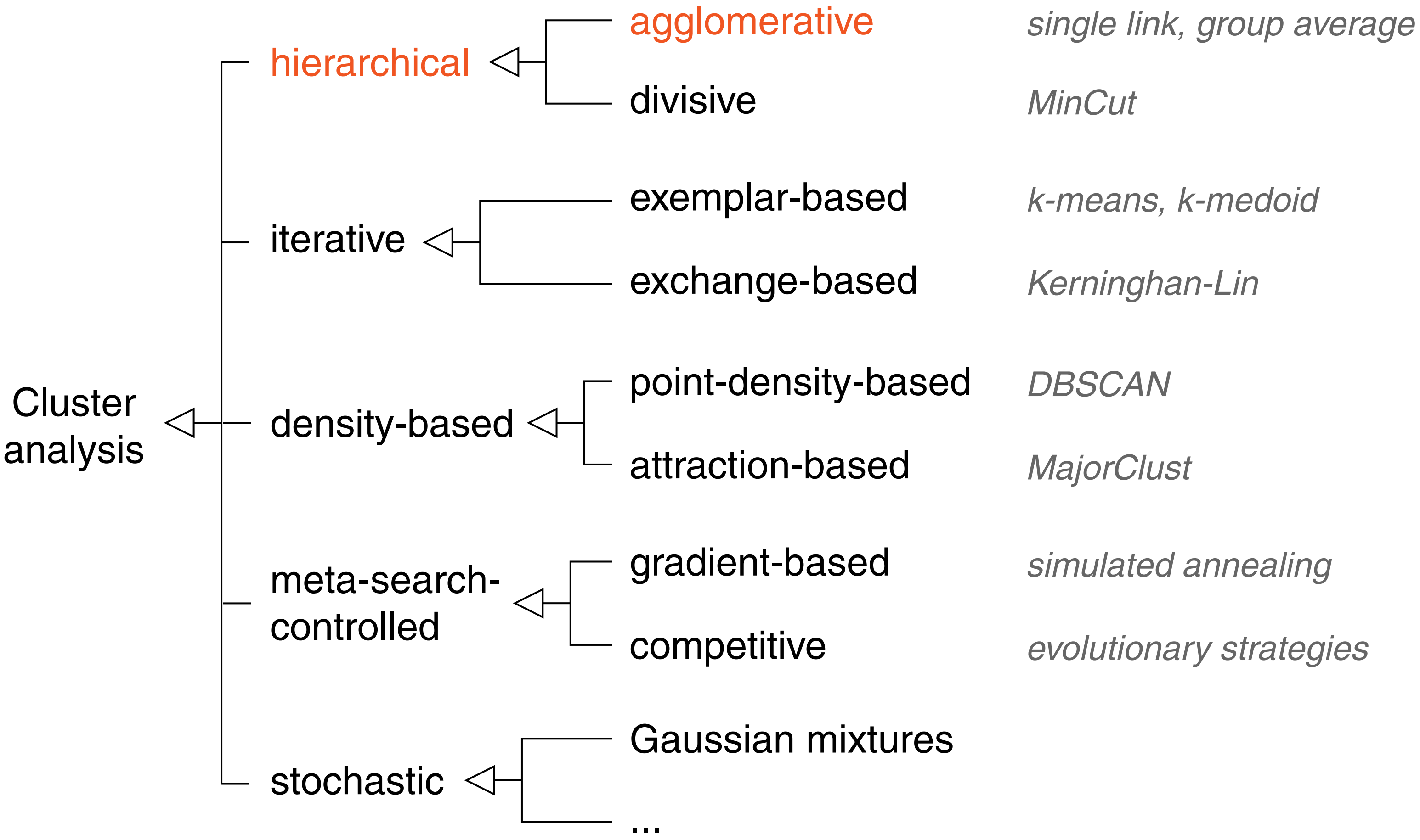
Chapter DM:II (continued)

II. Cluster Analysis

- Cluster Analysis Basics
- Hierarchical Cluster Analysis
- Iterative Cluster Analysis
- Density-Based Cluster Analysis
- Cluster Evaluation
- Constrained Cluster Analysis

Hierarchical Cluster Analysis

Merging Principles



Cluster analysis

hierarchical

agglomerative — *single link, group average*

divisive — *MinCut*

iterative

exemplar-based — *k-means, k-medoid*

exchange-based — *Kerninghan-Lin*

density-based

point-density-based — *DBSCAN*

attraction-based — *MajorClust*

meta-search-controlled

gradient-based — *simulated annealing*

competitive — *evolutionary strategies*

stochastic

Gaussian mixtures

...

Hierarchical Cluster Analysis

Hierarchical Agglomerative Algorithm

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

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

```
$\mathcal{C} = \{\{v\} \mid v \in V\}$ // initial clustering

WHILE $|\mathcal{C}| > 1$ DO
    update_distance_matrix$(\mathcal{C}, G, d_\mathcal{C})$
    $\{C, C'\} = \underset{\{C_i,C_j\}\in\mathcal{C}:\, C_i\neq C_j}{\text{argmin}} \; d_\mathcal{C}(C_i, C_j)$
    $\mathcal{C} = (\mathcal{C} \setminus \{C, C'\}) \;\cup\; \{C \cup C'\}$ // merging

ENDDO
RETURN$(T)$
```

Compare the above algorithm to the hierarchical divisive algorithm.

Hierarchical Cluster Analysis

Hierarchical Agglomerative Algorithm

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

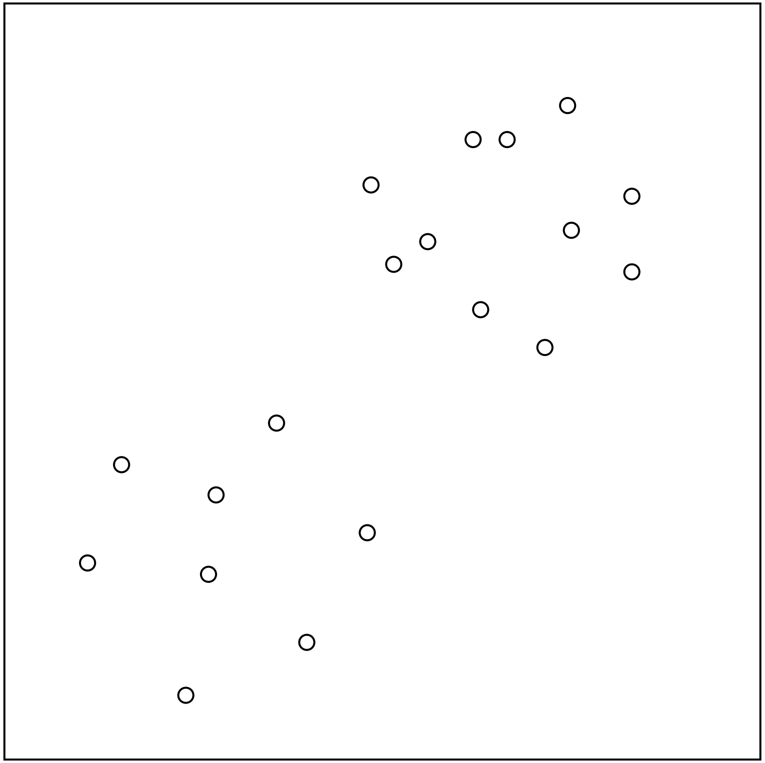
Output: $T = \langle V_T, E_T \rangle$. Cluster hierarchy or dendrogram.

```
$\mathcal{C} = \{\{v\} \mid v \in V\}$   // initial clustering
$V_T = \{v_C \mid C \in \mathcal{C}\}$,  $E_T = \emptyset$   // initial dendrogram
WHILE $|\mathcal{C}| > 1$ DO
    $\textit{update\_distance\_matrix}(\mathcal{C}, G, d_\mathcal{C})$
    $\{C, C'\} = \underset{\{C_i, C_j\} \in \mathcal{C}: C_i \neq C_j}{\text{argmin}} d_\mathcal{C}(C_i, C_j)$
    $\mathcal{C} = (\mathcal{C} \setminus \{C, C'\}) \cup \{C \cup C'\}$   // merging
    $V_T = V_T \cup \{v_{C,C'}\}$,  $E_T = E_T \cup \{\{v_{C,C'}, v_C\}, \{v_{C,C'}, v_{C'}\}\}$   // dendrogram
ENDDO
RETURN$(T)$
```

Compare the above algorithm to the hierarchical divisive algorithm.

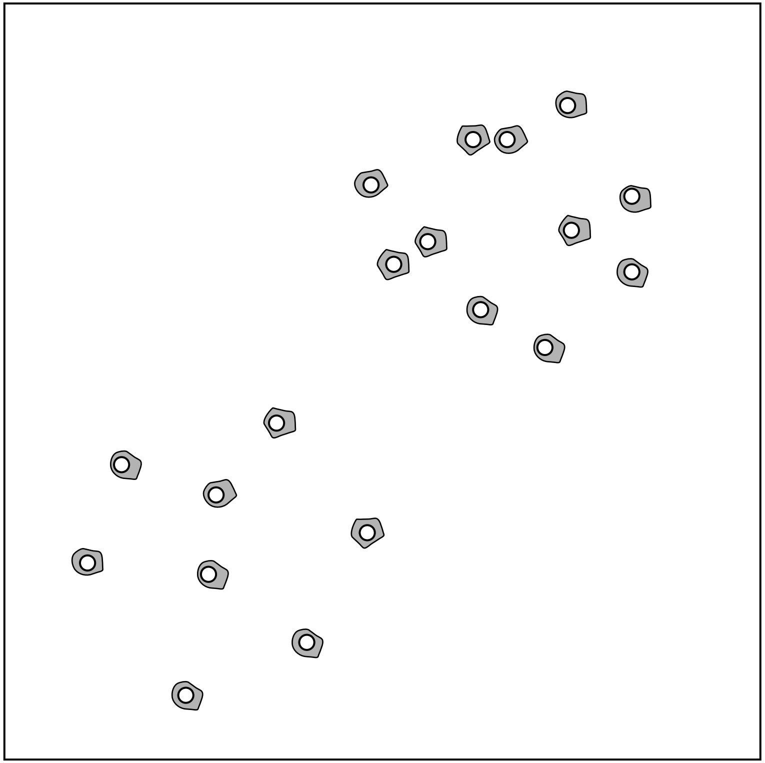
Hierarchical Cluster Analysis

Single Link: Cluster Distance Measure $d_{\mathcal{C}}$ = Nearest Neighbor



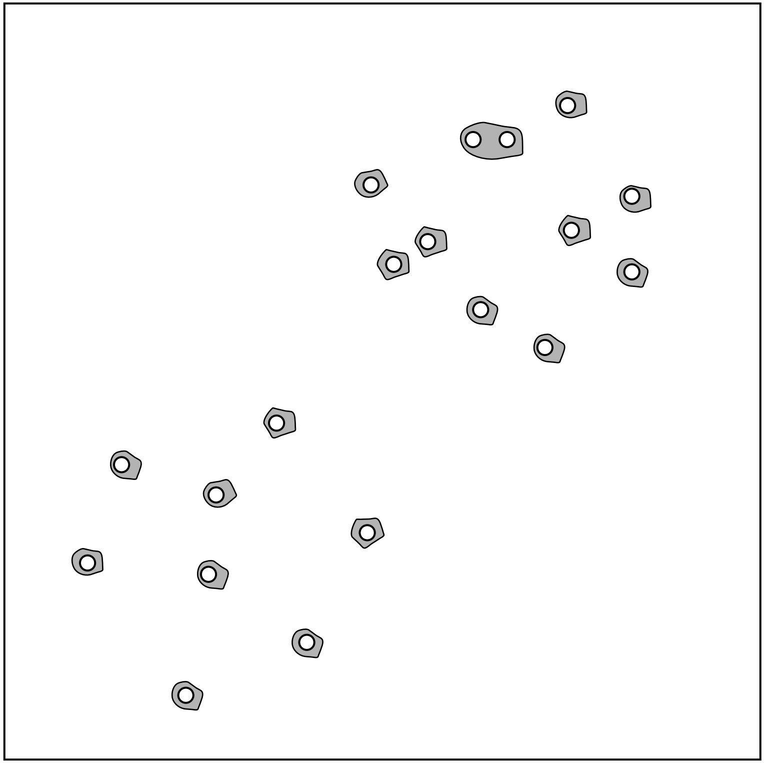
Hierarchical Cluster Analysis

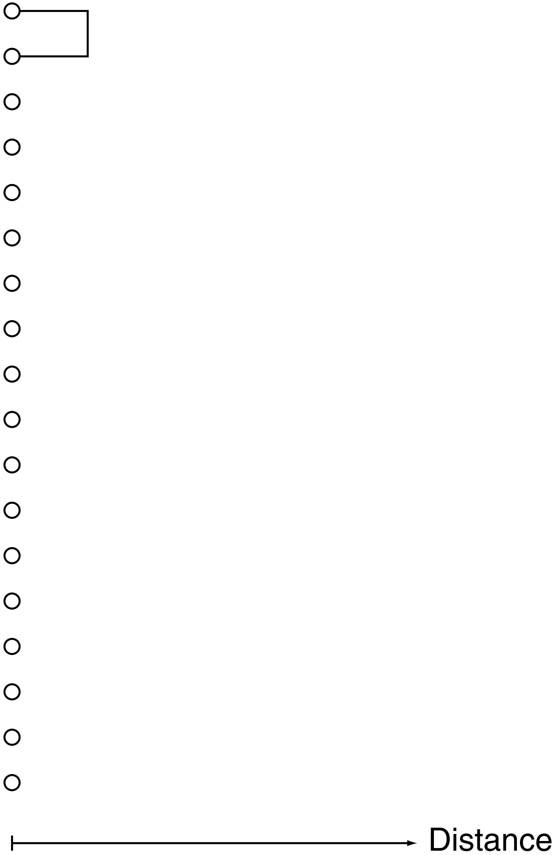
Single Link: Cluster Distance Measure $d_{\mathcal{C}}$ = Nearest Neighbor



Hierarchical Cluster Analysis

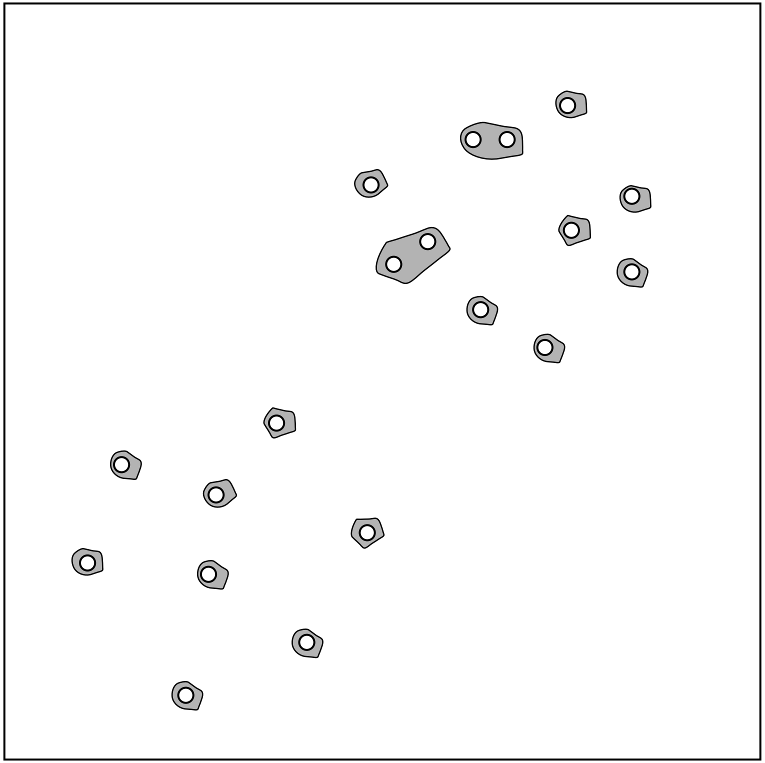
Single Link: Cluster Distance Measure $d_{\mathcal{C}}$ = Nearest Neighbor

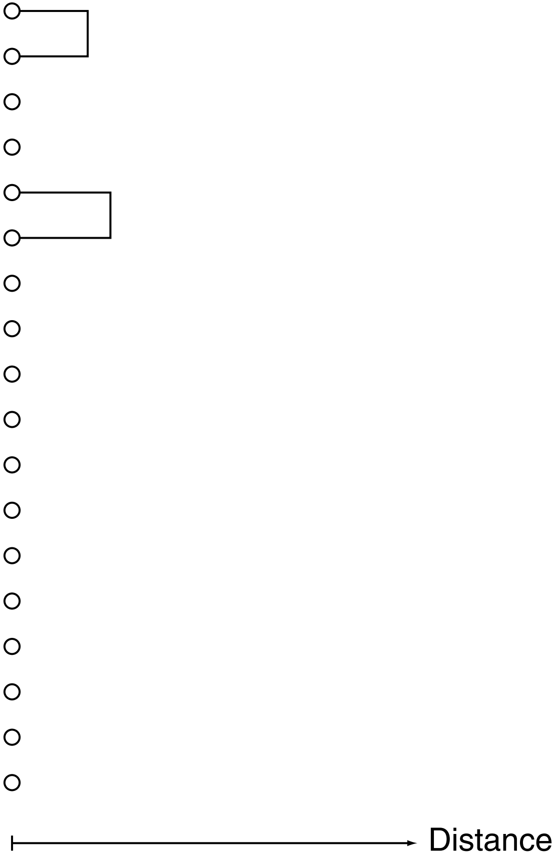




Hierarchical Cluster Analysis

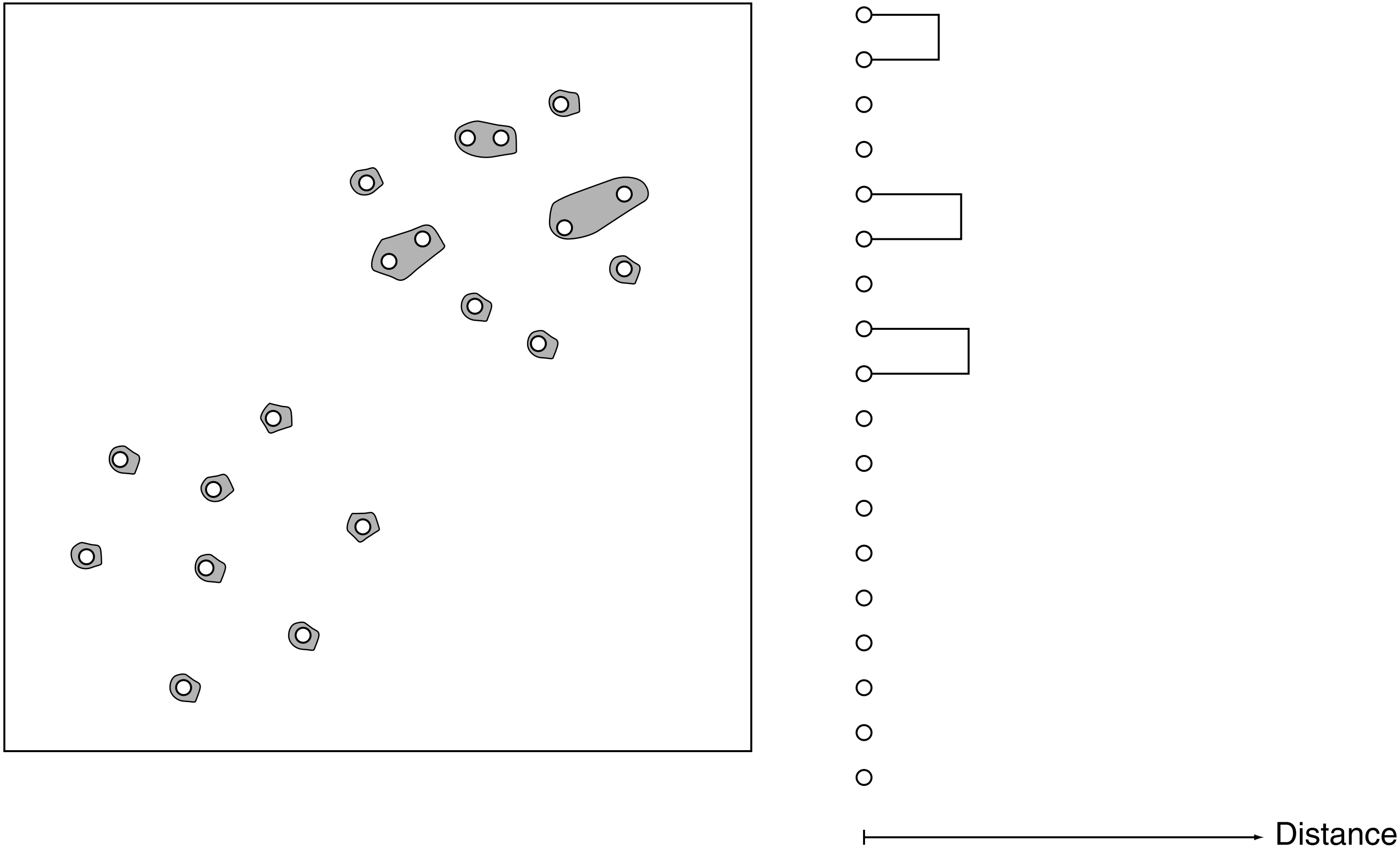
Single Link: Cluster Distance Measure $d_{\mathcal{C}}$ = Nearest Neighbor





