Freie Universität Berlin

Lecture: Machine Learning for Data Science

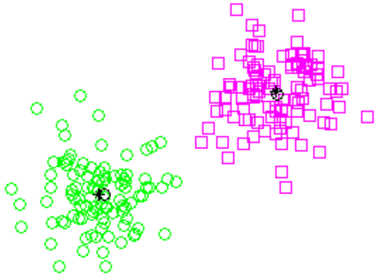
Winter semester 2021/22

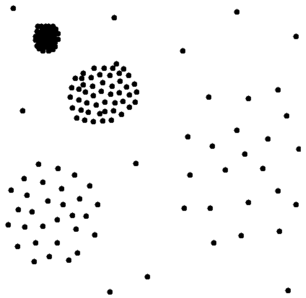
Lecture 12: Unsupervised learning –Density-based clustering

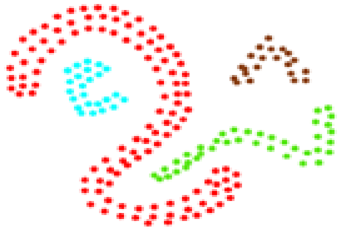
Prof. Dr. Eirini Ntoutsi

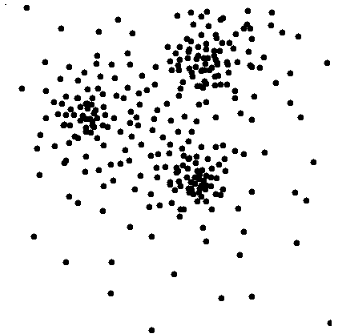
Clustering topics covered in this lecture

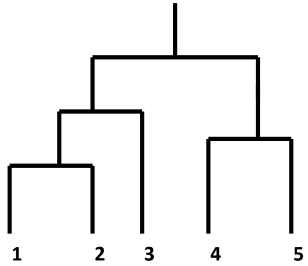
- Partitioning-based clustering
  - k-Means, k-Medoids
- Hierarchical clustering
- Density-based clustering
- Grid-based clustering
- Soft clustering
- Clustering evaluation

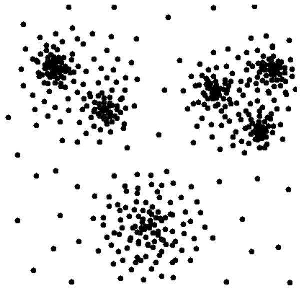


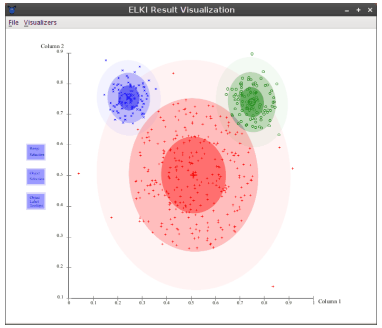












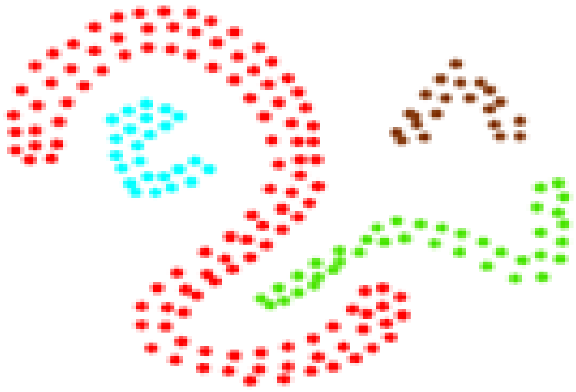
Outline

- Density-based clustering basics
- DBSCAN
- Grid-based clustering (shortly)
- Things you should know from this lecture & reading material

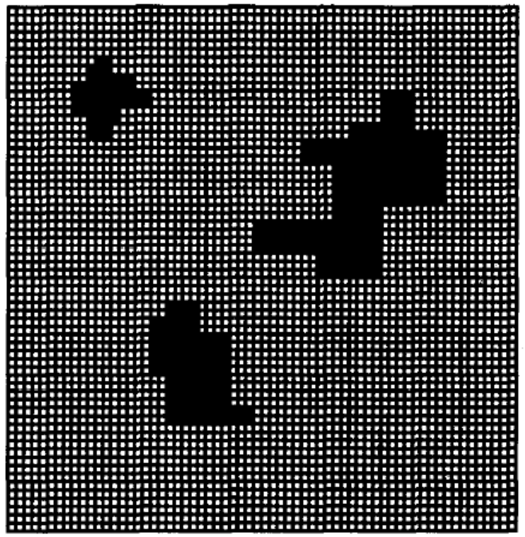
Density based clustering

- Clusters are regions of high density surrounded by regions of low density (noise)

- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Density-related parameter are required

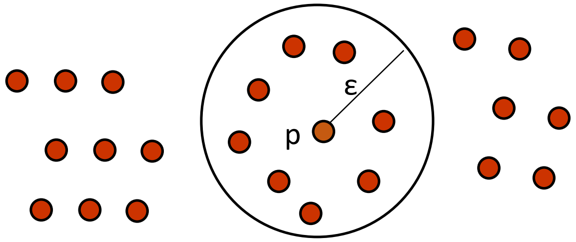


- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96)
  - OPTICS: Ankerst, et al (SIGMOD'99).
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98) (more grid-based)



The notion of density 1/2

- The density-based clustering approach (e.g., in DBSCAN)
  - Density is measured locally in the Eps-neighborhood (or ε-neighborhood) of each point
  - Density = number of points within a specified radius Eps (point itself included)
  - A cluster is a maximal set of density-connected points.



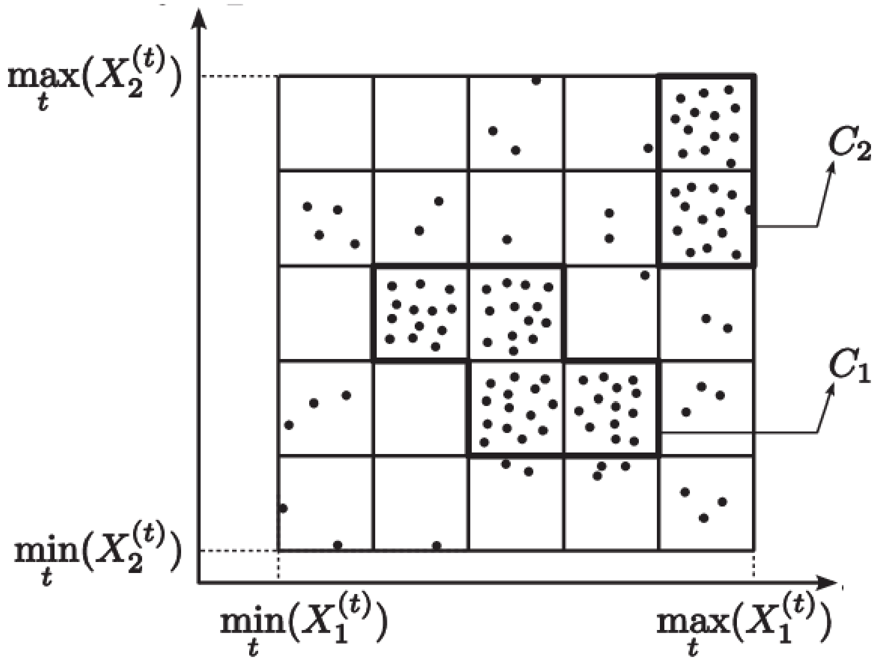
the ε-neighborhood of p: 9 points

(assessed based on Euclidean distance)

