Unsupervised learning

Data Science, AU, Fall 2023

Ira Assent

Where we are…

1. 11/10 Introduction, data preprocessing, PCA, clustering
2. 13/10 Unsupervised learning, more clustering, outlier detection
3. 23/10 Supervised learning, classical machine learning: DT, SVMs,…
4. 26/10 Neural networks, pitfalls, outlook

- We looked at data science basics
  - Data preprocessing: study your data, bring it to good shape before you run an analysis!
  - Principle components analysis: reduce dimensionality, retain most information
- Clustering
  - Grouping of data based on mutual similarity: pick appropriate similarity measures!
  - K-means / EM: group around representative (mean, medoid), iteratively update random initialization
    - Choice of parameter k: do proper evaluation, e.g. silhouette coefficient
- Worked with scikit-learn and pandas

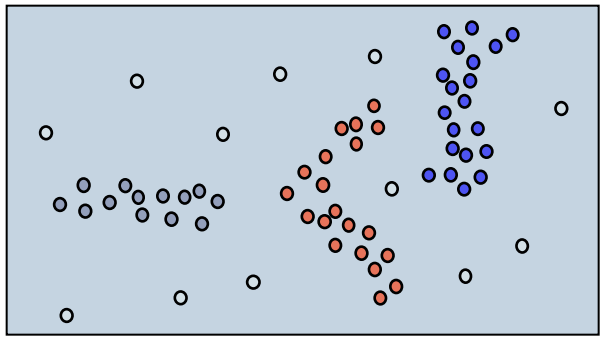
Any comments? Questions?

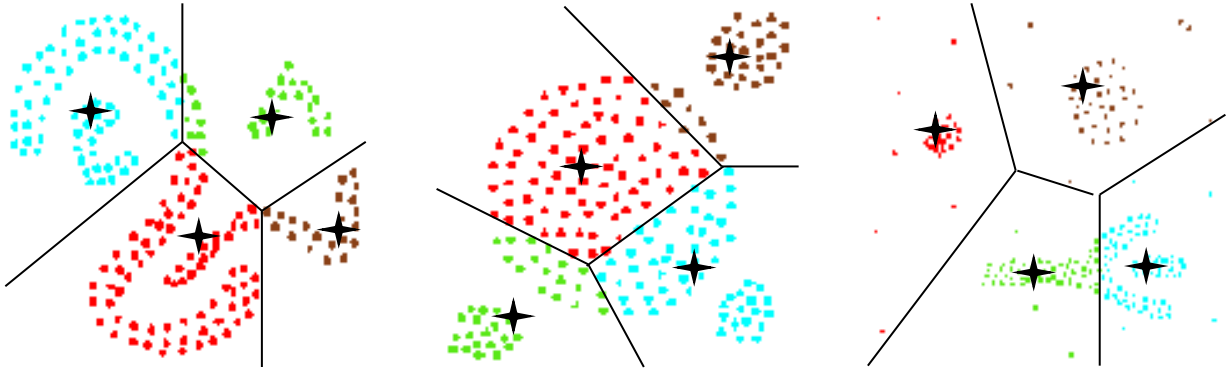
Today's learning goals

- Density-based clustering
  - Unsupervised grouping of data for overview
  - Overcome challenges in representation-based approaches
    - Handle arbitrary shapes and noise
    - At the cost of runtime
- Hierarchical clustering
  - There may be more than one way of grouping the data: strict or more loosely coupled
    - Hierarchy provides an overview
    - Helps in setting parameters
- Approximative clustering
  - When you need speed at the expense of accuracy
- Clustering evaluation measures
  - How do we know that it's the right grouping?
  - External vs internal measures
- Outlier detection
  - Finding deviating items and errors in the data

Density-Based Clustering

- Basic Idea:
  - Clusters are dense regions in the data space, separated by regions of lower object density
- Why Density-Based Clustering?





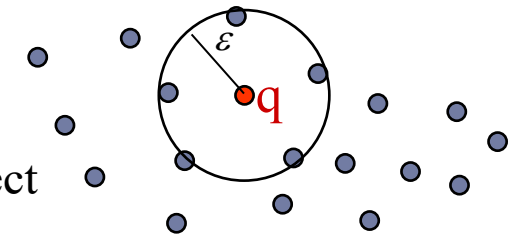
Results of a *k*-medoid algorithm for *k*=4

Different density-based approaches exist
Here we discuss the ideas underlying the DBSCAN algorithm

Density Based Clustering: Basic Concept

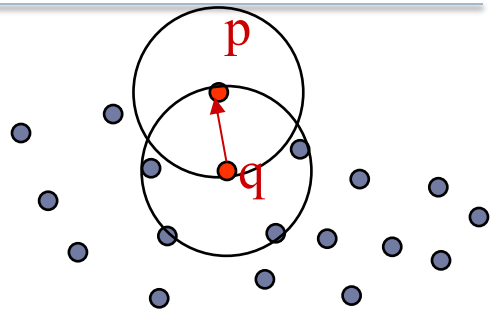
- Intuition for the formalization of the basic idea
  - For any point in a cluster, the local point density around that point has to exceed some threshold
  - The set of points from one cluster is spatially connected
- Local point density at a point *p* defined by two parameters
  - ε – radius for the neighborhood of point q:
    $N_\varepsilon(q) := \{p \text{ in data set } D \mid dist(p, q) \leq \varepsilon\}$
  - *MinPts* – minimum number of points in the given neighbourhood $N(p)$
- *q* is called a *core object* (or core point) w.r.t. ε, *MinPts* if
  $| N_\varepsilon(q) | \geq MinPts$

$MinPts = 5 \rightarrow$ q is a core object

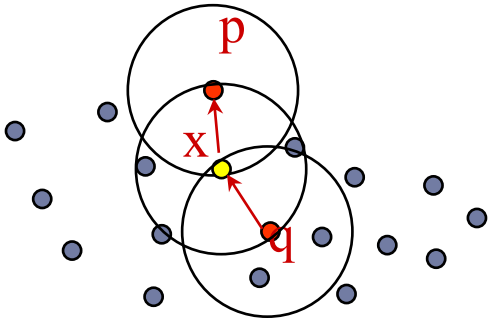


Density Based Clustering: Basic Definitions

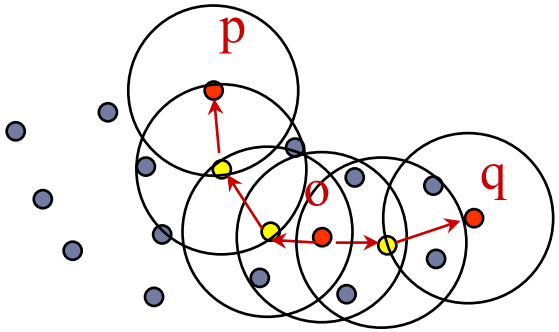
- *p directly density-reachable* from *q* w.r.t. ε, *MinPts* if
  1) $p \in N_{\varepsilon}(q)$ and
  2) *q* is a core object w.r.t. ε, *MinPts*



- *density-reachable*: transitive closure of *directly* density-reachable



- *p* is *density-connected* to a point *q* w.r.t. ε, *MinPts* if there is a point *o* such that both, *p* and *q* are density-reachable from *o* w.r.t. ε, *MinPts*.



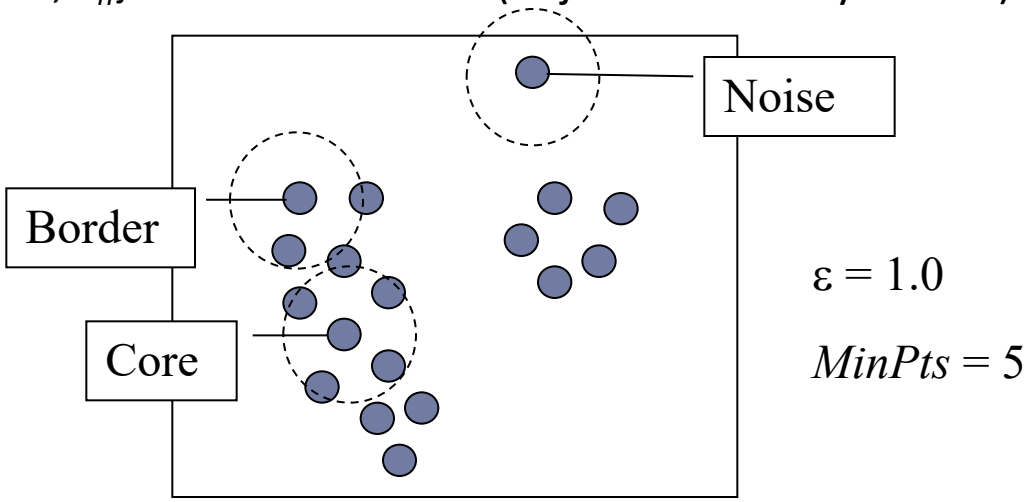
Density Based Clustering: Basic Definitions

- *Density-Based Cluster:* non-empty subset *S* of database *D* satisfying:

  1) *Maximality*: if *p* is in *S* and *q* is density-reachable from *p* then *q* is in *S*

  2) *Connectivity*: each object in *S* is density-connected to all other objects

- *Density-Based Clustering* of a database $D : \{S_1, ..., S_n; N\}$ where
  - $S_1, ..., S_n$: all density-based clusters in the database *D*
  - $N = D \setminus \{S_1, ..., S_n\}$ is called the *noise* (objects not in any cluster)



Density Based Clustering: DBSCAN Algorithm

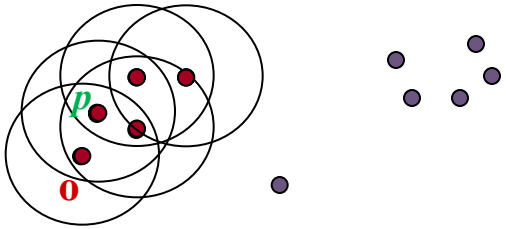
- Density Based Spatial Clustering of Applications with Noise
- Basic Theorem:
  - Each object in a density-based cluster C is density-reachable from any of its core-objects
  - Nothing else is density-reachable from core objects.

```
for each o ∈ D do
    if o is not yet classified then
        if o is a core-object then
            collect all objects density-reachable from o
            and assign them to a new cluster.
        else
            assign o to NOISE
```

  - density-reachable objects are collected by performing successive ε-neighborhood queries.

