Leibniz
Universität
Hannover

Fakultät für Elektrotechnik und Informatik
Institut für Verteilte Systeme
AG Intelligente Systeme - Data Mining group

Data Mining I

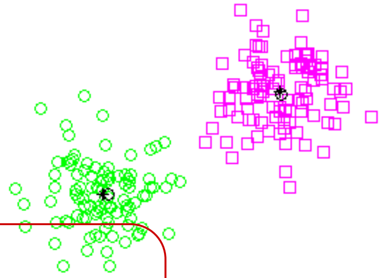
Summer semester 2019

Lecture 10: Clustering – 2: Density-based clustering

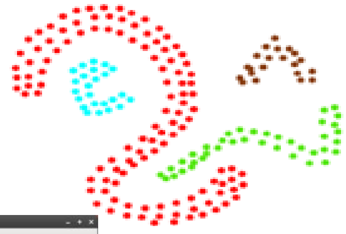
Lectures: Prof. Dr. Eirini Ntoutsi

TAs: Tai Le Quy, Vasileios Iosifidis, Maximilian Idahl, Shaheer Asghar

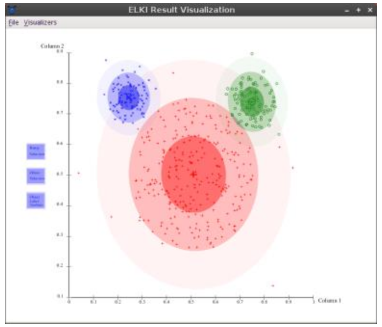
Clustering topics covered in DM1



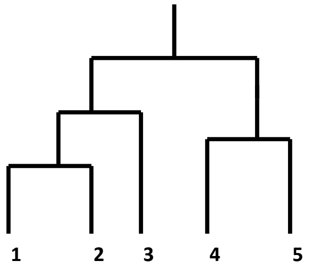
1. Partitioning-based clustering
   - kMeans, kMedoids
2. Density-based clustering
   - DBSCAN



3. Model-based clustering
   - EM



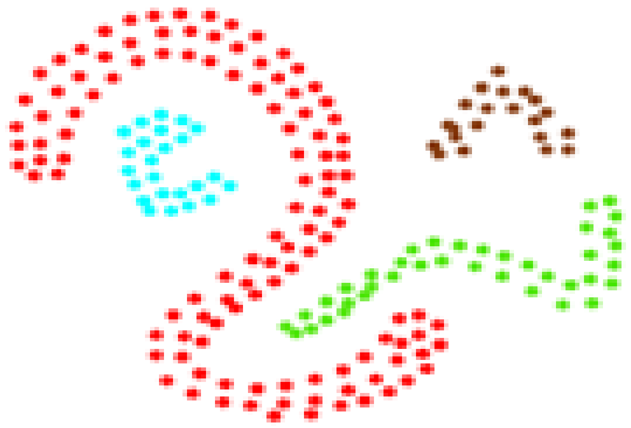
4. Hierarchical clustering



5. Clustering evaluation

Density based clustering

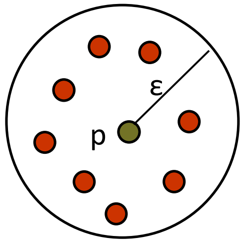
- Clusters are regions of high density surrounded by regions of low density (noise)
- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need density parameters as termination condition



- 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

- Density:
  - 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)



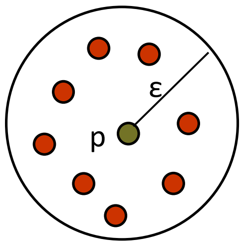
The ε-neighborhood of *p*: 9 points

- 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)

DBSCAN basic concepts

- Consider a dataset *D* of objects to be clustered
- Two parameters*:*
  - Eps (or ε): Maximum radius of the neighbourhood
  - MinPts: Minimum number of points in an Eps-neighbourhood of that point

- Eps-neighborhood of a point p in D
  - $N_{Eps}(p)$: $\{q$ belongs to $D \mid dist(p,q) <= Eps\}$