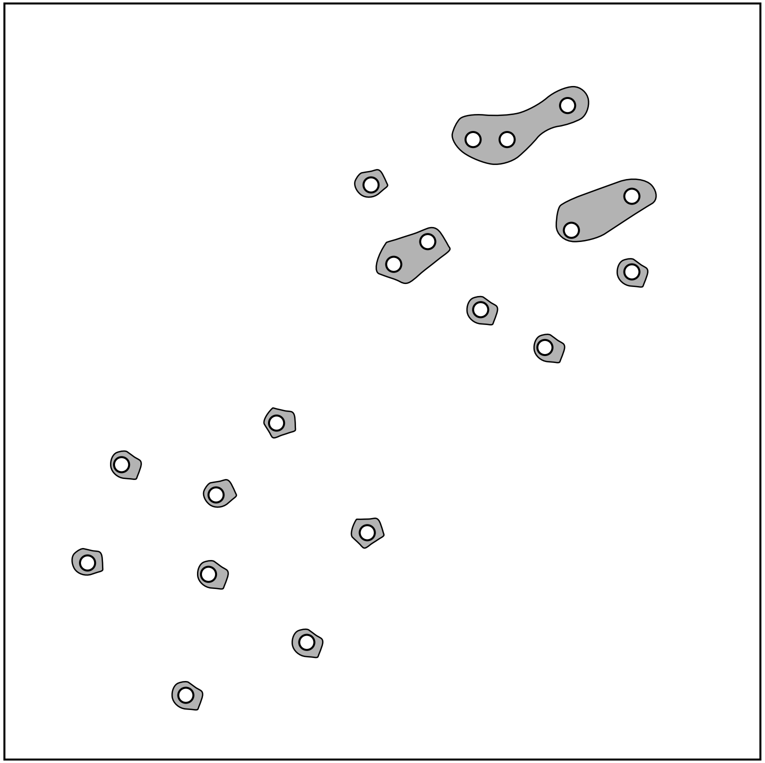
Hierarchical Cluster Analysis

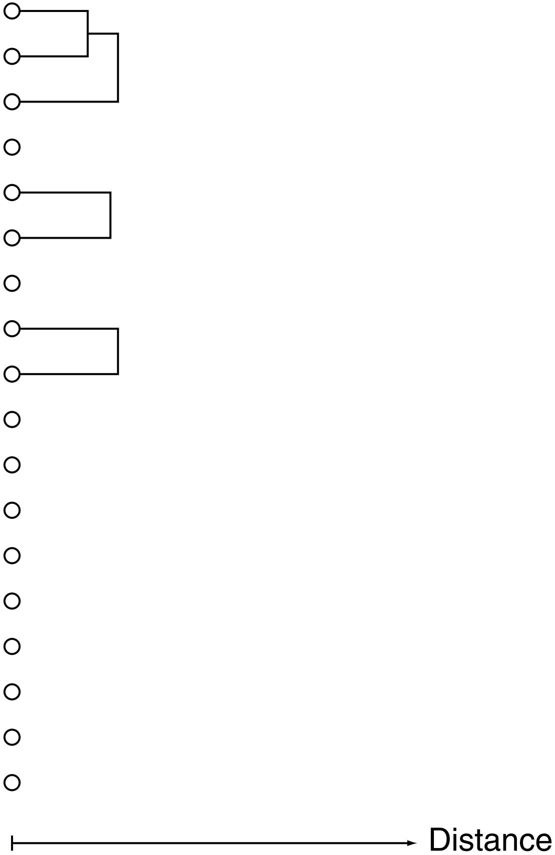
Single Link: Cluster Distance Measure $d_{\mathcal{C}}$ = Nearest Neighbor



Hierarchical Cluster Analysis

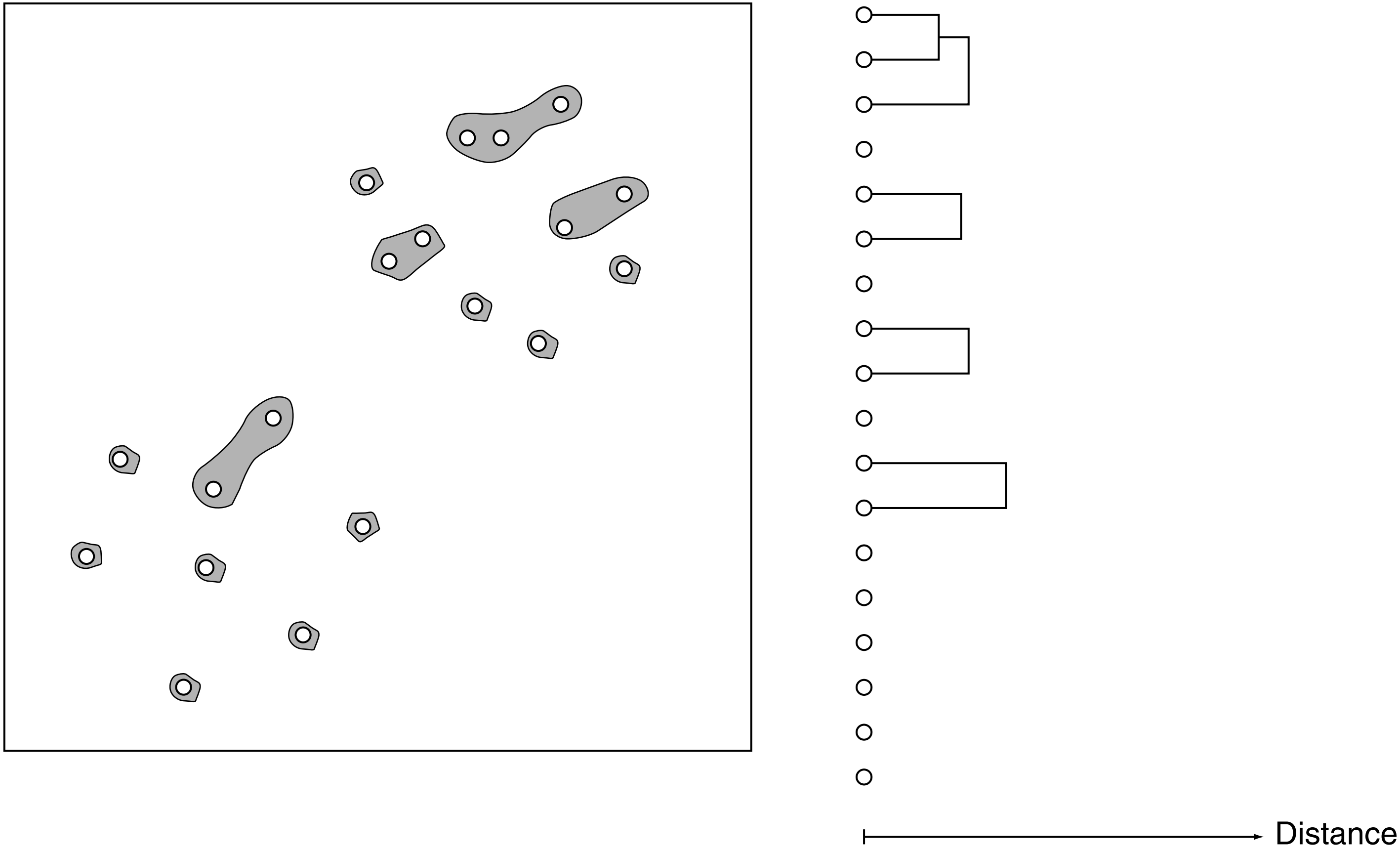
Single Link: Cluster Distance Measure $d_{\mathcal{C}}$ = Nearest Neighbor





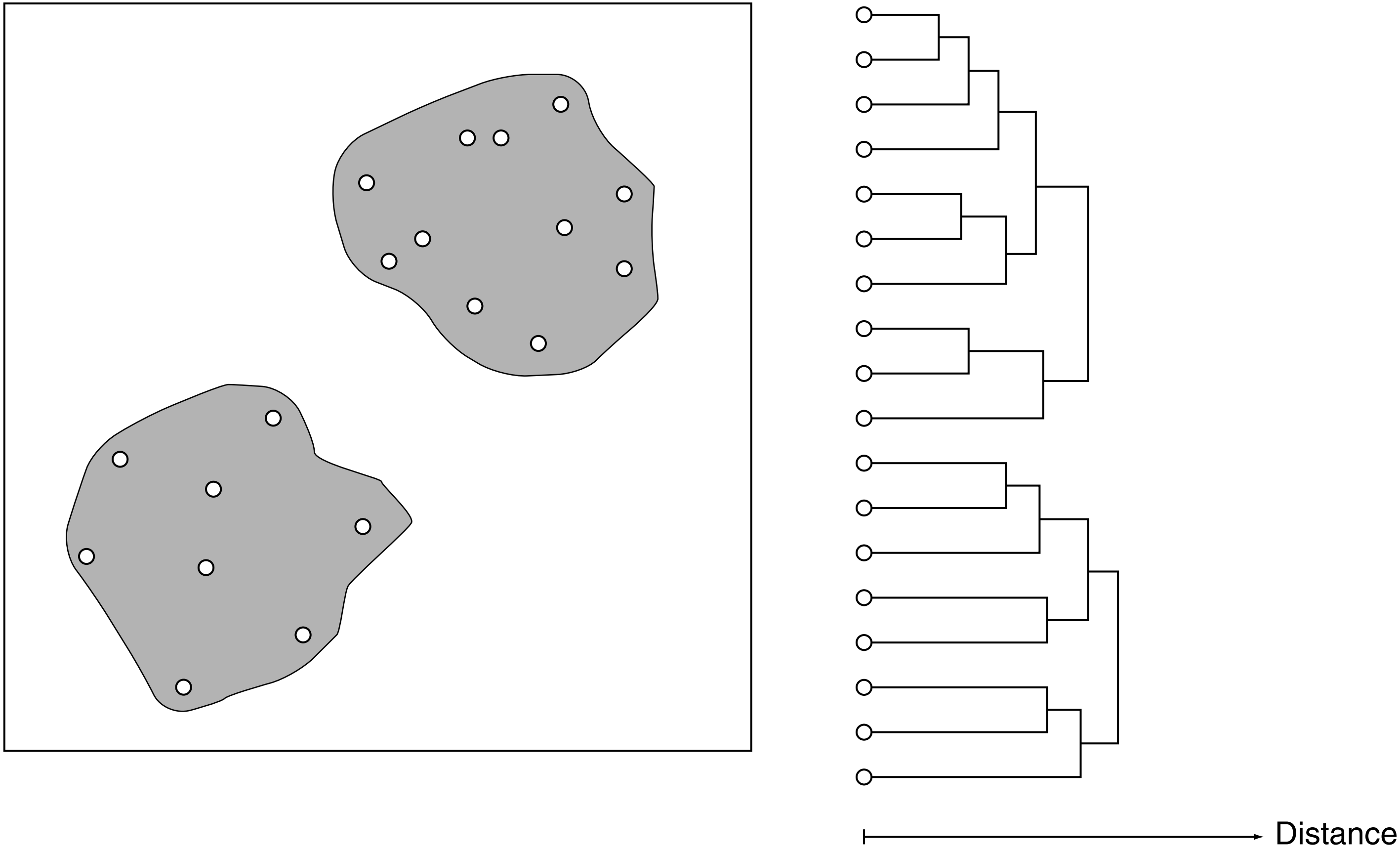
Hierarchical Cluster Analysis

Single Link: Cluster Distance Measure $d_{\mathcal{C}}$ = Nearest Neighbor



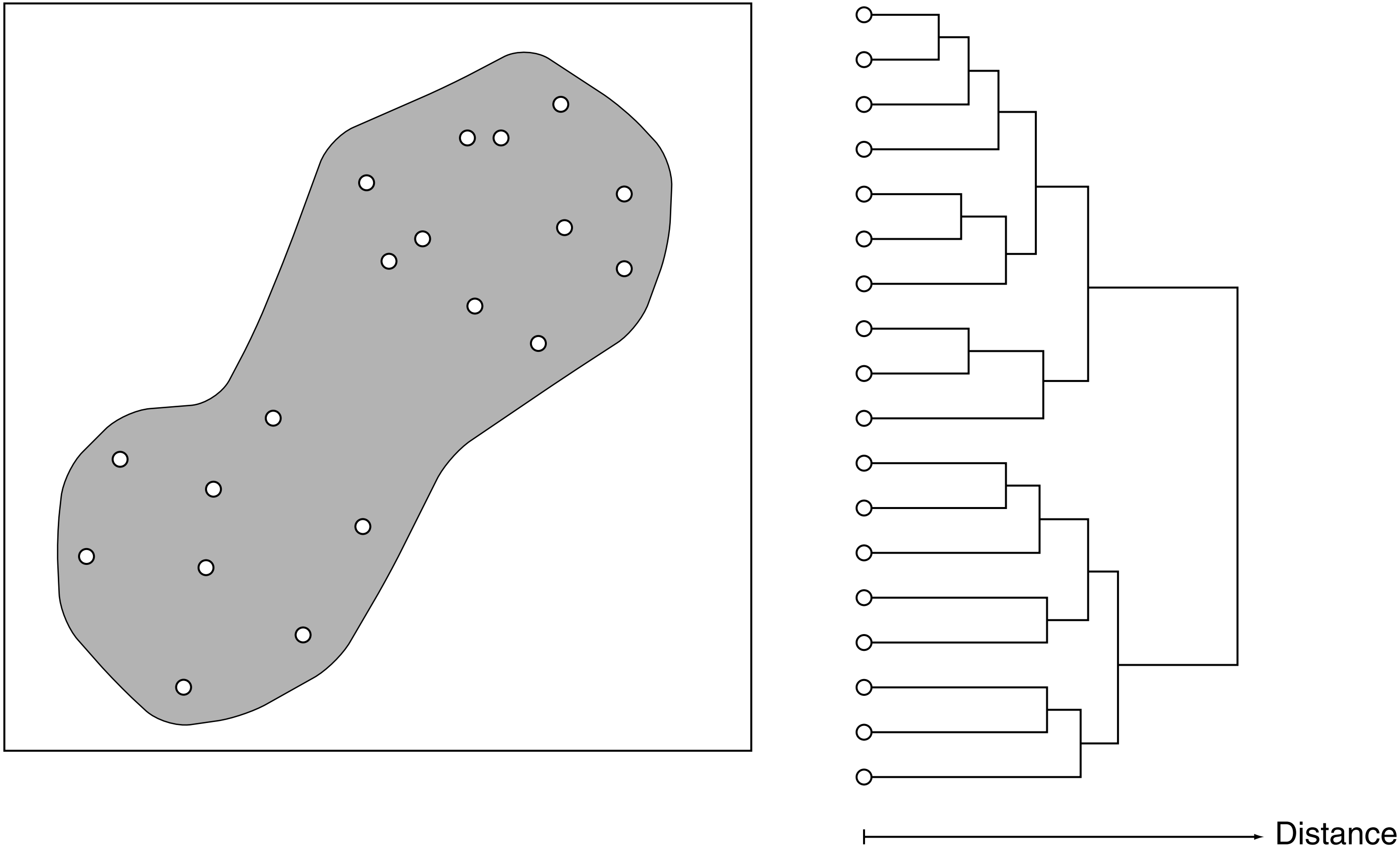
Hierarchical Cluster Analysis

Single Link: Cluster Distance Measure $d_{\mathcal{C}}$ = Nearest Neighbor



Hierarchical Cluster Analysis

Single Link: Cluster Distance Measure $d_{\mathcal{C}}$ = Nearest Neighbor



Hierarchical Cluster Analysis

Distance Re-Computation after each Merging Step [algorithm]

$t = 0$

	$\mathbf{C}_1$	$\mathbf{C}_2$	$\ldots$	$\mathbf{C}_n$
$\mathbf{C}_1$	0	$d_{\mathcal{C}}(\mathbf{C}_1, \mathbf{C}_2)$	$\ldots$	$d_{\mathcal{C}}(\mathbf{C}_1, \mathbf{C}_n)$
$\mathbf{C}_2$	-	0	$\ldots$	$d_{\mathcal{C}}(\mathbf{C}_2, \mathbf{C}_n)$
$\vdots$				
$\mathbf{C}_n$	-	-	$\ldots$	0

$\equiv$

	$\mathbf{x}_1$	$\mathbf{x}_2$	$\ldots$	$\mathbf{x}_n$
$\mathbf{x}_1$	0	$d(\mathbf{x}_1, \mathbf{x}_2)$	$\ldots$	$d(\mathbf{x}_1, \mathbf{x}_n)$
$\mathbf{x}_2$	-	0	$\ldots$	$d(\mathbf{x}_2, \mathbf{x}_n)$
$\vdots$				
$\mathbf{x}_n$	-	-	$\ldots$	0

Hierarchical Cluster Analysis

Distance Re-Computation after each Merging Step [algorithm]

$t = 0$

	$\mathbf{C}_1$	$\mathbf{C}_2$	$\ldots$	$\mathbf{C}_n$
$\mathbf{C}_1$	0	$d_{\mathcal{C}}(\mathbf{C}_1, \mathbf{C}_2)$	$\ldots$	$d_{\mathcal{C}}(\mathbf{C}_1, \mathbf{C}_n)$
$\mathbf{C}_2$	-	0	$\ldots$	$d_{\mathcal{C}}(\mathbf{C}_2, \mathbf{C}_n)$
$\vdots$				
$\mathbf{C}_n$	-	-	$\ldots$	0

$\equiv$

	$\mathbf{x}_1$	$\mathbf{x}_2$	$\ldots$	$\mathbf{x}_n$
$\mathbf{x}_1$	0	$d(\mathbf{x}_1, \mathbf{x}_2)$	$\ldots$	$d(\mathbf{x}_1, \mathbf{x}_n)$
$\mathbf{x}_2$	-	0	$\ldots$	$d(\mathbf{x}_2, \mathbf{x}_n)$
$\vdots$				
$\mathbf{x}_n$	-	-	$\ldots$	0

Hierarchical Cluster Analysis

Distance Re-Computation after each Merging Step [algorithm]

$t = 0$

	$\mathbf{C}_1$	$\mathbf{C}_2$	$\ldots$	$\mathbf{C}_n$
$\mathbf{C}_1$	0	$d_{\mathcal{C}}(\mathbf{C}_1, \mathbf{C}_2)$	$\ldots$	$d_{\mathcal{C}}(\mathbf{C}_1, \mathbf{C}_n)$
$\mathbf{C}_2$	-	0	$\ldots$	$d_{\mathcal{C}}(\mathbf{C}_2, \mathbf{C}_n)$
$\vdots$				
$\mathbf{C}_n$	-	-	$\ldots$	0

