark it *visited*.
  - If it's also a **core point**, add its neighbors to the queue (the cluster expands).
  - If it's a **border point**, just add it to the cluster, but don't expand further.
- **Repeat until queue is empty.**
  - Once there are no more unvisited points connected to the cluster, the cluster is complete.
- **Start again with another unvisited point.**
  - If it turns out to be a new core, grow another cluster.
  - If it's isolated, leave it as noise.

Georgia Tech

Practice:

ε= 1 unit
MinPts = 5



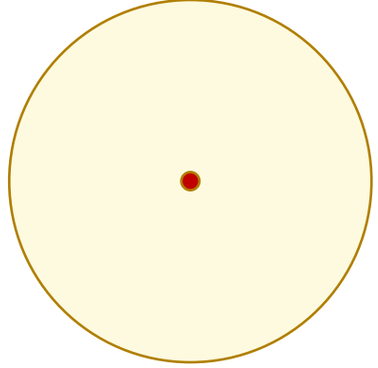
unvisited points: [↓ ↓ ↓]

Key Intuition

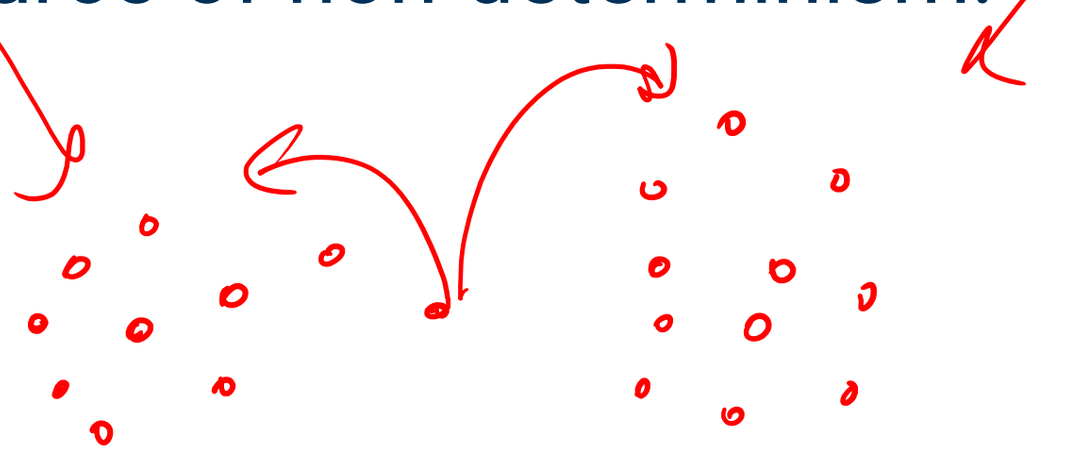
- **Core points** = dense regions (seeds of clusters).
- **Border points** = edges of clusters (belong to cluster but don’t expand).
- **Noise** = isolated points (too few neighbors)

Deterministic?

same input → same output

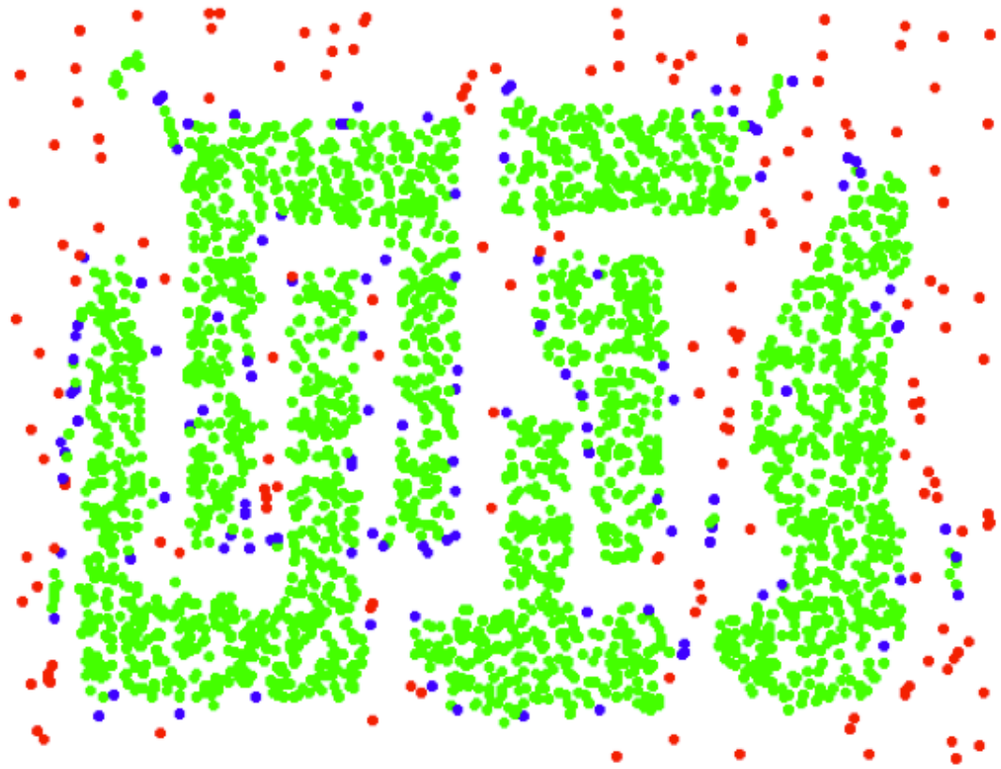


- **Almost deterministic**
- The result is deterministic **except for border points**.
- If a border point lies in the neighborhood of two clusters, it may get assigned differently depending on which cluster grows first.
- This is the only source of non-determinism.