The Eps-neighborhood of p

Core points vs border points vs noise points

- Let D be a dataset. Given a radius parameter *Eps* and a density parameter *MinPts* we can distinguish between:
  - 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.,:

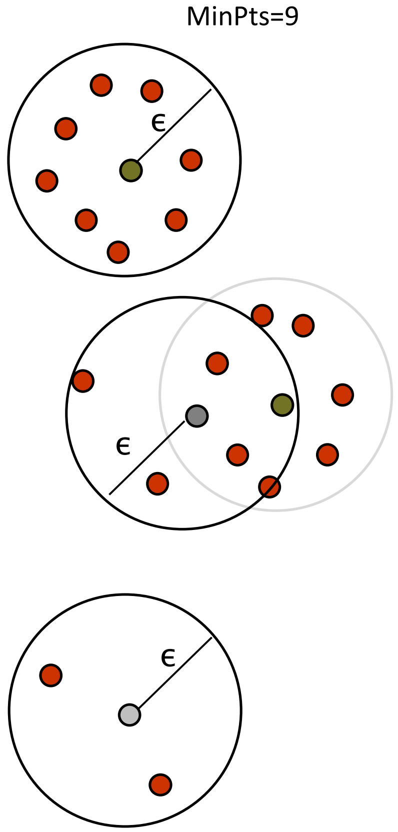
    $$|N_{Eps}(p)=\{q \mid dist(p,q) <= Eps\}| \geq MinPts$$

    - 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

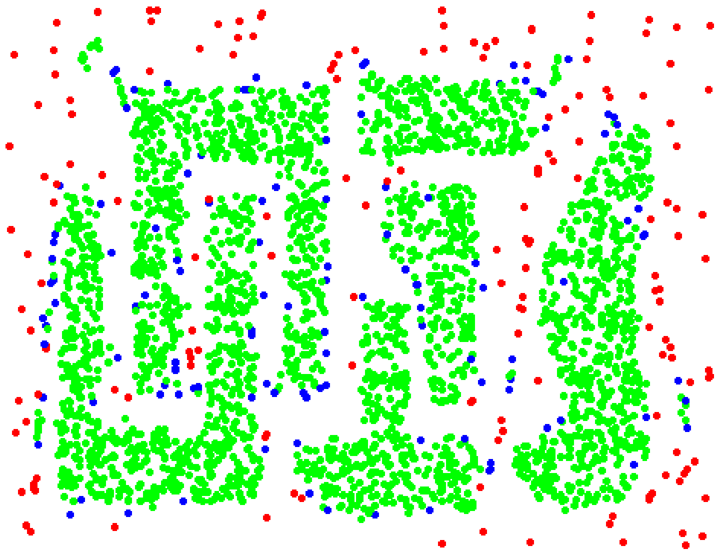
    - those are points that belong to the periphery of a cluster
  - Noise points