$\equiv$

	$\mathbf{x}_1$	$\mathbf{x}_2$	$\ldots$	$\mathbf{x}_n$
$\mathbf{x}_1$	0	$d(\mathbf{x}_1, \mathbf{x}_2)$	$\ldots$	$d(\mathbf{x}_1, \mathbf{x}_n)$
$\mathbf{x}_2$	-	0	$\ldots$	$d(\mathbf{x}_2, \mathbf{x}_n)$
$\vdots$				
$\mathbf{x}_n$	-	-	$\ldots$	0

↓

$t = i$

	$\mathbf{C}_{i_1}$	$\mathbf{C}_{i_2}$	$\ldots$	$\mathbf{C}i_{n-i}$
$\mathbf{C}_{i_1}$	0	$d_{\mathcal{C}}(\mathbf{C}_1, \mathbf{C}_{i_2})$	$\ldots$	$d_{\mathcal{C}}(\mathbf{C}_{i_1}, \mathbf{C}i_{n-i})$
$\mathbf{C}_{i_2}$	-	0	$\ldots$	$d_{\mathcal{C}}(\mathbf{C}_{i_2}, \mathbf{C}i_{n-i})$
$\vdots$				
$\mathbf{C}_{i_{n-i}}$	-	-	$\ldots$	0

Hierarchical Cluster Analysis

Distance Re-Computation after each Merging Step [algorithm]

$t = 0$

	$\mathbf{C}_1$	$\mathbf{C}_2$	$\ldots$	$\mathbf{C}_n$
$\mathbf{C}_1$	0	$d_{\mathcal{C}}(\mathbf{C}_1, \mathbf{C}_2)$	$\ldots$	$d_{\mathcal{C}}(\mathbf{C}_1, \mathbf{C}_n)$
$\mathbf{C}_2$	-	0	$\ldots$	$d_{\mathcal{C}}(\mathbf{C}_2, \mathbf{C}_n)$
$\vdots$				
$\mathbf{C}_n$	-	-	$\ldots$	0

$\equiv$

	$\mathbf{x}_1$	$\mathbf{x}_2$	$\ldots$	$\mathbf{x}_n$
$\mathbf{x}_1$	0	$d(\mathbf{x}_1, \mathbf{x}_2)$	$\ldots$	$d(\mathbf{x}_1, \mathbf{x}_n)$
$\mathbf{x}_2$	-	0	$\ldots$	$d(\mathbf{x}_2, \mathbf{x}_n)$
$\vdots$				
$\mathbf{x}_n$	-	-	$\ldots$	0

↓

$t = i$

	$\mathbf{C}_{i_1}$	$\mathbf{C}_{i_2}$	$\ldots$	$\mathbf{C}i_{n-i}$
$\mathbf{C}_{i_1}$	0	$d_{\mathcal{C}}(\mathbf{C}_1, \mathbf{C}_{i_2})$	$\ldots$	$d_{\mathcal{C}}(\mathbf{C}_{i_1}, \mathbf{C}i_{n-i})$
$\mathbf{C}_{i_2}$	-	0	$\ldots$	$d_{\mathcal{C}}(\mathbf{C}_{i_2}, \mathbf{C}i_{n-i})$
$\vdots$				
$\mathbf{C}_{i_{n-i}}$	-	-	$\ldots$	0

↓

$t = n - 1$ $\qquad$ $\mathbf{C}_{n_1}$

Hierarchical Cluster Analysis

Distance Measures of Hierarchical Agglomerative Algorithms [characteristics]

$d_{\mathcal{C}}(C, C') = \min\limits_{\substack{u \in C \\ v \in C'}} d(u, v)$	single link (nearest neighbor)
$d_{\mathcal{C}}(C, C') = \max\limits_{\substack{u \in C \\ v \in C'}} d(u, v)$	complete link (furthest / farthest neighbor)
$d_{\mathcal{C}}(C, C') = \dfrac{1}{\lvert C\rvert \cdot \lvert C'\rvert} \sum\limits_{\substack{u \in C \\ v \in C'}} d(u, v)$	group average link
$d_{\mathcal{C}}(C, C') = \sqrt{\dfrac{2 \cdot \lvert C\rvert \cdot \lvert C'\rvert}{\lvert C\rvert + \lvert C'\rvert}} \cdot \lVert\bar{u} - \bar{v}\rVert$	Ward criterion (variance) $\lVert\cdot\rVert = \lVert\cdot\rVert_2 =$ Euclidean norm

How the distance measures are employed:

- hierarchical agglomerative algorithm
- hierarchical divisive algorithm

Hierarchical Cluster Analysis

Ward Criterion

Ward is a variance criterion. It is the (double) increase of the error sum of squares, *ESS*, in the new cluster that results from merging the two clusters C and C'.

Derivation:

$$\mathit{ESS}(C) = \sum_{u \in C} ||\bar{u} - u||^2$$

Hierarchical Cluster Analysis

Ward Criterion

Ward is a variance criterion. It is the (double) increase of the error sum of squares, *ESS*, in the new cluster that results from merging the two clusters C and C'.

Derivation:

$$\mathit{ESS}(C) = \sum_{u \in C} ||\bar{u} - u||^2 \; = \; \sum_{u \in C} \Big(||\bar{u}||^2 - 2 \cdot \langle u, \bar{u} \rangle + ||u||^2 \Big)$$

Hierarchical Cluster Analysis

Ward Criterion

Ward is a variance criterion. It is the (double) increase of the error sum of squares, *ESS*, in the new cluster that results from merging the two clusters C and C'.

Derivation:

$$\textit{ESS}(C) = \sum_{u \in C} ||\bar{u} - u||^2 \; = \; \sum_{u \in C} \Big(||\bar{u}||^2 - 2 \cdot \langle u, \bar{u} \rangle + ||u||^2 \Big)$$

$$= |C| \cdot ||\bar{u}||^2 - 2|C| \cdot ||\bar{u}||^2 + \sum_{u \in C} ||u||^2$$

Hierarchical Cluster Analysis

Ward Criterion

Ward is a variance criterion. It is the (double) increase of the error sum of squares, *ESS*, in the new cluster that results from merging the two clusters C and C'.

Derivation:

$$\textit{ESS}(C) = \sum_{u \in C} ||\bar{u} - u||^2 \;=\; \sum_{u \in C} \Big(||\bar{u}||^2 - 2 \cdot \langle u, \bar{u} \rangle + ||u||^2 \Big)$$

$$= |C| \cdot ||\bar{u}||^2 - 2|C| \cdot ||\bar{u}||^2 + \sum_{u \in C} ||u||^2 \;=\; \sum_{u \in C} ||u||^2 \; - |C| \cdot ||\bar{u}||^2$$

Hierarchical Cluster Analysis

Ward Criterion

Ward is a variance criterion. It is the (double) increase of the error sum of squares, *ESS*, in the new cluster that results from merging the two clusters C and C'.

Derivation:

$$\textit{ESS}(C) = \sum_{u \in C} ||\bar{u} - u||^2 \ = \ \sum_{u \in C} \Big(||\bar{u}||^2 - 2 \cdot \langle u, \bar{u} \rangle + ||u||^2 \Big)$$

$$= |C| \cdot ||\bar{u}||^2 - 2|C| \cdot ||\bar{u}||^2 + \sum_{u \in C} ||u||^2 \ = \ \sum_{u \in C} ||u||^2 \ - |C| \cdot ||\bar{u}||^2$$

$$\textit{ESS}(C') = \sum_{v \in C'} ||v||^2 \ - |C'| \cdot ||\bar{v}||^2$$

Hierarchical Cluster Analysis

Ward Criterion

Ward is a variance criterion. It is the (double) increase of the error sum of squares, *ESS*, in the new cluster that results from merging the two clusters C and C'.

Derivation:

$$\mathit{ESS}(C) = \sum_{u \in C} ||\bar{u} - u||^2 \ = \sum_{u \in C} \Big(||\bar{u}||^2 - 2 \cdot \langle u, \bar{u} \rangle + ||u||^2 \Big)$$

$$= |C| \cdot ||\bar{u}||^2 - 2|C| \cdot ||\bar{u}||^2 + \sum_{u \in C} ||u||^2 \ = \sum_{u \in C} ||u||^2 \ - |C| \cdot ||\bar{u}||^2$$

$$\mathit{ESS}(C') = \sum_{v \in C'} ||v||^2 \ - |C'| \cdot ||\bar{v}||^2$$

$$\mathit{ESS}(C \cup C') = \sum_{w \in (C \cup C')} ||w||^2 \ - |C \cup C'| \cdot ||\bar{w}||^2, \quad \text{where } \ \bar{w} = \frac{|C| \cdot \bar{u} + |C'| \cdot \bar{v}}{|C| + |C'|}$$

Hierarchical Cluster Analysis

Ward Criterion

Ward is a variance criterion. It is the (double) increase of the error sum of squares, *ESS*, in the new cluster that results from merging the two clusters C and C'.

Derivation:

$$\mathit{ESS}(C) = \sum_{u \in C} ||\bar{u} - u||^2 \;=\; \sum_{u \in C} \left(||\bar{u}||^2 - 2 \cdot \langle u, \bar{u} \rangle + ||u||^2 \right)$$

$$= |C| \cdot ||\bar{u}||^2 - 2|C| \cdot ||\bar{u}||^2 + \sum_{u \in C} ||u||^2 \;=\; \sum_{u \in C} ||u||^2 \;-\; |C| \cdot ||\bar{u}||^2$$

$$\mathit{ESS}(C') = \sum_{v \in C'} ||v||^2 \;-\; |C'| \cdot ||\bar{v}||^2$$

$$\mathit{ESS}(C \cup C') = \sum_{w \in (C \cup C')} ||w||^2 \;-\; |C \cup C'| \cdot ||\bar{w}||^2, \quad \text{where} \;\; \bar{w} = \frac{|C| \cdot \bar{u} + |C'| \cdot \bar{v}}{|C| + |C'|}$$

$$\mathit{ESS}(C \cup C') - \mathit{ESS}(C) - \mathit{ESS}(C') \;=\; \ldots \;=\; \frac{|C| \cdot |C'|}{|C| + |C'|} \cdot ||\bar{u} - \bar{v}||^2$$

$\bar{u}$ and $\bar{v}$ denote the mean of the points $u \in C$ and $v \in C'$ respectively.

Hierarchical Cluster Analysis

Update Formula for Cluster Distances

After merging two clusters C and C' into a single new cluster, the resulting distances to other the clusters C_i, $d_{\mathcal{C}}(C \cup C', C_i)$, have to be computed.

By exploiting the already computed distances, the Lance-Williams update formula provides an efficient means (linear time in the current number of clusters) to obtain the desired new distances:

$$\begin{aligned} d_{\mathcal{C}}(C \cup C', C_i) = \; & \alpha \cdot d_{\mathcal{C}}(C, C_i) \quad + \\ & \beta \cdot d_{\mathcal{C}}(C', C_i) \;\; + \\ & \gamma \cdot d_{\mathcal{C}}(C, C') \quad + \\ & \delta \cdot |d_{\mathcal{C}}(C, C_i) - d_{\mathcal{C}}(C', C_i)| \end{aligned}$$

The constants $\alpha, \beta, \gamma, \delta$ are specific for single link, complete link, average link, and the ward criterion. The constants are derived on the basis of the respective computation rules for $d_{\mathcal{C}}$.

Hierarchical Cluster Analysis

Update Formula for Cluster Distances (continued)

After merging two clusters C and C' into a single new cluster, the resulting distances to other the clusters C_i, $d_{\mathcal{C}}(C \cup C', C_i)$, have to be computed.

Derivation of the update formula for single link, where $d_{\mathcal{C}}$ = nearest neighbor:

$$d_{\mathcal{C}}(C \cup C', C_i) \;=\; \min_{\substack{u \in (C \cup C') \\ v \in C_i}} d(u, v)$$ [distance measure]

Hierarchical Cluster Analysis

Update Formula for Cluster Distances (continued)

After merging two clusters C and C' into a single new cluster, the resulting distances to other the clusters C_i, $d_{\mathcal{C}}(C \cup C', C_i)$, have to be computed.

Derivation of the update formula for single link, where $d_{\mathcal{C}}$ = nearest neighbor:

$$d_{\mathcal{C}}(C \cup C', C_i) = \min_{\substack{u \in (C \cup C') \\ v \in C_i}} d(u, v) \qquad \text{[distance measure]}$$

$$= \min\{d_{\mathcal{C}}(C, C_i),\ d_{\mathcal{C}}(C', C_i)\}$$

Hierarchical Cluster Analysis

Update Formula for Cluster Distances (continued)

After merging two clusters C and C' into a single new cluster, the resulting distances to other the clusters C_i, $d_{\mathcal{C}}(C \cup C', C_i)$, have to be computed.

Derivation of the update formula for single link, where $d_{\mathcal{C}}$ = nearest neighbor:

$$\begin{aligned} d_{\mathcal{C}}(C \cup C', C_i) &= \min_{\substack{u \in (C \cup C') \\ v \in C_i}} d(u,v) \qquad \text{[distance measure]} \\ &= \min\{d_{\mathcal{C}}(C, C_i),\ d_{\mathcal{C}}(C', C_i)\} \\ &= 0.5 \cdot \big(d_{\mathcal{C}}(C, C_i) + d_{\mathcal{C}}(C', C_i)\big) - 0.5 \cdot |d_{\mathcal{C}}(C, C_i) - d_{\mathcal{C}}(C', C_i)| \end{aligned}$$

Hierarchical Cluster Analysis

Update Formula for Cluster Distances (continued)

After merging two clusters C and C' into a single new cluster, the resulting distances to other the clusters C_i, $d_{\mathcal{C}}(C \cup C', C_i)$, have to be computed.

Derivation of the update formula for single link, where $d_{\mathcal{C}}$ = nearest neighbor:

$$\begin{aligned}
d_{\mathcal{C}}(C \cup C', C_i) &= \min_{\substack{u \in (C \cup C') \\ v \in C_i}} d(u,v) \qquad \text{[distance measure]} \\
&= \min\{d_{\mathcal{C}}(C, C_i),\ d_{\mathcal{C}}(C', C_i)\} \\
&= 0.5 \cdot \big(d_{\mathcal{C}}(C, C_i) + d_{\mathcal{C}}(C', C_i)\big) - 0.5 \cdot |d_{\mathcal{C}}(C, C_i) - d_{\mathcal{C}}(C', C_i)| \\
&= \underset{\underset{\alpha}{\downarrow}}{0.5} \cdot d_{\mathcal{C}}(C, C_i) + \underset{\underset{\beta}{\downarrow}}{0.5} \cdot d_{\mathcal{C}}(C', C_i) + \underset{\underset{\delta}{\downarrow}}{(-0.5)} \cdot |d_{\mathcal{C}}(C, C_i) - d_{\mathcal{C}}(C', C_i)|
\end{aligned}$$

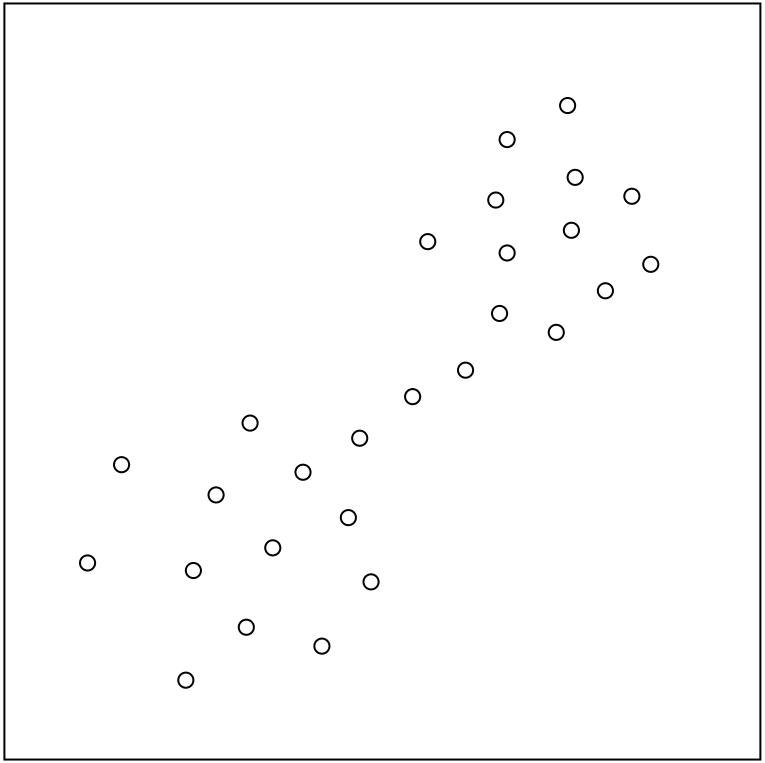
Remarks:

- Link-based algorithms can be used with arbitrary measures for distances and similarities.
- Single link can be operationalized straightforwardly with a minimum spanning tree algorithm.
- Variance-based approaches presume interval-based measurement scales for all features.
- The uniform pseudo code structure of the hierarchical agglomerative algorithm reveals the close relation of the different cluster analysis variants. However, this structural similarity must be regarded with caution: the features' measurement scales along with the point distance computation rule, $d(u, v)$, determine the basic merging characteristics of a cluster analysis algorithm.
- Basic idea of the Lance-Williams update formula: instead of analyzing after a merging step all members (points) of two clusters again, the formula exploits the cluster distances that were already computed in the preceding iteration before the merger.

  How large is the runtime improvement compared to a naive approach that exploits only the distance information in $G = \langle V, E, w \rangle$?