Examples

Original Points



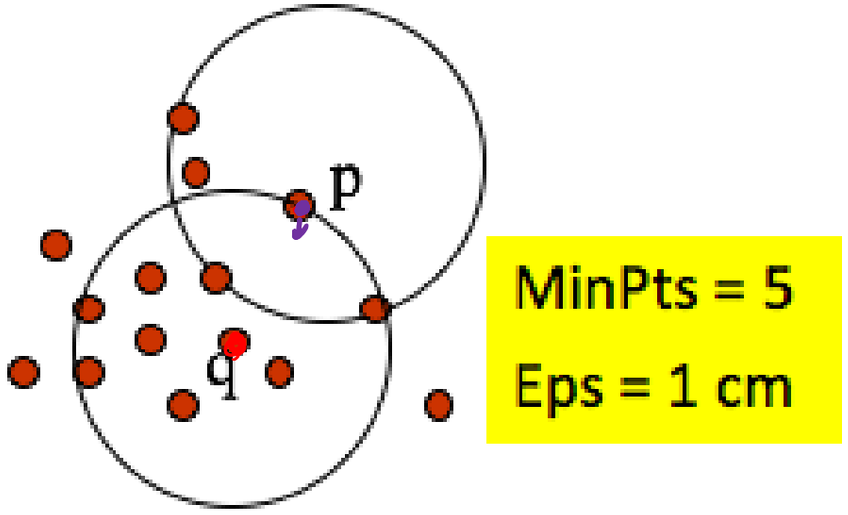
Point types: core, border and outliers

$\varepsilon = 10$, MinPts = 4

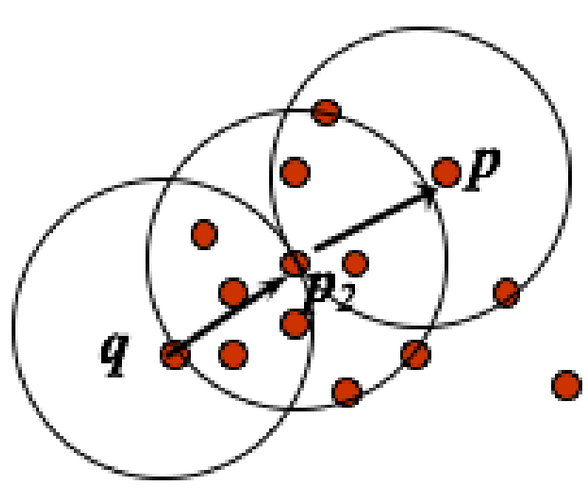
Density-based related points

- 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**

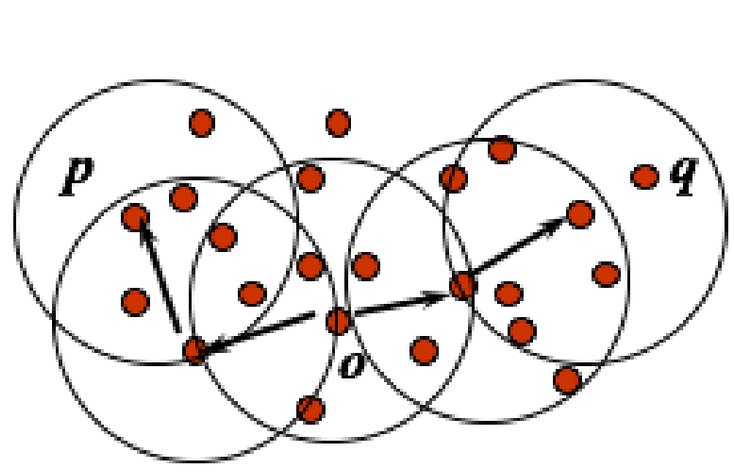
p is direct density reachable from q



Directly Density-Reachable



Density-Reachable



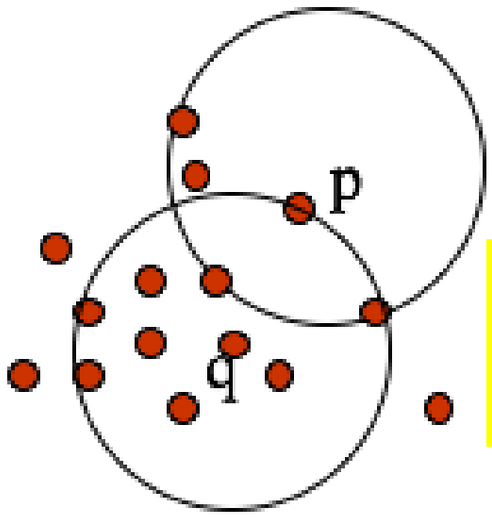
Density-Connected

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, …, p_n, $p_1 = q$, $p_n = p$*** such that ***p_{i+1}*** is directly density-reachable from ***p_i***
  - ***$p_1 = q \rightarrow p_2 \rightarrow \ldots \rightarrow p_n = q$***

