DATABASE SYSTEMS GROUP

Ludwig-Maximilians-Universität München
Institut für Informatik
Lehr- und Forschungseinheit für Datenbanksysteme

LMU

Knowledge Discovery in Databases

WiSe 2017/18

Kapitel 4: Clustering

Vorlesung: Prof. Dr. Peer Kröger

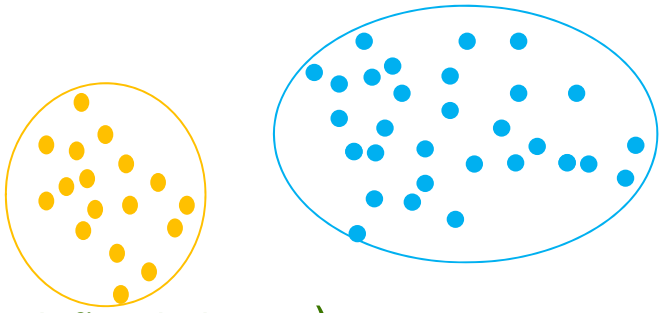
Übungen: Anna Beer, Florian Richter

Contents

What is Clustering?

Grouping a set of data objects into clusters

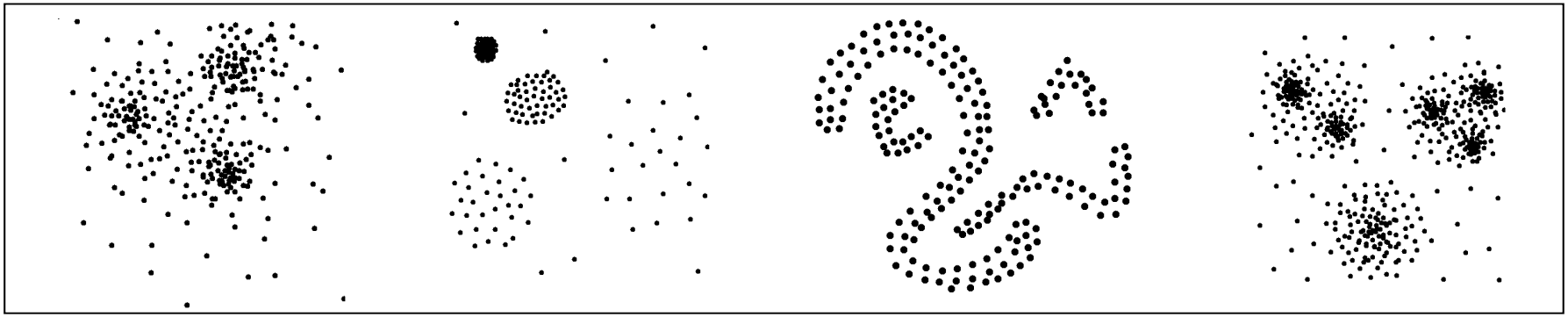
- Cluster: a collection of data objects
  1) *Similar* to one another within the same cluster
  2) *Dissimilar* to the objects in other clusters



Clustering = ***unsupervised "classification"*** (no predefined classes)

Typical usage

- As a *stand-alone tool* to get insight into data distribution
- As a *preprocessing step* for other algorithms



General Applications of Clustering

Preprocessing – as a data reduction (instead of sampling), e.g.

- Image data bases (color histograms for filter distances)
- Stream clustering (handle endless data sets for offline clustering)

Pattern Recognition and Image Processing

Spatial Data Analysis

- create thematic maps in Geographic Information Systems by clustering feature spaces
- detect spatial clusters and explain them in spatial data mining

Business Intelligence (especially market research)

WWW (Documents for Web Content Mining, Web-logs for Web Usage Mining, ...)