DBSCAN Algorithm: Example

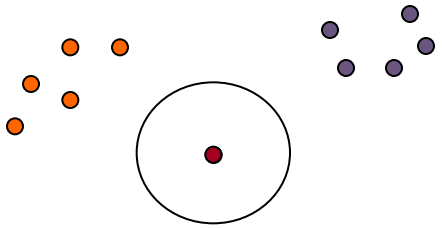
- Parameter
  - ε = 2.0
  - *MinPts* = 3



for each $o \in D$ **do**
  if o is not yet classified **then**
    if o is a core-object **then**
      collect all objects density-reachable from o
      and assign them to a new cluster.
    else
      assign o to NOISE

DBSCAN Algorithm: Example

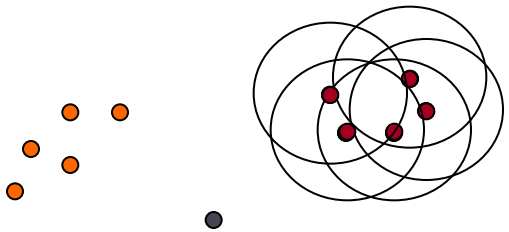
- Parameter
  - ε = 2.0
  - *MinPts* = 3



for each $o \in D$ **do**
  if o is not yet classified **then**
    if o is a core-object **then**
      collect all objects density-reachable from o
      and assign them to a new cluster.
    else
      assign o to NOISE

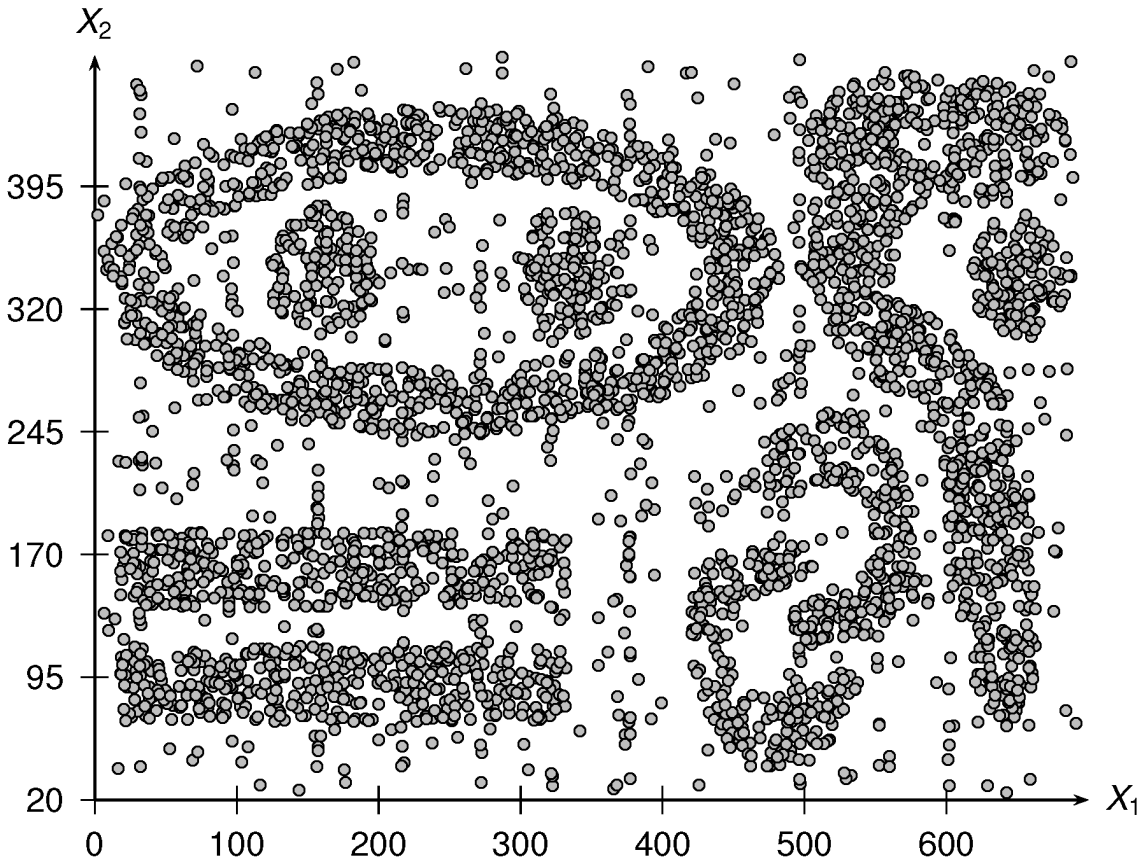
DBSCAN Algorithm: Example

- Parameter
  - ε = 2.0
  - *MinPts* = 3

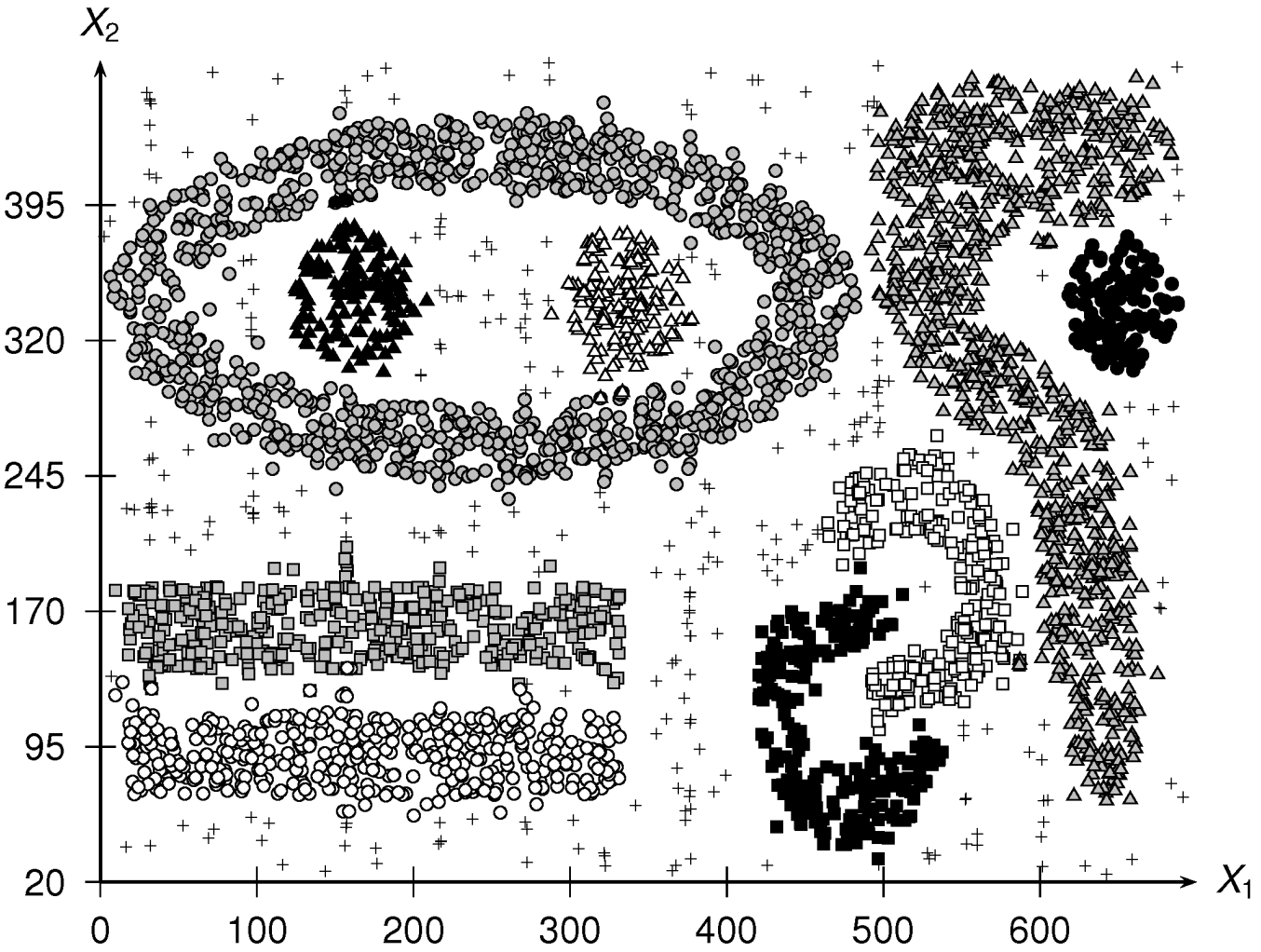


```
for each o ∈ D do
    if o is not yet classified then
        if o is a core-object then
            collect all objects density-reachable from o
            and assign them to a new cluster.
        else
            assign o to NOISE
```

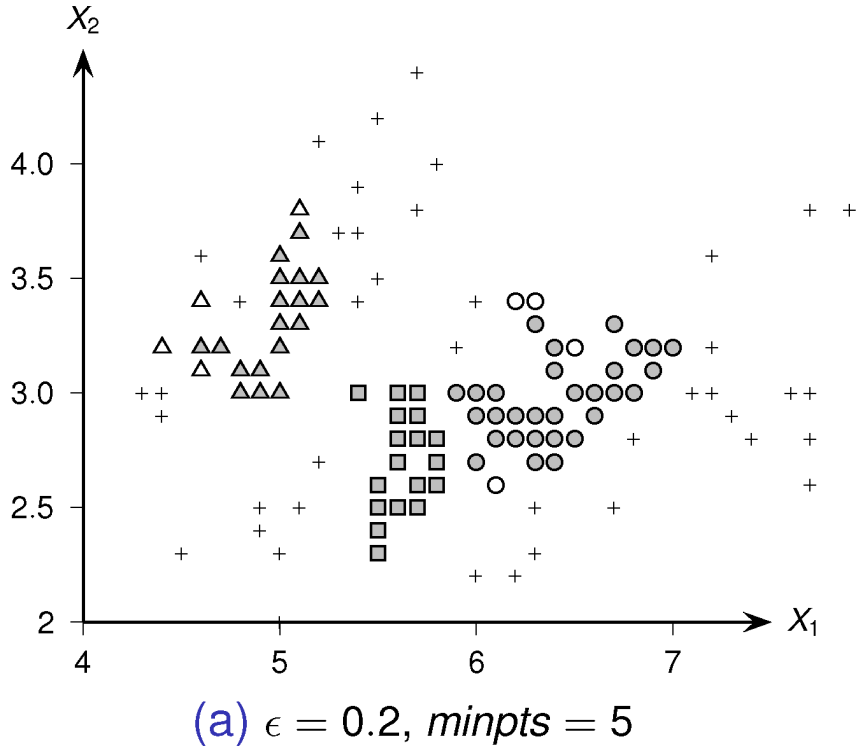
Synthetic data in 2D



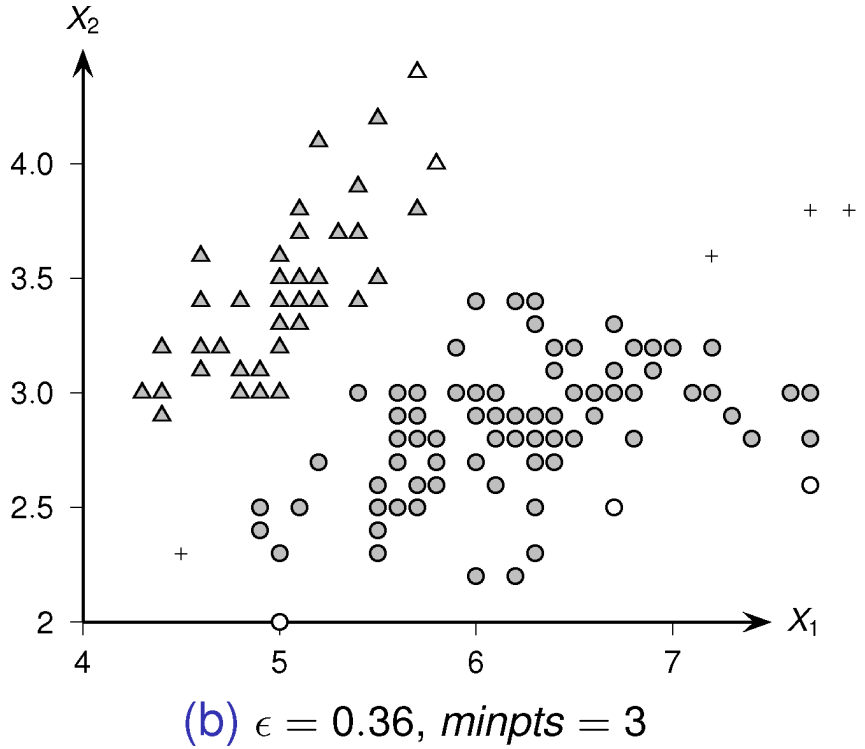
DBSCAN, e=15, minpts=10



DBSCAN, Iris data (sepal length, sepal width)



(a) $\epsilon = 0.2$, $minpts = 5$



(b) $\epsilon = 0.36$, $minpts = 3$

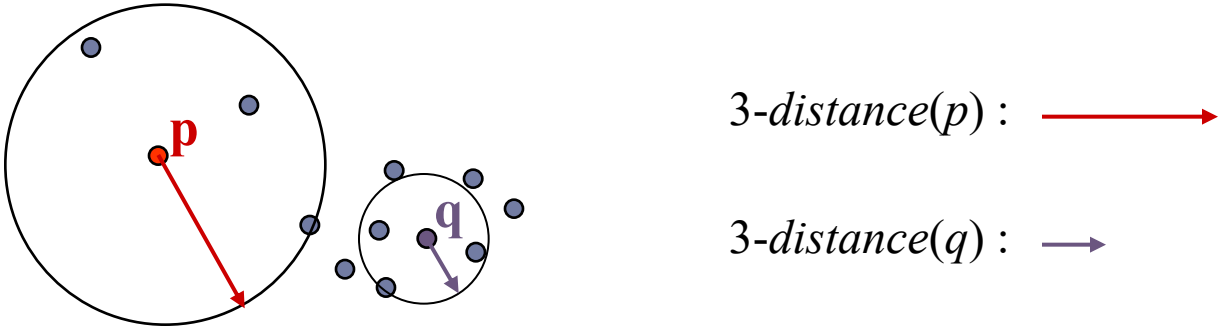
- Which one is best?