- Density depends on the specified radius Eps
  - In an extreme small radius, all points will have a density of 1 (only themselves)
  - In an extreme large radius, all points will have a density of n (the size of the dataset)

The notion of density 2/2

- The grid-based clustering approach (e.g., in CLIQUE)
  - A grid structure is used to capture the density of the dataset.
  - Density is measured locally in each grid cell
  - Density = number of points within each cell
  - A cluster is a set of connected dense cells
- Clustering depends on the grid structure
  - Grid parameters (cell size and density) are required

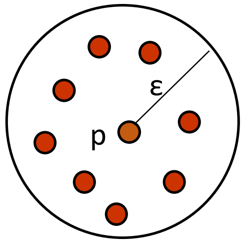


Outline

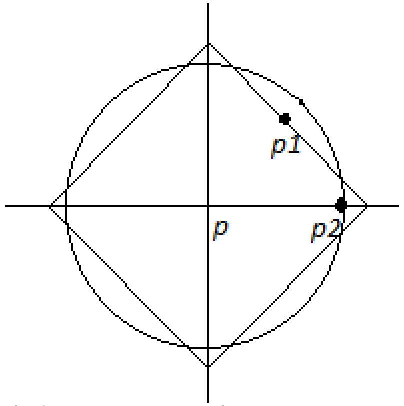
- Density-based clustering basics
- DBSCAN
- Grid-based clustering (shortly)
- Things you should know from this lecture & reading material

DBSCAN basic concepts

- Consider a dataset *D* of *n=|D| d*-dimensional objects to be clustered
- Two parameters:
  - *Eps* (or ε): Maximum radius of the neighborhood
  - *MinPts*: Minimum number of points in an Eps-neighborhood of that point (or, minimum density)
- Eps-neighborhood of a point *p* in *D*
  - $N_{Eps}(p)=\{q \text{ belongs to } D \mid dist(p,q) <= Eps\}$
- The choice of distance depends on the application per se
- The "shape of the neighborhood" depends on distance function



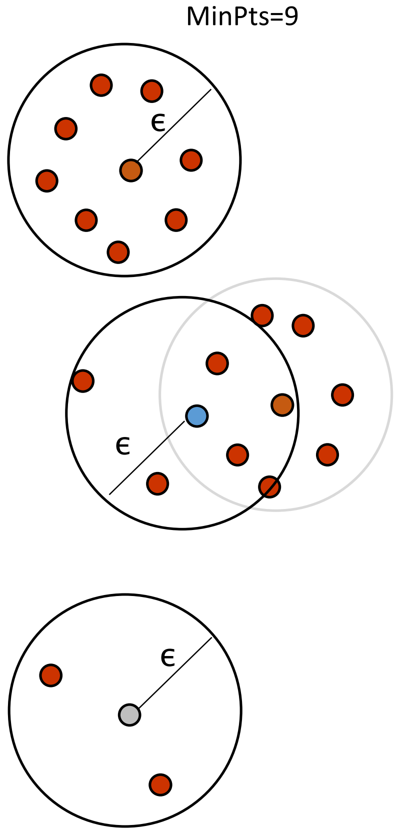
The Eps-neighborhood of p
(using Euclidean distance)



Euclidean vs Manhattan Eps-neighborhood of p

Core points vs border points vs noise points

- DBSCAN characterizes each point in *D* as either core, border or noise
  - Based on the radius parameter Eps and the density parameter MinPts



- Core points: A point is a core point if it has more than a specified number of points (*MinPts*) within a specified radius (*Eps*), i.e.,:

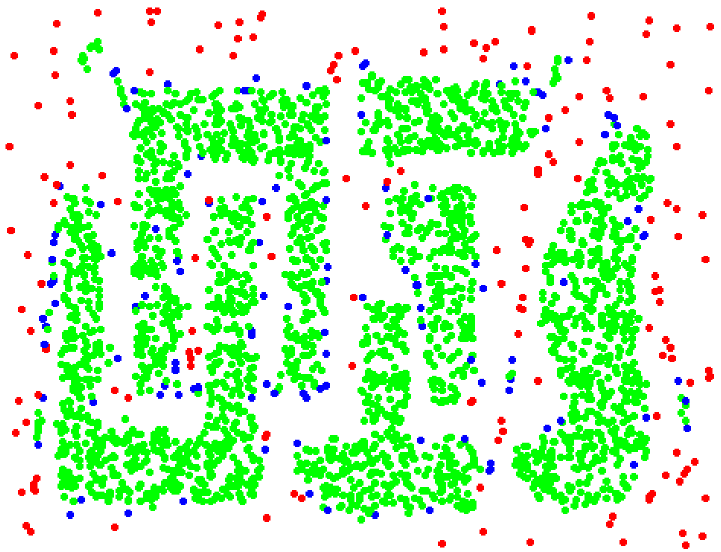
$$|N_{Eps}(p)=\{q \mid dist(p,q) <= Eps \}| \geq MinPts$$

  - these are points that are at the interior of a cluster

- Border points: A border point has fewer than MinPts within Eps radius, but it is in the neighborhood of a core point
  - those are points that belong to the periphery of a cluster

- Noise points
  - neither a core point nor a border point

Core, Border and Noise points



Eps = 10, MinPts = 4

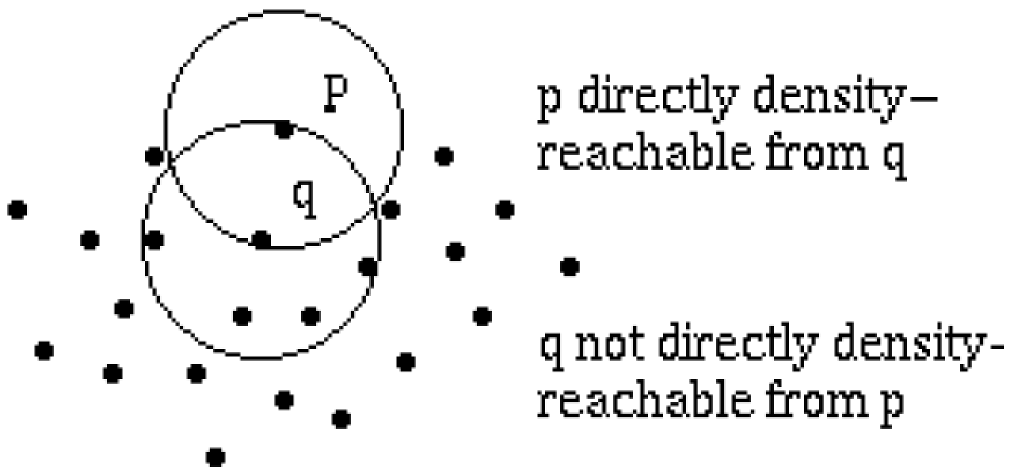
Original points

Point types: core, border and noise

- Core points are points that are at the interior of a cluster
- Border points belong to the periphery of a cluster
- Noise points do not belong to any cluster