Biology

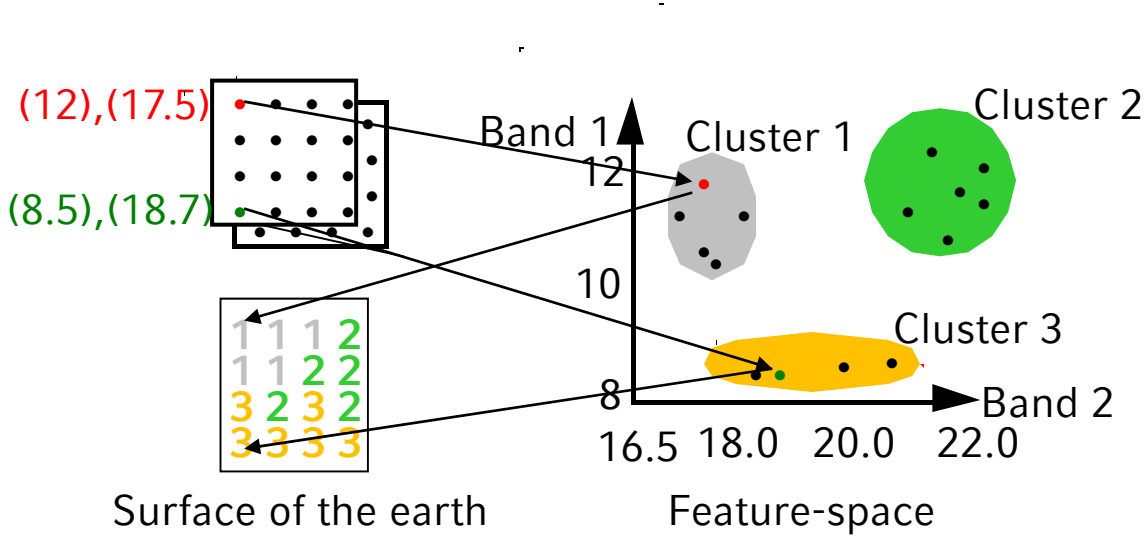
- Clustering of gene expression data

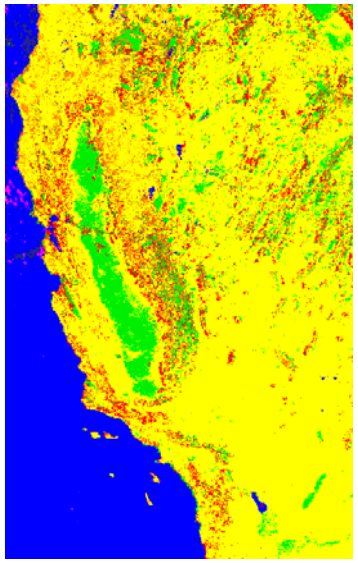
.....

An Application Example: Thematic Maps

Satellite images of a region in different wavelengths (bands)

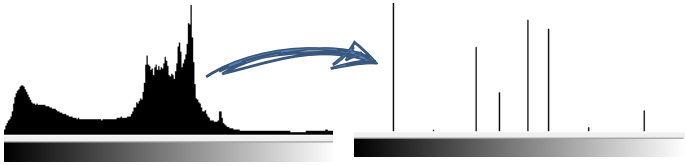
- Each point on the surface maps to a high-dimensional feature vector $p = (x_1, \ldots, x_d)$ where x_i is the recorded intensity at the surface point in band i.
- Assumption: each different land-use reflects and emits light of different wavelengths in a characteristic way.



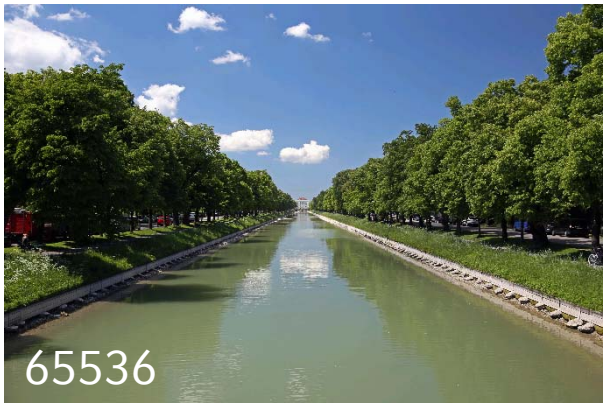


An Application Example: Downsampling Images

- Reassign color values to k distinct colors
- Cluster pixels using color difference, not spatial data