(a)

(b)

Determining the Parameters ε and *MinPts*

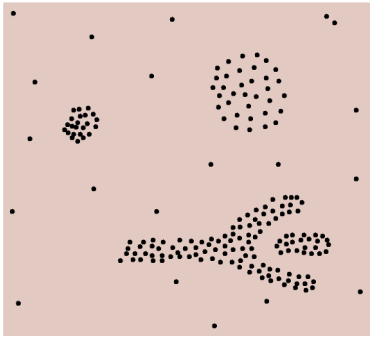
- Cluster: Point density higher than specified by ε and *MinPts*
- Idea: use the point density of the least dense cluster in the data set as parameters – but how to determine this?
- Heuristic: look at the distances to the *k*-nearest neighbors

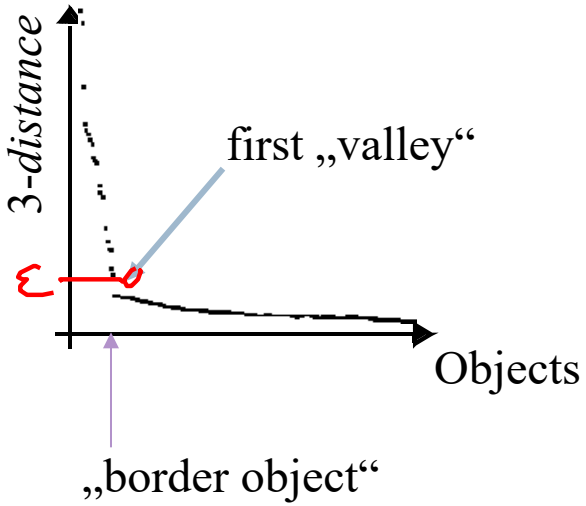


- Function *k-distance*(*p*): distance from *p* to the its *k*-nearest neighbor
- *k-distance plot*: *k*-distances of all objects, sorted in decreasing order

Determining the Parameters ε and *MinPts*

- Example *k*-distance plot
  1. dim = 2 → MinPts = 3
  2. Identify border object
  3. Set ε

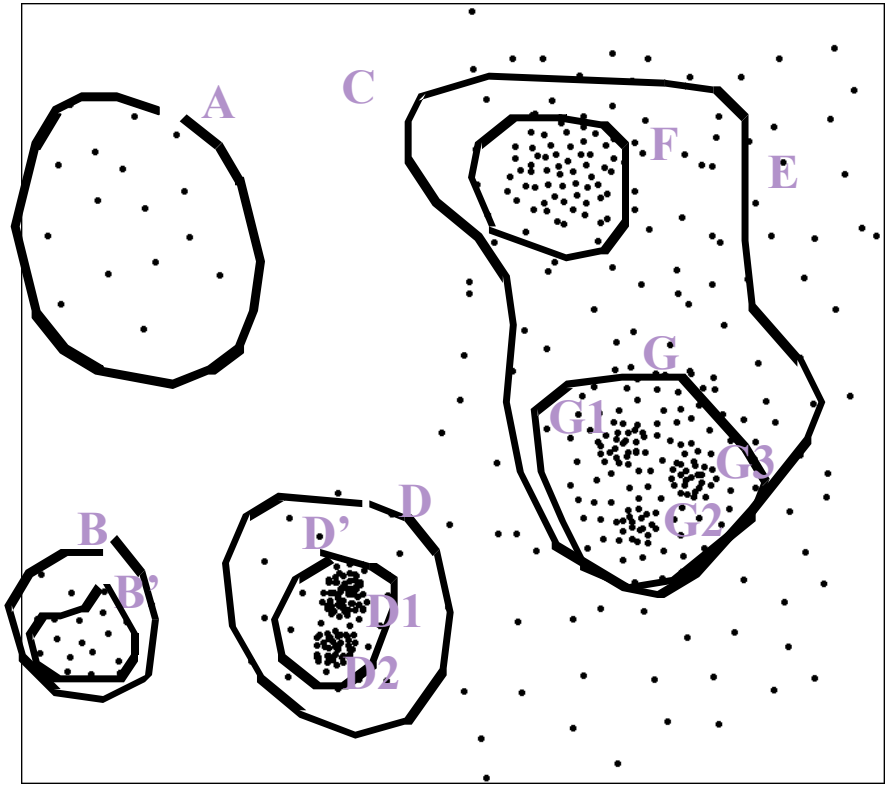


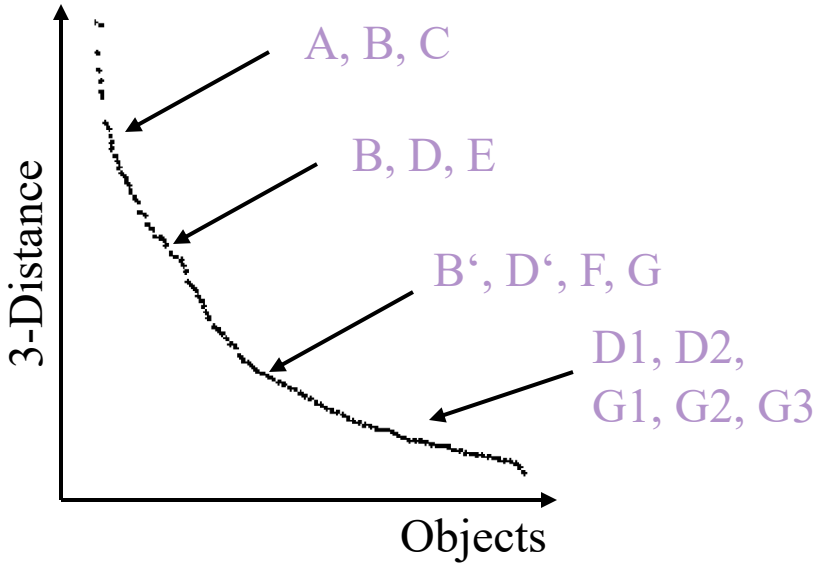


- Heuristic method:
  - Fix a value for *MinPts*
  - (default: $2 \times d - 1$, d = dimension of data space)
  - User selects “border object” *o* from the *MinPts-distance* plot; ε is set to *MinPts-distance*(o)

Determining the Parameters ε and *MinPts*

- Problematic example



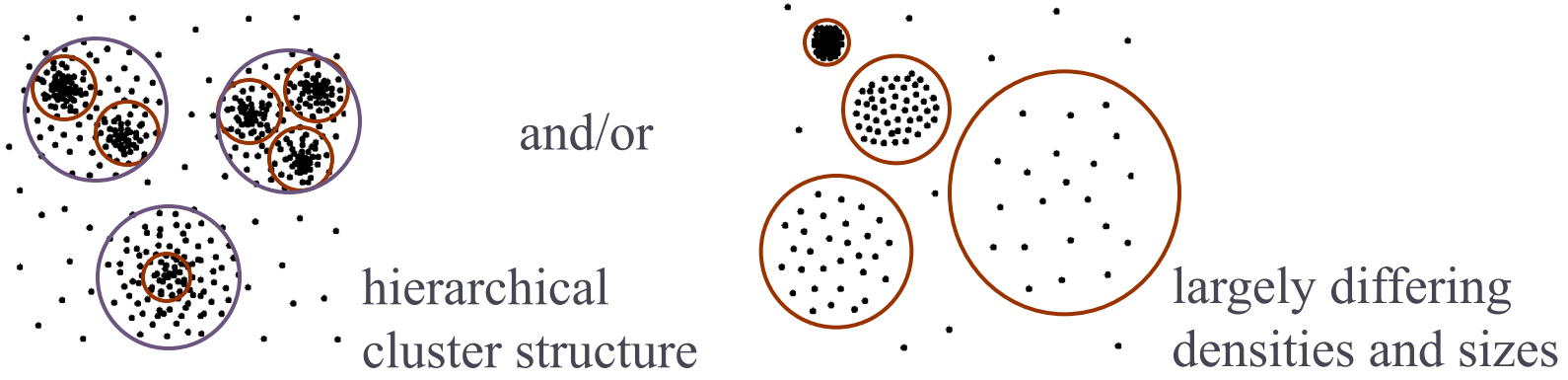


Density Based Clustering: Discussion

- Advantages
  - Clusters can have arbitrary shape and size, i.e. clusters are not restricted to have convex shapes
  - Number of clusters is determined automatically
  - Can separate clusters from surrounding noise
  - Can be supported by spatial index structures
- Disadvantages
  - Input parameters may be difficult to determine
  - In some situations very sensitive to input parameter setting

Hierarchical Clustering

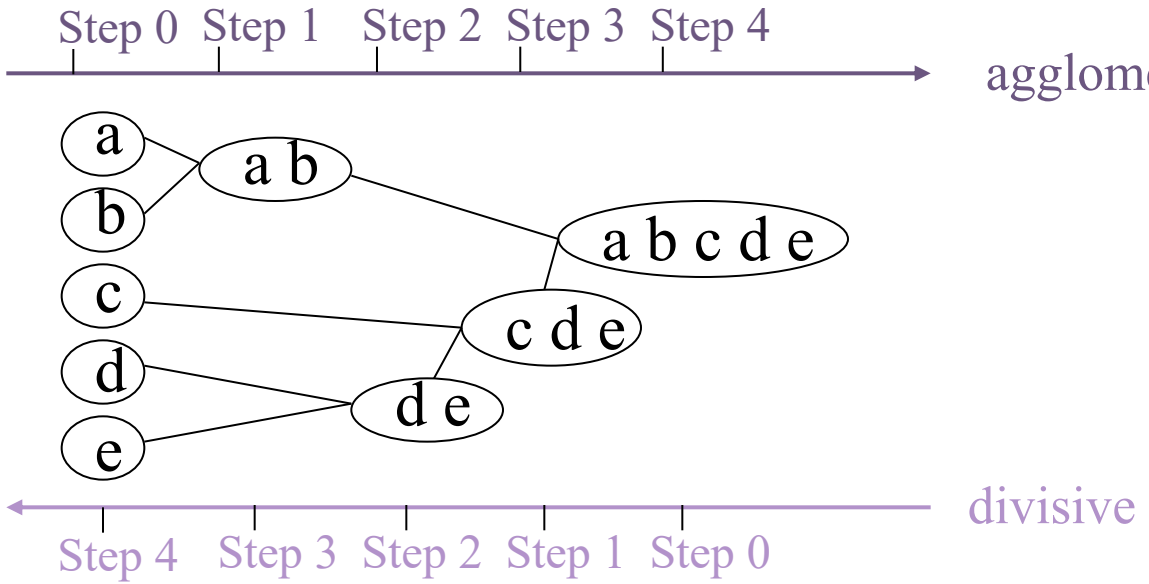
- Global parameters to separate all clusters with a partitioning clustering method may not exist