p is density reachable from q

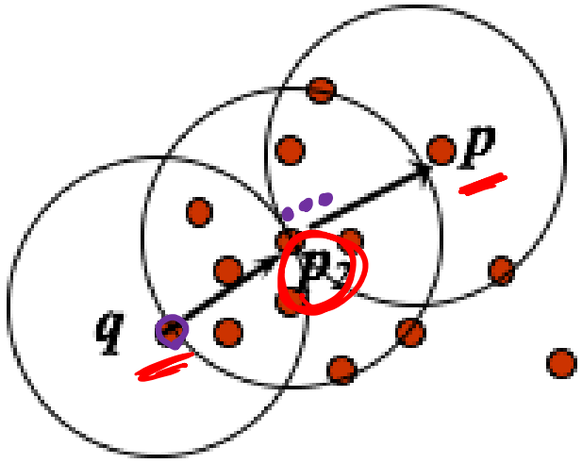
✗ Not a 2-way relationship



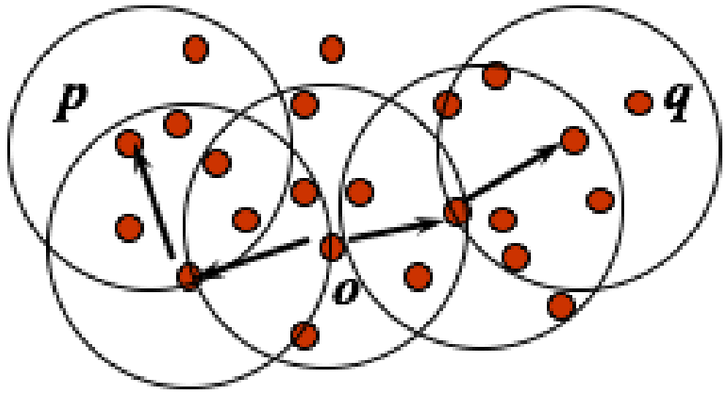
MinPts = 5

Eps = 1 cm

Directly Density-Reachable



Density-Reachable

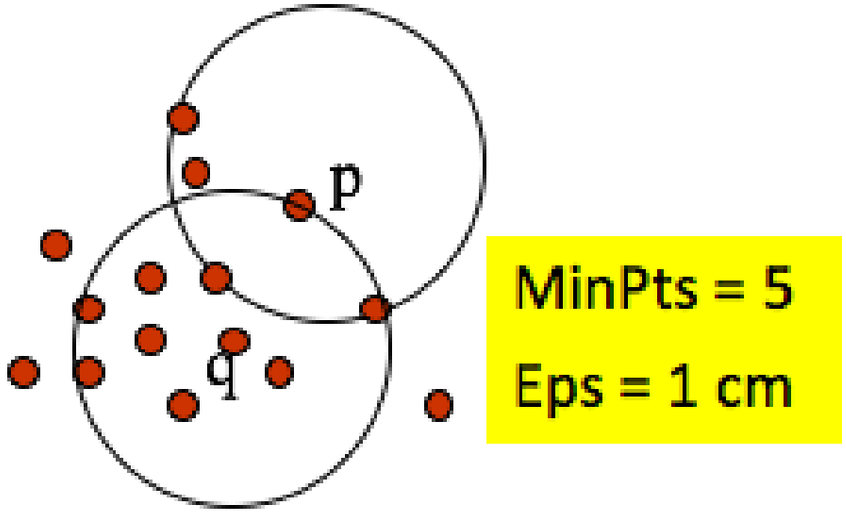


Density-Connected

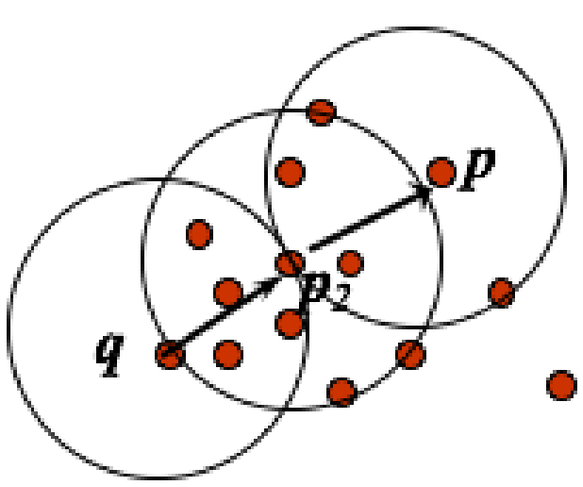
Density-based related points

- 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***

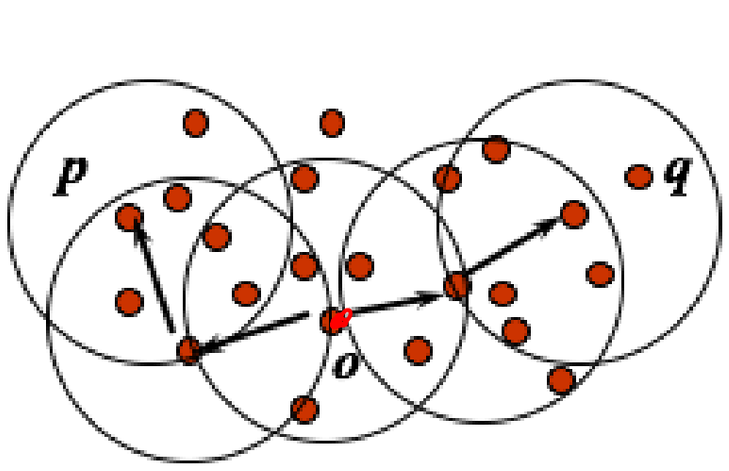
p & q are density connected



Directly Density-Reachable



Density-Reachable