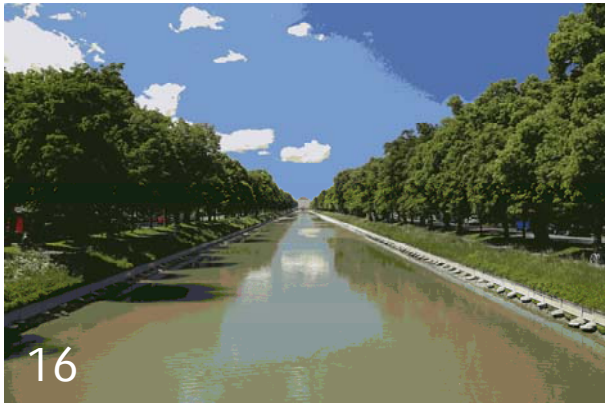


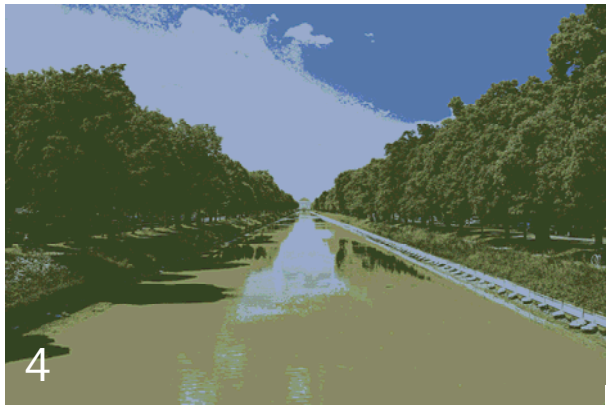
58483 KB



19496 KB

9748 KB





Major Clustering Approaches

Partitioning algorithms
- Find k partitions, minimizing some objective function

Probabilistic Model-Based Clustering (EM)

Density-based
- Find clusters based on connectivity and density functions

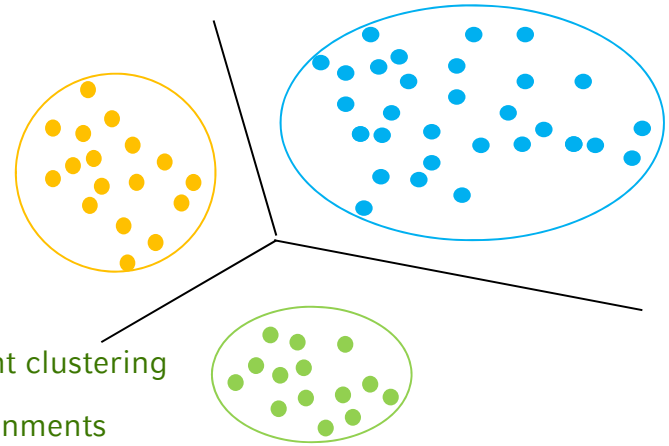
Hierarchical algorithms
- Create a hierarchical decomposition of the set of objects

Other methods
- Grid-based
- Neural networks (SOM's)
- Graph-theoretical methods
- Subspace Clustering
- . . .

Contents

Partitioning Algorithms: Basic Concept

- Goal: Construct a partition of a database D of n objects into a set of k ($k < n$) clusters $C_1, \ldots, C_k$ ($C_i \subset D, C_i \cap C_j = \emptyset \Leftrightarrow C_i \neq C_j, \bigcup C_i = D$) minimizing an objective function.
  - Exhaustively enumerating all possible partitions into k sets in order to find the global minimum is too expensive.



- Popular heuristic methods:
  - Choose k representatives for clusters, e.g., randomly
  - Improve these initial representatives iteratively:
    - Assign each object to the cluster it "fits best" in the current clustering
    - Compute new cluster representatives based on these assignments
    - Repeat until the change in the objective function from one iteration to the next drops below a threshold

- Examples of representatives for clusters
  - k-means: Each cluster is represented by the center of the cluster
  - k-medoid: Each cluster is represented by one of its objects

Contents

1) Introduction to clustering
2) Partitioning Methods
   - K-Means
   - Variants: K-Medoid, K-Mode, K-Median
   - Choice of parameters: Initialization, Silhouette coefficient
3) Probabilistic Model-Based Clusters: Expectation Maximization
4) Density-based Methods: DBSCAN
5) Hierarchical Methods
   - Agglomerative and Divisive Hierarchical Clustering
   - Density-based hierarchical clustering: OPTICS
6) Evaluation of Clustering Results
7) Further Clustering Topics
   - Scaling Up Clustering Algorithms
   - Outlier Detection

