VK Multimedia Information Systems

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

Dienstags, 16.oo Uhr s.t., E.1.42

Selection of Project B

- Find a project that
  - … is personally interesting for you
  - … you can finish in time (40-60h)
  - … has something to do with the course
- Opportunities
  - Try before buy: Thesis, etc.
  - Get some work done (for you, employer, …)
  - Be creative
  - Contribute to open source

Projects 1

- SURF/SIFT/GLOH – Primus, Garnik, Katzian
- Video Summary – Mueller, Guggenberger
- Bildsegmentierung – Korak, Lettmayer, Schager

Projects 2

- Salient points / regions – Muenzer, Pairitsch
- “Scalable Recognition with a Vocabulary Tree” – Wang
- 2-D Barcodes: Druck, Scan & Parsing

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

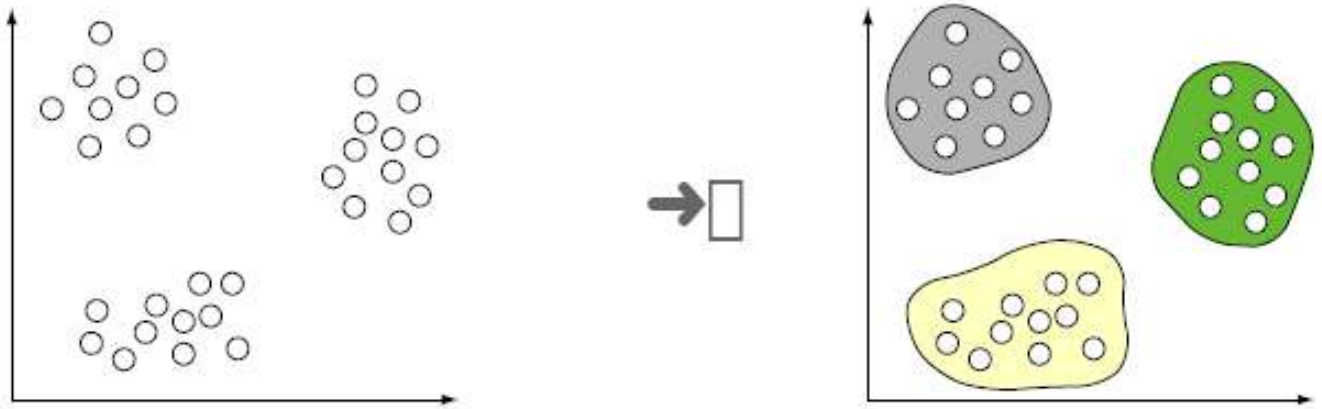
Indexing of multimedia documents

- 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

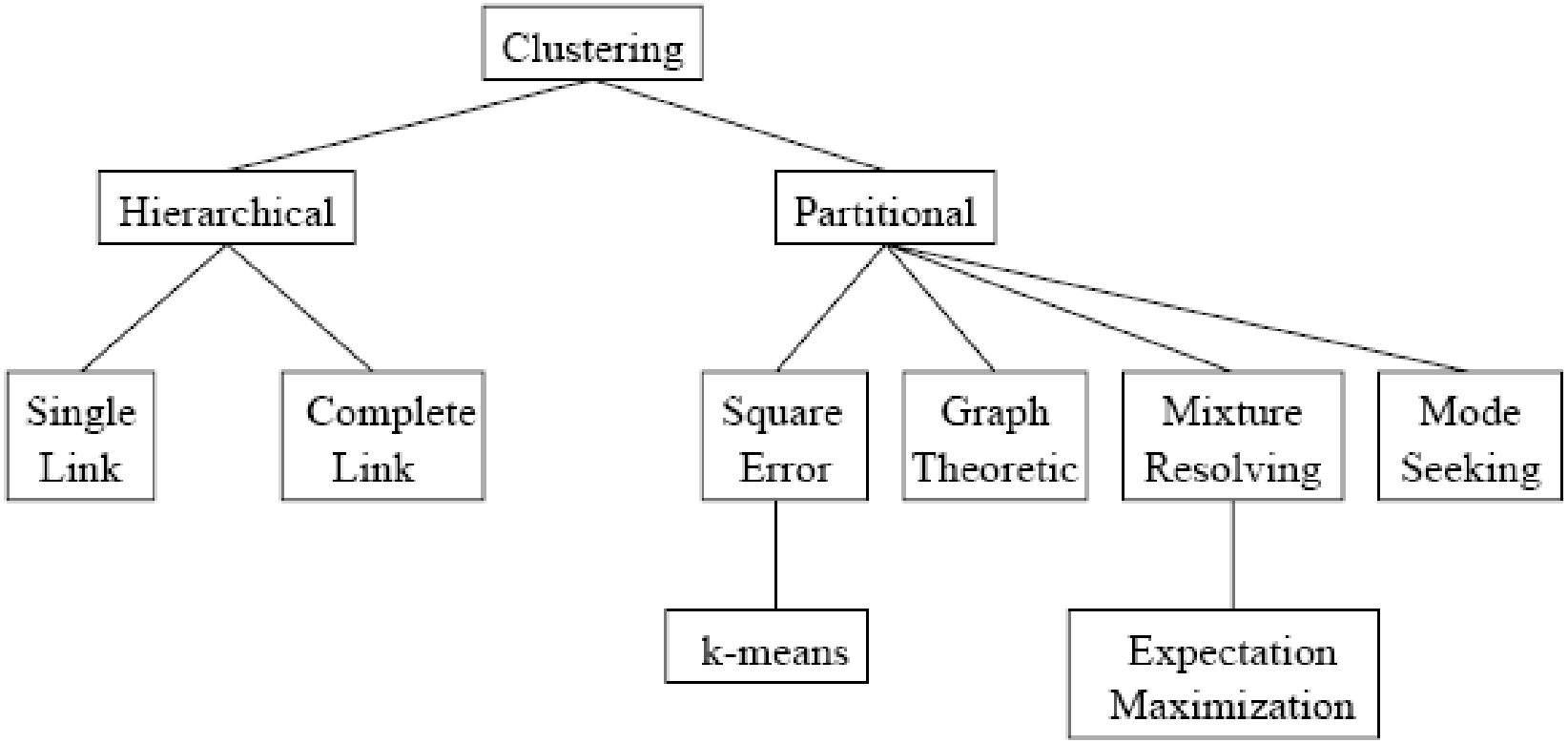
- For 2 dimensions:



Clustering: Definition

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

Clustering Techniques



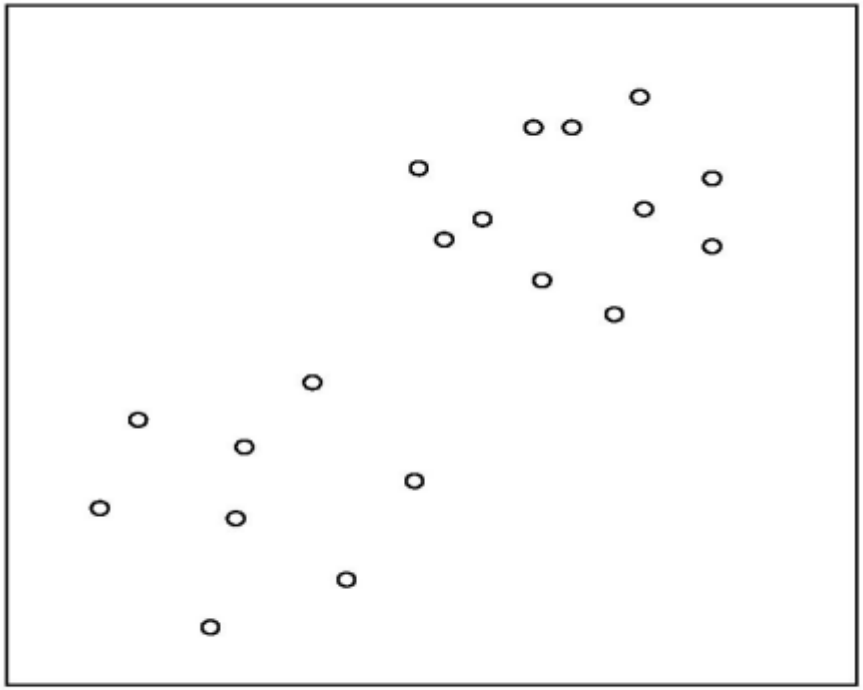
Hierarchical Clustering

Input: $G = \langle V, E, w \rangle$. Weighted graph.
$d_{\mathcal{C}}$. Distance measure between two clusters.

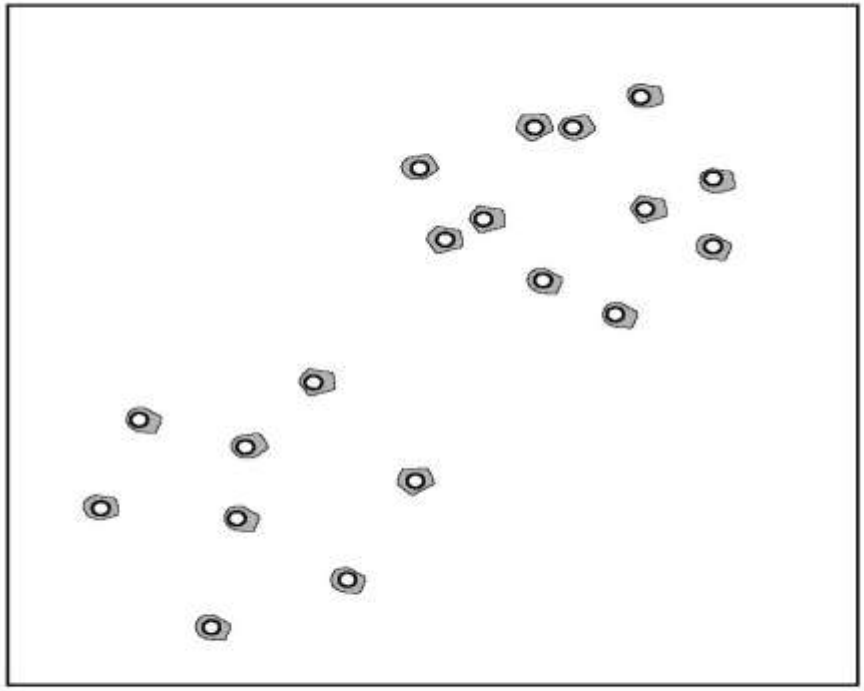
Output: $T = \langle V_T, E_T \rangle$. Cluster hierarchy or dendrogram.