Density-Connected

Georgia Tech

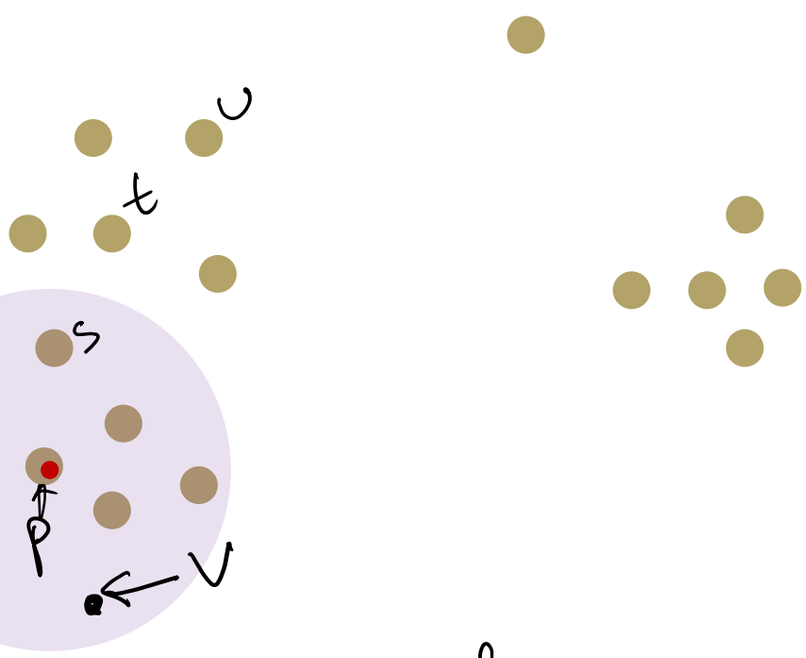
t is direct density reachable from s

p is " " " from s

v is " " " from t

s is " " " from t

ε= 1 unit
MinPts = 5



v density reachable from s

v is d d r from p

v $\xrightarrow{dr}$ s

v & u are density connected

Outline

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

The DBSCAN Algorithm

Algorithm is super sensitive to eps

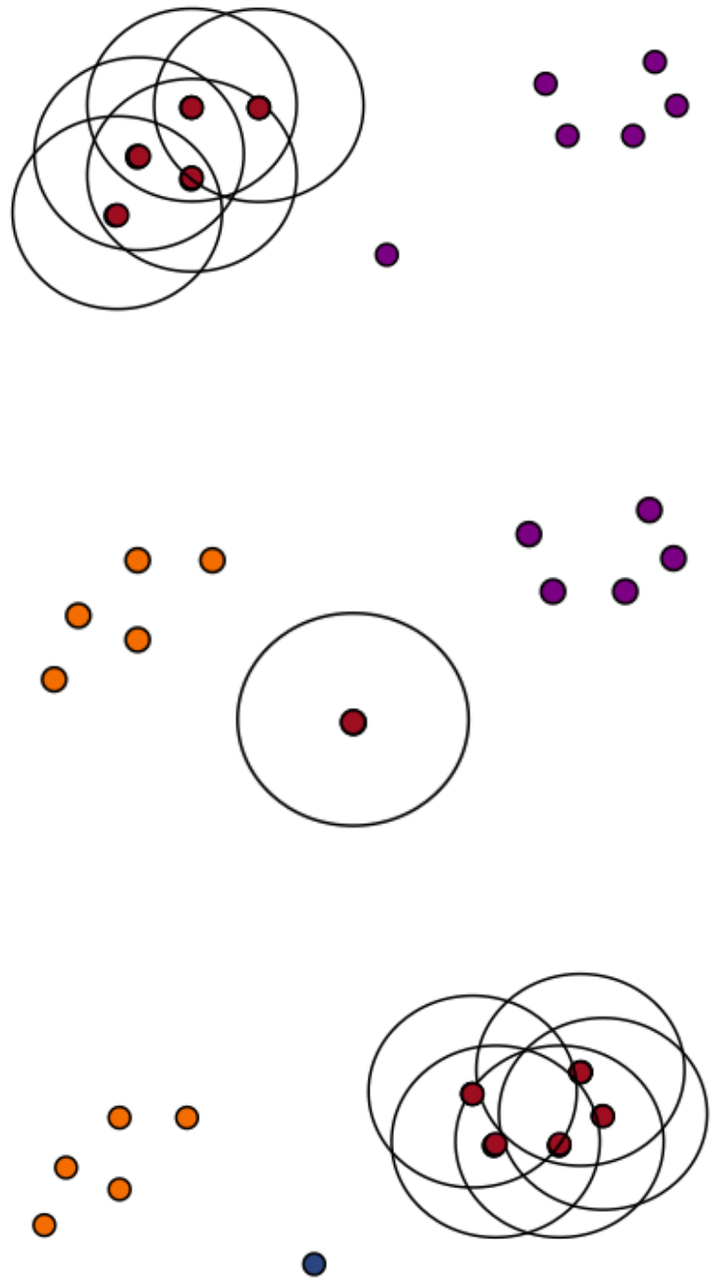
data

```
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)
```