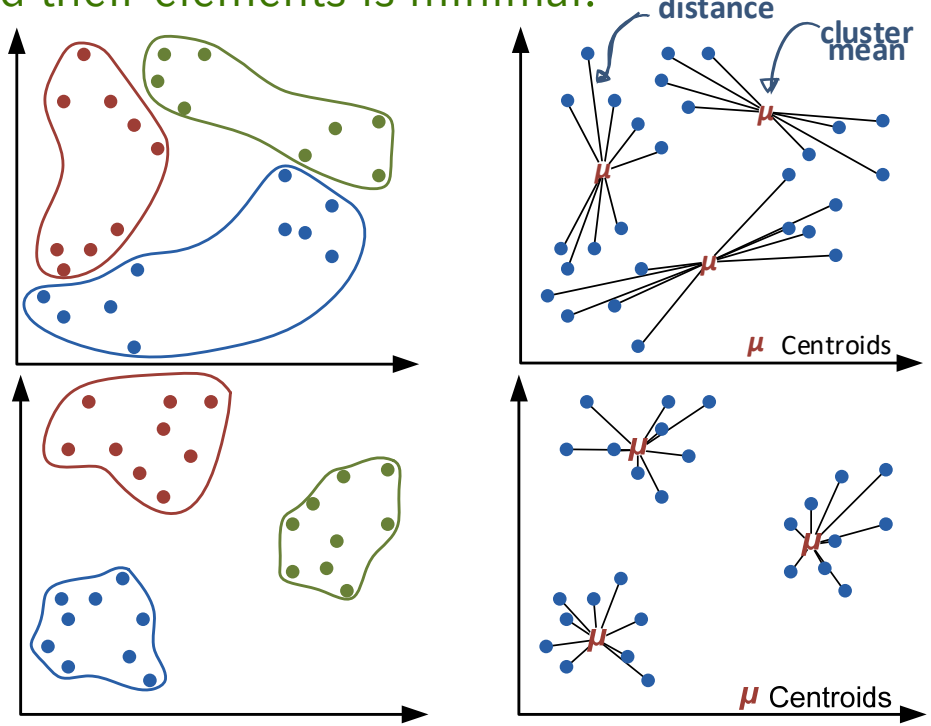
K-Means Clustering: Basic Idea

Idea of K-means: find a clustering such that the *within-cluster variation* of each cluster is small and use the *centroid* of a cluster as representative.

Objective: For a given *k*, form *k* groups so that the sum of the (squared) distances between the mean of the groups and their elements is minimal.

Poor Clustering
(large sum of distances)

Optimal Clustering
(minimal sum of distances)



S.P. Lloyd: Least squares quantization in PCM. In IEEE Information Theory, 1982 (original version: technical report, Bell Labs, 1957)
J. MacQueen: *Some methods for classification and analysis of multivariate observation*, In Proc. of the 5th Berkeley Symp. on Math. Statist. and Prob., 1967.

K-Means Clustering: Basic Notions

Objects $p = (p_1, \ldots, p_d)$ are points in a d-dimensional vector space

(the mean μ_S of a set of points S must be defined: $\mu_S = \frac{1}{|S|}\sum_{p \in S} p$)

$C(p)$: the cluster p is assigned to

Measure for the compactness of a **cluster** C_j (sum of squared errors):

$$SSE(C_j) = \sum_{p \in C_j} dist\left(p, \mu_{C_j}\right)^2$$

Measure for the compactness of a **clustering** $\mathcal{C}$**:**

$$SSE(\mathcal{C}) = \sum_{C_j \in \mathcal{C}} SSE(C_j) = \sum_{p \in DB} dist\left(p, \mu_{C(p)}\right)^2$$

Optimal Partitioning: $\underset{\mathcal{C}}{\operatorname{argmin}}\, SSE(\mathcal{C})$

