

# VK Multimedia Information Systems



Mathias Lux, [mlux@itec.uni-klu.ac.at](mailto: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



<http://www.uni-klu.ac.at>

- Abgaben Videosummary
- Ergebnisse Evaluierung
- Clustering
  - HAC & K-MEANS

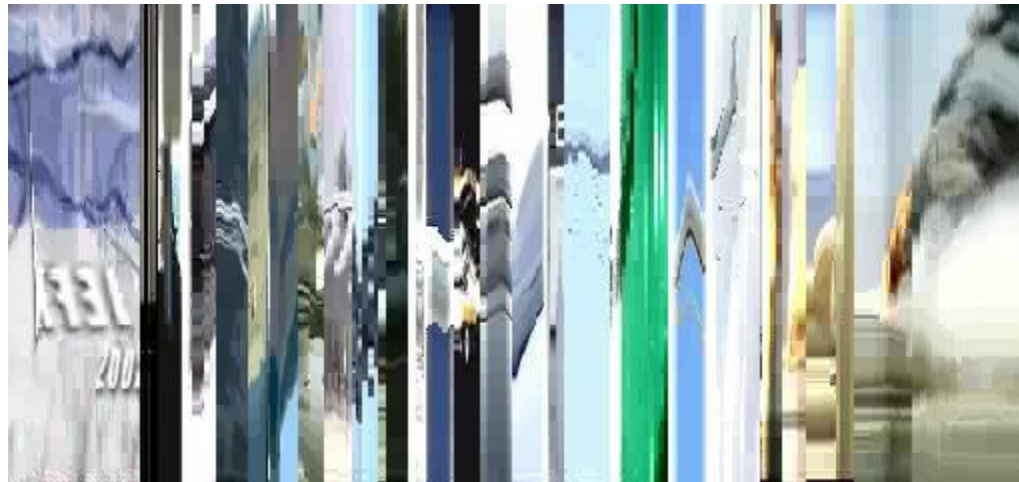


# Abgaben



<http://www.uni-klu.ac.at>

510 frames a 2 pixel





# Ergebnisse Evaluierung ...



<http://www.uni-klu.ac.at>

# What is Clustering?



<http://www.uni-klu.ac.at>

- 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



<http://www.uni-klu.ac.at>

- Automatic Media Organization
  - Creating (hierarchical) groups
- Data Mining & Analysis
  - Finding Patterns & outliers
- Visualization
  - Visual analysis
  - e.g. huge graphs / lists ...

# Clustering: Applications



<http://www.uni-klu.ac.at>

## 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



<http://www.uni-klu.ac.at>

- Object has  $d$  features
  - $d$  ... number of dimensions
- For 2 dimensions:



# Clustering: Definition



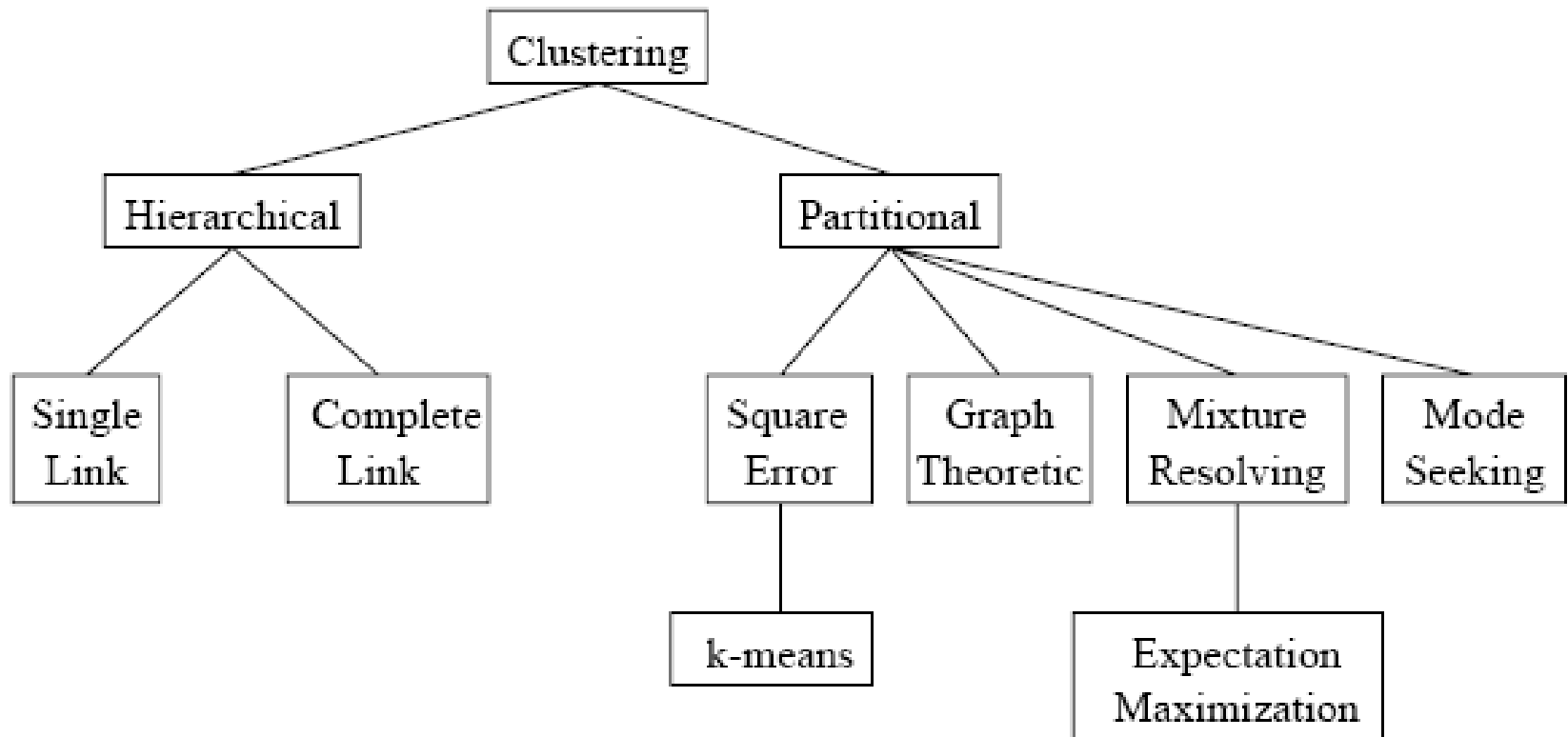
<http://www.uni-klu.ac.at>

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

# Clustering Techniques



<http://www.uni-klu.ac.at>



# Hierarchical Clustering



<http://www.uni-klu.ac.at>

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

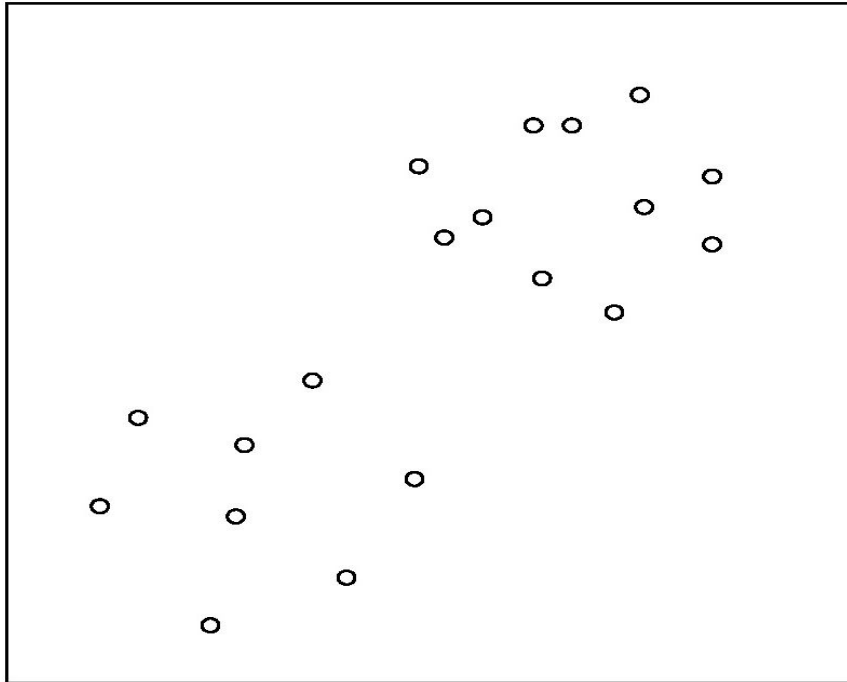
Output:  $T = \langle V_T, E_T \rangle$ . Cluster hierarchy or dendrogram.