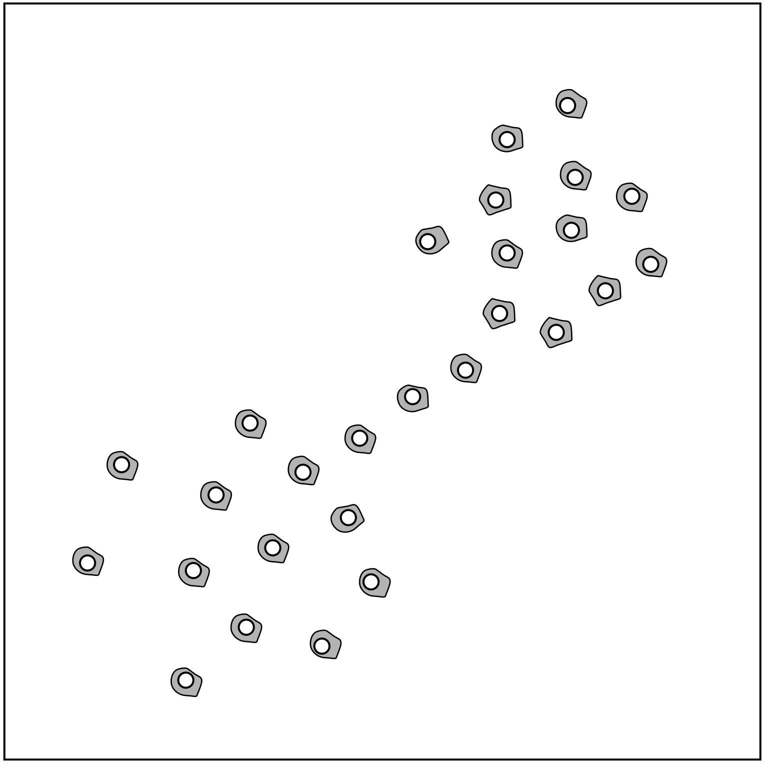
Hierarchical Cluster Analysis

Chaining Problem of Single Link ($d_{\mathcal{C}}$ = Nearest Neighbor)



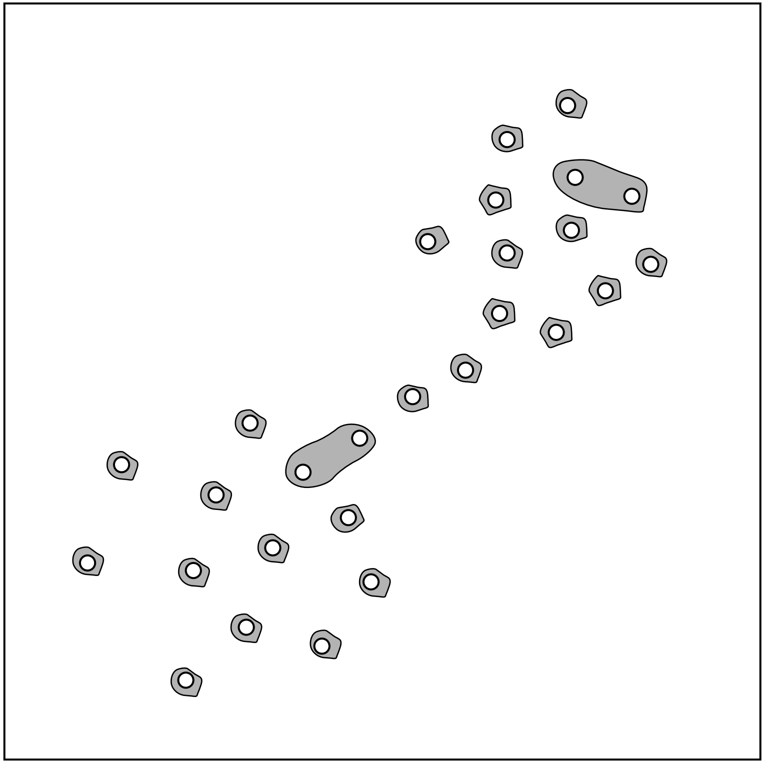
Hierarchical Cluster Analysis

Chaining Problem of Single Link ($d_{\mathcal{C}}$ = Nearest Neighbor)



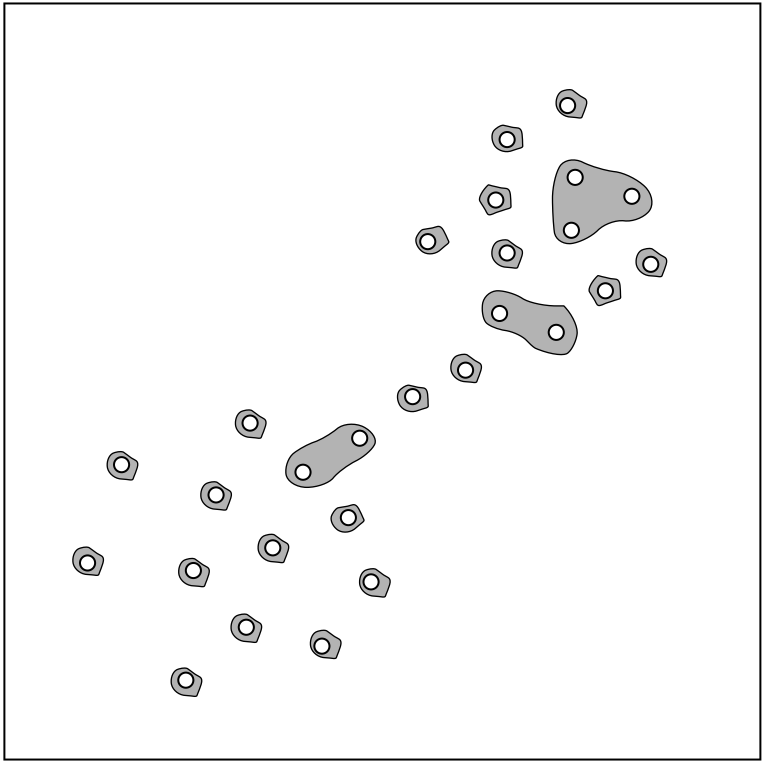
Hierarchical Cluster Analysis

Chaining Problem of Single Link ($d_{\mathcal{C}}$ = Nearest Neighbor)



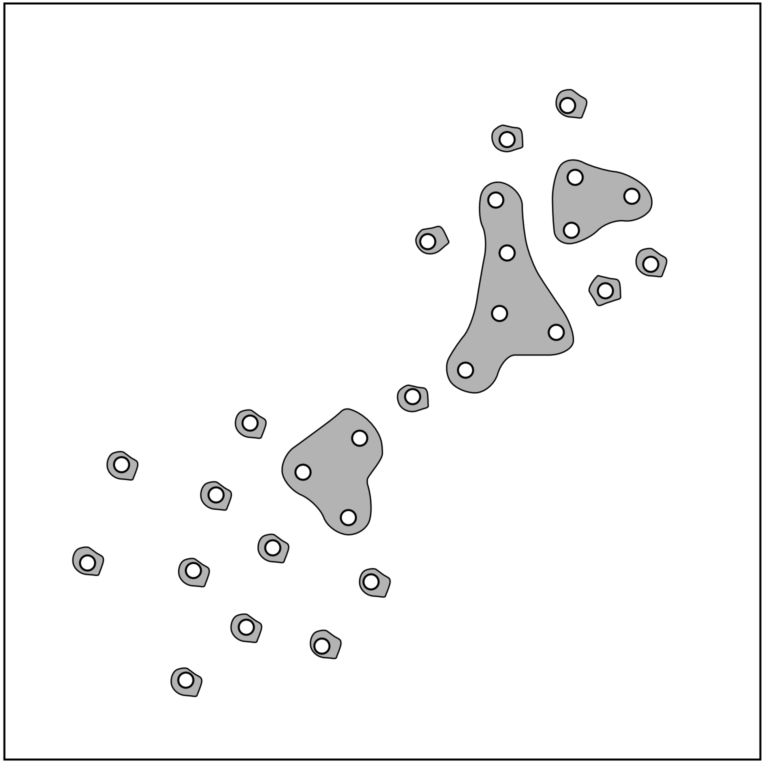
Hierarchical Cluster Analysis

Chaining Problem of Single Link ($d_{\mathcal{C}}$ = Nearest Neighbor)



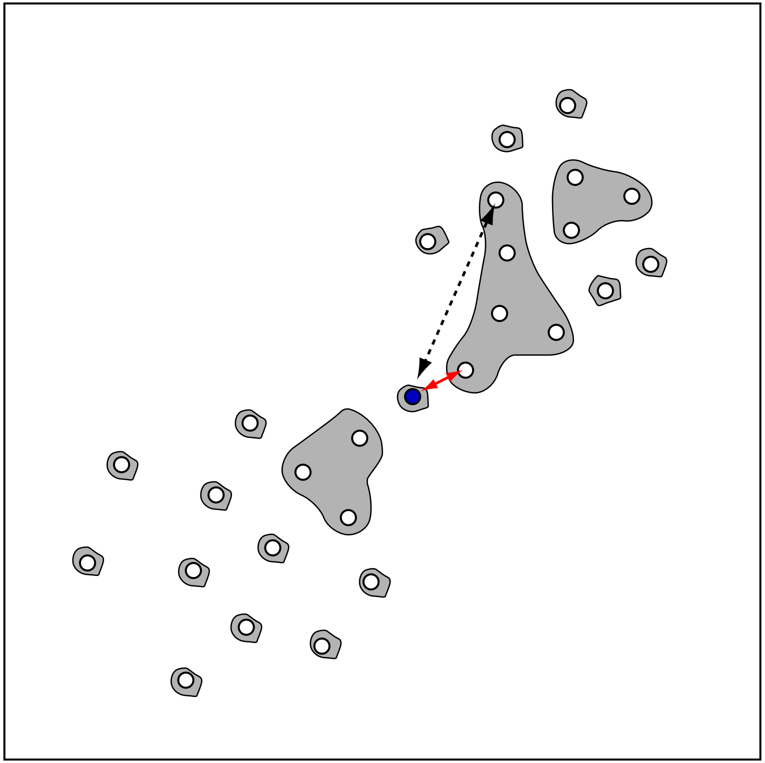
Hierarchical Cluster Analysis

Chaining Problem of Single Link ($d_{\mathcal{C}}$ = Nearest Neighbor)



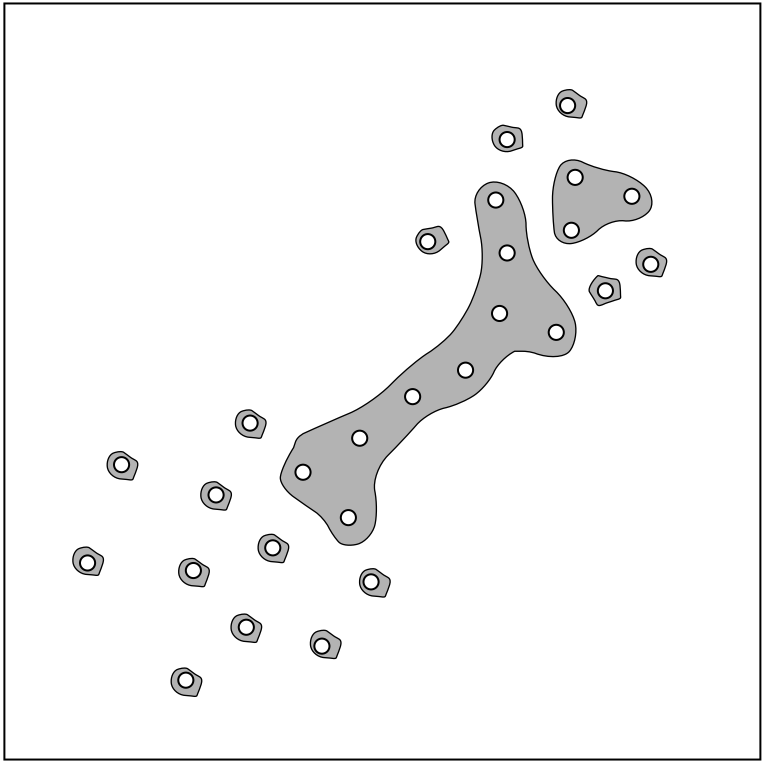
Hierarchical Cluster Analysis

Chaining Problem of Single Link ($d_{\mathcal{C}}$ = Nearest Neighbor) [characteristics]



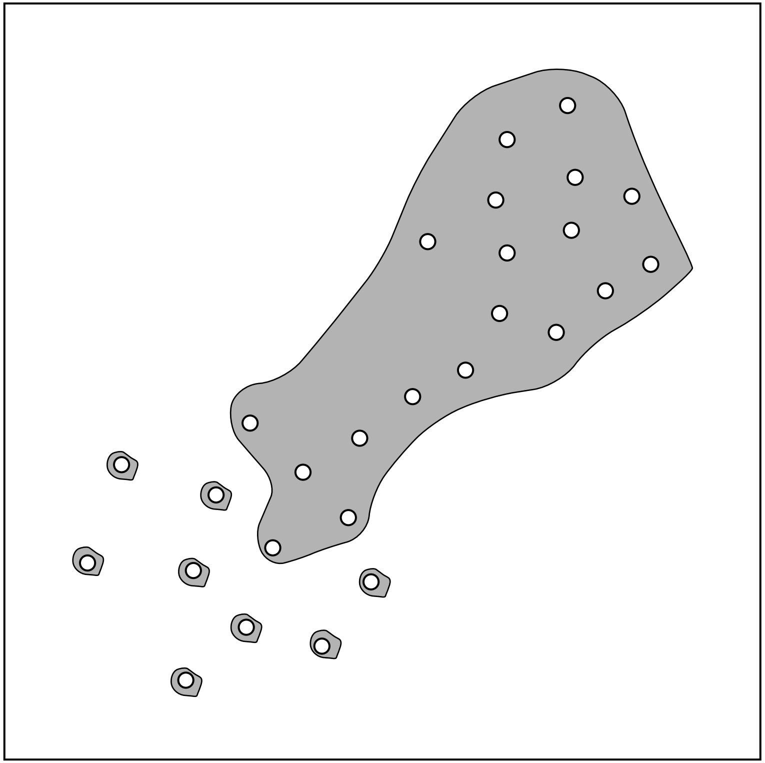
Hierarchical Cluster Analysis

Chaining Problem of Single Link ($d_{\mathcal{C}}$ = Nearest Neighbor)



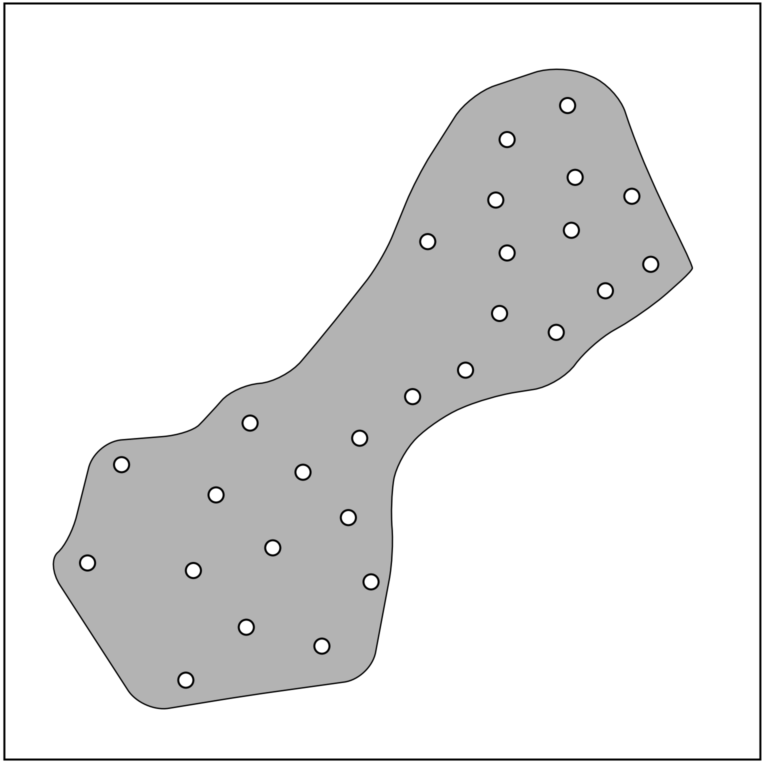
Hierarchical Cluster Analysis

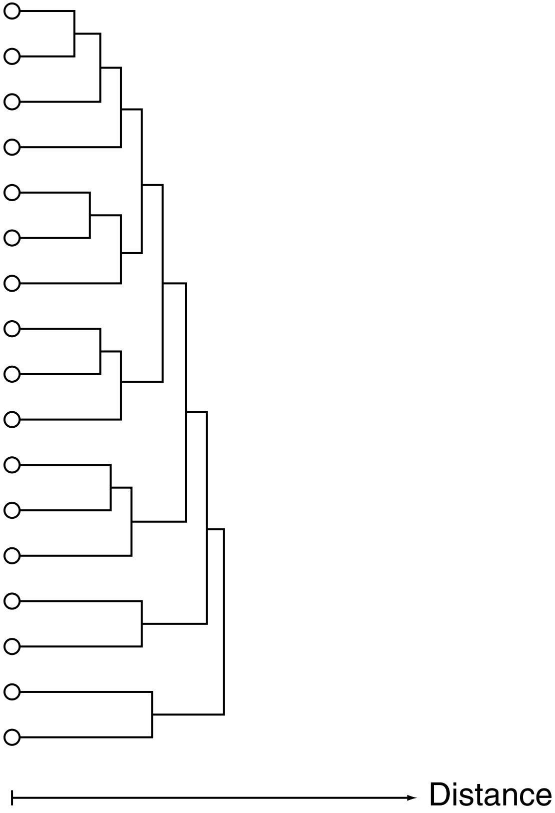
Chaining Problem of Single Link ($d_{\mathcal{C}}$ = Nearest Neighbor)



Hierarchical Cluster Analysis

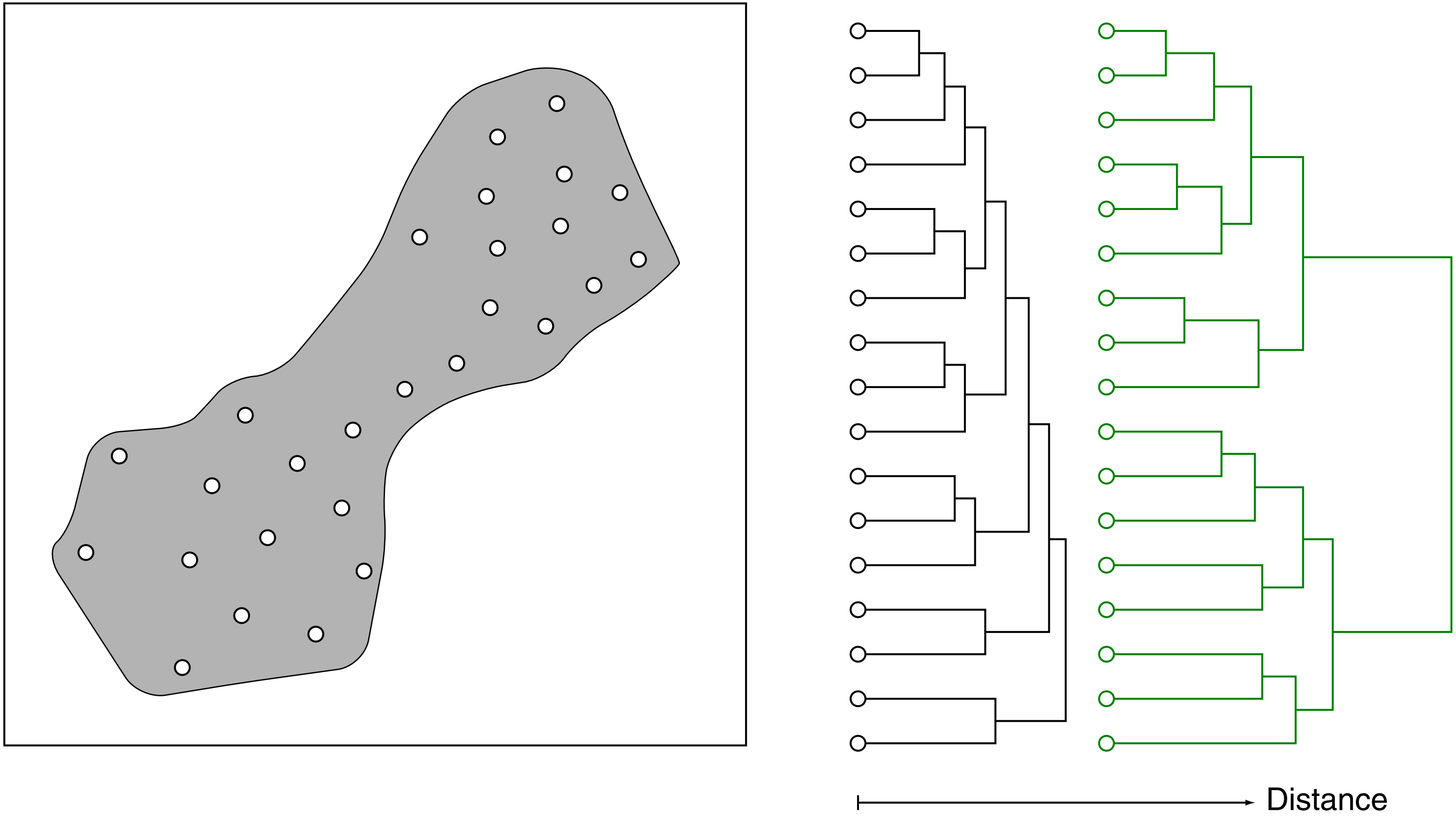
Chaining Problem of Single Link ($d_{\mathcal{C}}$ = Nearest Neighbor)





Hierarchical Cluster Analysis

Chaining Problem of Single Link ($d_{\mathcal{C}}$ = Nearest Neighbor)

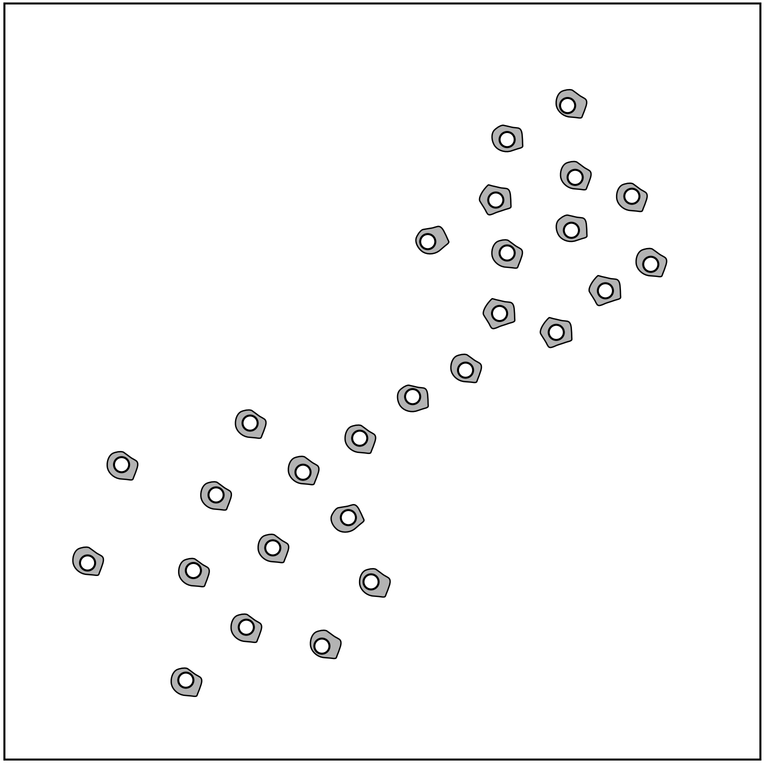


Remarks:

- A k-nearest-neighbor variant may help to mitigate the chaining problem.
- A k-nearest-neighbor variant will prefer larger clusters as agglomeration candidates: larger clusters contain more points and hence are more likely to become a nearest neighbor than smaller clusters.