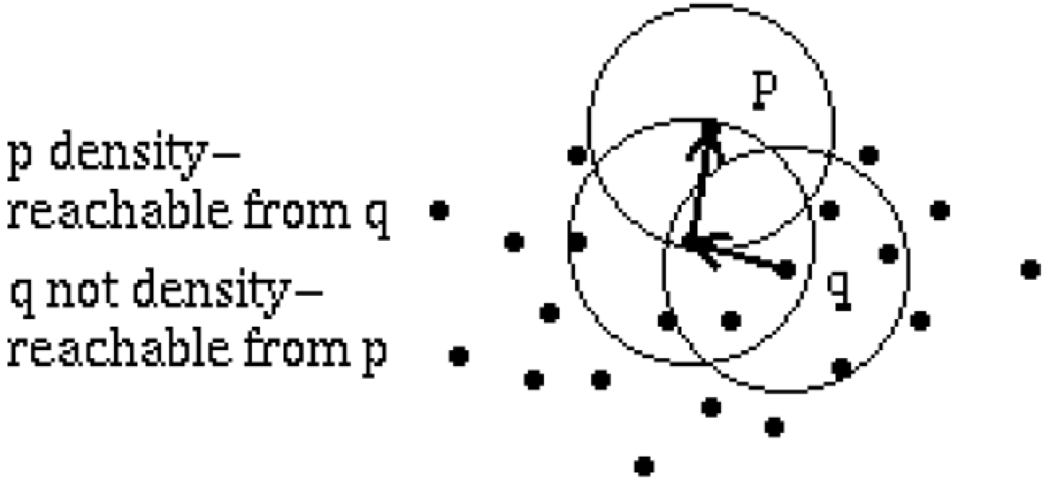
Direct reachability

- Directly density-reachable: A point p is directly density-reachable from a point q w.r.t. Eps, MinPts if
  - p belongs to $N_{Eps}(q)$ and
  - q is a core point, i.e.,: $|N_{Eps}(q)| >= MinPts$
- not a symmetric relation



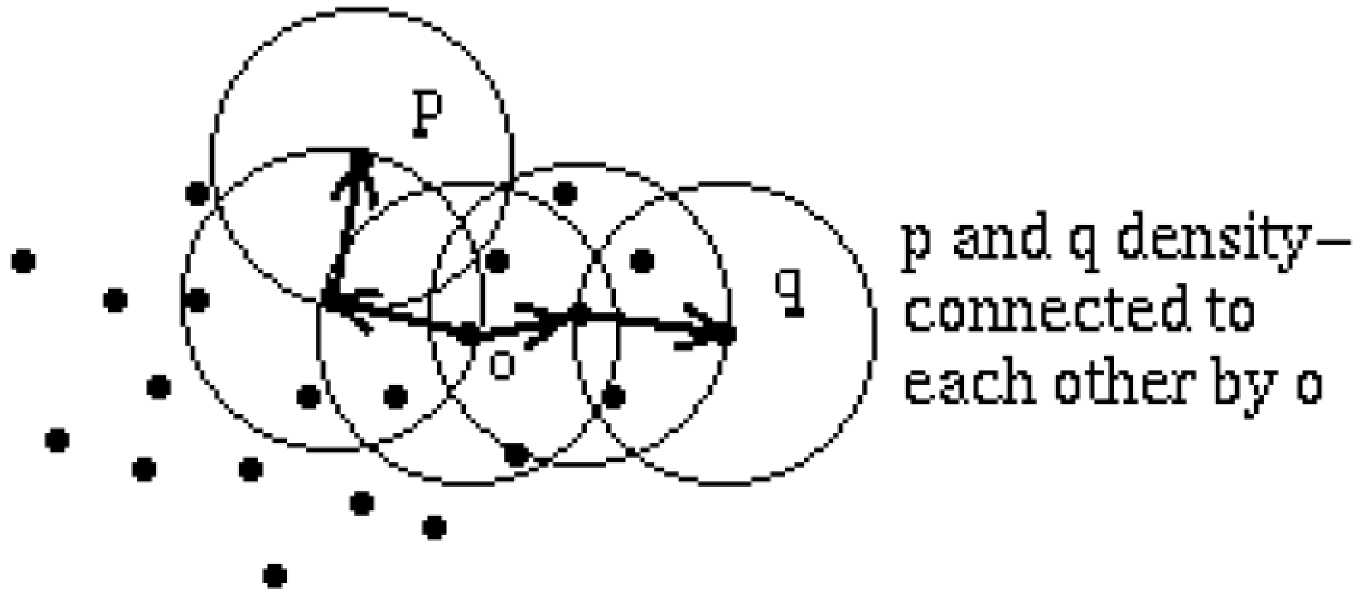
Reachability

- Density-reachable: A point p is density-reachable from a point q w.r.t. Eps, MinPts 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
- not a symmetric relation



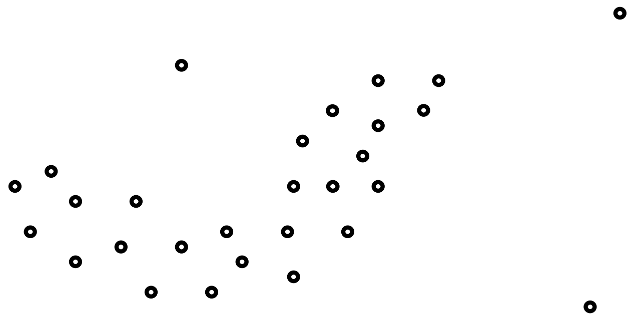
Connectivity

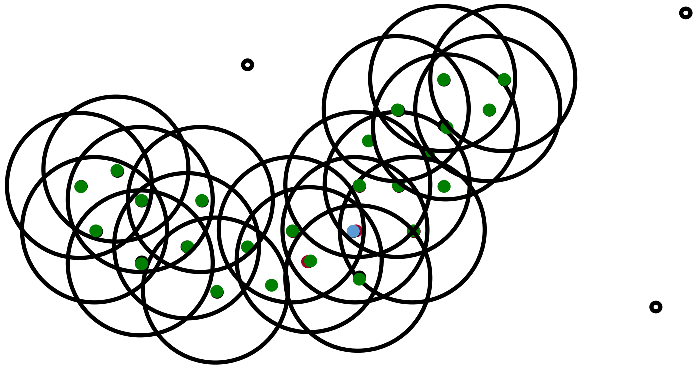
- Density-connected: A point p is density-connected to a point q w.r.t. Eps, MinPts if there is a point o such that both, p and q are density-reachable from o w.r.t. Eps and MinPts
- Density-connectedness is symmetric



