Knowledge Discovery in Databases
WiSe 2017/18

Kapitel 4: Clustering

Vorlesung: Prof. Dr. Peer Kröger

Übungen: Anna Beer, Florian Richter
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
   - Ensemble Clustering
   - Discussion: an alternative view on DBSCAN
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.

![Diagram showing clustering and feature space](image_url)
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
65536 256 16
8 4 2
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
- . . .
1) Introduction to clustering

2) Partitioning Methods
   - K-Means
   - K-Medoid
   - Choice of parameters: Initialization, Silhouette coefficient

3) Expectation Maximization: a statistical approach

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
   - Ensemble Clustering
   - Discussion: an alternative view on DBSCAN
   - Outlier Detection
Partitioning Algorithms: Basic Concept

- Goal: Construct a partition of a database $D$ of $n$ objects into a set of $k$ ($k < n$) clusters $C_1, ..., C_k$ ($C_i \subset D, C_i \cap C_j = \emptyset \iff C_i \neq C_j, \bigcup C_i = D$) 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

Clustering ➔ Partitioning Methods
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)
**K-Means Clustering: Basic Notions**

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

(the mean $\mu_S$ 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):

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

Measure for the compactness of a clustering $C$:

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

Optimal Partitioning: $\arg\min_C SSE(C)$

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:

1. **Initialization:** Choose $k$ arbitrary representatives
2. 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.

---

**Algorithm Steps:**

1. **Initialization:** Choose $k$ arbitrary representatives.
2. **Repeat:**
   - Assign each object to the cluster with the nearest representative.
   - Compute the new centroids of the clusters.

**Steps Diagram:**

- **Initial step:** Choose $k$ representatives.
- **Assignment step:** Assign each object to the nearest cluster.
- **Centroid update step:** Compute the new centroids.
- **Check convergence:** Repeat until the centroids do not change.

---

**Clustering → Partitioning Methods → K-Means**
K-Means Clustering: Discussion

Strengths
- Relatively efficient: $O(tkn)$, where $n = \#$ objects, $k = \#$ clusters, and $t = \#$ iterations
- Typically: $k, t << n$
- 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):
  
  \[ TD(C) = \sum_{p \in C} \text{dist}(p, m_{c(p)}) \]
  
  and
  
  \[ TD(C) = \sum_{C_i \in C} TD(C_i) \]

- 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 $T_{D_{current}}$.

  Repeat
  1. For each pair (medoid $M$, non-medoid $N$)
     - compute the value $T_{D_{N\leftrightarrow M}}$, 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 $T_{D_{N\leftrightarrow M}}$ is minimal
  3. If $T_{D_{N\leftrightarrow M}} < T_{D_{current}}$
     - Swap $N$ with $M$
     - Set $T_{D_{current}} := T_{D_{N\leftrightarrow M}}$

  Until nothing changes

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

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

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

$$\text{Median} \{x\} = x$$
$$\text{Median} \{x, y\} \in \{x, y\}$$

$$\text{Median}(X) = \text{Median}(X - \min X - \max X), \quad \text{if } |X| > 2$$

- 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.

Clustering → Partitioning Methods → Variants: K-Medoid, K-Mode, K-Median
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:

\[
d(M, X) \text{ is minimal} \iff \forall j \in \{1, ..., d\} : \forall c \in A_j : f(m_j, j, X) \geq f(c, j, X)
\]

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

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

\( K \)-Modes algorithm proceeds similar to k-Means algorithm
K-Mode Clustering: Example

<table>
<thead>
<tr>
<th>Employee-ID</th>
<th>Profession</th>
<th>Household Pets</th>
</tr>
</thead>
<tbody>
<tr>
<td>#133</td>
<td>Technician</td>
<td>Cat</td>
</tr>
<tr>
<td>#134</td>
<td>Manager</td>
<td>None</td>
</tr>
<tr>
<td>#135</td>
<td>Cook</td>
<td>Cat</td>
</tr>
<tr>
<td>#136</td>
<td>Programmer</td>
<td>Dog</td>
</tr>
<tr>
<td>#137</td>
<td>Programmer</td>
<td>None</td>
</tr>
<tr>
<td>#138</td>
<td>Technician</td>
<td>Cat</td>
</tr>
<tr>
<td>#139</td>
<td>Programmer</td>
<td>Snake</td>
</tr>
<tr>
<td>#140</td>
<td>Cook</td>
<td>Cat</td>
</tr>
<tr>
<td>#141</td>
<td>Advisor</td>
<td>Dog</td>
</tr>
</tbody>
</table>

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

Clustering → Partitioning Methods → Variants: K-Medoid, K-Mode, K-Median
**K-Means/Median/Mode/Medoid Clustering: Discussion**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>numerical data (mean)</td>
<td>ordered attribute data</td>
<td>categorical attribute data</td>
<td>metric data</td>
</tr>
<tr>
<td>efficiency</td>
<td>high $O(tkn)$</td>
<td>high $O(tkn)$</td>
<td>high $O(tkn)$</td>
<td>low $O(tk(n - k)^2)$</td>
</tr>
<tr>
<td>sensitivity to outliers</td>
<td>high</td>
<td>low</td>
<td>low</td>
<td>low</td>
</tr>
</tbody>
</table>

- **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
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-parcellation ≠ convex hull of cluster