which is very computationally expensive



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

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.

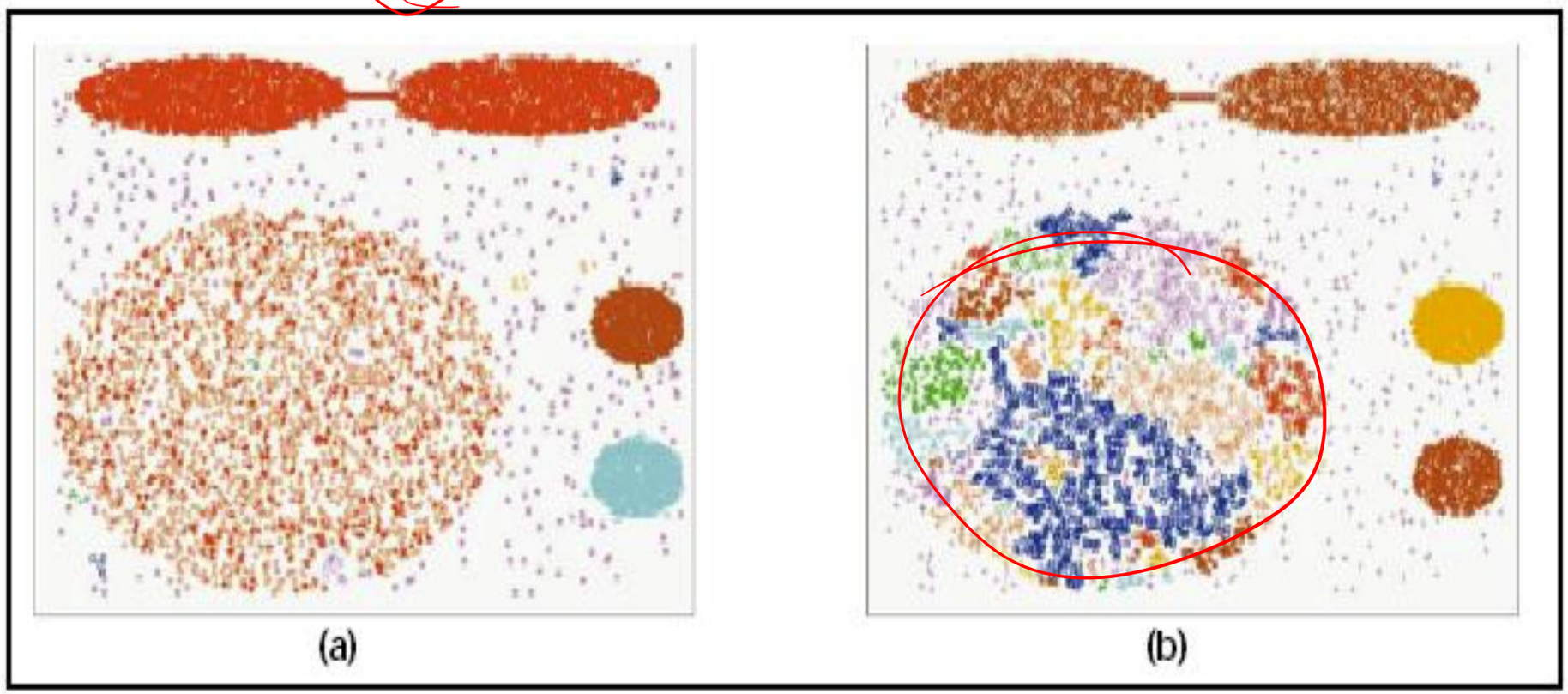


Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.

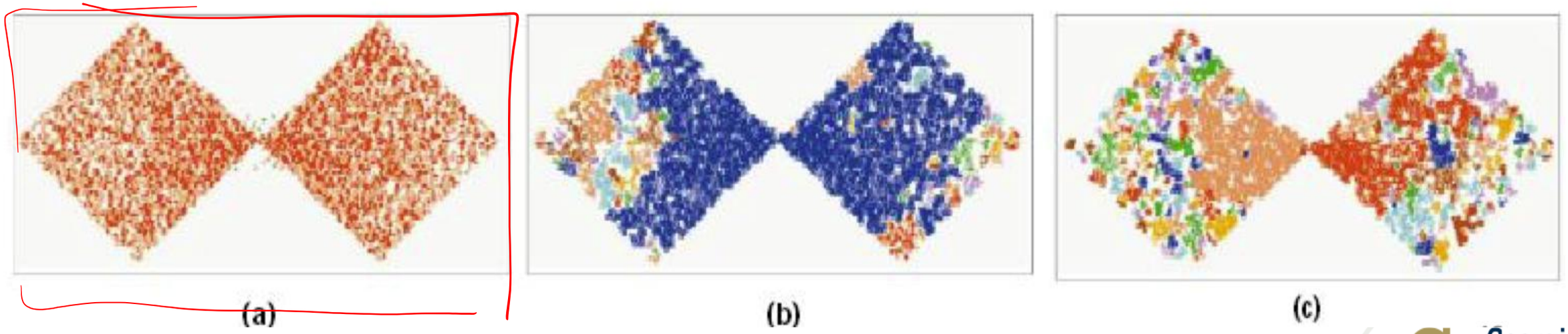
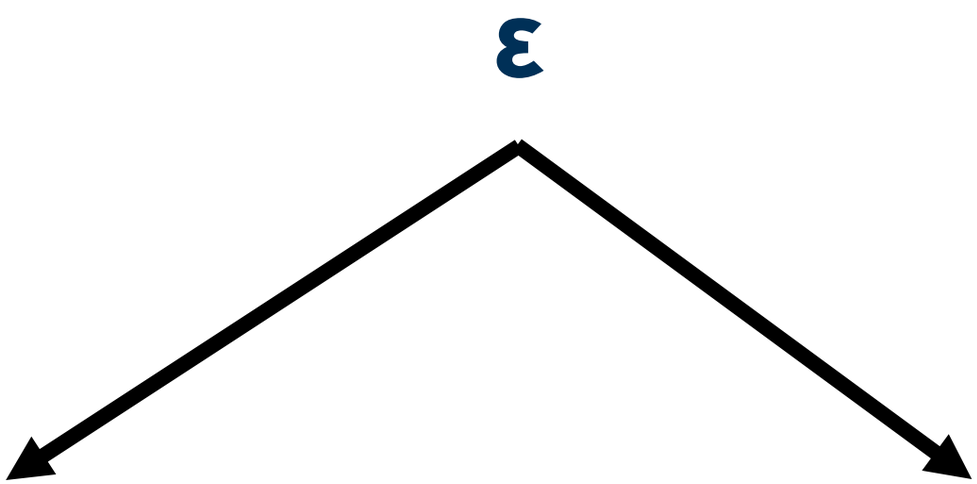