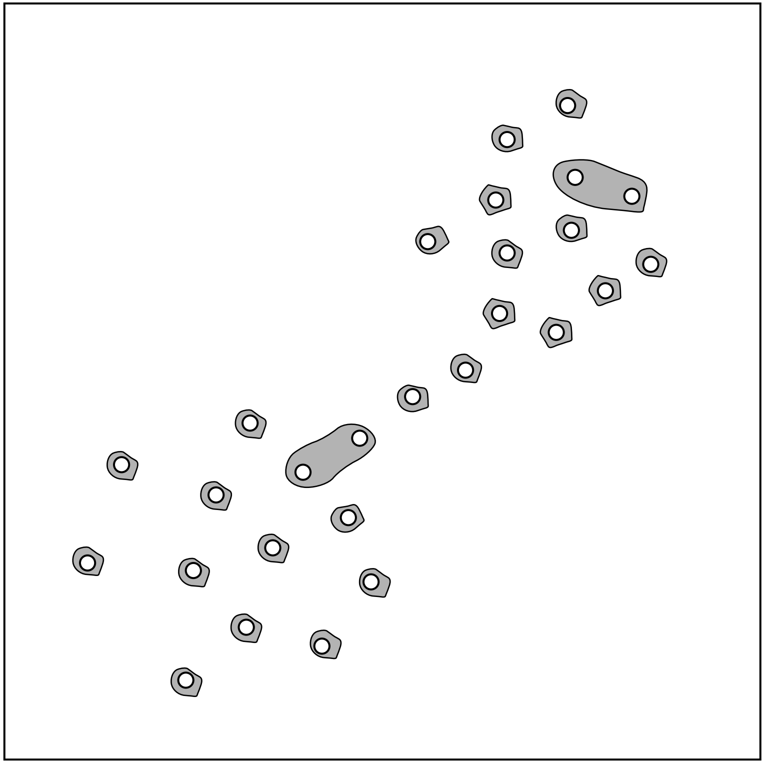
Hierarchical Cluster Analysis

Chaining Problem of Single Link ($d_\mathcal{C}$ = k-Nearest-Neighbor)



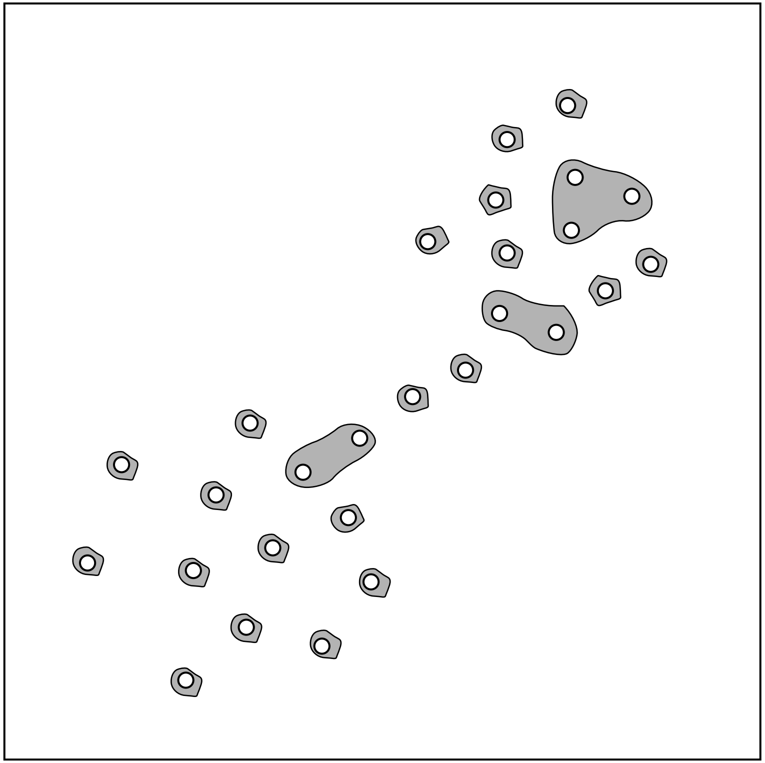
Hierarchical Cluster Analysis

Chaining Problem of Single Link ($d_{\mathcal{C}}$ = k-Nearest-Neighbor)



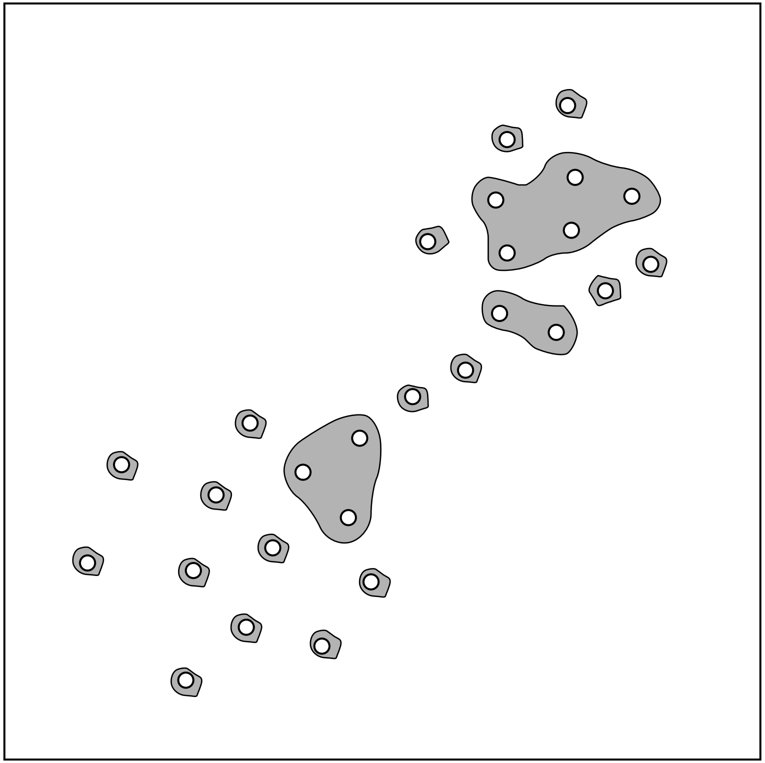
Hierarchical Cluster Analysis

Chaining Problem of Single Link ($d_{\mathcal{C}}$ = k-Nearest-Neighbor)



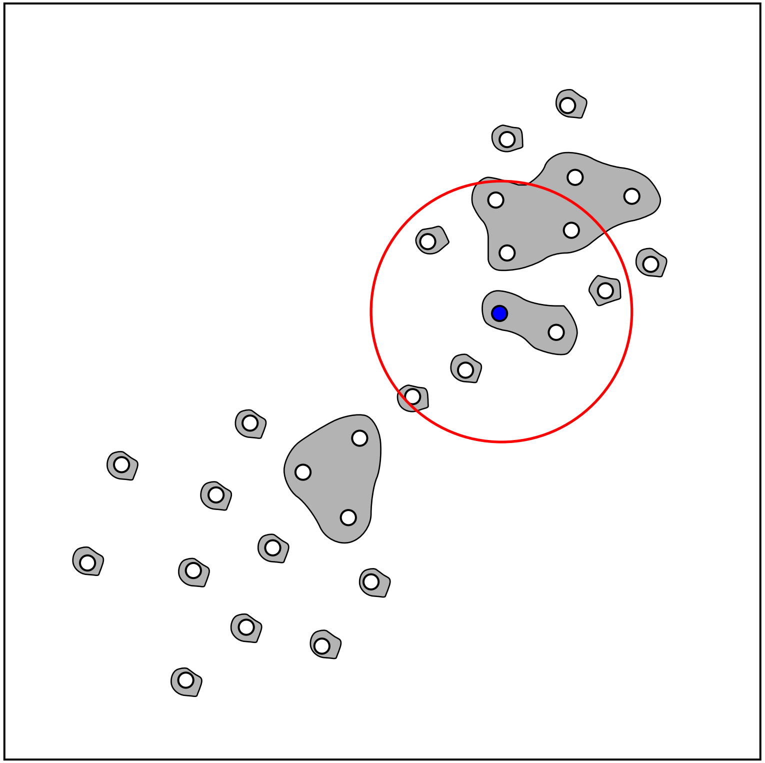
Hierarchical Cluster Analysis

Chaining Problem of Single Link ($d_{\mathcal{C}}$ = k-Nearest-Neighbor)



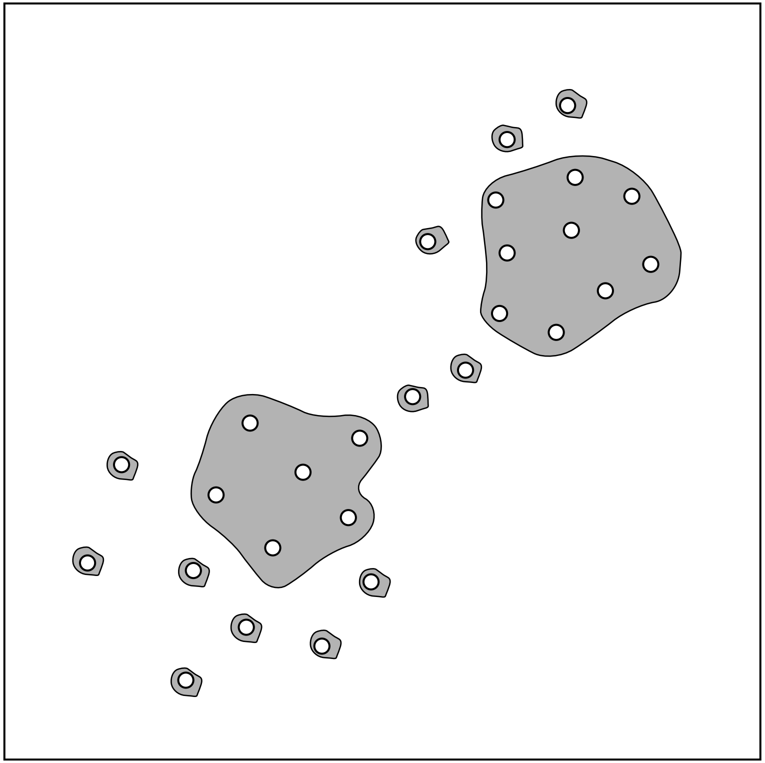
Hierarchical Cluster Analysis

Chaining Problem of Single Link ($d_{\mathcal{C}}$ = k-Nearest-Neighbor)



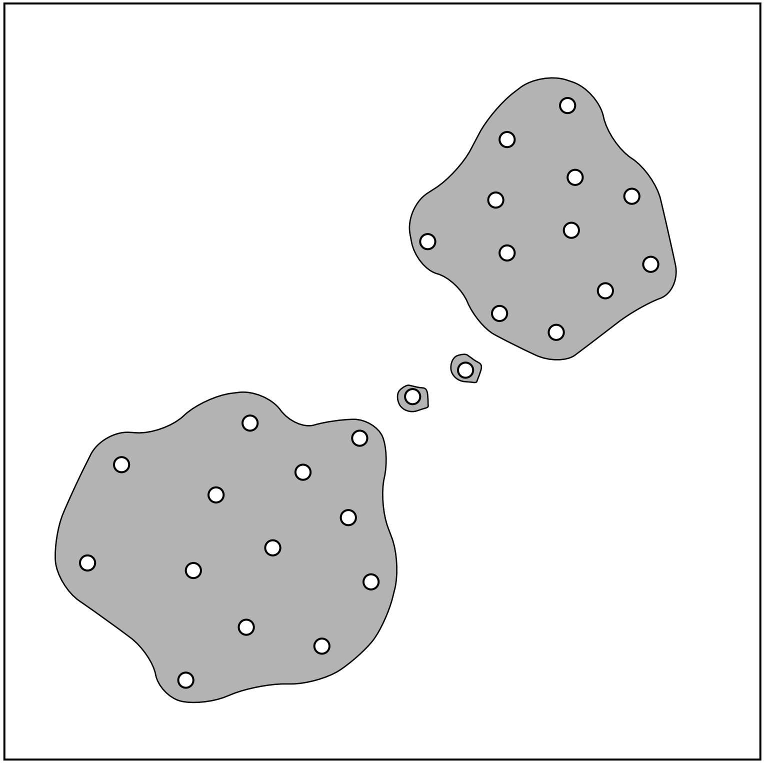
Hierarchical Cluster Analysis

Chaining Problem of Single Link ($d_{\mathcal{C}}$ = k-Nearest-Neighbor)



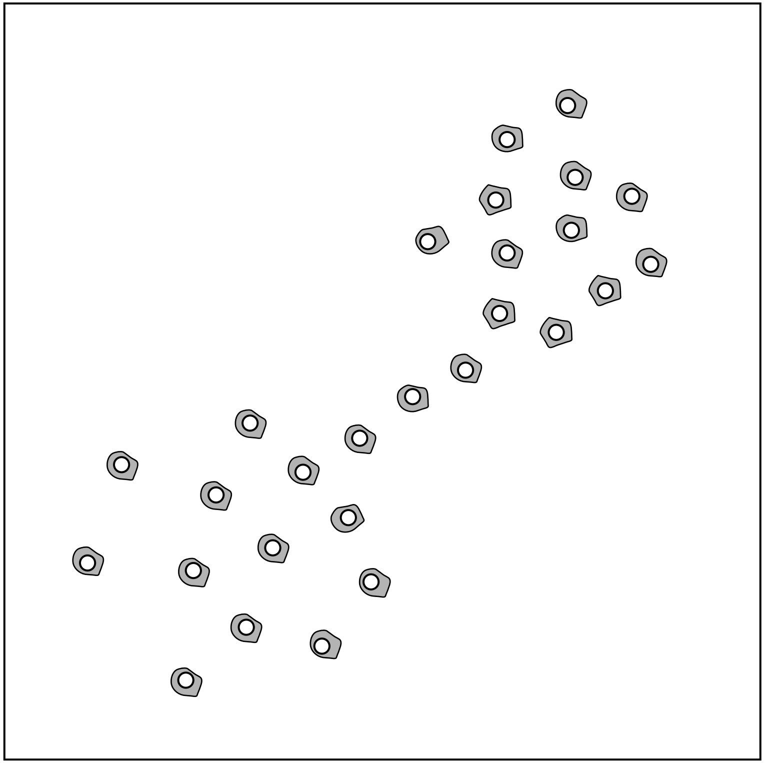
Hierarchical Cluster Analysis

Chaining Problem of Single Link ($d_{\mathcal{C}}$ = k-Nearest-Neighbor)



Hierarchical Cluster Analysis

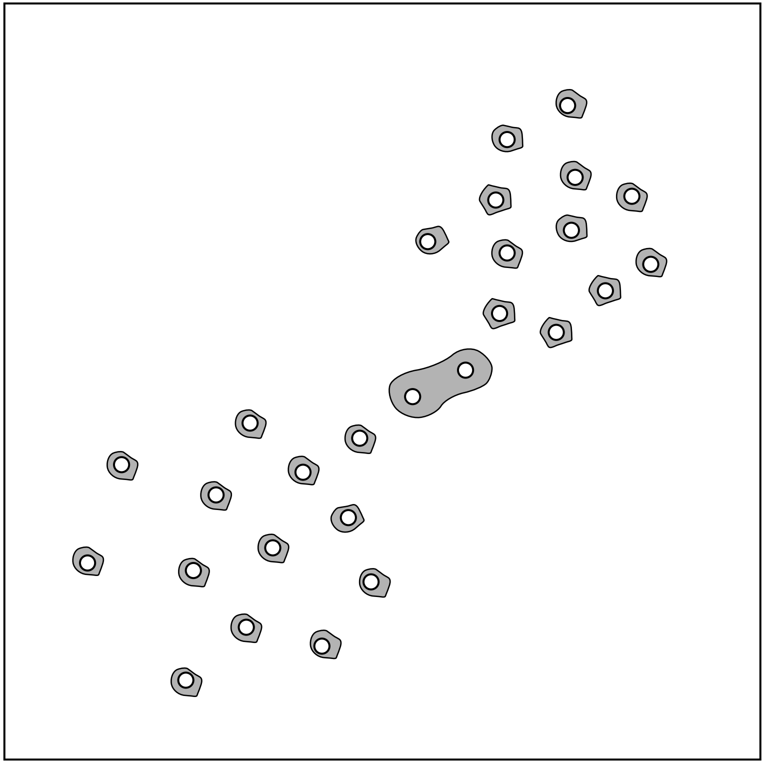
Chaining Problem of Single Link ($d_{\mathcal{C}}$ = k-Nearest-Neighbor)



In certain situations k-nearest-neighbor can fail as well.

Hierarchical Cluster Analysis

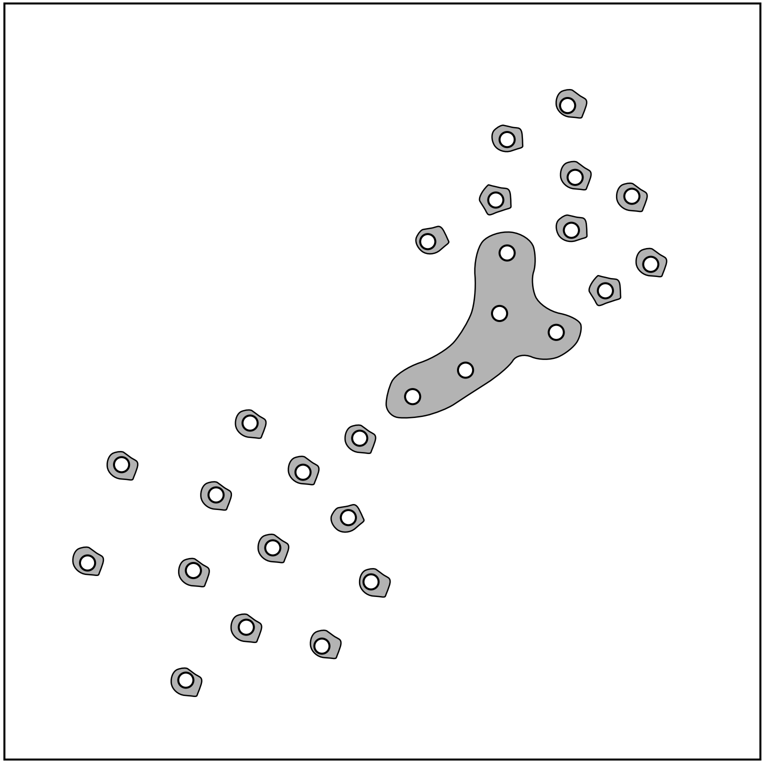
Chaining Problem of Single Link ($d_{\mathcal{C}}$ = k-Nearest-Neighbor)



In certain situations k-nearest-neighbor can fail as well.

