VK Multimedia
Information Systems

Mathias Lux, mlux@itec.uni-klu.ac.at

Dienstags, 16.00 Uhr c.t., E X.XX

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 2.0 License. See http://creativecommons.org/licenses/by-nc-sa/2.0/at/
Selected Topics

- Abgaben Video summary
- Ergebnisse Evaluierung
- Clustering
  - HAC & K-MEANS
Abgaben

510 frames a 2 pixel
Ergebnisse Evaluierung ...
What is Clustering?

- Clustering is **unsupervised classification** with:
  - Maximized similarity in groups
  - Minimized similarity between groups
- Clustering creates **structure**

Clustering slides adapted from Benno Stein, University of Weimar
http://www.uni-weimar.de/cms/Lecture-Notes.550.0.html
and “Data Clustering: A Review”, Jain, Murty & Flynn, 1999
Clustering: Applications

- Automatic Media Organization
  - Creating (hierarchical) groups

- Data Mining & Analysis
  - Finding Patterns & outliers

- Visualization
  - Visual analysis
  - e.g. huge graphs / lists ...
Clustering: Applications

Indexing of Multimedia Documents
- Documents are organized in Clusters
- Centroids / Medoids describe Cluster

Search Process:
- Select best matching clusters
- Linear search within cluster members

Evaluation
- How many 'good results' are missed
- What is the average performance gain
Clustering: Example

- Object has $d$ features
  - $d$ ... number of dimensions

- For 2 dimensions:
Clustering: Definition

Let $X$ be a set of Objects. A Clustering $C$ of $X$ with $C = \{C_1, C_2, \ldots, C_k\}$, $C_i \subseteq X$ is a pairwise disjunctive segmentation of $X$ in subsets $C_i$ with $\bigcup_{C_i \in C} C_i = X$. 
Clustering Techniques

Hierarchical
- Single Link
- Complete Link

Partitional
- Square Error
- Graph Theoretic
- Mixture Resolving
- Mode Seeking
- k-means

Expectation Maximization
Hierarchical Clustering

Input: $G = \langle V, E, w \rangle$. Weighted graph.
\(d_C\). Distance measure between two clusters.

Output: $T = \langle V_T, E_T \rangle$. Cluster hierarchy or dendrogram.

1. $\mathcal{C} = \{\{v\} | v \in V\}$  // define initial clustering
2. $V_T = \{v_C | C \in \mathcal{C}\}$, $E_T = \emptyset$  // define initial dendrogram
3. WHILE $|\mathcal{C}| > 1$ DO
4. update_distance_matrix($\mathcal{C}, G, d_C$)
5. $\{C, C'\} = \arg\min_{\{C_i, C_j\} \in \mathcal{C}: C_i \neq C_j} d_C(C_i, C_j)$
6. $\mathcal{C} = (\mathcal{C} \setminus \{C, C'\}) \cup \{C \cup C'\}$  // clustering
7. $V_T = V_T \cup \{v_{C, C'}\}$, $E_T = E_T \cup \{(v_C, C'), (v_C, v_C'), (v_{C, C'}, v_{C'})\}$  // dendrogram
8. ENDDO
9. RETURN($T$)
HAC: Example
HAC: Example
HAC: Example
HAC: Example
HAC: Example
HAC: Example
HAC: Example
HAC: Example
HAC: Example
Cluster Distance

\[ d_C(C, C') = \min_{u \in C} \max_{v \in C'} d(u, v) \quad \text{Single-Link (Nearest-Neighbor)} \]

\[ d_C(C, C') = \max_{u \in C} \min_{v \in C'} d(u, v) \quad \text{Complete-Link (Furthest-Neighbor)} \]

\[ d_C(C, C') = \frac{1}{|C| \cdot |C'|} \sum_{u \in C} \sum_{v \in C'} d(u, v) \quad (\text{Group-}) \text{Average-Link} \]

\[ d_C(C, C') = \sqrt{\frac{2 \cdot |C| \cdot |C'|}{|C| + |C'|} \cdot ||\bar{u} - \bar{v}||} \quad \text{Ward (Varianz)} \]
Single Link Problem: Chaining
Single Link Problem: Chaining
Single Link Problem: Chaining
Single Link Problem: Chaining
Single Link Problem: Chaining
Single Link Problem: Chaining
Single Link Problem: Chaining
Single Link Problem: Chaining
Single Link Problem: Chaining

![Diagram of a network showing a single link problem with Chaining technique applied.]
Single Link Problem: Chaining
Complete Link Problem: Overlap
Complete Link Problem: Overlap
Complete Link Problem: Overlap
Complete Link Problem: Overlap
Complete Link
Problem: Overlap
Complete Link Problem: Overlap
Complete Link
Problem: Overlap
Complete Link
Problem: Overlap
## Hierarchical Clustering: Comparison

<table>
<thead>
<tr>
<th></th>
<th>Single Link</th>
<th>Complete Link</th>
<th>Average Link</th>
<th>Ward</th>
</tr>
</thead>
<tbody>
<tr>
<td># clusters</td>
<td>small</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
</tr>
<tr>
<td>cluster type</td>
<td>stretched</td>
<td>small</td>
<td>compact</td>
<td>spherical</td>
</tr>
<tr>
<td>chaining tendency</td>
<td>high</td>
<td>low</td>
<td>low</td>
<td>low</td>
</tr>
<tr>
<td>outlier detection</td>
<td>high</td>
<td>very low</td>
<td>low</td>
<td>low</td>
</tr>
</tbody>
</table>
Partitional Clustering

- Only one partition of the data
  - No structure (dendrogram)
- Usually based on an optimization criterion
  - Iterated until “optimal” results
  - Multiple starting points
    - e.g. initial clusters
- Benefits for large data sets
  - But number of clusters has to be known
Iterative Clustering Algorithm

Input: \( G = (V, E, w) \). Weighted graph.
\( d \). Distance function for nodes in \( V \).
\( e \). Minimization criterion for cluster representatives, based on \( d \).
\( k \). Number of desired clusters.

Output: \( r_1, \ldots, r_k \). Cluster representatives.

1. \( t = 0 \)
2. \textbf{FOR} \( i = 1 \) to \( k \) \textbf{DO} \( r_i(t) = \text{choose}(V) \) \quad // \text{init representatives}
3. \textbf{REPEAT}
4. \textbf{FOR} \( i = 1 \) to \( k \) \textbf{DO} \( C_i = \emptyset \)
5. \textbf{FOREACH} \( v \in V \) \textbf{DO} \quad // \text{find nearest representative (cluster)}
6. \( x = \arg\min_{i: i \in \{1, \ldots, k\}} d(r_i(t), v) \), \( C_x = C_x \cup \{v\} \)
7. \textbf{ENDDO}
8. \textbf{FOR} \( i = 1 \) to \( k \) \textbf{DO} \( r_i(t) = \text{minimize}(e, C_i) \) \quad // \text{update}
9. \textbf{UNTIL} \( (\forall r_i : d(r_i(t), r_i(t-1)) < \varepsilon \quad \lor \quad t > t_{\text{max}}) \)
10. \textbf{RETURN} \( \{r_1(t), \ldots, r_k(t)\} \)
Iterative Clustering Algorithm

1. Select an initial partition of the patterns with a fixed number of clusters and cluster centers.

2. Assign each object to its closest cluster center and compute the new cluster centers as the centroids of the clusters. Repeat this step until convergence is achieved, i.e., until the cluster membership is stable.

3. Merge and split clusters based on some heuristic information, optionally repeating step 2.
Iterative Clustering Algorithm

- Cluster representatives: Centroids (Medoids)
- Initial cluster representatives chosen randomly
- Optimization is based on the sum of squared error (distance to centroid)
Iterative Clustering Algorithm

- Choose $k$ cluster centers to coincide with $k$ randomly-chosen objects or $k$ randomly defined points inside the hypervolume containing the objects.
- Assign each object to the closest cluster center (centroid).
- Recompute the cluster centers (centroids) using the current cluster memberships.
- If a convergence criterion is not met, go to step 2. Typical convergence criteria are: no (or minimal) reassignment of patterns to new cluster centers, or minimal decrease in squared error.
K-Means Example
K-Means Example
K-Means Example
K-Means Example
K-Means Example
K-Means Example
K-Means Example
K-Means Example
Cluster Center

\[ e(C) = \sum_{i=1}^{k} \sum_{v \in C_i} (v - r_i)^2 \quad r_i = \bar{C}_i \]

Centroid-Berechnung
(k-Means)

\[ e(C) = \sum_{i=1}^{k} \sum_{v \in C_i} |v - r_i| \quad r_i \in C_i \]

Medoid-Berechnung
(k-Medoid)

\[ e(C) = \sum_{i=1}^{k} \max_{v \in C_i} |v - r_i| \quad r_i \in C_i \]

k-Center

\[ e(C) = \sum_{i=1}^{k} \sum_{v \in V} \mu_{v_i}^2 \cdot (v - r_i)^2 \quad r_i = \frac{\sum_{v \in V} \mu_{v_i}^2 \cdot v}{\sum_{v \in V} \mu_{v_i}^2} \]

Fuzzy-k-Means
Method Comparison

● **K-Means & Fuzzy K-Means** are based on interval scaled features
  • Cluster center is artificial

● **K-Medoid & K-Center** work with arbitrary distance and similarity functions
  • Cluster center is part of the objects
  • Medoid is more robust against outliers
K-Means Problems
K-Means Problems
K-Means Problems
Thank you ...

... for your attention