



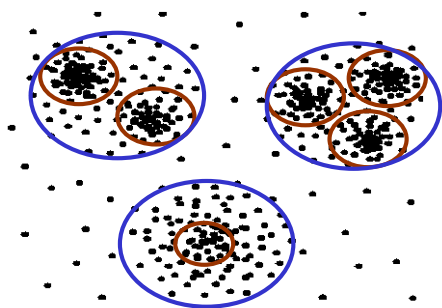
Big Data Analytics

– Week 3: Density-Based Hierarchical Clustering –

Knowledge Discovery and Data Mining
Prof. Dr. Emmanuel Müller

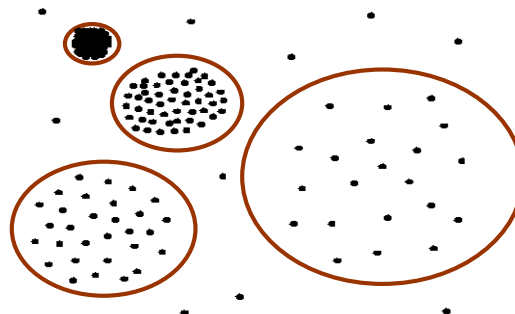
From Partitioning to Hierarchical Clustering

- Global parameters to separate all clusters with a partitioning clustering method may not exist



*hierarchical
cluster structure*

and/or

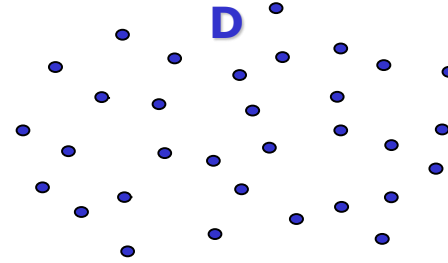
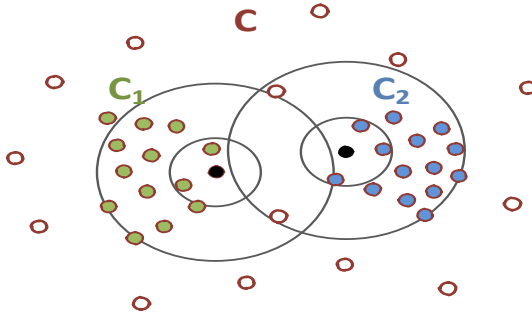


*largely differing
densities and sizes*

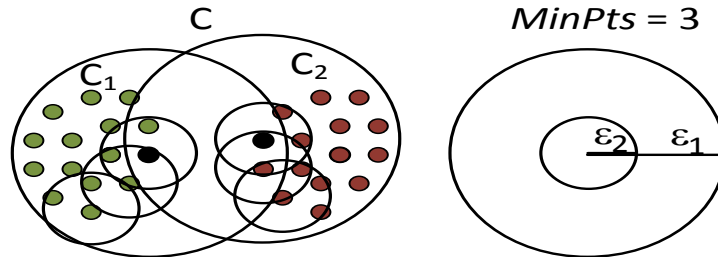
- Need a hierarchical clustering algorithm in these situations

Density-Based Hierarchical Clustering

Observation: Dense clusters are completely contained by less dense clusters



Idea: Process objects in the “right” order and keep track of point density in their neighborhood



The Algorithm OPTICS

Ankerst M., Breunig M., Kriegel H.-P., Sander J.: „OPTICS: Ordering Points To Identify the Clustering Structure“, *In SIGMOD 1999*, pp. 49-60.

OPTICS: **O**rdering **P**oints **T**o **I**dentify the **C**lustering **S**tructure

Basic data structure: controllList

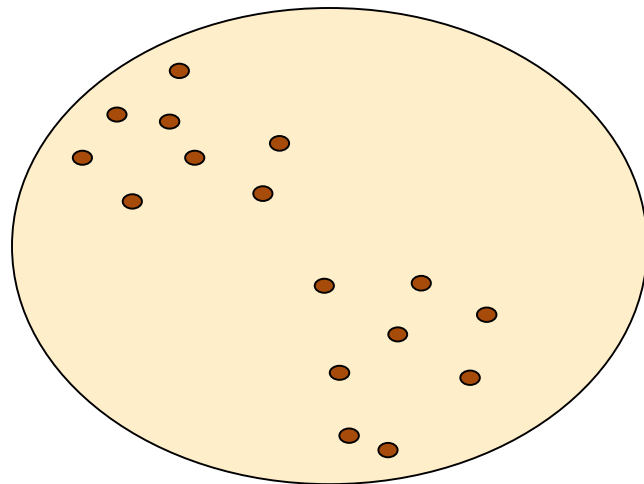
- Memorize shortest reachability distances seen so far (“distance of a jump to that point”)