    neither a core point nor a border point



Core, Border and Noise points

Original points



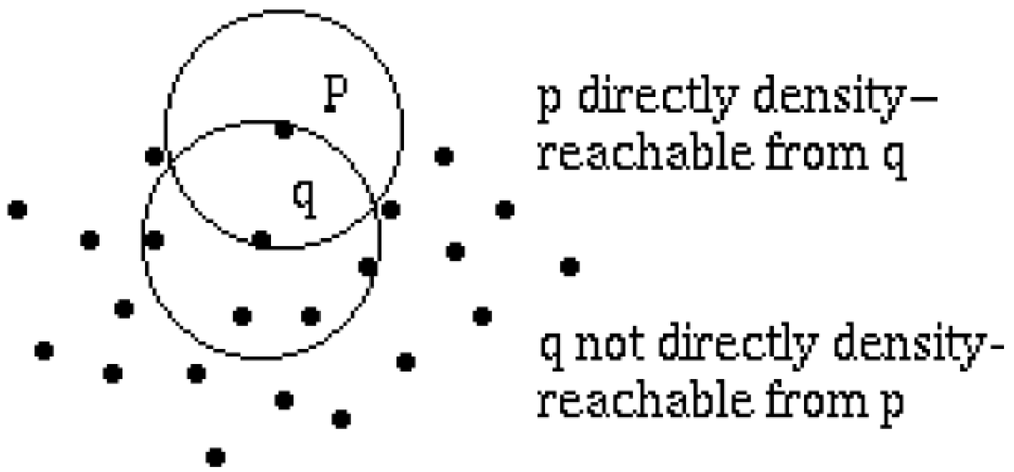
Point types: core, border and noise

Eps = 10, MinPts = 4

- 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

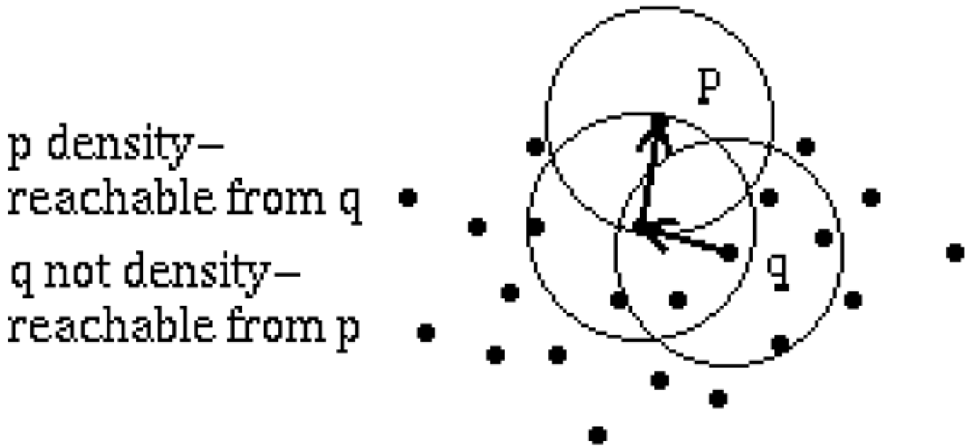
Direct reachability

- 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.,: $|N_{Eps}(q)| >= MinPts$