- Need a hierarchical clustering algorithm in these situations

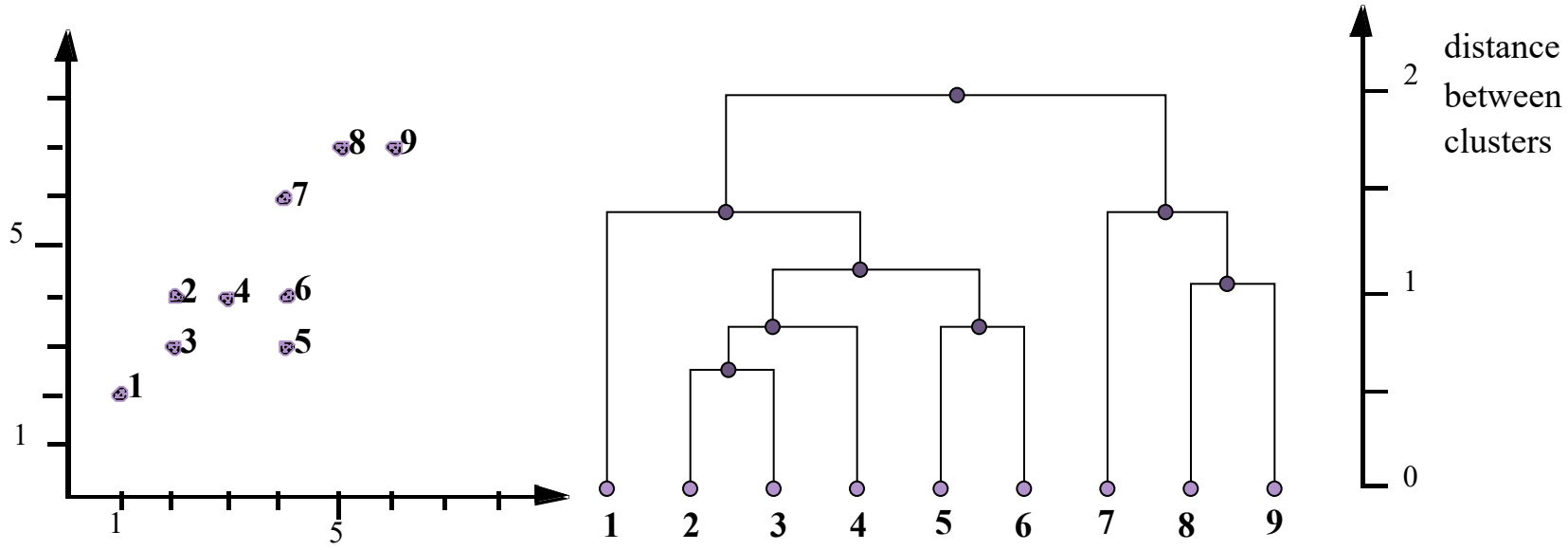
Hierarchical Clustering: Basic Notions

- Hierarchical decomposition of the data set (with respect to a given similarity measure) into a set of nested clusters
- Result represented by a so called *dendrogram* (greek δενδρο = tree)
  - Nodes in the dendrogram represent possible clusters
  - can be constructed bottom-up (agglomerative approach) or top down (divisive approach)



Hierarchical Clustering: Example

- Interpretation of the dendrogram
  - The root represents the whole data set
  - A leaf represents a single objects in the data set
  - An internal node represent the union of all objects in its sub-tree
  - The height of an internal node represents the distance between its two child nodes



Agglomerative Hierarchical Clustering

1. Initially, each object forms its own cluster
2. Compute all pairwise distances between the initial clusters (objects)
3. Merge the closest pair (A, B) in the set of the current clusters into a new cluster $C = A \cup B$
4. Remove A and B from the set of current clusters; insert C into the set of current clusters
5. If the set of current clusters contains only C (i.e., if C represents all objects from the database): STOP
6. Else: determine the distance between the new cluster C and all other clusters in the set of current clusters; go to step 3.

- Requires a distance function for clusters (sets of objects)

Single Link Method and Variants

- Given: a distance function *dist*(*p*, *q*) for database objects
- The following distance functions for clusters (i.e., sets of objects) *X* and *Y* are commonly used for hierarchical clustering:

Single-Link: $$dist\_sl(X,Y) = \min_{x \in X, y \in Y} dist(x,y)$$

Complete-Link: $$dist\_cl(X,Y) = \max_{x \in X, y \in Y} dist(x,y)$$

Average-Link: $$dist\_al(X,Y) = \frac{1}{|X| \cdot |Y|} \cdot \sum_{x \in X, y \in Y} dist(x,y)$$

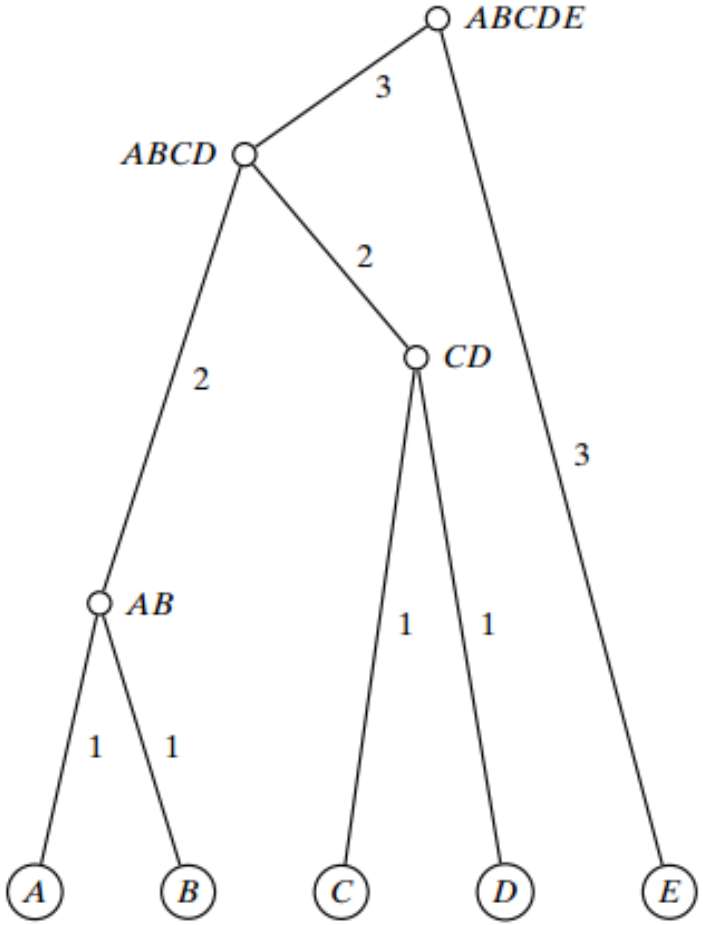
Single Link Method Example

δ	E
$ABCD$	(3)

δ	CD	E
AB	(2)	3
CD		3

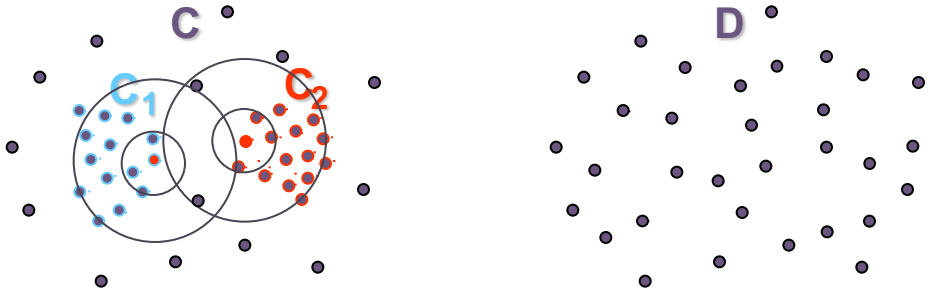
δ	C	D	E
AB	3	2	3
C		(1)	3
D			5

δ	B	C	D	E
A	(1)	3	2	4
B		3	2	3
C			1	3
D				5

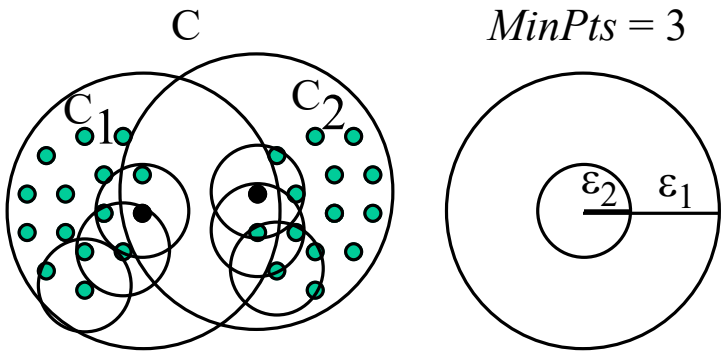


Density-Based Hierarchical Clustering

- *Observation*: Dense clusters are completely contained by less dense clusters

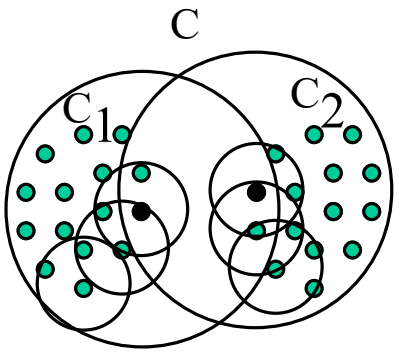


- *Idea*: Process objects in the "right" order and keep track of point density in their neighborhood



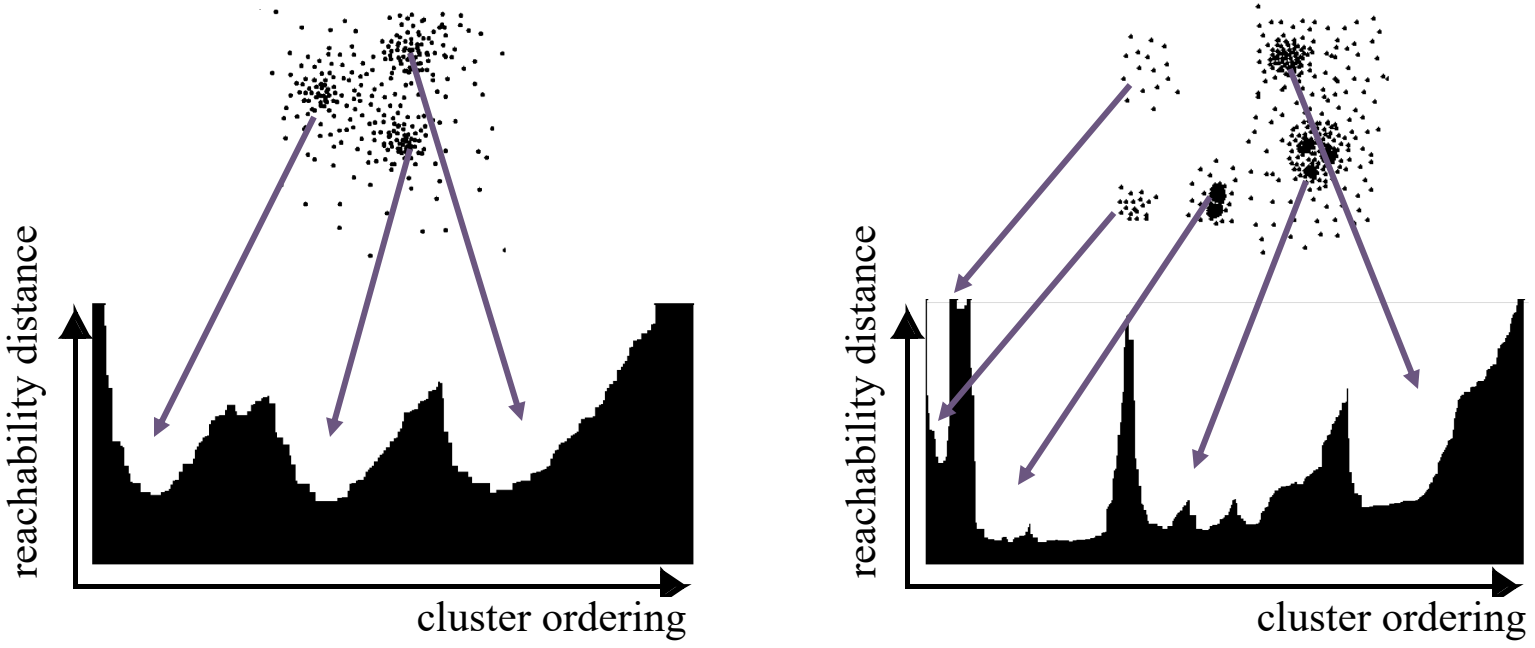
The Algorithm OPTICS

- OPTICS: **O**rdering **P**oints **T**o **I**dentify the **C**lustering **S**tructure
- Basic data structure: controlList
  - 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
  - reachability-distance of points



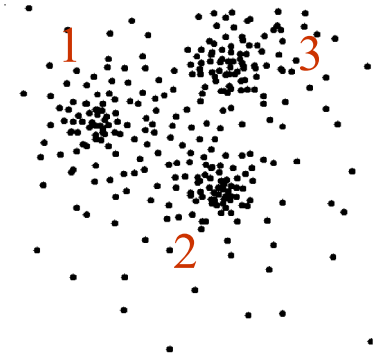
OPTICS: The Reachability Plot

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

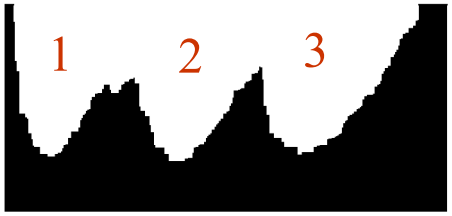


OPTICS: Parameter Sensitivity

- Relatively insensitive to parameter settings
- Good result if parameters are just "large enough"

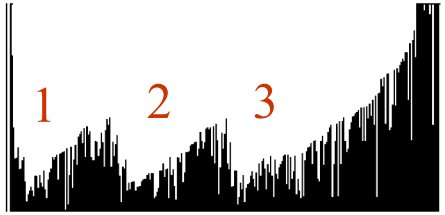


MinPts = 10, ε = 10



MinPts = 10, **ε = 5**

***MinPts* = 2**, ε = 10

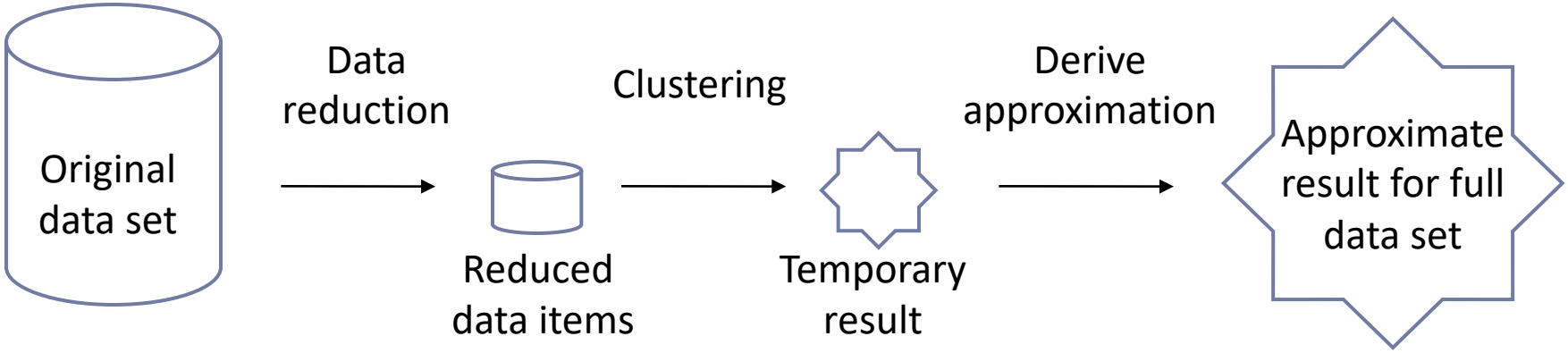


Hierarchical Clustering: Discussion

- 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 $O(n^2)$

Scaling-Up Clustering Algorithms via Data Reduction/Summarization

Basic Idea:



Small Loss in Accuracy

Large Performance Boost

Most simple approach: Random Sampling

BIRCH

- Birch: Balanced Iterative Reducing and Clustering using Hierarchies
- Incrementally construct a CF (Clustering Feature) tree, a hierarchical data structure for multiphase clustering
  - Phase 1: scan DB to build an initial in-memory CF tree (a multi-level compression of the data that tries to preserve the inherent clustering structure of the data)
  - Phase 2: use an arbitrary clustering algorithm to cluster the leaf nodes of the CF-tree
- Scales linearly: finds a good clustering with a single scan and improves the quality with a few additional scans
- Weakness: handles only numeric data, and sensitive to the order of data records

BIRCH – Discussion

Advantages

- Compression factor can be tuned to the available main memory
- Efficient construction of a micro-clustering (O(n))
- Good clustering result for partitioning iterative-refinement clustering algorithms such as k-means and k-medoid when applied to only the leaf nodes of a CF-tree

Disadvantages

- Only for data from a Euclidean vector space (linear sum, square sum, mean, etc must be defined)
- Sensitive to the order of the data records
- Entries limited by the page size
- Different parameters to tune

Clustering evaluation and validation

- External measures:
  - Take criteria into account that are not part of the clustering data
    - E.g. class labels
- Internal measures:
  - Based only on clustering data
    - E.g. distances as in TD or silhouette coefficient
- Measures can be relative
  - Compare two clusterings instead of obtaining ”objective” goodness value
  - Some measures, e.g. silhouette can be used in both ways

External measures

- External measures
  - assume correct / ground-truth clustering is known and basis for evaluation
  - ground-truth clustering T = { T_1, T_2,… T_k},
  - Cluster T_j consists of all the points with label j
    - E.g. three Iris types give rise to three ground truth clusters
  - C = {C_1,..., C_r} a clustering of the same dataset to be evaluated
- Use r × k contingency table $\mathbf{N}(i,j) = n_{ij} = |C_i \cap T_j|$
  - n_{ij} number of points in both cluster C_i and in ground-truth partition T_j

Purity

- Purity quantifies the extent to which a cluster C_i contains entities from only one partition

$$purity_i = \frac{1}{n_i} \max_{j=1}^{k} \{n_{ij}\}$$

- The purity of clustering C is defined as the weighted sum of the clusterwise purity values

$$purity = \sum_{i=1}^{r} \frac{n_i}{n} purity_i = \frac{1}{n} \sum_{i=1}^{r} \max_{j=1}^{k} \{n_{ij}\}$$

- where the ratio n_i/n denotes the fraction of points in cluster C_i

Purity of k-means clustering

- What would you expect to give best result?

  a) Purity of Iris clustering with k=3

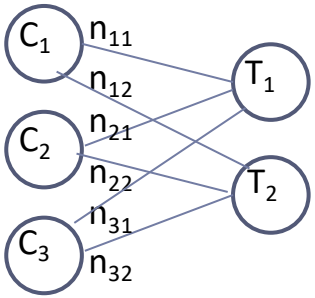
  b) Purity of Iris clustering with k=50

- What in the ground trust is used / not used?

Maximum matching

- Evaluated based on best mapping between clusters and ground truth
  - Such that sum of common points between machted clusters is maximized
  - only one cluster can match with a given partition (unlike purity)
- Matching M in G between C and T
  - Modelled as bipartite graph over vertices V = C ∪ T, edges are E = {(C_i, T_j)} with edge weights w(C_i, T_j) = n_{ij}
    - Means: connect the two clusterings, connection weight is the contingency
  - Matching is subset of E where edges in M are pairwise nonadjacent
    - no common vertex
- Maximum weight matching

$$match = \arg\max_{M} \left\{ \frac{w(M)}{n} \right\}$$



where w(M) is sum of all edge weights in matching M

$$w(M) = \sum_{e \in M} w(e)$$

F-measure

- Given cluster C_i, let j_i denote the partition that contains the maximum number of points from C_i
  - Means the ground truth cluster it mostly contains
- precision of cluster C_i is its purity $prec_i = \frac{1}{n_i} \max_{j=1}^{k} \{n_{ij}\} = \frac{n_{ij_i}}{n_i}$
- recall of cluster C_i is $recall_i = \frac{n_{ij_i}}{|T_{j_i}|} = \frac{n_{ij_i}}{m_{j_i}}$ where mij = $|T_{j_i}|$
- F-measure is harmonic mean of precision and recall
  - So we look at how pure it is and at the same time how much of the ground truth partition it contains

$$F_i = \frac{2}{\frac{1}{prec_i} + \frac{1}{recall_i}} = \frac{2 \cdot prec_i \cdot recall_i}{prec_i + recall_i} = \frac{2\, n_{ij_i}}{n_i + m_{j_i}}$$

- F-measure for clustering is mean of clusterwise F-measure

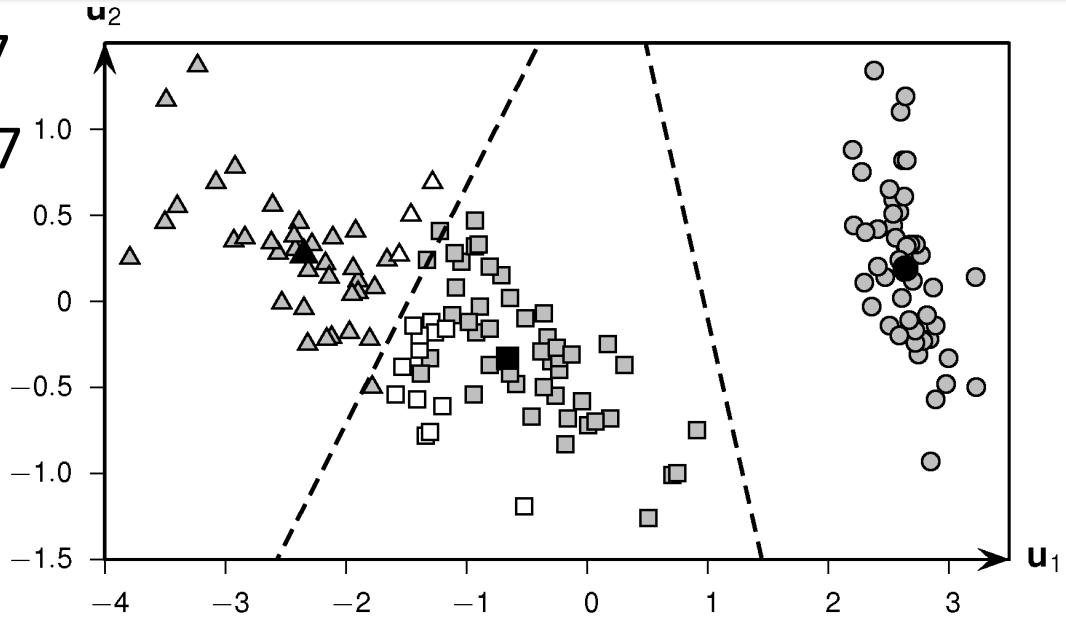
$$F = \frac{1}{r} \sum_{i=1}^{r} F_i$$

Iris principle components data, good clustering

Purity 0.887

Match 0.887

F1 0.85



Contingency table:

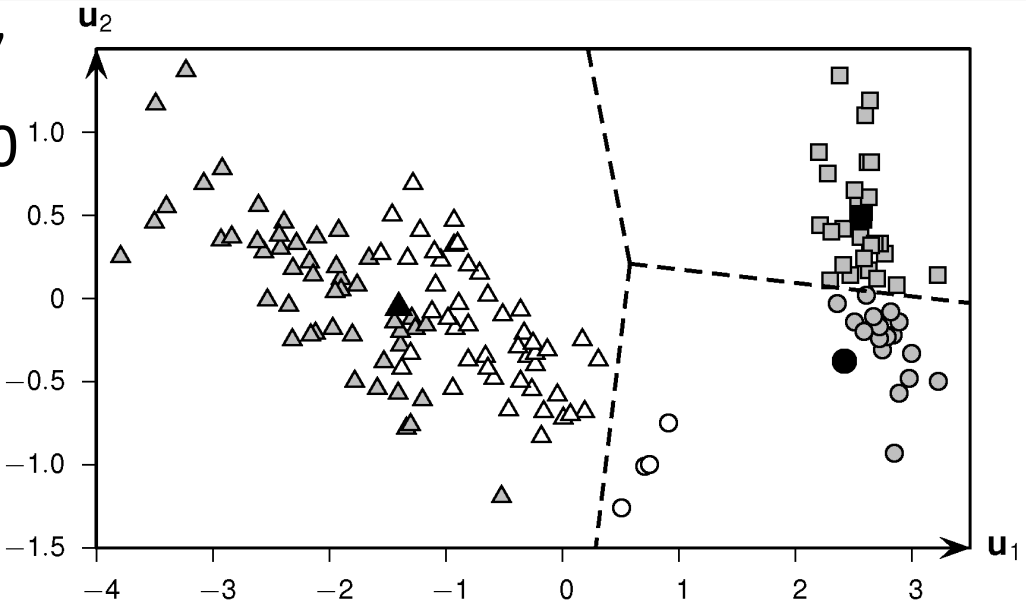
	iris-setosa T_1	iris-versicolor T_2	iris-virginica T_3	n_i
C_1(squares)	0	47	14	61
C_2(circles)	50	0	0	50
C_3(triangles)	0	3	36	39
m_j	50	50	50	$n = 150$

Iris principle components data, poor clustering

Purity 0.667

Match 0.560

F 0.658



Contingency table:

	iris-setosa T_1	iris-versicolor T_2	iris-virginica T_3	n_i
C_1 (*squares*)	30	0	0	30
C_2 (*circles*)	20	4	0	24
C_3 (*triangles*)	0	46	50	96
m_j	50	50	50	$n = 150$

Evaluation measures - Discussion

- Internal measures:
  - Recall e.g. silhouette coefficient
  - Make no assumptions about ground truth
  - In unsupervised learning this is very often the case in practice
- External measures:
  - Make use of external information such as domain expertise
  - Sometimes to be generated for testing sample
- Different measures capture different aspects of a good clustering
  - This is actively discussed in the research community
  - For example, can a cluster be “matched” to two ground truth clusters (or the other way around?) Should this be punished somehow?

Outlier Detection

- Outlier: Data that deviates from remainder of the data
- Example:
  - Usually, you charge about $ 2000 a month to your credit card in the US
  - If there is a transaction of $ 10000 one day from Europe: are you on holiday or your card?
  - And if these $ 10000 are charged in the US - is that still suspicious?
  - The more attributes, the more difficult the decision

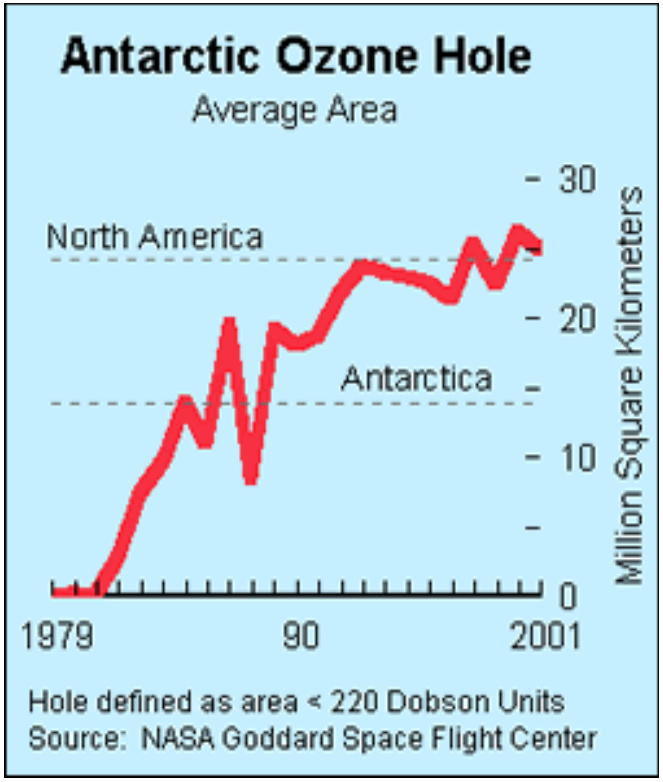
Outlier Detection

- Hawkins Definition of an Outlier
  - An object that deviates so much from the rest of the data set as to arouse suspicion that it was generated by a different mechanism
- Problem
  - Find top *n* outlier points
- Applications:
  - Credit card fraud detection
  - Telecom fraud detection
  - Customer segmentation
  - Medical analysis

Importance of Detecting Anomalies

Ozone Depletion History

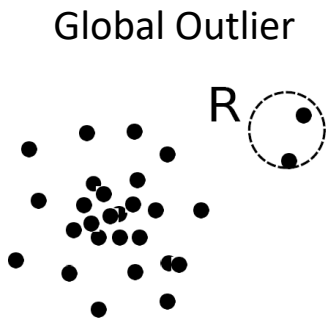
- In 1985 three researchers (Farman, Gardinar and Shanklin) were puzzled by data gathered by the British Antarctic Survey showing that ozone levels for Antarctica had dropped 10% below normal level
- Why did the Nimbus 7 satellite, which had instruments aboard for recording ozone levels, not record similarly low ozone concentrations?
- The ozone concentrations recorded by the satellite were so low they were being treated as noise by a computer program and discarded!



Sources:
http://exploringdata.cqu.edu.au/ozone.html
http://www.epa.gov/ozone/science/hole/size.html

Types of Outliers (I)

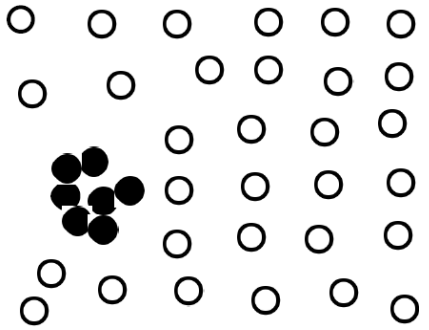
- Global outlier (or point anomaly)
  - Object significantly deviates from the rest of the data set
  - Ex. Intrusion detection in computer networks
  - Issue: Find an appropriate measurement of deviation



- Contextual outlier (or *conditional outlier*)
  - Object deviates significantly based on a selected context
  - Ex. 20^{o} C in Aarhus: outlier? (depending on summer or winter?)
  - Attributes of data objects should be divided into two groups
    - Contextual attributes: defines the context, e.g., time & location
    - Behavioral attributes: characteristics of the object, used in outlier evaluation, e.g., temperature
  - Can be viewed as a generalization of *local outliers*—whose density significantly deviates from its local area
  - Issue: How to define or formulate meaningful context?

Types of Outliers (II)

- Collective Outliers
  - Set of objects *collectively* deviate significantly from the whole data set, even if the individual data objects may not be outliers
  - Applications: E.g., *intrusion detection*:
    - When a number of computers keep sending denial-of-service packages
  - Detection of collective outliers
    - Consider not only individual objects, but also groups of objects
    - Need background knowledge on relationship among data objects, such as distance or similarity measure
- A data set may have multiple types of outlier
- One object may belong to more than one type of outlier



Collective Outlier

Challenges of Outlier Detection

- Modeling normal objects and outliers properly
  - Hard to enumerate all possible normal behaviors in an application
  - The border between normal and outlier objects is often a gray area
- Application-specific outlier detection
  - Relationships among objects are often application-dependent
  - E.g., clinic data: a small deviation could be an outlier; while in marketing analysis, larger fluctuations
- Noise in outlier detection
  - Noise may distort normal objects
    - blur distinction between normal objects and outliers
- Understandability
  - Understand why these are outliers: Justification of the detection
  - Specify the degree of an outlier: the unlikelihood of the object being generated by a normal mechanism

Outlier Detection I: Supervised Methods

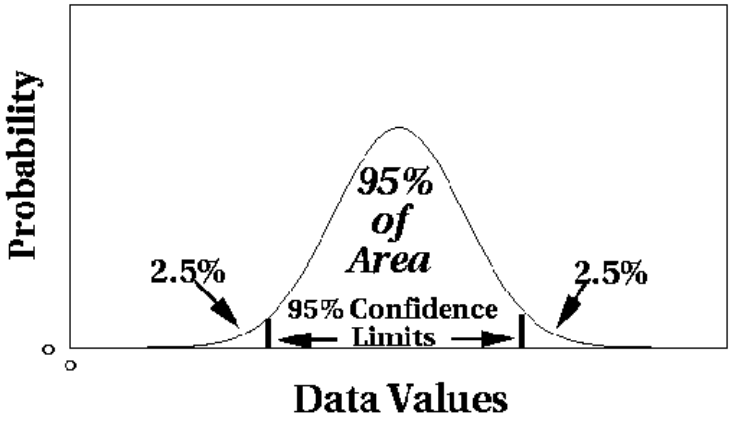
- If user-labeled examples of outliers can be obtained
  - Modeling outlier detection as a classification problem
    - Samples examined by domain experts used for training & testing
  - Methods for Learning a classifier for outlier detection effectively:
    - Model normal objects & report those not matching the model as outliers, or
    - Model outliers and treat those not matching the model as normal
  - Challenges
    - Imbalanced classes, i.e., outliers are rare: Boost the outlier class and make up some artificial outliers
    - Catch as many outliers as possible, i.e., recall is more important than accuracy (i.e., not mislabeling normal objects as outliers)