Image Credit: George Karypis.



High value (what will happen?)

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

Low value (what will happen?)

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

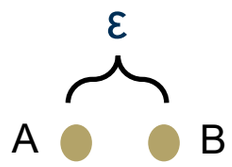
Minimum number of Points (**MinPts**)

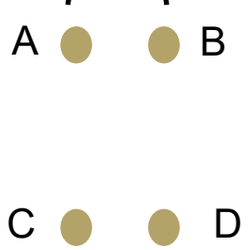
MinPts = 1?

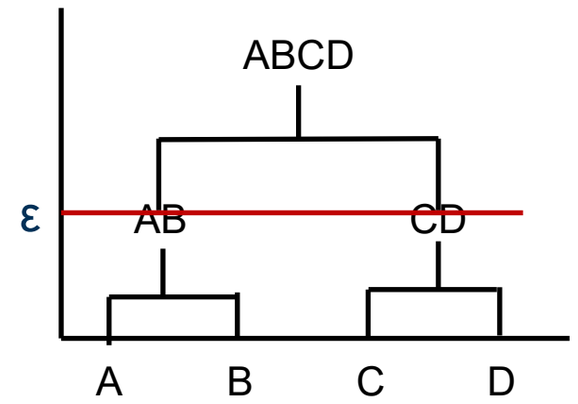
Every point will be a cluster on its own, Why?

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

MinPts = 2?







Dendogram cut at height ε

So, MinPts should be at least 3

no. of dimensions

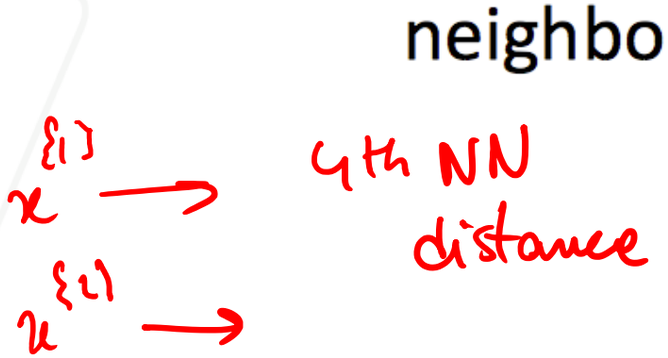
Rule of thumb, MinPts >= D+1;

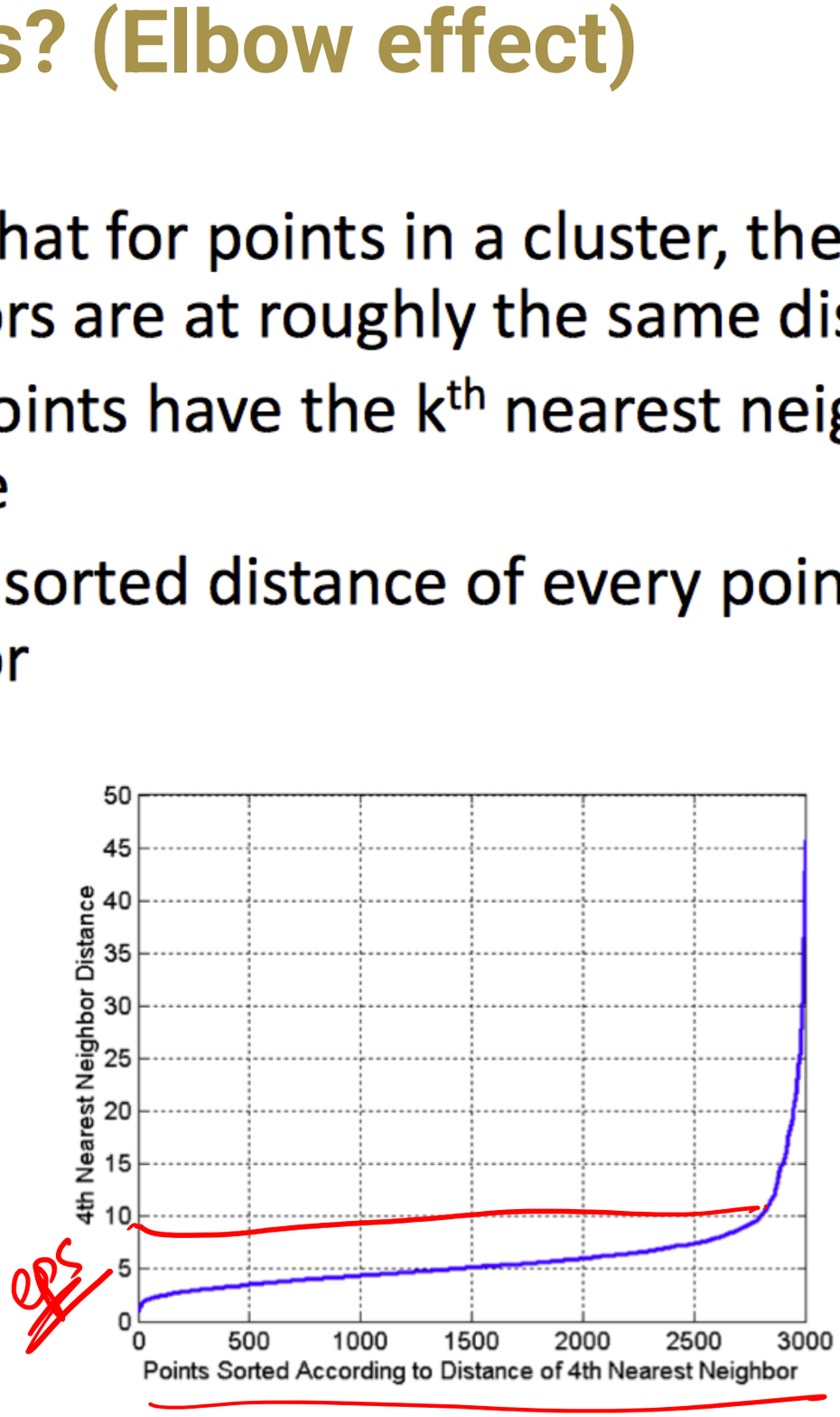
For noisy data => MinPts = 2*D (yield more significant clusters)