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

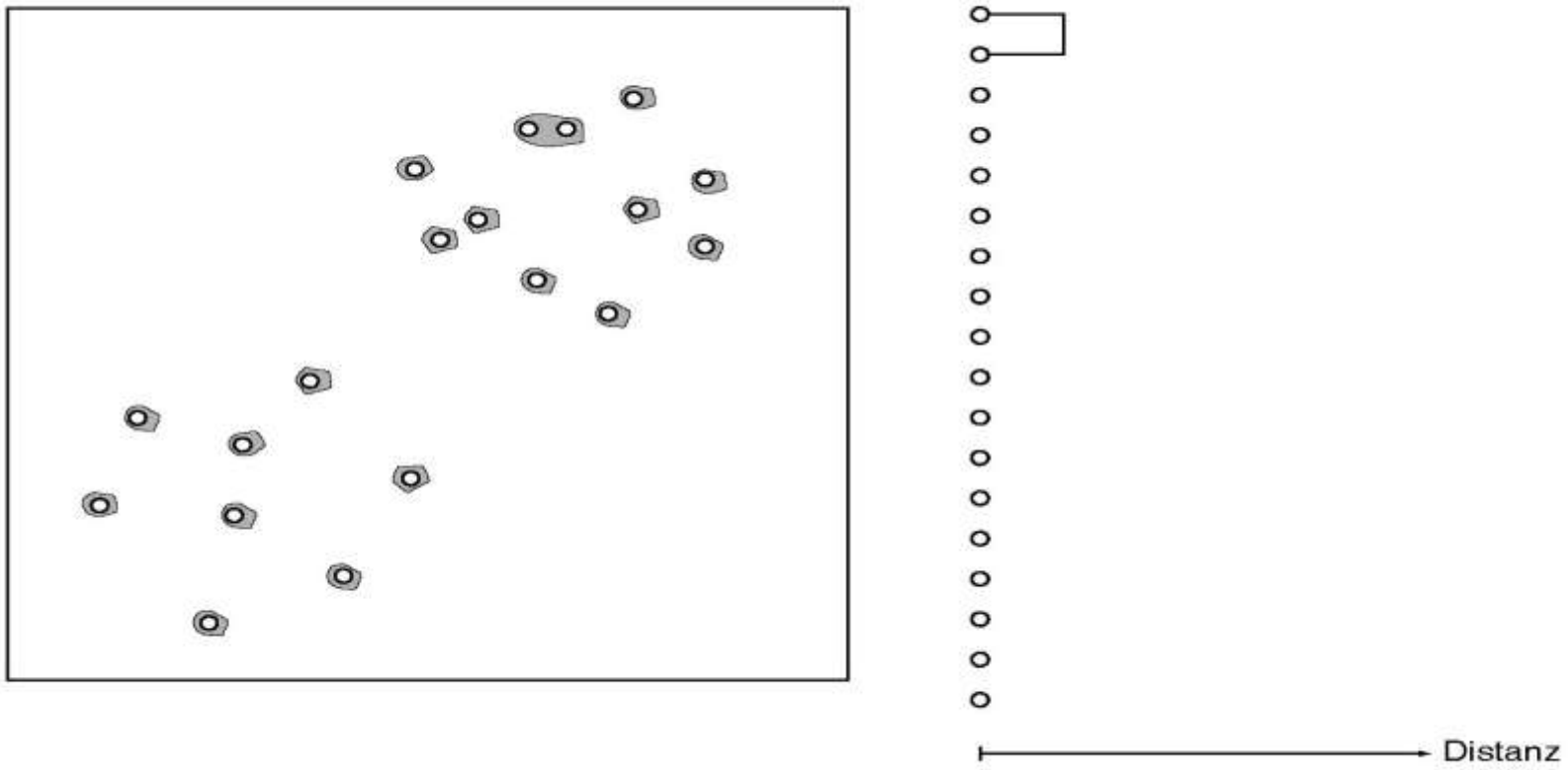
HAC: Example



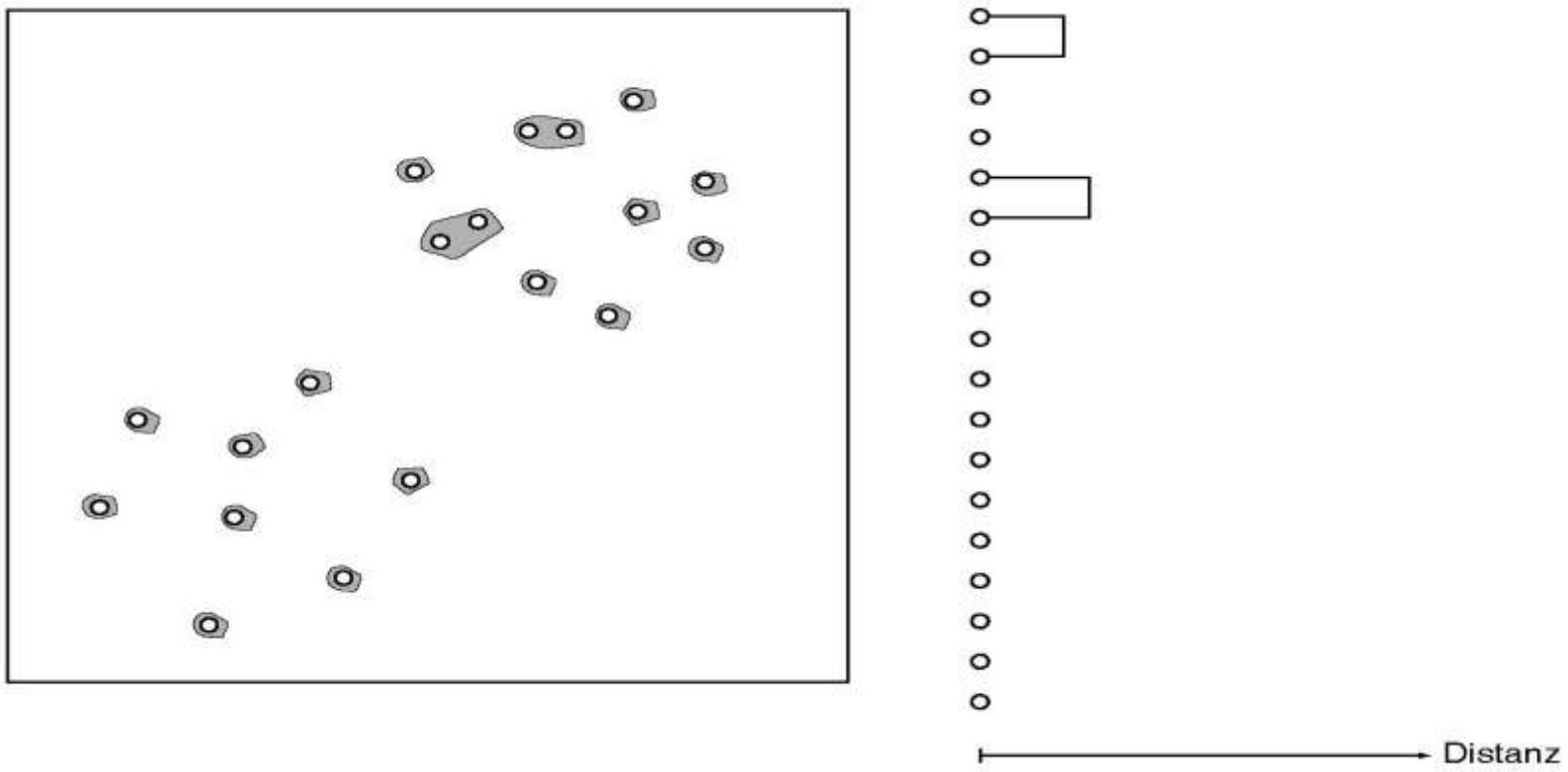
HAC: Example



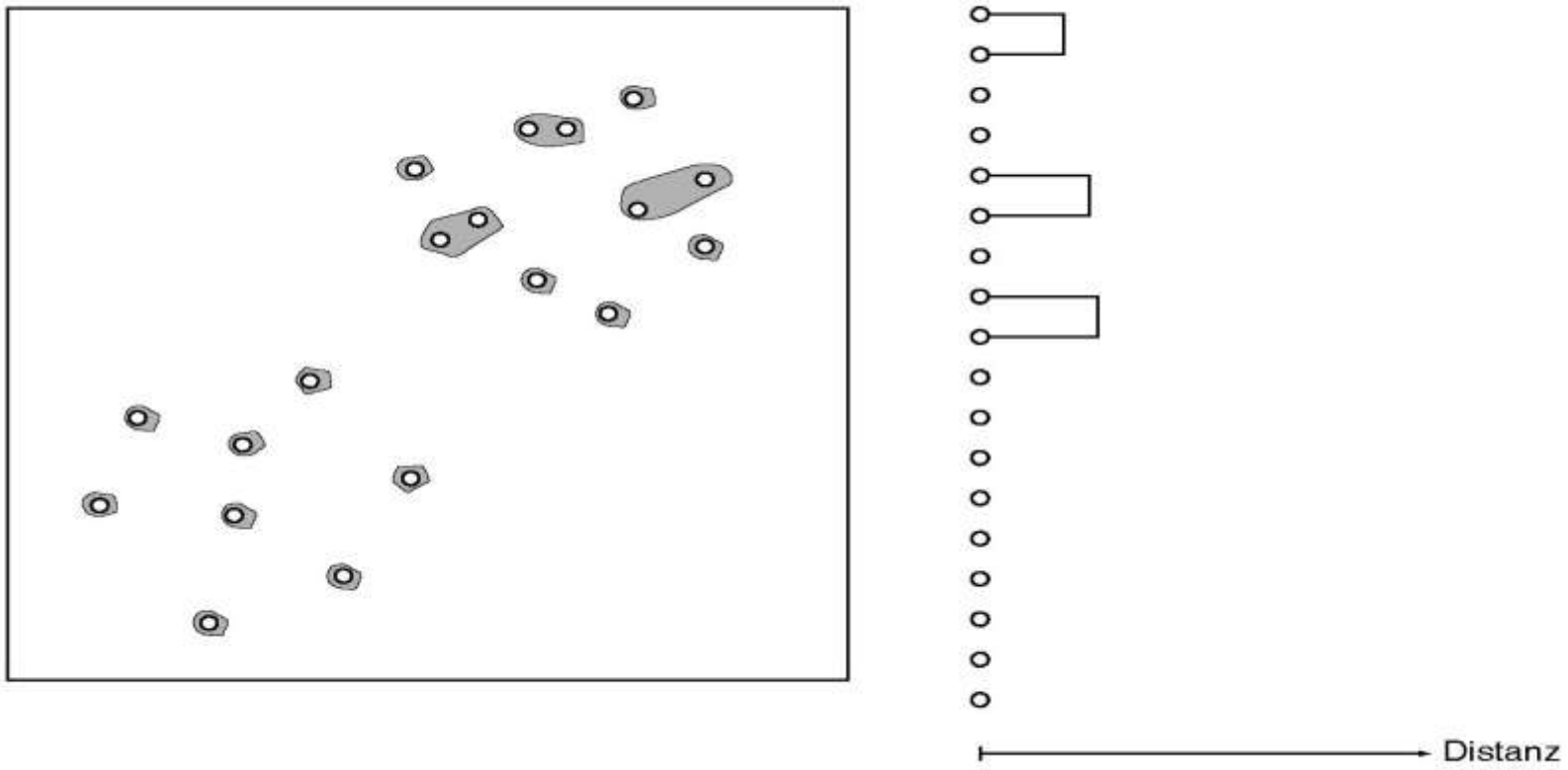
HAC: Example



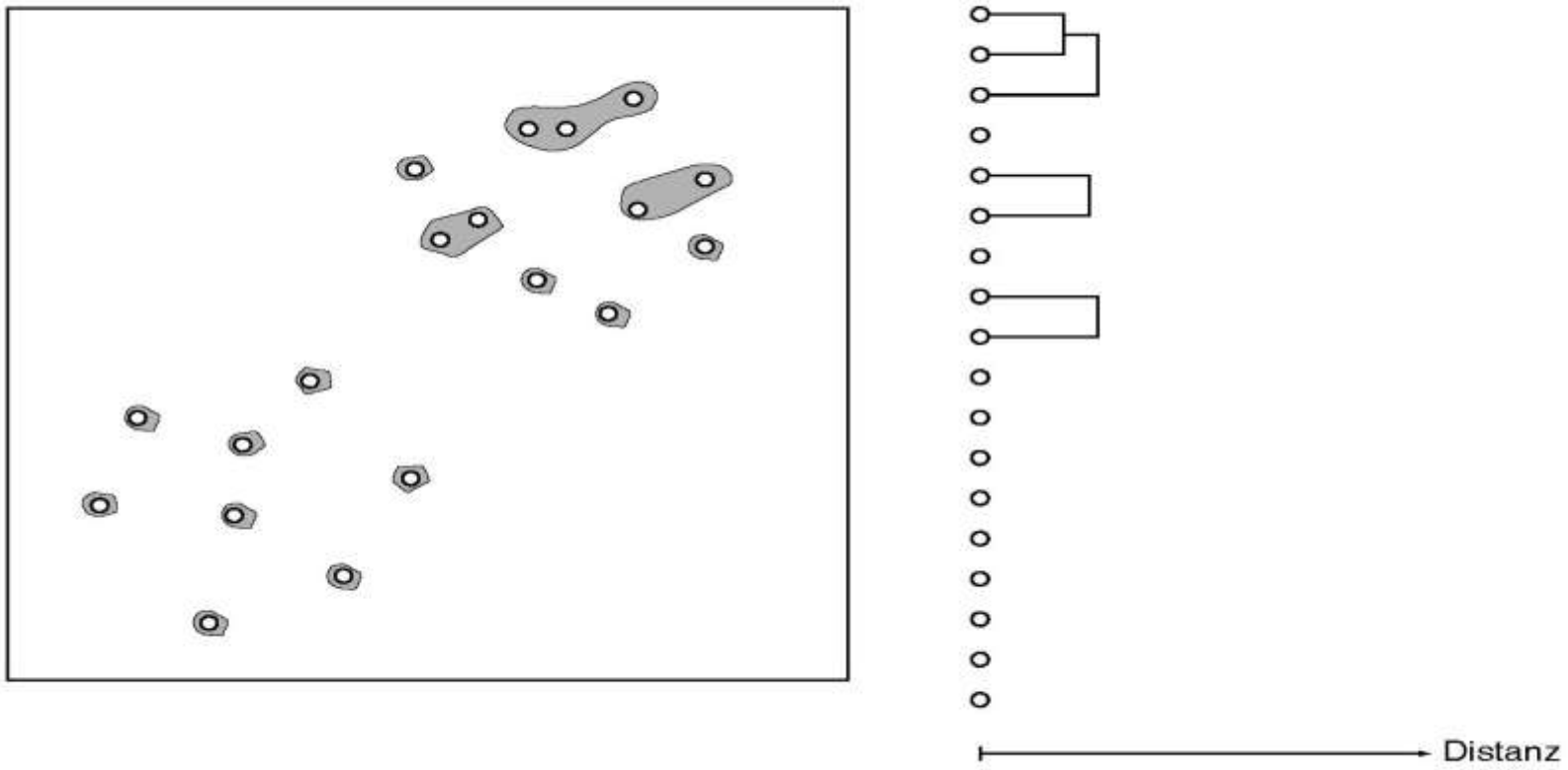
HAC: Example



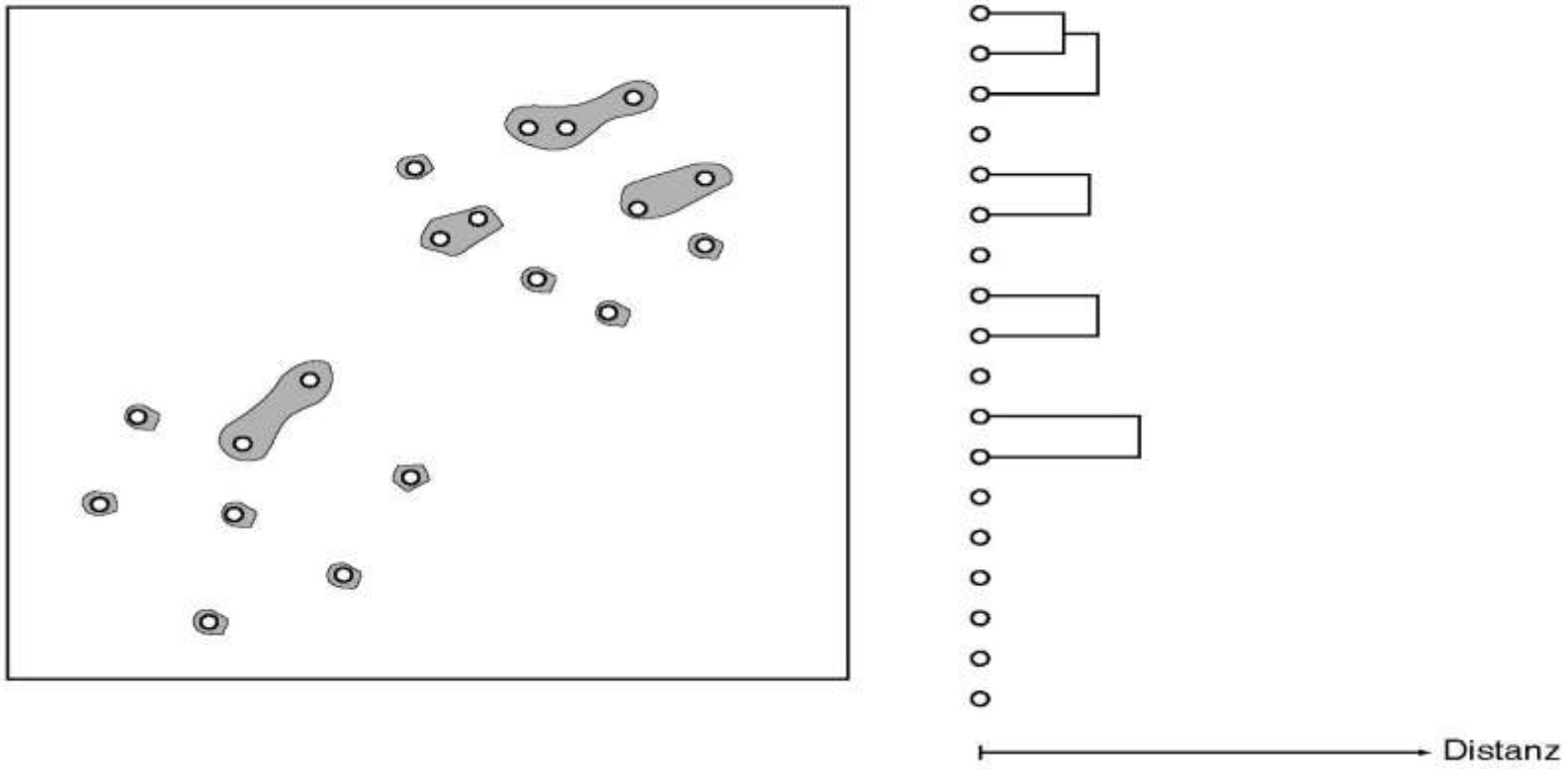
HAC: Example



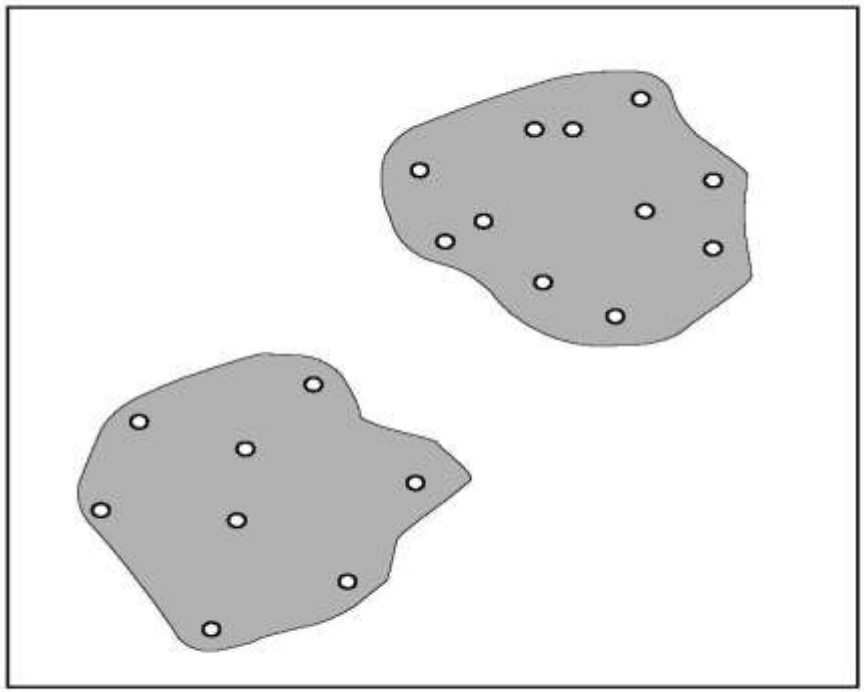