Cluster

- A cluster is a maximal set of density-connected points

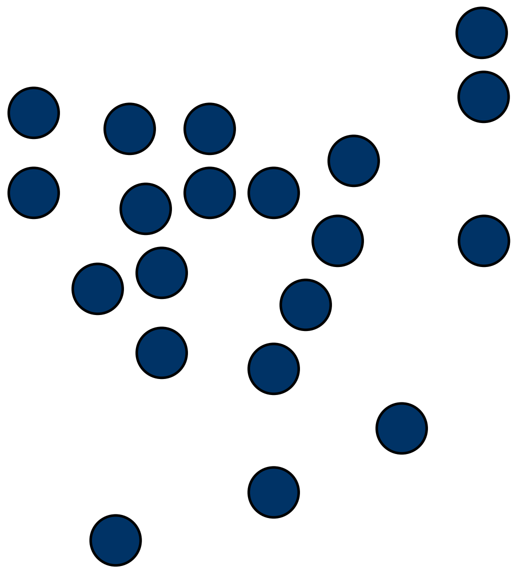




- A cluster satisfies two properties:
  - All points within the cluster are mutually density-connected.
  - If a point is density-reachable from any point of the cluster, it is part of the cluster as well.

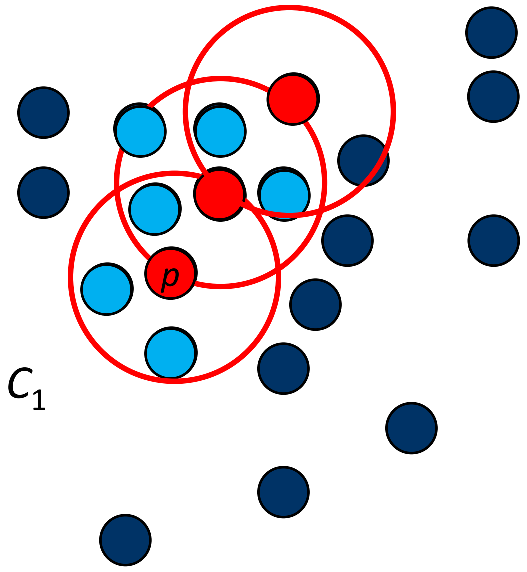
DBSCAN algorithm

- Arbitrary select a point p to start
- Retrieve all points density-reachable from p w.r.t. Eps and MinPts.
- If p is a core point, a cluster is formed starting with p and by expanding through its neighbors.
- 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.



DBSCAN algorithm

- Arbitrary select a point p to start
- Retrieve all points density-reachable from p w.r.t. Eps and MinPts.
- If p is a core point, a cluster is formed starting with p and by expanding through its neighbors.
- 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.



DBSCAN pseudocode

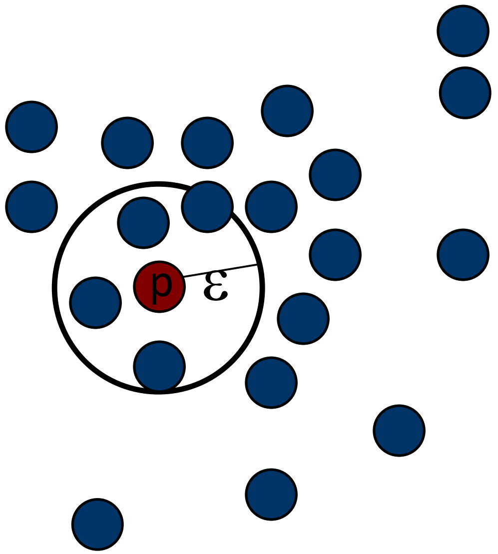
ALGORITHM 1: Pseudocode of Original Sequential DBSCAN Algorithm

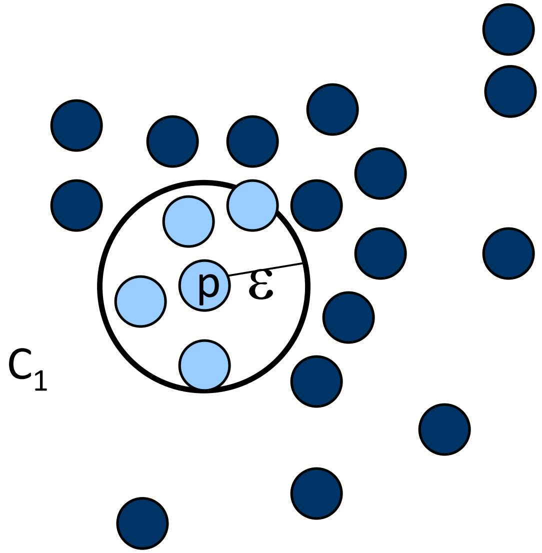
```
Input: DB: Database
Input: ε: Radius
Input: minPts: Density threshold
Input: dist: Distance function
Data: label: Point labels, initially undefined
foreach point p in database DB do                        // Iterate over every point
    if label(p) ≠ undefined then continue                // Skip processed points
    Neighbors N ← RangeQuery(DB, dist, p, ε)             // Find initial neighbors
    if |N| < minPts then                                 // Non-core points are noise
        label(p) ← Noise
        continue
    c ← next cluster label                               // Start a new cluster
    label(p) ← c
    Seed set S ← N \ {p}                                 // Expand neighborhood
    foreach q in S do
        if label(q) = Noise then  label(q) ← c
        if label(q) ≠ undefined then continue
        Neighbors N ← RangeQuery(DB, dist, q, ε)
        label(q) ← c
        if |N| < minPts then continue                    // Core-point check
        S ← S ∪ N
```

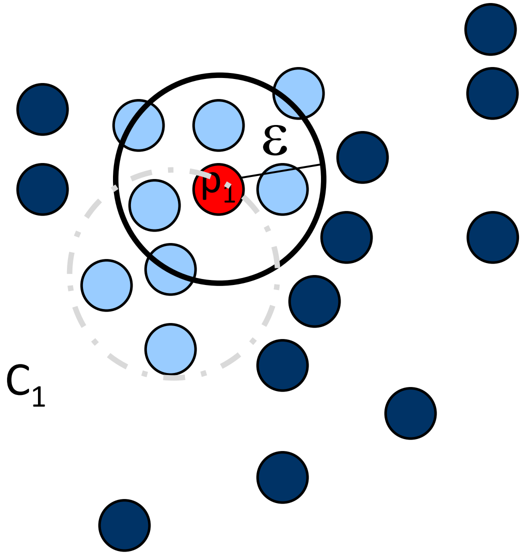
Source: https://dl.acm.org/doi/abs/10.1145/3068335