Optimizing the within-cluster variation is computationally challenging

(NP-hard) → use efficient heuristic algorithms

K-Means Clustering: Algorithm

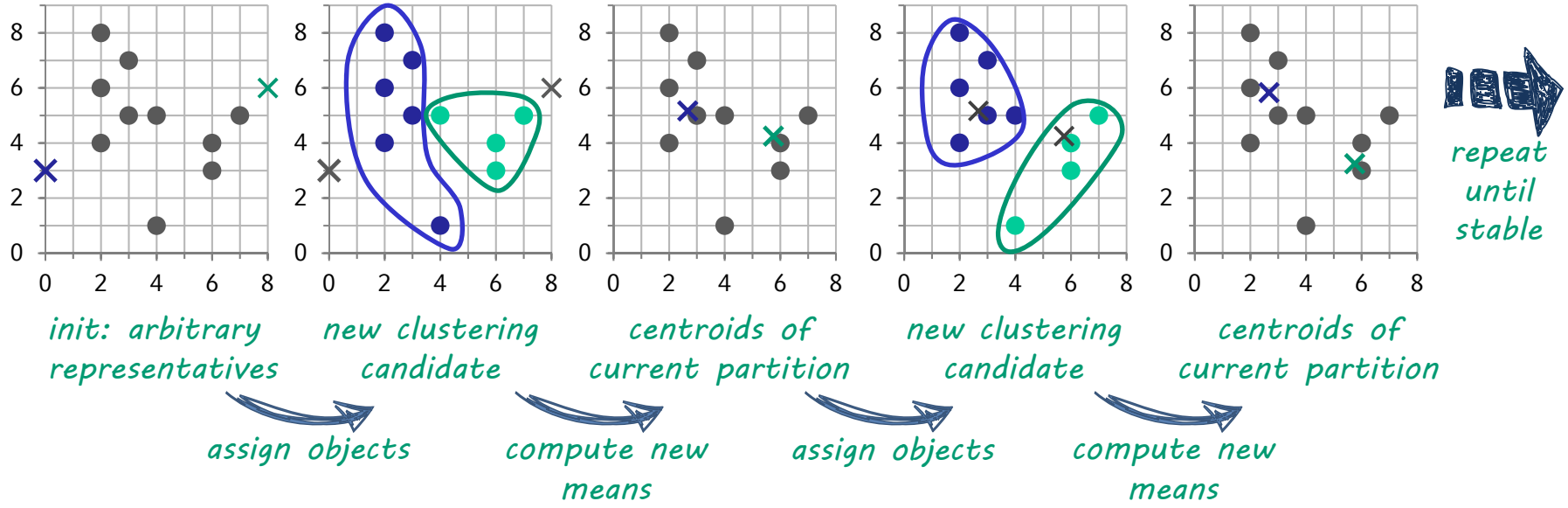
k-Means algorithm (Lloyd's algorithm):

Given *k*, the *k*-means algorithm is implemented in 2 main steps:

Initialization: Choose *k* arbitrary representatives

Repeat until representatives do not change:

1. Assign each object to the cluster with the nearest representative.
2. Compute the centroids of the clusters of the current partitioning.



K-Means Clustering: Discussion

Strengths

- Relatively efficient: $O(tkn)$, where n = # objects, k = # clusters, and t = # iterations
- Typically: k, $t \ll n$
- Easy implementation

Weaknesses

- Applicable only when mean is defined
- Need to specify *k*, the number of clusters, in advance
- Sensitive to noisy data and outliers
- Clusters are forced to convex space partitions (Voronoi Cells)
- Result and runtime strongly depend on the initial partition; often terminates at a *local optimum* – however: methods for a good initialization exist

Several variants of the *k*-means method exist, e.g., ISODATA

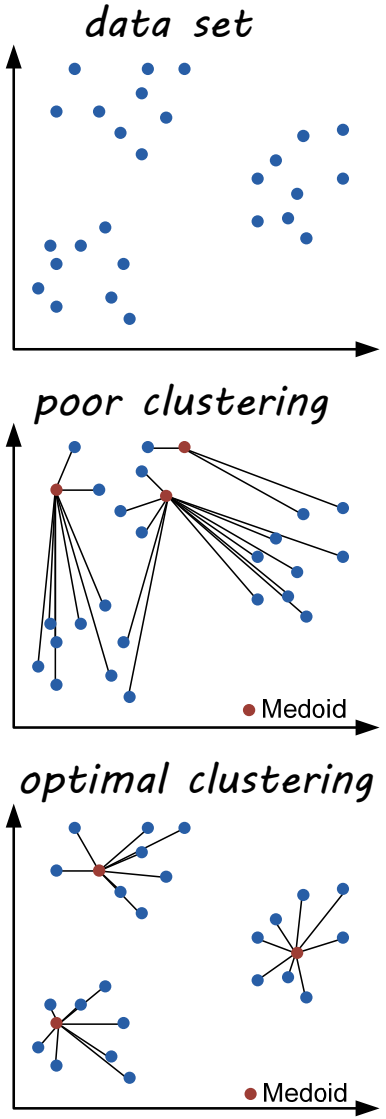
- Extends *k*-means by methods to eliminate very small clusters, merging and split of clusters; user has to specify additional parameters