HAC: Example

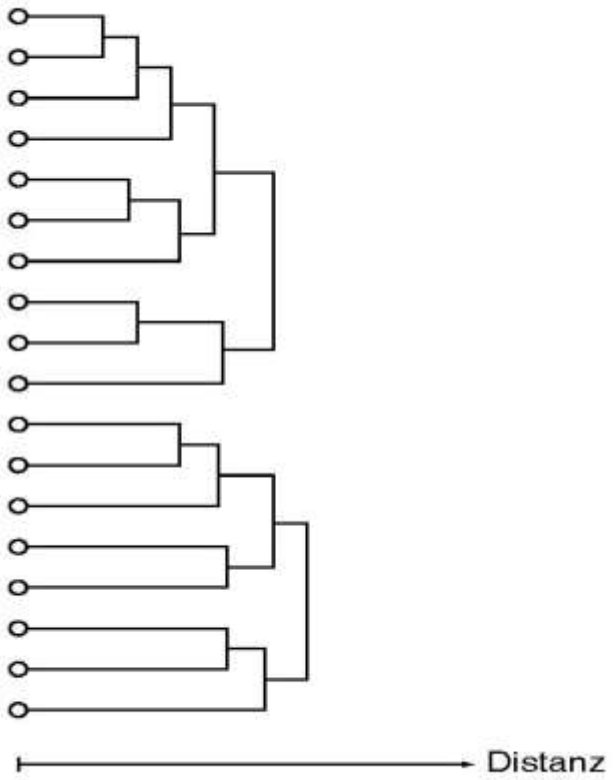


HAC: Example

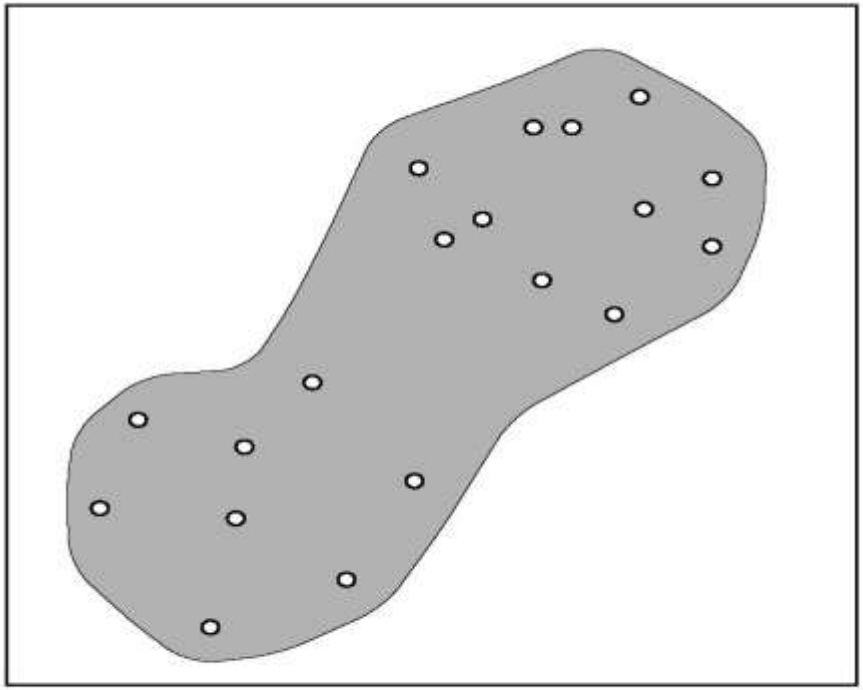


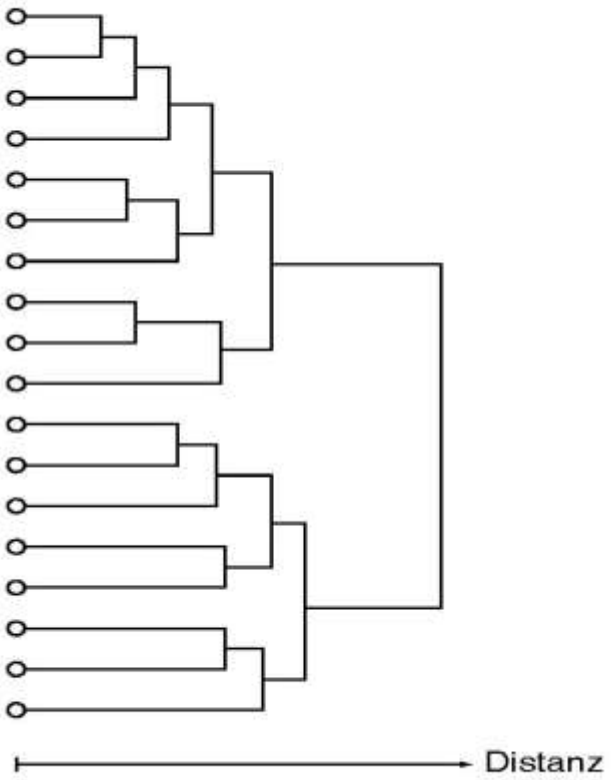
HAC: Example





HAC: Example





Cluster Distance

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

Single-Link
(Nearest-Neighbor)

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