DBSCAN: An example

MinPts = 5







1. Check the ε-neighborhood of p;
2. If p has less than MinPts neighbors then mark p as outlier and continue with the next object
3. Otherwise mark p as processed and put all the neighbors in cluster C_1

1. Check the unprocessed objects in C_1
2. If no core object, return C_1
3. Otherwise, randomly pick up one core object p_1, mark p_1 as processed, and put all unprocessed neighbors of p_1 in cluster C_1

Source: http://www.cse.buffalo.edu/faculty/azhang/cse601/density-based.ppt

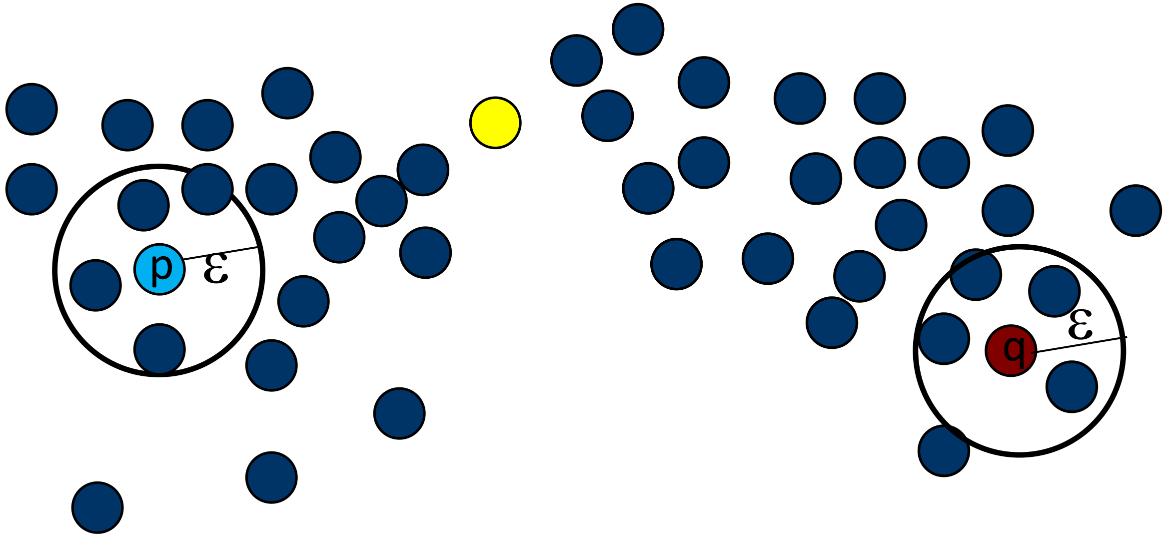
Short break (5’)

Is the result of DBSCAN dependent on the order in which we visit the data?

- Think for 1’
- Discuss with your neighbours
- Discuss in the class

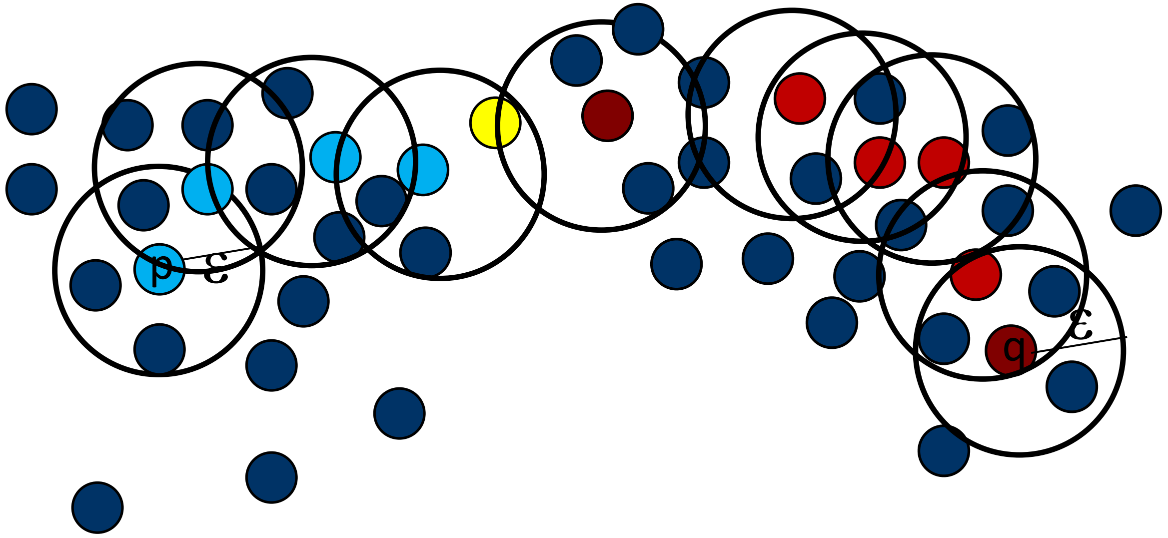
Does the processing order affect the clustering result?