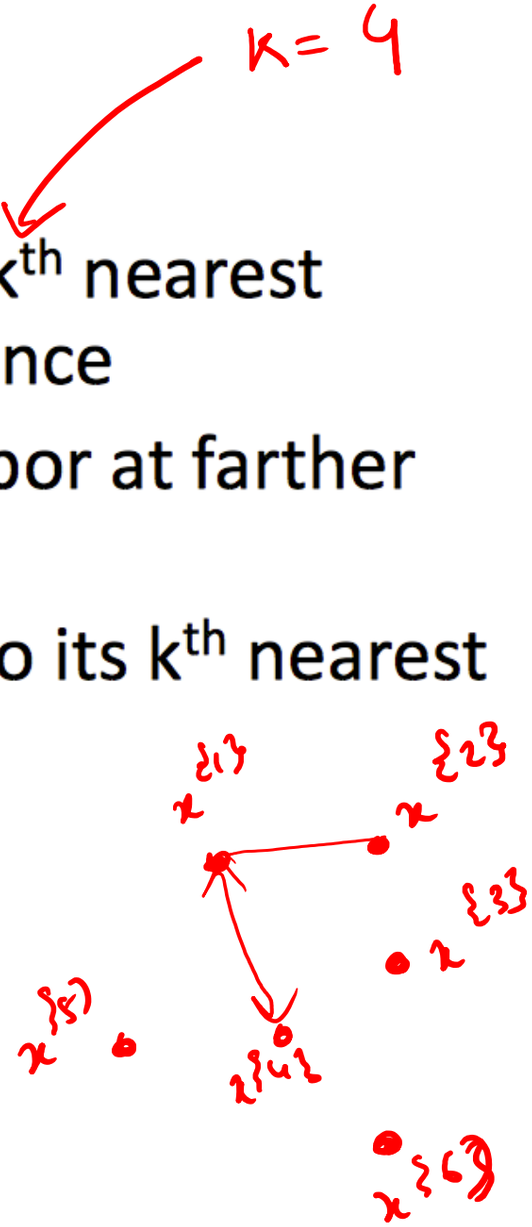
Georgia Tech

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



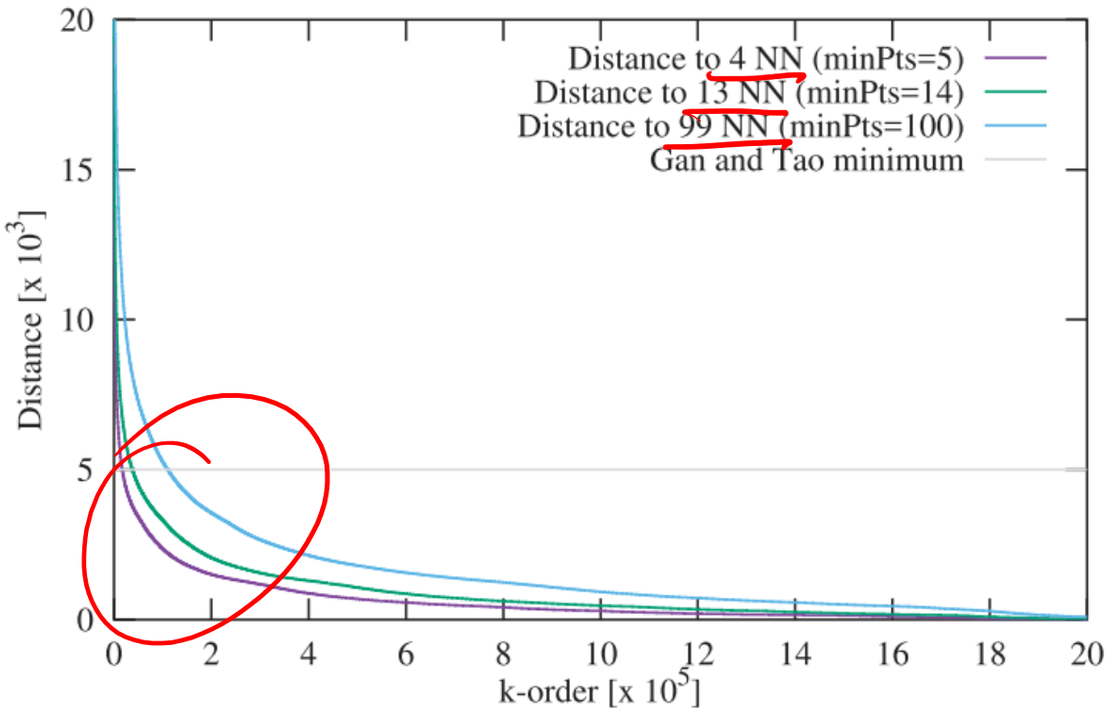




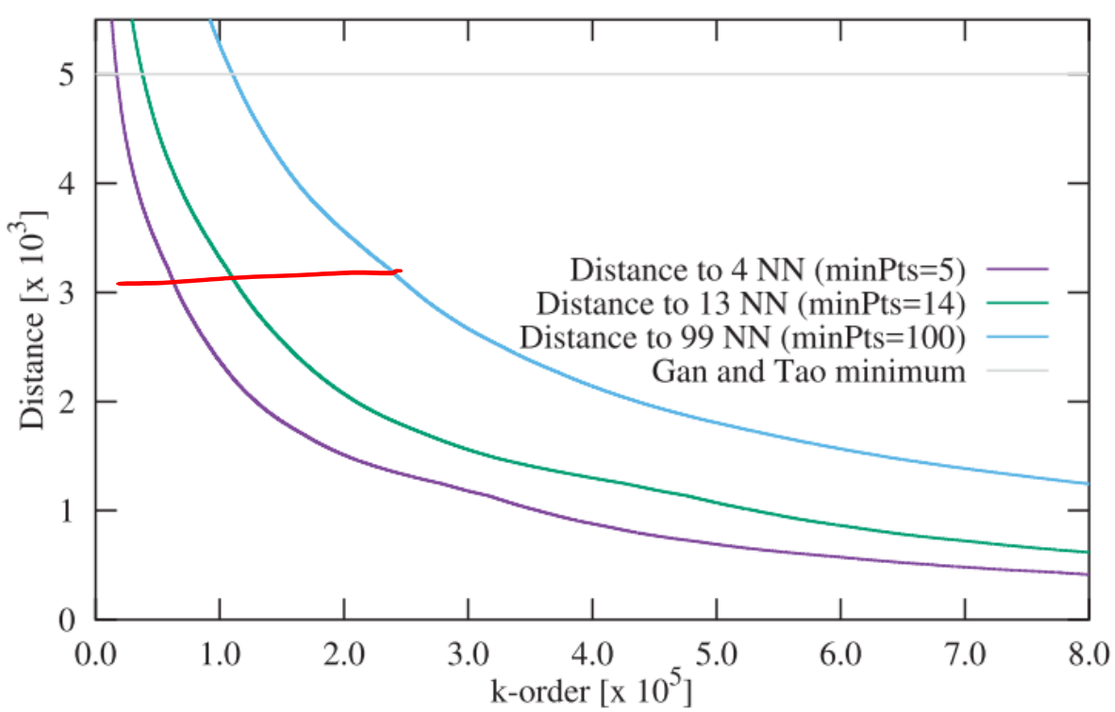
Here we have 3000 points and x-axis shows just a point index.

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)

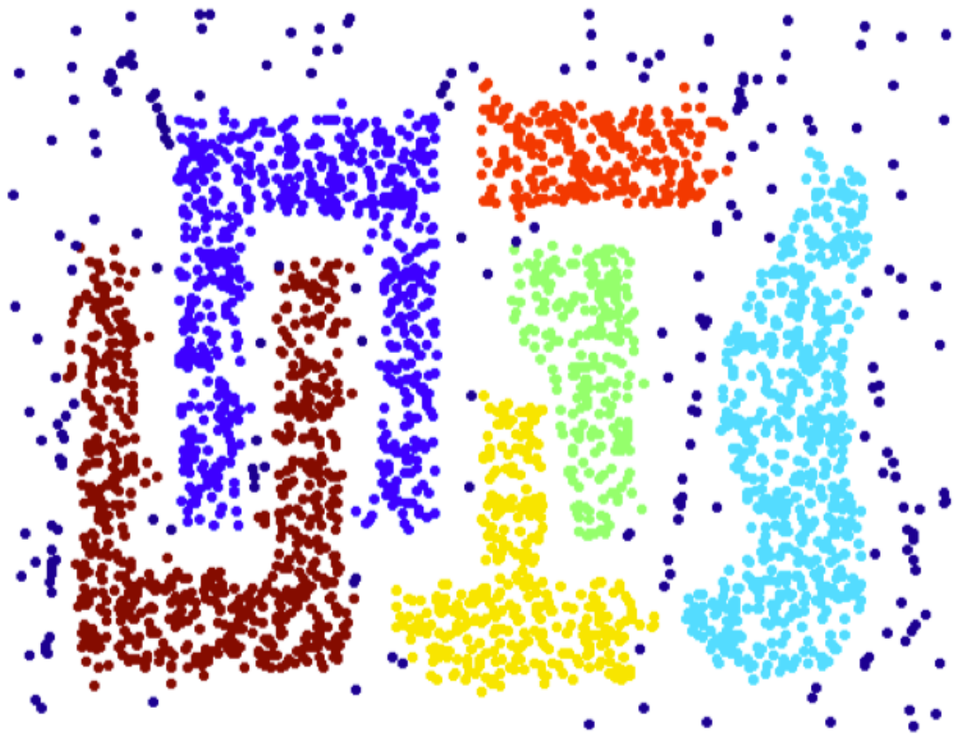
minPts often does not have a significant impact on the clustering results

[Erich Schuber et al]