Complete-Link
(Furthest-Neighbor)

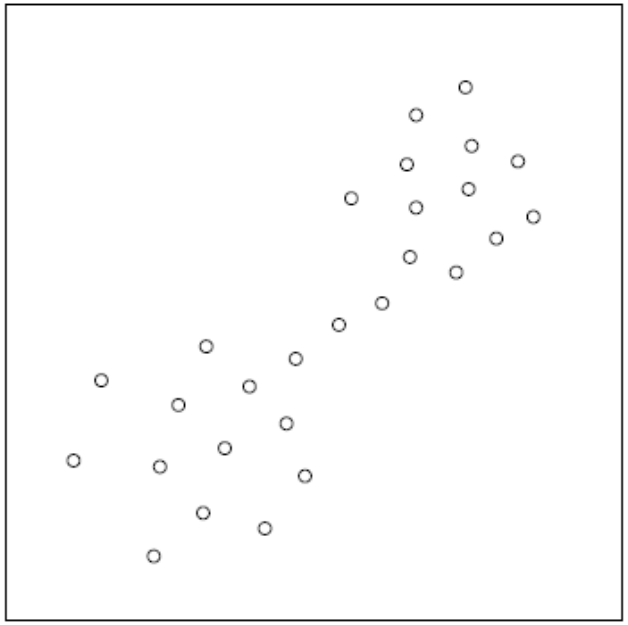
$$d_{\mathcal{C}}(C, C') = \frac{1}{|C| \cdot |C'|} \sum_{\substack{u \in C \\ v \in C'}} d(u, v)$$

(Group-)Average-Link

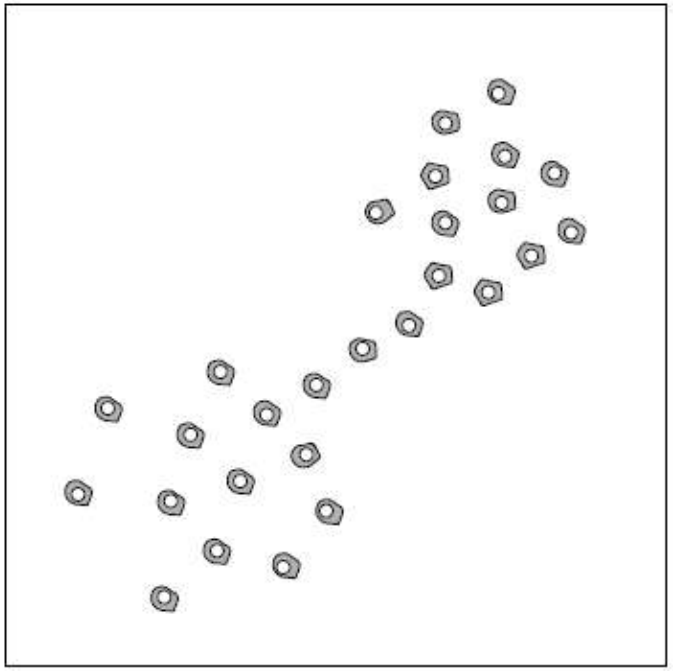
$$d_{\mathcal{C}}(C, C') = \sqrt{\frac{2 \cdot |C| \cdot |C'|}{|C| + |C'|}} \cdot ||\bar{u} - \bar{v}||$$

Ward (Varianz)