Visit each point

- Make always a shortest jump

Output:

- order of points
- core-distance of points
- reachability-distance of points



Core Distance and Reachability Distance

Parameters: “generating” distance ε , fixed value $MinPts$

core-distance _{$\varepsilon, MinPts$} (o)

“smallest distance such that o is a core object”
(if that distance is $\leq \varepsilon$; “?” otherwise)

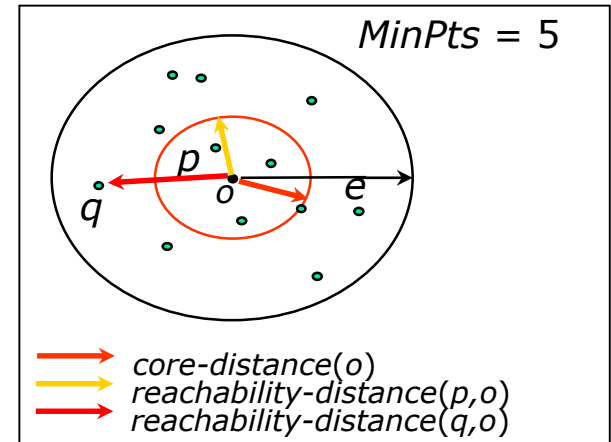
reachability-distance _{$\varepsilon, MinPts$} (p, o)

“smallest distance such that p is
directly density-reachable from o ”
(if that distance is $\leq \varepsilon$; “?” otherwise)

$$reach-dist(p, o) = \begin{cases} dist(p, o) & \Leftrightarrow dist(p, o) > core-dist(o) \\ core-dist(o) & \Leftrightarrow dist(p, o) < core-dist(o) \\ undef & \Leftrightarrow dist(p, o) > \varepsilon \end{cases}$$

value

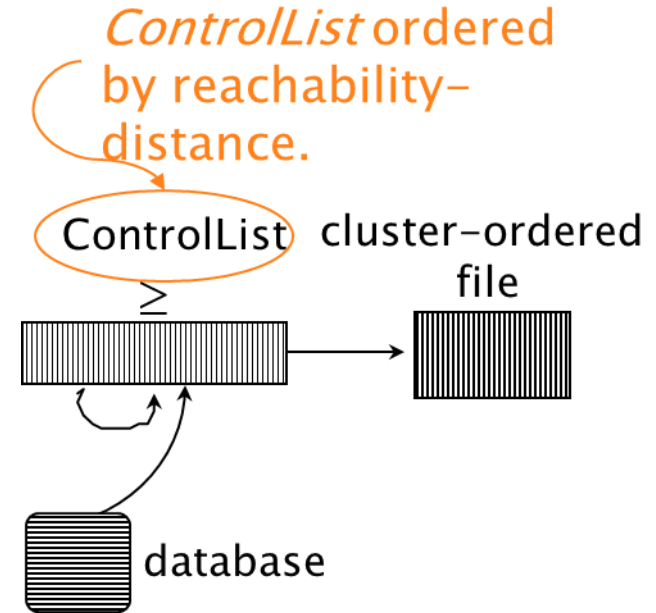
condition



The Algorithm OPTICS

```

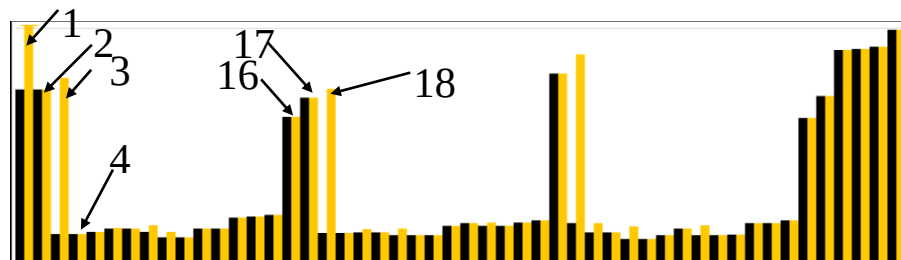
foreach  $o \in \text{Database}$ 
  // initially,  $o.\text{processed} = \text{false}$  for all objects  $o$ 
  if  $o.\text{processed} = \text{false}$ ;
    insert ( $o$ , “?”) into ControlList;
  while ControlList is not empty
    select first element ( $o$ ,  $r\_dist$ ) from ControlList;
    retrieve  $N_\epsilon(o)$  and determine  $c\_dist = \text{core-distance}(o)$ ;
    set  $o.\text{processed} = \text{true}$ ;
    write ( $o$ ,  $r\_dist$ ,  $c\_dist$ ) to file;
    if  $o$  is a core object at any distance  $\leq \epsilon$ 
      foreach  $p \in N_\epsilon(o)$  not yet processed;
        determine  $r\_dist_p = \text{reachability-distance}(p, o)$ ;
        if  $(p, \_) \notin \text{ControlList}$ 
          insert ( $p$ ,  $r\_dist_p$ ) in ControlList;
        else if  $(p, \text{old\_}r\_dist) \in \text{ControlList}$  and  $r\_dist_p < \text{old\_}r\_dist$ 
          update ( $p$ ,  $r\_dist_p$ ) in ControlList;
  