MinPts = 5

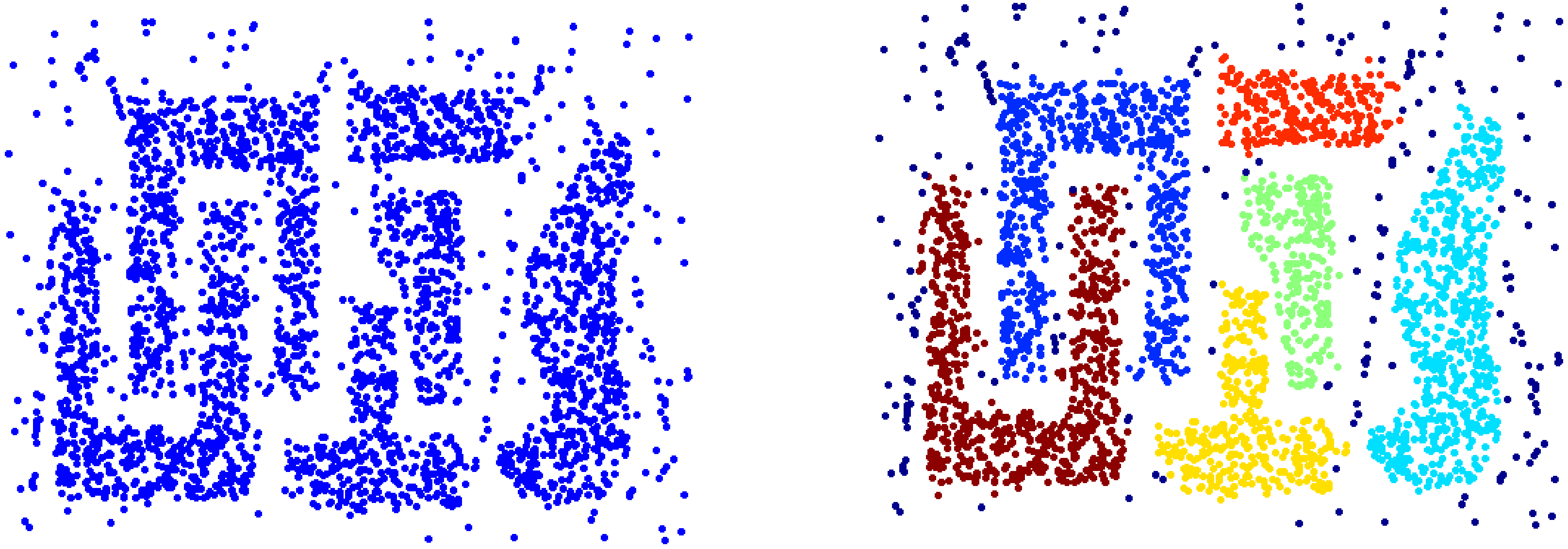


Border points might change cluster membership depending on processing order

MinPts = 5



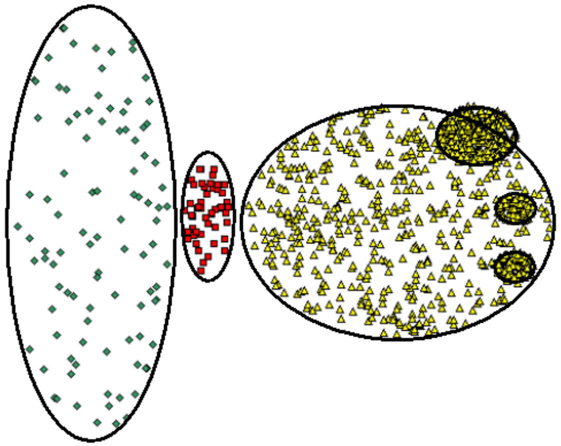
When DBSCAN works well?

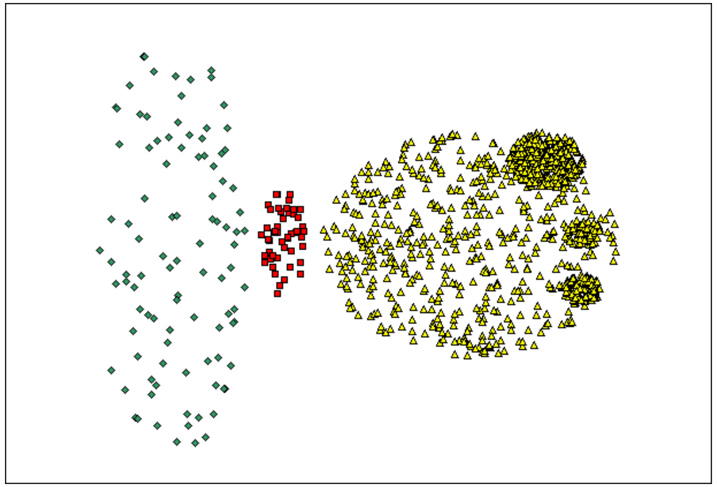


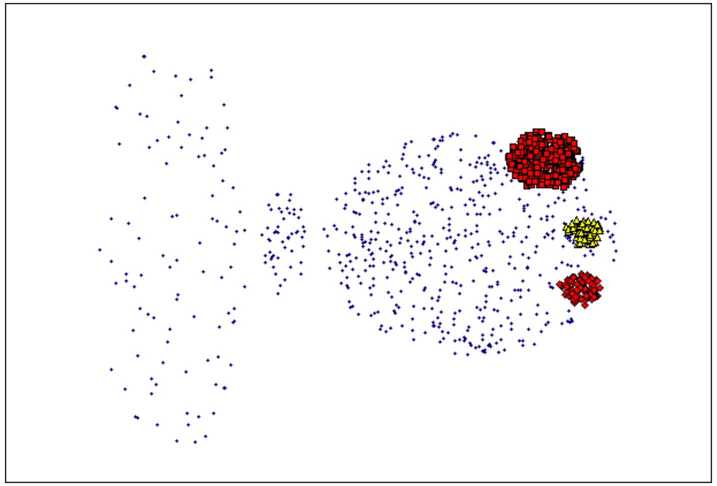
Clusters

- Resistant to noise
- Can handle clusters of different shapes and sizes

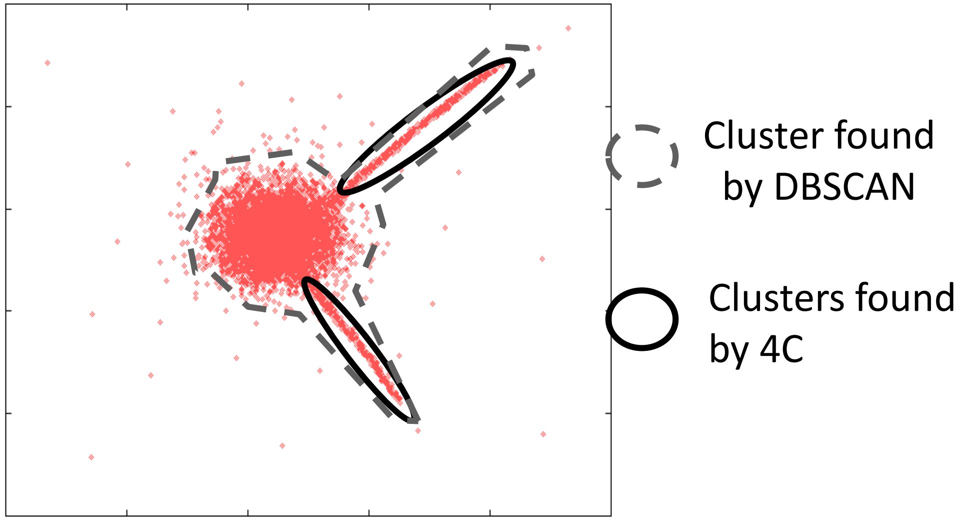
When DBSCAN does not work well?







Original points



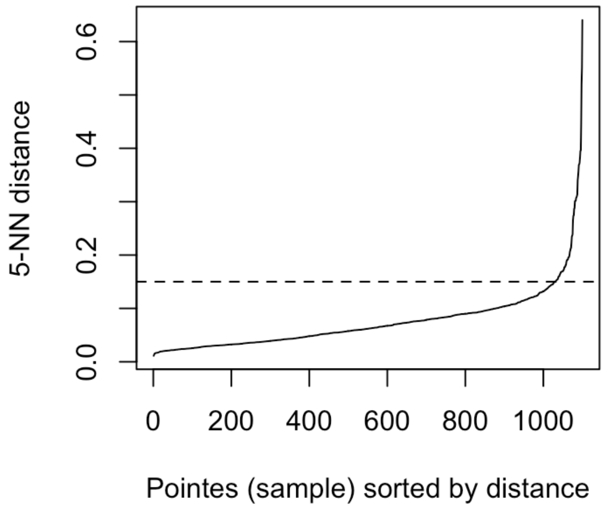
- DBSCAN fails to identify clusters of varying densities
- Problems in high-dimensional data due to curse of dimensionality

DBSCAN: determining Eps and MinPts

- Intuition
  - for points in a cluster, their k^{th} nearest neighbors are at roughly the same distance
  - whereas noise points have the k^{th} nearest neighbor at farther distance

- So, the idea is to calculate, the distance of every point to its *k* nearest neighbor. The value of *k* will be specified by the user and corresponds to MinPts.