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

# HAC: Example



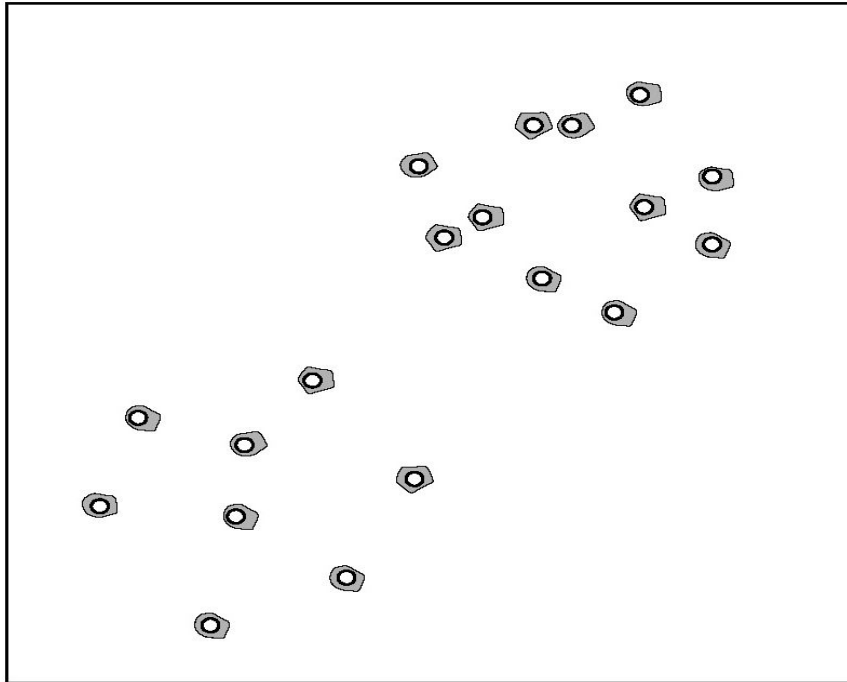
<http://www.uni-klu.ac.at>



# HAC: Example



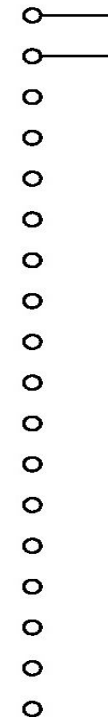
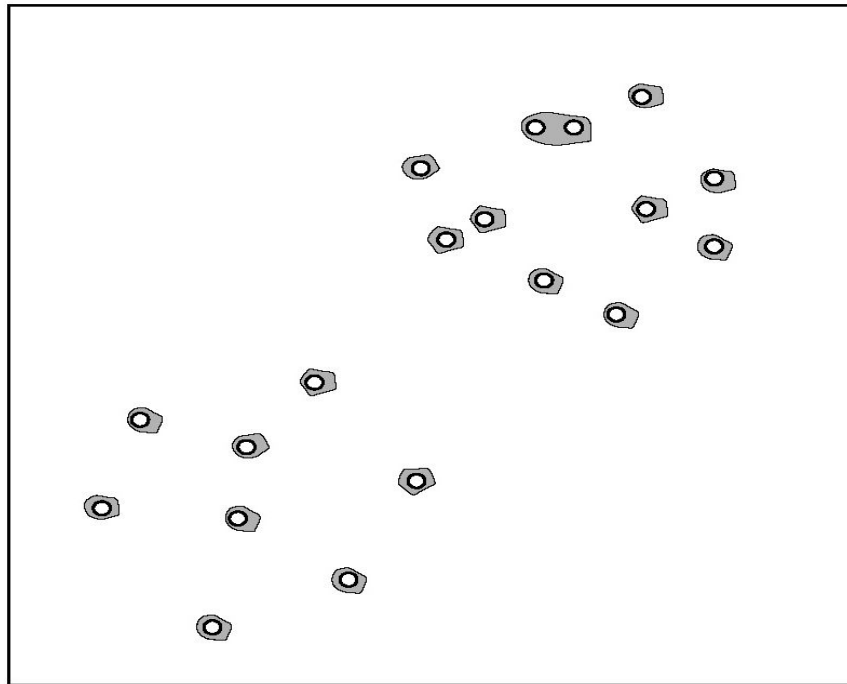
<http://www.uni-klu.ac.at>



# HAC: Example



<http://www.uni-klu.ac.at>

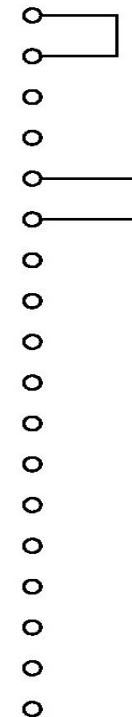
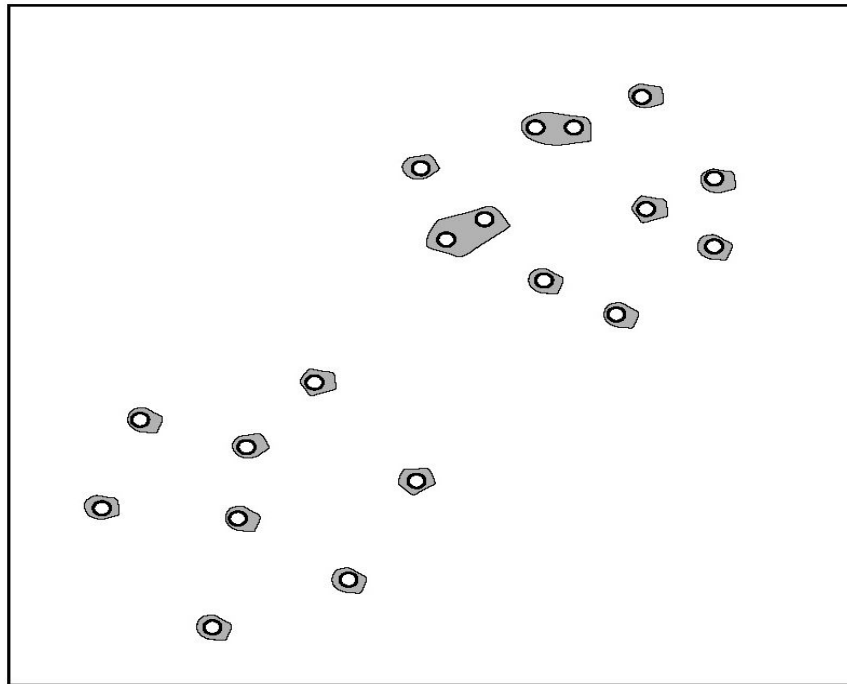


→ Distanz

# HAC: Example



<http://www.uni-klu.ac.at>

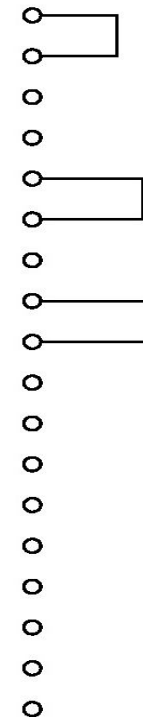
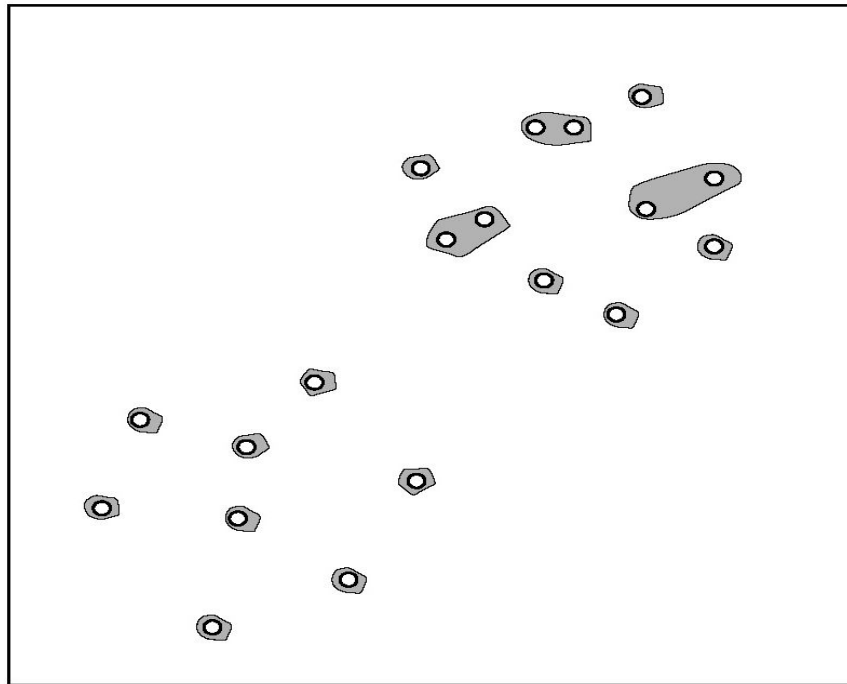


→ Distanz

# HAC: Example



<http://www.uni-klu.ac.at>

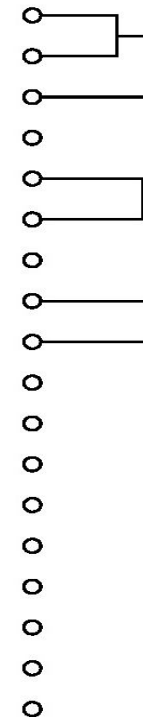
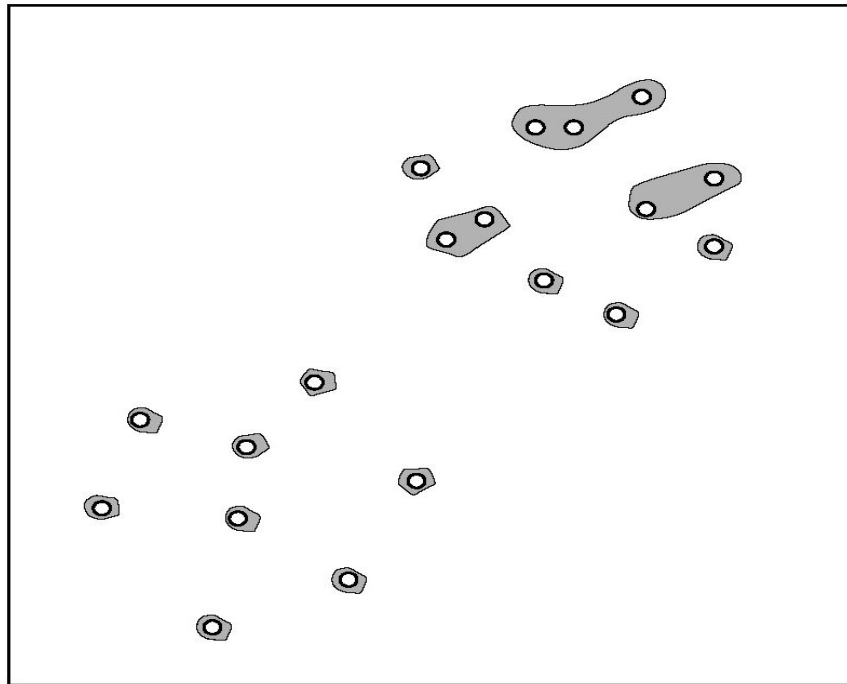