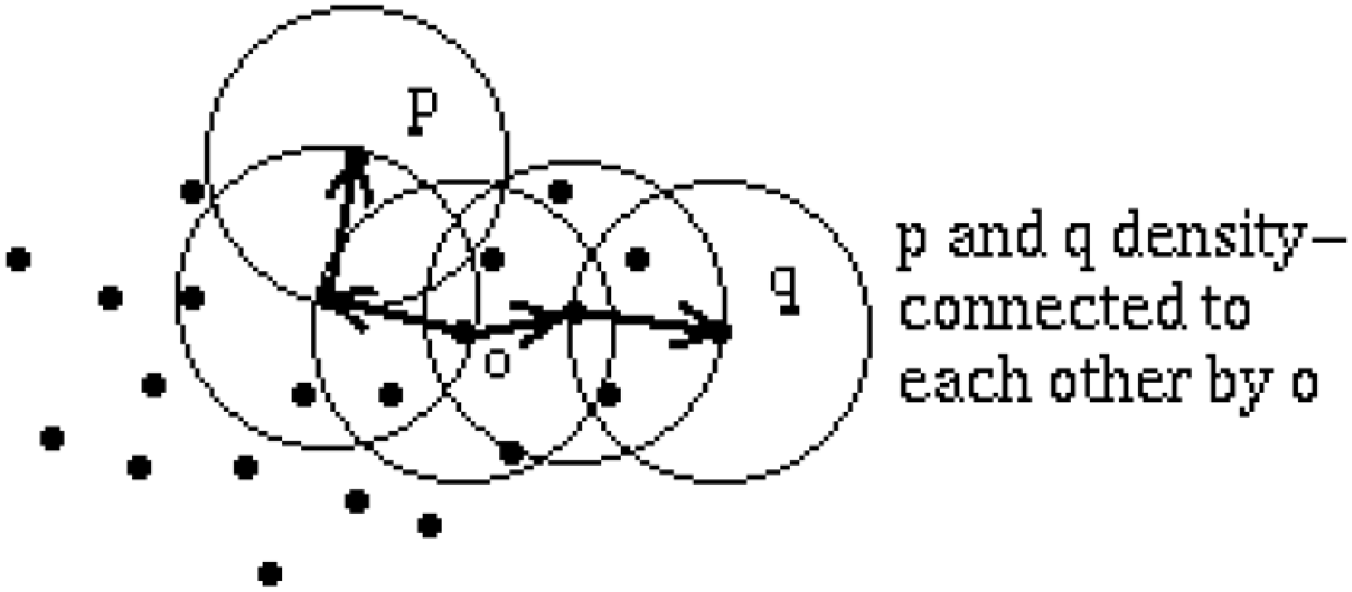
Reachability

- 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



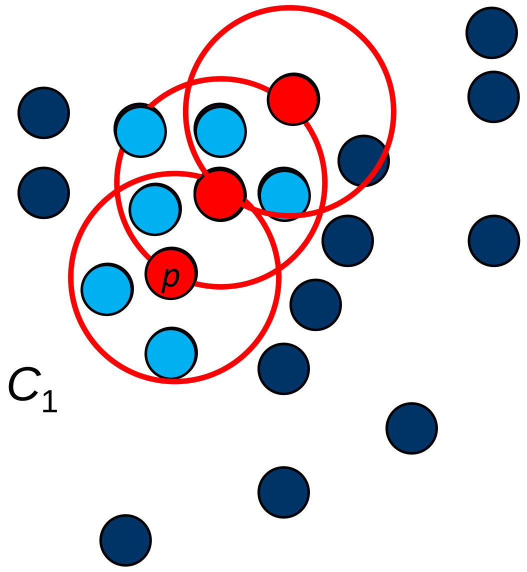
Cluster

- 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 pseudocode I

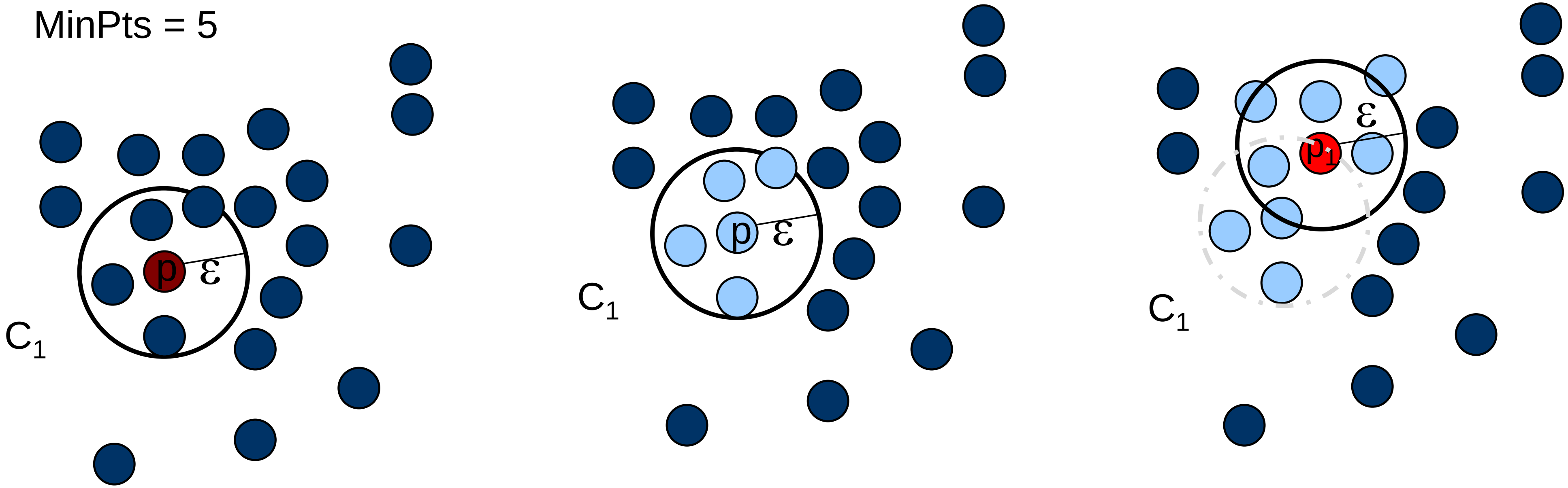
```
DBSCAN(Dataset DB, Real Eps, Integer MinPts)
  // initially all objects are unclassified,
  // o.ClId = unclassified for all o ∈ DB

  ClusterId := nextId(NOISE);
  for i from 1 to |DB| do
    Object := DB.get(i);
    if Object.ClId = unclassified then
      if ExpandCluster(DB, Object, ClusterId, Eps, MinPts)
      then ClusterId:=nextId(ClusterId);
```