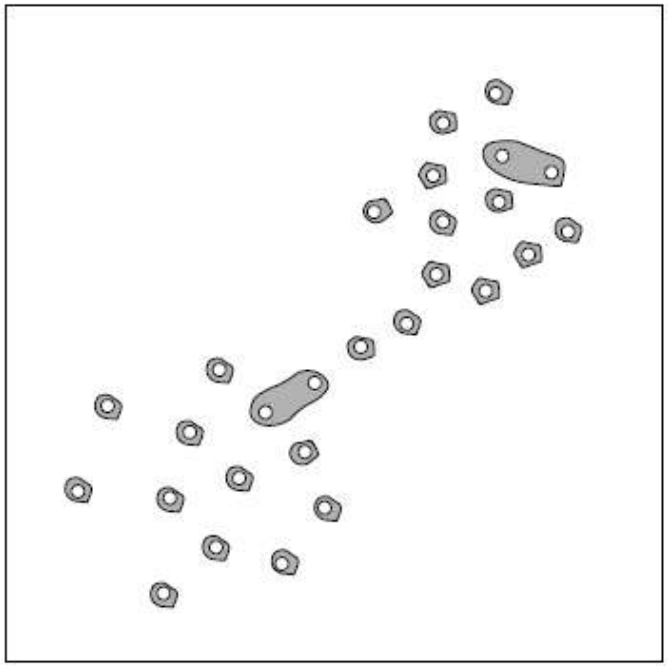
Single Link Problem: Chaining



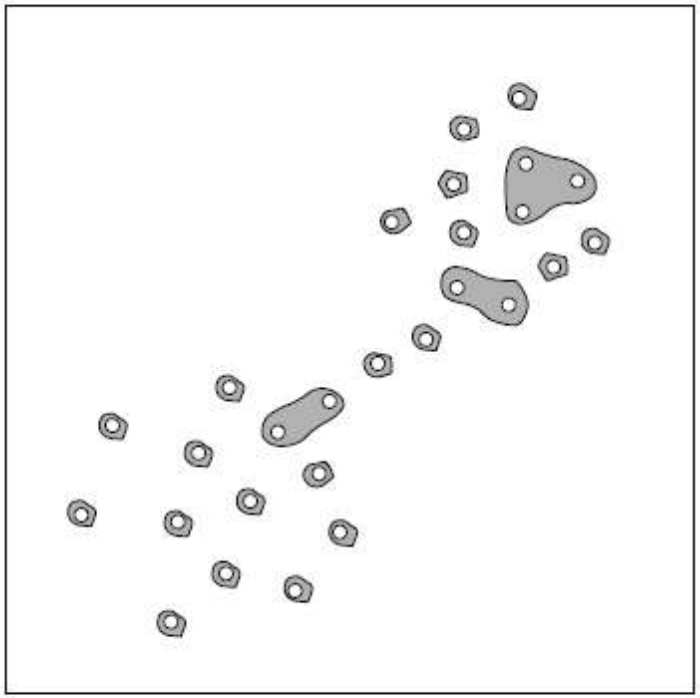
Single Link Problem: Chaining



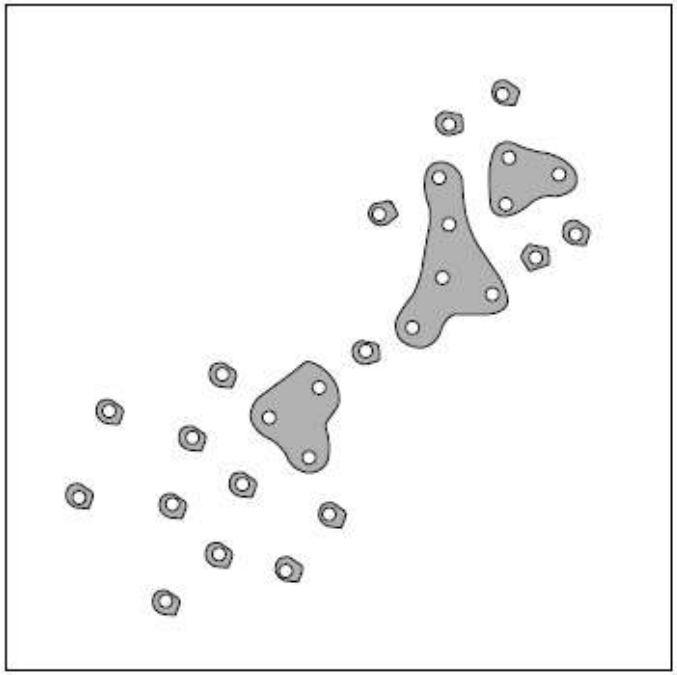
Single Link Problem: Chaining



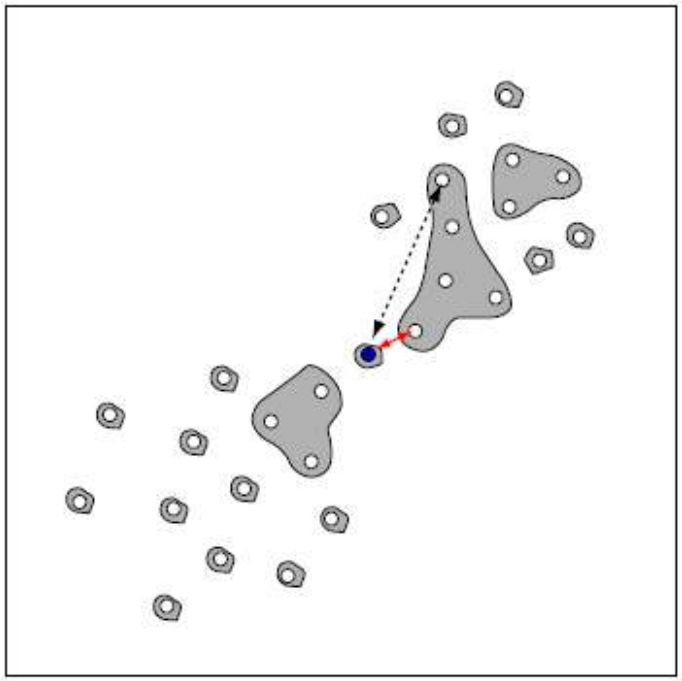
Single Link Problem: Chaining



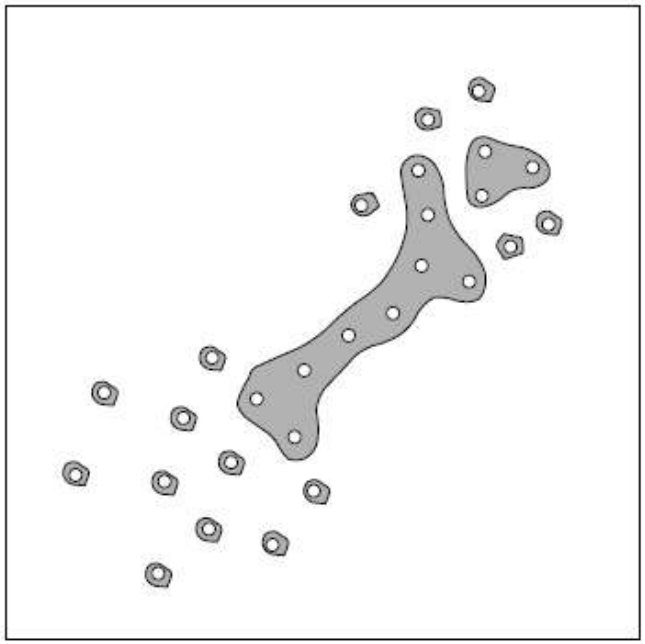
Single Link Problem: Chaining



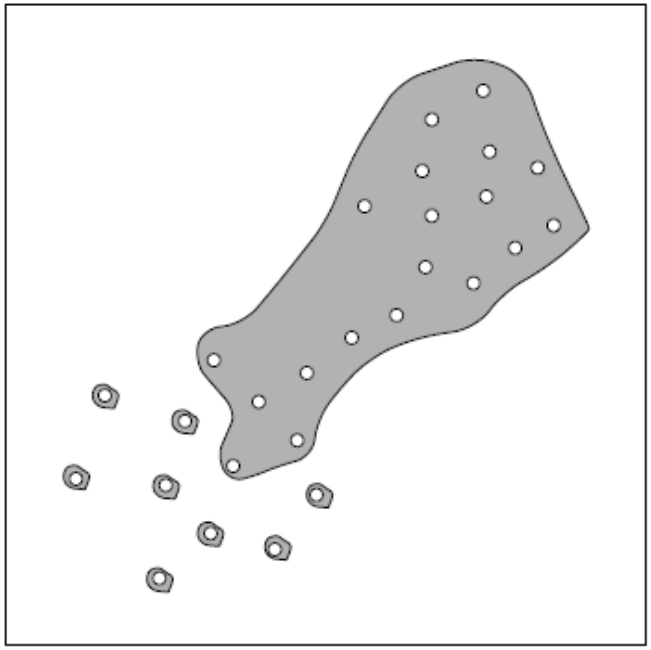
Single Link Problem: Chaining



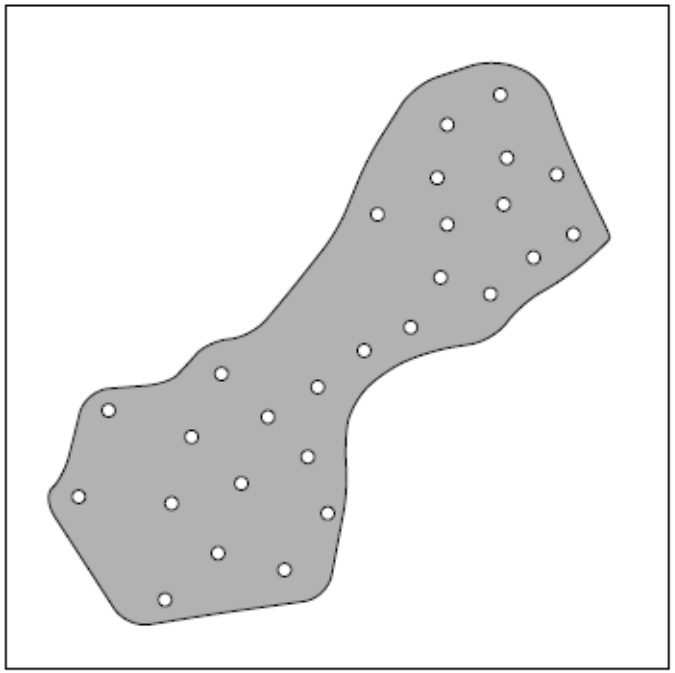
Single Link Problem: Chaining

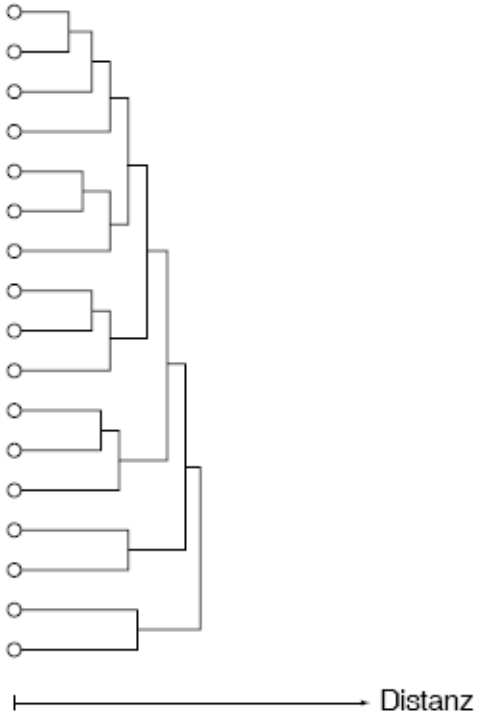


Single Link Problem: Chaining

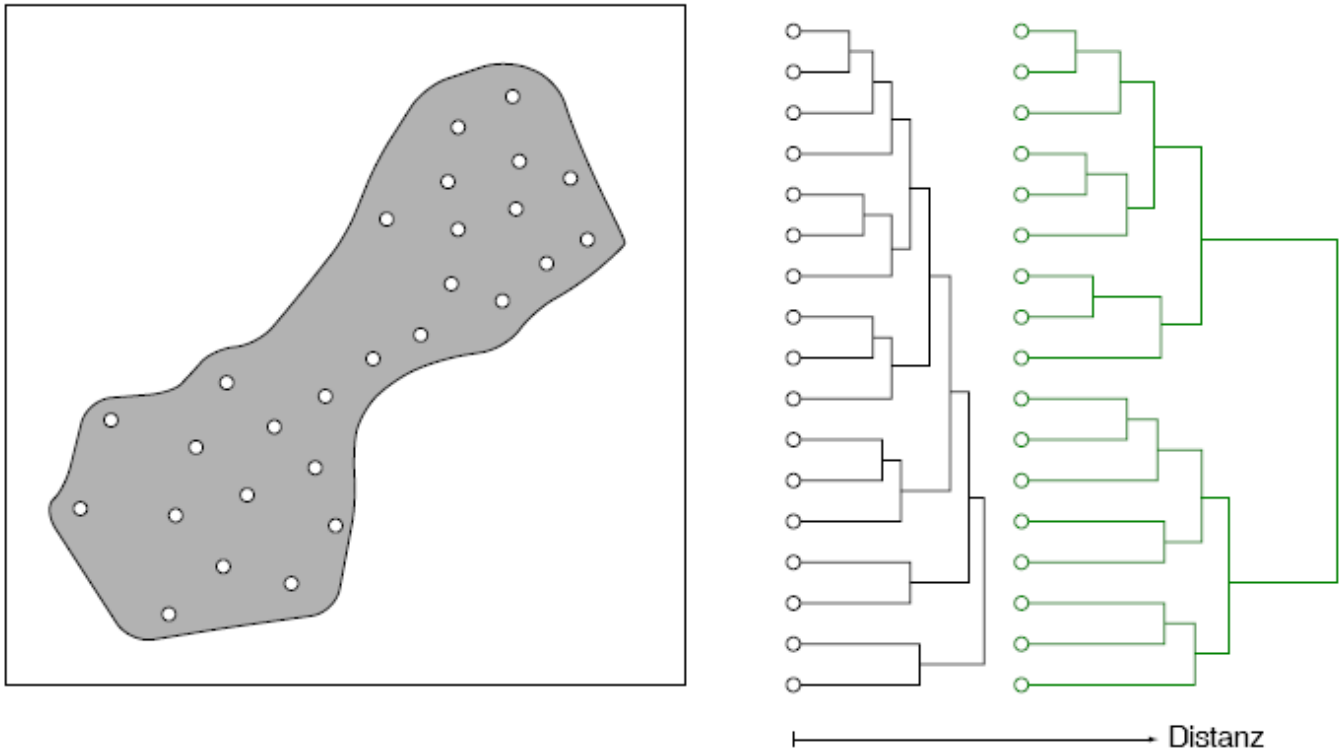


Single Link Problem: Chaining

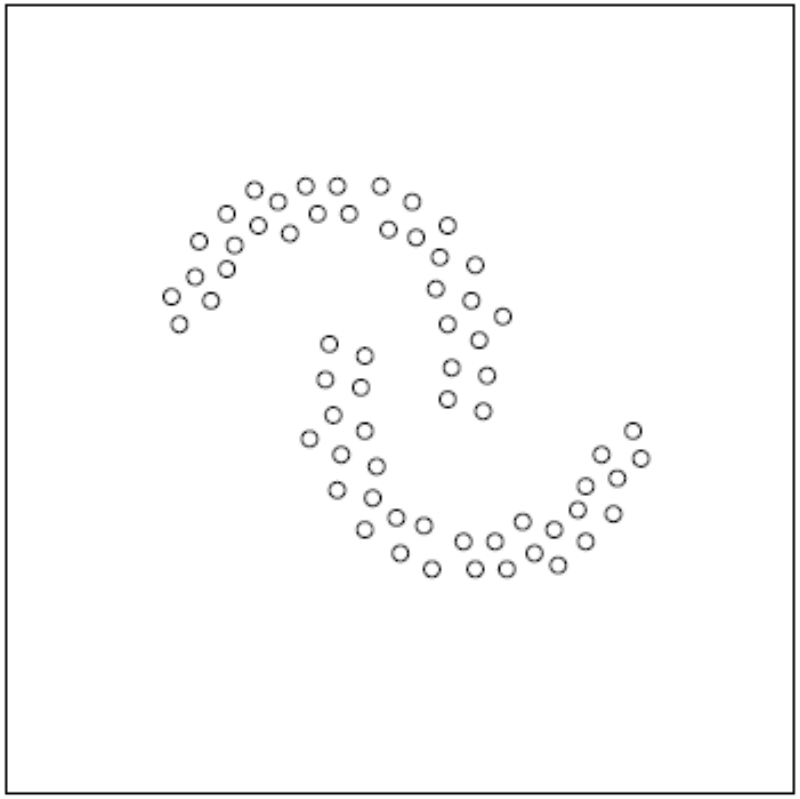




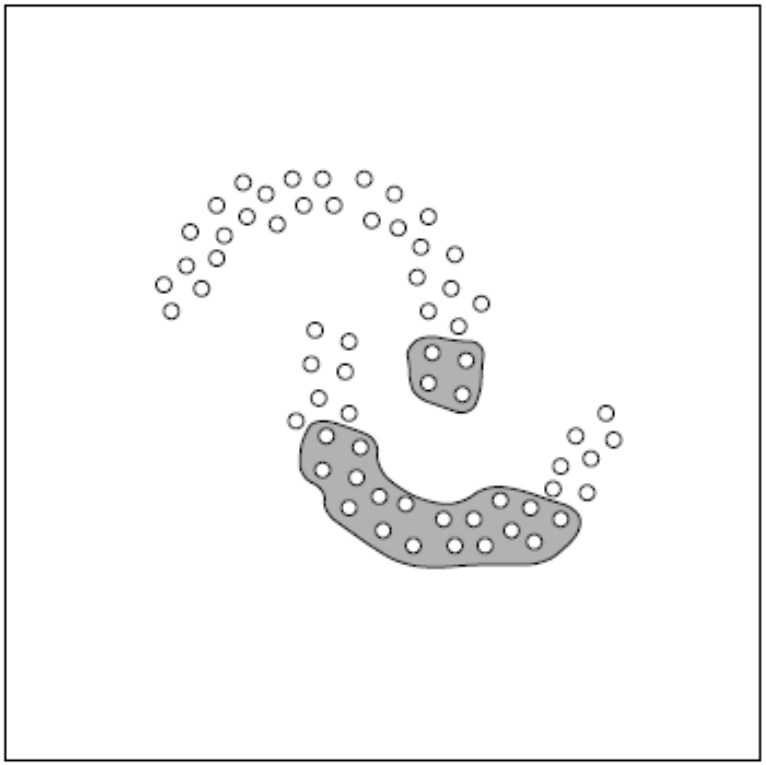