Contents

1) Introduction to clustering
2) Partitioning Methods
   - K-Means
   - Variants: K-Medoid, K-Mode, K-Median
   - Choice of parameters: Initialization, Silhouette coefficient
3) Probabilistic Model-Based Clusters: Expectation Maximization
4) Density-based Methods: DBSCAN
5) Hierarchical Methods
   - Agglomerative and Divisive Hierarchical Clustering
   - Density-based hierarchical clustering: OPTICS
6) Evaluation of Clustering Results
7) Further Clustering Topics
   - Scaling Up Clustering Algorithms
   - Outlier Detection

K-Medoid, *K*-Modes, *K*-Median Clustering: Basic Idea

- Problems with K-Means:
  - Applicable only when mean is defined (vector space)
  - Outliers have a strong influence on the result
- The influence of outliers is intensified by the use of the *squared* error → use the absolute error (total distance instead): $TD(C) = \sum_{p \in C} dist(p, m_{c(p)})$ and $TD(\mathcal{C}) = \sum_{C_i \in \mathcal{C}} TD(C_i)$
- Three alternatives for using the Mean as representative:
  - *Medoid*: representative object "in the middle"
  - *Mode*: value that appears most often
  - *Median*: (artificial) representative object "in the middle"
- Objective as for k-Means: Find *k* representatives so that, the sum of the distances between objects and their closest representative is minimal.



K-Medoid Clustering: PAM Algorithm

Partitioning Around Medoids [Kaufman and Rousseeuw, 1990]

- Given *k*, the *k*-medoid algorithm is implemented in 3 steps:

  Initialization: Select *k* objects arbitrarily as initial medoids (representatives); assign each remaining (non-medoid) object to the cluster with the nearest representative, and compute $TD_{current}$.

  Repeat

  1. **For** *each* pair (medoid *M*, non-medoid *N*)
     - compute the value $TD_{N \leftrightarrow M}$, i.e., the value of TD for the partition that results when "swapping" *M* with *N*
  2. Select the best pair (*M, N*) for which $TD_{N \leftrightarrow M}$ is minimal
  3. **If** $TD_{N \leftrightarrow M} < TD_{current}$
     - Swap *N* with *M*
     - Set $TD_{current} := TD_{N \leftrightarrow M}$

  Until nothing changes

- Problem of PAM: high complexity ($O(tk(n-k)^2)$)

Kaufman L., Rousseeuw P. J., *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley & Sons, 1990.

K-Medoid Clustering: CLARANS Algorithm

Optimization: CLARANS [Ng & Han 1994]

- Trading accuracy for speed
- Two additional tuning parameters: *maxneighbor* und *numlocal*
- At most *maxneighbor* of pairs (M,N) are considered in each iteration (Step 1)
- Best first: take the first pair (M,N) that reduces the TD-value instead of evaluating all (*maxneighbor*) pairs (worst-case still *maxneighbor*)
- Termination after *numlocal* iteration even if convergence is not yet reached

Kaufman L., Rousseeuw P. J., *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley & Sons, 1990.

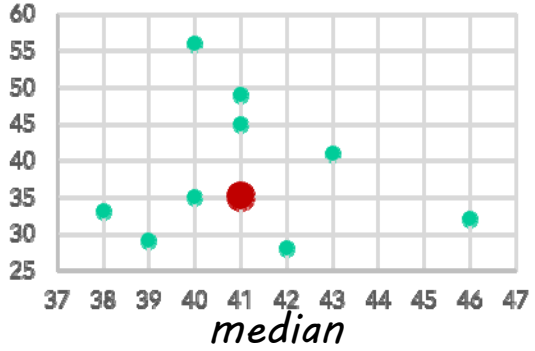
K-Median Clustering

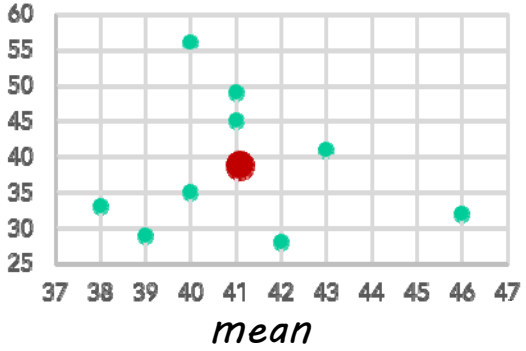
Problem: Sometimes, data is not numerical