—————→ Distanz

# HAC: Example



<http://www.uni-klu.ac.at>

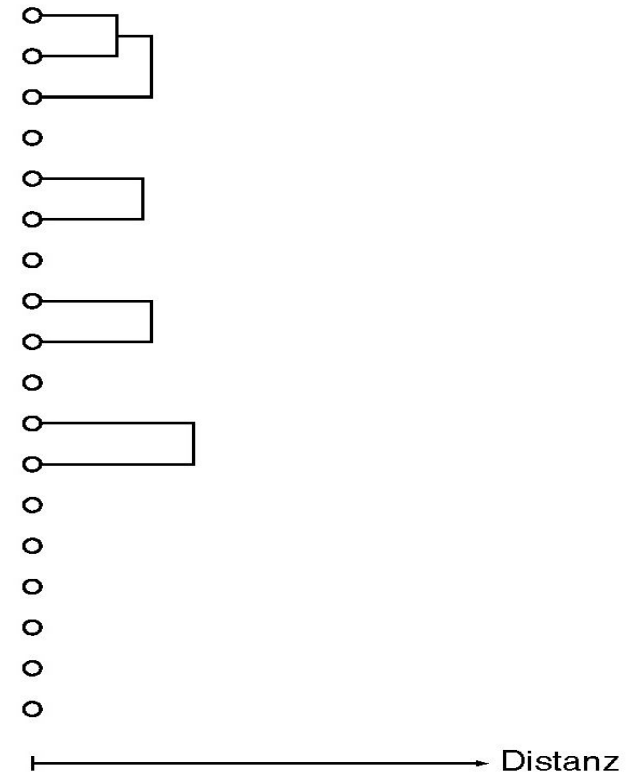
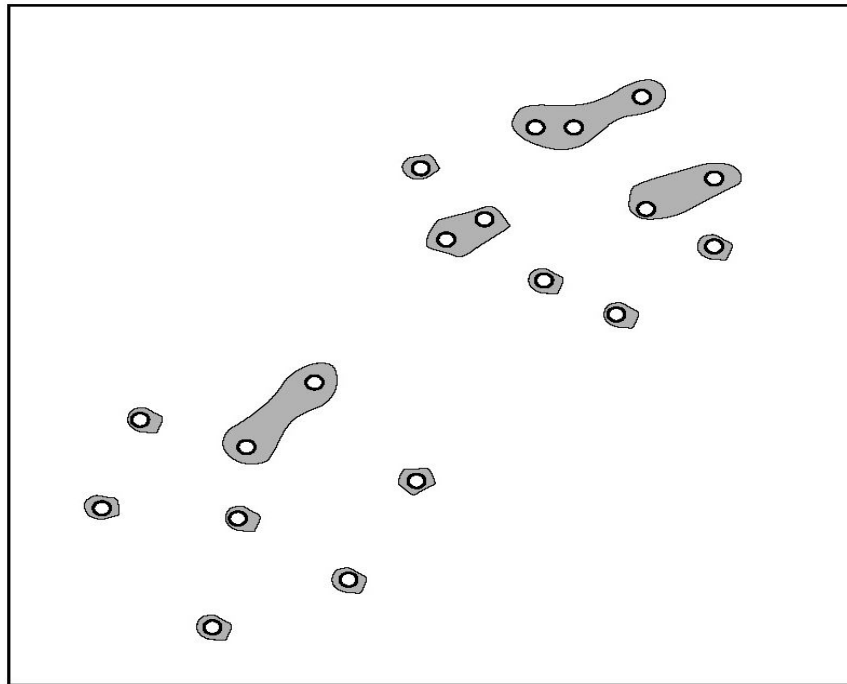


→ Distanz

# HAC: Example



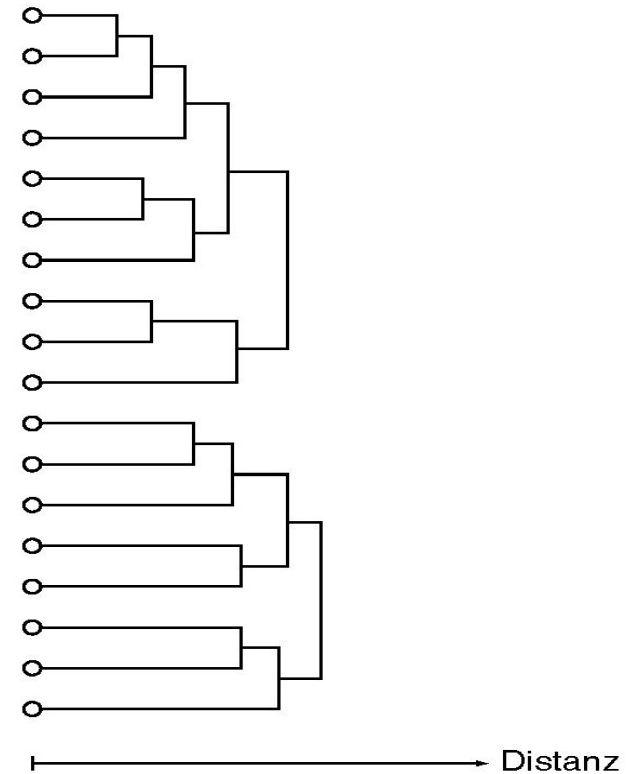
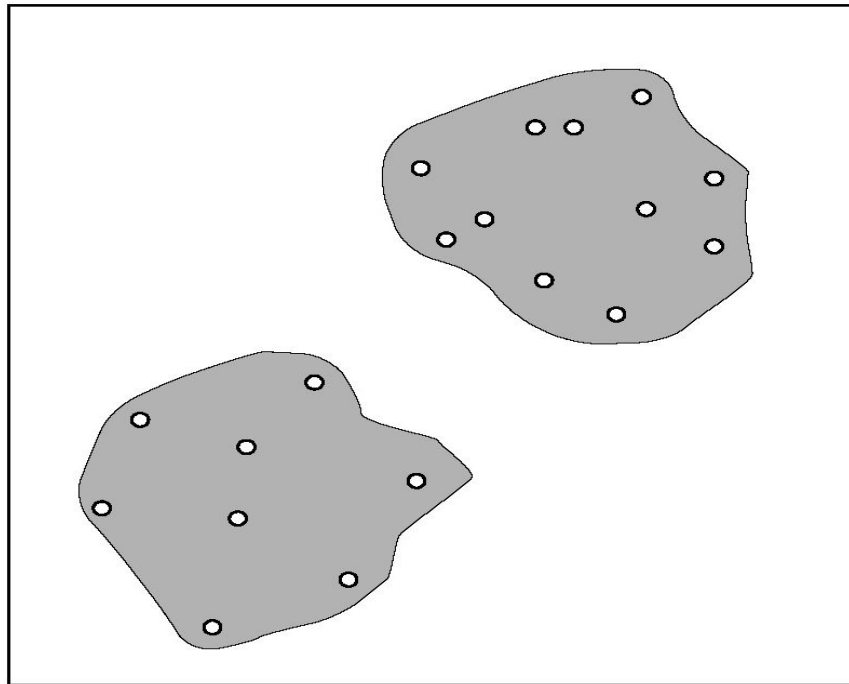
<http://www.uni-klu.ac.at>



# HAC: Example



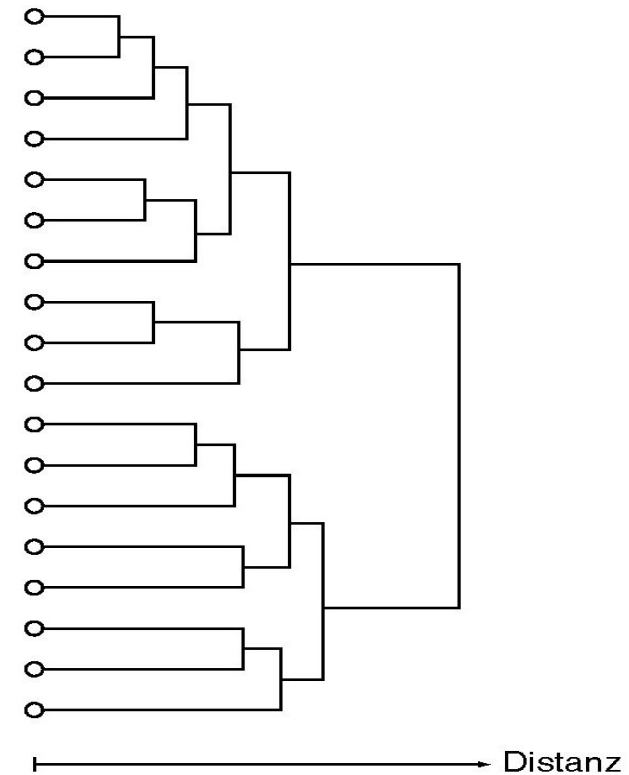
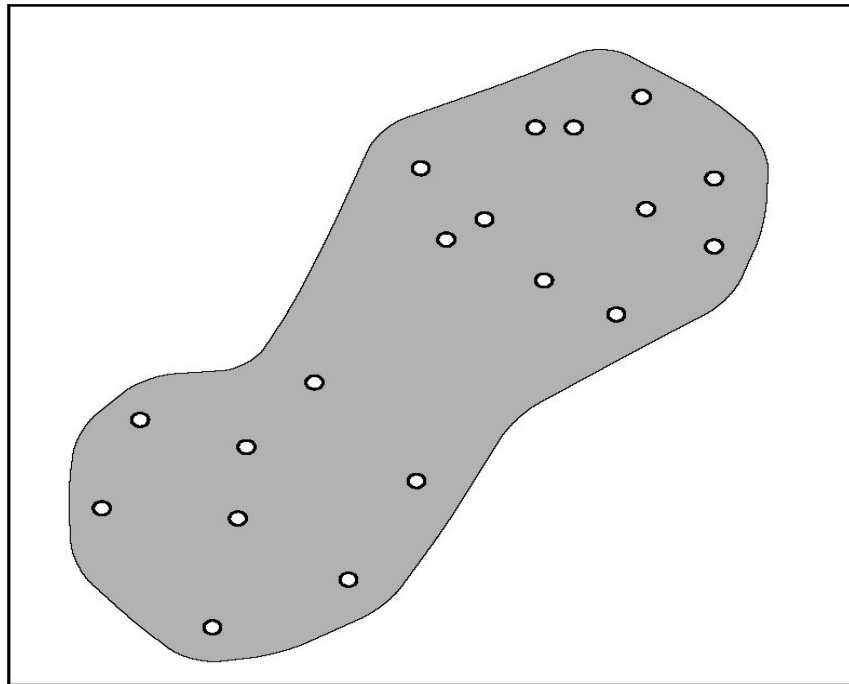
<http://www.uni-klu.ac.at>



# HAC: Example



<http://www.uni-klu.ac.at>



# Cluster Distance



<http://www.uni-klu.ac.at>

$$d_c(C, C') = \min_{\substack{u \in C \\ v \in C'}} d(u, v)$$

Single-Link  
(Nearest-Neighbor)

$$d_c(C, C') = \max_{\substack{u \in C \\ v \in C'}} d(u, v)$$

Complete-Link  
(Furthest-Neighbor)

$$d_c(C, C') = \frac{1}{|C| \cdot |C'|} \sum_{\substack{u \in C \\ v \in C'}} d(u, v)$$

(Group-)Average-Link

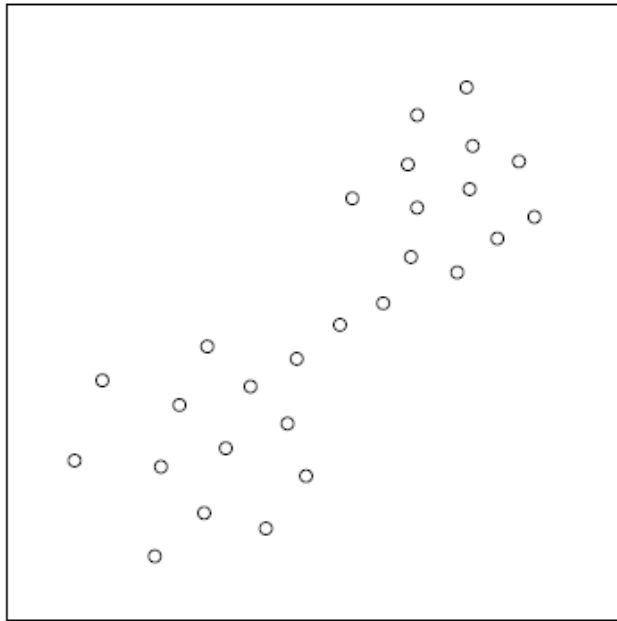
$$d_c(C, C') = \sqrt{\frac{2 \cdot |C| \cdot |C'|}{|C| + |C'|}} \cdot \|\bar{u} - \bar{v}\|$$

Ward (Varianz)

# Single Link Problem: Chaining



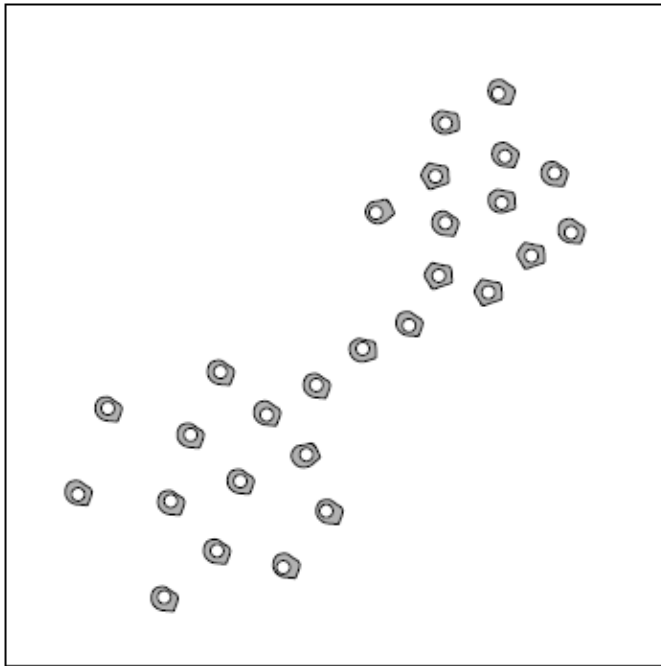
<http://www.uni-klu.ac.at>



# Single Link Problem: Chaining



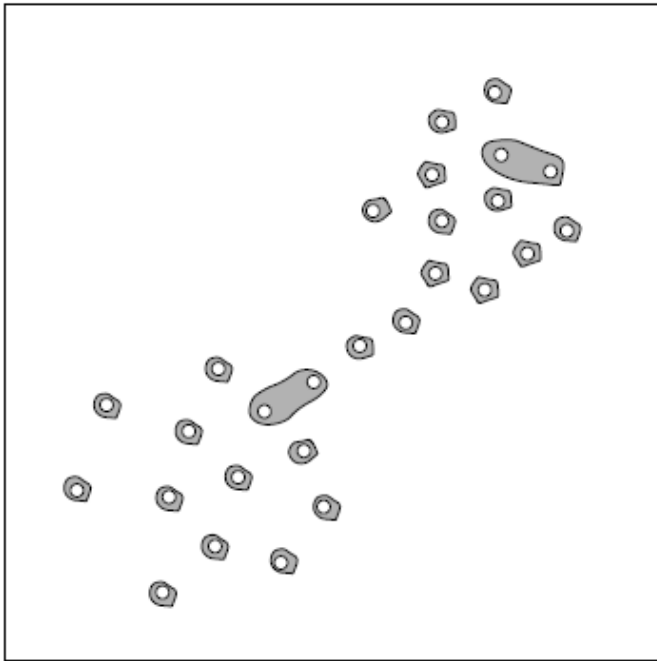
<http://www.uni-klu.ac.at>



# Single Link Problem: Chaining



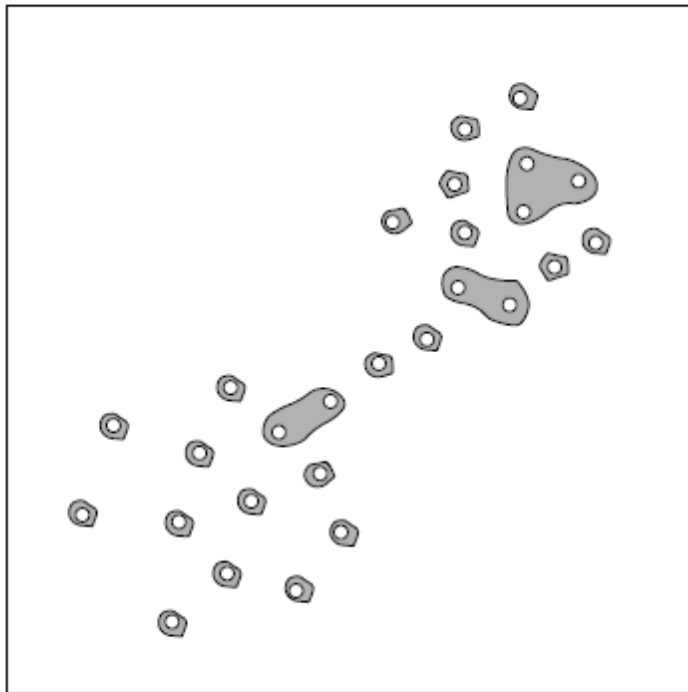
<http://www.uni-klu.ac.at>



# Single Link Problem: Chaining



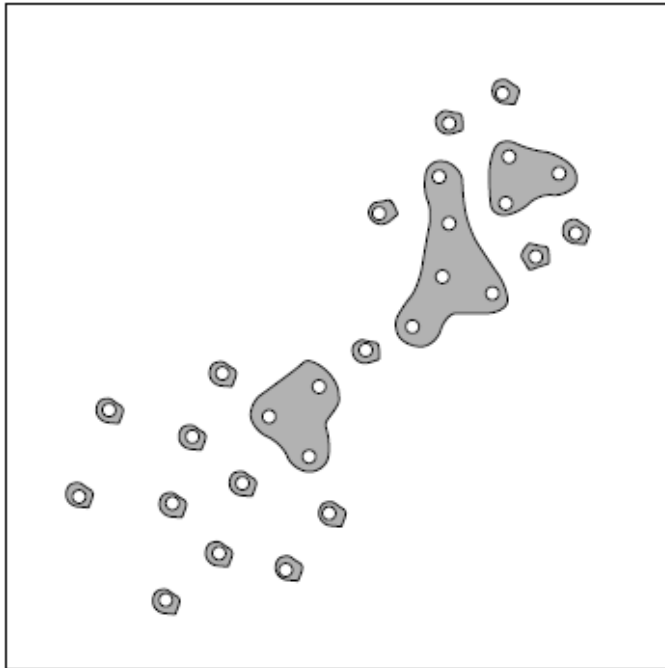
<http://www.uni-klu.ac.at>



# Single Link Problem: Chaining



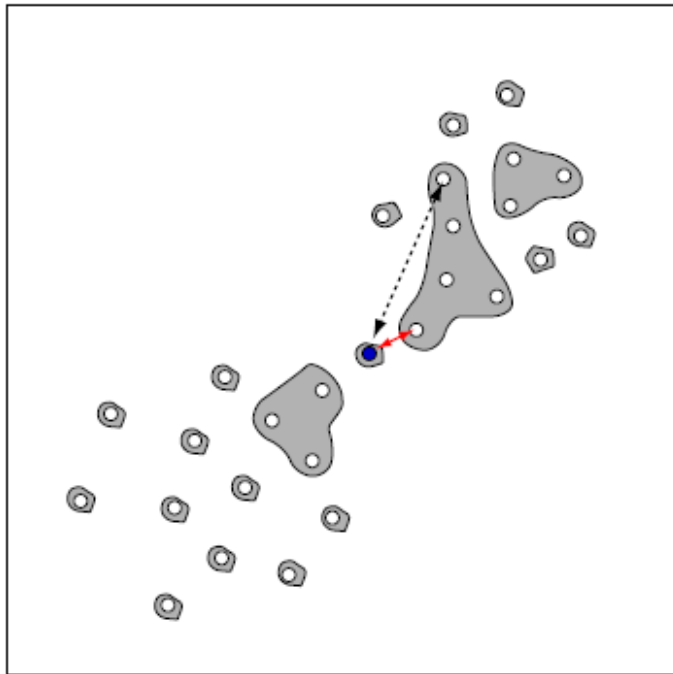
<http://www.uni-klu.ac.at>



# Single Link Problem: Chaining



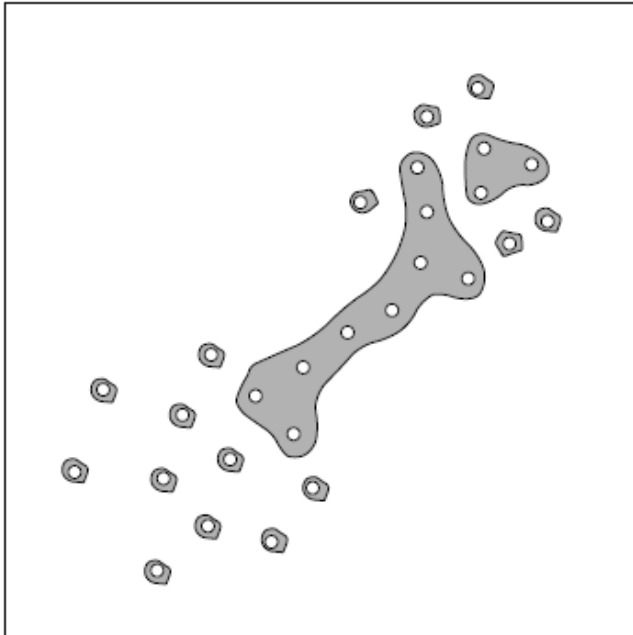
<http://www.uni-klu.ac.at>



# Single Link Problem: Chaining



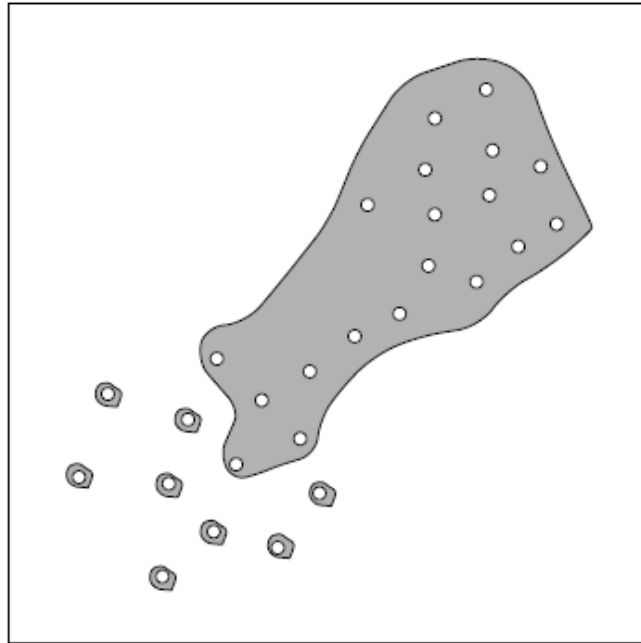
<http://www.uni-klu.ac.at>



# Single Link Problem: Chaining



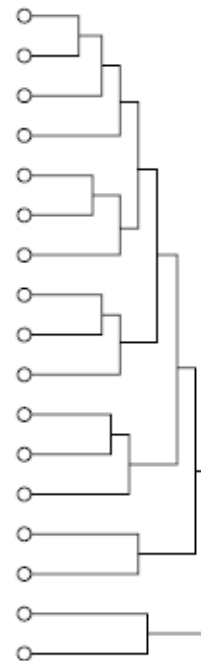
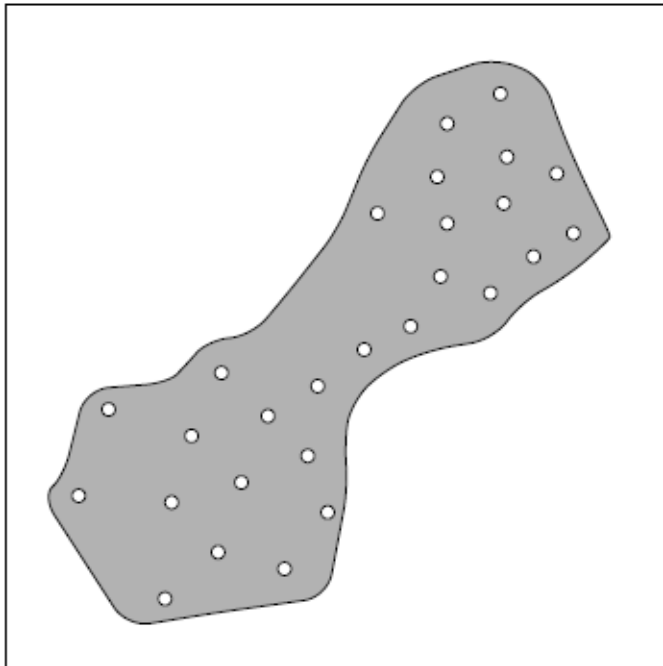
<http://www.uni-klu.ac.at>



# Single Link Problem: Chaining



<http://www.uni-klu.ac.at>

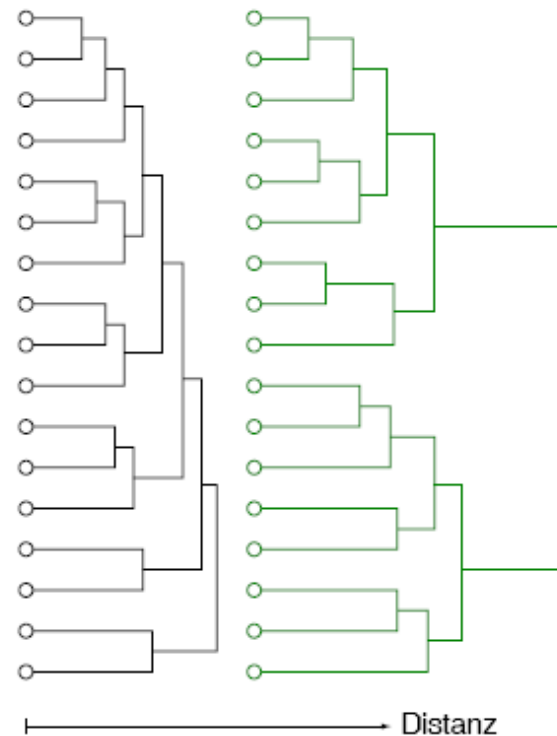
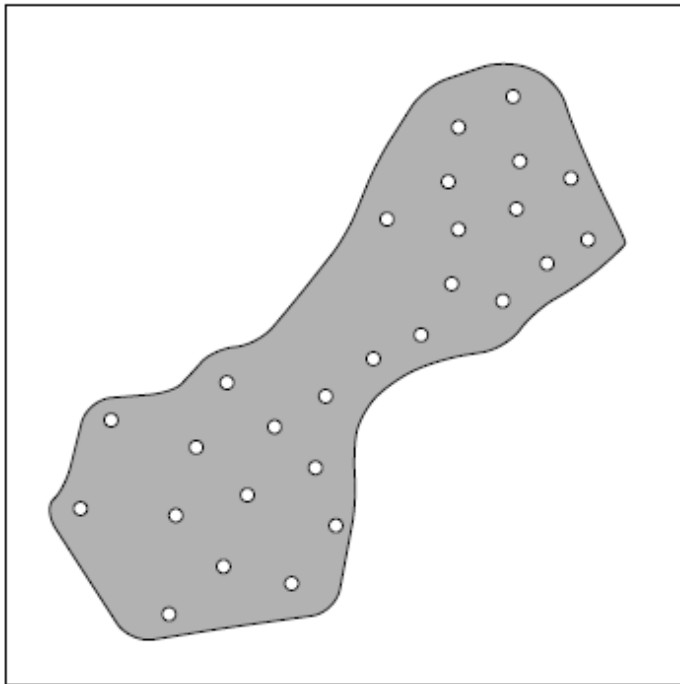


→ Distanz

# Single Link Problem: Chaining



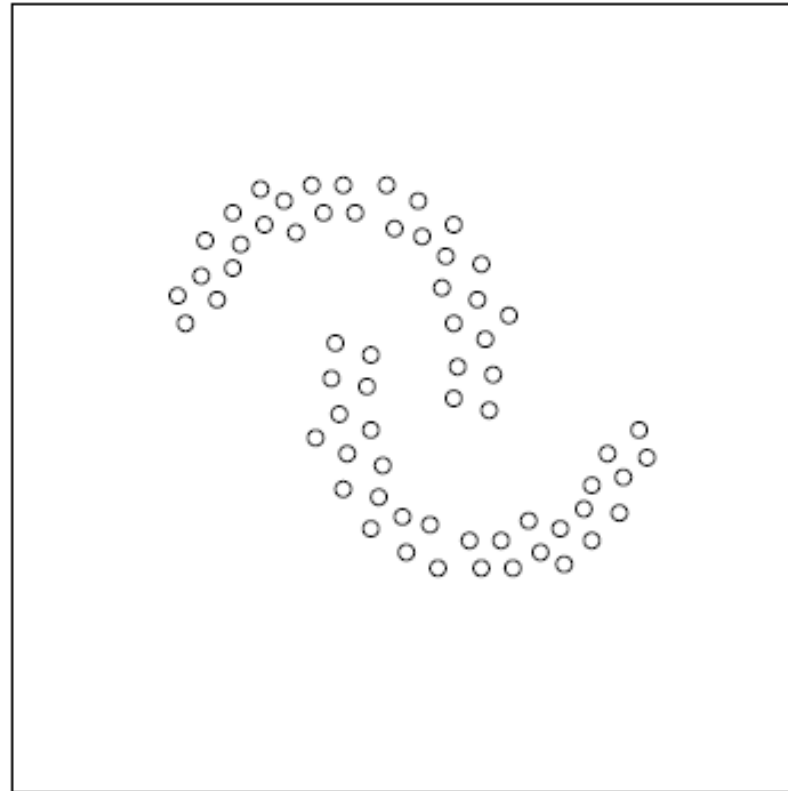
<http://www.uni-klu.ac.at>



# Complete Link Problem: Overlap



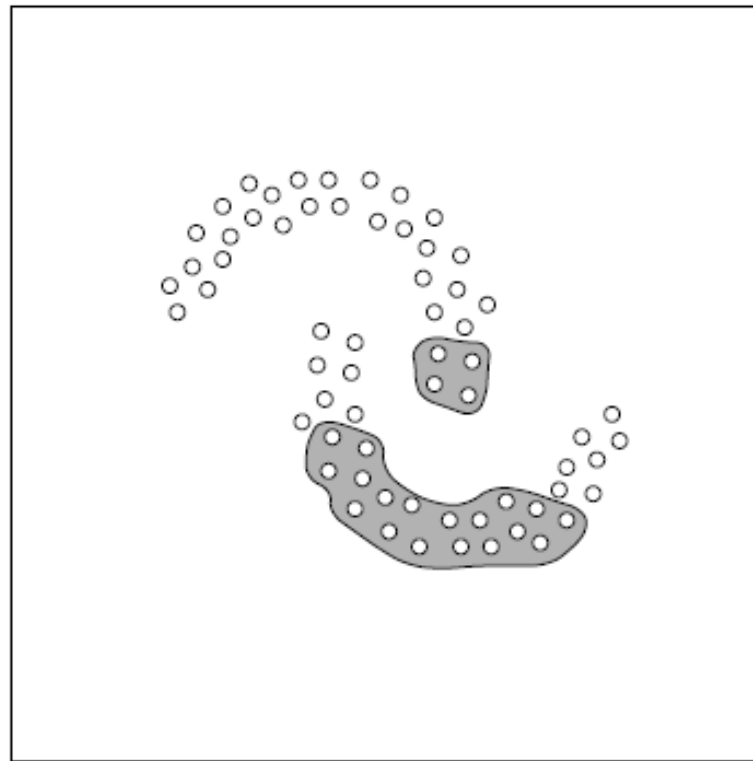
<http://www.uni-klu.ac.at>



# Complete Link Problem: Overlap



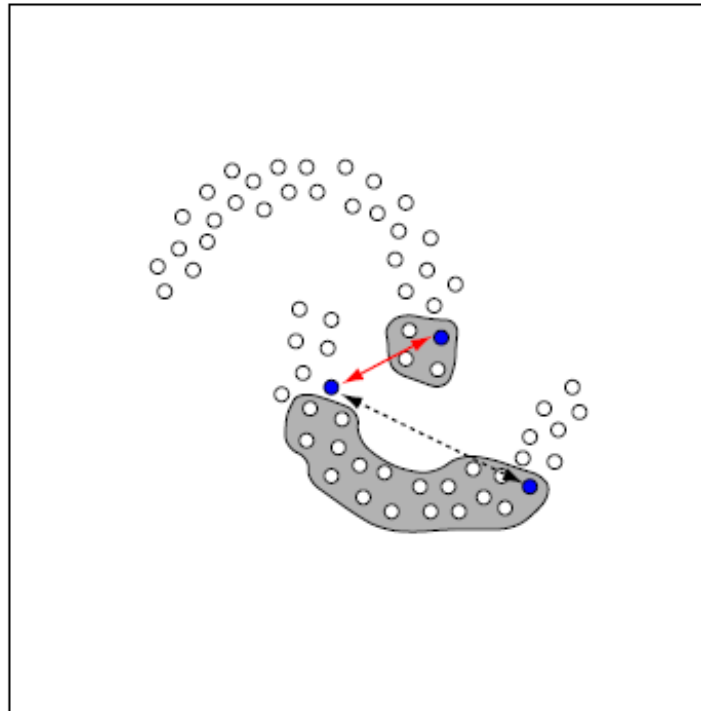
<http://www.uni-klu.ac.at>



# Complete Link Problem: Overlap



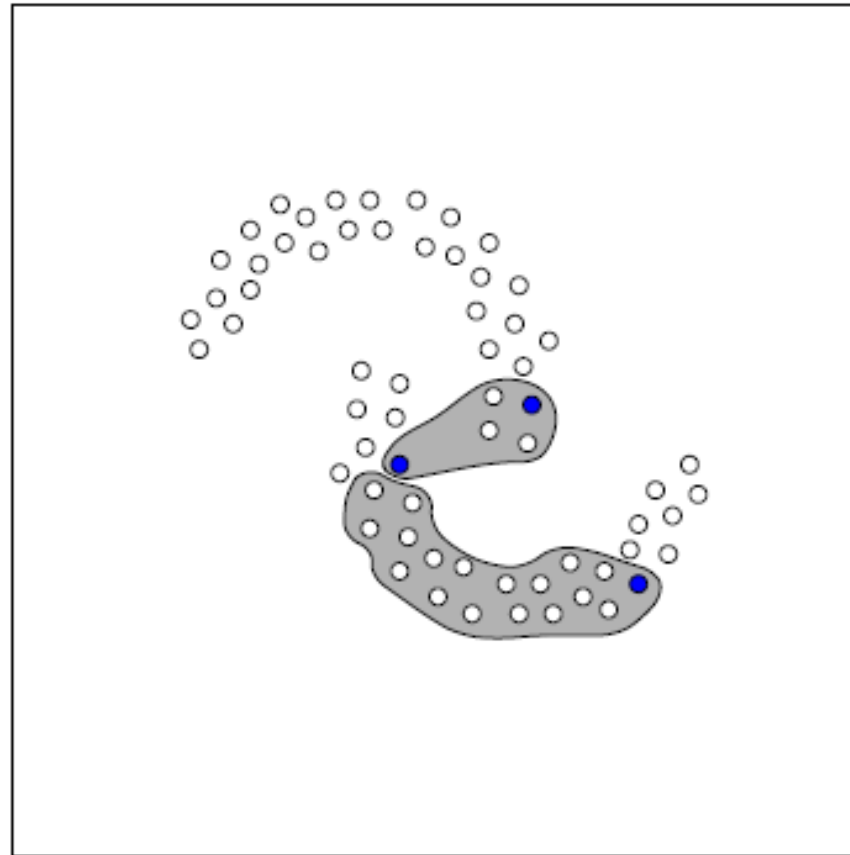
<http://www.uni-klu.ac.at>



# Complete Link Problem: Overlap



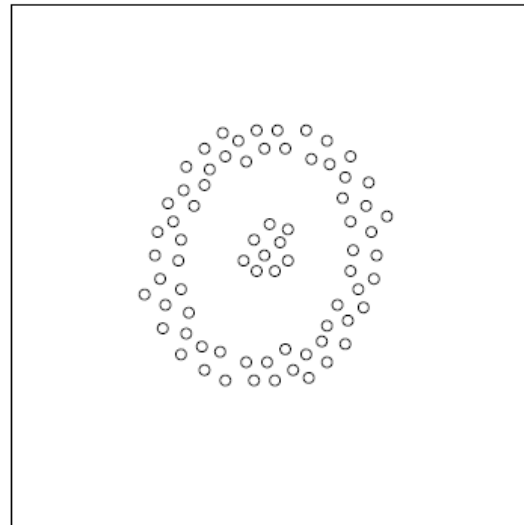
<http://www.uni-klu.ac.at>



# Complete Link Problem: Overlap



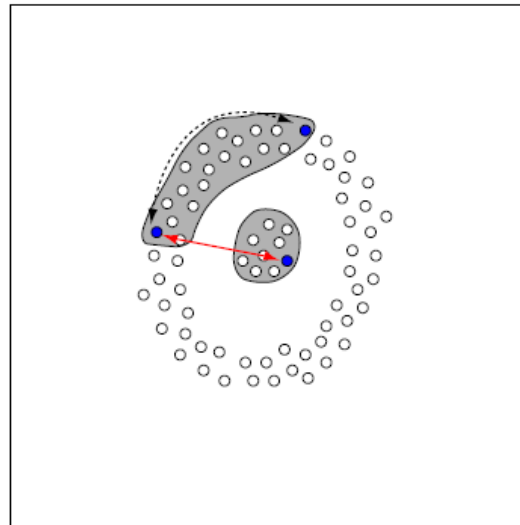
<http://www.uni-klu.ac.at>



# Complete Link Problem: Overlap



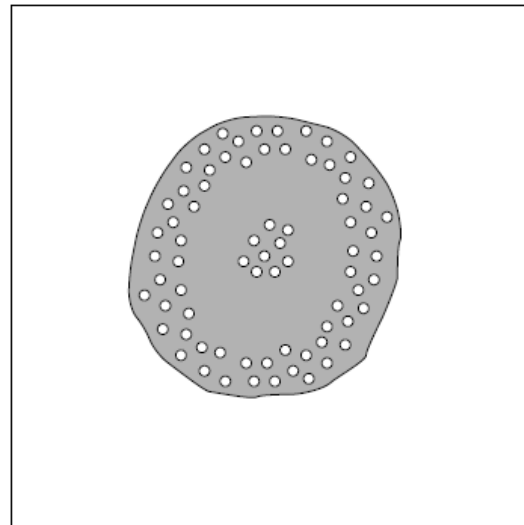
<http://www.uni-klu.ac.at>



# Complete Link Problem: Overlap



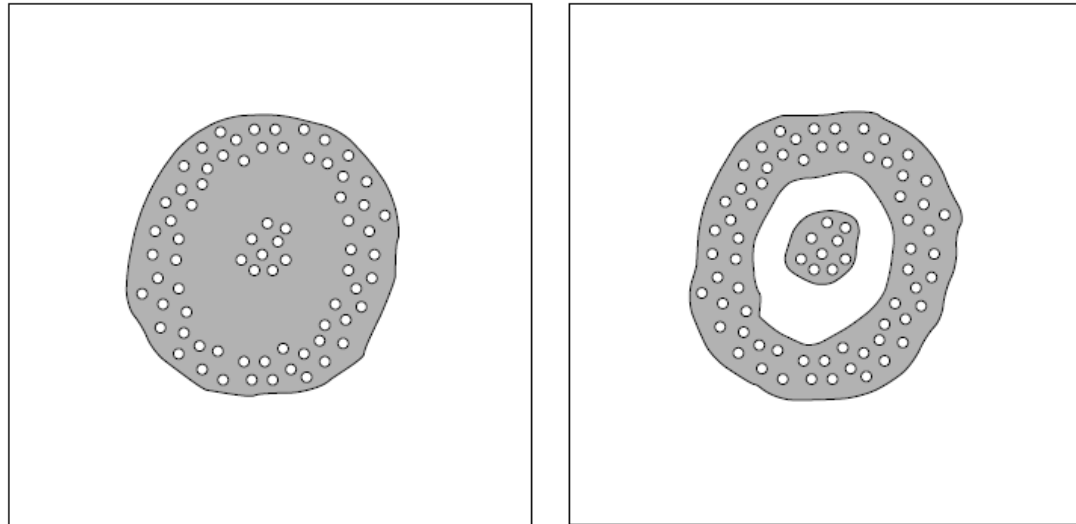
<http://www.uni-klu.ac.at>



# Complete Link Problem: Overlap



<http://www.uni-klu.ac.at>



# Hierarchical Clustering: Comparison



<http://www.uni-klu.ac.at>

|                   | Single Link      | Complete Link   | Average Link   | Ward             |
|-------------------|------------------|-----------------|----------------|------------------|
| # clusters        | <i>small</i>     | <i>high</i>     | <i>medium</i>  | <i>medium</i>    |
| cluster type      | <i>stretched</i> | <i>small</i>    | <i>compact</i> | <i>spherical</i> |
| chaining tendency | <i>high</i>      | <i>low</i>      | <i>low</i>     | <i>low</i>       |
| outlier detection | <i>high</i>      | <i>very low</i> | <i>low</i>     | <i>low</i>       |

# Partitional Clustering



<http://www.uni-klu.ac.at>

- 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



<http://www.uni-klu.ac.at>

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

Output:  $r_1, \dots, r_k$ . Cluster representatives.

```
1.  $t = 0$ 
2. FOR  $i = 1$  to  $k$  DO  $r_i(t) = \text{choose}(V)$  // init representatives
3. REPEAT
4.   FOR  $i = 1$  to  $k$  DO  $C_i = \emptyset$ 
5.   FOREACH  $v \in V$  DO // find nearest representative (cluster)
6.      $x = \underset{i: i \in \{1, \dots, k\}}{\operatorname{argmin}} d(r_i(t), v), C_x = C_x \cup \{v\}$ 
7.   ENDDO
8.   FOR  $i = 1$  to  $k$  DO  $r_i(t) = \text{minimize}(e, C_i)$  // update
9. UNTIL  $(\forall r_i : d(r_i(t), r_i(t-1)) < \varepsilon \vee t > t_{\max})$ 
10. RETURN  $(\{r_1(t), \dots, r_k(t)\})$ 
```

# Iterative Clustering Algorithm



<http://www.uni-klu.ac.at>

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



<http://www.uni-klu.ac.at>

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

# Iterative Clustering Algorithm



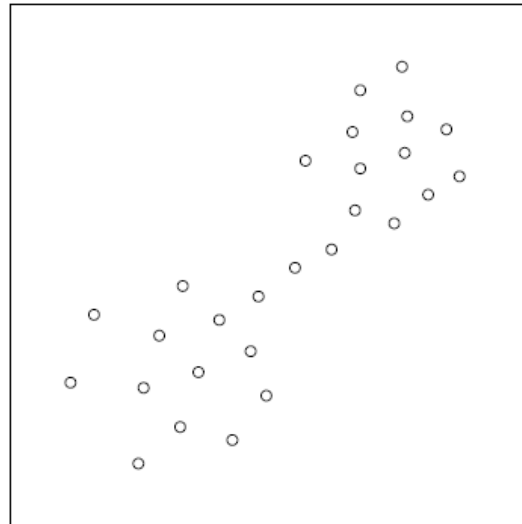
<http://www.uni-klu.ac.at>

- 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



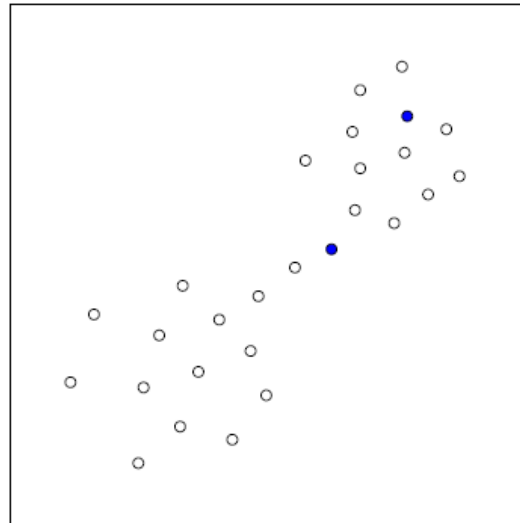
<http://www.uni-klu.ac.at>



# K-Means Example



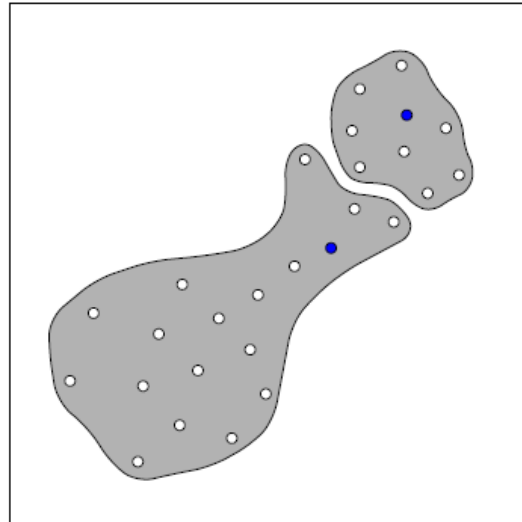
<http://www.uni-klu.ac.at>



# K-Means Example



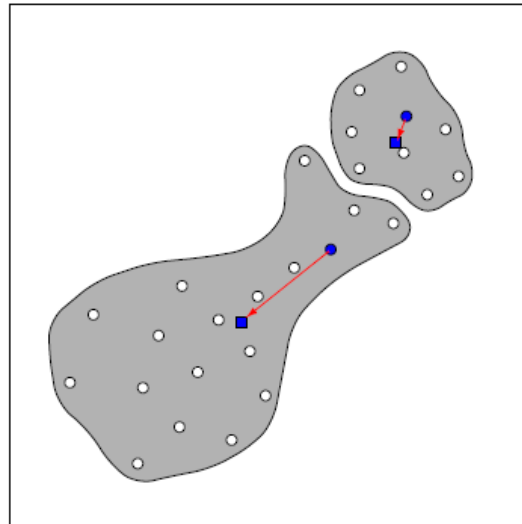
<http://www.uni-klu.ac.at>



# K-Means Example



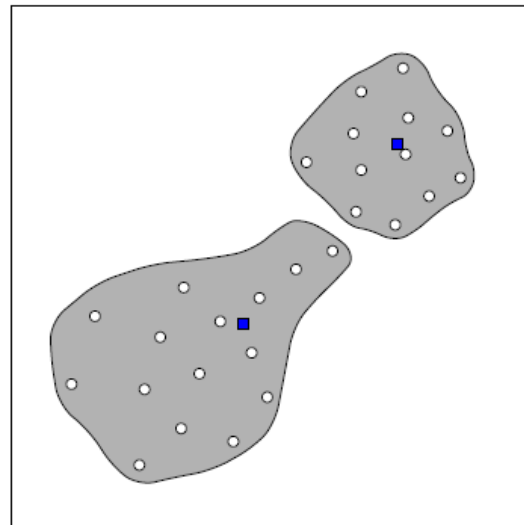
<http://www.uni-klu.ac.at>



# K-Means Example



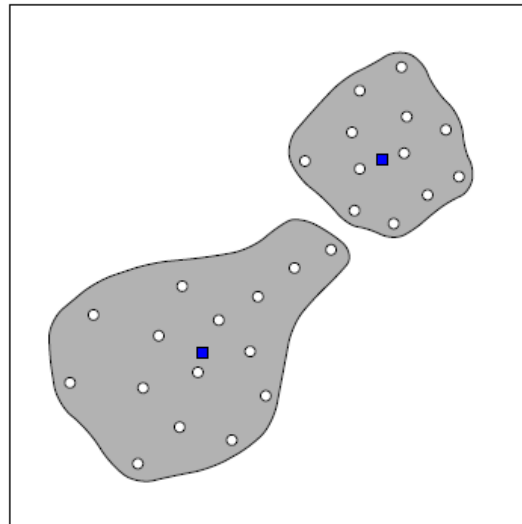
<http://www.uni-klu.ac.at>



# K-Means Example



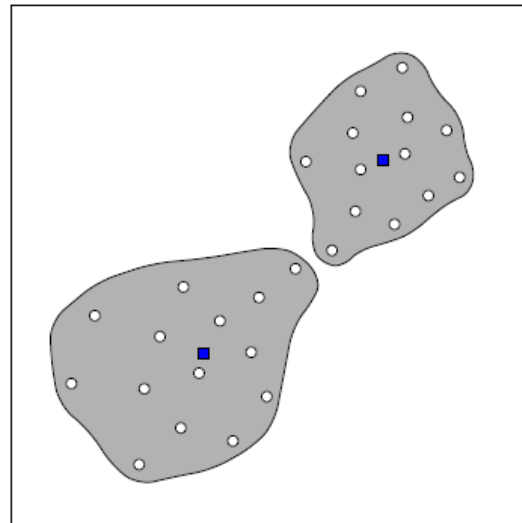
<http://www.uni-klu.ac.at>



# K-Means Example



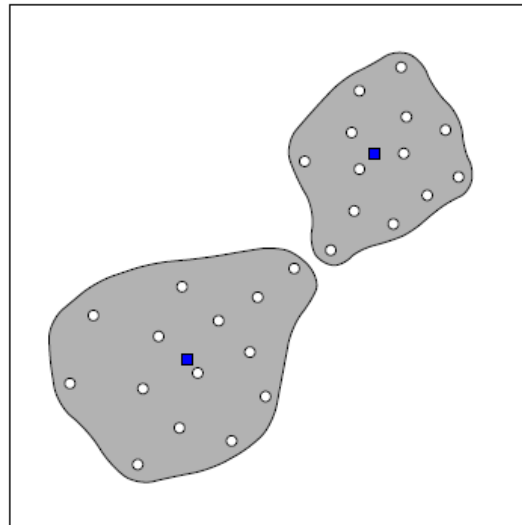
<http://www.uni-klu.ac.at>



# K-Means Example



<http://www.uni-klu.ac.at>



# Cluster Center



<http://www.uni-klu.ac.at>

$$e(\mathcal{C}) = \sum_{i=1}^k \sum_{v \in C_i} (v - r_i)^2$$

$$r_i = \bar{v}_{C_i}$$

Centroid-  
Berechnung  
( $k$ -Means)

$$e(\mathcal{C}) = \sum_{i=1}^k \sum_{v \in C_i} |v - r_i|$$

$$r_i \in C_i$$

Medoid-  
Berechnung  
( $k$ -Medoid)

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

$$r_i \in C_i$$

$k$ -Center

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

Fuzzy-  
 $k$ -Means

# Method Comparison



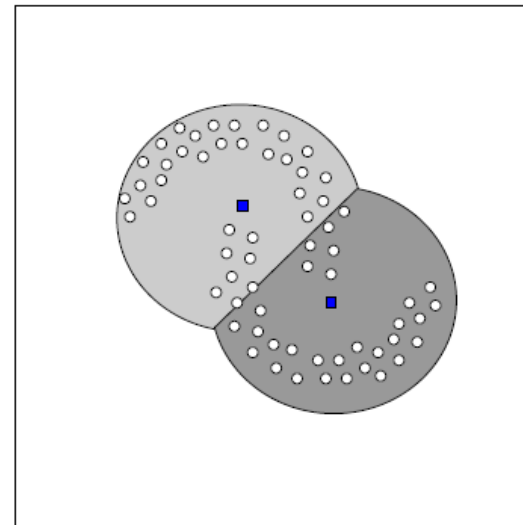
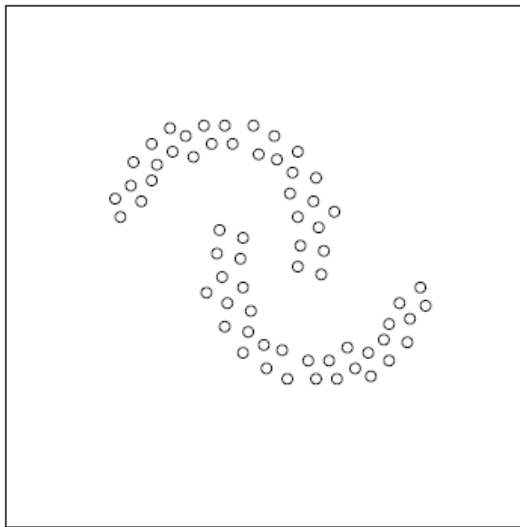
<http://www.uni-klu.ac.at>

- 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



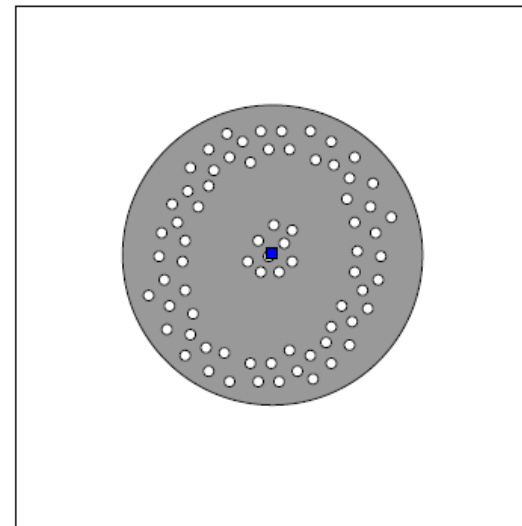
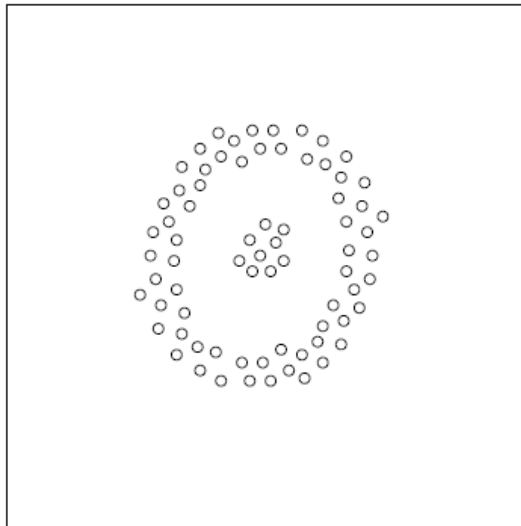
<http://www.uni-klu.ac.at>



# K-Means Problems



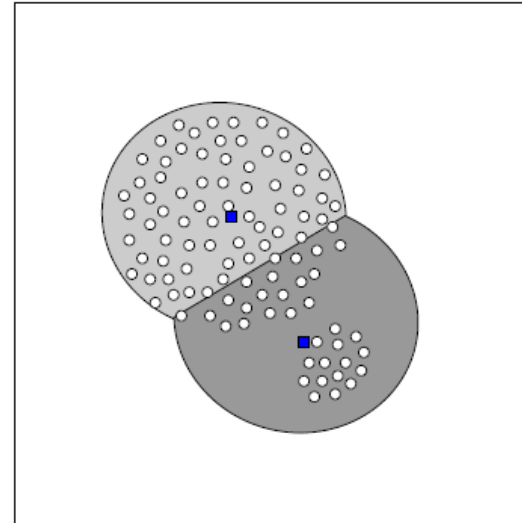
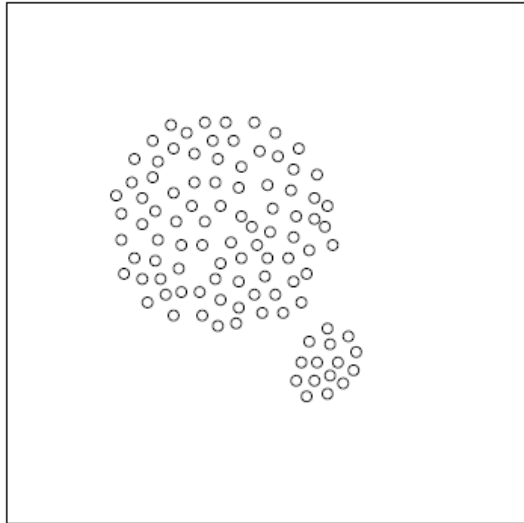
<http://www.uni-klu.ac.at>



# K-Means Problems



<http://www.uni-klu.ac.at>



# Thank you ...



<http://www.uni-klu.ac.at>

... for your attention