Single Link Problem: Chaining



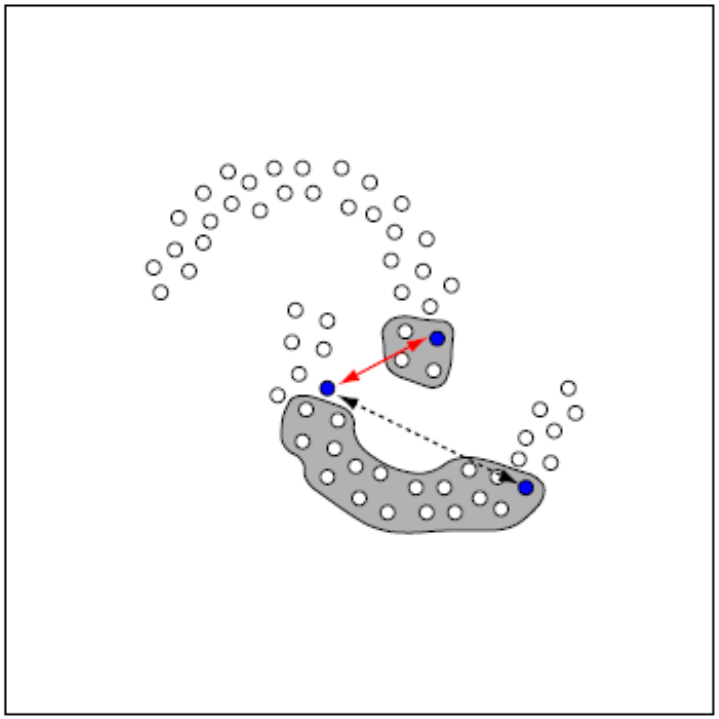
Complete Link Problem: Overlap



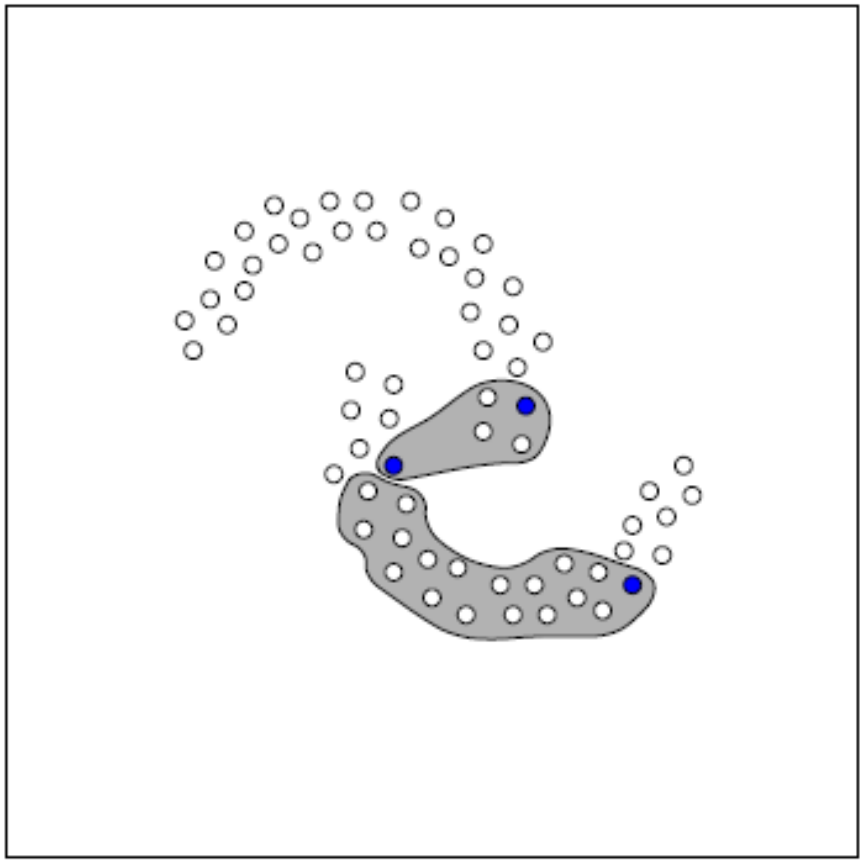
Complete Link Problem: Overlap



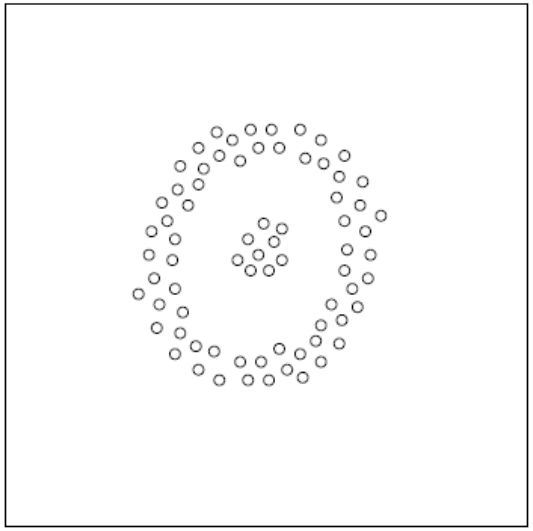
Complete Link Problem: Overlap



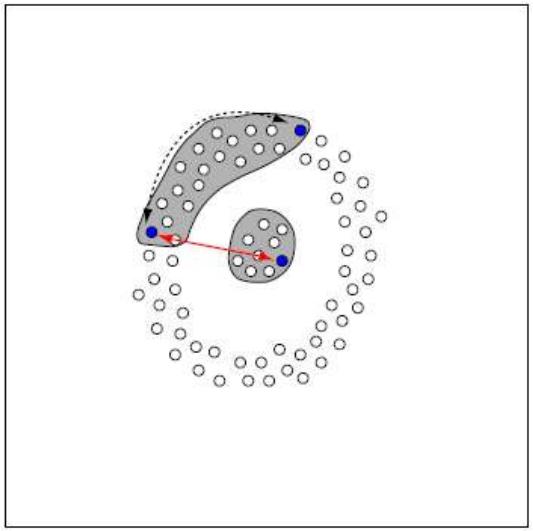
Complete Link Problem: Overlap



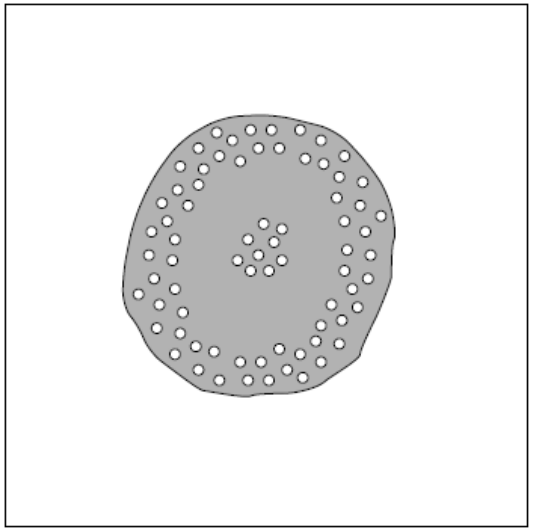
Complete Link Problem: Overlap



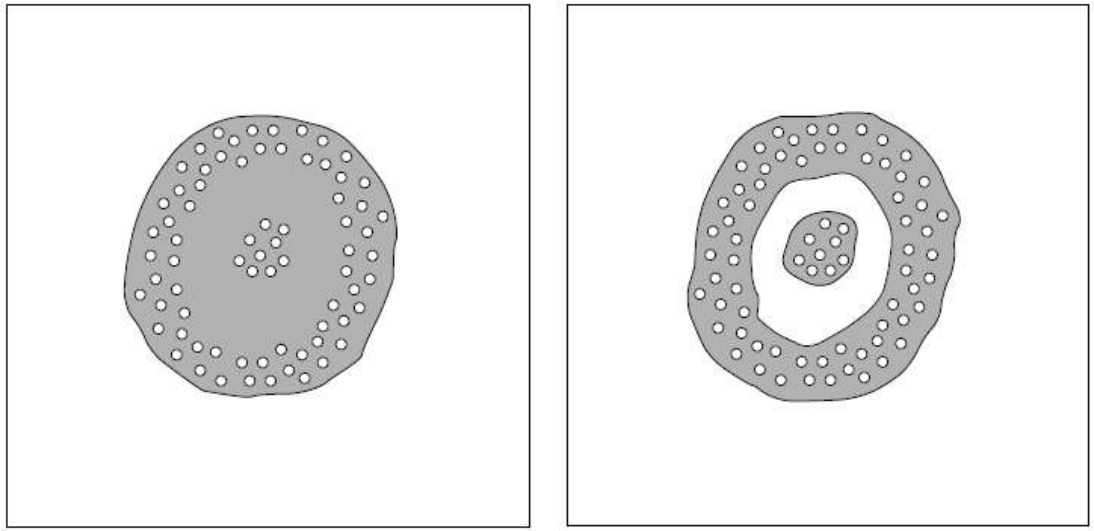
Complete Link Problem: Overlap



Complete Link Problem: Overlap



Complete Link Problem: Overlap



Hierarchical Clustering: Comparison

	Single Link	Complete Link	Average Link	Ward
# clusters	*small*	*high*	*medium*	*medium*
cluster type	*stretched*	*small*	*compact*	*spherical*
chaining tendency	*high*	*low*	*low*	*low*
outlier detection	*high*	*very low*	*low*	*low*

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

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

1. $t = 0$
2. **FOR** $i = 1$ to k **DO** $r_i(t) = \mathit{choose}(V)$ // 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. $x = \underset{i:\, i \in \{1,\ldots,k\}}{\operatorname{argmin}} \; d(r_i(t), v), \quad C_x = C_x \cup \{v\}$
7. **ENDDO**
8. **FOR** $i = 1$ to k **DO** $r_i(t) = \mathit{minimize}(e, C_i)$ // update
9. **UNTIL**$(\forall r_i : d(r_i(t), r_i(t-1)) < \varepsilon \;\vee\; t > t_{\max})$
10. **RETURN**$(\{r_1(t), \ldots, r_k(t)\})$

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.