```



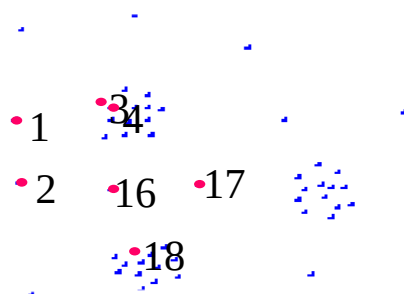
OPTICS: Properties

“Flat” density-based clusters wrt. $\varepsilon^* \leq \varepsilon$ and *MinPts* afterwards:

- Starts with an object o where $c\text{-dist}(o) \leq \varepsilon^*$ and $r\text{-dist}(o) > \varepsilon^*$
- Continues while $r\text{-dist} \leq \varepsilon^*$



■ Core-distance ■ Reachability-distance

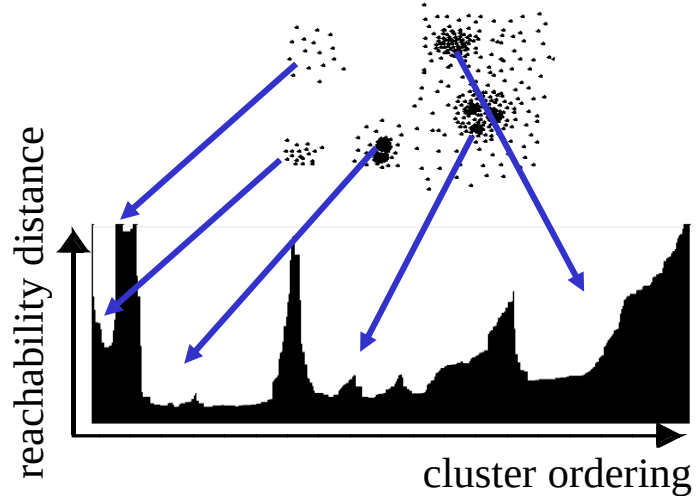
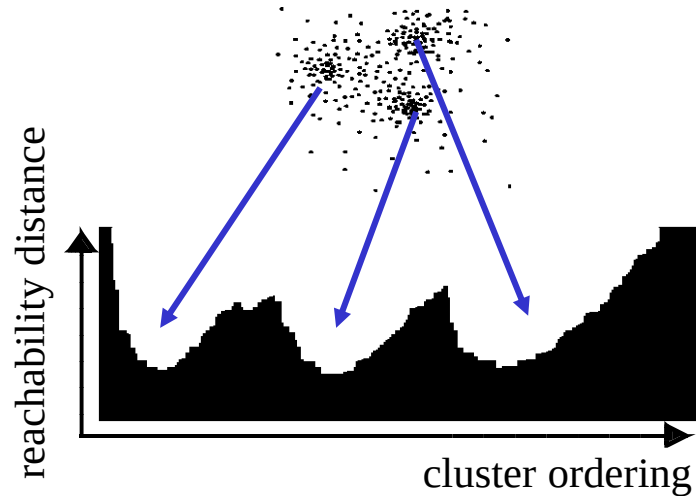


Performance: approx. runtime(DBSCAN(ε , *MinPts*))

- $O(n * \text{runtime}(\varepsilon\text{-neighborhood-query}))$
 - without spatial index support (worst case): $O(n^2)$
 - e.g. tree-based spatial index support: $O(n * \log(n))$

OPTICS: The Reachability Plot

- represents the density-based clustering structure
- easy to analyze
- independent of the dimension of the data



Hierarchical Clustering: Discussion

Advantages

- Does not require the number of clusters to be known in advance
- No (standard methods) or very robust parameters (OPTICS)
- Computes a complete hierarchy of clusters
- Good result visualizations integrated into the methods
- A “flat” partition can be derived afterwards
(e.g. via a cut through the dendrogram or the reachability plot)

Disadvantages

- May not scale well
 - Runtime for the standard methods: $O(n^2 \log n^2)$
 - Runtime for OPTICS: without index support $O(n^2)$
- User has to choose the final clustering