When DBSCAN Works Well

- Robust to noise
- Can detect arbitrarily-shaped clusters

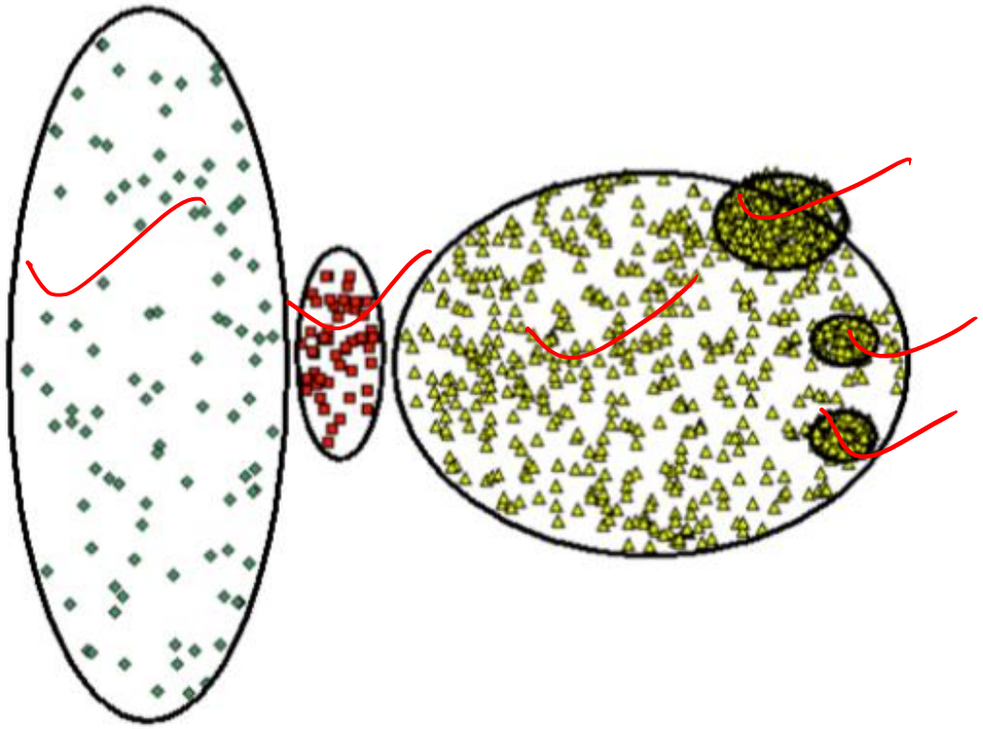
Original Points



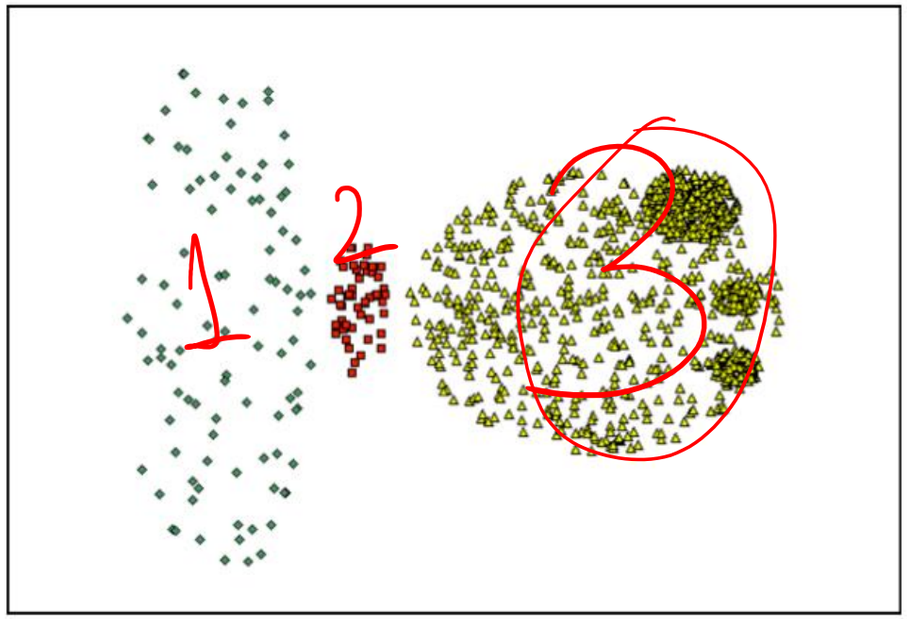
Clusters

When DBSCAN Does NOT Work Well

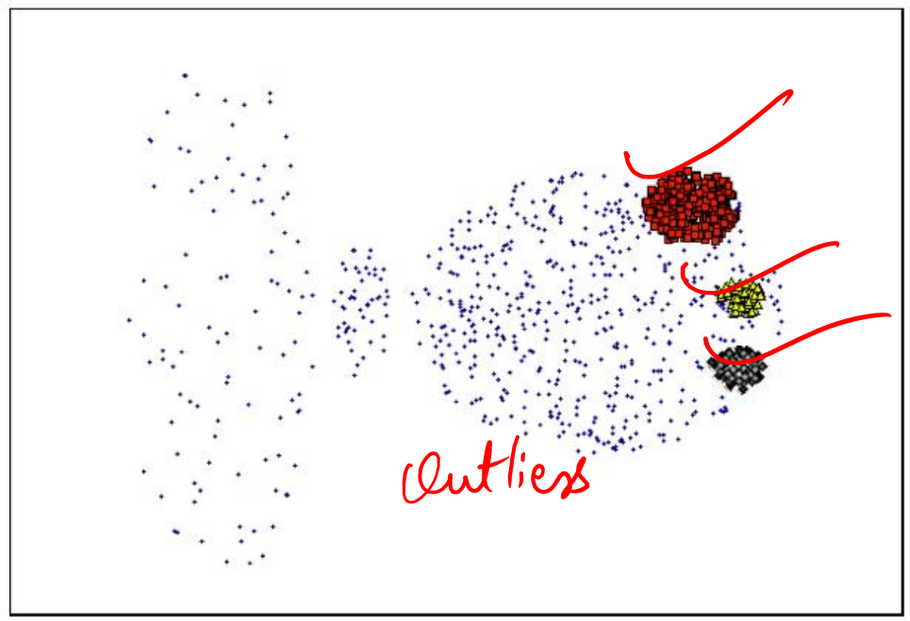
- Cannot handle varying densities
- Sensitive to parameters—hard to determine the best setting of parameters



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

Georgia Tech

Which clustering method to choose?

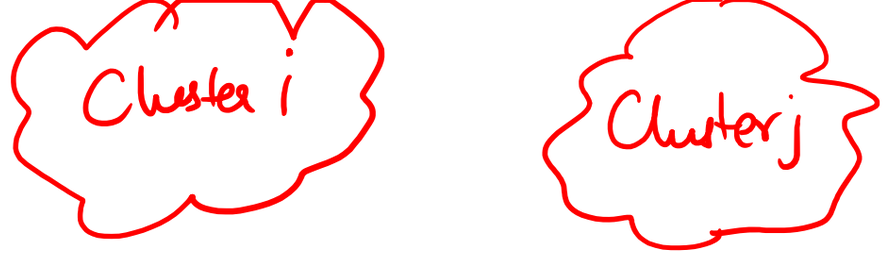
Algorithm	Strengths	Best Contexts	Limitations
K-Means	Simple, fast, scalable to large datasets	Well-separated, spherical clusters; when you know/guess (k)	Struggles with non-spherical shapes, sensitive to outliers
GMM (Gaussian Mixture Models)	Probabilistic clusters, allows soft assignments	Data with overlapping clusters; when clusters are elliptical	Sensitive to initialization, assumes Gaussian distribution
Hierarchical Clustering	Produces dendrogram (multi-level clustering), no need to pre-specify (k)	Small–medium datasets; when hierarchy/taxonomy is useful (e.g. document organization)	Computationally expensive for large (n), sensitive to linkage choice
DBSCAN	Can find arbitrarily shaped clusters, robust to noise/outliers	Spatial/geometric data; clusters of varying shapes and sizes	Struggles with varying densities, parameter tuning (epsilon, minPts)

Georgia Tech

Clustering Evaluation

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

dispersion

Should be low

distance

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

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

Calculate the DB of i cluster from other clusters

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

LOW

HIGH

Worst Case

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

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

Quick Knowledge Check

1. DBSCAN’s key idea is to find clusters as…A. Groups minimizing MSE around means B. Dense regions separated by low-density gaps. C. Points on a grid with equal spacing

2. A **border** point is: A. Not within ε of any point. B. Within ε of a core point but not itself core. C. The first point visited

3. Increase ε (holding MinPts fixed). What most likely happens? A. More clusters, more noise. B. Fewer clusters; clusters may merge. C. No effect. D. All points become noise

4. Which statement is **true** about DBSCAN?. A. You must pre-specify the number of clusters k. B. It struggles with arbitrarily shaped clusters. C. It is robust to noise and finds non-spherical clusters. D. It cannot mark outliers

5. The **k-distance (elbow) plot** helps choose ε by: A. Picking the global maximum of distances. B. Picking the knee of sorted k-th nearest neighbor distances. C. Minimizing average distance to centroids. D. Maximizing silhouette score

6. **Density-reachability** means p is reachable from q if: A. There’s a path of core points each within ε of the next. B. They share the same nearest centroid. C. Their Euclidean distance is minimum. D. They are in the same k-means cluster