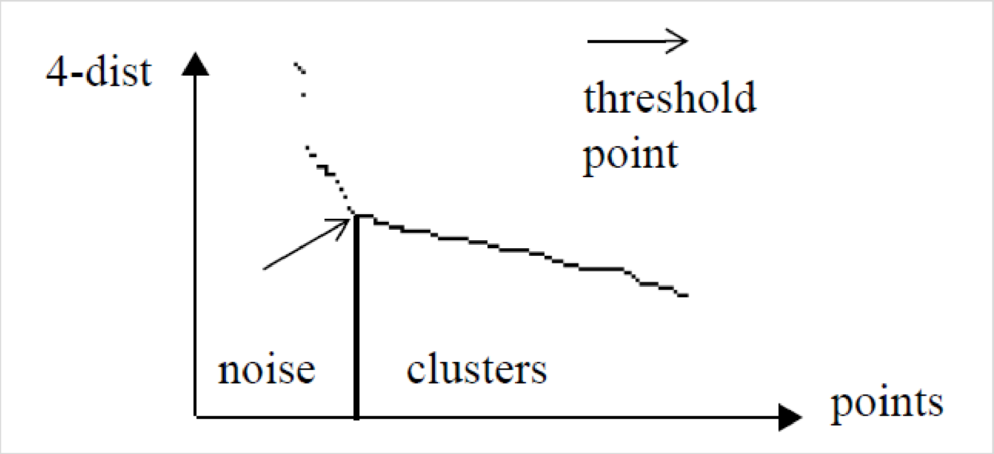
- Next, these k-distances are plotted in an ascending order. The aim is to determine the "knee", which corresponds to the optimal *eps* parameter.
  - A knee corresponds to a threshold where a sharp change occurs along the *k*-distance curve."



Source: http://www.sthda.com/english/wiki/dbscan-density-based-clustering-for-discovering-clusters-in-large-datasets-with-noise-unsupervised-machine-learning

DBSCAN: determining Eps and MinPts

Ordering points to identify the clustering structure (OPTICS algorithm)



The sorted k-dist graph

All points with a higher *k*-dist value (left of the threshold) are considered to be noise, all other points (right of the threshold) are assigned to some cluster.

From the DBSCAN paper: "our experiments indicate that the k-dist graphs for k > 4 do not significantly differ from the 4-dist graph and, furthermore, they need considerably more computation. Therefore, we eliminate the parameter MinPts by setting it to 4 for all databases (for 2-dimensional data)."

Short break (5’)

What is the complexity of DBSCAN?

- Think for 1’
- Discuss with your neighbours
- Discuss in the class

Complexity

- For a dataset *D* consisting of *n* points, the time complexity of DBSCAN is
  - O(*n* * time to find points in the Eps-neighborhood)

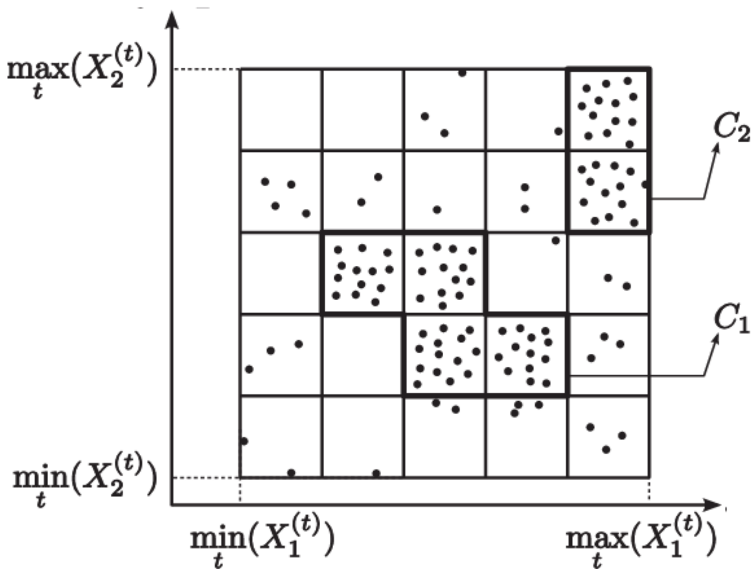
- Worst case $O(n^2)$

- In low-dimensional spaces *O(nlogn)*;
  - efficient data structures (e.g., kd-trees) allow for efficient retrieval of all points within a given distance of a specified point

Outline

- Density-based clustering basics
- DBSCAN
- Grid-based clustering (shortly)
- Things you should know from this lecture & reading material

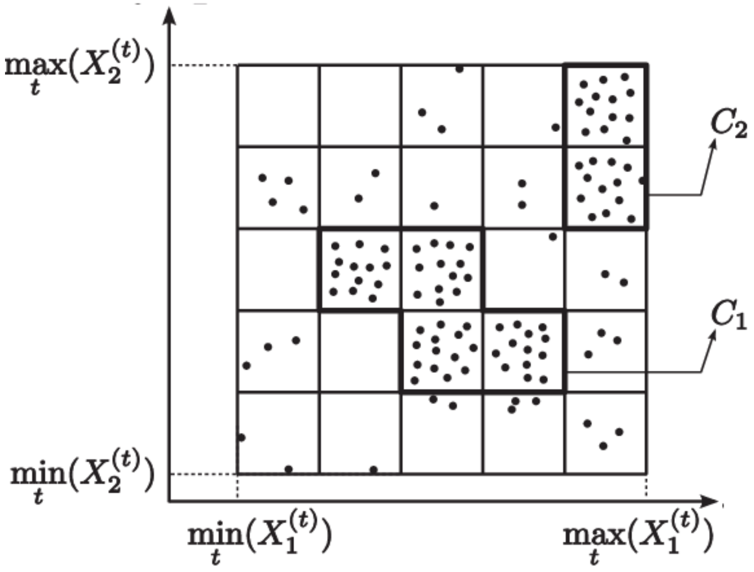
Density based on grid

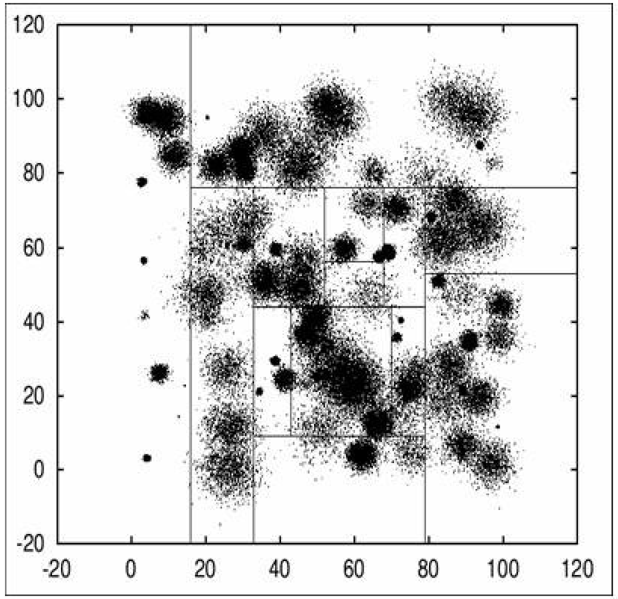


- A grid structure is used to capture the density of the dataset.
- Density is measured locally in each grid cell
  - Density = number of points within each cell
- A cluster is a set of connected dense cells
  - Dense cells are first identified
  - Neighboring dense cells form clusters
  - Similarly to DBSCAN, a cluster is a maximal set of connected dense cells