Outlier Detection II: Unsupervised Methods

- An outlier is expected to be far away from any groups of normal objects
- Weakness: Cannot detect collective outlier effectively
  - Normal objects may not share any strong patterns, but the collective outliers may share high similarity in a small area
- Ex. In some intrusion or virus detection, normal activities are diverse
  - Unsupervised methods may have a high false positive rate but still miss many real outliers
  - Supervised methods can be more effective, e.g., identify attacking some key resources, but can only recover outliers present in training data
- Many clustering methods can be adapted for unsupervised methods
  - Find clusters, then outliers: not belonging to any cluster
  - Problem 1: Hard to distinguish noise from outliers
  - Problem 2: Costly since first clustering: but far less outliers than normal objects
    - Newer methods: tackle outliers directly

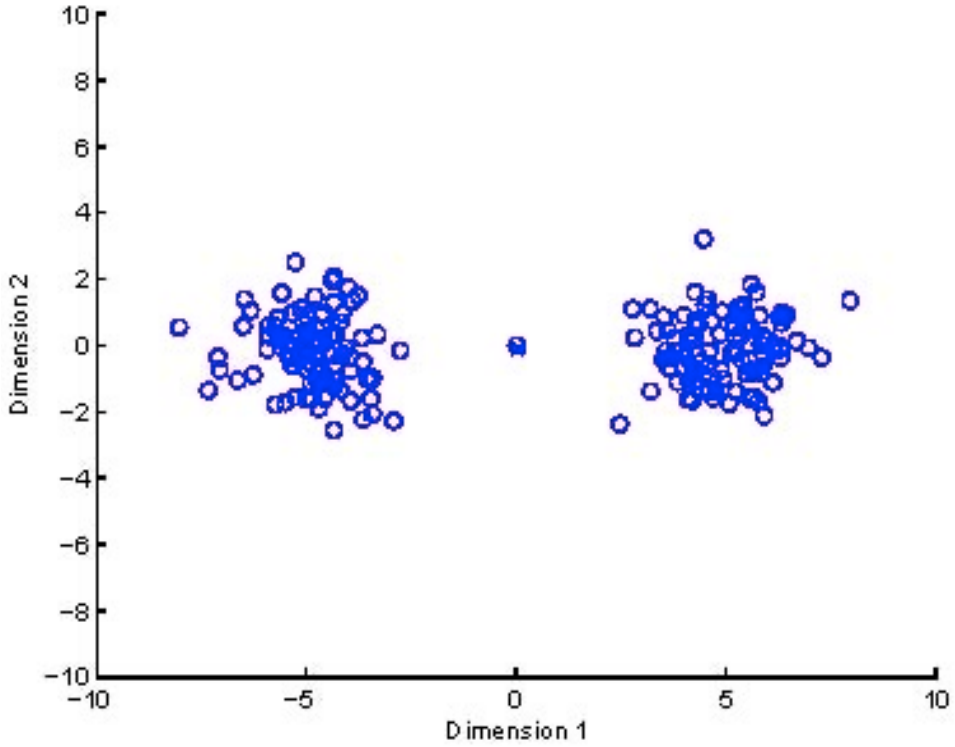
Outlier Discovery: Statistical Approaches

- Assume an underlying model that generates data set (e.g. normal distribution)
- Discordance tests depending on
  - data distribution
  - distribution parameter (e.g., mean, variance)
  - number of expected outliers
- Typical approach: assuming normal distribution, outliers are those objects that are more than three times the standard deviation from the mean in a given attribute



Limitations of statistical approaches

- Most tests are for single attributes only
- In many cases, data distribution may not be known
- Statistical measures might be sensitive to outliers themselves
- Not all outliers are detected: e.g.



O. Zaïane, lecture “Principles of Knowledge Discovery in Data”, U of Alberta, CA

What else could go wrong

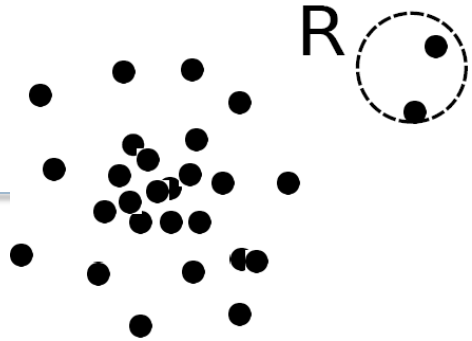
- Now, let's assume we do not have such outliers in the middle, only in extremes
  - We fit a Normal / Gaussian distribution to the values of the attribute and look at extremes
  - What could be an issue for very noisy data?

Further Problems with Statistical Tests

- Robustness
  - Mean and standard deviation are very sensitive to outliers
  - These values are computed for the complete data set (including potential outliers)
  - The *MDist* is used to determine outliers although the *MDist* values are influenced by these outliers

- Discussion
  - Data distribution is fixed
  - Low flexibility (no mixture model)
  - Global method
  - Outputs a label but can also output a score

Model-based Methods



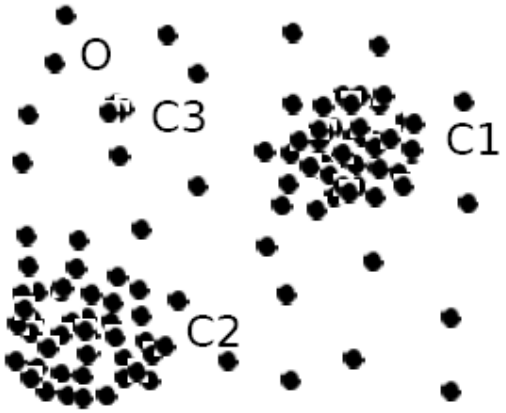
- Statistical methods (also known as model-based methods) assume that the normal data follow some statistical model (a stochastic model)
  - The data not following the model are outliers.
- Example: use Gaussian distribution to model normal data
  - For each object estimate the probability that it fits the Gaussian distribution
  - If probability is very low, likely an outlier
- Effectiveness of statistical methods: highly depends on whether the assumption of statistical model holds in the real data
- There are rich alternatives to use various statistical models
  - E.g., parametric (as in above model) vs. non-parametric (e.g. kernel per data object)

Parametric Methods: Mixture Model

- Assumption of single data distribution may be overly simplified
- Use several parametric distributions to fit the data
  - E.g. two normal distributions
  - probability that o is generated by mixture of two Gaussians is

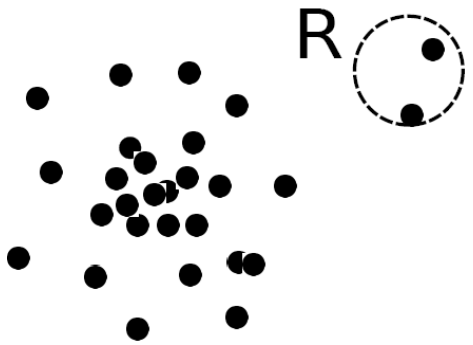
$$Pr(o|\Theta_1, \Theta_2) = f_{\Theta_1}(o) + f_{\Theta_2}(o)$$

  where $f_{\theta 1}$ and $f_{\theta 2}$ are the probability density functions
- Then use EM algorithm to learn the parameters μ_1, σ_1, μ_2, σ_2 from data
- An object o is an outlier if it does not belong to any cluster



Proximity-Based / Distance-Based Approach

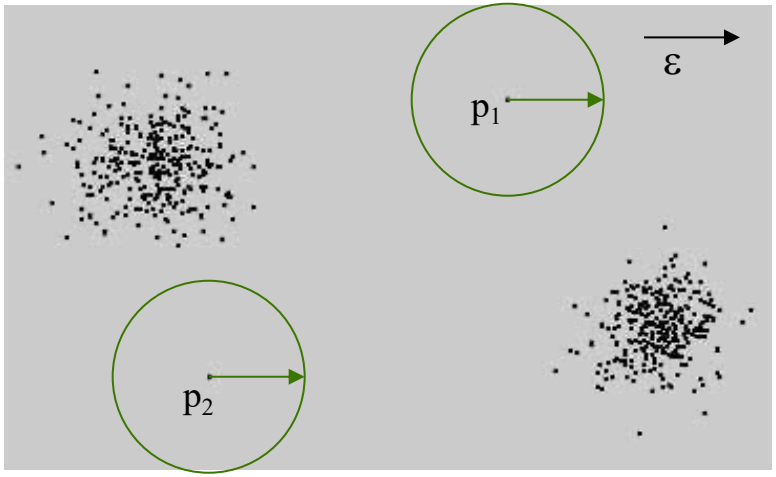
- Introduced to overcome the main limitations imposed by statistical methods
  - Multi-dimensional analysis without knowing data distribution
- An object is an outlier if the nearest neighbors of the object are far away, i.e., the proximity of the object is significantly deviates from the proximity of most of the other objects in the same data set
- Example: Model the proximity of an object using its 3 nearest neighbors
  - Objects in region R are substantially different from other objects in the data set.
  - Thus the objects in R are outliers



Distance-based Approaches

- DB(ε,π)-Outliers
  - Basic model
    - Given a radius ε and a percentage π
    - A point *p* is considered an outlier if at most π percent of all other points have a distance to *p* less than ε

$$OutlierSet(\varepsilon,\pi) = \{p \mid \frac{Card(\{q \in DB \mid dist(p,q) < \varepsilon\})}{Card(DB)} \leq \pi\}$$



Discussion Proximity-Based Methods

- The effectiveness of proximity-based methods

  highly relies on the proximity measure
- Cannot easily detect groups of outliers
- Proximity-based outlier detection usually encompasses both distance-based and density-based approaches

- Seen as a formalization of the outlier notion by Hawkins
- Shown to be a generalization of defining an outlier as an object that is more than a factor times the standard deviation from the mean
- Easy to implement, yet global view, requires parameterization

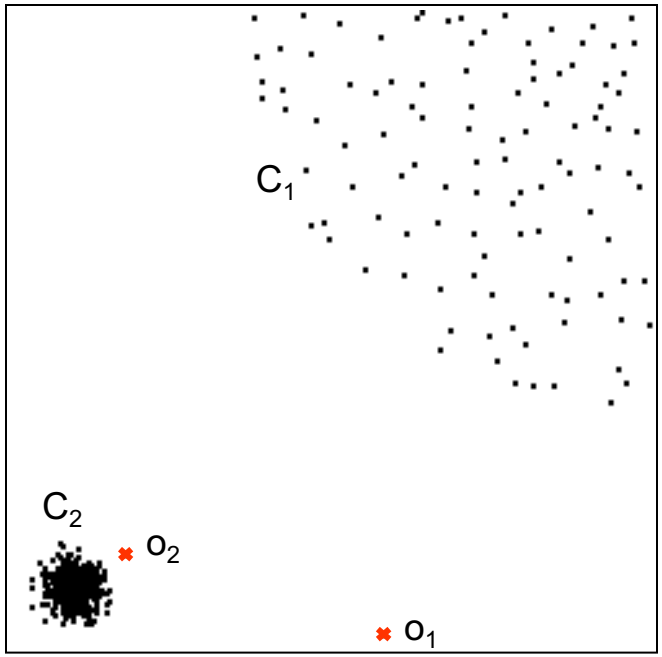
Density-based outlier detection approaches

- General idea
  - Compare the density around a point with the density around its local neighbors
  - The relative density of a point compared to its neighbors is computed as an outlier score
  - Approaches also differ in how to estimate density

- Basic assumption
  - The density around a normal data object is similar to the density around its neighbors
  - The density around an outlier is considerably different to the density around its neighbors

Outlier Discovery: Local Outlier Factors

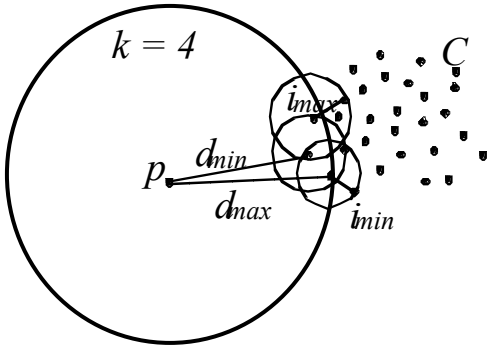
- o_1: clear *DB*(*pct*, *dmin*)-outlier for
  - *Pct:* greater than 99%
  - *dmin*: very large
- o_2: Not well captured
  - either: not an *DB*(*pct*, *dmin*)-outlier
  - or: many points in C_1 are outliers, too!



- only ”global” outliers are captured
- objects that are outlying relative to their local neighborhoods also “deviating”

Local outliers

- Local outlier factor of points *p:*
  - Basic Idea: Look at the *k*-nearest neighbor distance of *p* relative to the *k*-nearest neighbor distances of these *k* neighbors of *p*

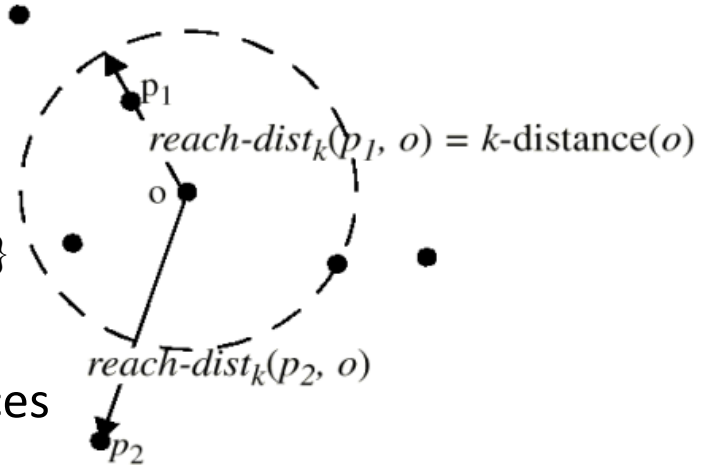


- Instead of ”global” outliers, search for ”local” outliers
- Each object has an outlier factor, which is the degree to which the object deviates

Density-based Approaches

- Model
  - Reachability distance
    - Introduces a smoothing factor

$$reach-dist_k(p,o) = \max\{k-\text{distance}(o), dist(p,o)\}$$



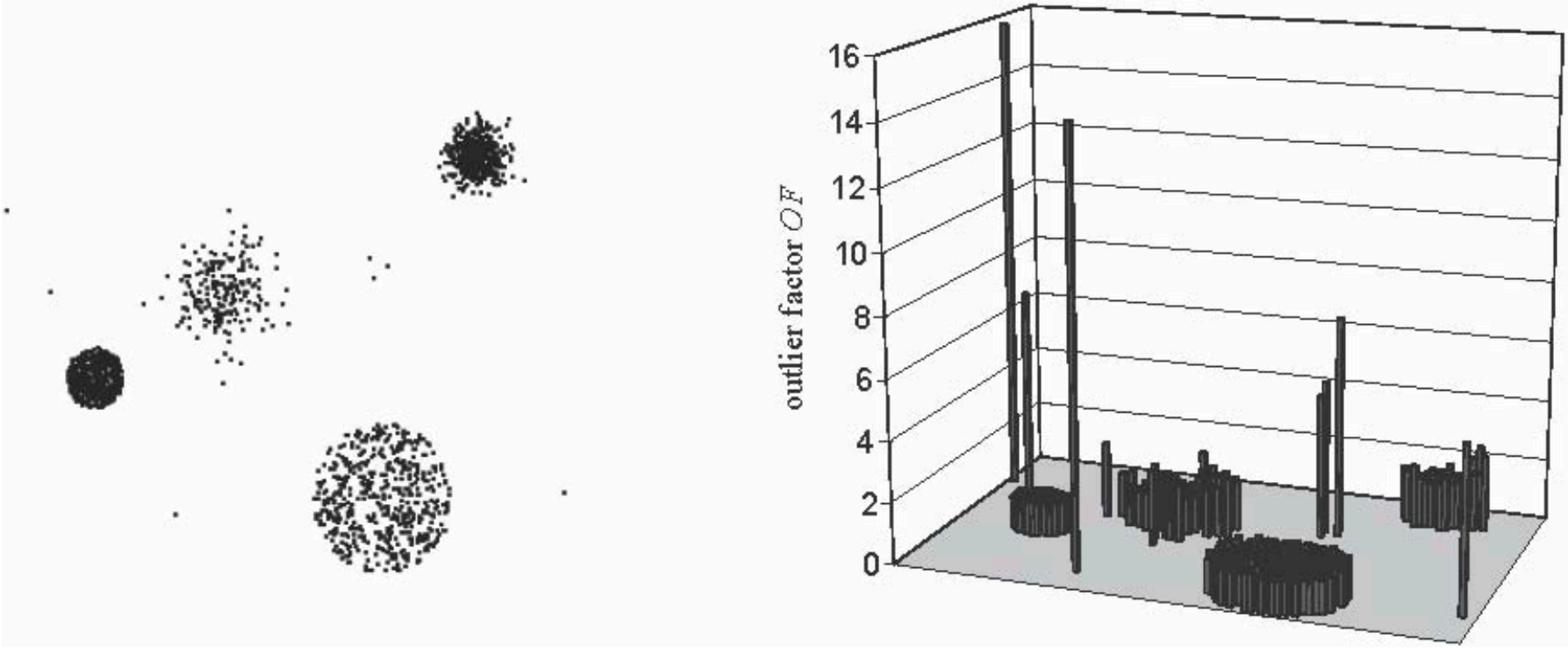
  - Local reachability distance (lrd) of point *p*
    - Inverse of the average reachability distances of the *k*NNs of *p*

$$lrd_k(p) = 1/\left(\frac{\sum_{o\in kNN(p)} reach-dist_k(p,o)}{Card(kNN(p))}\right)$$

  - Local outlier factor (LOF) of point *p*
    - Average ratio of lrds of neighbors of *p* and lrd of *p*

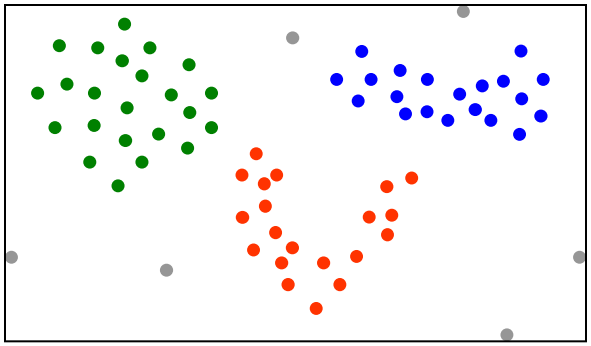
$$LOF_k(p) = \frac{\sum_{o\in kNN(p)} \frac{lrd_k(o)}{lrd_k(p)}}{Card(kNN(p))}$$

LOF evaluation



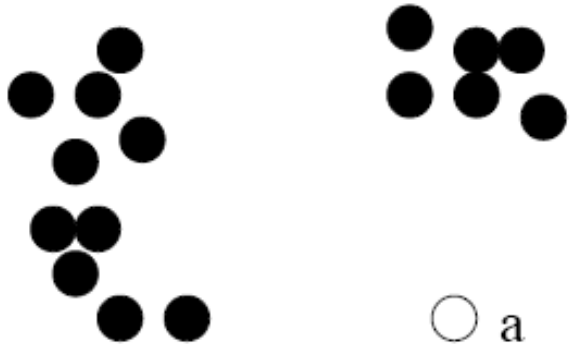
(Density Based) Clustering and Outlier Detection

- Complementarity (same result per object) of clustering and outlier detection
- E.g. DBSCAN
  - A cluster is a maximal set of density-connected objects, separated by sparsely populated areas in feature space
  - Every object not contained in any cluster is an outlier

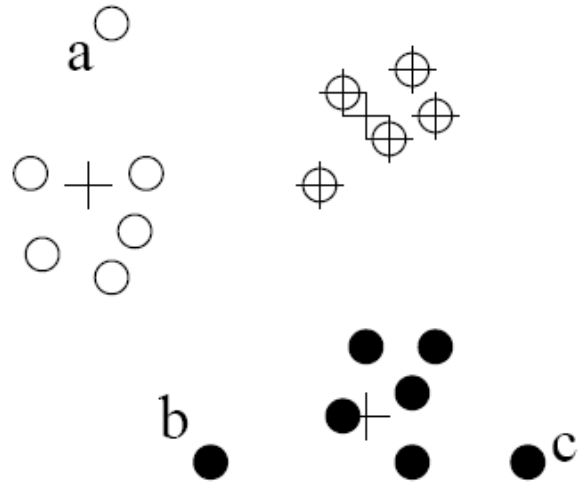


Clustering-Based Outlier Detection

- Example 1: DBSCAN noise as outliers



- Example 2: Far from its closest cluster
  - Using e.g. k-means, partition data points into clusters
  - For each object o, assign an outlier score based on its distance from its closest center
    - If dist(o, c_o)/avg_dist(c_o) is large, likely an outlier



Clustering-Based Method: Strength and Weakness

- Strength
  - Detect outliers without requiring any labeled data
  - Works for many types of data
  - Clusters can be regarded as summaries of the data
  - Once the cluster are obtained, need only compare any object against the clusters to determine whether it is an outlier (fast)

- Weakness
  - Effectiveness depends highly on the clustering method used—they may not be optimized for outlier detection
  - High computational cost: Need to first find clusters

Summary

- Density-based clustering
  - Handle arbitrary shapes and noise
  - Can be slow, finding parameters can be cumbersome (use e.g. OPTICS)
- Hierarchical clustering
  - Hierarchical overview over different possible clusterings
  - Can be converted into “flat” clustering if necessary
- Approximative clustering
  - When you need speed at the expense of accuracy
- Clustering evaluation measures
  - External measures: given ground truth grouping, check quality
  - Internal measures: consider the structure of the clusters (e.g. silhouette coefficient)
- Outlier detection
  - Finding deviating items and errors in the data
  - Statistics: find deviations per attribute / pairs of attributes or assume distribution
  - Distance or density-based
  - Score based methods provide ranking (LOF), no outlier / inlier cut-off necessary

References

- https://jakevdp.github.io/PythonDataScienceHandbook/ Data Science in Python
- https://wesmckinney.com/book/ Python for Data Science
- https://scikit-learn.org Scikit-learn
- https://pandas.pydata.org/ Pandas
- https://dataminingbook.info/book_html/ Mohammed J. Zaki, Wagner Meira, Jr., Data Mining and Machine Learning: Fundamental Concepts and Algorithms, 2nd Edition, Cambridge University Press, March 2020. ISBN: 978-1108473989.
- J. Han and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2000.
- M. H. Dunham. Data Mining: Introductory and Advanced Topics. Prentice Hall, 2003
- Outlier detection tutorial https://imada.sdu.dk/~zimek/publications/KDD2010/kdd10-outlier-tutorial.pdf
- Zimek, Arthur, and Peter Filzmoser. "There and back again: Outlier detection between statistical reasoning and data mining algorithms." Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8.6 (2018): e1280. https://wires.onlinelibrary.wiley.com/doi/pdfdirect/10.1002/widm.1280