Hierarchical Cluster Analysis

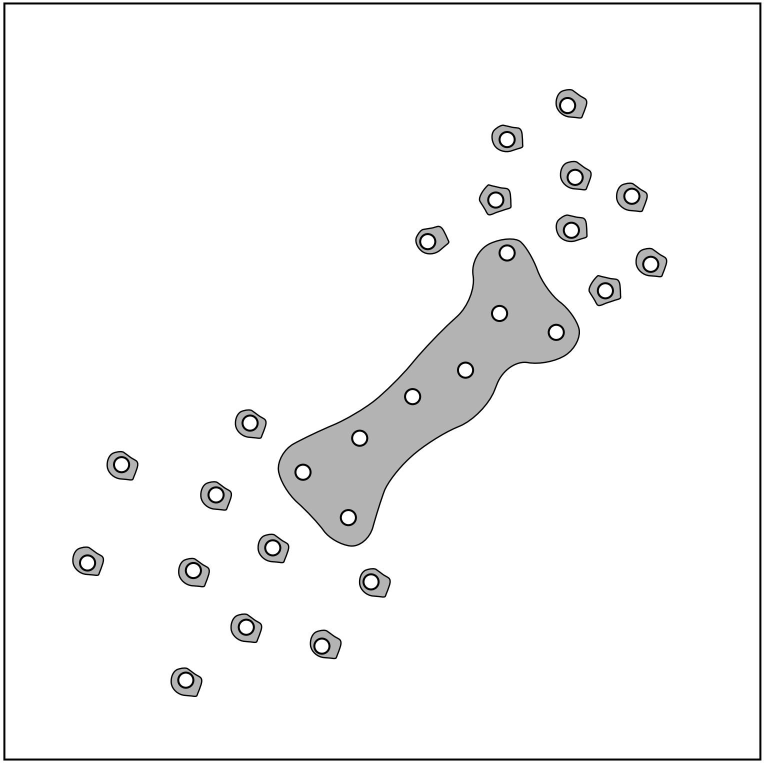
Chaining Problem of Single Link ($d_{\mathcal{C}}$ = k-Nearest-Neighbor)



In certain situations k-nearest-neighbor can fail as well.

Hierarchical Cluster Analysis

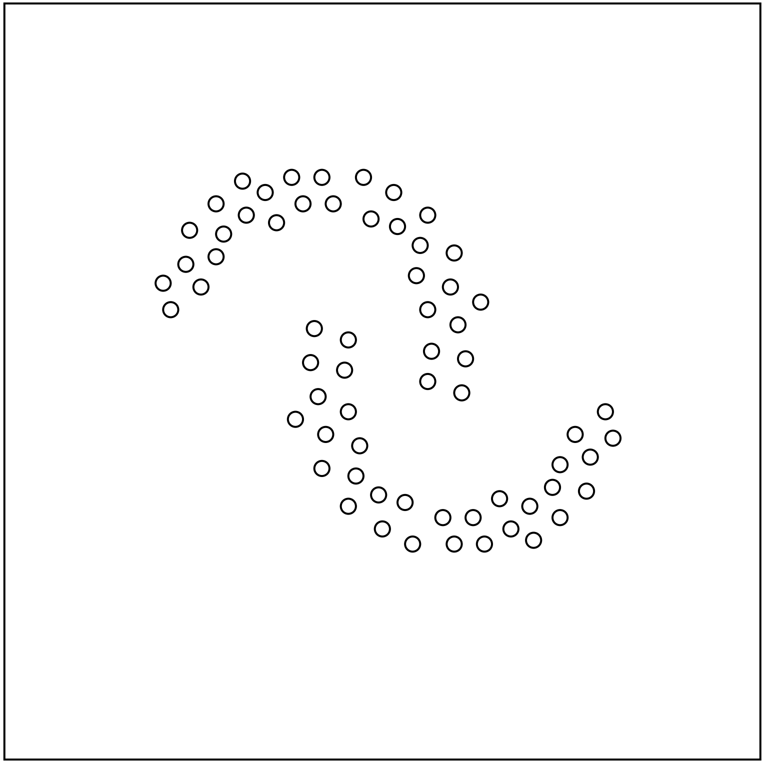
Chaining Problem of Single Link ($d_{\mathcal{C}}$ = k-Nearest-Neighbor)



In certain situations k-nearest-neighbor can fail as well.

Hierarchical Cluster Analysis

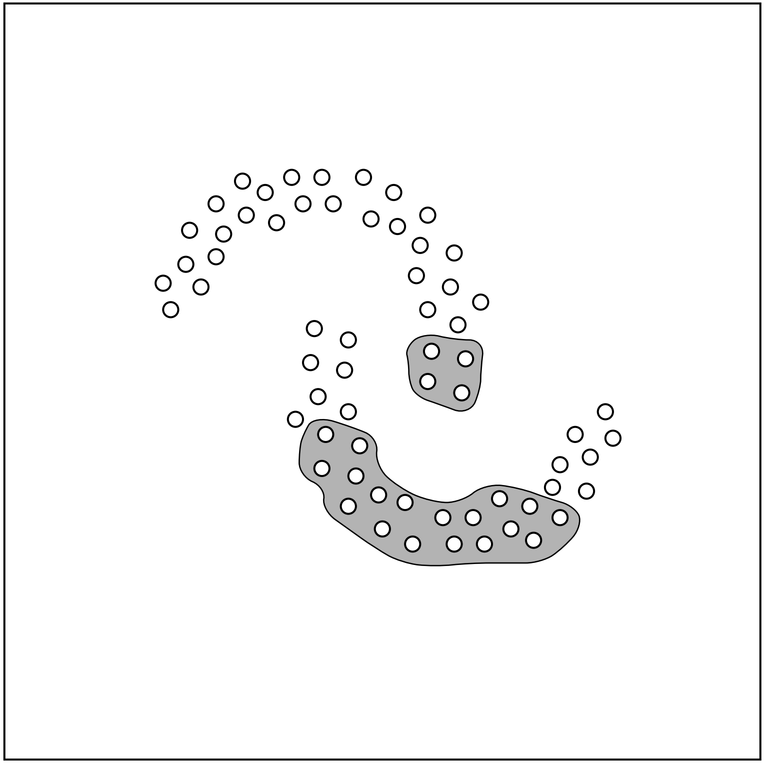
Nesting Problem of Complete Link ($d_{\mathcal{C}}$ = Furthest Neighbor)



Particular pattern recognition tasks or the detection of hyperspheres requires to deal with nested clusters.

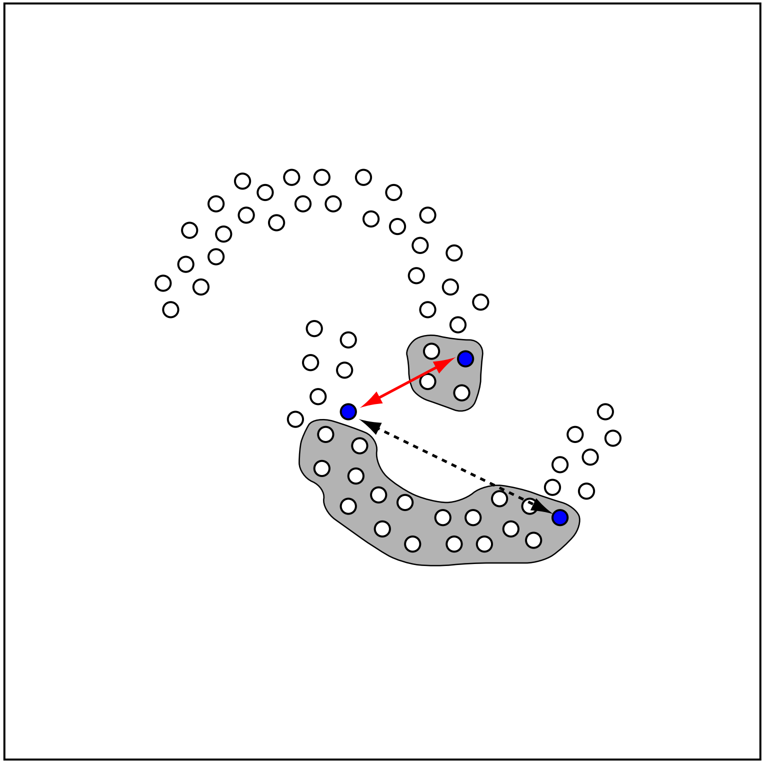
Hierarchical Cluster Analysis

Nesting Problem of Complete Link ($d_{\mathcal{C}}$ = Furthest Neighbor)



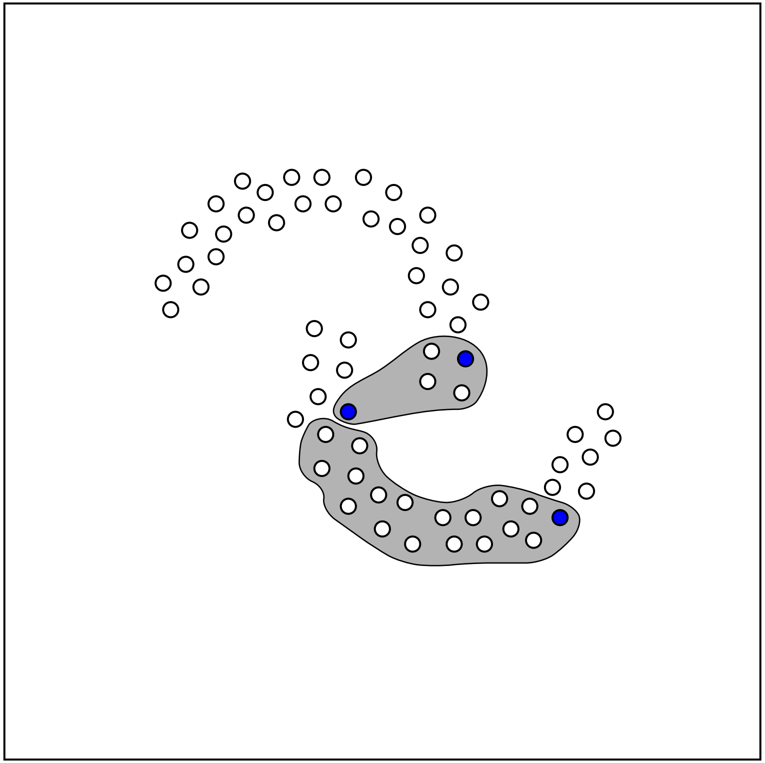
Hierarchical Cluster Analysis

Nesting Problem of Complete Link ($d_{\mathcal{C}}$ = Furthest Neighbor) [characteristics]



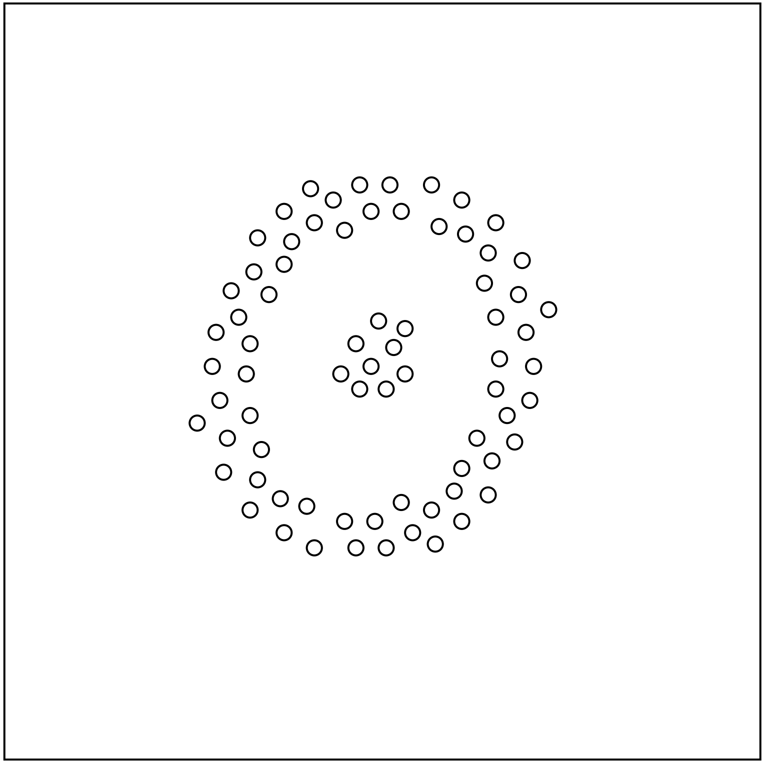
Hierarchical Cluster Analysis

Nesting Problem of Complete Link ($d_{\mathcal{C}}$ = Furthest Neighbor)



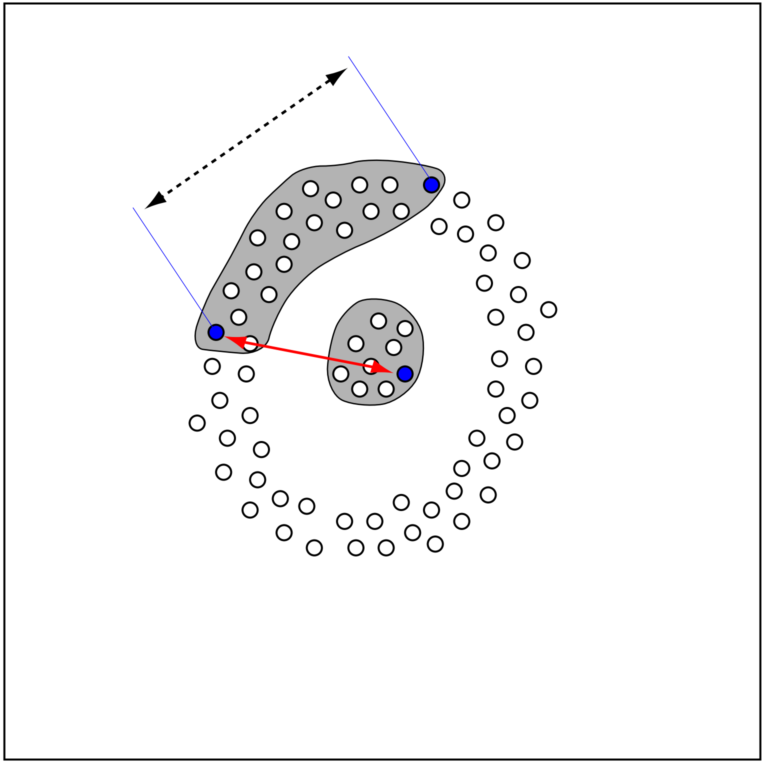
Hierarchical Cluster Analysis

Nesting Problem of Complete Link ($d_{\mathcal{C}}$ = Furthest Neighbor)



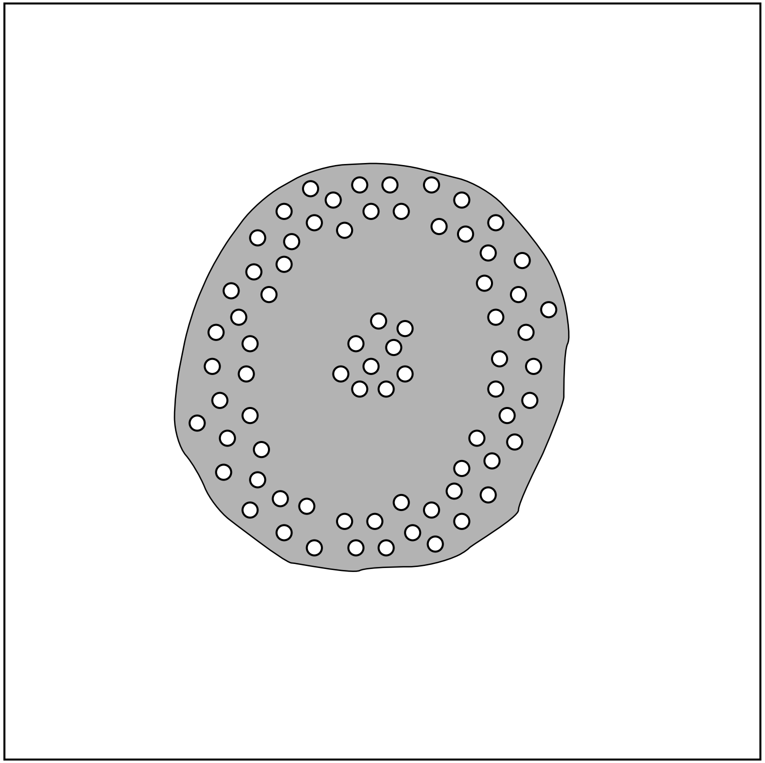
Hierarchical Cluster Analysis

Nesting Problem of Complete Link ($d_{\mathcal{C}}$ = Furthest Neighbor) [characteristics]



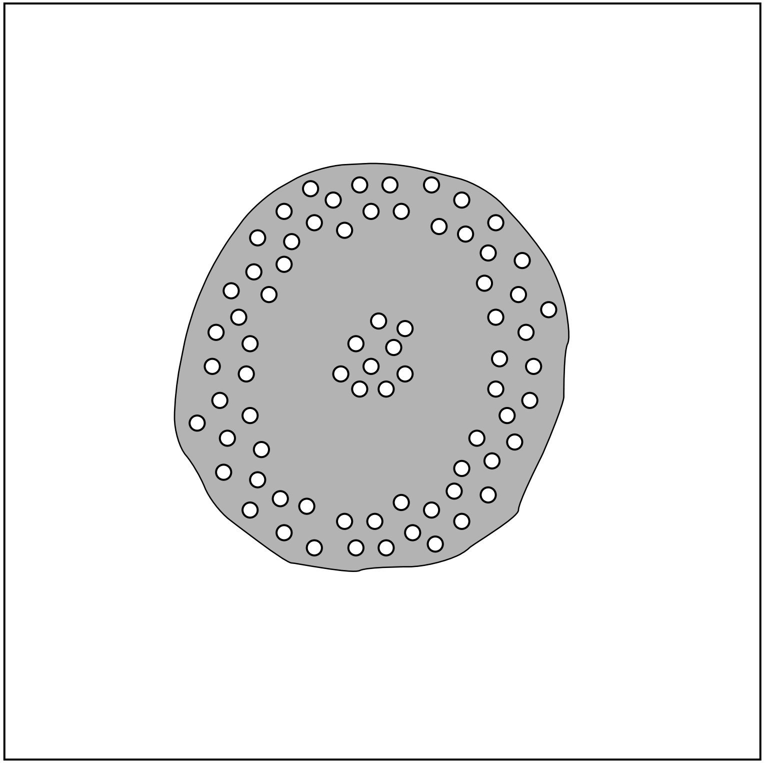
Hierarchical Cluster Analysis

Nesting Problem of Complete Link ($d_{\mathcal{C}}$ = Furthest Neighbor)