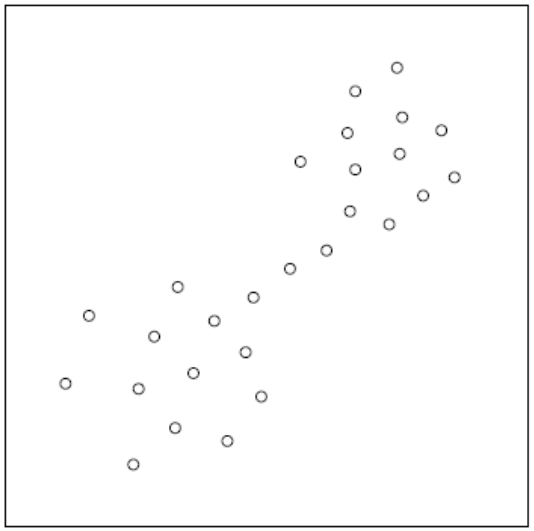
Iterative Clustering Algorithm

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

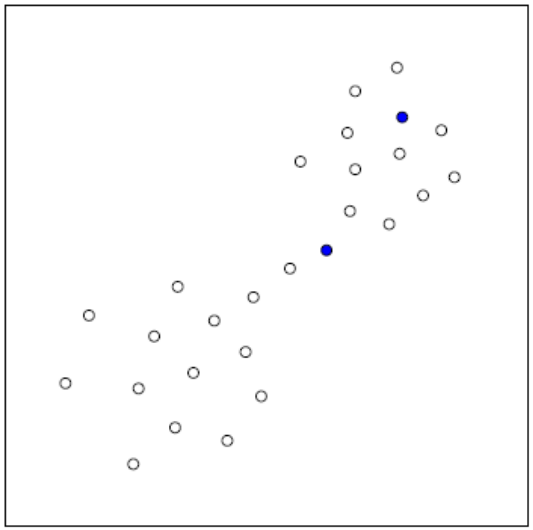
Iterative Clustering Algorithm

- 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

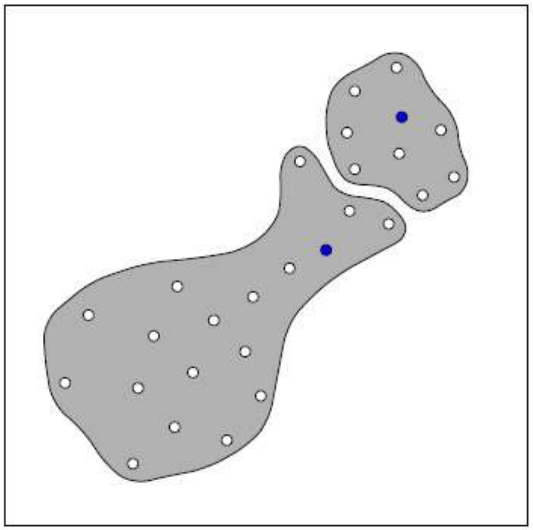
K-Means Example



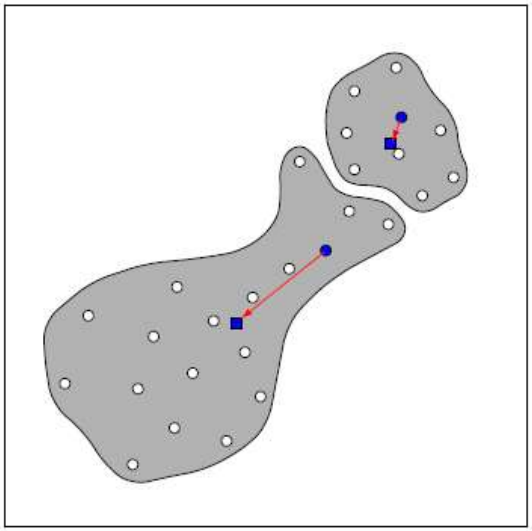
K-Means Example



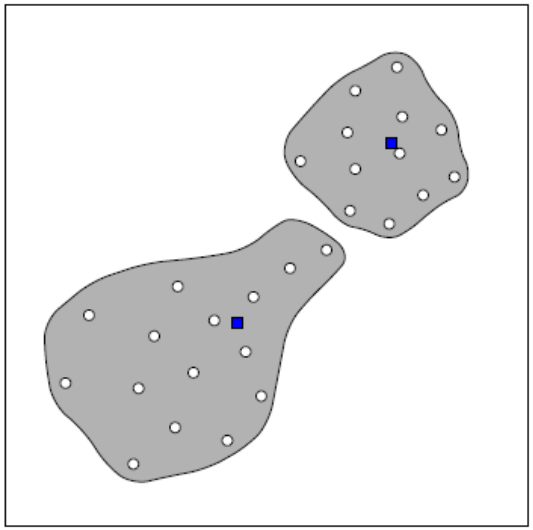
K-Means Example



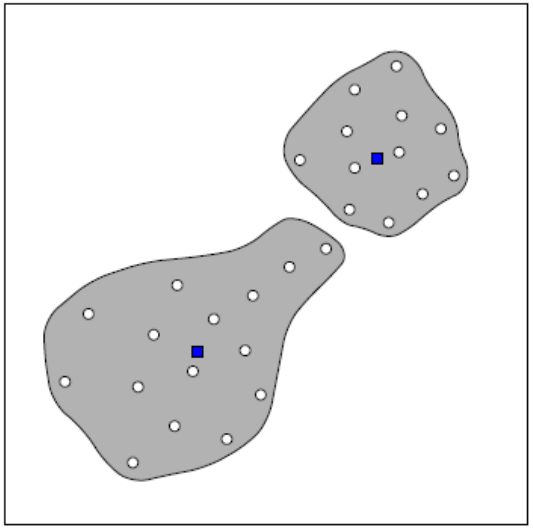
K-Means Example



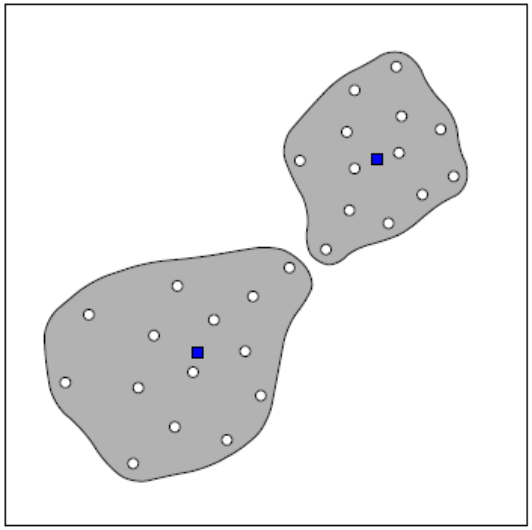
K-Means Example



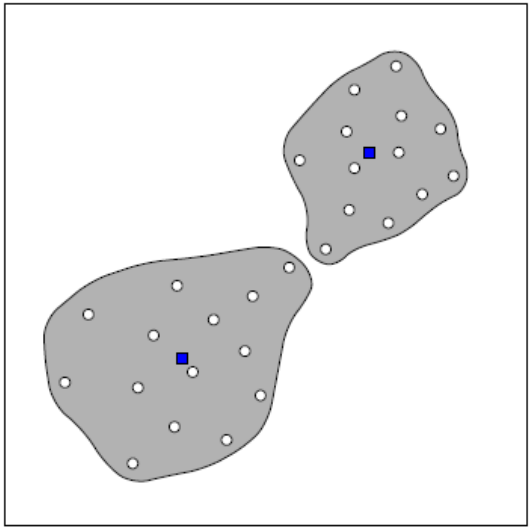
K-Means Example



K-Means Example



K-Means Example



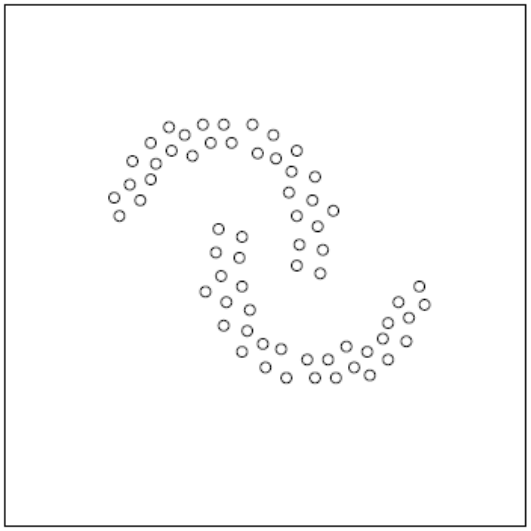
Cluster Center

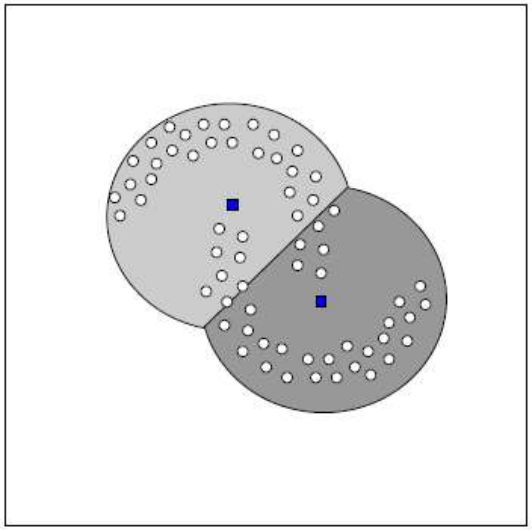
$e(\mathcal{C}) = \sum_{i=1}^{k} \sum_{v \in C_i} (v - r_i)^2$	$r_i = \bar{v}_{C_i}$	Centroid-Berechnung (k-Means)
$e(\mathcal{C}) = \sum_{i=1}^{k} \sum_{v \in C_i} \lvert v - r_i \rvert$	$r_i \in C_i$	Medoid-Berechnung (k-Medoid)
$e(\mathcal{C}) = \sum_{i=1}^{k} \max_{v \in C_i} \lvert v - r_i \rvert$	$r_i \in C_i$	k-Center
$e(\mathcal{C}) = \sum_{i=1}^{k} \sum_{v \in V} \mu_{v_i}^2 \cdot (v - r_i)^2$	$r_i = \frac{\sum_{v \in V} \mu_{v_i}^2 \cdot v}{\sum_{v \in V} \mu_{v_i}^2}$	Fuzzy-k-Means

Method Comparison

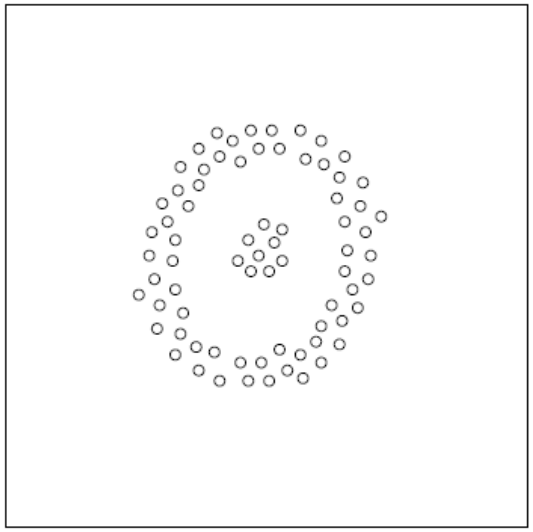
- K-Means & Fuzzy K-Means are based on interval scaled features
  - Cluster center is artificial
- K-Medoid & K-Center work with arbitrary distance and similarity functions
  - Cluster center is part of the objects
  - Medoid is more robust against outliers

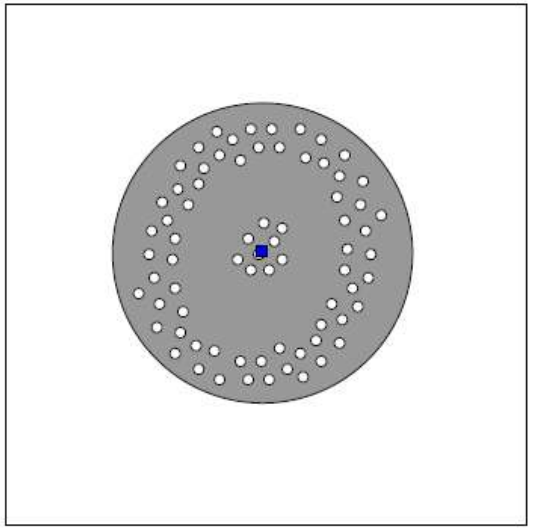
K-Means Problems



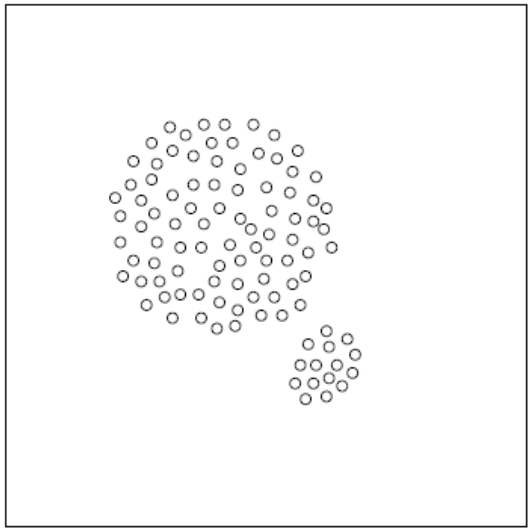


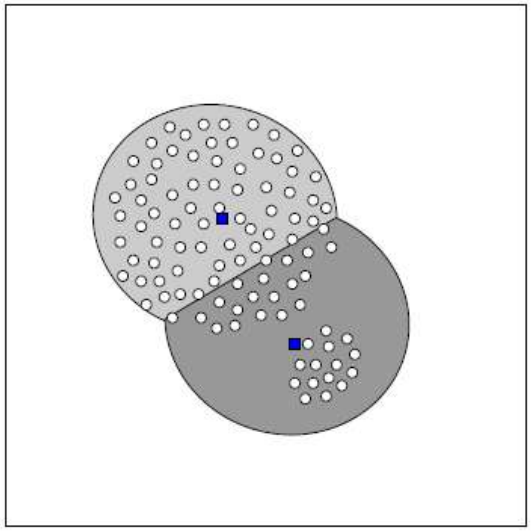
K-Means Problems





K-Means Problems





Thanks ...