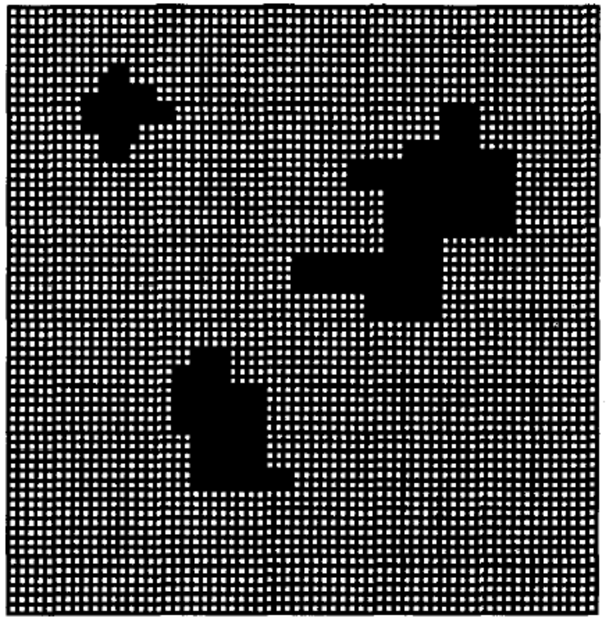
Density based on grid

- Clustering depends on the grid structure
- Grid parameters (cell size and density) are required
  - Typically global parameters → fixed-grid approaches
- Adaptive-grid approaches also exist





Grid-based methods



- A variety of algorithms
    - STING (VLDB'97), WaveCluster (VLDB'98),…
    - CLIQUE (SIGMOD'98) for high-dimensional data
- Appealing features
    - No assumption on the number of clusters
    - Discovering clusters of arbitrary shapes
    - Ability to handle outliers
- But, as already mentioned
    - The result depends on the grid parameters (cell size and cell density, which are typically global)
        - Approaches exist for adaptive size grids

Outline

- Hierarchical clustering basics
- Hierarchical clustering methods
- Bisecting k-Means
- Things you should know from this lecture & reading material

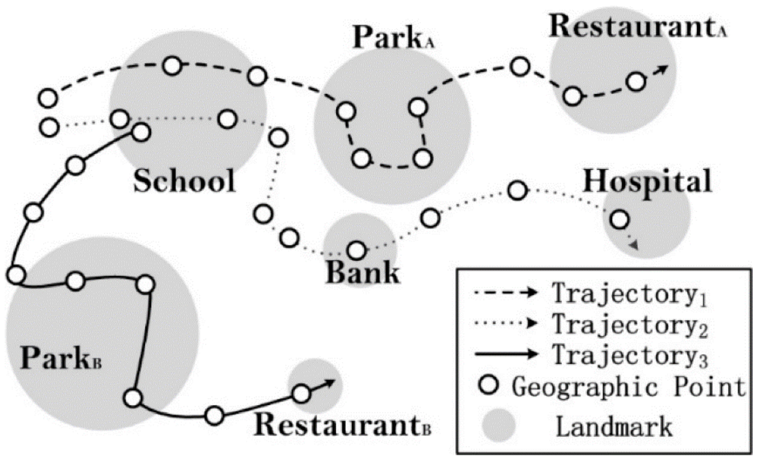
Overview and Reading

- Overview
  - Density-based clustering
    - DBSCAN
    - Core, border, noisy points
  - Grid-based clustering basics

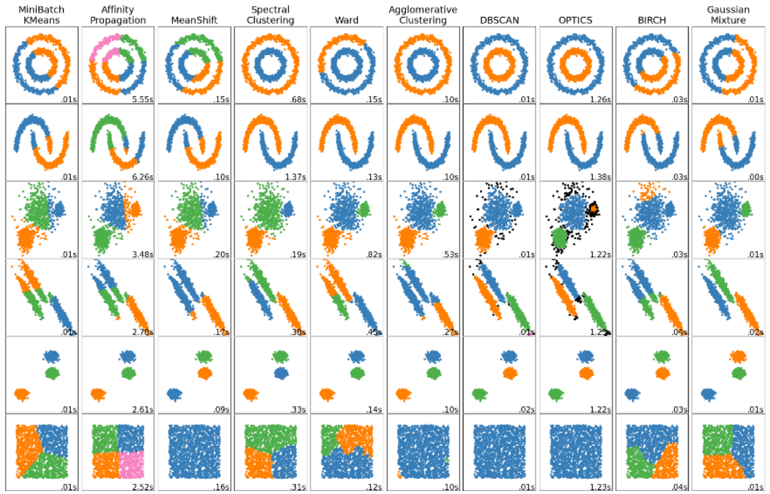
- Reading
  - Tan P.-N., Steinbach M., Kumar V book, Chapter 8.
  - Data Clustering: A Review, https://www.cs.rutgers.edu/~mlittman/courses/lightai03/jain99data.pdf
  - Nando de Freitas youtube video: https://www.youtube.com/watch?v=voN8omBe2r4

Hands on experience

- Try density-based clustering on mobility data (you can use your own mobility data)
  - Do you recognize any clusters in your activities?
    - E.g., going to University, out and about ….
- Or, existing GPS trajectory data
  - E.g., Geolife GPS trajectory dataset
- Or, try toy datasets from scikit-learn



Source: https://www.mdpi.com/2220-9964/6/7/212/htm



Thank you

Questions/Feedback/Wishes?

Acknowledgements

- The slides are based on
  - KDD I lecture at LMU Munich (Johannes Aßfalg, Christian Böhm, Karsten Borgwardt, Martin Ester, Eshref Januzaj, Karin Kailing, Peer Kröger, Eirini Ntoutsi, Jörg Sander, Matthias Schubert, Arthur Zimek, Andreas Züfle)
  - Introduction to Data Mining book slides at http://www-users.cs.umn.edu/~kumar/dmbook/
  - Thank you to all TAs contributing to their improvement, namely Vasileios Iosifidis, Damianos Melidis, Tai Le Quy, Han Tran.