DBSCAN pseudocode II

```
ExpandCluster(DB, StartObject, ClusterId, Eps, MinPts): Boolean
 seeds:= RQ(StartObjekt, Eps);
 if |seeds| < MinPts then // StartObject is not a core object
    StartObject.ClId := NOISE;
        return false;
 else // else: StartObject is a core object
        forall o ∈ seeds do o.ClId := ClusterId;
    remove StartObject from seeds;
    while seeds ≠ Empty do
        select an object o from the set of seeds;
                Neighborhood := RQ(o, Eps);
                if |Neighborhood| ≥ MinPts then // o is a core object
                           for i from 1 to |Neighborhood| do
                               p := Neighborhood.get(i);
                               if p.ClId in {UNCLASSIFIED, NOISE} then
                                  if p.ClId = UNCLASSIFIED then
                                     add p to the seeds;
                                    p.ClId := ClusterId;
                end if;
            end for;
                end if;
                remove o from the seeds;
    end while;
 end if
return true;
```

DBSCAN: An example

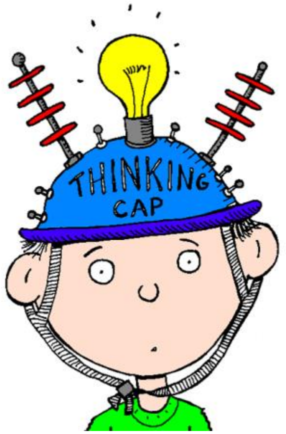


1. Check the ε-neighborhood of p;
2. If p has less than MinPts neighbors then mark p as outlier and continue with the next object
3. Otherwise mark p as processed and put all the neighbors in cluster C_1

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

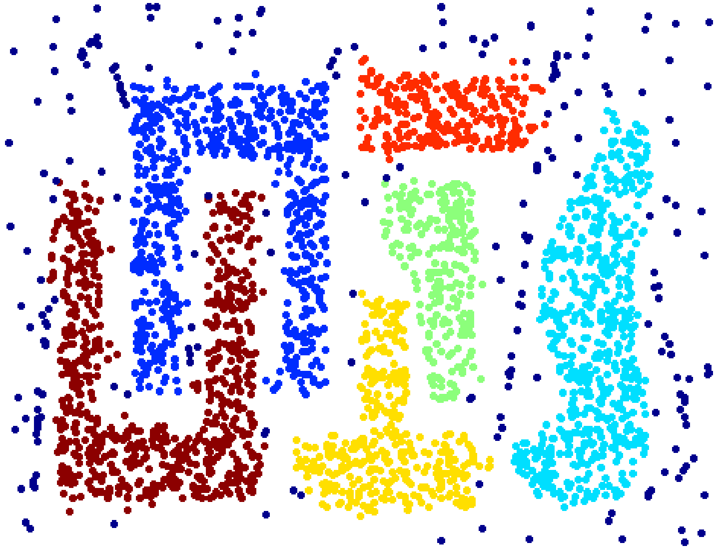
Short break (5’)



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

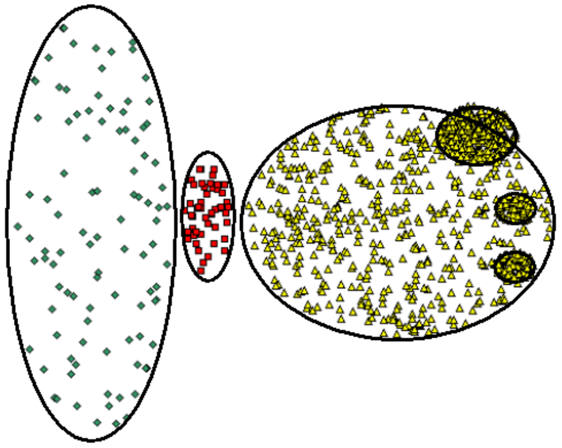
When DBSCAN works well?