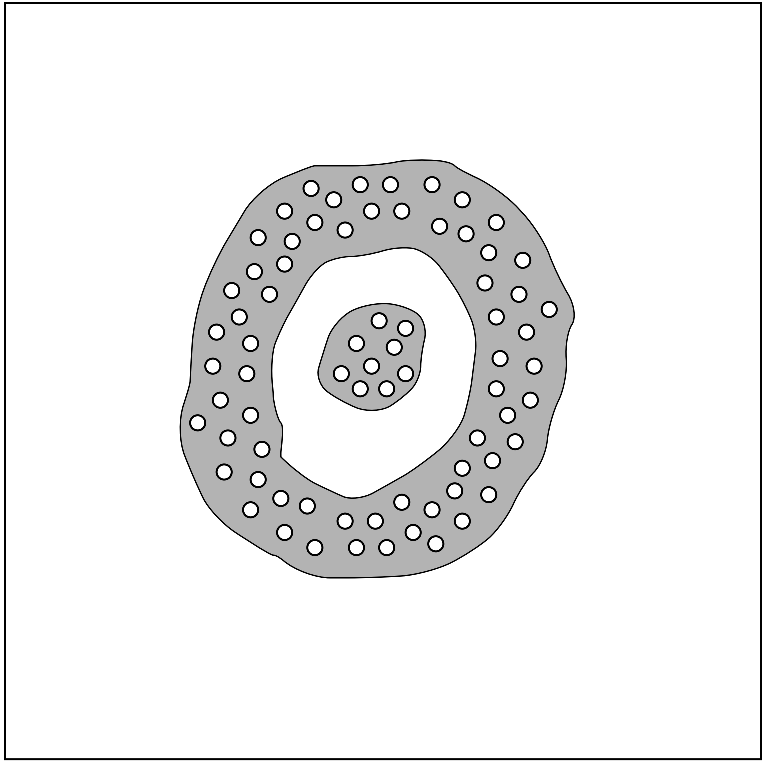


Hierarchical Cluster Analysis

Nesting Problem of Complete Link ($d_{\mathcal{C}}$ = Furthest Neighbor)



Reality



Wish

Hierarchical Cluster Analysis

Characteristics of Hierarchical Agglomerative Algorithms [distance measures]

Geometrical characteristics:

	single link	complete link	average link	Ward criterion
characteristic	contractive:	dilating:	conservative:	conservative:
cluster number	low	high	medium	medium
cluster form	extended	small	compact	spherical
chaining tendency	strong	low	low	low
outlier-detecting	very good	poor	medium	medium

Hierarchical Cluster Analysis

Characteristics of Hierarchical Agglomerative Algorithms [distance measures]

Geometrical characteristics:

	single link	complete link	average link	Ward criterion
characteristic	contractive:	dilating:	conservative:	conservative:
cluster number	low	high	medium	medium
cluster form	extended	small	compact	spherical
chaining tendency	strong	low	low	low
outlier-detecting	very good	poor	medium	medium

Data-related characteristics:

noisy data	susceptible	susceptible	unaffected	unaffected
feature transformation	invariant	invariant	–	–

Hierarchical Cluster Analysis

Characteristics of Hierarchical Agglomerative Algorithms [distance measures]

Geometrical characteristics:

	single link	complete link	average link	Ward criterion
characteristic	contractive:	dilating:	conservative:	conservative:
cluster number	low	high	medium	medium
cluster form	extended	small	compact	spherical
chaining tendency	strong	low	low	low
outlier-detecting	very good	poor	medium	medium

Data-related characteristics:

noisy data	susceptible	susceptible	unaffected	unaffected
feature transformation	invariant	invariant	–	–

Characteristics of the cluster distance measure $d_{\mathcal{C}}$:

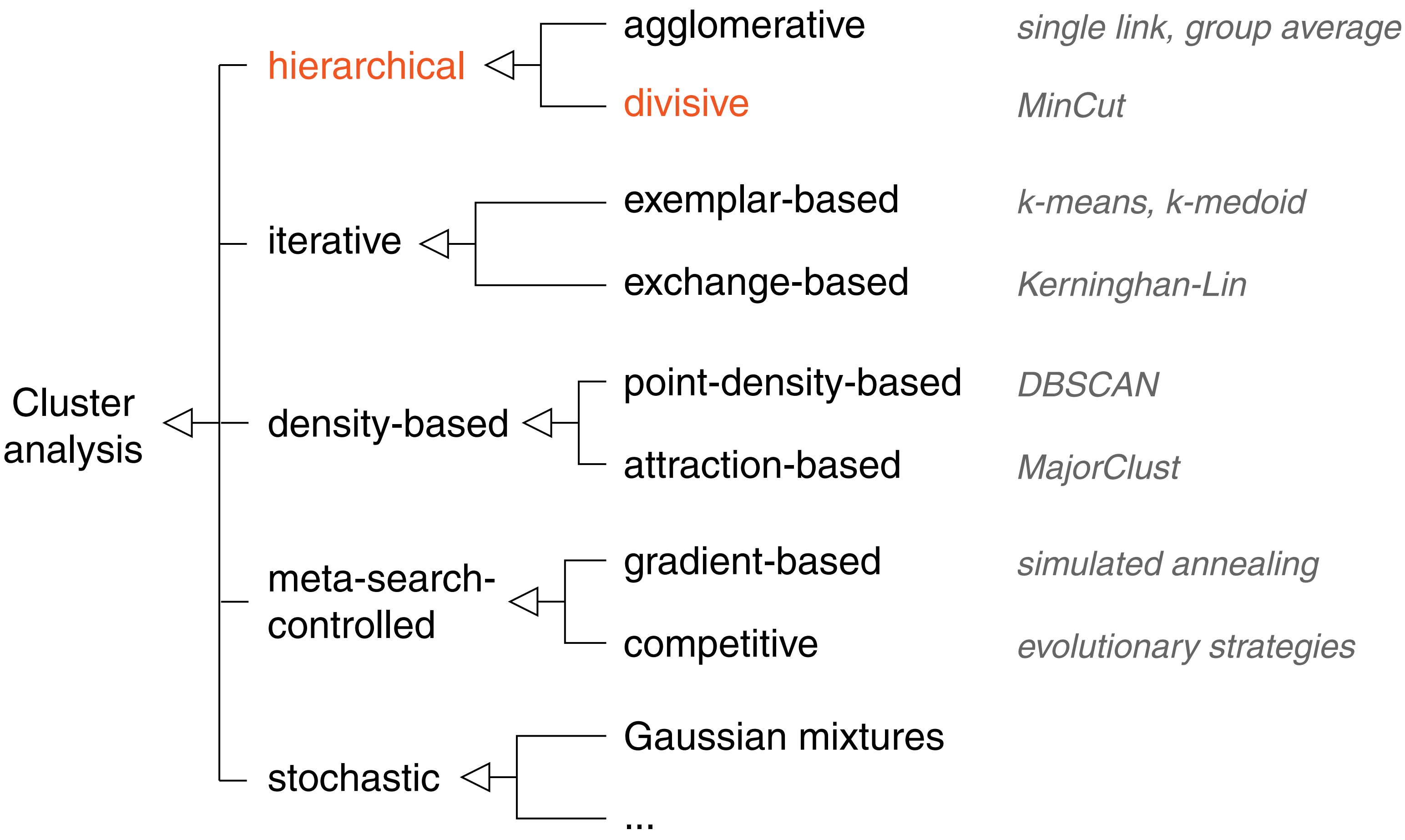
monotonicity	✓	✓	✓	✓
order dependence	✓	✓	✓	✓
consistency	$\longrightarrow 0$	$\longrightarrow \infty$	✓	$\longrightarrow \infty$

Remarks:

- The previous table also shows the usage frequency of the algorithms: single link and complete link are the most popular hierarchical agglomerative algorithms.
  [Jain/Murty/Flynn 1999]
- The Ward criterion has been well-proven for cluster of equal sizes.
- Average link prefers spherical cluster forms, but it will also be able to detect “potato-shaped” clusters. [Kaufman/Rousseeuw 1990, p.47]
- Chaining will also happen when the median distance is employed.
- The median distance and is not a monotonic cluster distance measure.
  [Kaufman/Rousseeuw 1990, pp. 205+240]

Hierarchical Cluster Analysis

Merging Principles



Cluster analysis

hierarchical
- agglomerative — *single link, group average*
- divisive — *MinCut*

iterative
- exemplar-based — *k-means, k-medoid*
- exchange-based — *Kerninghan-Lin*

density-based
- point-density-based — *DBSCAN*
- attraction-based — *MajorClust*

meta-search-controlled
- gradient-based — *simulated annealing*
- competitive — *evolutionary strategies*

stochastic
- Gaussian mixtures
- ...

Hierarchical Cluster Analysis

Hierarchical Divisive Algorithm

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

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

```
$\mathcal{C} = \{V\}$ // initial clustering

WHILE $\exists C_x : (C_x \in \mathcal{C} \;\wedge\; |C_x| > 1)$ DO
    $\{C, C'\} = \underset{\substack{\{C_i, C_j\}: \\ C_i \cup C_j = C_x \,\wedge\, C_i \cap C_j = \emptyset}}{\text{argmax}} \; d_{\mathcal{C}}(C_i, C_j)$
    $\mathcal{C} = (\mathcal{C} \setminus \{C_x\}) \;\cup\; \{C, C'\}$ // splitting

ENDDO
RETURN$(T)$
```

Compare the above algorithm to the hierarchical agglomerative algorithm.

Hierarchical Cluster Analysis

Hierarchical Divisive Algorithm

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

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

```
1. $\mathcal{C} = \{V\}$ // initial clustering
2. $V_T = \{v_C \mid C \in \mathcal{C}\}$, $E_T = \emptyset$ // initial dendrogram
3. WHILE $\exists C_x : (C_x \in \mathcal{C} \;\wedge\; |C_x| > 1)$ DO
4.     $\{C, C'\} = \underset{\substack{\{C_i, C_j\}: \\ C_i \cup C_j = C_x \wedge C_i \cap C_j = \emptyset}}{\text{argmax}} \; d_{\mathcal{C}}(C_i, C_j)$
5.     $\mathcal{C} = (\mathcal{C} \setminus \{C_x\}) \;\cup\; \{C, C'\}$ // splitting
6.     $V_T = V_T \cup \{v_C, v_{C'}\}$, $E_T = E_T \cup \{\{v_{C_x}, v_C\}, \{v_{C_x}, v_{C'}\}\}$ // dendrogram
7. ENDDO
8. RETURN$(T)$
```

Compare the above algorithm to the hierarchical agglomerative algorithm.

Remarks:

- ❑ The cluster distance measure $d_{\mathcal{C}}$ can be chosen as with the hierarchical agglomerative algorithms. However, the worst-case complexity is exponential instead of quadratic.

- ❑ Hierarchical divisive algorithm are often designed according to the *monothetic* paradigm: within each decision step only a single feature is considered.

  The monothetic paradigm is particularly useful for features with ordinal and interval-based measurement scales: instead of considering all possible partitionings, a set of feature vectors is split with regard to a location parameter such as a feature's median or a feature's mean.

- ❑ In contrast to hierarchical agglomerative algorithms, a hierarchical divisive algorithm cannot repair a "wrong" partitioning from a previous iteration.

- ❑ A powerful hierarchical divisive algorithm is based on the *cut weight* of a graph:

$$sim_{\mathcal{C}}(C, C') = \sum_{e \in cut(\{C, C'\})} w(e) \quad \text{or} \quad d_{\mathcal{C}}(C, C') = \frac{1}{sim_{\mathcal{C}}(C, C')}$$

Hierarchical Cluster Analysis

MinCut

Definition 4 (Cut, Minimum Cut)

Let $G = \langle V, E, w \rangle$ be a graph with a non-negative weight function w. Let $U \subset V$ be a non-empty subset of the node set V and let $\bar{U}$ be defined as $\bar{U} = V \setminus U$. Then the cut between U and $\bar{U}$ is defined as follows:

$$cut(\{U, \bar{U}\}) = \{\{u, v\} \mid \{u, v\} \in E, u \in U, v \in \bar{U}\}$$

Hierarchical Cluster Analysis

MinCut

Definition 4 (Cut, Minimum Cut)

Let $G = \langle V, E, w \rangle$ be a graph with a non-negative weight function w. Let $U \subset V$ be a non-empty subset of the node set V and let $\bar{U}$ be defined as $\bar{U} = V \setminus U$. Then the cut between U and $\bar{U}$ is defined as follows:

$$cut(\{U, \bar{U}\}) = \{\{u, v\} \mid \{u, v\} \in E, u \in U, v \in \bar{U}\}$$

The weight (or the capacity) of $cut(\{U, \bar{U}\})$ is defined as follows:

$$w(\{U, \bar{U}\}) = \sum_{e \in cut(\{U, \bar{U}\})} w(e)$$

Hierarchical Cluster Analysis

MinCut

Definition 4 (Cut, Minimum Cut)

Let $G = \langle V, E, w \rangle$ be a graph with a non-negative weight function w. Let $U \subset V$ be a non-empty subset of the node set V and let $\bar{U}$ be defined as $\bar{U} = V \setminus U$. Then the cut between U and $\bar{U}$ is defined as follows:

$$cut(\{U, \bar{U}\}) = \{\{u, v\} \mid \{u, v\} \in E, u \in U, v \in \bar{U}\}$$

The weight (or the capacity) of $cut(\{U, \bar{U}\})$ is defined as follows:

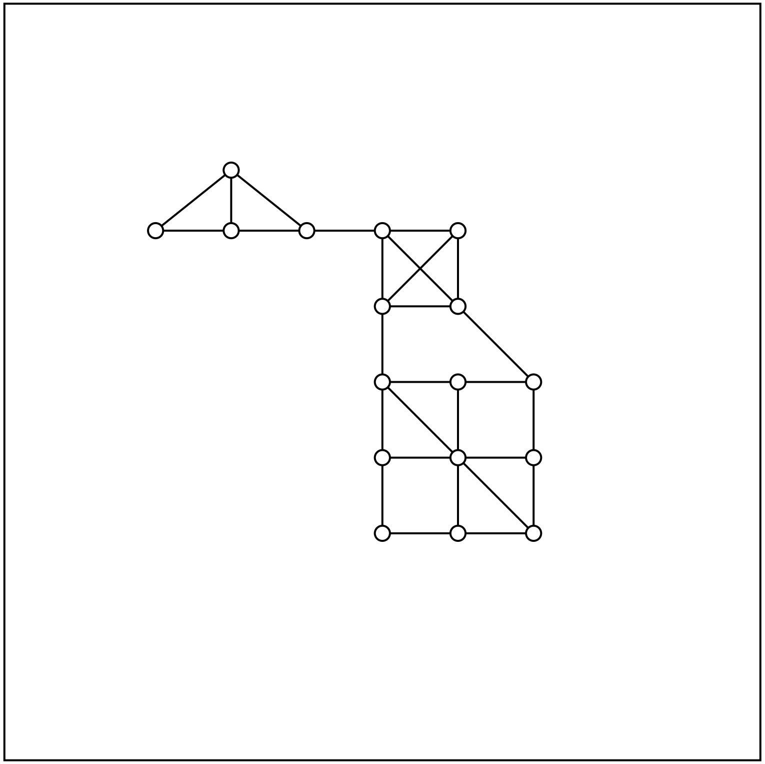
$$w(\{U, \bar{U}\}) = \sum_{e \in cut(\{U, \bar{U}\})} w(e)$$

$cut(\{U, \bar{U}\})$ is called minimum capacity cut of G, iff for all splittings $\{W, \bar{W}\}$, $W, \bar{W} \neq \emptyset$ holds:

$$w(\{U, \bar{U}\}) \ \leq \ w(\{W, \bar{W}\})$$

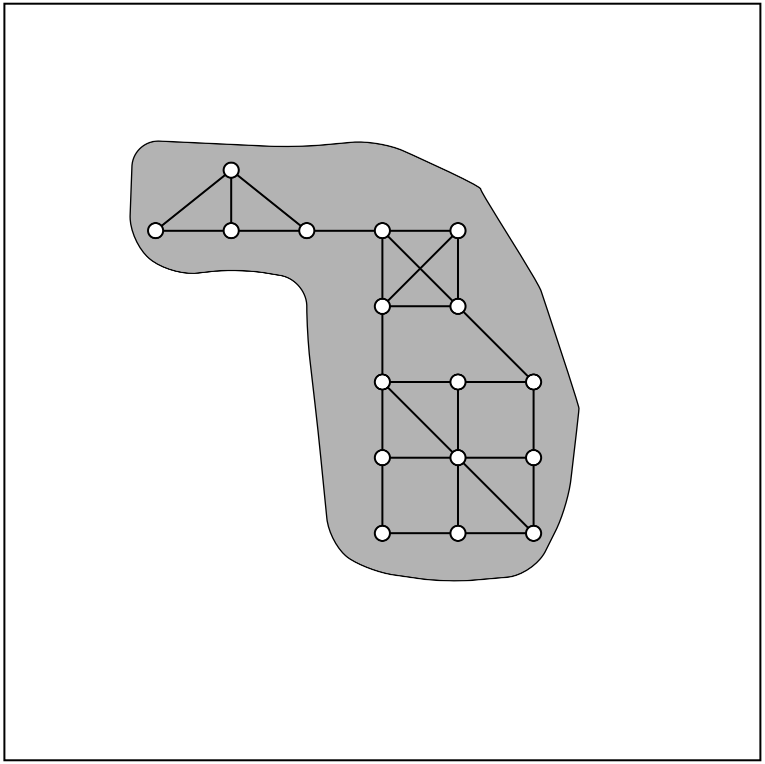
Hierarchical Cluster Analysis

MinCut



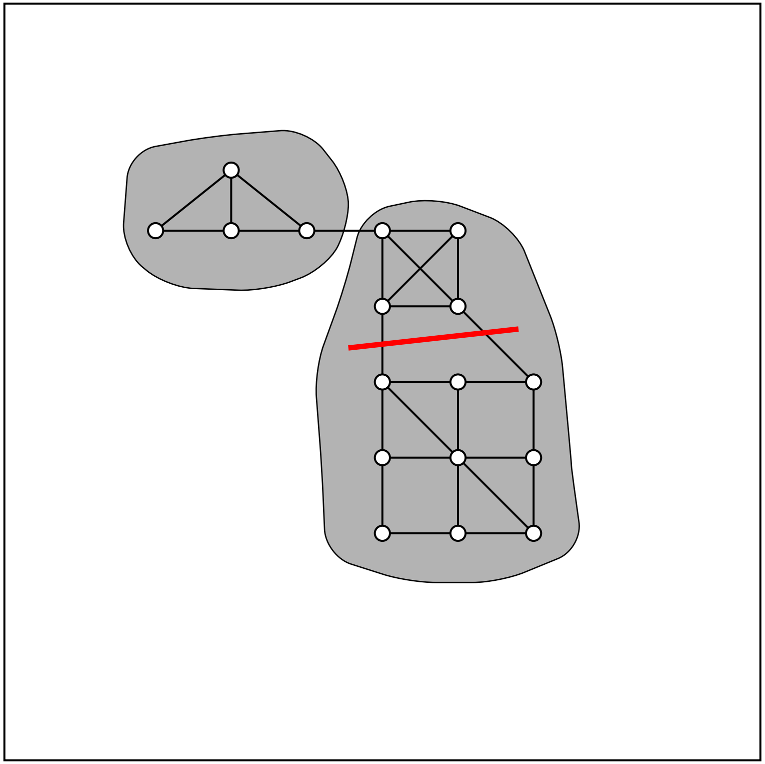
Hierarchical Cluster Analysis

MinCut



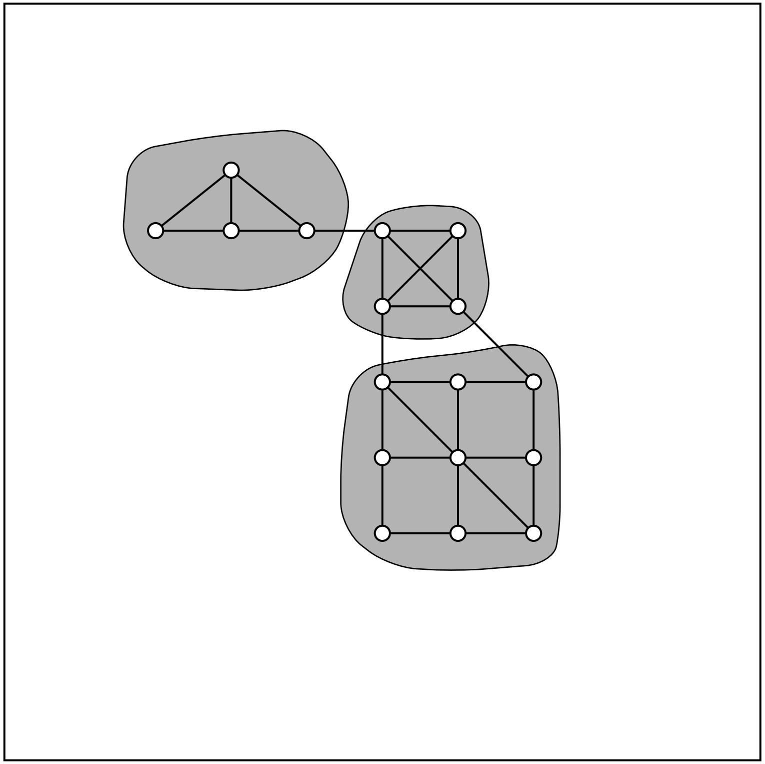
Hierarchical Cluster Analysis

MinCut



Hierarchical Cluster Analysis

MinCut

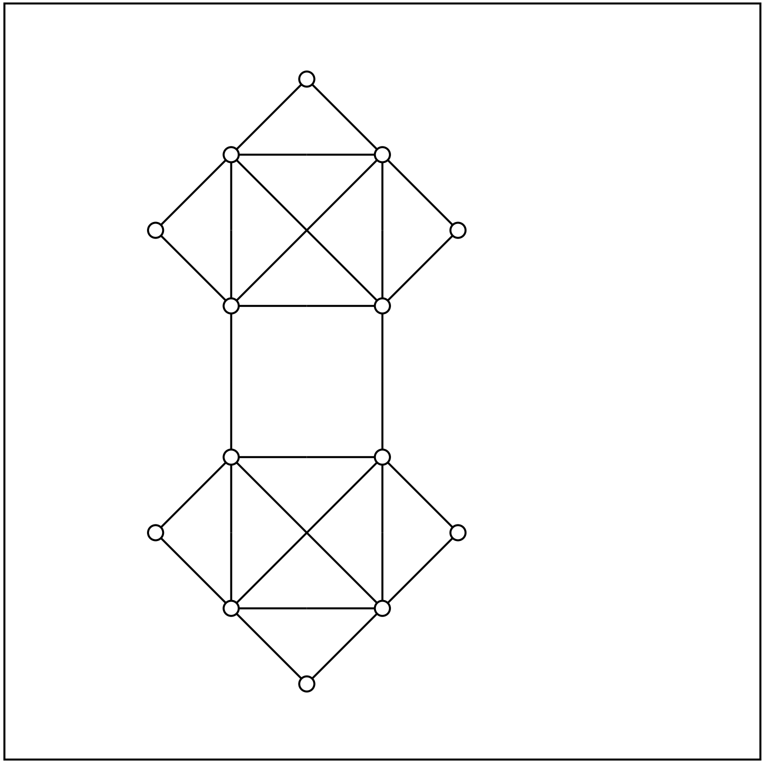


Remarks:

- ❑ Each partitioning requires the computation of a minimum capacity cut. Note that no node is labeled as source or sink.
- ❑ The runtime complexity of the best known algorithm for the computation of a minimum capacity cut is in $O(|V| \cdot |E| + |V|^2 \cdot \log |V|)$. [Nagamochi/Ono/Ibaraki 1994]
- ❑ $|V| - 1$ computations of a minimum capacity cut are necessary to obtain a complete partitioning (= one node per cluster).
- ❑ The effort for the computation of a minimum s-t-cut, i.e., a cut that considers a source s and a sink t, is in $O(|V|^2 \log(|E|))$.
- ❑ The effort for the computation of a balanced minimum cut (k-way, $k \geq 2$) is NP complete.
- ❑ In the literature on the subject, MinCut is not classified as a hierarchical algorithm but treated as a special approach.

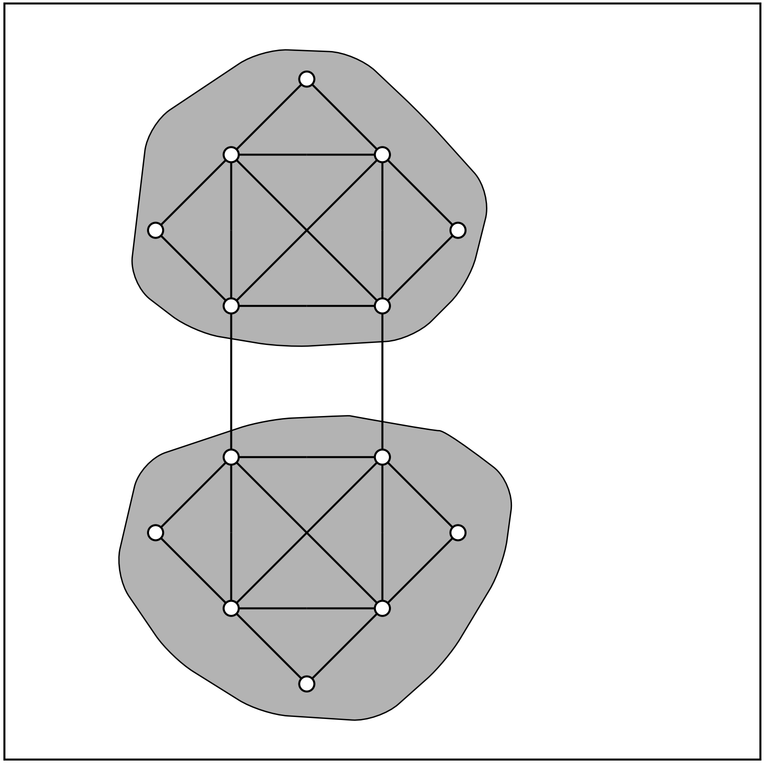
Hierarchical Cluster Analysis

Splitting Problem of MinCut



Hierarchical Cluster Analysis

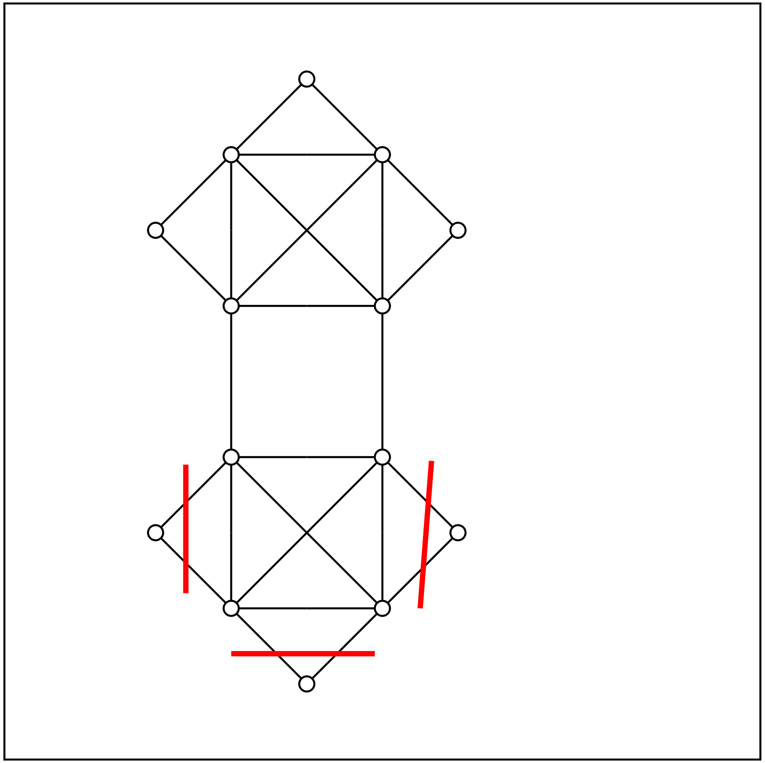
Splitting Problem of MinCut



Wish

Hierarchical Cluster Analysis

Splitting Problem of MinCut



Reality

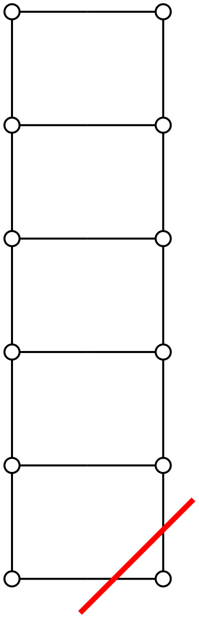
Solution: Normalization of the cut capacity with regard to the node number.

Hierarchical Cluster Analysis

Splitting Problem of MinCut

Normalized cut capacity: $\overline{w}(\{U, \bar{U}\}) = \dfrac{w(\{U, \bar{U}\})}{w(\{U, V\})} + \dfrac{w(\{U, \bar{U}\})}{w(\{\bar{U}, V\})}$

Illustration of $\overline{w}$:



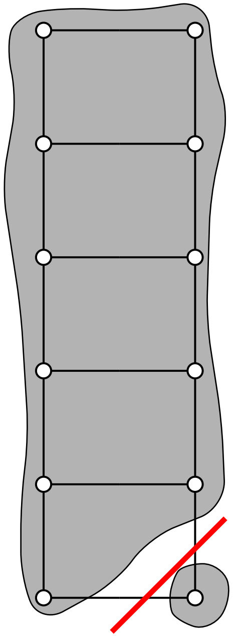
$cut(\{U, \bar{U}\}) = \{\{u, v\} \mid \{u, v\} \in E, u \in U, v \in \bar{U}\},\qquad w(\{U, \bar{U}\}) = \sum_{e \in cut(\{U, \bar{U}\})} w(e)$

Hierarchical Cluster Analysis

Splitting Problem of MinCut

Normalized cut capacity: $\overline{w}(\{U,\bar{U}\}) = \frac{w(\{U,\bar{U}\})}{w(\{U,V\})} + \frac{w(\{U,\bar{U}\})}{w(\{\bar{U},V\})}$

Illustration of $\overline{w}$:



$w(\{U, \bar{U}\}) = 2$

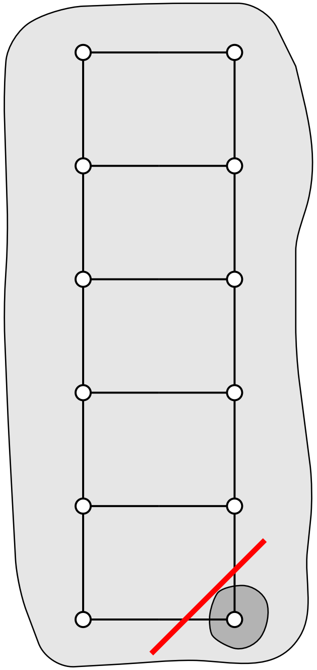
$cut(\{U,\bar{U}\}) = \{\{u,v\} \mid \{u,v\} \in E, u \in U, v \in \bar{U}\}, \qquad w(\{U,\bar{U}\}) = \sum_{e \in cut(\{U,\bar{U}\})} w(e)$

Hierarchical Cluster Analysis

Splitting Problem of MinCut

Normalized cut capacity: $\overline{w}(\{U, \bar{U}\}) = \frac{w(\{U, \bar{U}\})}{w(\{U, V\})} + \frac{w(\{U, \bar{U}\})}{w(\{\bar{U}, V\})}$

Illustration of $\overline{w}$:



$w(\{U, V\}) = 2$

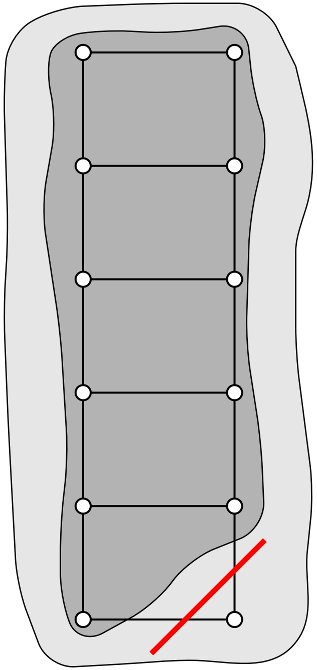
$cut(\{U, \bar{U}\}) = \{\{u, v\} \mid \{u, v\} \in E, u \in U, v \in \bar{U}\},\qquad w(\{U, \bar{U}\}) = \sum_{e \in cut(\{U, \bar{U}\})} w(e)$

Hierarchical Cluster Analysis

Splitting Problem of MinCut

Normalized cut capacity: $\overline{w}(\{U,\bar{U}\}) = \frac{w(\{U,\bar{U}\})}{w(\{U,V\})} + \frac{w(\{U,\bar{U}\})}{w(\{\bar{U},V\})}$

Illustration of $\overline{w}$:



$w\,(\{\bar{U},\,V\}) = 30$

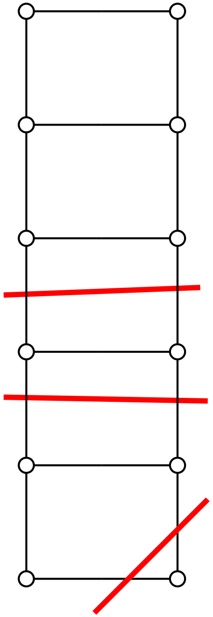
$cut(\{U,\bar{U}\}) = \{\{u,v\} \mid \{u,v\} \in E, u \in U, v \in \bar{U}\},\qquad w(\{U,\bar{U}\}) = \sum_{e \in cut(\{U,\bar{U}\})} w(e)$

Hierarchical Cluster Analysis

Splitting Problem of MinCut

Normalized cut capacity: $$\overline{w}(\{U,\bar{U}\}) = \frac{w(\{U,\bar{U}\})}{w(\{U,V\})} + \frac{w(\{U,\bar{U}\})}{w(\{\bar{U},V\})}$$

Illustration of $\overline{w}$:



$\overline{w}(\{U, \overline{U}\}) = 2/16 + 2/16 \approx 0.25$

$\overline{w}(\{U, \overline{U}\}) = 2/10 + 2/22 \approx 0.29$

$\overline{w}(\{U, \overline{U}\}) = 2/2 + 2/30 \approx 1.07$

$$cut(\{U,\bar{U}\}) = \{\{u,v\} \mid \{u,v\} \in E, u \in U, v \in \bar{U}\}, \qquad w(\{U,\bar{U}\}) = \sum_{e \in cut(\{U,\bar{U}\})} w(e)$$

Remarks:

- The computation of a minimum cut of normalized cut capacity is NP complete.
- Efficient approximations for the computation of $\overline{w}(\{U, \bar{U}\})$ have been developed and used for image segmentation and gene expression cluster analysis. [Shi/Malik 2000]

Hierarchical Cluster Analysis

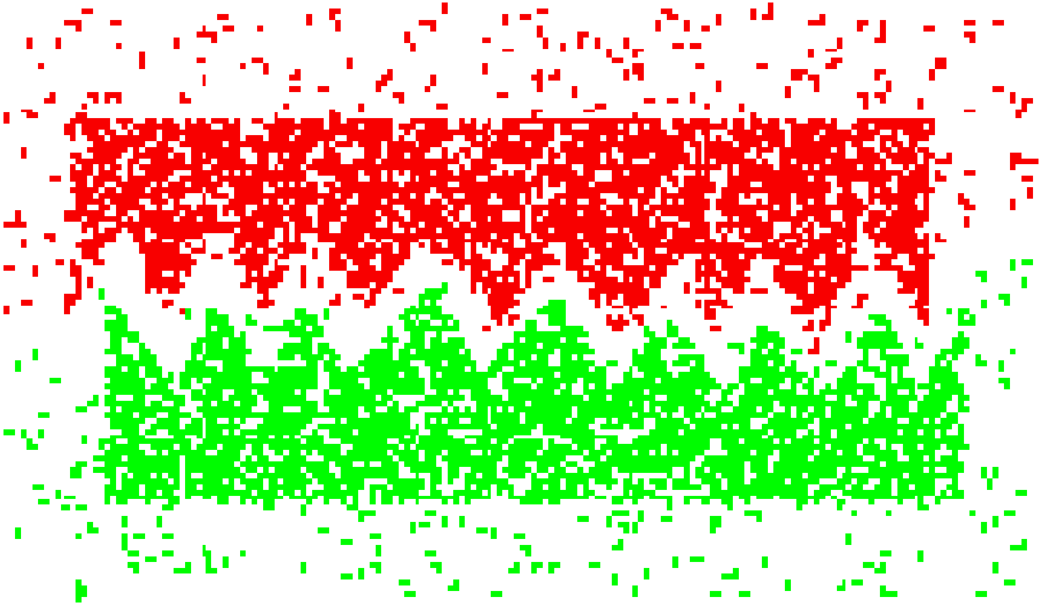
Combination of Hierarchical Algorithms

The system Chameleon combines graph thinning, graph partitioning, and a hierarchical cluster analysis [Karypis/Han/Kumar 2000] :

Hierarchical Cluster Analysis

Combination of Hierarchical Algorithms

The system Chameleon combines graph thinning, graph partitioning, and a hierarchical cluster analysis [Karypis/Han/Kumar 2000] :

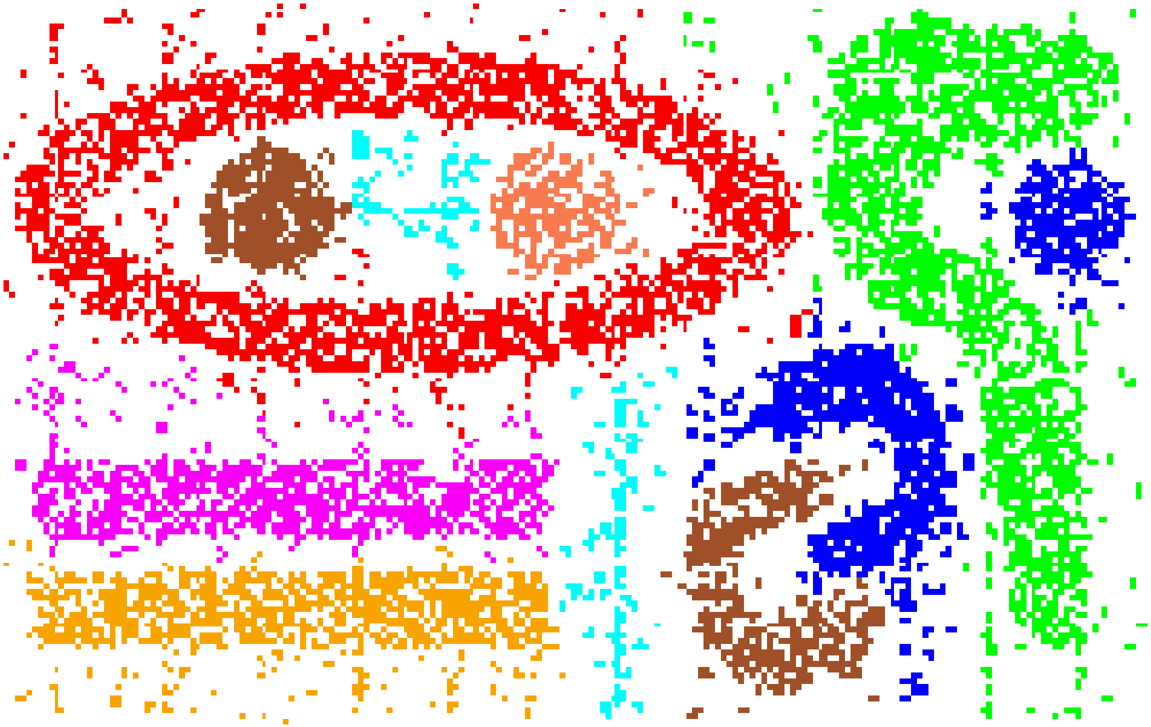


The cluster distance $d_{\mathcal{C}}(C, C')$ is defined as $d_{\mathcal{C}} = \dfrac{1}{R_I(C, C') \cdot (R_C(C, C'))^{\alpha}}$

Hierarchical Cluster Analysis

Combination of Hierarchical Algorithms

Chameleon (continued) [Karypis/Han/Kumar 2000] :



The hyperparameter α in $d_{\mathcal{C}}$ is task-depending and has to be determined by the user (via trial and error).