Idea: If there is an ordering on the data $X = \{x_1, x_2, x_3, \ldots, x_n\}$, use median instead of mean

$$Median(\{x\}) = x$$
$$Median(\{x, y\}) \in \{x, y\}$$
$$Median(X) = Median(X - \min X - \max X), \qquad if\ |X| > 2$$

- A median is computed in each dimension independently and can thus be a combination of multiple instances
  → median can be efficiently computed for ordered data
- Different strategies to determine the "middle" in an array of even length possible





K-Mode Clustering: First Approach [Huang 1997]

Given: $X \subseteq \Omega = A_1 \times A_2 \times \cdots \times A_d$ is a set of n objects, each described by d categorical attributes A_i $(1 \leq i \leq d)$

Mode: a mode of X is a vector $M = [m_1, m_2, \cdots, m_d] \in \Omega$ that minimizes

$$d(M, X) = \sum_{x_i \in X} d(x_i, M)$$

where d is a distance function for categorical values (e.g. Hamming Dist.)

→ Note: M is not necessarily an element of X

→ For Hamming: the mode is determined by the most frequent value in each attribute

Huang, Z.: *A Fast Clustering Algorithm to Cluster very Large Categorical Data Sets in Data Mining*, In DMKD, 1997.

K-Mode Clustering: First Approach [Huang 1997]

Theorem to determine a Mode:

Let $f(c,j,X) = \frac{1}{n} \cdot |\{x \in X |\, x[j] = c\}|$ be the relative frequency of category c of attribute A_j in the data, then:

$$d(M,X) \text{ is minimal} \Leftrightarrow \forall j \in \{1,\ldots,d\}: \forall c \in A_j: f(\mathrm{m_j}, \mathrm{j}, \mathrm{X}) \geq f(c,j,X)$$

→ this allows to use the k-means paradigm to cluster categorical data without loosing its efficiency

→ Note: the mode of a dataset might be not unique

K-Modes algorithm proceeds similar to k-Means algorithm

Huang, Z.: *A Fast Clustering Algorithm to Cluster very Large Categorical Data Sets in Data Mining*, In DMKD, 1997.

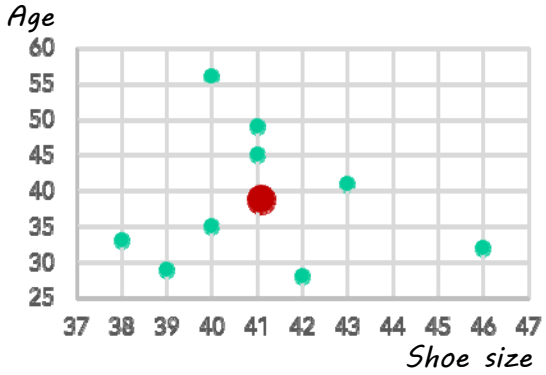
K-Mode Clustering: Example

Employee-ID	Profession	Household Pets
#133	Technician	Cat
#134	Manager	None
#135	Cook	Cat
#136	Programmer	Dog
#137	Programmer	None
#138	Technician	Cat
#139	Programmer	Snake
#140	Cook	Cat
#141	Advisor	Dog

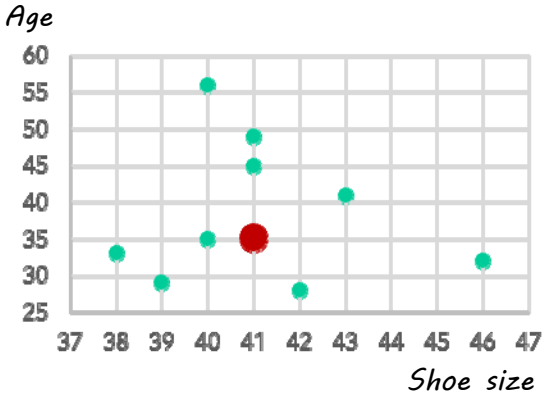
Profession: (**Programmer: 3**, Technician: 2, Cook: 2, Advisor: 1, Manager:1)
Household Pet: (**Cat: 4**, Dog: 2, None: 2, Snake: 1)

Mode is (Programmer, Cat)
Remark: (Programmer, Cat) ∉ DB

K-Means/Medoid/Mode/Median overview

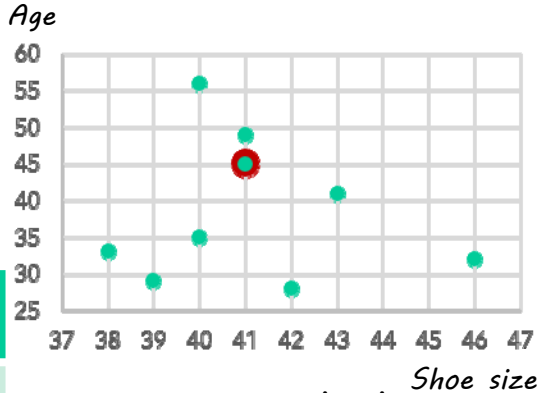


mean



median

Employee-ID	Profession	Shoe size	Age
#133	Technician	42	28
#134	Manager	41	45
#135	Cook	46	32
#136	Programmer	40	35
#137	Programmer	41	49
#138	Technician	43	41
#139	Programmer	39	29
#140	Cook	38	33
#141	Advisor	40	56



medoid

Profession: Programmer
Shoe size: 40/41
Age: n.a.

mode

K-Means/Median/Mode/Medoid Clustering: Discussion

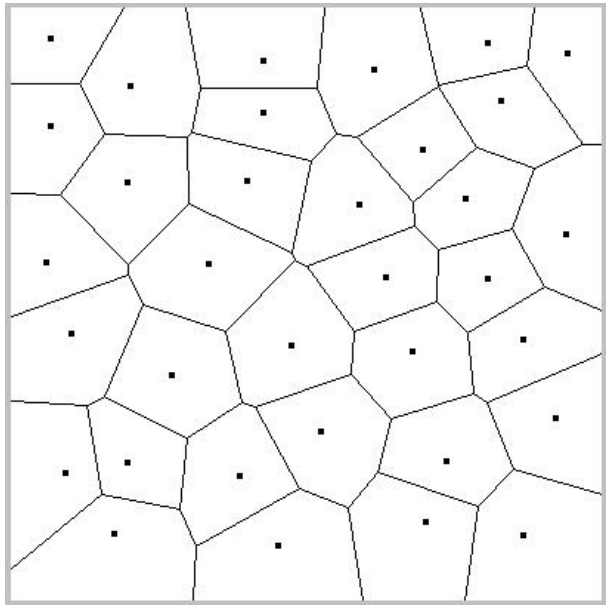
	k-Means	*k*-Median	K-Mode	K-Medoid
data	numerical data (mean)	ordered attribute data	categorical attribute data	metric data
efficiency	high $O(tkn)$	high $O(tkn)$	high $O(tkn)$	low $O(tk(n-k)^2)$
sensitivity to outliers	high	low	low	low

- Strength
  - Easy implementation (→ many variations and optimizations in the literature)
- Weakness
  - Need to specify *k*, the number of clusters, in advance
  - Clusters are forced to convex space partitions (Voronoi Cells)
  - Result and runtime strongly depend on the initial partition; often terminates at a *local optimum* – however: methods for a good initialization exist

Voronoi Model for convex cluster regions

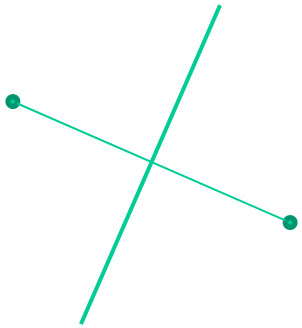
Definition: Voronoi diagram

- For a given set of points $P = \{p_i | \; i = 1 \ldots k\}$ (here: cluster representatives), a Voronoi diagram partitions the data space in Voronoi cells, one cell per point.
- The cell of a point $p \in P$ covers all points in the data space for which p is the nearest neighbors among the points from P.



Observations

- The Voronoi cells of two neighboring points $p_i, p_j \in P$ are separated by the perpendicular hyperplane („Mittelsenkrechte“) between p_i and p_j.
- As Voronoi cells are intersections of half spaces, they are convex regions.



Voronoi Model for convex cluster regions(2)

Voronoi-parcellation ≠ convex hull of cluster