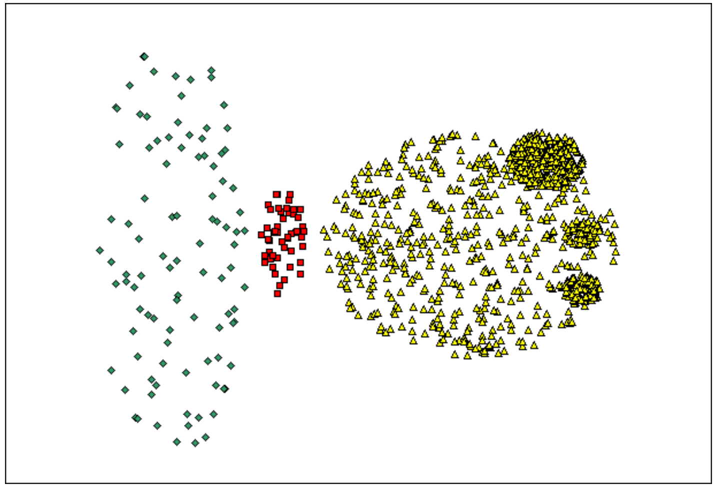
Clusters

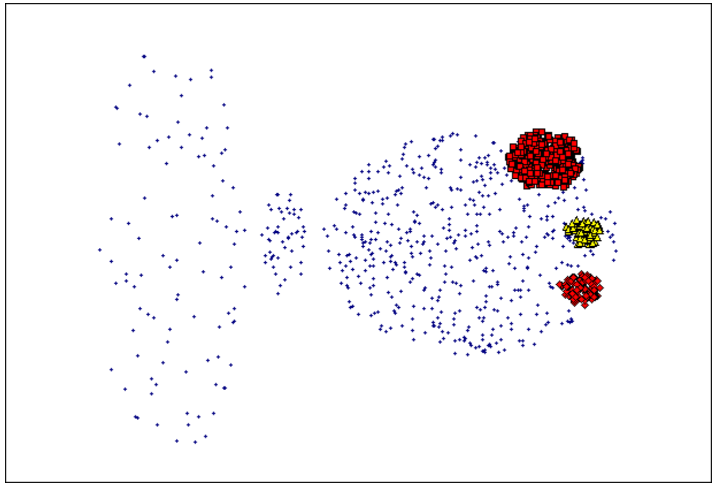
- Resistant to noise
- Can handle clusters of different shapes and sizes

When DBSCAN does not work well?



Original points





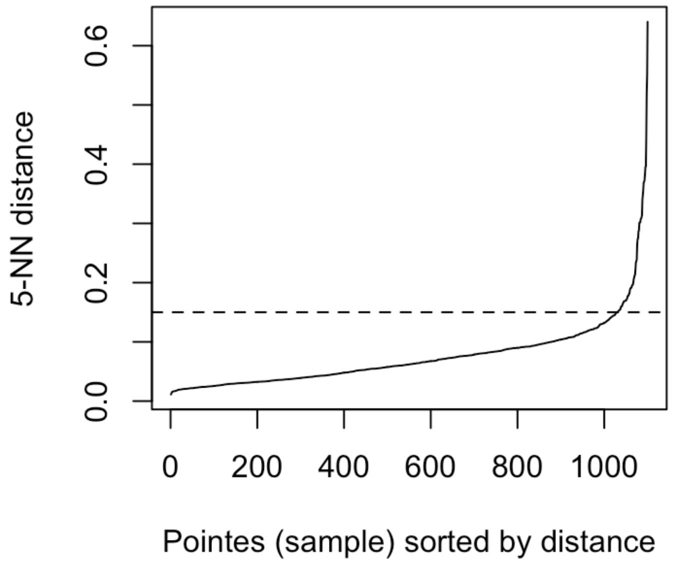
- DBScan can fail to identify clusters of varying densities
- Problems in high-dimensional data due to curse of dimensionality

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.

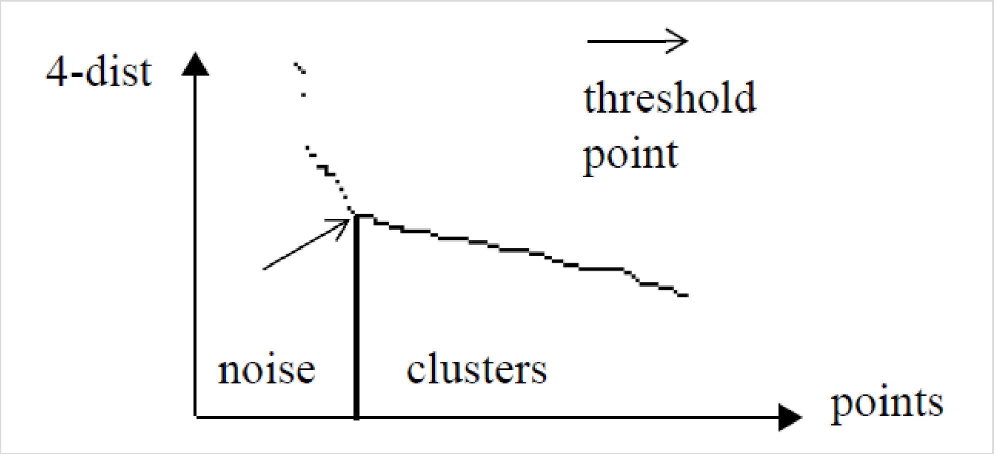
- Next, these k-distances are plotted in an ascending order. The aim is to determine the "knee", which corresponds to the optimal *eps* parameter.
  - A knee corresponds to a threshold where a sharp change occurs along the *k*-distance curve."



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 (3’)

What is the complexity of DBSCAN?

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

Complexity

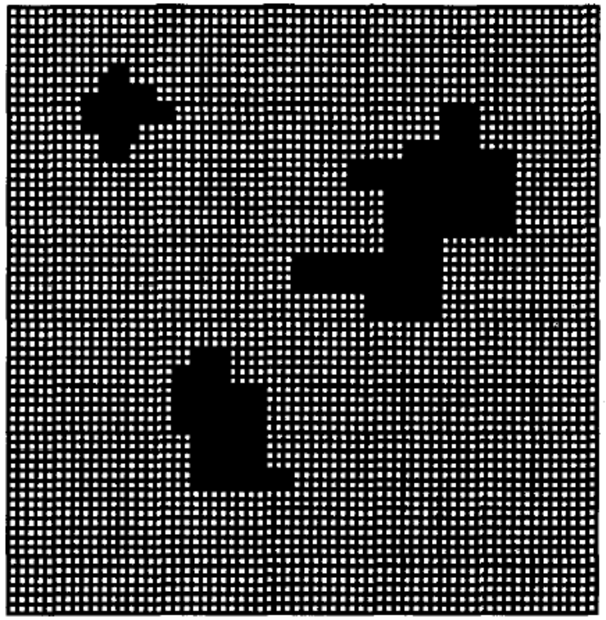
- For a dataset *D* consisting of *n* points, the time complexity of DBSCAN is
  - *O(n x time to find points in the Eps-neighborhood)*
- Worst case $O(n^2)$
- In low-dimensional spaces *O(nlogn)*;
  - efficient data structures (e.g., *kd-trees*) allow for efficient retrieval of all points within a given distance of a specified point

Things you should know from this lecture

- Density-based clustering
- DBSCAN
- Core, border, noisy points

Grid-based methods

- Another density-based clustering approach.
- A grid structure is used to capture the density of the dataset.
  - A cluster is a set of connected dense cells
    - 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
  - The result depends on the grid parameters (cell size and cell density, which are typically global)
    - Approaches exist for dynamic size grids



Homework/ tutorial

- Homework
  - Try DBSCAN (e.g., in ELKI: https://elki-project.github.io/howto/clustering, SciKit: http://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html or, write your own implementation) using your own GPS data for 1 week, 1 month etc
    - Are there any clear patterns in your data?
- Readings:
  - Tan P.-N., Steinbach M., Kumar V book, Chapter 8. Also online: https://www-users.cs.umn.edu/~kumar/dmbook/ch8.pdf
  - The original DBSCAN paper at KDD96, https://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf