Chapter ML:XI

XI. Cluster Analysis

- Data Mining Overview
- Cluster Analysis Basics
- Hierarchical Cluster Analysis
- Iterative Cluster Analysis
- Density-Based Cluster Analysis
- Cluster Evaluation
- Constrained Cluster Analysis

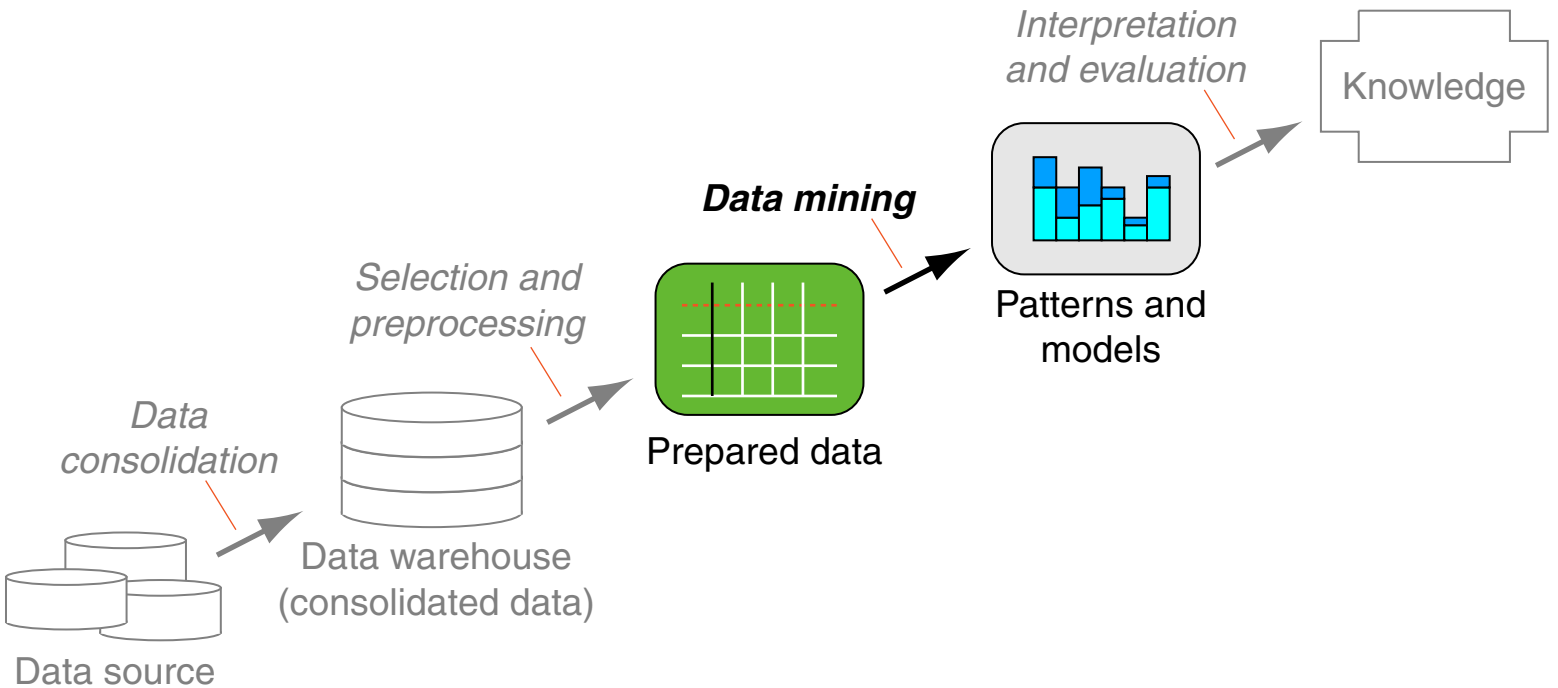
Data Mining Overview

Definition 1 (Data Mining)
Data mining is the systematic, usually automated or semi-automated discovery and extraction of so far unknown relations from huge data sets.

Data mining involves the following steps:

1. specification of the task
2. selection of the data
3. data preprocessing and data transformation
4. pattern recognition
5. presentation

Data Mining Overview



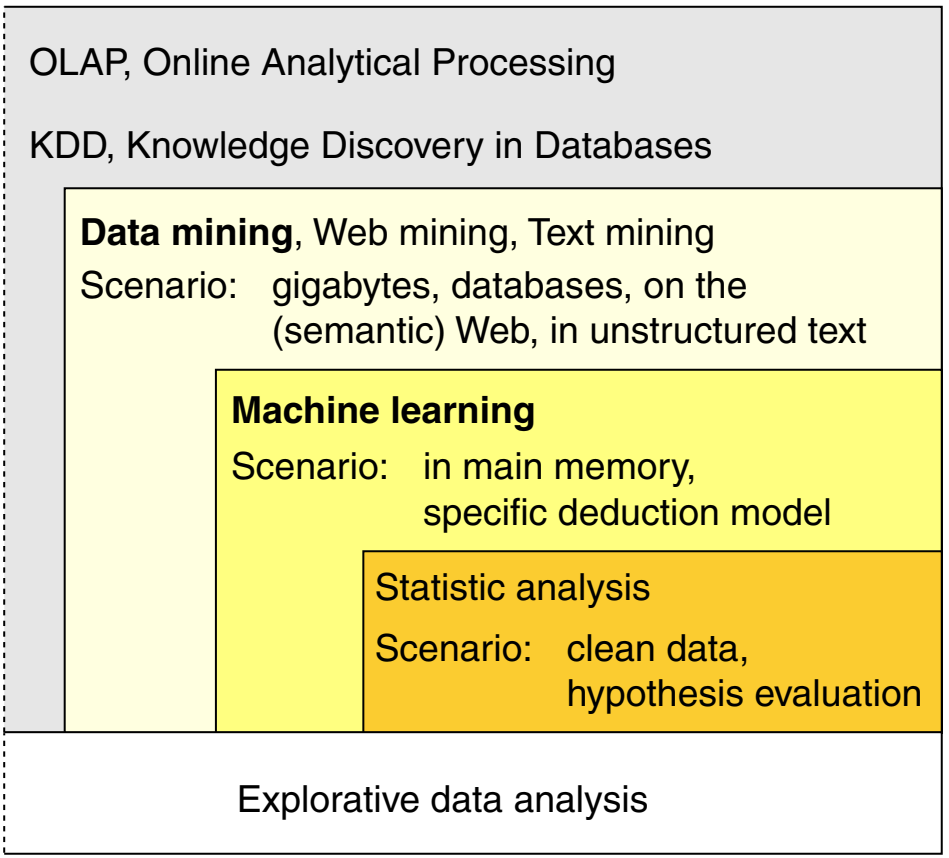
Definition 2 (Knowledge Discovery in Databases, KDD)
Knowledge Discovery in Databases is the process of identifying valid, new, relevant, and interpretable patterns in huge data sets.

[Fayyad 1996, Wrobel 1998]

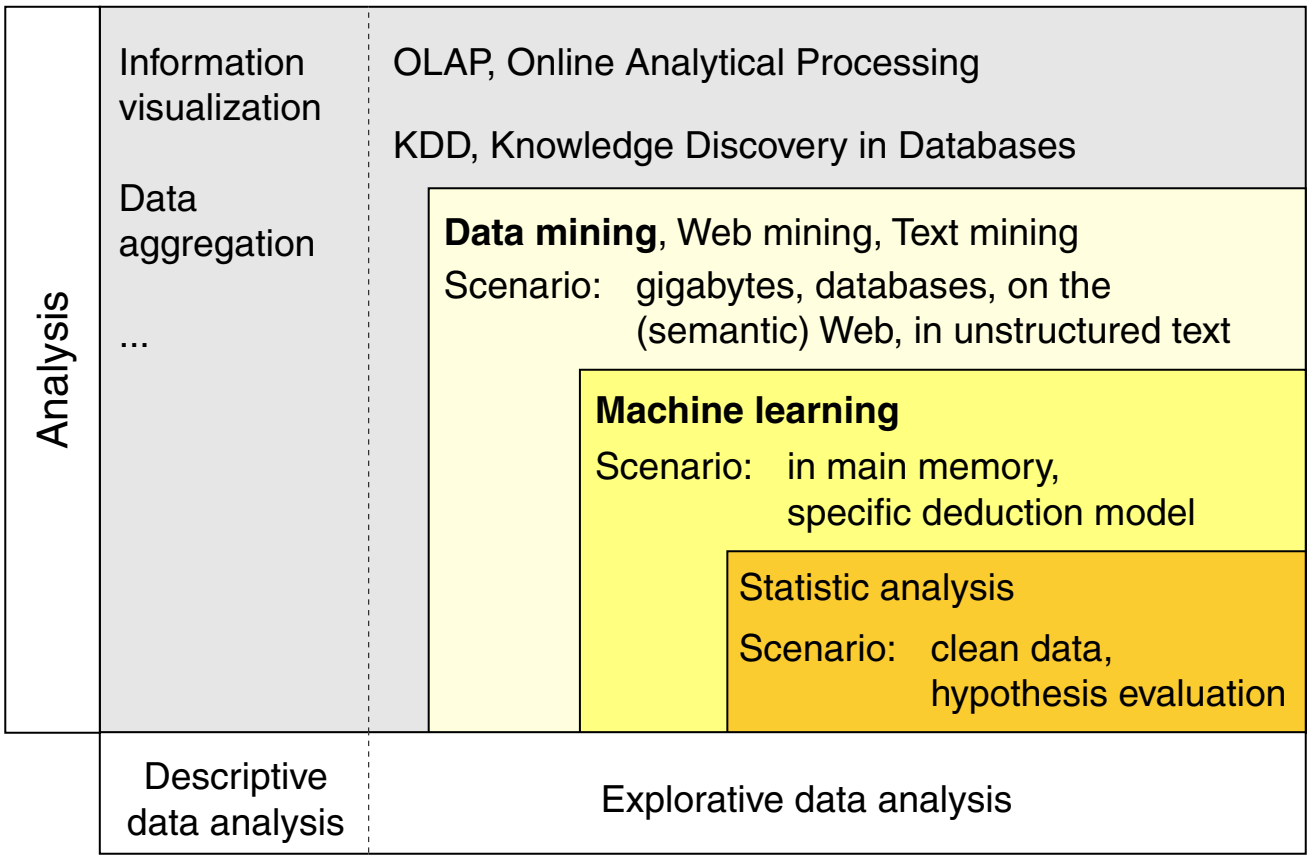
Remarks:

- Data mining technology belongs to the field of *explorative data analysis*. Explorative data analysis deals with both data presentation and search for structures, peculiarities, and anomalies. It is employed if the research question is fuzzy or if the choice of the statistical model is unclear.
- The data mining definition does not use the notion of “information”: under the viewpoint of semiotics, data mining operates on the sigmatic layer only.
  The *interpretation* of discovered patterns, i.e., the examination of information with regard to new findings and a subjective knowledge gain, which happens on the pragmatic layer, belongs to the field of KDD.
- In the business world, the terms data mining and knowledge discovery in databases, KDD, are used synonymously. Note however, that data mining designates only a single step within a KDD process, namely the analysis step for pattern recognition.
- Web data mining is the transfer and usage of data mining technology for information extraction on the Internet and especially the World Wide Web. Text mining is the identification of relevant information in text.

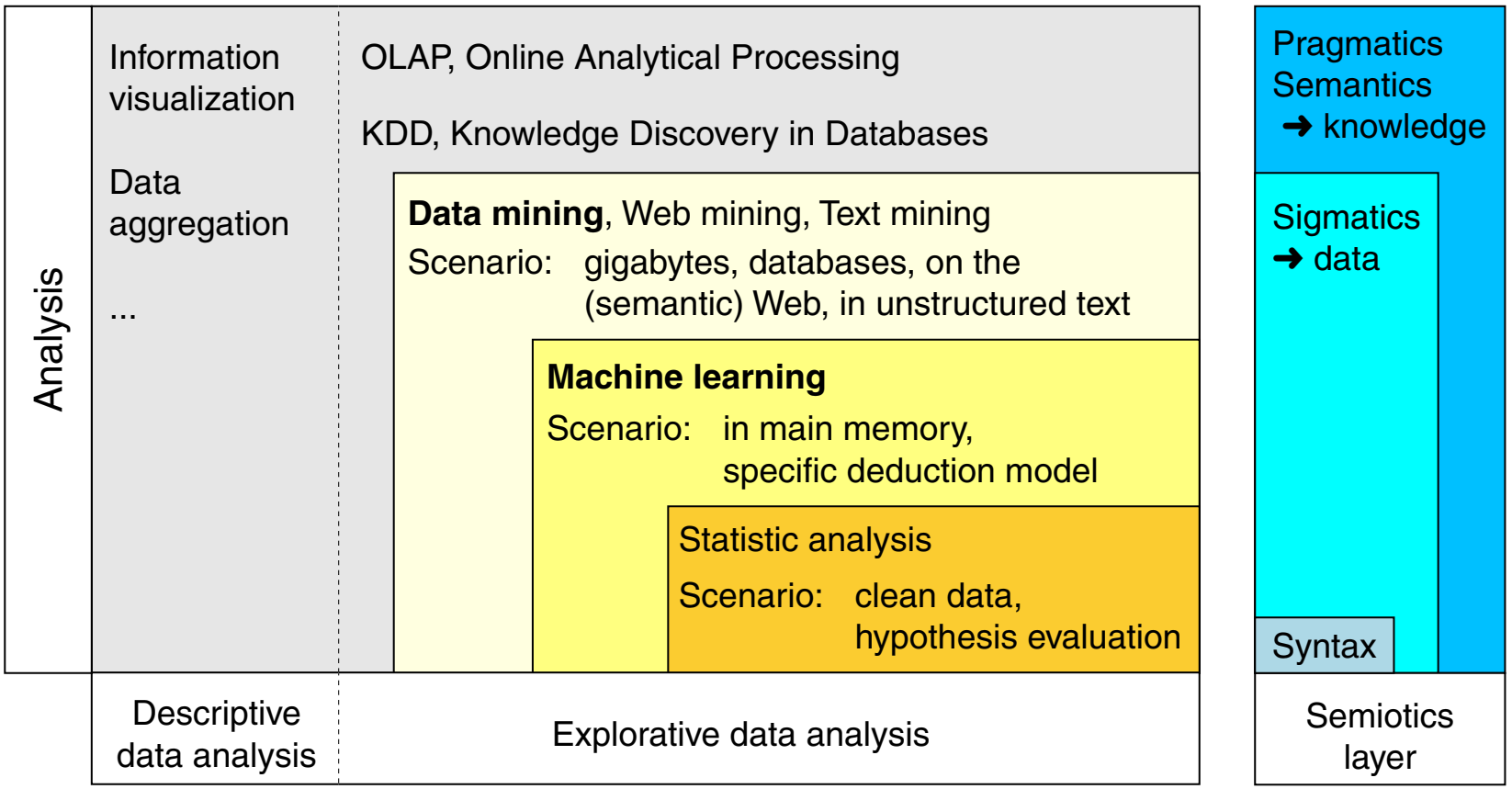
Data Mining Overview



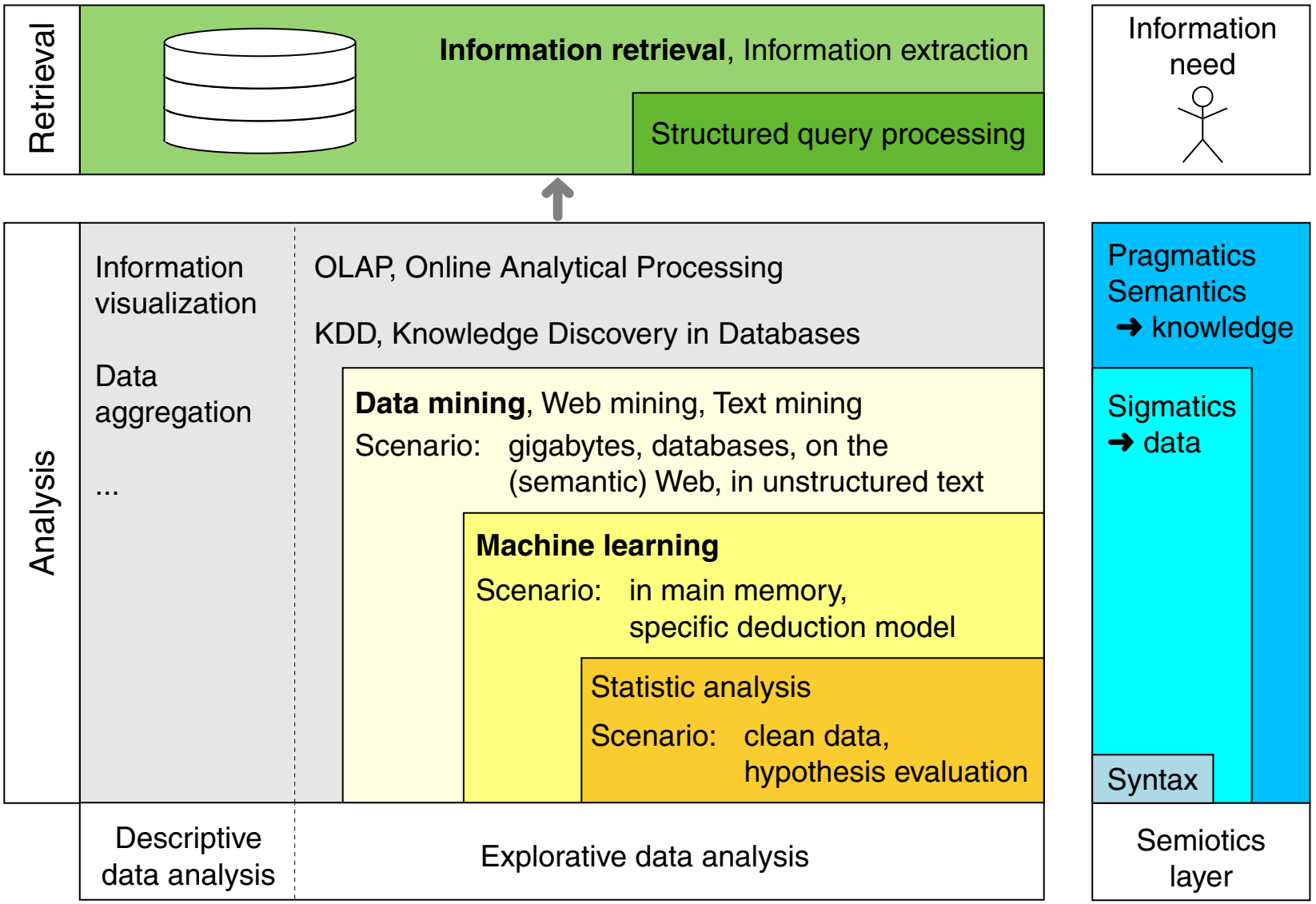
Data Mining Overview



Data Mining Overview



Data Mining Overview



Remarks:

- A clear separation between machine learning and data mining is not always possible. A key difference, however, results from the sizes of the analyzed data sets: machine learning applications are usually executed in main memory. The field of data mining arose from the necessity to apply analysis methods to large data bases.
- The foci of machine learning are the processes and theories of learning and deduction, such as analogical reasoning, learning from examples, or reinforcement-driven learning. The major driving force behind data mining is the business world with their large data bases.
- The following count to relevant data mining problems: undirected association analysis to identify dependencies between consumer products (market basket analysis), cluster analysis and categorization, filtering of process data, forecasting and prediction.

Data Mining Overview

Methods and Tools

- cluster analysis
- Learning of propositional or description-logical rules. Example:

  ```
  IF status=married AND house_owner=true THEN creditor=good
  ```

- Learning of association rules. Example:

  “75% of the buyers of product A will buy the products B, C, and D as well.”

- principal component analysis (PCA), factor analysis
- multi-dimensional scaling (MDS)

Chapter ML:XI (continued)

XI. Cluster Analysis

Cluster Analysis Basics

Cluster analysis is the unsupervised classification of a set of objects in groups, pursuing the following objectives:

1. maximize the similarities within the groups (intra groups)
2. minimize the similarities between the groups (inter groups)

Cluster Analysis Basics

Cluster analysis is the unsupervised classification of a set of objects in groups, pursuing the following objectives:

1. maximize the similarities within the groups (intra groups)
2. minimize the similarities between the groups (inter groups)

Applications:

- identification of similar groups of buyers
- “higher-level” image processing: object recognition
- search of similar gene profiles
- specification of syndromes
- analysis of traffic data in computer networks
- visualization of complex graphs
- text categorization in information retrieval

Remarks:

- The setting of a cluster analysis is reverse to the setting of a variance analysis:
  - A variance analysis verifies whether a nominal feature defines groups such that the members of the different groups differ significantly with regard to a numerical feature. I.e., the nominal feature is in the role of the independent variable, while the numerical feature(s) is (are) in role of dependent variable(s). Example: The type of a product packaging may define the number of customers in a supermarket who look at the product.
  - A cluster analysis in turn can be used to identify such a nominal feature, namely by constructing a suited feature domain for the nominal variable: each cluster corresponds implicitly to a value of the domain. Example: Equivalent but differently presented products in a supermarket are clustered with regard to the number of customers who buy the products.
- Cluster analysis is a tool for structure *generation*. Nearly nothing is known about the nominal variable that is to be identified. In particular, there is no knowledge about the number of domain values (the number of clusters).
- Variance analysis is a tool for structure *verification*.

Cluster Analysis Basics

Let $\mathbf{x}_1, \ldots \mathbf{x}_n$ denote the p-dimensional feature vectors of n objects:

	Feature 1	Feature 2	...	Feature p
$\mathbf{x}_1$	x_{1_1}	x_{1_2}	...	x_{1_p}
$\mathbf{x}_2$	x_{2_1}	x_{2_2}	...	x_{2_p}
$\vdots$				
$\mathbf{x}_n$	x_{n_1}	x_{n_2}	...	x_{n_p}

Cluster Analysis Basics

Let $\mathbf{x}_1, \ldots \mathbf{x}_n$ denote the p-dimensional feature vectors of n objects:

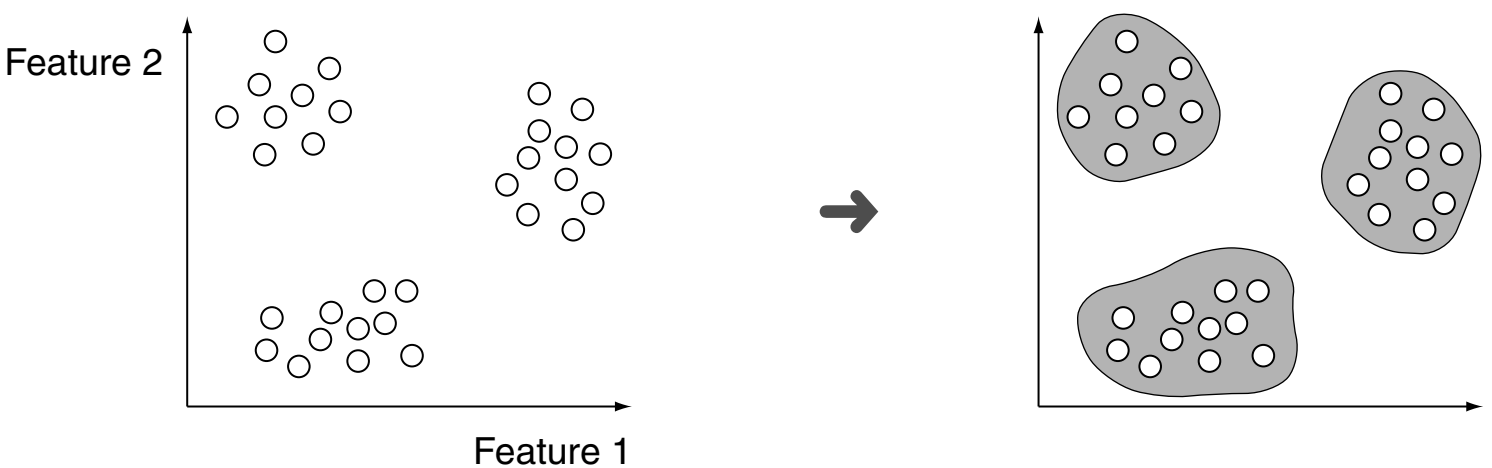
	Feature 1	Feature 2	...	Feature p	no Target concept
$\mathbf{x}_1$	x_{1_1}	x_{1_2}	...	x_{1_p}	c_1
$\mathbf{x}_2$	x_{2_1}	x_{2_2}	...	x_{2_p}	c_2
$\vdots$					$\vdots$
$\mathbf{x}_n$	x_{n_1}	x_{n_2}	...	x_{n_p}	c_n

Cluster Analysis Basics

Let $\mathbf{x}_1, \ldots \mathbf{x}_n$ denote the p-dimensional feature vectors of n objects:

	Feature 1	Feature 2	...	Feature p	no Target concept
$\mathbf{x}_1$	x_{1_1}	x_{1_2}	...	x_{1_p}	c_1
$\mathbf{x}_2$	x_{2_1}	x_{2_2}	...	x_{2_p}	c_2
$\vdots$					$\vdots$
$\mathbf{x}_n$	x_{n_1}	x_{n_2}	...	x_{n_p}	c_n

30 two-dimensional feature vectors ($n = 30$, $p = 2$):



Cluster Analysis Basics

Definition 3 (Exclusive Clustering [splitting])

Let X be a set of feature vectors. An exclusive clustering $\mathcal{C}$ of X, $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$, $C_i \subseteq X$, is a partitioning of X into non-empty, mutually exclusive subsets C_i with $\bigcup_{C_i \in \mathcal{C}} C_i = X$.

Cluster Analysis Basics

Definition 3 (Exclusive Clustering [splitting]**)**
Let X be a set of feature vectors. An exclusive clustering $\mathcal{C}$ of X, $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$, $C_i \subseteq X$, is a partitioning of X into non-empty, mutually exclusive subsets C_i with $\bigcup_{C_i \in \mathcal{C}} C_i = X$.

Algorithms for cluster analysis are unsupervised learning methods:

- the learning process is self-organized
- there is no (external) teacher
- the optimization criterion is task- and domain-*independent*

Cluster Analysis Basics

Definition 3 (Exclusive Clustering [splitting]**)**
Let X be a set of feature vectors. An exclusive clustering $\mathcal{C}$ of X, $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$, $C_i \subseteq X$, is a partitioning of X into non-empty, mutually exclusive subsets C_i with $\bigcup_{C_i \in \mathcal{C}} C_i = X$.

Algorithms for cluster analysis are unsupervised learning methods:

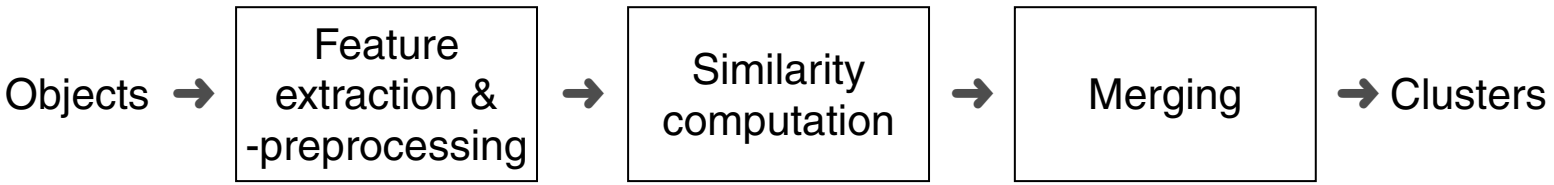
- the learning process is self-organized
- there is no (external) teacher
- the optimization criterion is task- and domain-*independent*

Supervised learning:

- a learning objective such as the target concept is provided
- the optimization criterion *depends* on the task or the domain
- information is provided about *how* the optimization criterion can be maximized. Keyword: instructive feedback

Cluster Analysis Basics

Main Stages of a Cluster Analysis

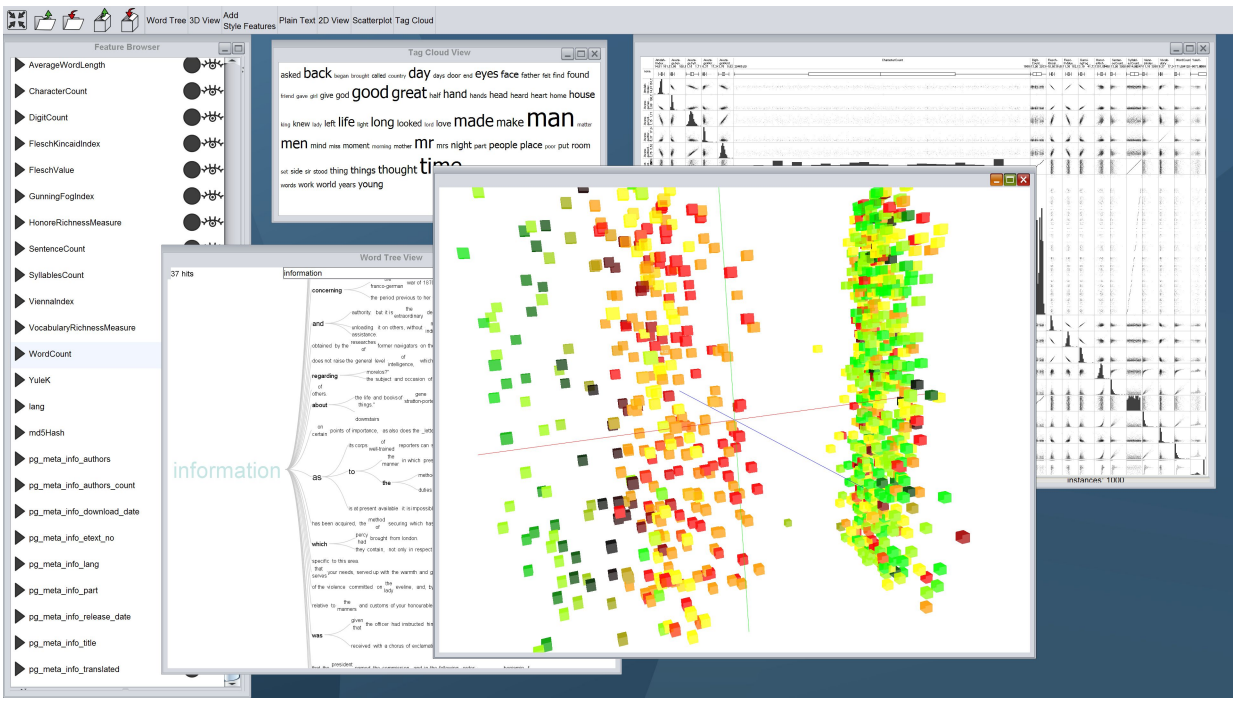


Cluster Analysis Basics

Feature Extraction and Preprocessing

Required are (possibly new) features of high variance. Approaches:

- analysis of dispersion parameters
- dimension reduction: PCA, factor analysis, MDS
- visual inspection: scatter plots, box plots



[Webis 2012, VDM tool]

Cluster Analysis Basics

Feature Extraction and Preprocessing

Required are (possibly new) features of high variance. Approaches:

- analysis of dispersion parameters
- dimension reduction: PCA, factor analysis, MDS
- visual inspection: scatter plots, box plots

Feature standardization can dampen the structure and make things worse:

Cluster Analysis Basics

Computation of Distances or Similarities

	$\mathbf{x}_1$	$\mathbf{x}_2$	$\ldots$	$\mathbf{x}_n$
$\mathbf{x}_1$	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

Remarks:

- Usually, the distance matrix is defined implicitly by a metric on the feature space.
- The distance matrix can be understood as the adjacency matrix of a weighted, undirected graph G, $G = \langle V, E, w \rangle$. The set X of feature vectors is mapped one-to-one (bijection) onto a set of nodes V. The distance $d(\mathbf{x}_i, \mathbf{x}_j)$ corresponds to the weight $w(\{u, v\})$ of edge $\{u, v\} \in E$ between those nodes u and v that are associated with $\mathbf{x}_i$ and $\mathbf{x}_j$ respectively.

Cluster Analysis Basics

Computation of Distances or Similarities (continued)

Properties of a distance function:

1. $d(\mathbf{x}_1, \mathbf{x}_2) \geq 0$
2. $d(\mathbf{x}_1, \mathbf{x}_1) = 0$
3. $d(\mathbf{x}_1, \mathbf{x}_2) = d(\mathbf{x}_2, \mathbf{x}_1)$
4. $d(\mathbf{x}_1, \mathbf{x}_3) \leq d(\mathbf{x}_1, \mathbf{x}_2) + d(\mathbf{x}_2, \mathbf{x}_3)$

Cluster Analysis Basics

Computation of Distances or Similarities (continued)

Properties of a distance function:

1. $d(\mathbf{x}_1, \mathbf{x}_2) \geq 0$
2. $d(\mathbf{x}_1, \mathbf{x}_1) = 0$
3. $d(\mathbf{x}_1, \mathbf{x}_2) = d(\mathbf{x}_2, \mathbf{x}_1)$
4. $d(\mathbf{x}_1, \mathbf{x}_3) \leq d(\mathbf{x}_1, \mathbf{x}_2) + d(\mathbf{x}_2, \mathbf{x}_3)$

Minkowsky metric for features with interval-based measurement scales:

$$d(\mathbf{x}_1, \mathbf{x}_2) = \left(\sum_{i=1}^{p} |x_{1_i} - x_{2_i}|^r \right)^{1/r}$$

where

- $r = 1$. Manhattan or Hamming distance, L_1 norm
- $r = 2$. Euclidean distance, L_2 norm
- $r = \infty$. Maximum distance, L_∞ norm or L_{max} norm

Cluster Analysis Basics

Computation of Distances or Similarities (continued)

Cluster analysis does not presume a particular measurement scale.

- ➜ Generalization of the distance function towards a (dis)similarity function by omitting the triangle inequality. (Dis)similarities can be quantified between all kinds of features irrespective of the given levels of measurement.

Cluster Analysis Basics

Computation of Distances or Similarities (continued)

Cluster analysis does not presume a particular measurement scale.

➜ Generalization of the distance function towards a (dis)similarity function by omitting the triangle inequality. (Dis)similarities can be quantified between all kinds of features irrespective of the given levels of measurement.

Similarity coefficients given two feature vectors, $\mathbf{x}_1$, $\mathbf{x}_2$, with binary features:

$$\text{Simple Matching Coefficient (SMC)} = \frac{f_{11} + f_{00}}{f_{11} + f_{00} + f_{01} + f_{10}}$$

$$\text{Jaccard Coefficient (J)} = \frac{f_{11}}{f_{11} + f_{01} + f_{10}}$$

where

f_{11} = number of features with a value of 1 in both $\mathbf{x}_1$ and $\mathbf{x}_2$

f_{00} = number of features with a value of 0 in both $\mathbf{x}_1$ and $\mathbf{x}_2$

f_{01} = number of features with value 0 in $\mathbf{x}_1$ and value 1 in $\mathbf{x}_2$

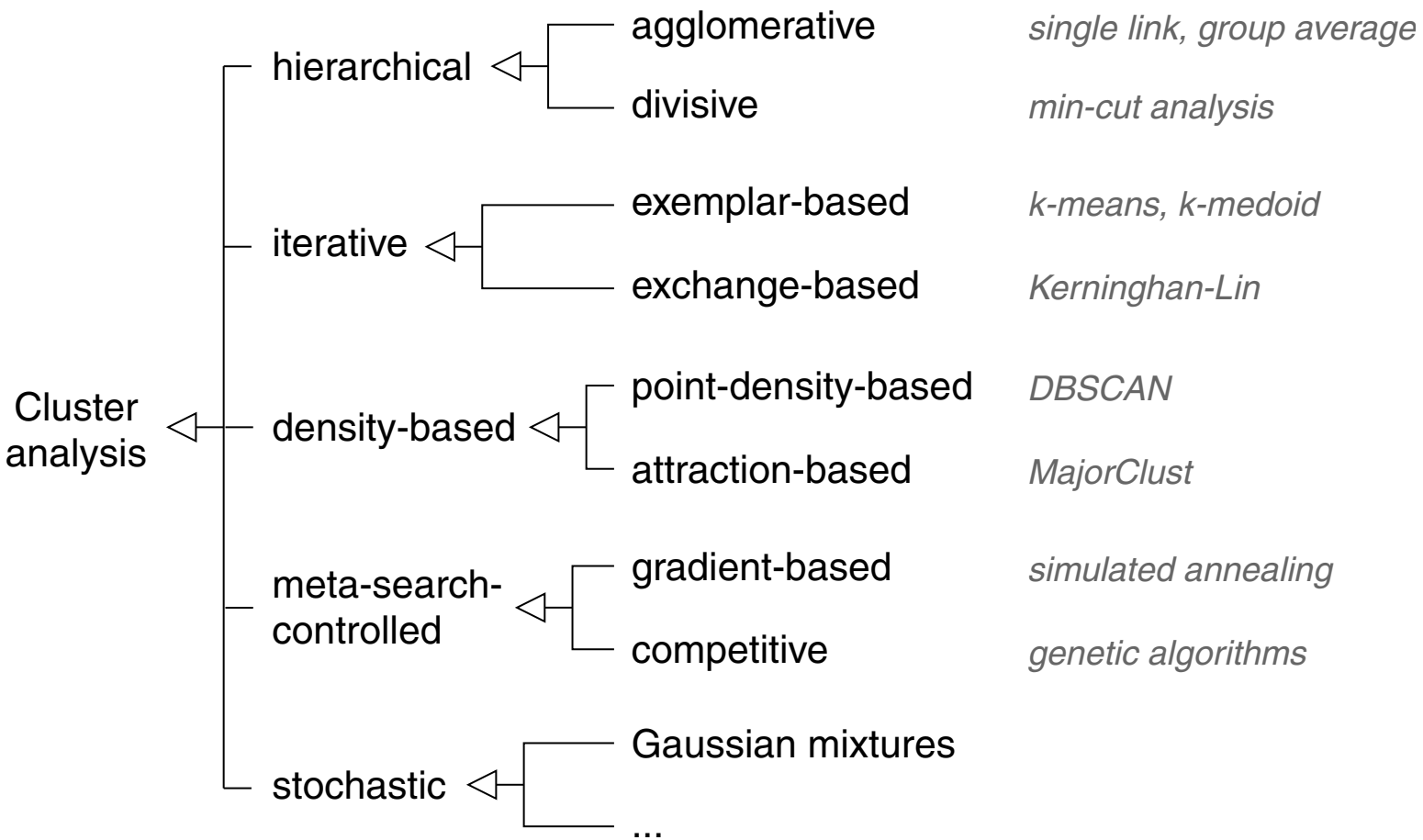
f_{10} = number of features with value 1 in $\mathbf{x}_1$ and value 0 in $\mathbf{x}_2$

Remarks:

- The definitions for the above similarity coefficients can be extended towards features with a nominal measurement scale.
- Particular heterogeneous metrics have been developed, such as HEOM and HVDM, which allow the combined computation of feature values from different measurement scales.
- The computation of the correlation between all features of two feature vectors (not: between between two features over all feature vectors) allows to compare feature profiles.
  Example: Q correlation coefficient
- The development of a suited, realistic, and expressive similarity measure may pose the biggest challenge within a cluster analysis tasks. Typical problems:
  - (unwanted) structure damping due to normalization
  - (unwanted) sensitivity concerning outliers
  - (not recognized) feature correlations
  - (not considered) varying feature importances
- Similarity measures can be transformed straightforward into dissimilarity measures—and vice versa.

Cluster Analysis Basics

Merging Principles

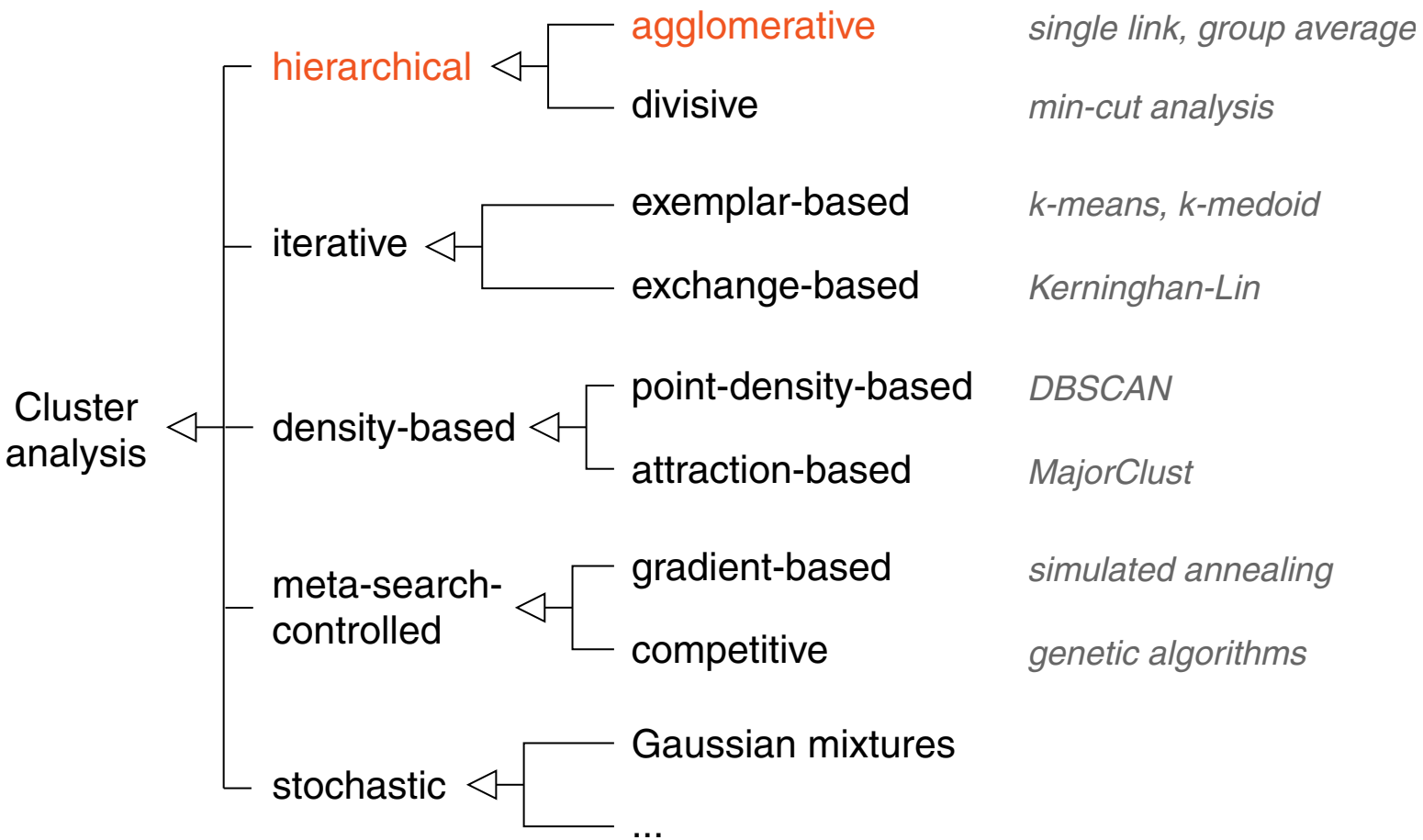


Chapter ML:XI (continued)

XI. Cluster Analysis

Hierarchical Cluster Analysis

Merging Principles



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.

1. $\mathcal{C} = \{\{v\} \mid v \in V\}$ `// initial clustering`
2.
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. $\mathcal{C} = (\mathcal{C} \setminus \{C, C'\}) \cup \{C \cup C'\}$ `// merging`
7.
8. **ENDDO**
9. **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.

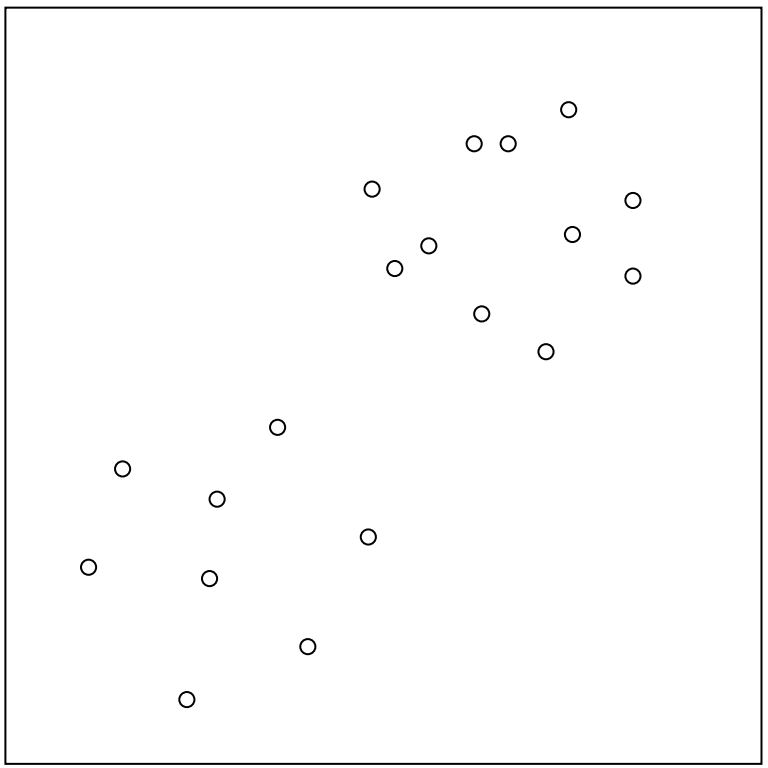
Output: $T = \langle V_T, E_T \rangle$. Cluster hierarchy or dendrogram.

1. $\mathcal{C} = \{\{v\} \mid v \in V\}$ `// initial clustering`
2. $V_T = \{v_C \mid C \in \mathcal{C}\}$, $E_T = \emptyset$ `// 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. $\mathcal{C} = (\mathcal{C} \setminus \{C, C'\}) \cup \{C \cup C'\}$ `// merging`
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)

Compare the above algorithm to the hierarchical divisive algorithm.

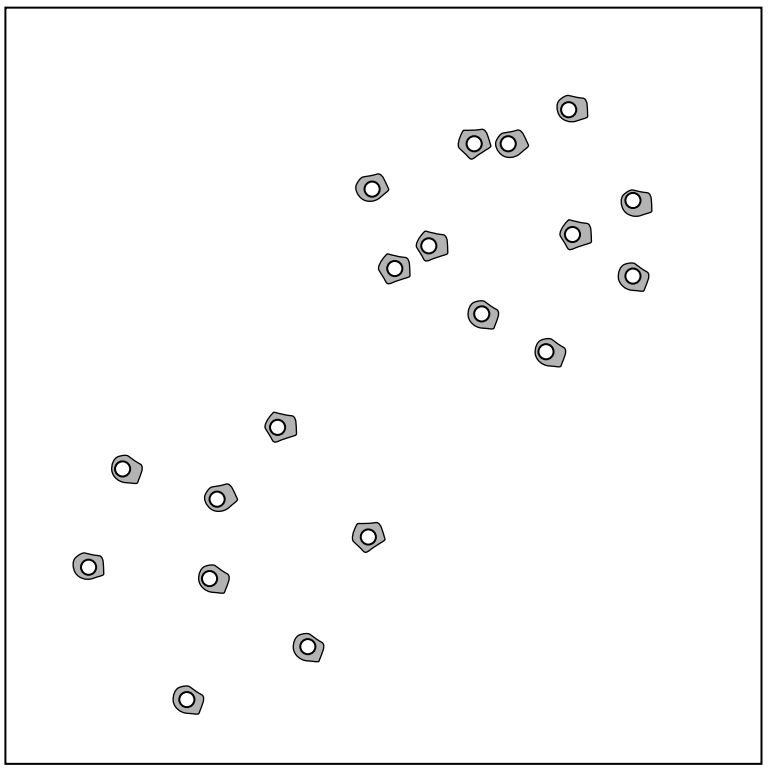
Hierarchical Cluster Analysis

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



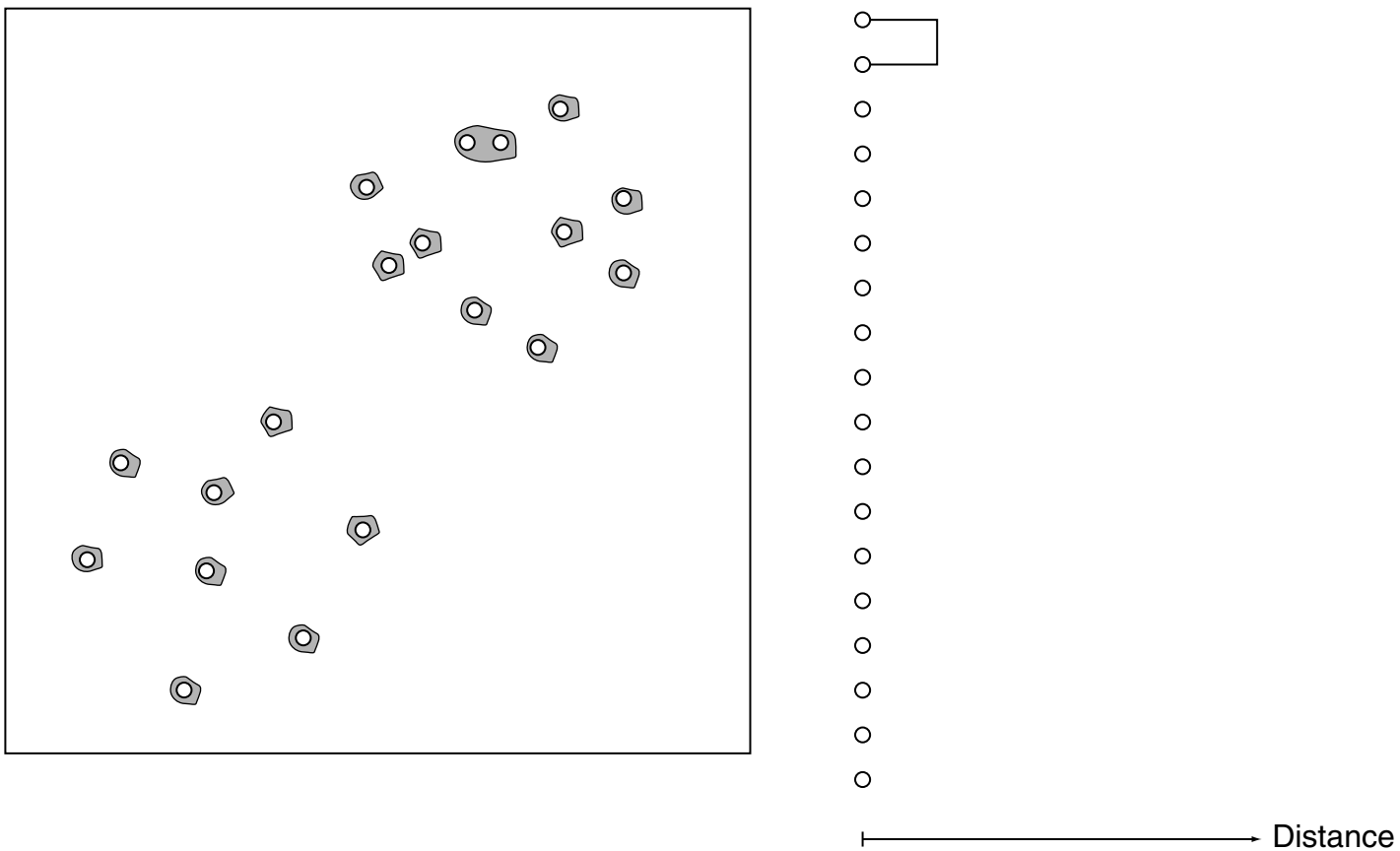
Hierarchical Cluster Analysis

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



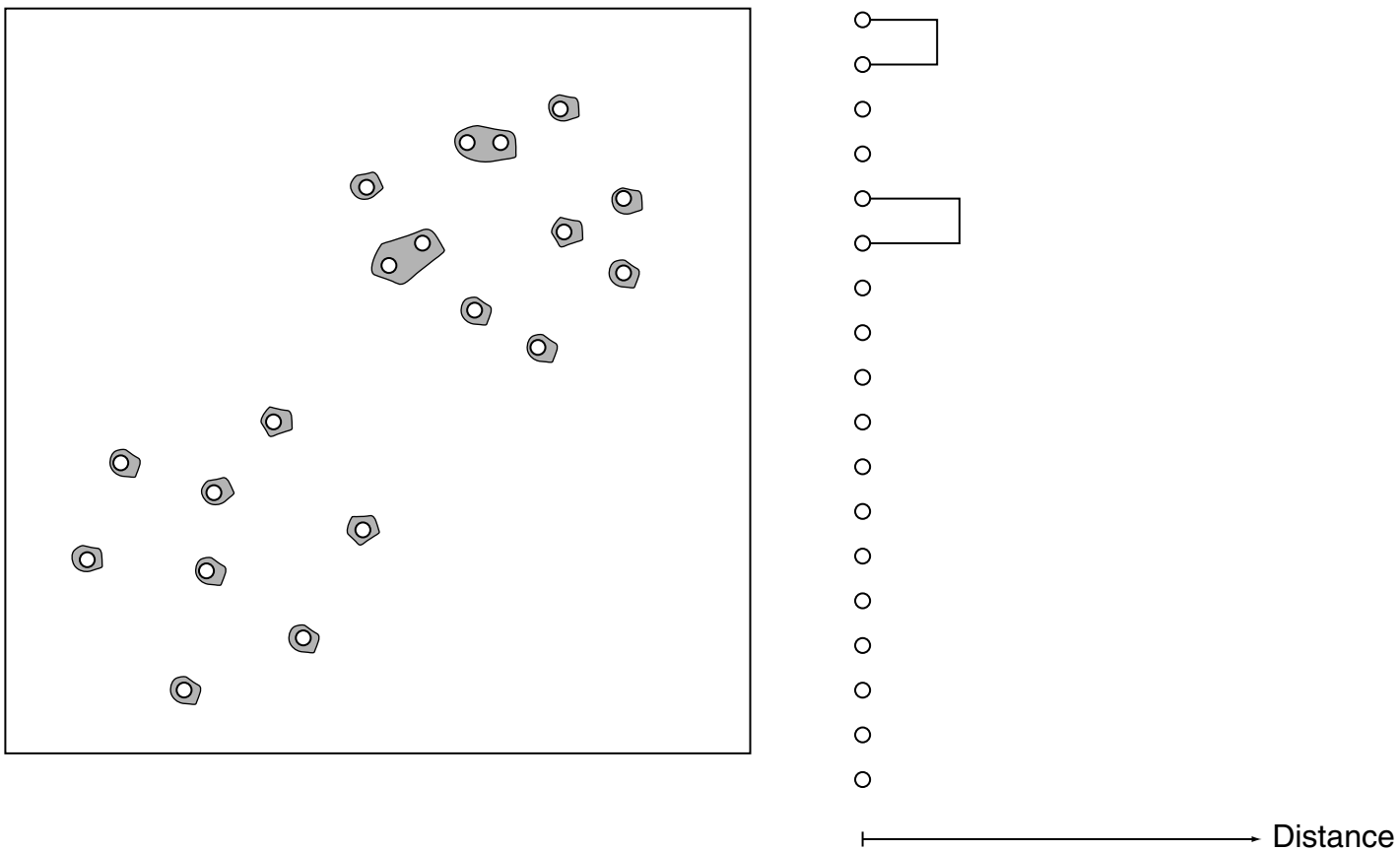
Hierarchical Cluster Analysis

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



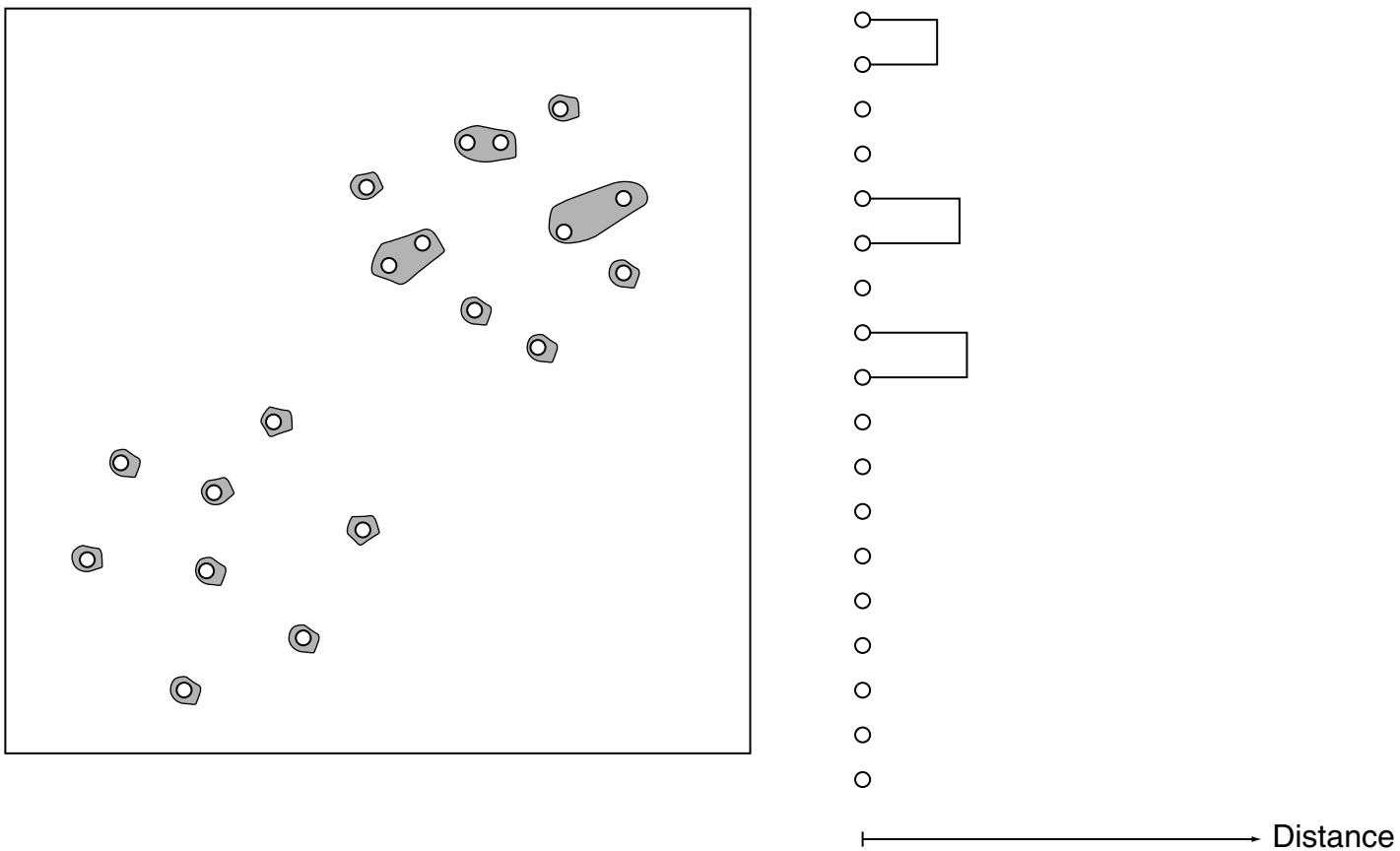
Hierarchical Cluster Analysis

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



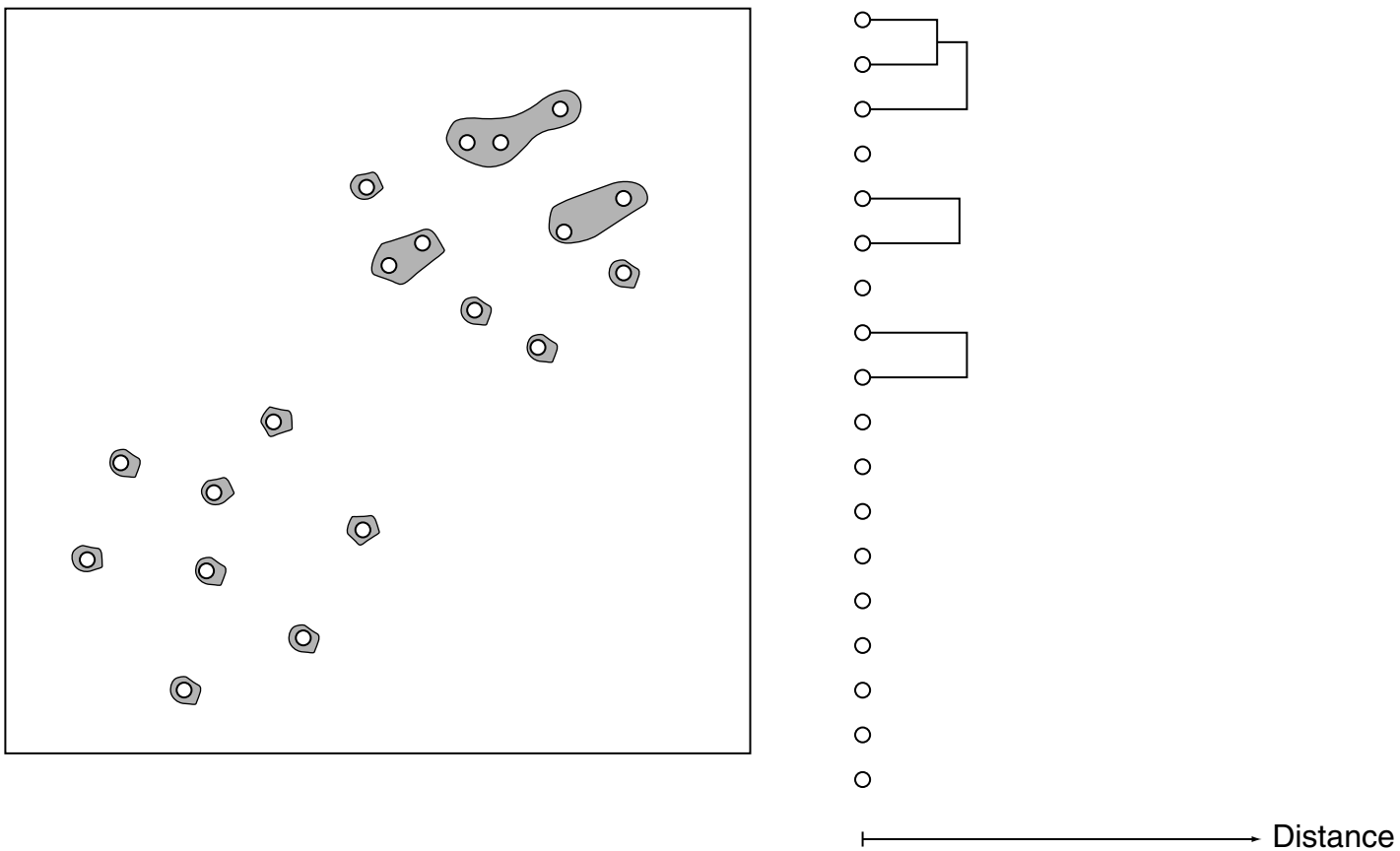
Hierarchical Cluster Analysis

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



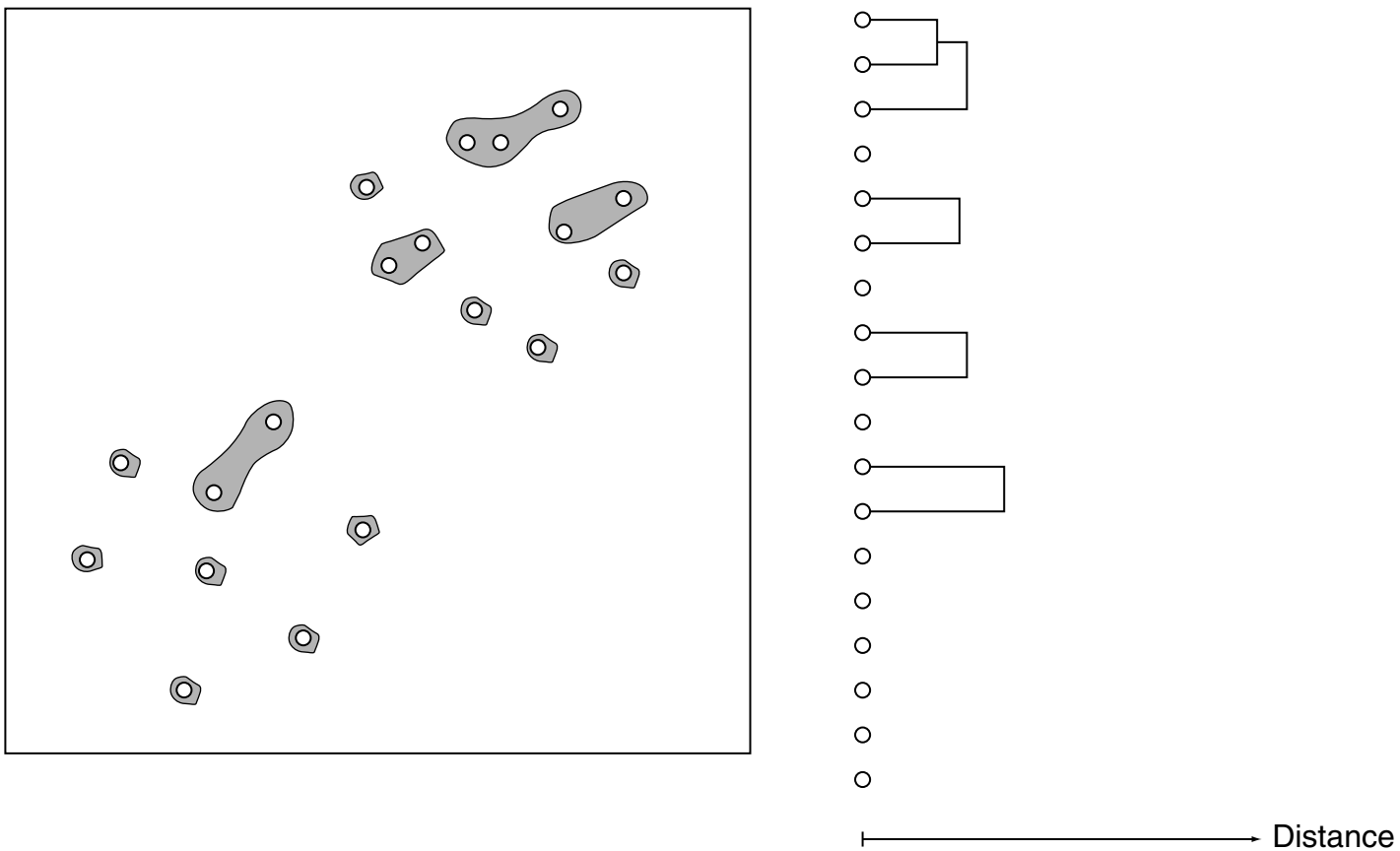
Hierarchical Cluster Analysis

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



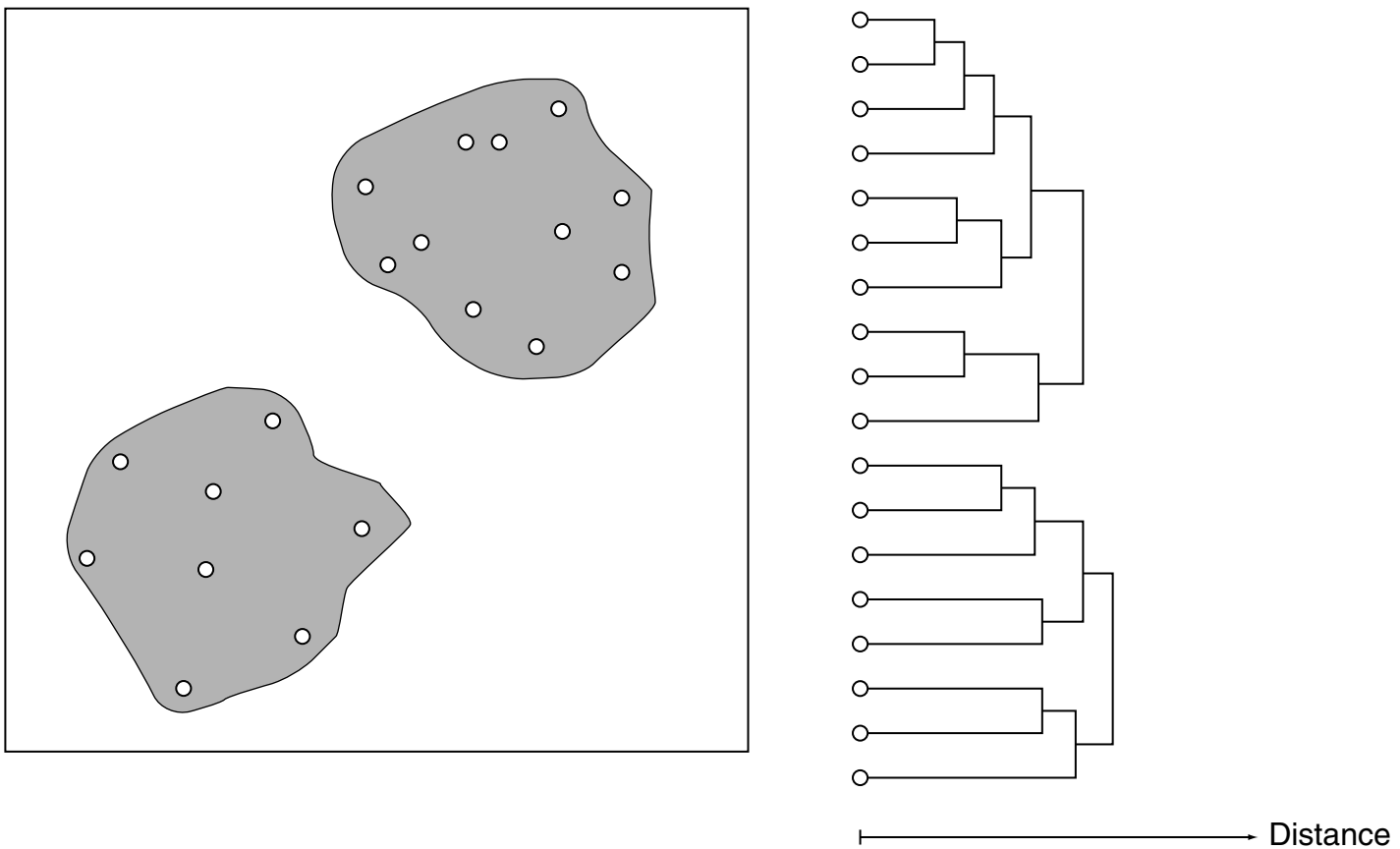
Hierarchical Cluster Analysis

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



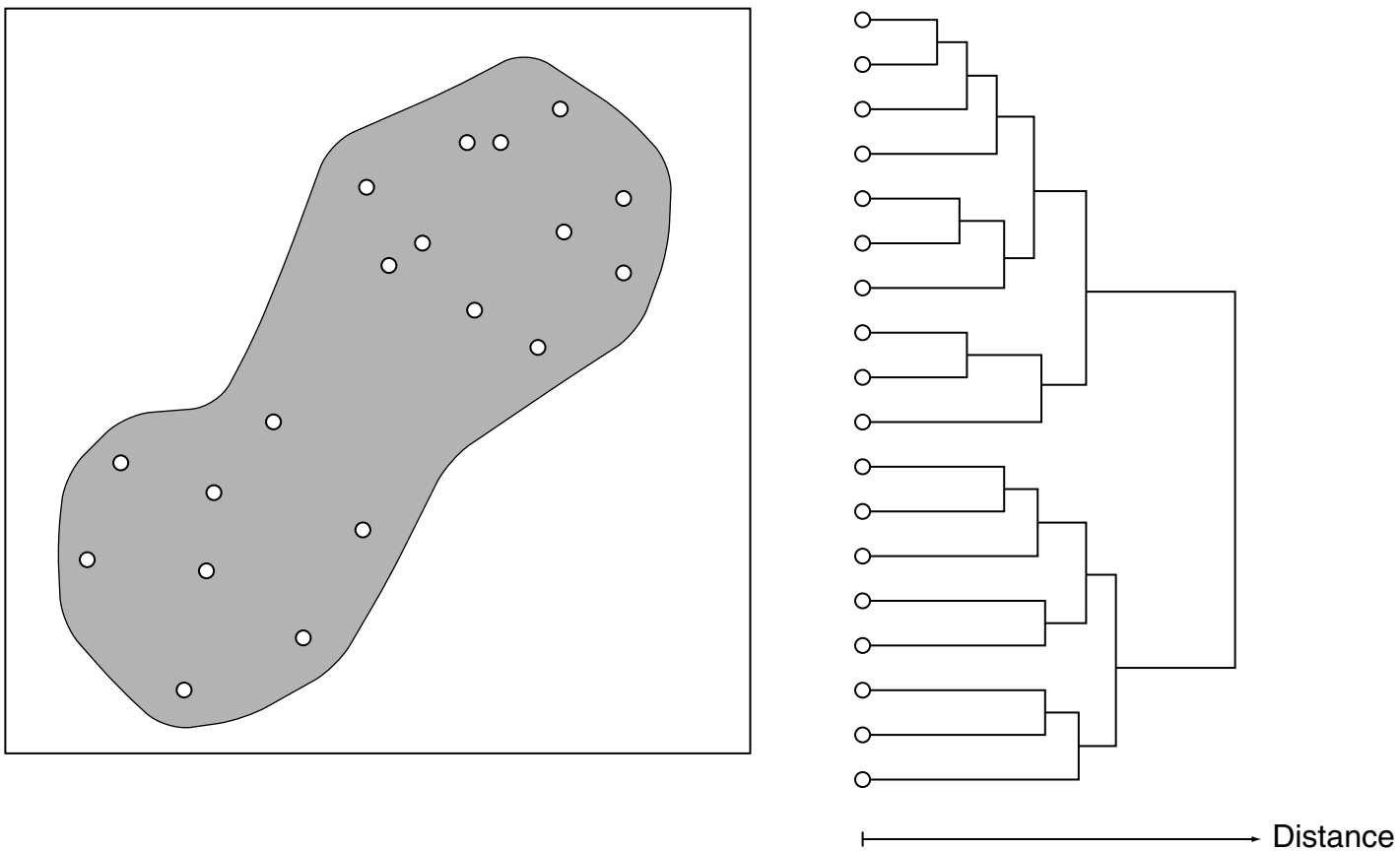
Hierarchical Cluster Analysis

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 Measures of Hierarchical Agglomerative Algorithms [characteristics]

$d_{\mathcal{C}}(C, C') = \min_{\substack{u \in C \\ v \in C'}} d(u, v)$	single link (nearest neighbor)
$d_{\mathcal{C}}(C, C') = \max_{\substack{u \in C \\ v \in C'}} d(u, v)$	complete link (furthest neighbor)
$d_{\mathcal{C}}(C, C') = \frac{1}{\lvert C \rvert \cdot \lvert C' \rvert} \sum_{\substack{u \in C \\ v \in C'}} d(u, v)$	group average link
$d_{\mathcal{C}}(C, C') = \sqrt{\frac{2 \cdot \lvert C \rvert \cdot \lvert C' \rvert}{\lvert C \rvert + \lvert C' \rvert}} \cdot \lvert\lvert \bar{u} - \bar{v} \rvert\rvert$	Ward criterion (variance)

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:

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

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:

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

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:

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

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:

$$\begin{aligned}
\mathit{ESS}(C) &= \sum_{u \in C} ||\bar{u} - u||^2 \;=\; \sum_{u \in C} (||\bar{u}||^2 - 2 \cdot \langle u, \bar{u} \rangle + ||u||^2) \\
&= |C| \cdot ||\bar{u}||^2 - 2|C| \cdot ||\bar{u}||^2 + \sum_{u \in C} ||u||^2 \;=\; \sum_{u \in C} ||u||^2 \;- |C| \cdot ||\bar{u}||^2 \\
\mathit{ESS}(C') &= \sum_{v \in C'} ||v||^2 \;- |C'| \cdot ||\bar{v}||^2
\end{aligned}$$

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:

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

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

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

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

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:

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

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

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

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

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

$\bar{u}$ 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 actual number of clusters) to obtain the desired new distances:

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

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:

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

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:

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

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

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:

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

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

$$= 0.5 \cdot (d_{\mathcal{C}}(C, C_i) + d_{\mathcal{C}}(C', C_i)) - 0.5 \cdot |d_{\mathcal{C}}(C, C_i) - d_{\mathcal{C}}(C', C_i)|$$

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:

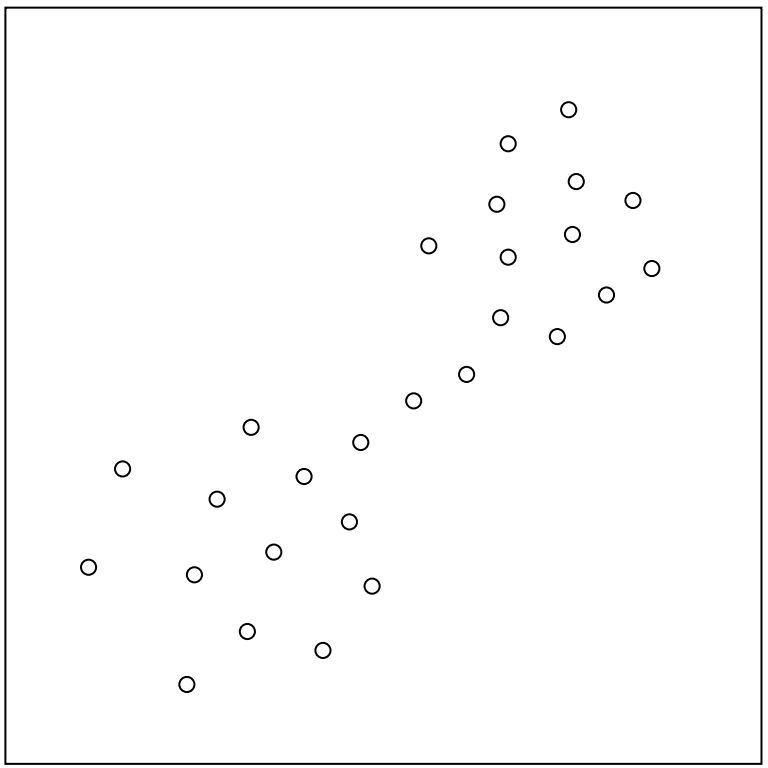
$$
\begin{aligned}
d_{\mathcal{C}}(C \cup C', C_i) &= \min_{\substack{u \in C \cup C' \\ v \in C_i}} d(u, v) \quad \text{[distance measure]} \\
&= \min\{d_{\mathcal{C}}(C, C_i),\ d_{\mathcal{C}}(C', C_i)\} \\
&= 0.5 \cdot (d_{\mathcal{C}}(C, C_i) + d_{\mathcal{C}}(C', C_i)) - 0.5 \cdot |d_{\mathcal{C}}(C,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

Remarks:

- Link-based algorithms can be used with arbitrary measures for distances and similarities.
- Single link can be operationalized straightforward 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 all members (points) of two clusters again, the formula exploits the cluster distances that were computed in the preceding iteration.
  How large is the runtime improvement compared to a naive approach that exploits only the distance information in $G = \langle V, E, w \rangle$?

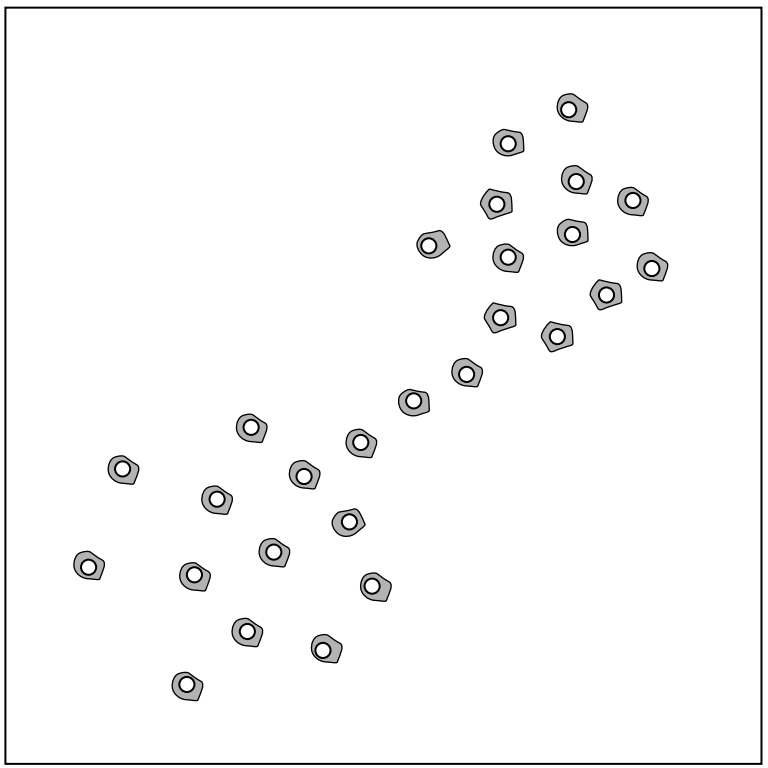
Hierarchical Cluster Analysis

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



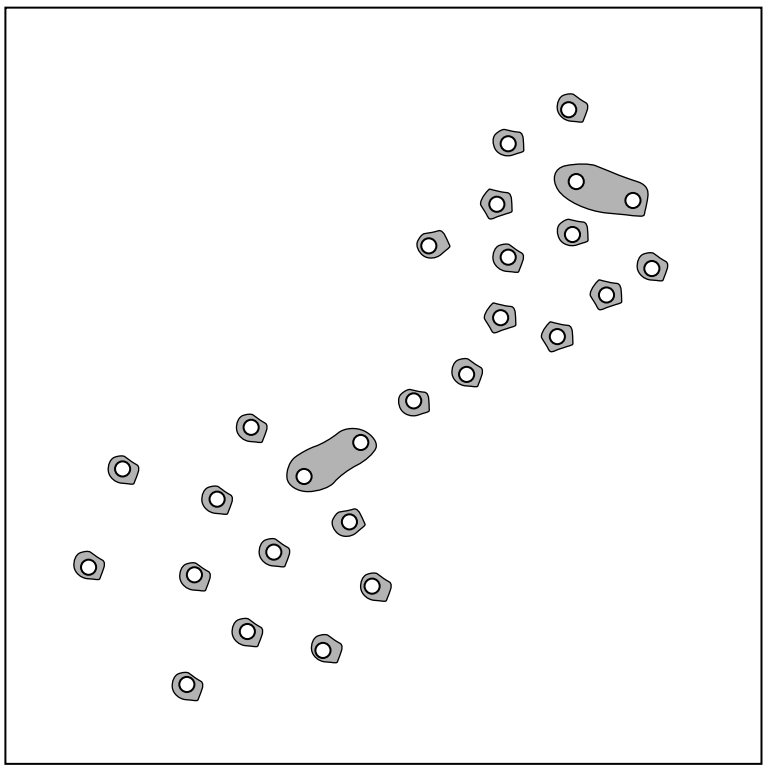
Hierarchical Cluster Analysis

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



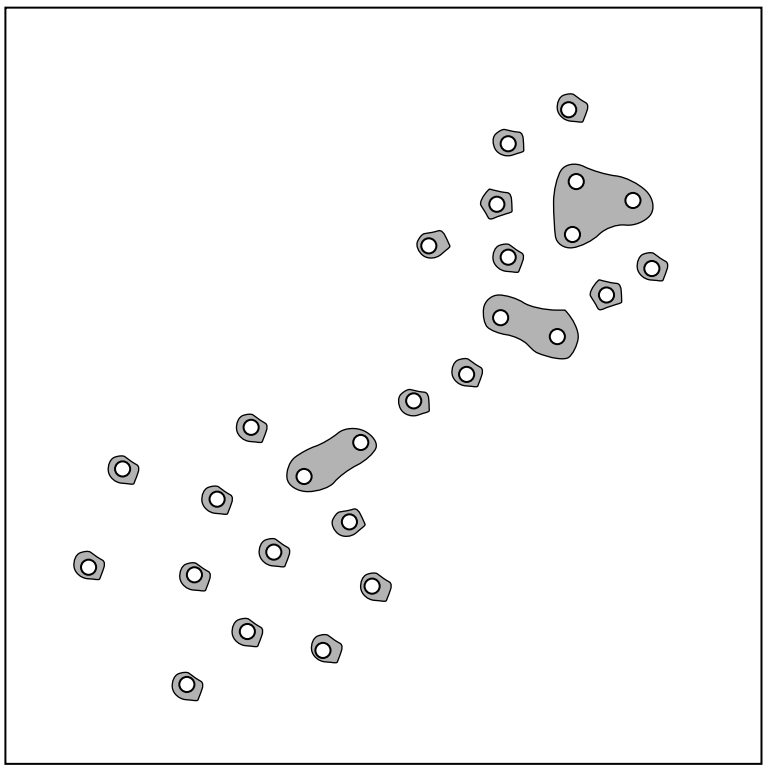
Hierarchical Cluster Analysis

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



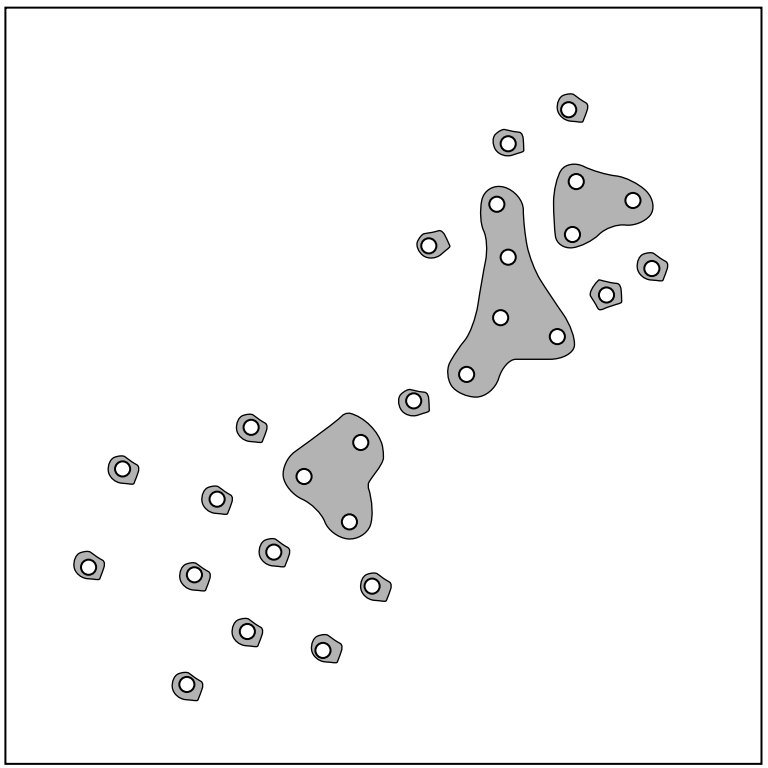
Hierarchical Cluster Analysis

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



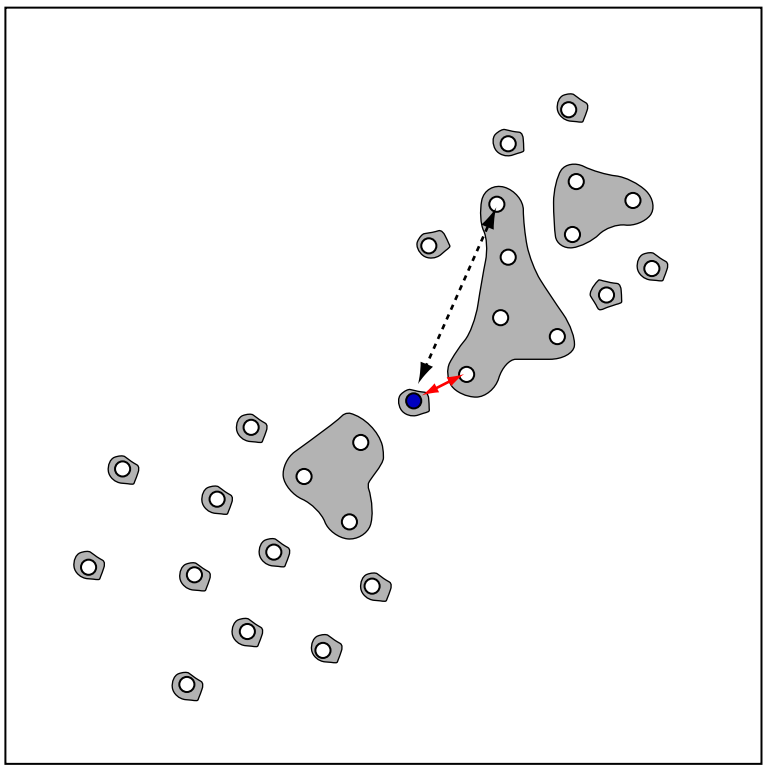
Hierarchical Cluster Analysis

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



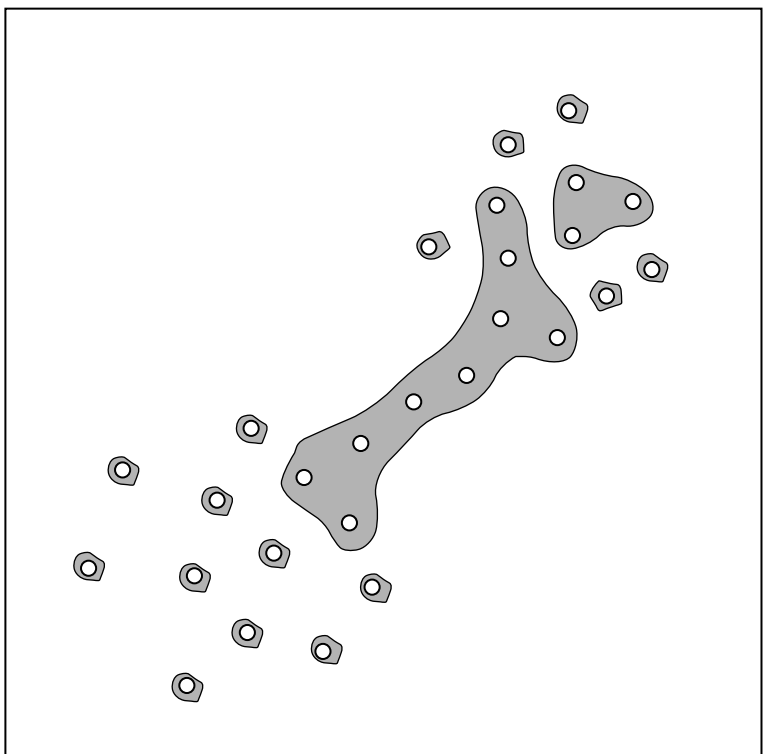
Hierarchical Cluster Analysis

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



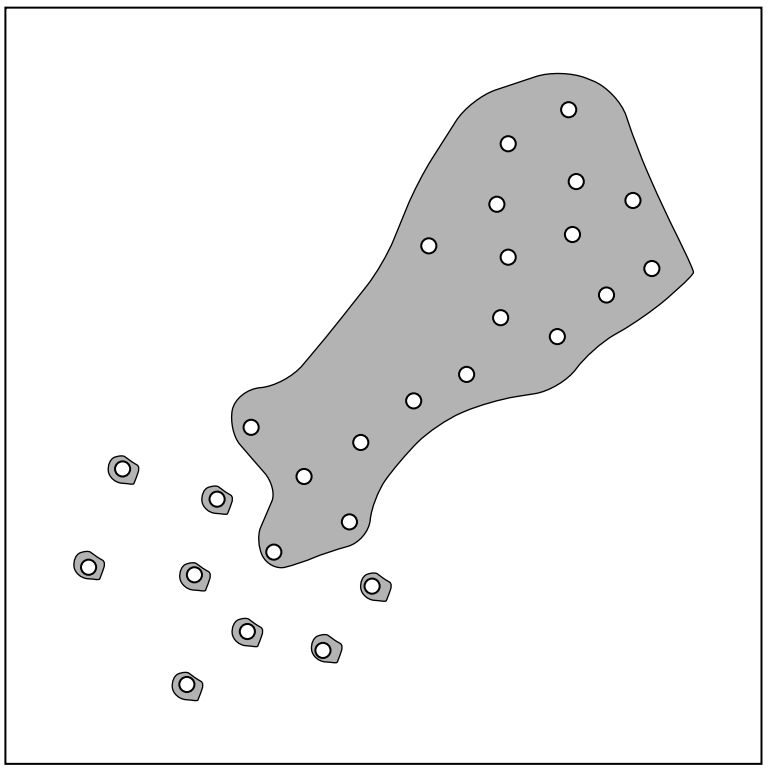
Hierarchical Cluster Analysis

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



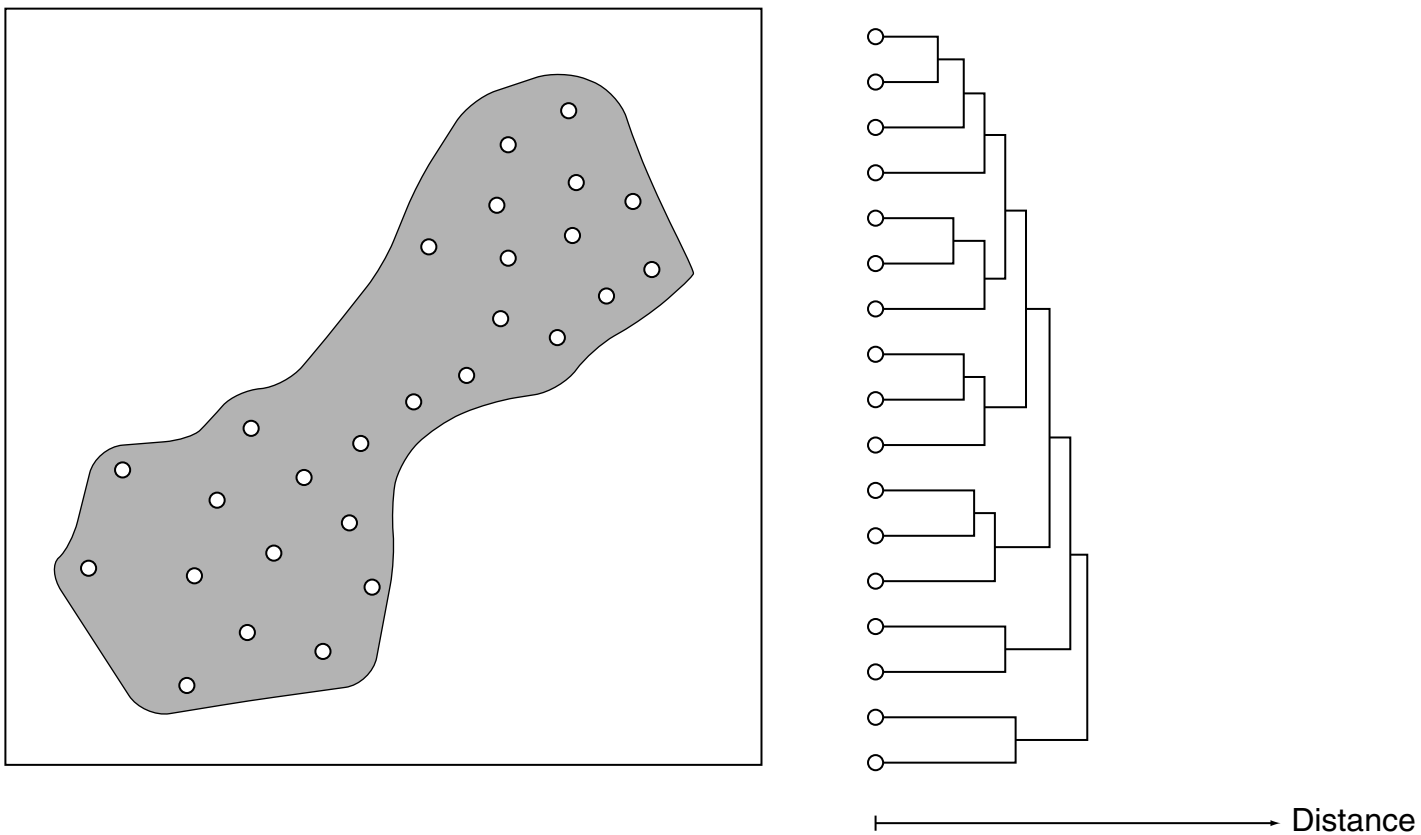
Hierarchical Cluster Analysis

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



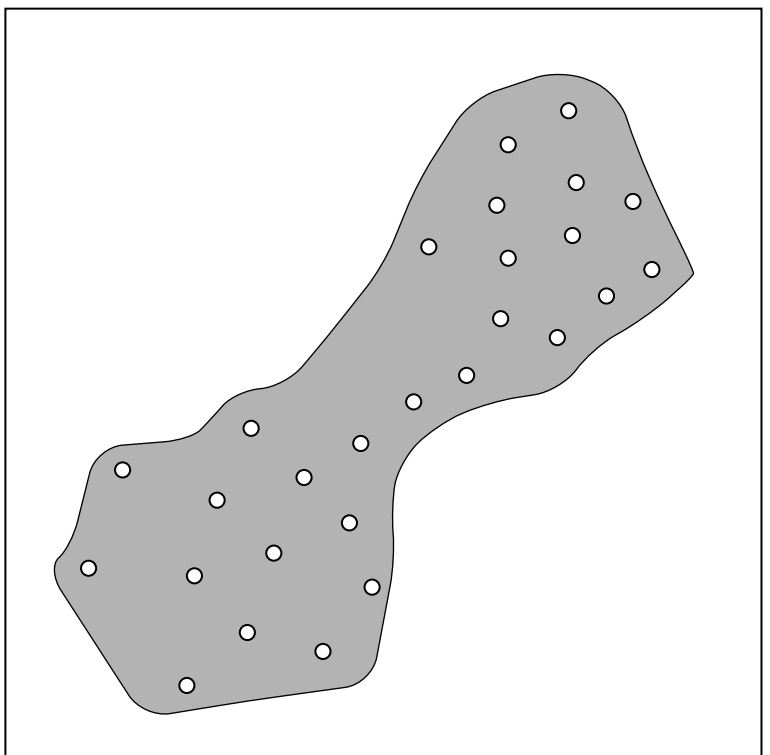
Hierarchical Cluster Analysis

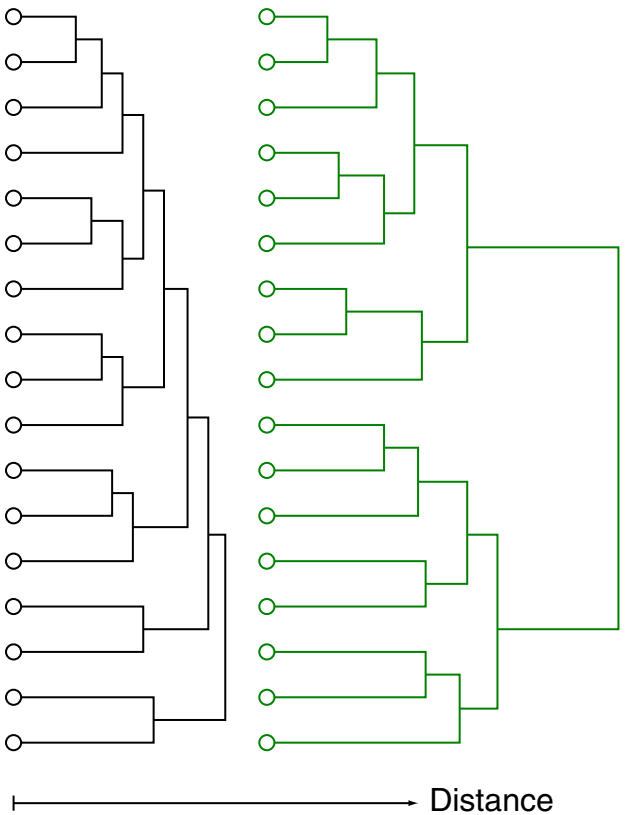
Chaining Problem of Single Link (d_C = Nearest Neighbor)



Hierarchical Cluster Analysis

Chaining Problem of Single Link (d_C = Nearest Neighbor)



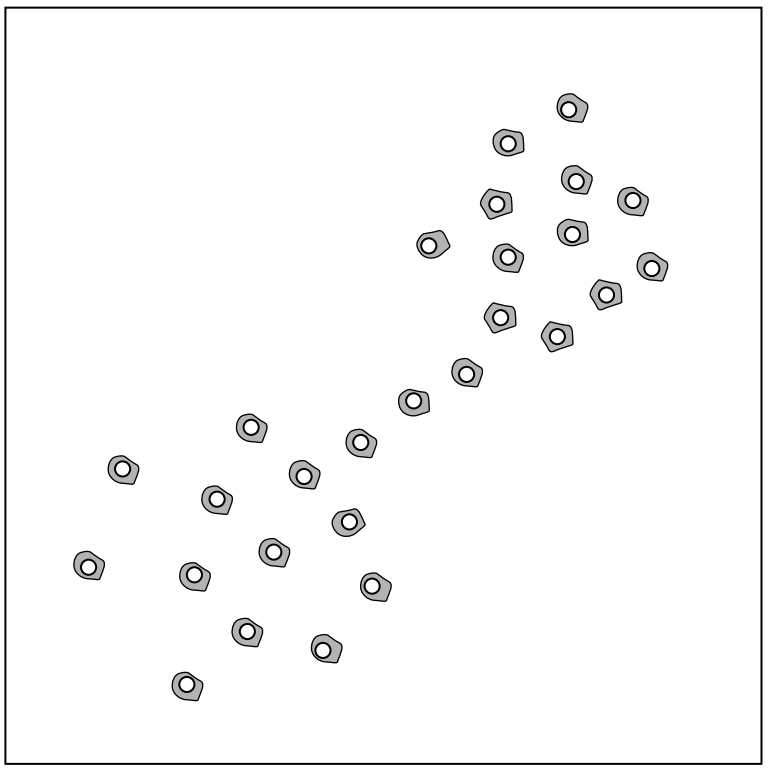


Remarks:

- 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.

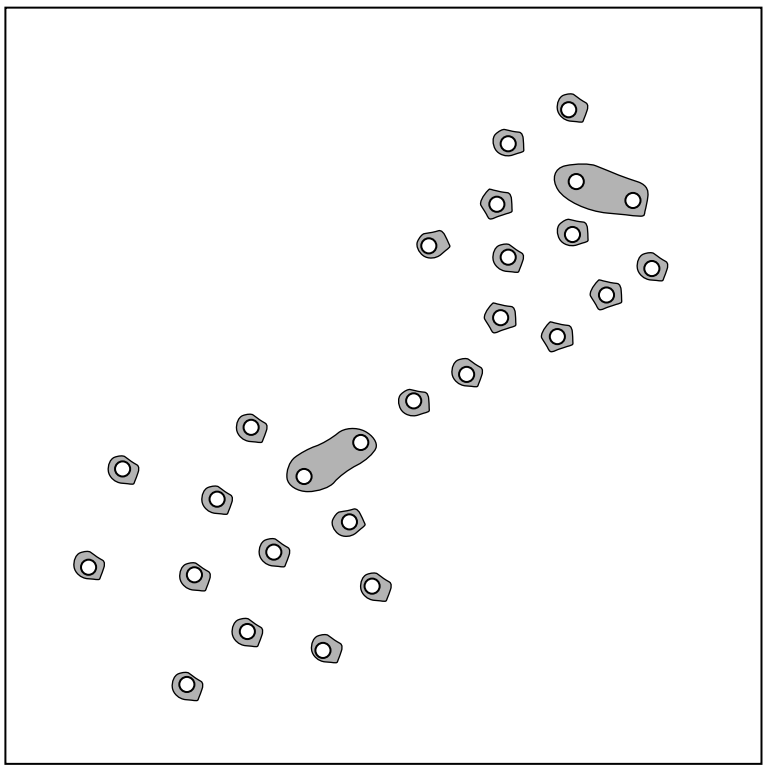
Hierarchical Cluster Analysis

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



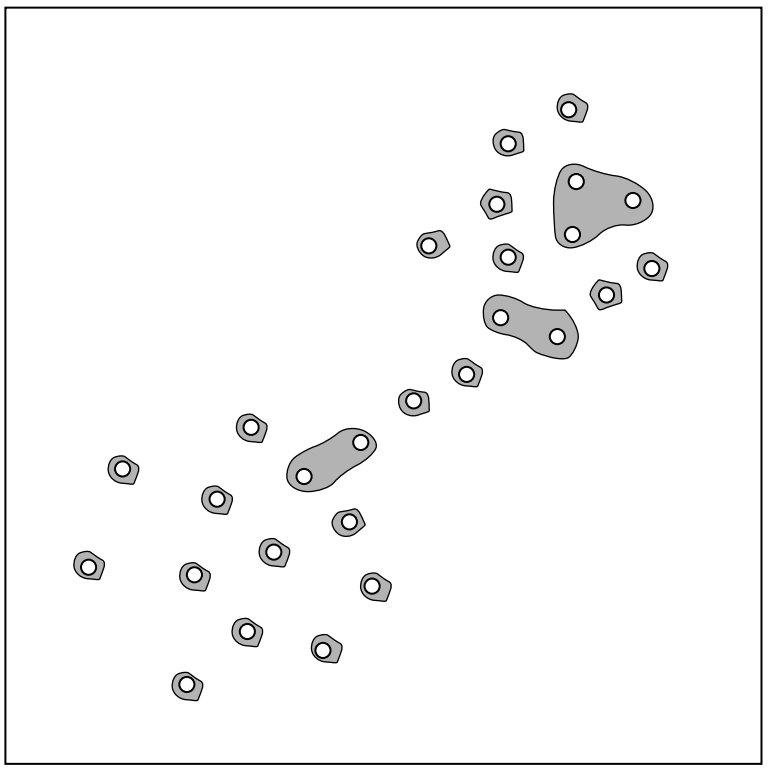
Hierarchical Cluster Analysis

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



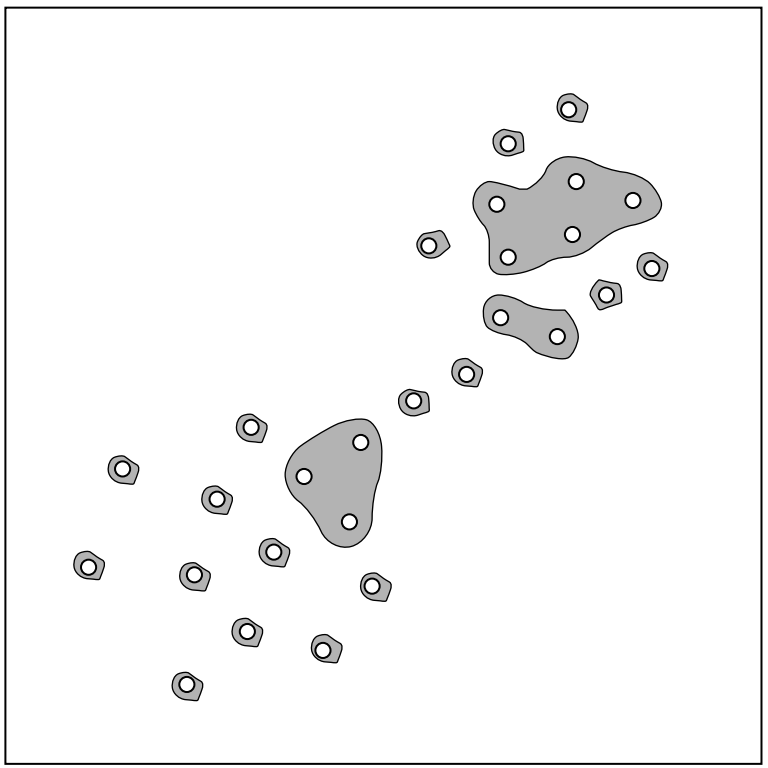
Hierarchical Cluster Analysis

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



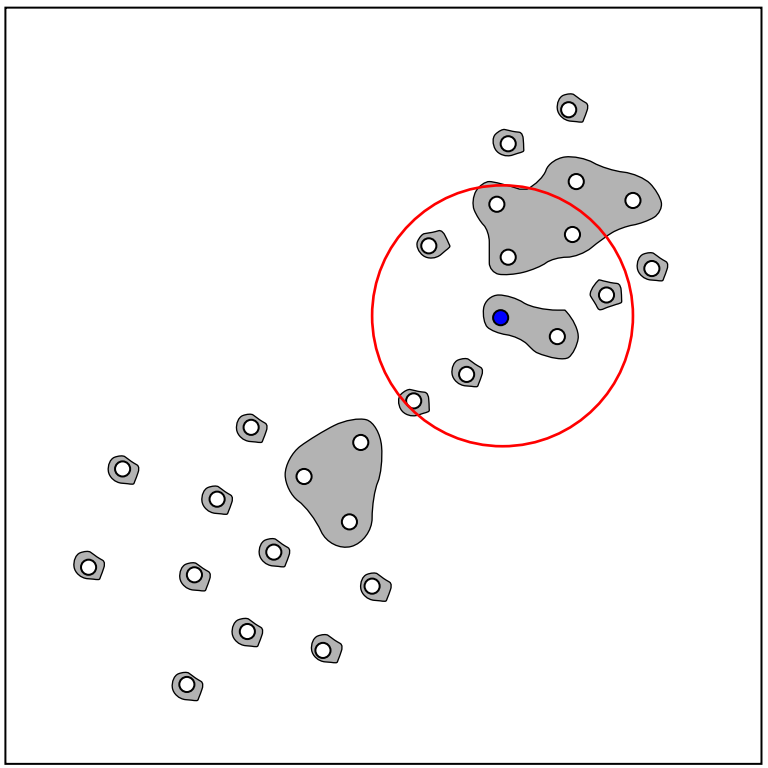
Hierarchical Cluster Analysis

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



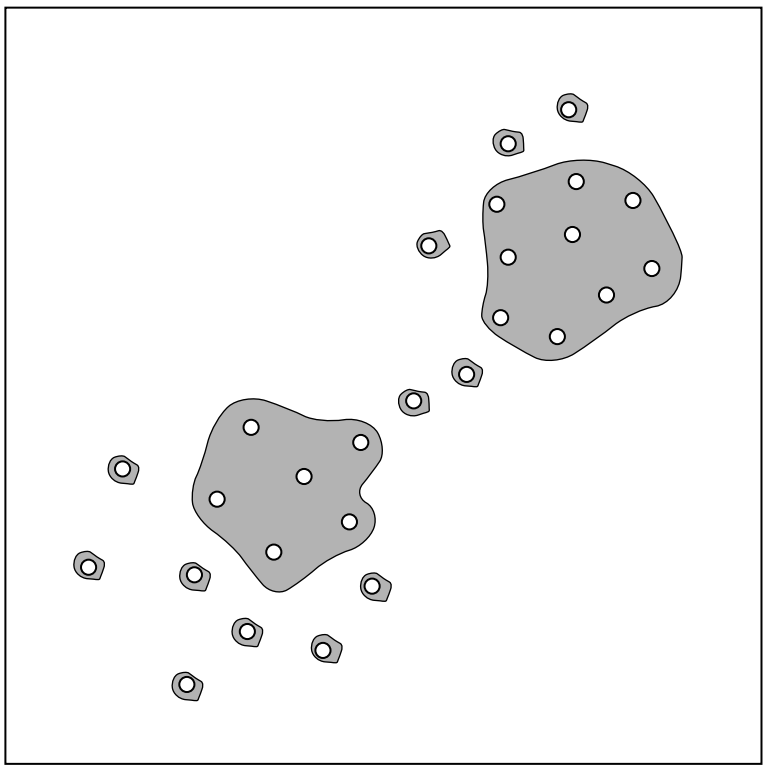
Hierarchical Cluster Analysis

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



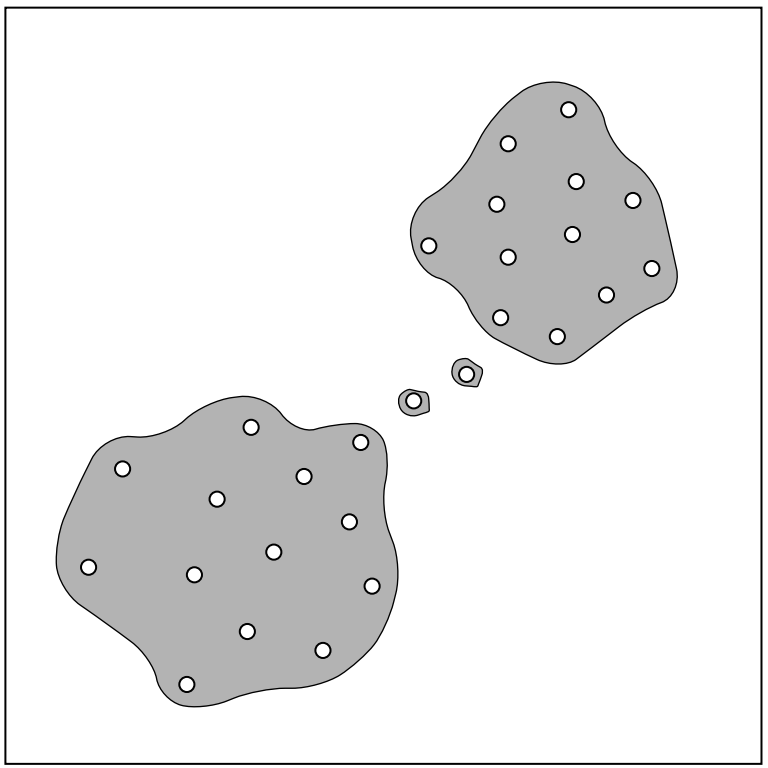
Hierarchical Cluster Analysis

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



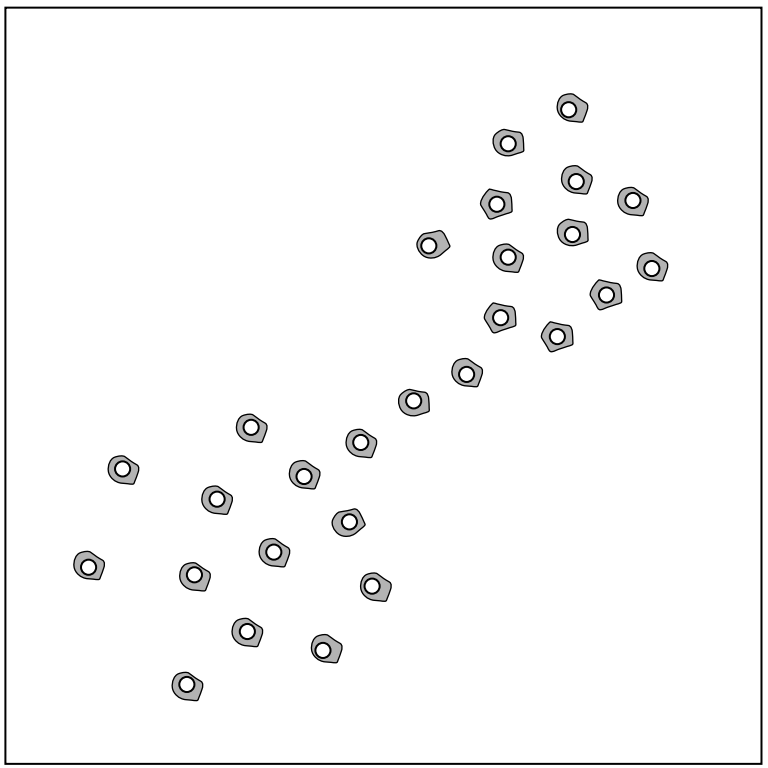
Hierarchical Cluster Analysis

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



Hierarchical Cluster Analysis

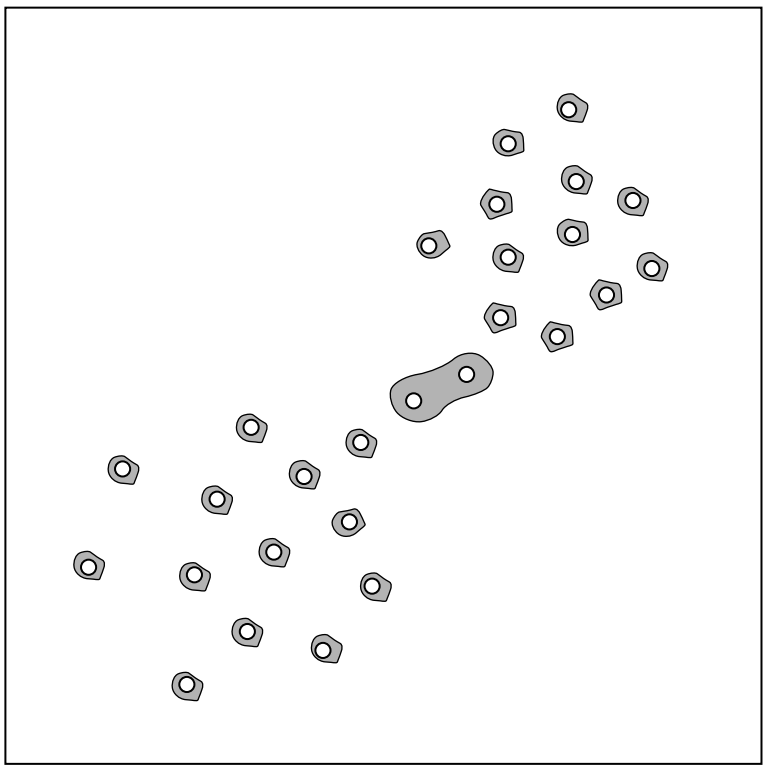
Chaining Problem of Single Link ($d_{\mathcal{C}}$ = k-Nearest-Neighbor)



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

Hierarchical Cluster Analysis

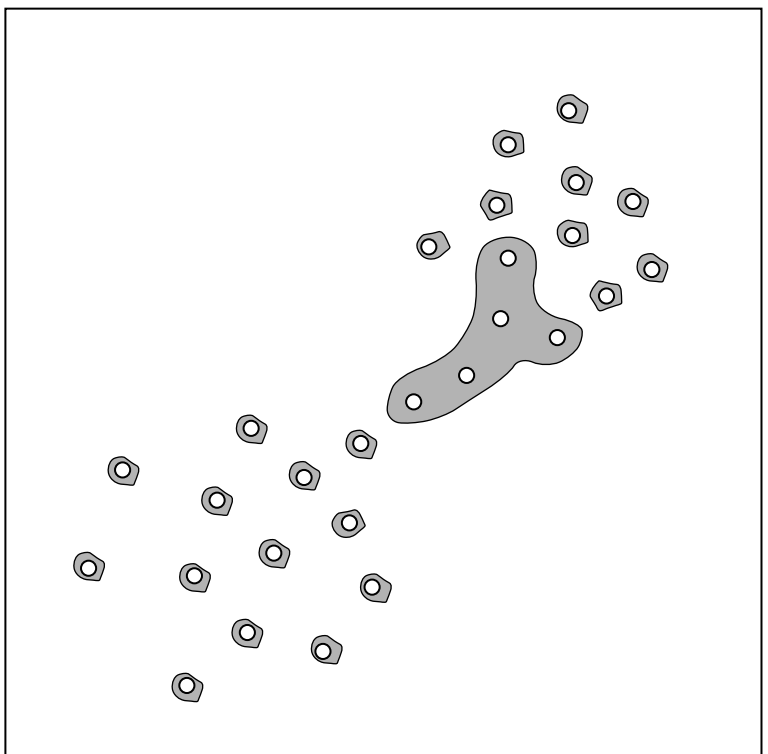
Chaining Problem of Single Link ($d_{\mathcal{C}}$ = k-Nearest-Neighbor)



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

Hierarchical Cluster Analysis

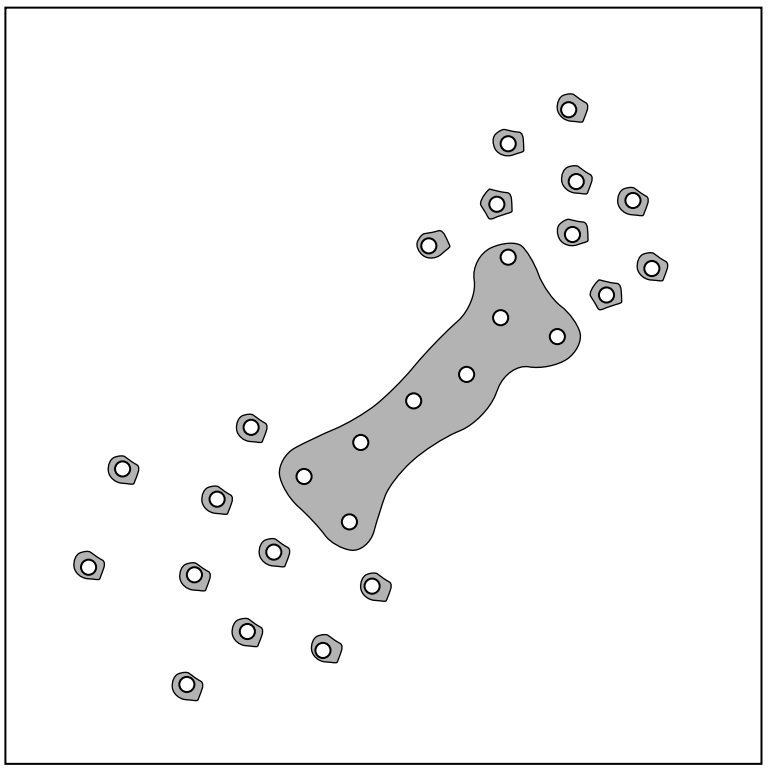
Chaining Problem of Single Link ($d_{\mathcal{C}}$ = k-Nearest-Neighbor)



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

Hierarchical Cluster Analysis

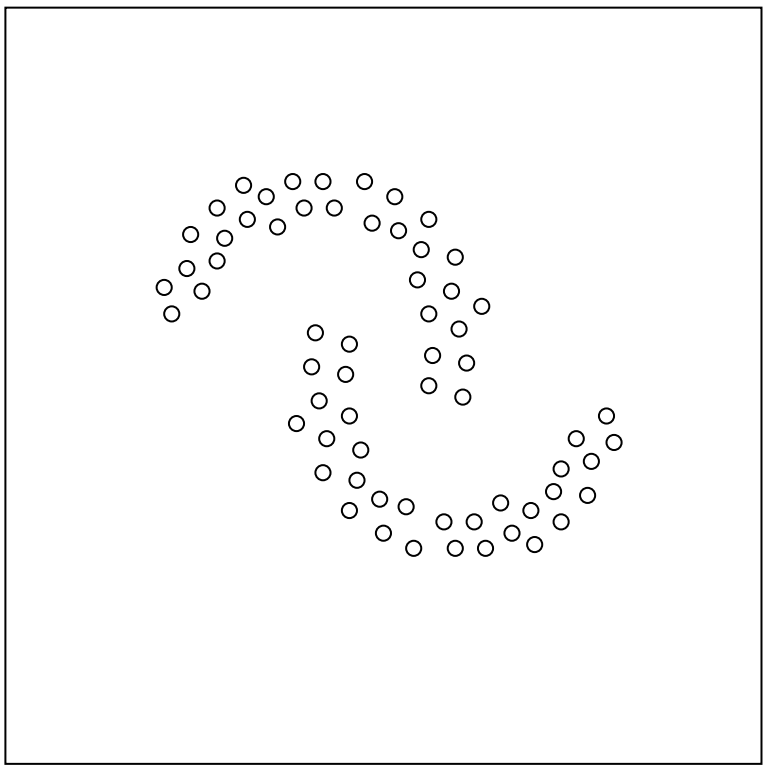
Chaining Problem of Single Link ($d_{\mathcal{C}}$ = k-Nearest-Neighbor)



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

Hierarchical Cluster Analysis

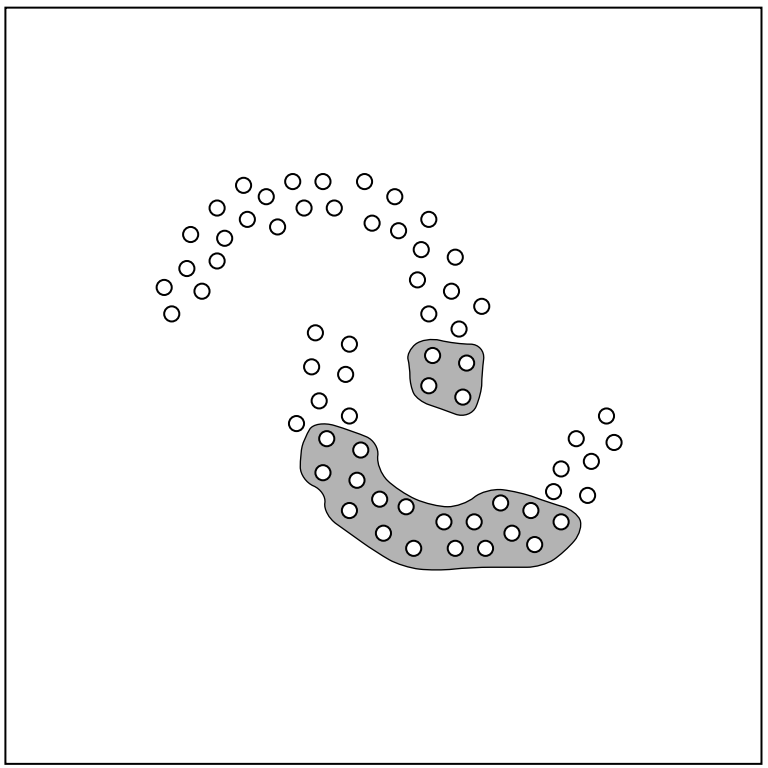
Nesting Problem of Complete Link ($d_{\mathcal{C}}$ = Furthest Neighbor)



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

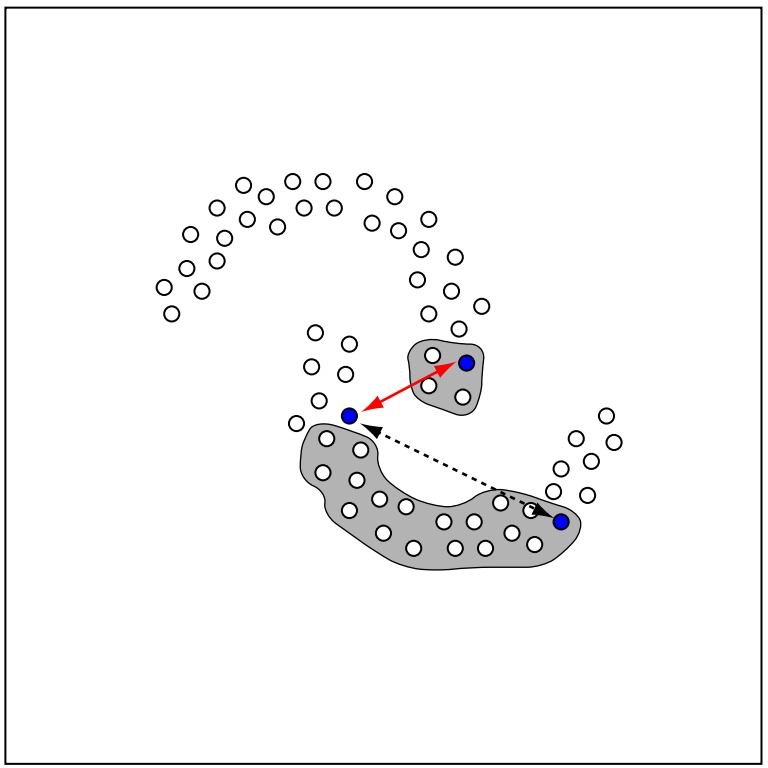
Hierarchical Cluster Analysis

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



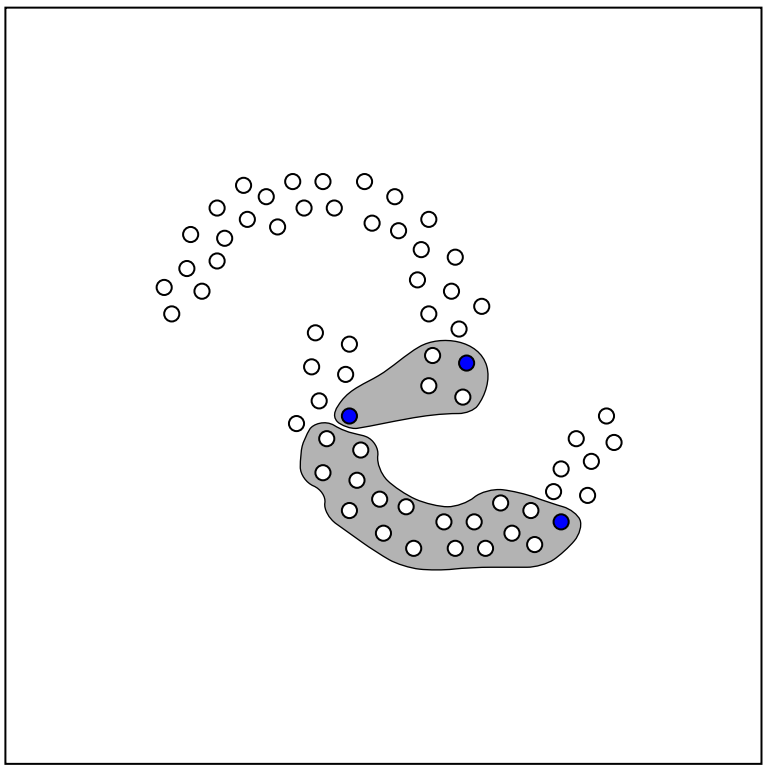
Hierarchical Cluster Analysis

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



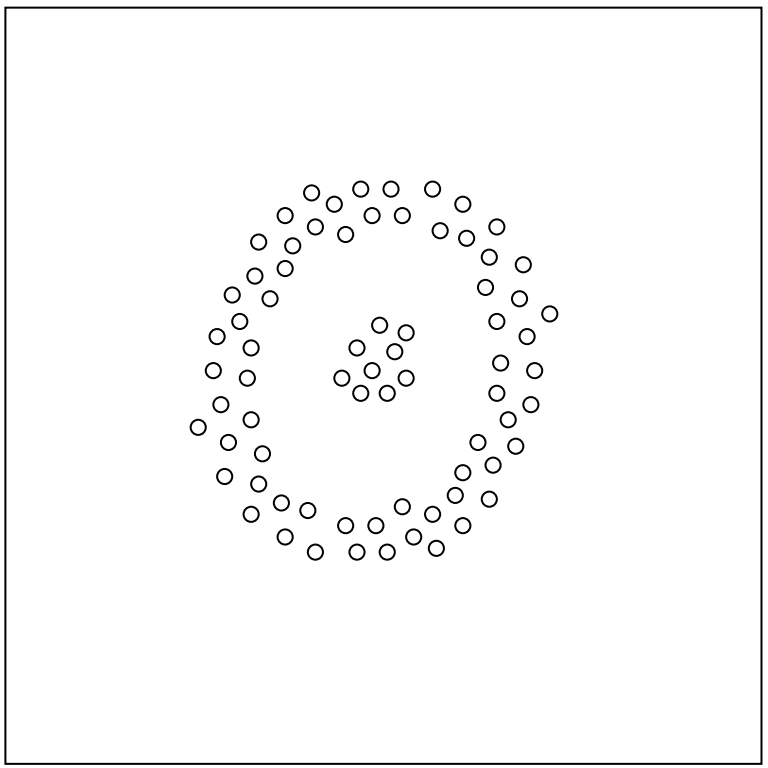
Hierarchical Cluster Analysis

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



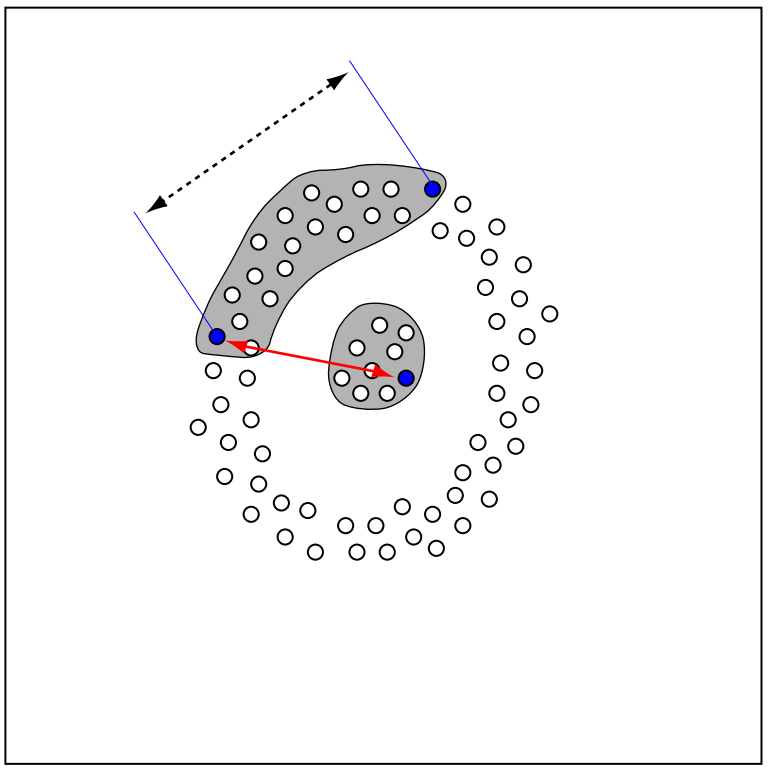
Hierarchical Cluster Analysis

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



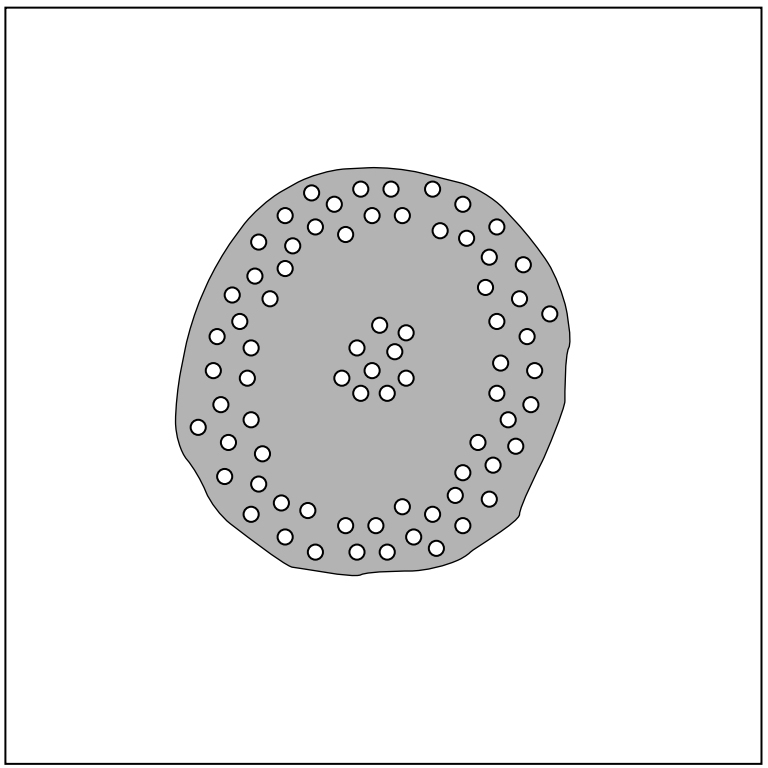
Hierarchical Cluster Analysis

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



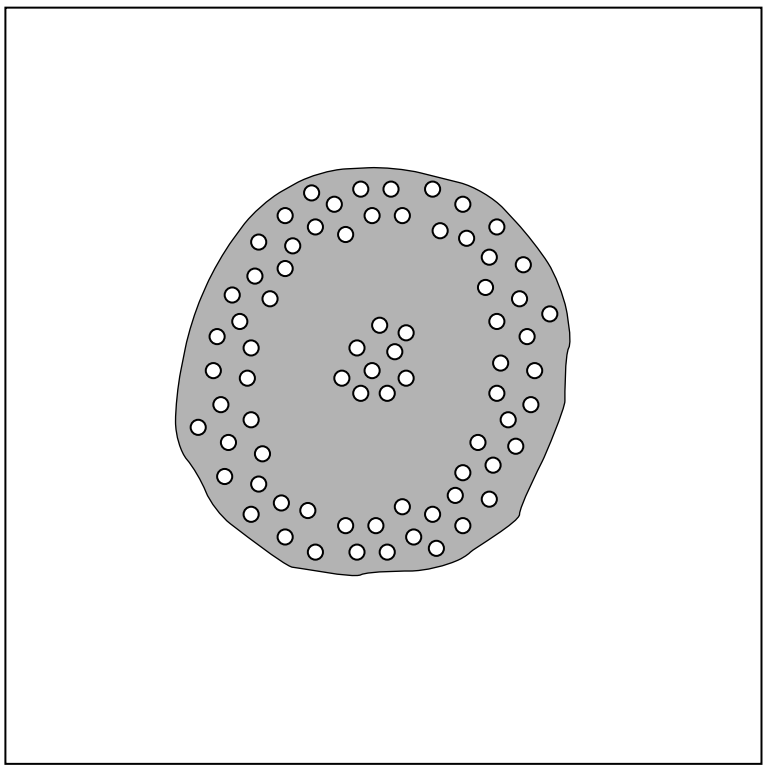
Hierarchical Cluster Analysis

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

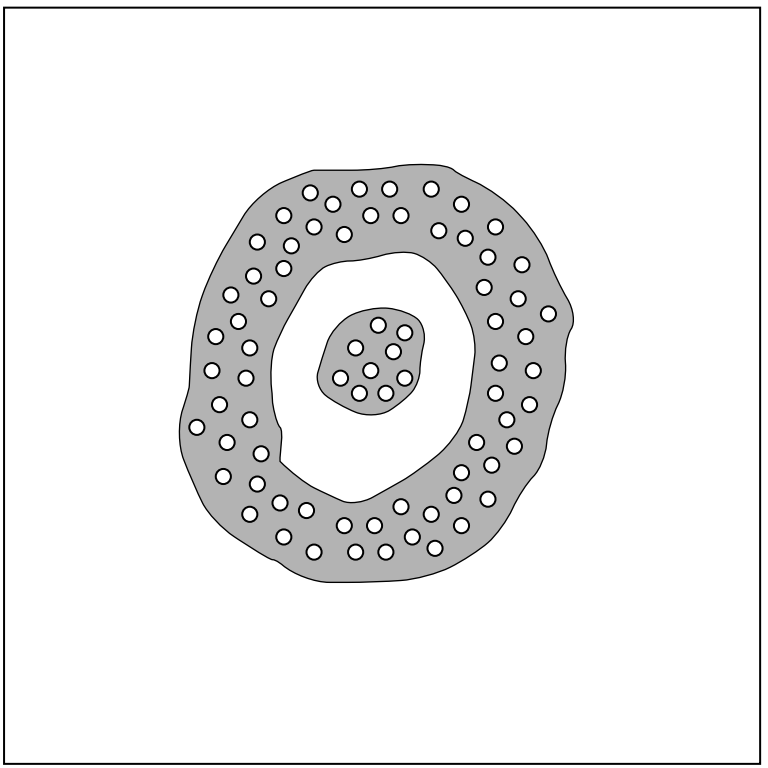


Hierarchical Cluster Analysis

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 $d_{\mathcal{C}}$:

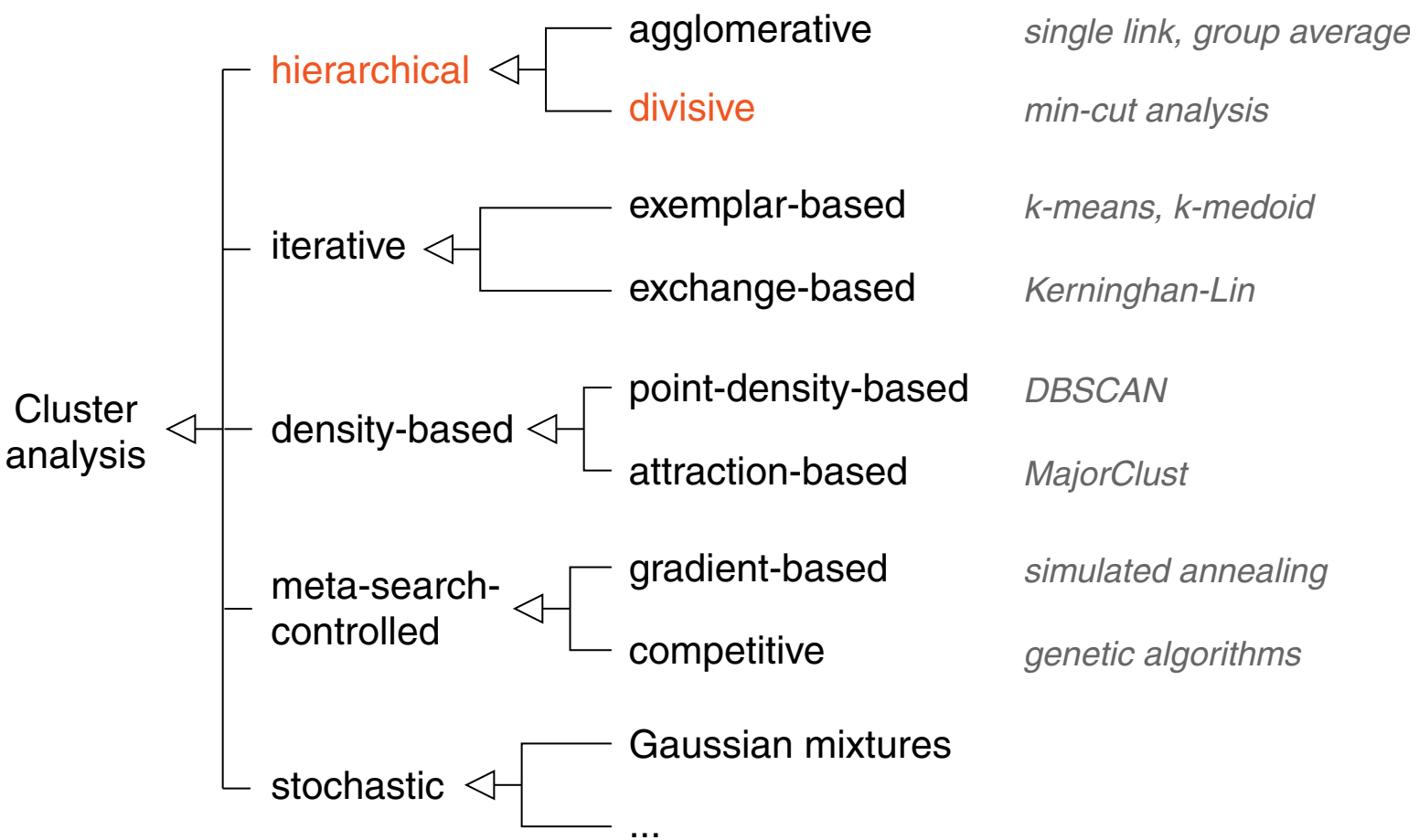
monotonicity	✓	✓	✓	✓
order dependence	✓	✓	✓	✓
consistency	$\longrightarrow 0$	$\longrightarrow \infty$	✓	$\longrightarrow \infty$

Remarks:

- The previous table also shows the usage frequency of the algorithms: single link and complete link are the most popular hierarchical agglomerative algorithms.
- 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.
- Chaining will also happen when the median distance is employed.
- The median distance and is not a monotonic cluster distance measure.

Hierarchical Cluster Analysis

Merging Principles



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.
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.
7. **ENDDO**
8. **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. $V_T = \{v_C \mid C \in \mathcal{C}\}, \quad E_T = \emptyset$ // initial dendrogram
3. **WHILE** $\exists C_x : (C_x \in \mathcal{C} \quad \wedge \quad |C_x| > 1)$ **DO**
4. $\{C, C'\} = \underset{\substack{\{C_i, C_j\}: \\ C_i \cup C_j = C_x \quad \wedge \quad C_i \cap C_j = \emptyset}}{\text{argmax}} d_{\mathcal{C}}(C_i, C_j)$
5. $\mathcal{C} = (\mathcal{C} \setminus \{C_x\}) \quad \cup \quad \{C, C'\}$ // splitting
6. $V_T = V_T \cup \{v_C, v_{C'}\}, \quad E_T = E_T \cup \{\{v_{C_x}, v_C\}, \{v_{C_x}, v_{C'}\}\}$ // 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 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 that occurred during the first iterations.
- A powerful hierarchical divisive algorithm is given with

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

Hierarchical Cluster Analysis

MinCut Cluster Analysis

Definition 4 (Cut, Minimum Cut)

Let $G = \langle V, E, w \rangle$ be a graph with a non-negative weight function w. Moreover, 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 Cluster Analysis

Definition 4 (Cut, Minimum Cut)

Let $G = \langle V, E, w \rangle$ be a graph with a non-negative weight function w. Moreover, 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:

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

Moreover, let $w(\{U, \bar{U}\})$ denote the weight (or the capacity) of $cut(\{U, \bar{U}\})$:

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

Hierarchical Cluster Analysis

MinCut Cluster Analysis

Definition 4 (Cut, Minimum Cut)

Let $G = \langle V, E, w \rangle$ be a graph with a non-negative weight function w. Moreover, 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:

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

Moreover, let $w(\{U, \bar{U}\})$ denote the weight (or the capacity) of $cut(\{U, \bar{U}\})$:

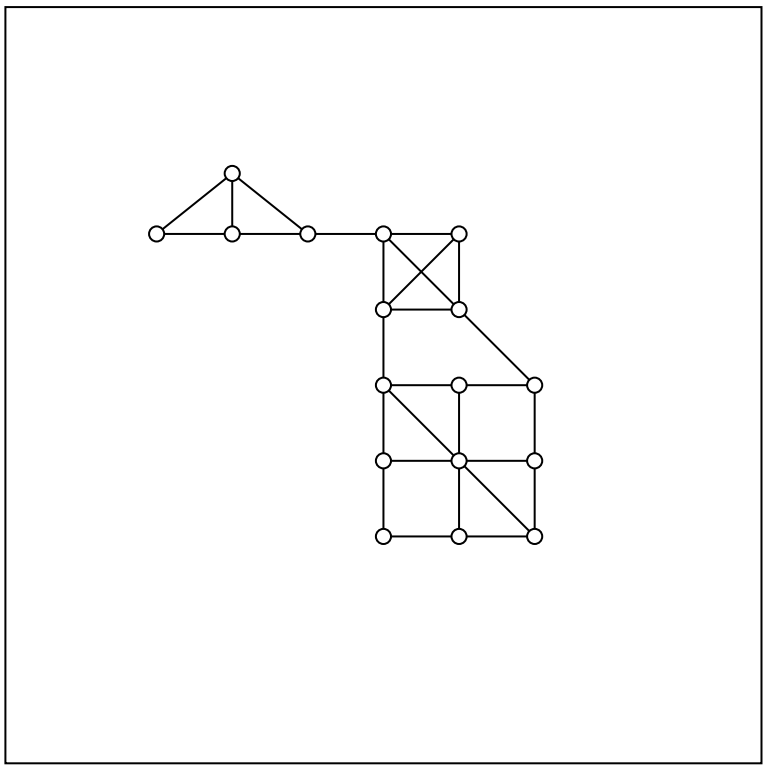
$$w(\{U, \bar{U}\}) = \sum_{e \in cut(\{U, \bar{U}\})} w(e)$$

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

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

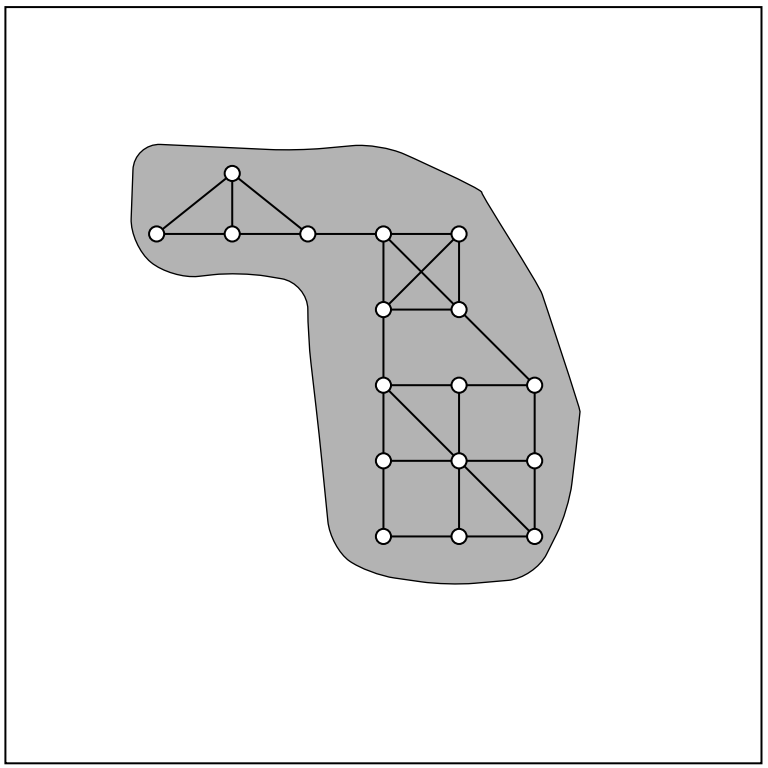
Hierarchical Cluster Analysis

MinCut Cluster Analysis



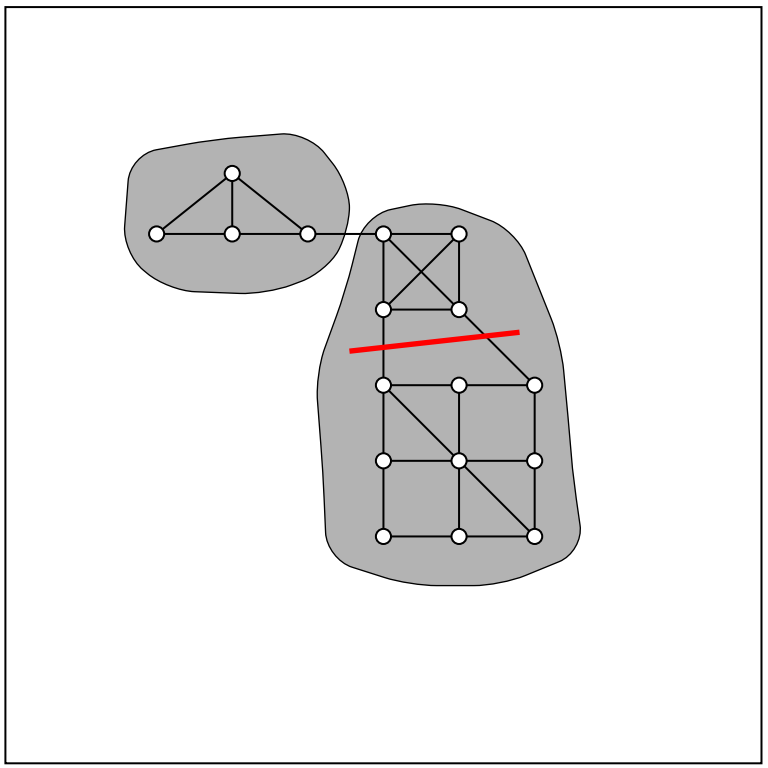
Hierarchical Cluster Analysis

MinCut Cluster Analysis



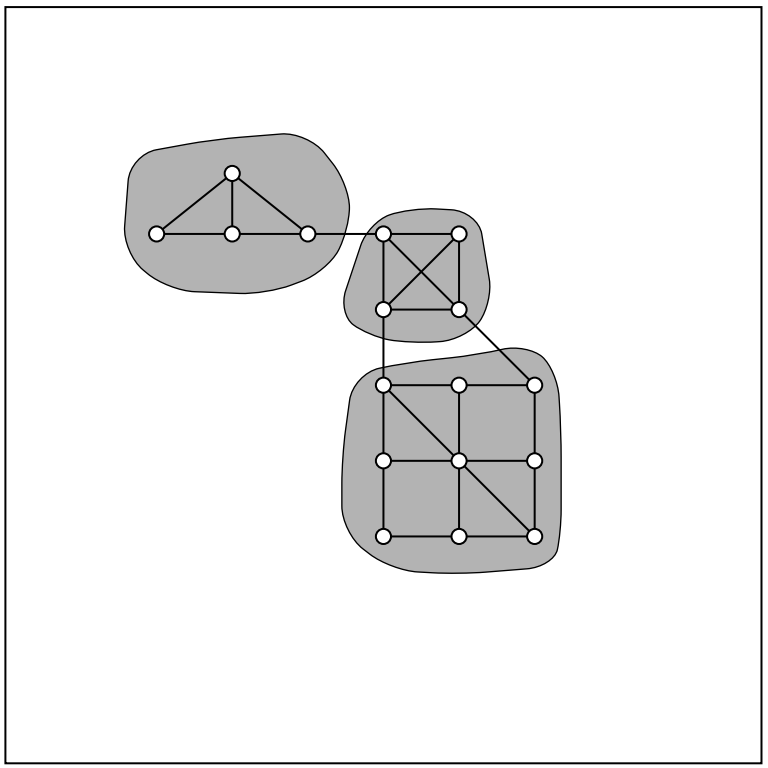
Hierarchical Cluster Analysis

MinCut Cluster Analysis



Hierarchical Cluster Analysis

MinCut Cluster Analysis

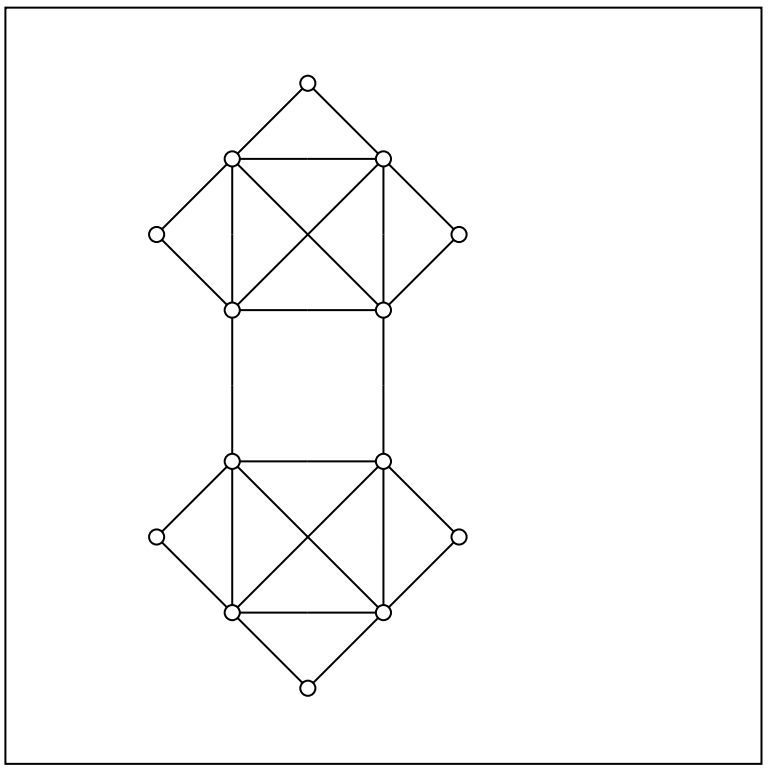


Remarks:

- 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 cluster analysis is not classified as a hierarchical algorithm.

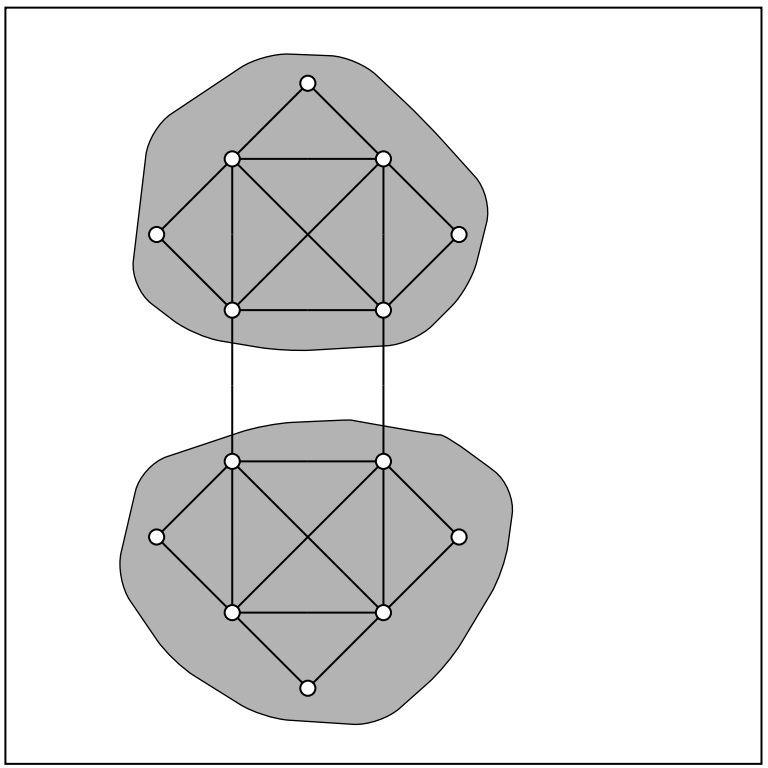
Hierarchical Cluster Analysis

Splitting Problem of the MinCut Cluster Analysis



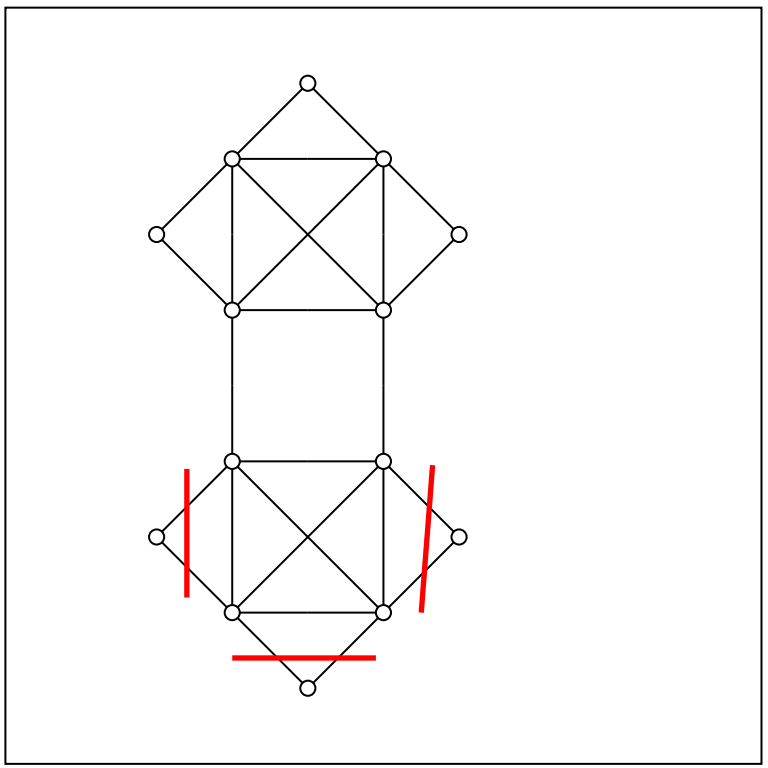
Hierarchical Cluster Analysis

Splitting Problem of the MinCut Cluster Analysis



Hierarchical Cluster Analysis

Splitting Problem of the MinCut Cluster Analysis



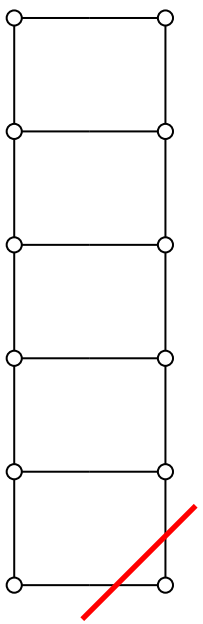
Solution: Normalization of the cut capacity with regard to the node number.

Hierarchical Cluster Analysis

Splitting Problem of the MinCut Cluster Analysis

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 $\overline{w}$:



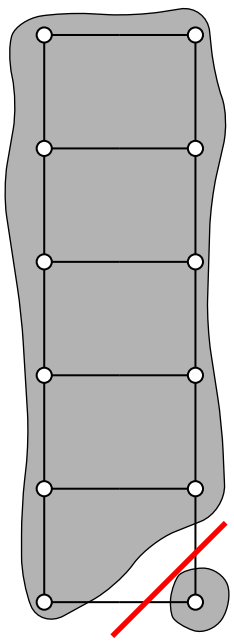
$cut(\{U,\bar{U}\}) = \{\{u,v\} \mid \{u,v\} \in E, u \in U, v \in \bar{U}\}, \qquad w(\{U,\bar{U}\}) = \sum_{e \in cut(\{U,\bar{U}\})} w(e)$

Hierarchical Cluster Analysis

Splitting Problem of the MinCut Cluster Analysis

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 $\overline{w}$:



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

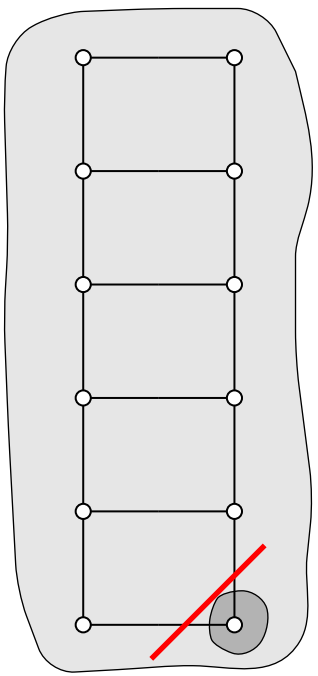
$cut(\{U, \bar{U}\}) = \{\{u, v\} \mid \{u, v\} \in E, u \in U, v \in \bar{U}\}, \qquad w(\{U, \bar{U}\}) = \sum_{e \in cut(\{U, \bar{U}\})} w(e)$

Hierarchical Cluster Analysis

Splitting Problem of the MinCut Cluster Analysis

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 $\overline{w}$:



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

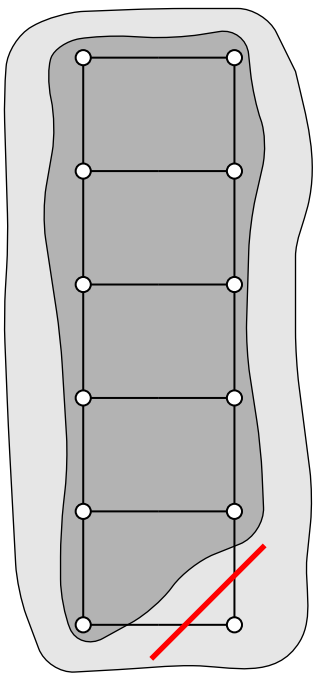
$cut(\{U,\bar{U}\}) = \{\{u,v\} \mid \{u,v\} \in E, u \in U, v \in \bar{U}\}, \qquad w(\{U,\bar{U}\}) = \sum_{e \in cut(\{U,\bar{U}\})} w(e)$

Hierarchical Cluster Analysis

Splitting Problem of the MinCut Cluster Analysis

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 $\overline{w}$:



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

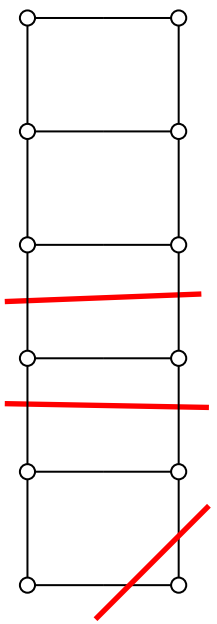
$cut(\{U,\bar{U}\}) = \{\{u,v\} \mid \{u,v\} \in E, u \in U, v \in \bar{U}\}, \qquad w(\{U,\bar{U}\}) = \sum_{e \in cut(\{U,\bar{U}\})} w(e)$

Hierarchical Cluster Analysis

Splitting Problem of the MinCut Cluster Analysis

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 $\overline{w}$:



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

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

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

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

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

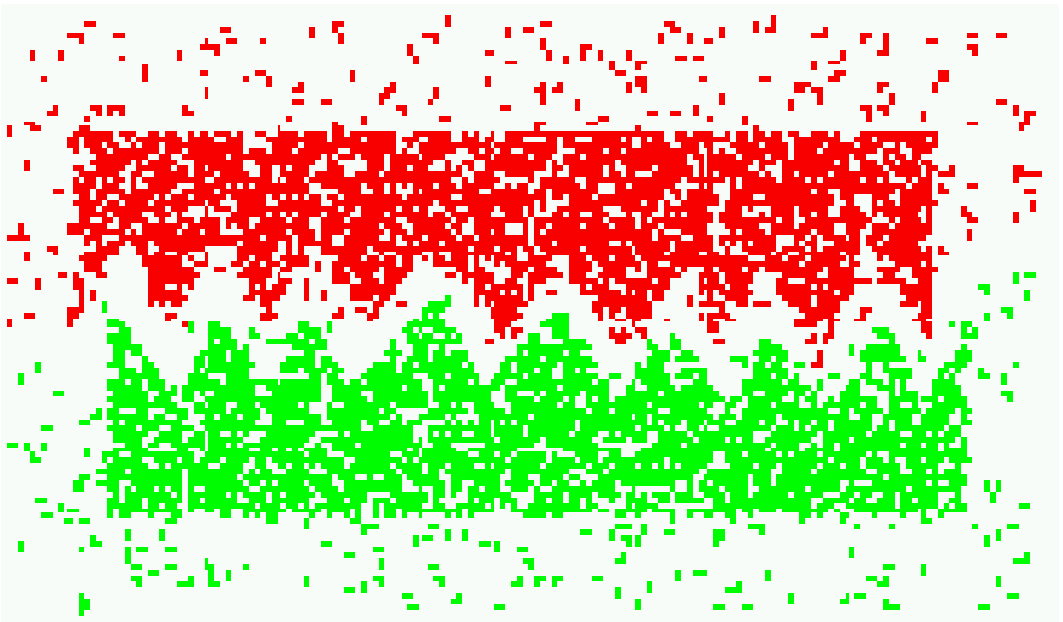
Combination of Hierarchical Algorithms

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] :

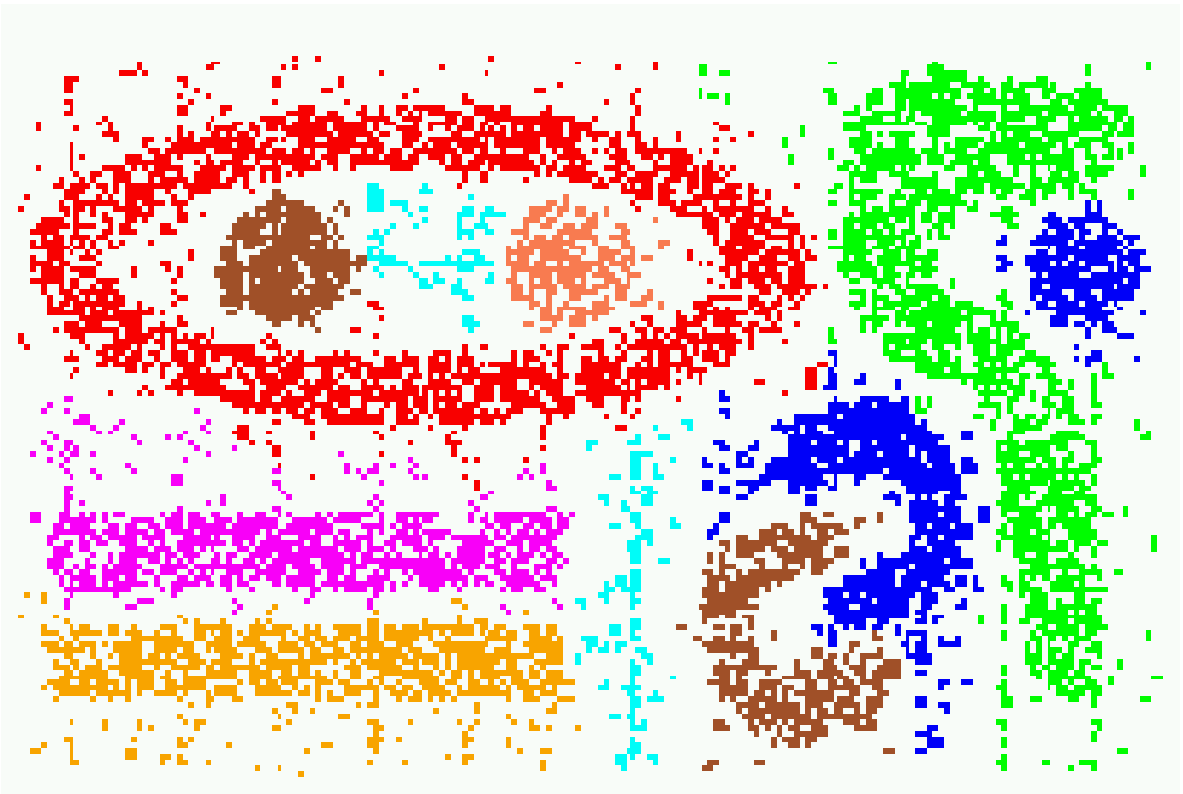


The cluster distance $d_{\mathcal{C}}(C, C')$ is defined as $d_{\mathcal{C}} = \dfrac{1}{R_I(C, C') \cdot (R_C(C, C'))^{\alpha}}$

Hierarchical Cluster Analysis

Combination of Hierarchical Algorithms

Chameleon [Karypis/Han/Kumar 2000] :



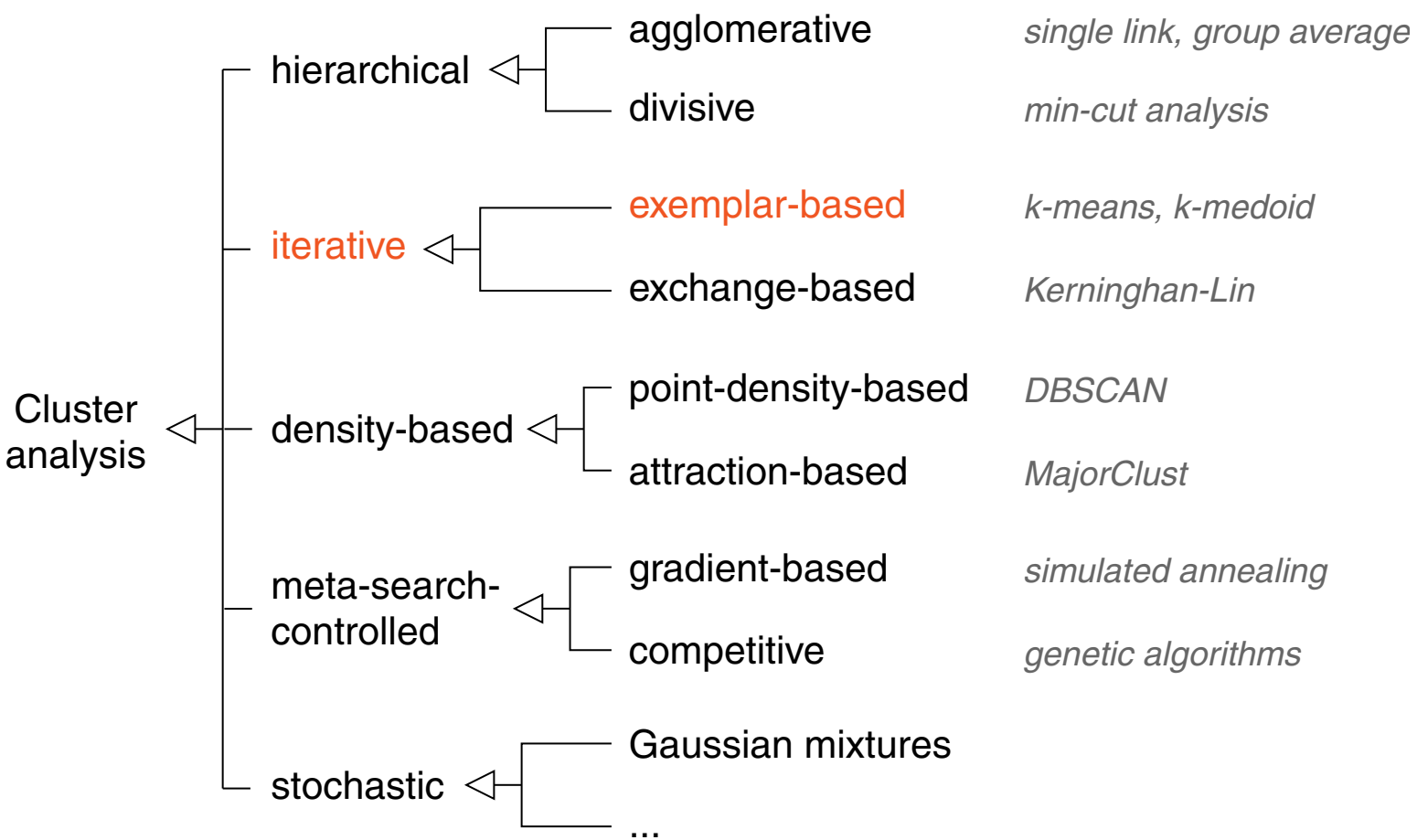
The parameter α in $d_{\mathcal{C}}$ is task-depending and has to be determined (via trial and error) by the user.

Chapter ML:XI (continued)

XI. Cluster Analysis

Iterative Cluster Analysis

Merging Principles



Iterative Cluster Analysis

Exemplar-Based Algorithm

Input: $G = \langle V, E, w \rangle$. Weighted graph.
d. Distance measure for two 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.
2.  FOR i = 1 to k DO r_i(t) = choose(V) // init representatives
3.
4.
5.
6.    FOREACH v ∈ V DO // find nearest representative (cluster)
7.      i = argmin_{j: j∈{1,...,k}} d(r_j(t), v), C_i = C_i ∪ {v}
8.    ENDDO
9.    FOR i = 1 to k DO r_i(t) = minimize(e(C_i)) // update
10.
11.
```

Iterative Cluster Analysis

Exemplar-Based Algorithm

Input: $G = \langle V, E, w \rangle$. Weighted graph.
d. Distance measure for two 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.

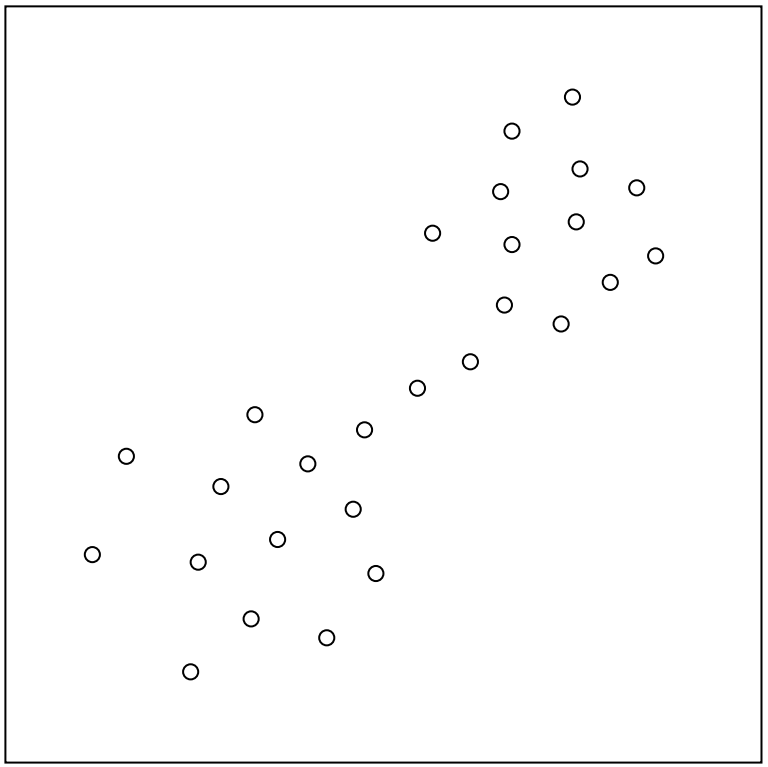
1. $t = 0$
2. **FOR** $i = 1$ to k **DO** $r_i(t) =$ *choose*(V) // init representatives
3. **REPEAT**
4. $t = t + 1$
5. **FOR** $i = 1$ to k **DO** $C_i = \emptyset$
6. **FOREACH** $v \in V$ **DO** // find nearest representative (cluster)
7. $i = \underset{j: j \in \{1,\ldots,k\}}{\text{argmin}} \; d(r_j(t), v)$, $C_i = C_i \cup \{v\}$
8. **ENDDO**
9. **FOR** $i = 1$ to k **DO** $r_i(t) =$ *minimize*$(e(C_i))$ // update
10. **UNTIL** (*convergence*$(r_1(t), \ldots, r_k(t))$ **OR** $t > t_{\text{max}}$)
11. **RETURN**$(\{r_1(t), \ldots, r_k(t)\})$

Remarks:

- The cluster representatives are called centroids or, more general, medoids.
- The function *choose*(V) operationalizes a random sampling without replacement (in German: “zufälliges Ziehen ohne Zurücklegen”).
- If the data is from a metric space, then as distance function d the Euclidean distance between two data points is usually chosen. An alternative and more general approach is to choose the shortest path between two points in G.
- If the data is from a metric space, then as minimization criterion e the sum of the squared distances to the cluster representatives (= variance criterion) is usually chosen: For points $v \in V$ from $\mathbf{R}^p$, the components of the optimum cluster representative (= vector of minimum variance) are given by the component-wise arithmetic mean of the points in the cluster.

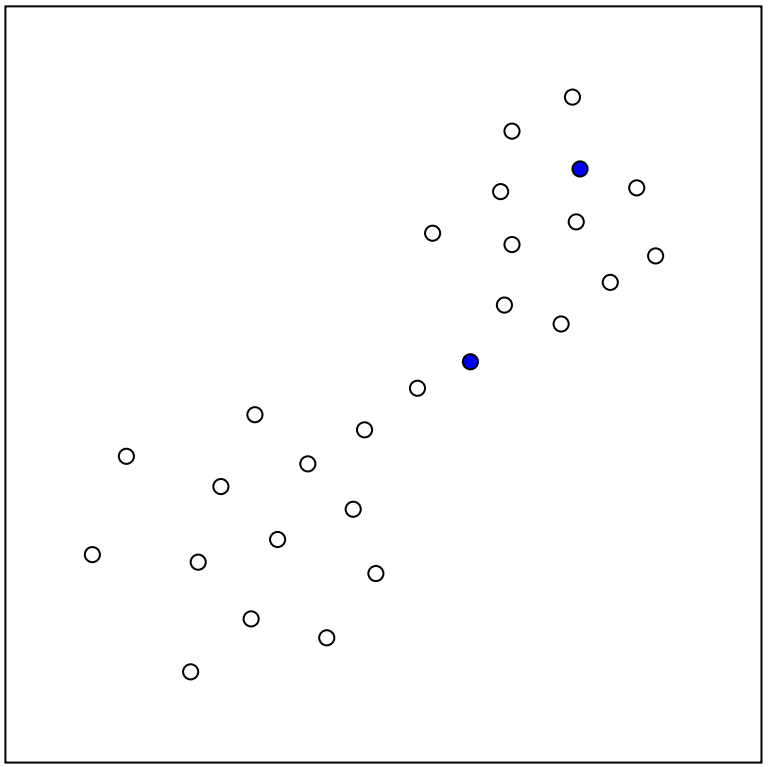
Iterative Cluster Analysis

k-Means with Minimization Criterion e = Variance



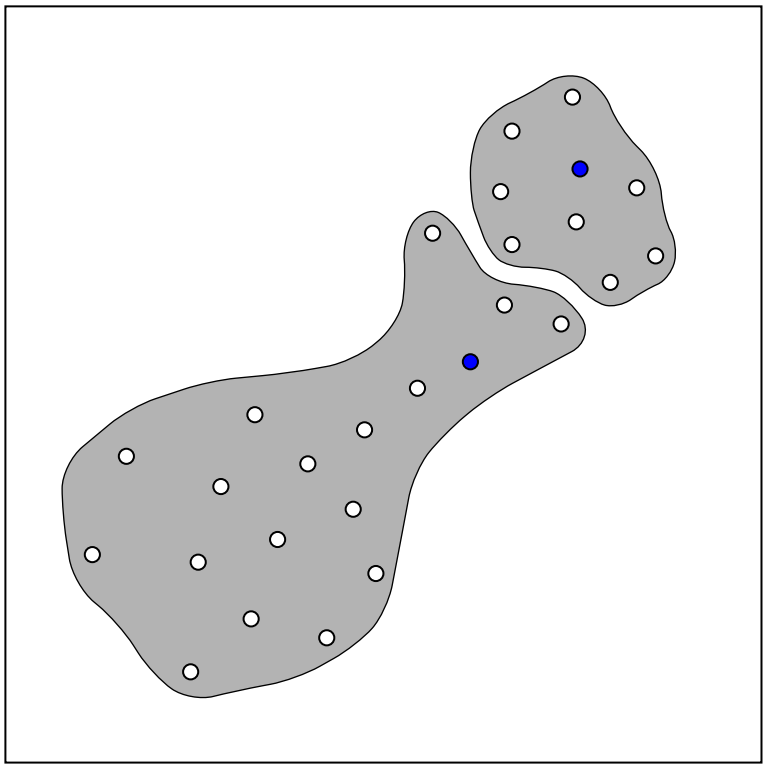
Iterative Cluster Analysis

k-Means with Minimization Criterion e = Variance



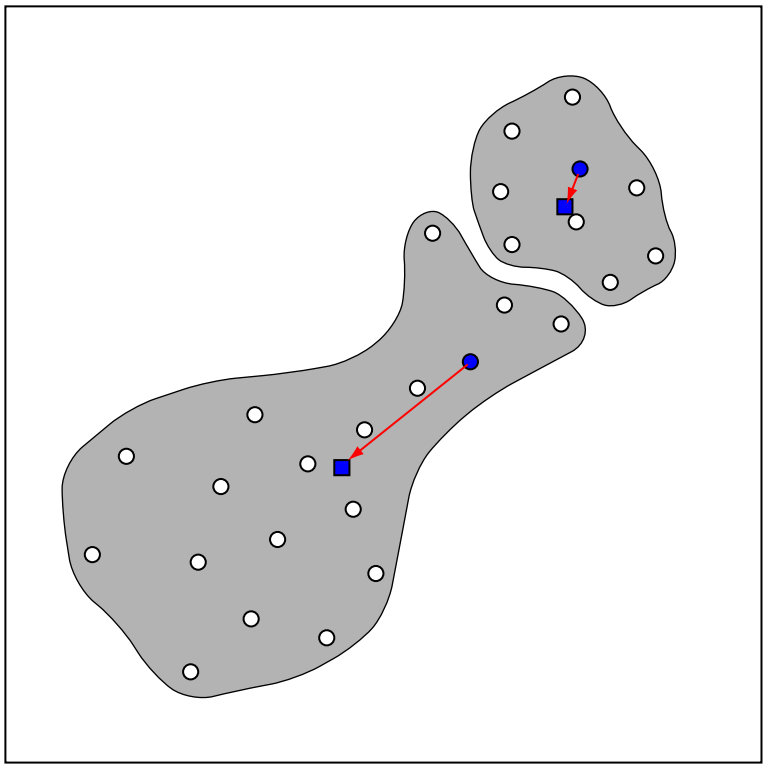
Iterative Cluster Analysis

k-Means with Minimization Criterion e = Variance



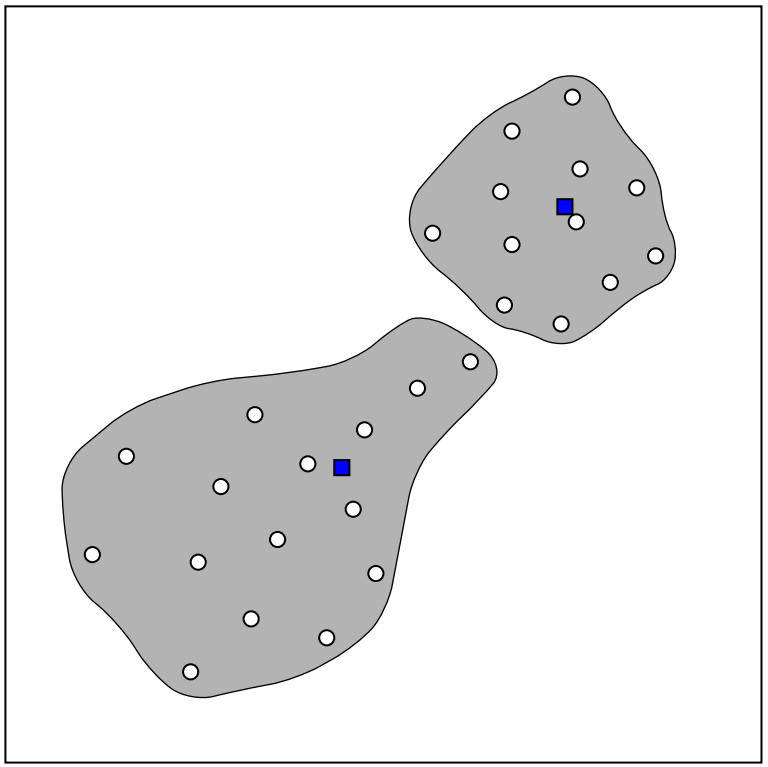
Iterative Cluster Analysis

k-Means with Minimization Criterion e = Variance



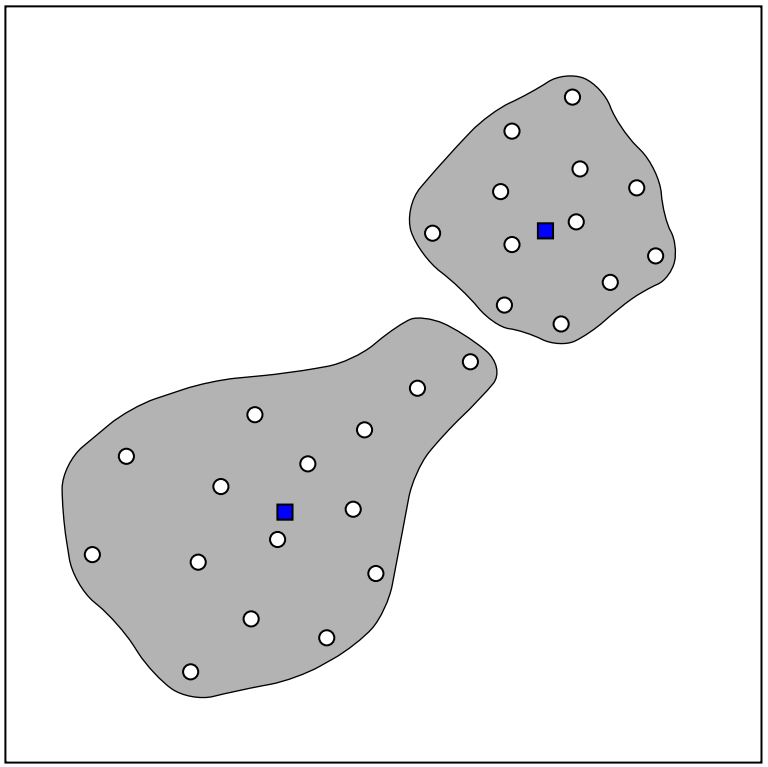
Iterative Cluster Analysis

k-Means with Minimization Criterion e = Variance



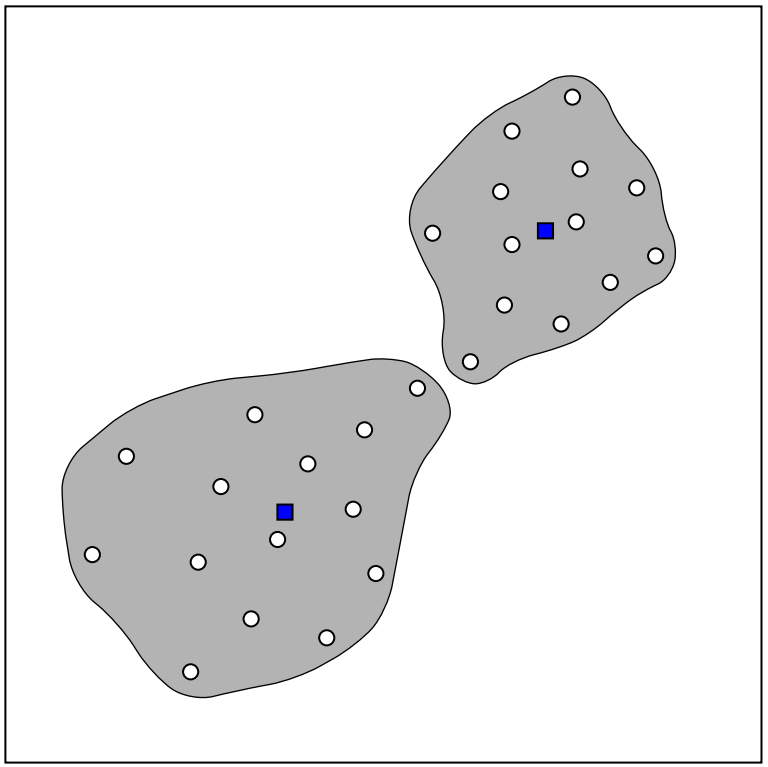
Iterative Cluster Analysis

k-Means with Minimization Criterion e = Variance



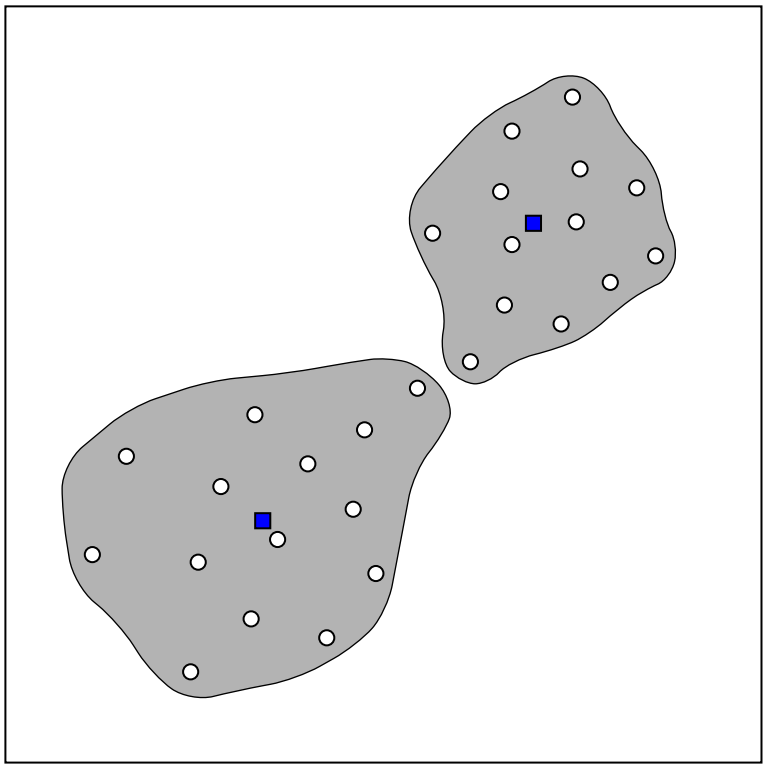
Iterative Cluster Analysis

k-Means with Minimization Criterion e = Variance



Iterative Cluster Analysis

k-Means with Minimization Criterion e = Variance



Iterative Cluster Analysis

Minimization Criteria of Exemplar-Based Algorithms

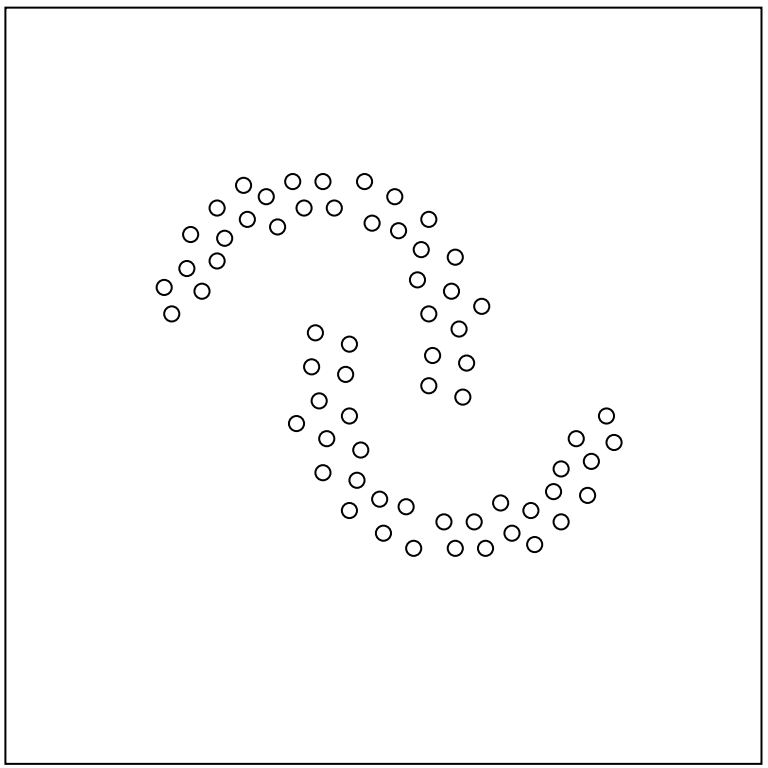
$e(C_i) = \sum_{v \in C_i} (v - r_i)^2$	$r_i = \bar{v}_{C_i}$	centroid computation via variance minimization (k-means)
$e(C_i) = \sum_{v \in C_i} \lvert v - r_i \rvert$	$r_i \in C_i$	medoid computation (k-medoid)
$e(C_i) = \max_{v \in C_i} \lvert v - r_i \rvert$	$r_i \in C_i$	k-center
$e(C_i) = \sum_{v \in V} (\mu_i(v))^2 \cdot (v - r_i)^2$	$r_i = \frac{\sum_{v \in V} (\mu_i(v))^2 \cdot v}{\sum_{v \in V} (\mu_i(v))^2}$	Fuzzy k-means

Remarks:

- $\bar{v}_{C_i}$ denotes the arithmetic mean of the points $v \in C_i$.
- To simplify notation the cluster representative is denoted with r_i instead of with $r_i(t)$.
- The sum of the squared distances to a cluster representative r_i becomes minimum, if r_i is the arithmetic mean of the points in C_i. Hence, the computation of the centroid in k-means corresponds to a local—i.e., cluster-specific—minimization of the variance.
- The medoid or central element of a cluster denotes a point $r_i \in C_i$ that minimizes the sum of the distances from r_i to all other points in C_i. An advantage of medoids compared to centroids is their robustness with respect to outliers and, as a consequence, an improved convergence behavior (= smaller number of iterations).
- Within Fuzzy k-means, $\mu_i(v)$ denotes the membership value of the point $v \in V$ with respect to cluster C_i.
- k-medoid and k-center can employ arbitrary distance measures and similarity measures.
- k-means and Fuzzy k-means presume interval-based measurement scales for all features.
- k-means can be operationalized straightforward as Kohonen self-organizing map, SOM, a particular kind of neural network:
  - The SOM network is comprised of an input layer with p nodes, which correspond one-to-one to the features, and a so-called “competitive layer” with k nodes.
  - Based on the network’s current edge weights the training algorithm determines for a feature vector the so-called “winning neuron”, whose edge weights are raised according to a learning rate η.

Iterative Cluster Analysis

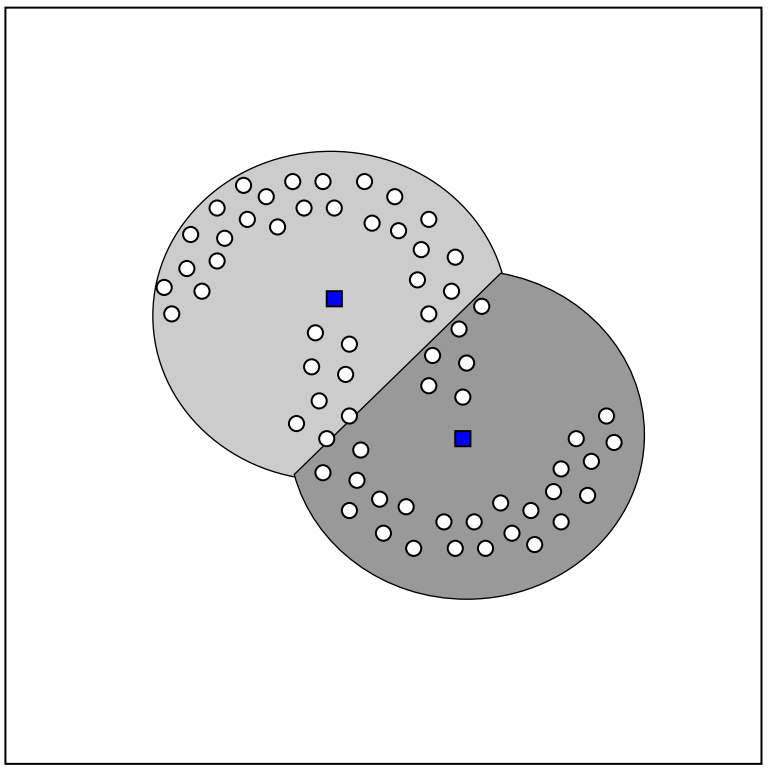
k-Means versus Single Link



Exemplar-based algorithms fail to detect nested clusters.

Iterative Cluster Analysis

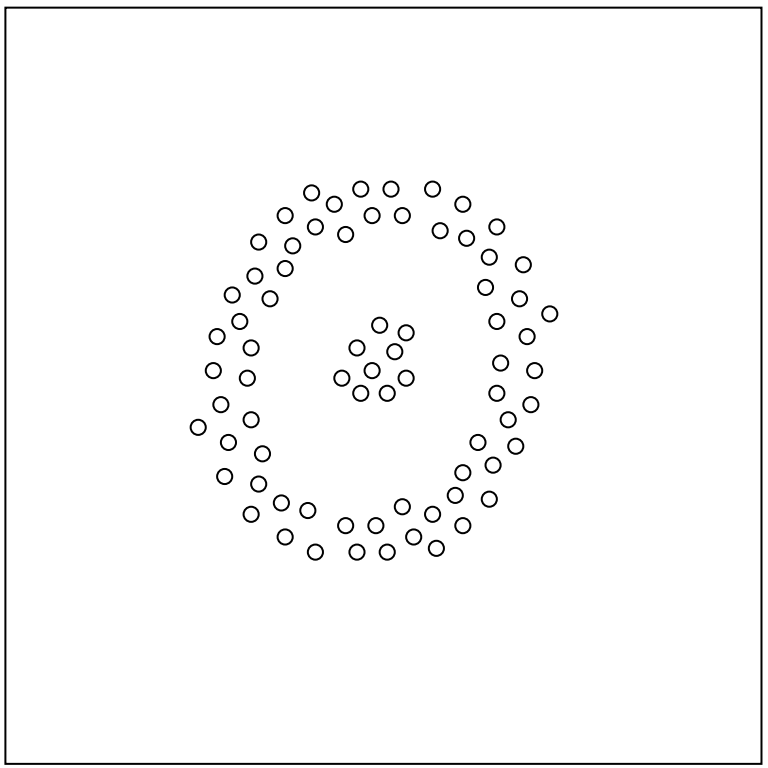
k-Means versus Single Link



Exemplar-based algorithms fail to detect nested clusters.

Iterative Cluster Analysis

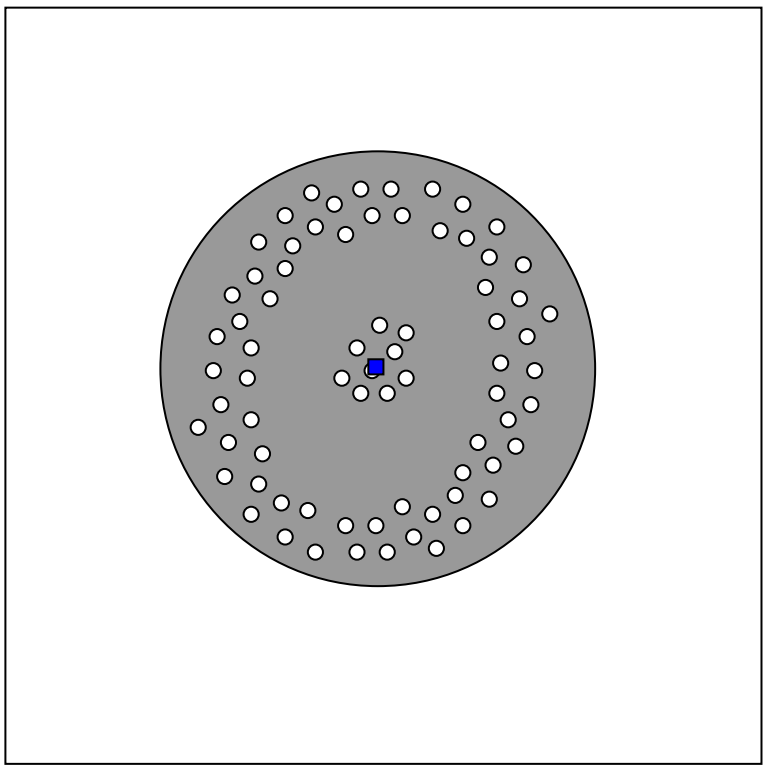
k-Means versus Single Link



Exemplar-based algorithms fail to detect nested clusters.

Iterative Cluster Analysis

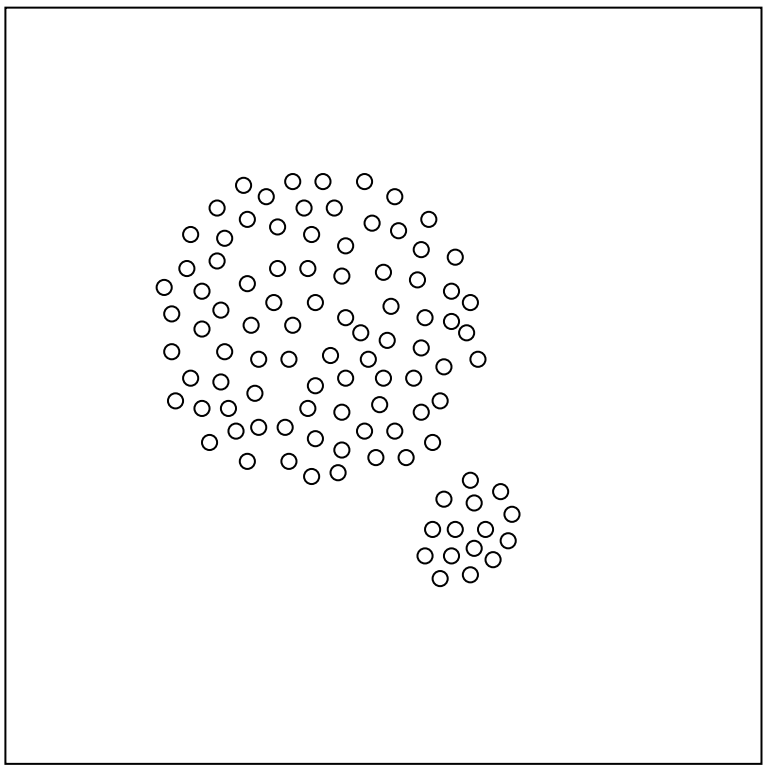
k-Means versus Single Link



Exemplar-based algorithms fail to detect nested clusters.

Iterative Cluster Analysis

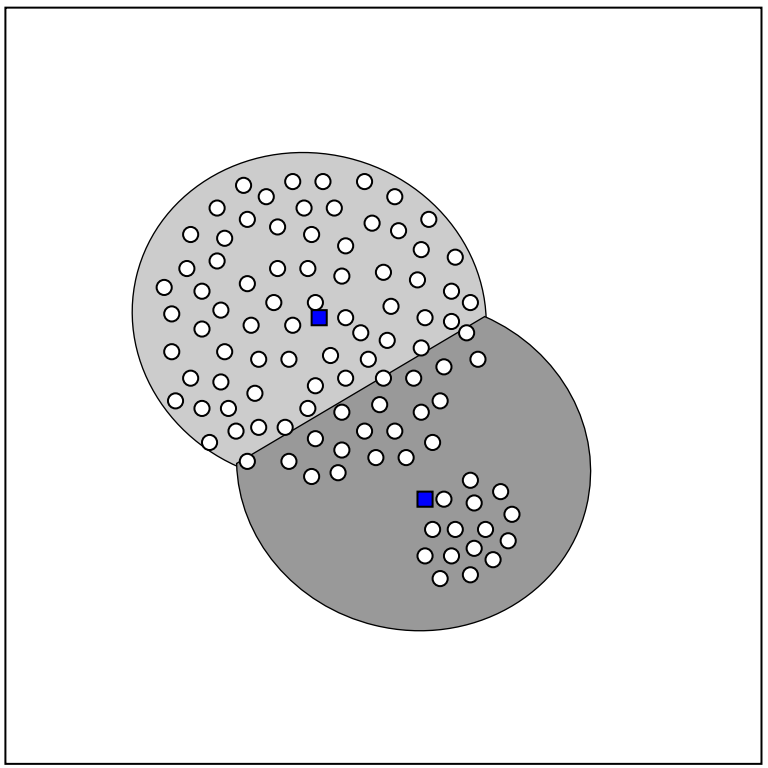
k-Means versus Single Link



Exemplar-based algorithms fail to detect clusters with large difference in size.

Iterative Cluster Analysis

k-Means versus Single Link



Exemplar-based algorithms fail to detect clusters with large difference in size.

Iterative Cluster Analysis

Exclusive versus Non-Exclusive Algorithms

Let $\mathcal{C} = \{C_1, \ldots, C_k\}$ be a partitioning of a set V with $\bigcup_{i=1\ldots k} C_i = V$.

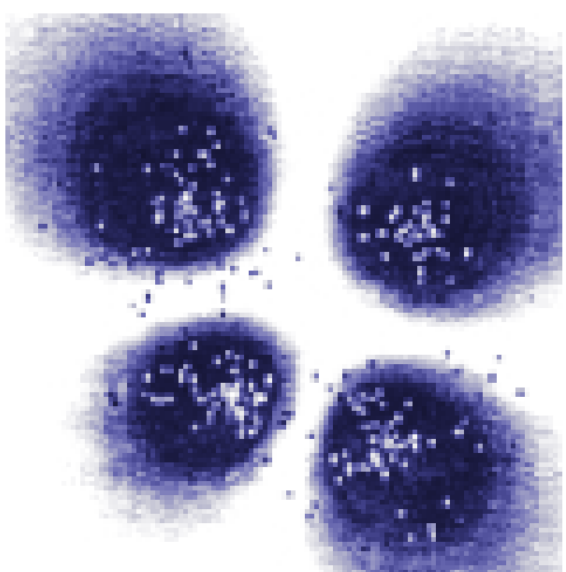
- exclusive algorithms: $\forall i, j \in \{1, \ldots, k\} : i \neq j$ implies $C_i \cap C_j = \emptyset$
- non-exclusive algorithms allows for multiple cluster membership

Iterative Cluster Analysis

Exclusive versus Non-Exclusive Algorithms

Let $\mathcal{C} = \{C_1, \ldots, C_k\}$ be a partitioning of a set V with $\bigcup_{i=1\ldots k} C_i = V$.

- exclusive algorithms: $\forall i, j \in \{1, \ldots, k\} : i \neq j$ implies $C_i \cap C_j = \emptyset$
- non-exclusive algorithms allows for multiple cluster membership
- Fuzzy cluster analysis quantifies cluster membership of the $v \in V$ by means of a membership function $\mu_i(v),\ i \in \{1, \ldots, k\}$. [minimization criterion]

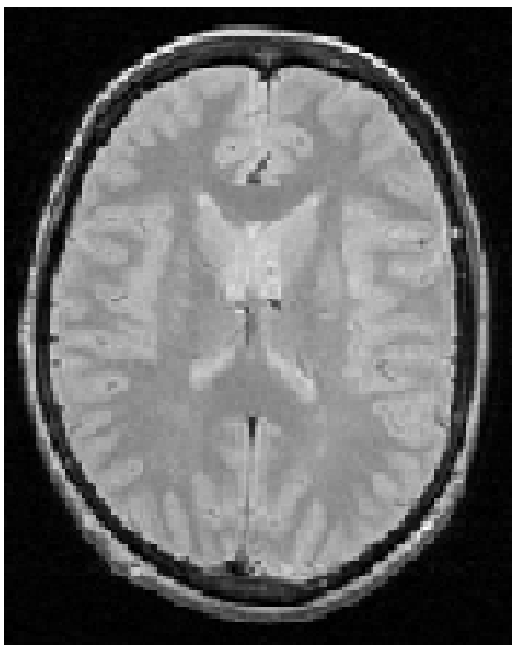


[Höppner/Klawonn/Kruse 1997]

Iterative Cluster Analysis

Exclusive versus Non-Exclusive Algorithms

Application of Fuzzy cluster analysis to represent and envision cerebral tissue:



[Pham/Prince/Dagher/Xn 1996]

Remarks:

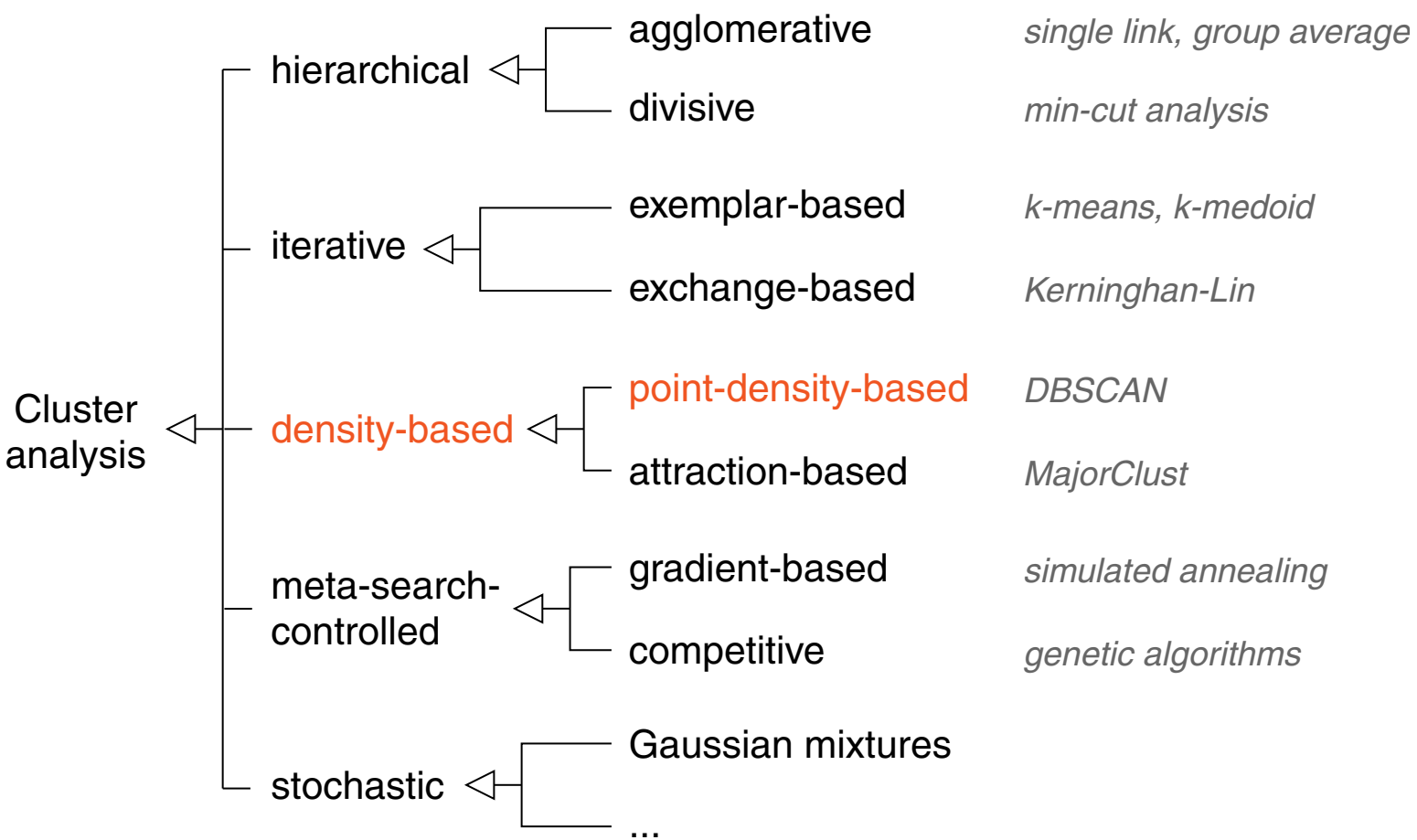
- The domain of the linguistic variable of the Fuzzy model is comprised of k elements, which correspond to the clusters $C_1, \ldots, C_k$.
- Usually a normalization constraint for the membership function is stated: $\sum_{i=1 \ldots k} \mu_i(v) = 1$
- A drawback of Fuzzy k-means variants that neglect normalization is that points with small membership function values for a cluster are treated as outliers, instead of moving the cluster towards these points. Hence it is useful to apply the iteration procedure with a normalization constraint—at least within an initialization phase.
- A categorization by a Fuzzy cluster analysis is beneficial if no clear class structure is given or if various feature vectors belong to several classes at the same time.
- A disadvantage of Fuzzy cluster analysis is the fact that the concept of cluster representatives does not exist.

Chapter ML:XI (continued)

XI. Cluster Analysis

Density-Based Cluster Analysis

Merging Principles



Density-Based Cluster Analysis

Density-based algorithms strive to partition the graph $G = \langle V, E, w \rangle$, better: the set of points V, into regions of equal density.

Approaches to density estimation:

- parameter-based: the type of the underlying data distribution is known
- parameterless: construction of histograms, superposition of kernel density estimators

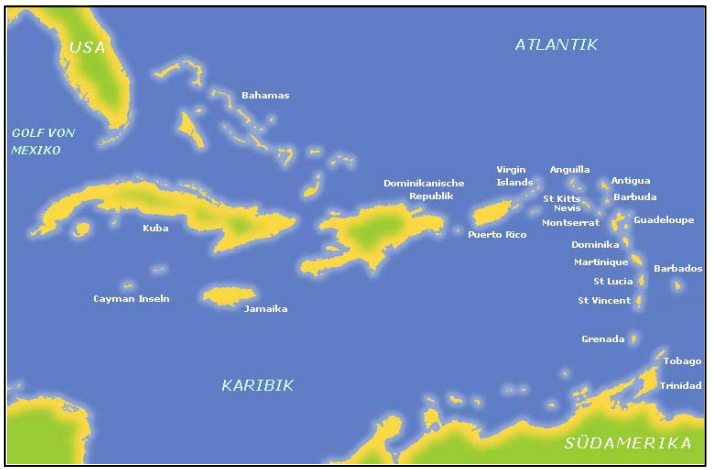
Density-Based Cluster Analysis

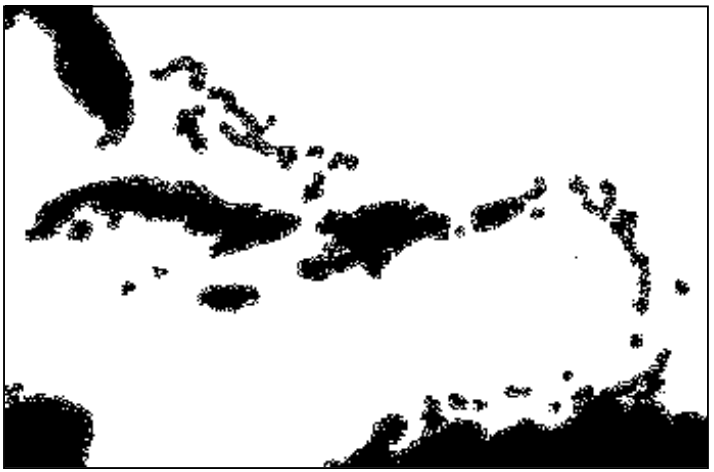
Density-based algorithms strive to partition the graph $G = \langle V, E, w \rangle$, better: the set of points V, into regions of equal density.

Approaches to density estimation:

- parameter-based: the type of the underlying data distribution is known
- parameterless: construction of histograms, superposition of kernel density estimators

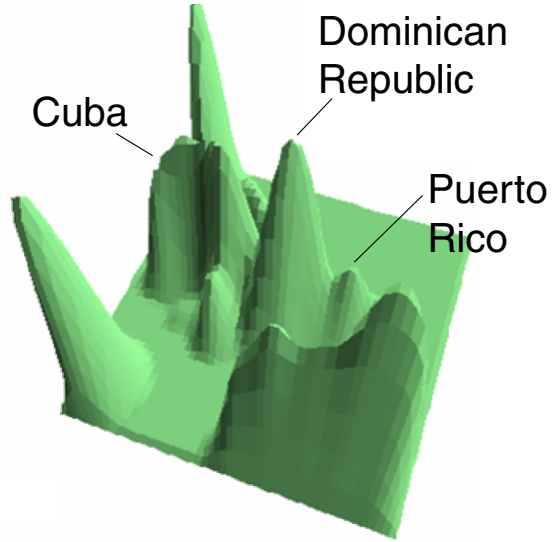
Example (Caribbean Islands) :





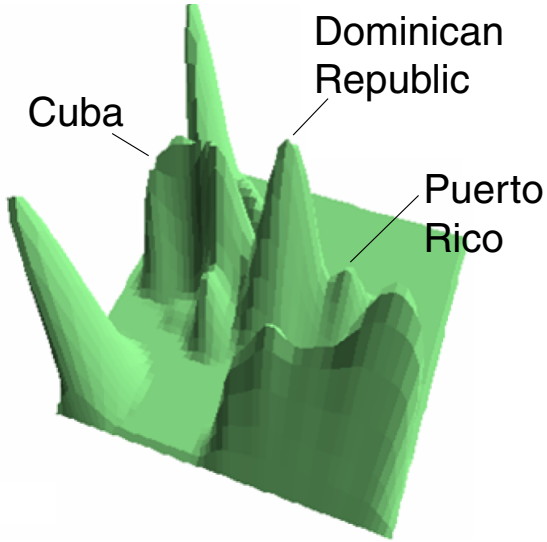
Density-Based Cluster Analysis

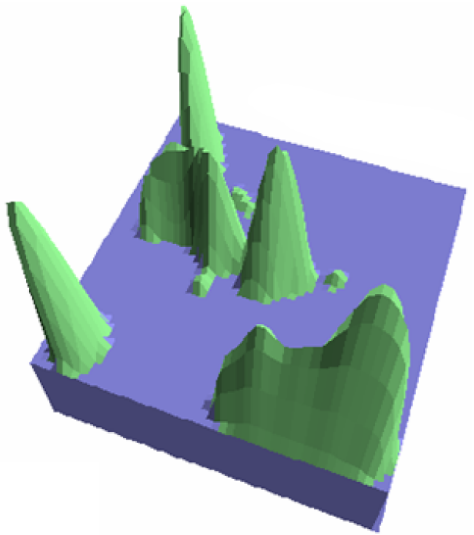
Density Estimation with Gaussian Kernel for the Example



Density-Based Cluster Analysis

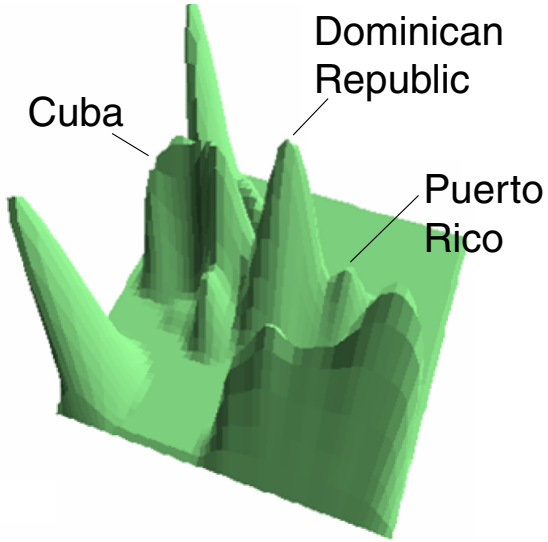
Density Estimation with Gaussian Kernel for the Example

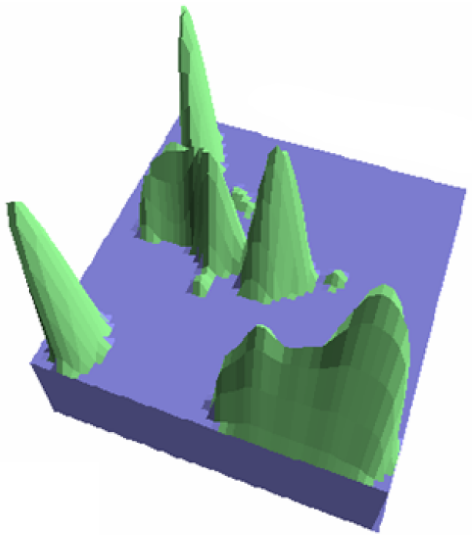


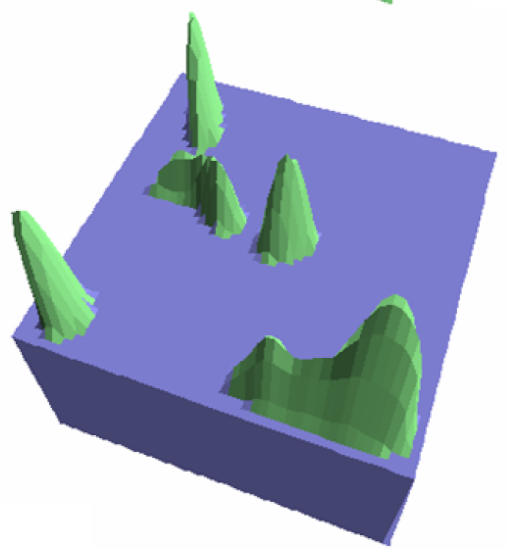


Density-Based Cluster Analysis

Density Estimation with Gaussian Kernel for the Example

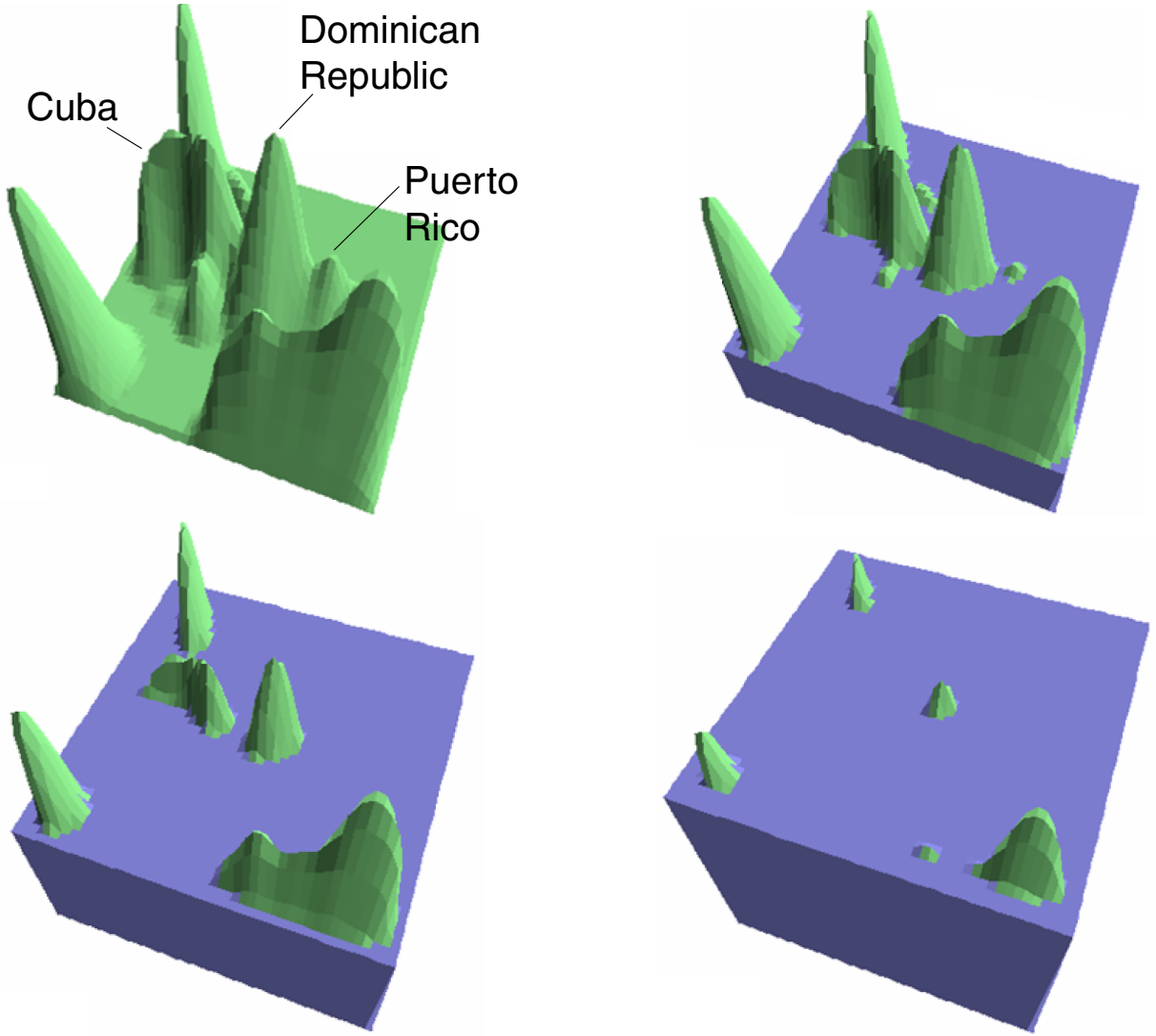






Density-Based Cluster Analysis

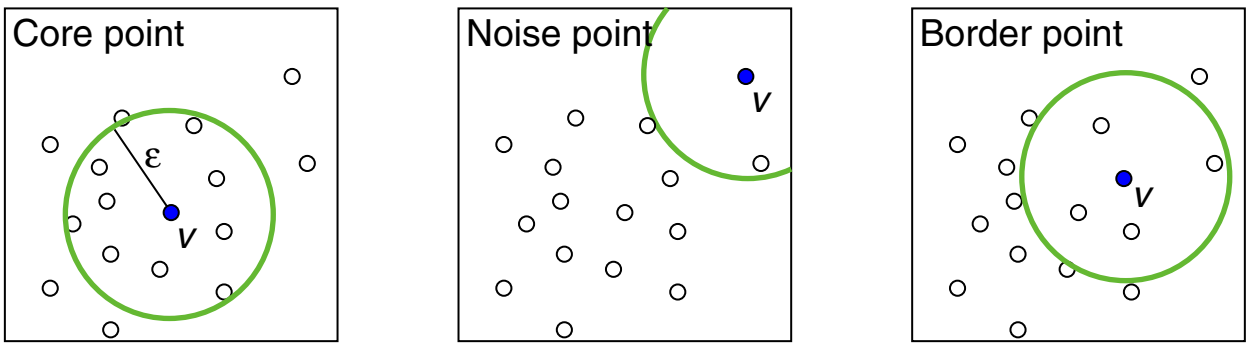
Density Estimation with Gaussian Kernel for the Example



Density-Based Cluster Analysis

DBSCAN: Density Estimation Principle [Ester et al. 1996]

Let $N_\varepsilon(v)$ denote the ε-neighborhood of some point $v \in V$. Differentiation between three kinds of points:



1. v is a core point $\Leftrightarrow$ $|N_\varepsilon(v)| \geq$ *MinPts*
2. v is a noise point $\Leftrightarrow$
   v is not density-reachable from any core point
3. v is a border point otherwise

Density-Based Cluster Analysis

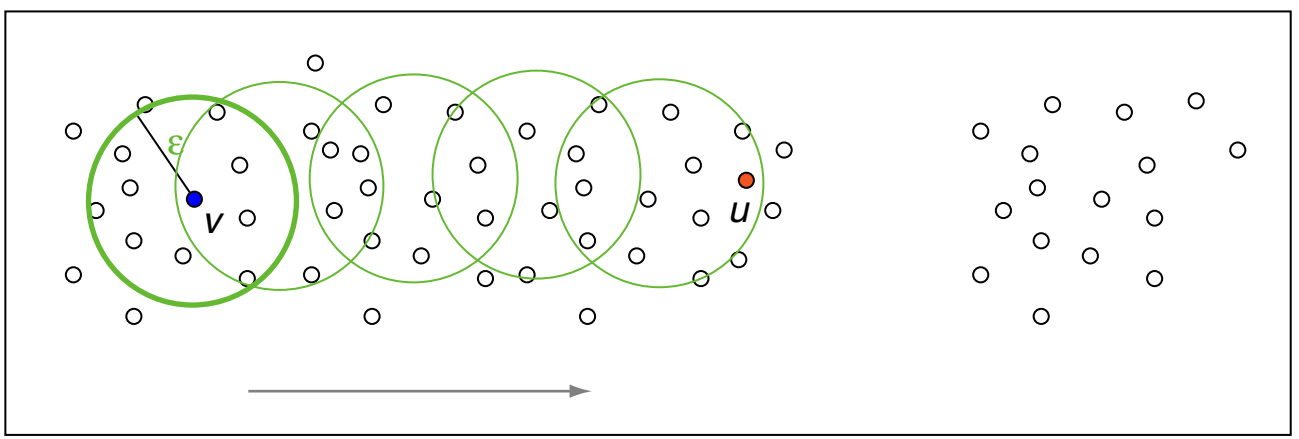
DBSCAN: Density Estimation Principle

A point u is density-reachable from a point v, if either of the following conditions hold:

(a) $u \in N_\varepsilon(v)$, where v is a core point.

(b) There exists a set of points $\{v_1, \ldots, v_l\}$, where

$v_{i+1} \in N_\varepsilon(v_i)$ and v_i is core point, $i = 1, \ldots, l-1$, with $v_1 = v$, $v_l = u$.



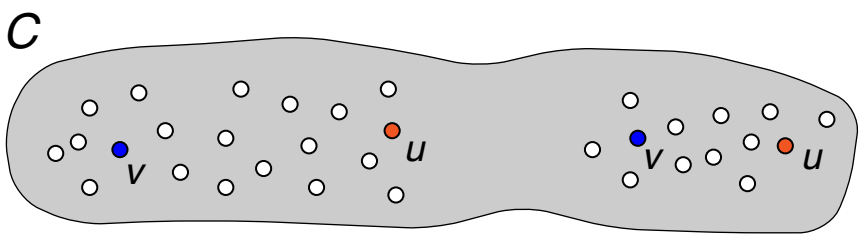
Condition (b) can be considered as the transitive application of Condition (a).

Density-Based Cluster Analysis

DBSCAN: Cluster Interpretation

A cluster $C \subseteq V$ fulfills the following two conditions:

1. $\forall u, v$: If $v \in C$ and u is density-reachable from v, then $u \in C$.



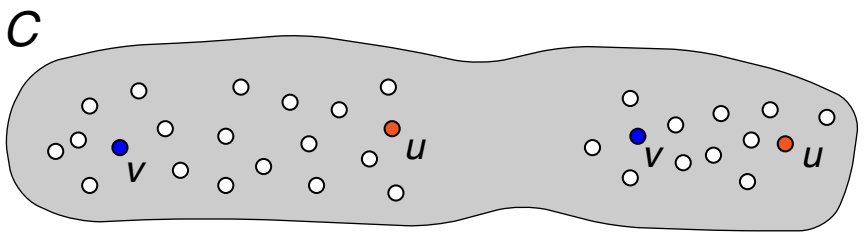
Maximality condition

Density-Based Cluster Analysis

DBSCAN: Cluster Interpretation

A cluster $C \subseteq V$ fulfills the following two conditions:

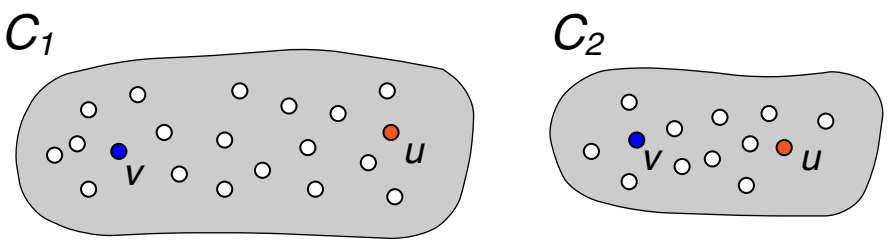
1. $\forall u, v$: If $v \in C$ and u is density-reachable from v, then $u \in C$.



Maximality condition

2. $\forall u, v$: u is density-connected with v, which is defined as follows:

   There exists a point t wherefrom u and v are density-reachable.



Connectivity condition

Density-Based Cluster Analysis

DBSCAN: Algorithm

Input: $G = \langle V, E, w \rangle$. Weighted graph.
d. Distance measure for two nodes in V.
ε. Neighborhood radius.
MinPts. Lower bound for point number in ε-neighborhood.

Output: $\gamma : V \to \mathbf{Z}$. Cluster assignment function.

```
1.
2.
3.

4.    N_ε(v) = neighborhood(G, d, v, ε)
5.    IF |N_ε(v)| ≥ MinPts THEN // v is core point
6.      i = i + 1
7.      C_i = density_reachable_hull(G, d, N_ε(v)) // form a new cluster
8.      FOREACH v ∈ C_i DO γ(v) = i
9.    ELSE γ(v) = −1 // v is _preliminarily_ classified as noise

10.
11.
```

Density-Based Cluster Analysis

DBSCAN: Algorithm

Input: $G = \langle V, E, w \rangle$. Weighted graph.
d. Distance measure for two nodes in V.
ε. Neighborhood radius.
MinPts. Lower bound for point number in ε-neighborhood.

Output: $\gamma : V \to \mathbf{Z}$. Cluster assignment function.

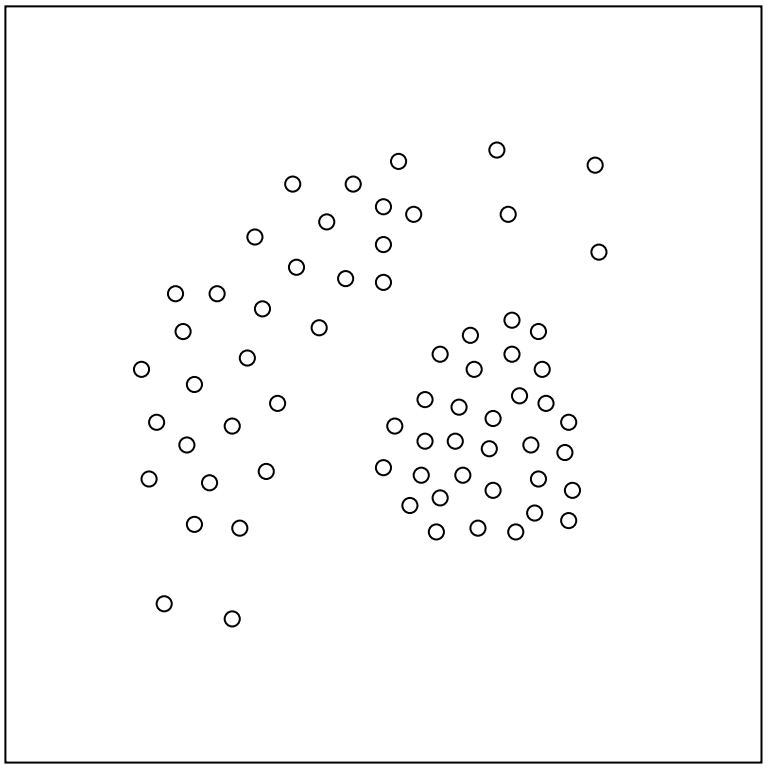
```
1.  i = 0
2.  WHILE ∃v : (v ∈ V AND γ(v) = ⊥) DO // ⊥ = unclassified
3.    v = choose_unclassified_point(V)

4.    N_ε(v) = neighborhood(G, d, v, ε)
5.    IF |N_ε(v)| ≥ MinPts THEN // v is core point
6.      i = i + 1
7.      C_i = density_reachable_hull(G, d, N_ε(v)) // form a new cluster
8.      FOREACH v ∈ C_i DO γ(v) = i
9.    ELSE γ(v) = −1 // v is _preliminarily_ classified as noise

10. ENDDO
11. RETURN(γ)
```

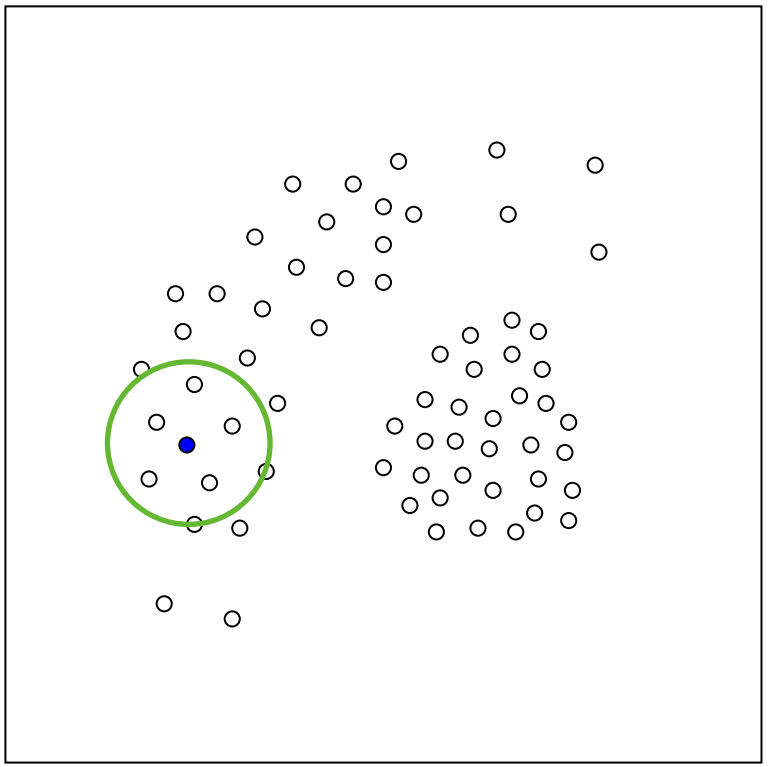
Density-Based Cluster Analysis

DBSCAN



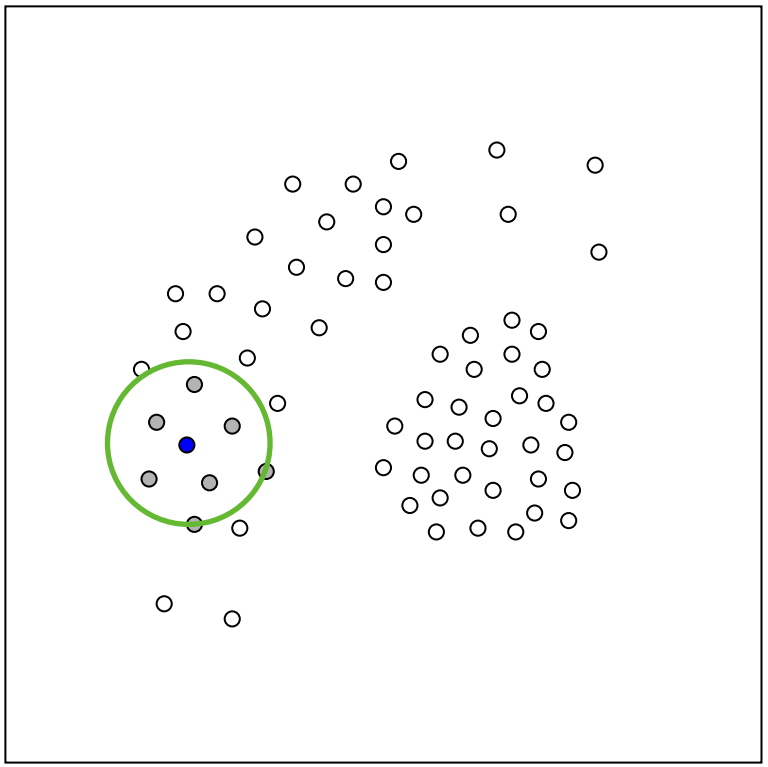
Density-Based Cluster Analysis

DBSCAN



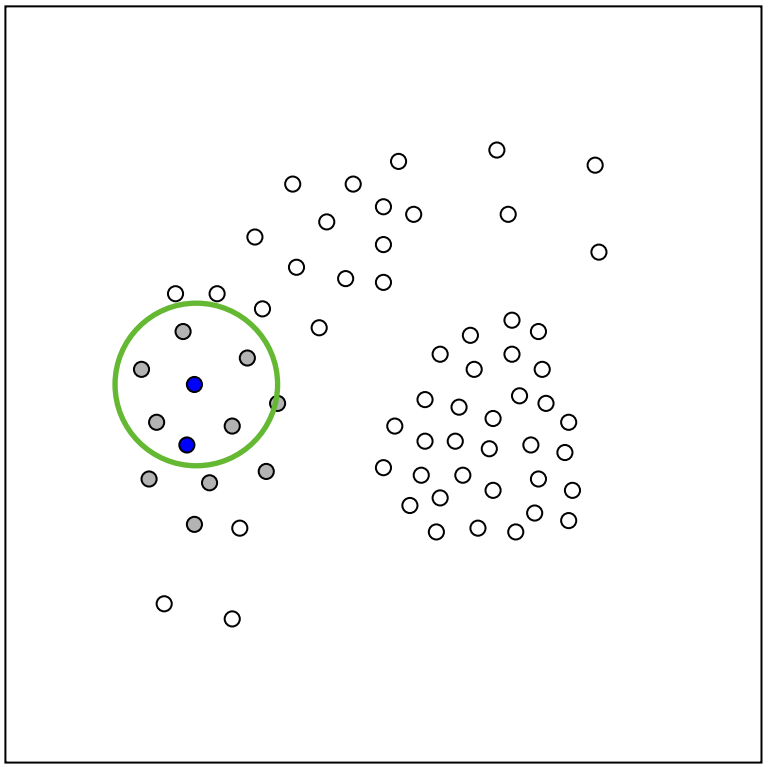
Density-Based Cluster Analysis

DBSCAN



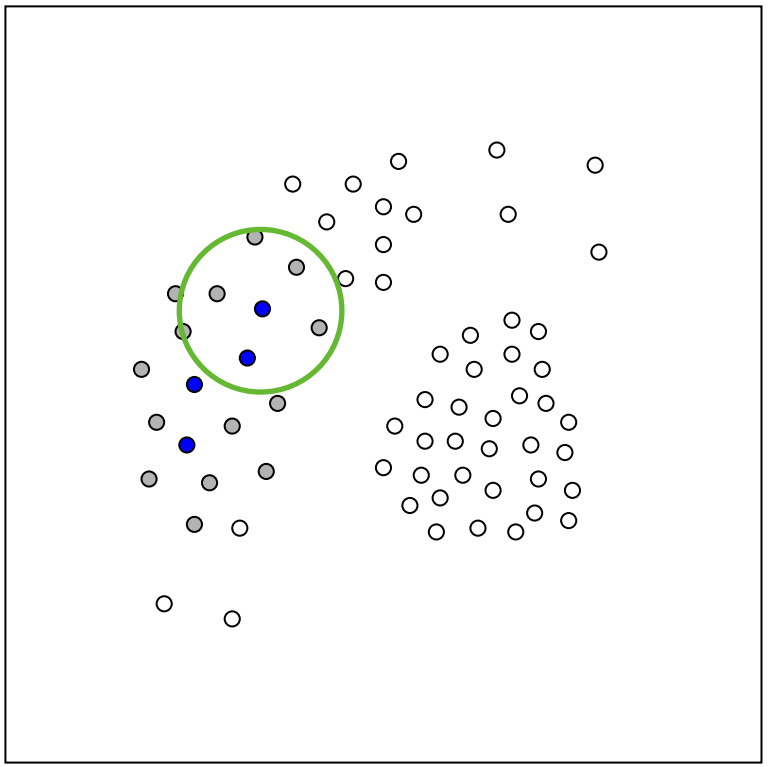
Density-Based Cluster Analysis

DBSCAN



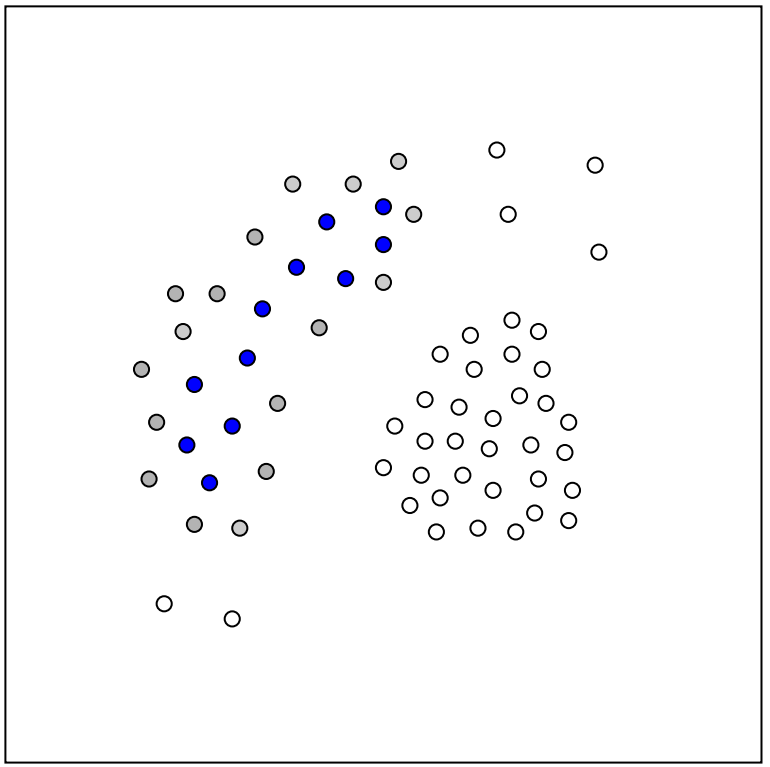
Density-Based Cluster Analysis

DBSCAN



Density-Based Cluster Analysis

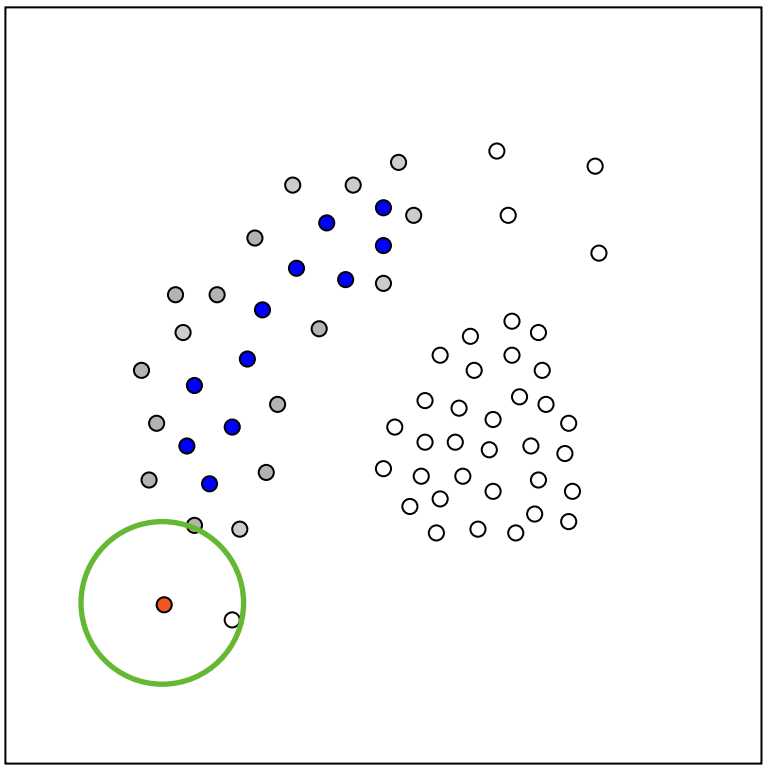
DBSCAN



- Core point
- Border point

Density-Based Cluster Analysis

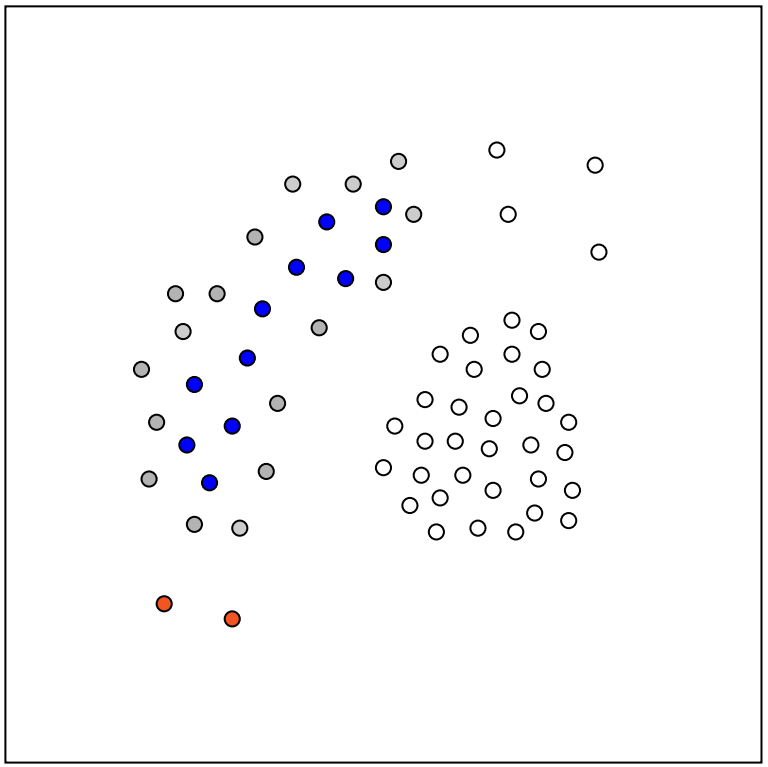
DBSCAN



- Core point
- Border point

Density-Based Cluster Analysis

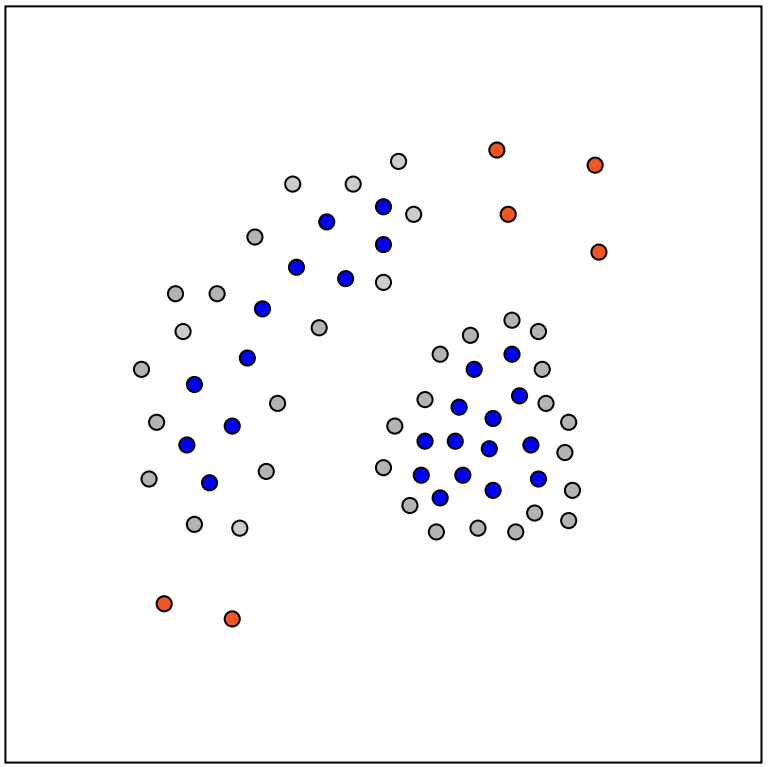
DBSCAN



- Core point
- Border point
- Noise point

Density-Based Cluster Analysis

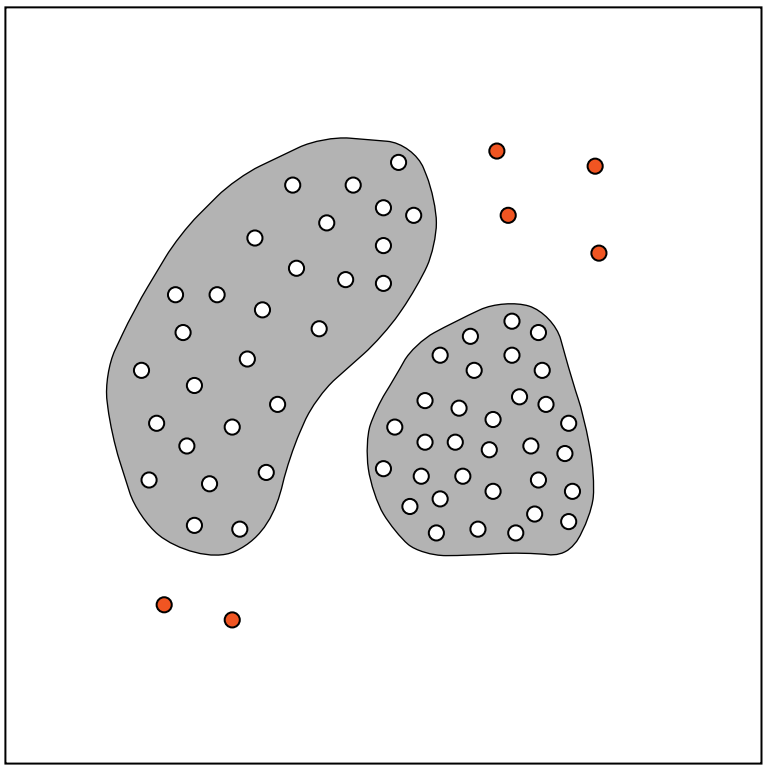
DBSCAN



- Core point
- Border point
- Noise point

Density-Based Cluster Analysis

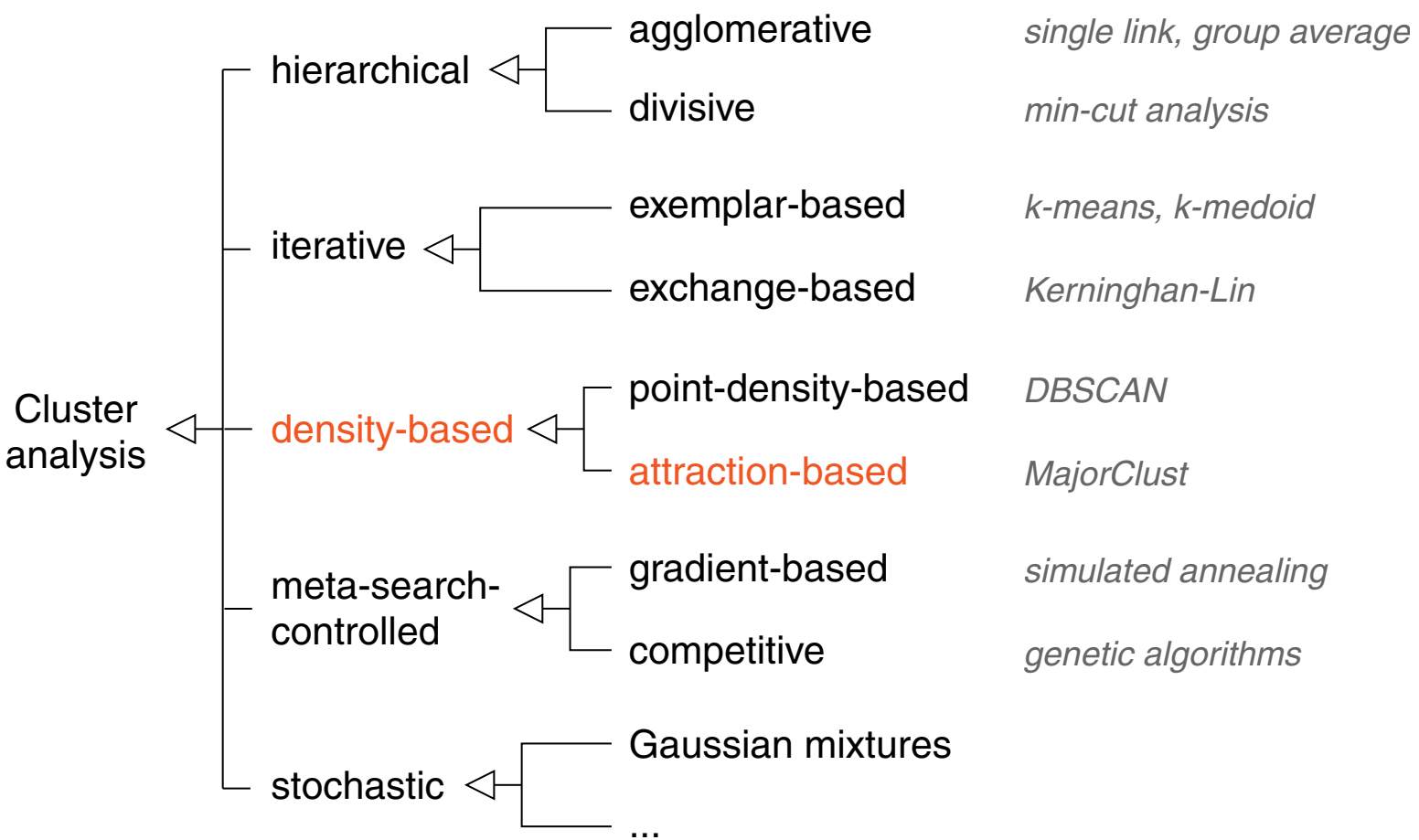
DBSCAN



Noise point

Density-Based Cluster Analysis

Merging Principles



Density-Based Cluster Analysis

MajorClust: Density Estimation Principle [Stein/Niggemann 1999]

The weighted edges in a graph $G = \langle V, E, w \rangle$ are interpreted as attracting forces, whereas members of the same cluster combine their forces. Illustration:

Unique membership situation, leading to a merge of two clusters:

Density-Based Cluster Analysis

MajorClust: Density Estimation Principle [Stein/Niggemann 1999]

The weighted edges in a graph $G = \langle V, E, w \rangle$ are interpreted as attracting forces, whereas members of the same cluster combine their forces. Illustration:

Unique membership situation, leading to a merge of two clusters:

Unique membership situation, leading to a change of cluster membership:

Density-Based Cluster Analysis

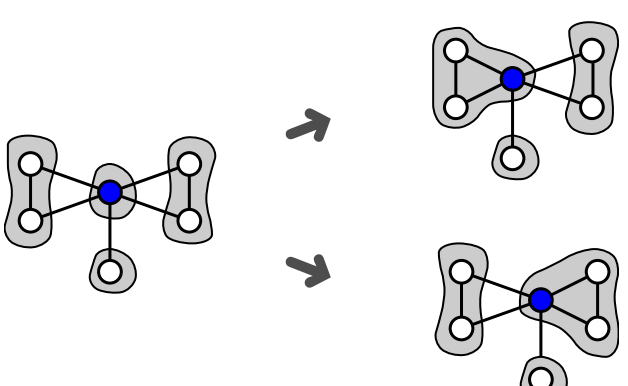
MajorClust: Density Estimation Principle [Stein/Niggemann 1999]

The weighted edges in a graph $G = \langle V, E, w \rangle$ are interpreted as attracting forces, whereas members of the same cluster combine their forces. Illustration:

Unique membership situation, leading to a merge of two clusters:

Unique membership situation, leading to a change of cluster membership:

Ambiguous membership situation:



Density-Based Cluster Analysis

MajorClust: Algorithm

Input: $G = \langle V, E, w \rangle$. Weighted graph.
d. Distance measure for two nodes in V.

Output: $\gamma : V \to \mathbf{N}$. Cluster assignment function.

1.
2.
3.
4.
5. **FOREACH** $v \in V$ **DO**
6. $\gamma^* = \underset{i:\, i \in \{1,\ldots,|V|\}}{\text{argmax}} \sum_{\{u,v\}:\, \{u,v\} \in E \wedge \gamma(u) = i} w(u,v)$
7. **IF** $\gamma(v) \neq \gamma^*$ **THEN** $\gamma(v) = \gamma^*$, $t = \mathit{False}$ **ENDIF** `// relabeling`
8. **ENDDO**
9.
10.

Density-Based Cluster Analysis

MajorClust: Algorithm

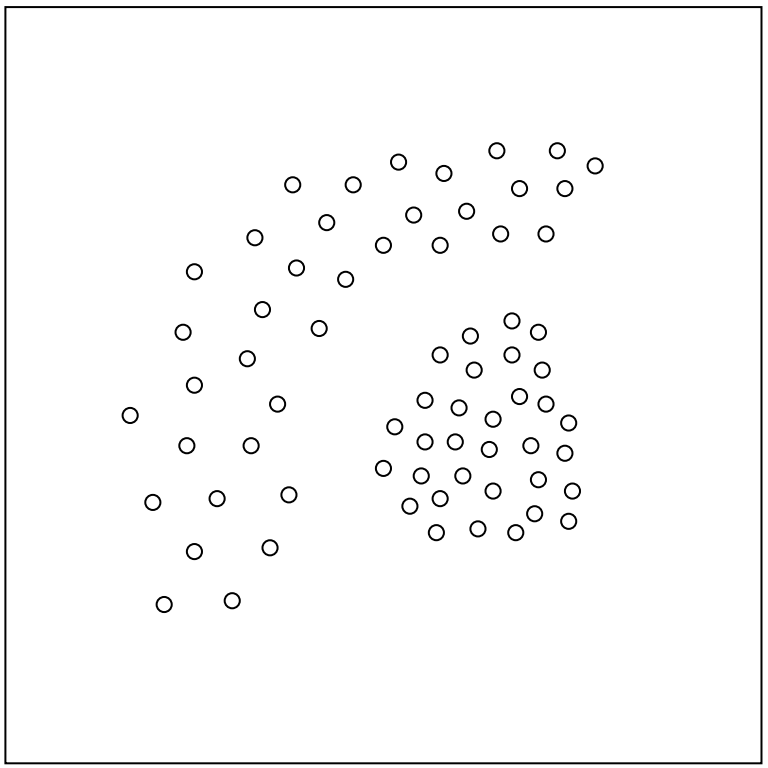
Input: $G = \langle V, E, w \rangle$. Weighted graph.
d. Distance measure for two nodes in V.

Output: $\gamma : V \to \mathbf{N}$. Cluster assignment function.

1. $i = 0$, $t =$ *False*
2. **FOREACH** $v \in V$ **DO** $i = i + 1$, $\gamma(v) = i$ **ENDDO**
3. **UNLESS** t **DO**
4. $t =$ *True*
5. **FOREACH** $v \in V$ **DO**
6. $\gamma^* = \underset{i:\, i \in \{1,\ldots,|V|\}}{\text{argmax}} \sum_{\{u,v\}:\, \{u,v\} \in E \wedge \gamma(u) = i} w(u, v)$
7. **IF** $\gamma(v) \neq \gamma^*$ **THEN** $\gamma(v) = \gamma^*$, $t =$ *False* **ENDIF** `// relabeling`
8. **ENDDO**
9. **ENDDO**
10. **RETURN**(γ)

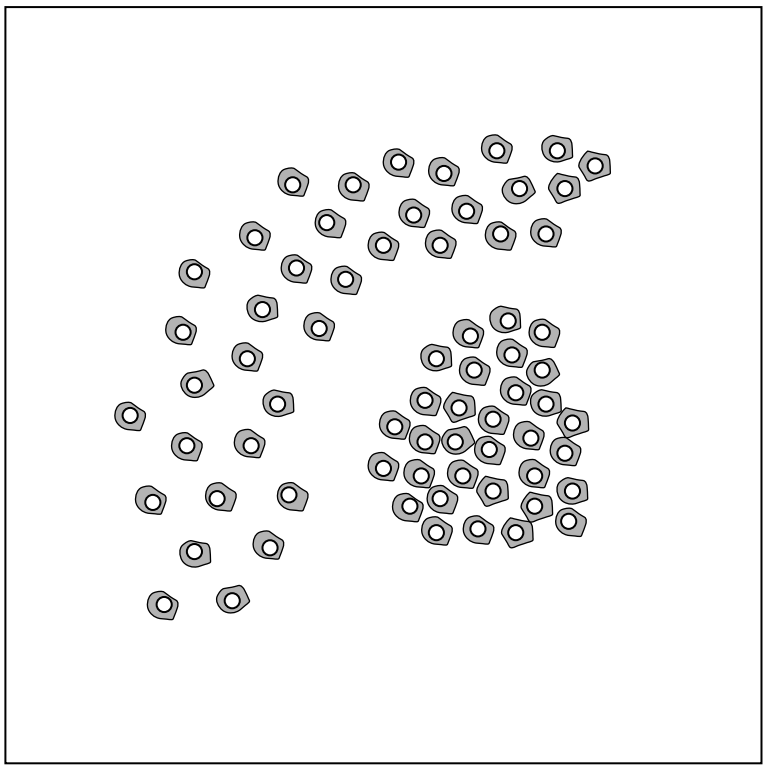
Density-Based Cluster Analysis

MajorClust



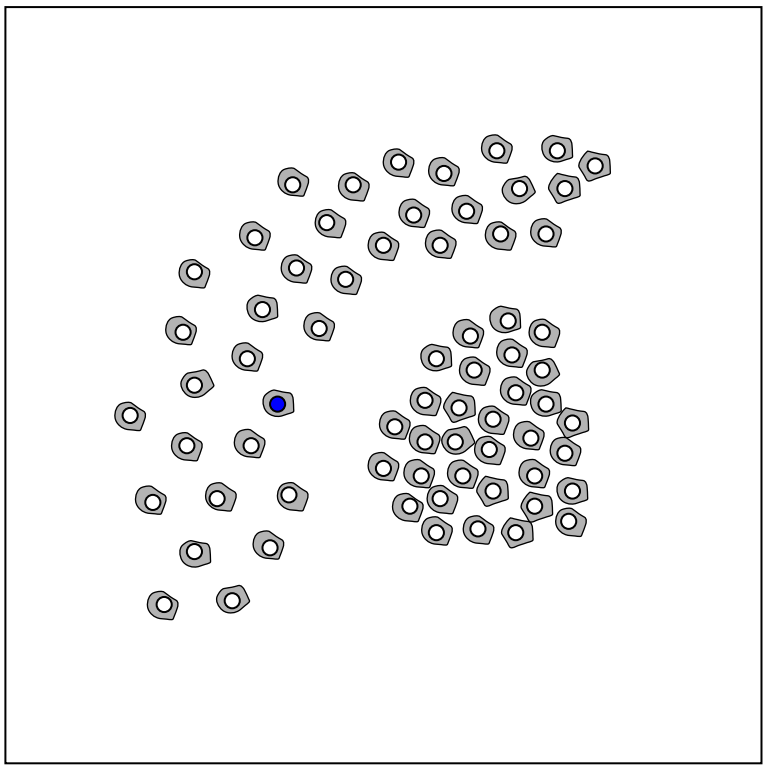
Density-Based Cluster Analysis

MajorClust



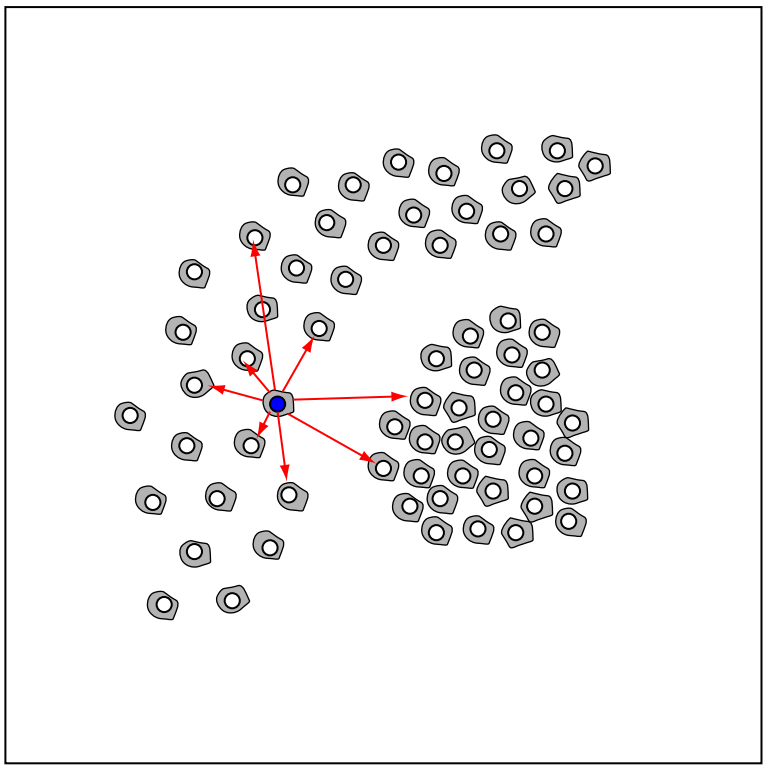
Density-Based Cluster Analysis

MajorClust



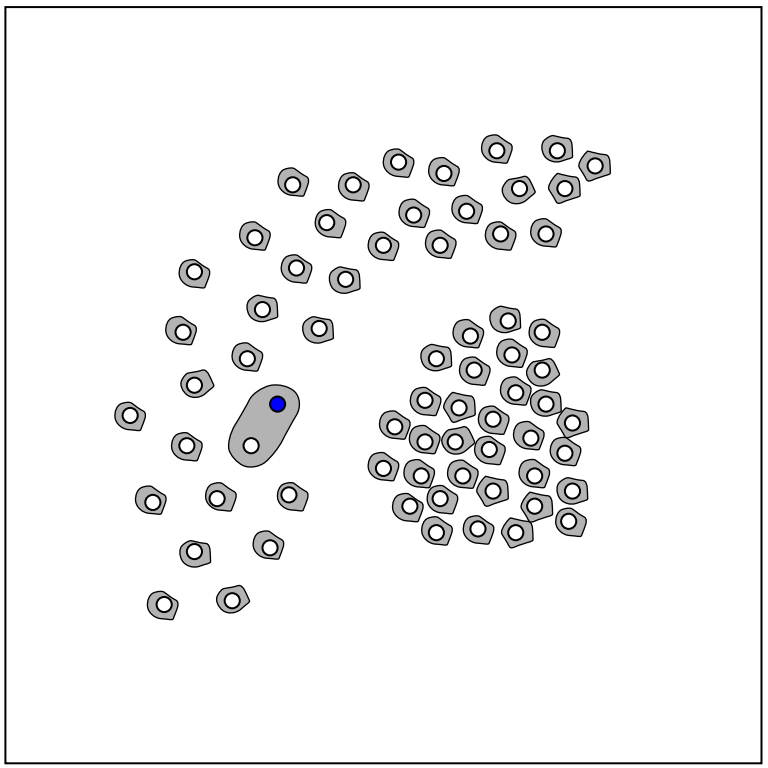
Density-Based Cluster Analysis

MajorClust



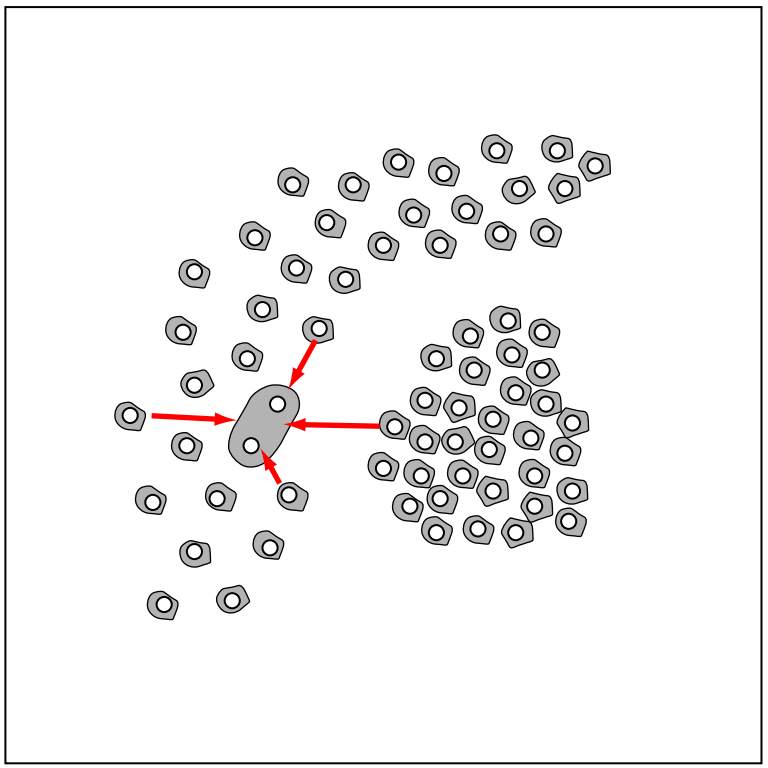
Density-Based Cluster Analysis

MajorClust



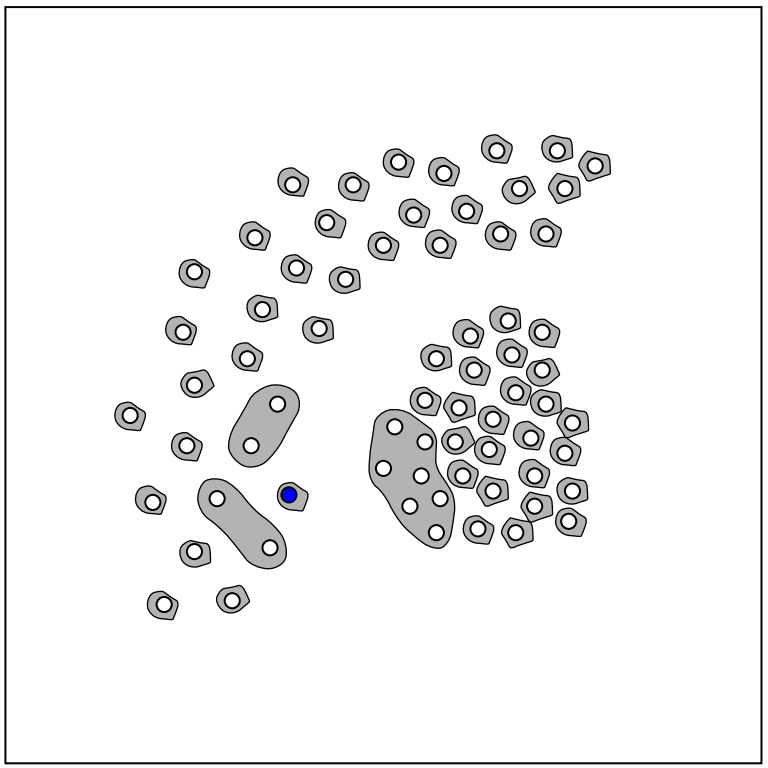
Density-Based Cluster Analysis

MajorClust



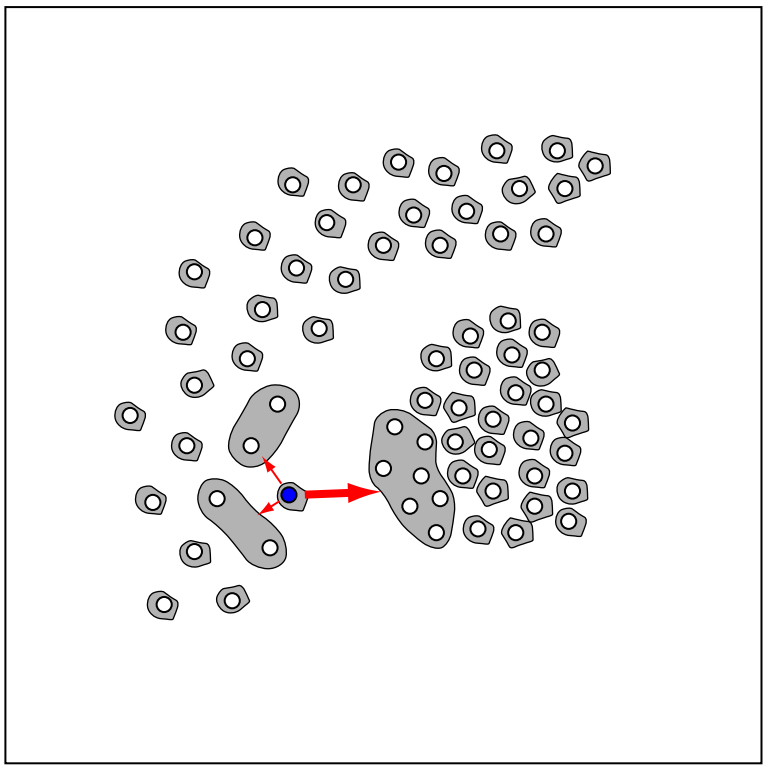
Density-Based Cluster Analysis

MajorClust



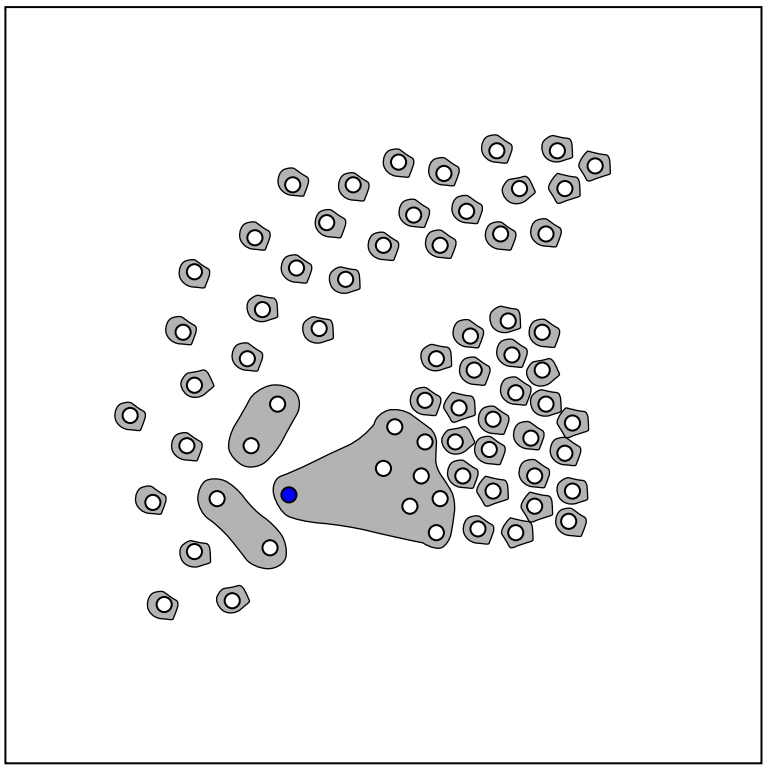
Density-Based Cluster Analysis

MajorClust



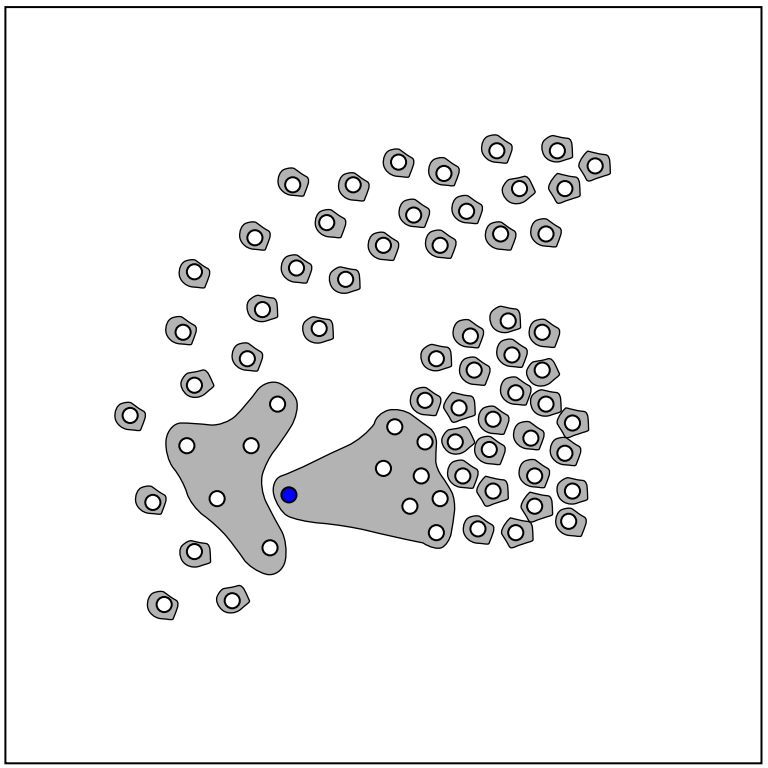
Density-Based Cluster Analysis

MajorClust



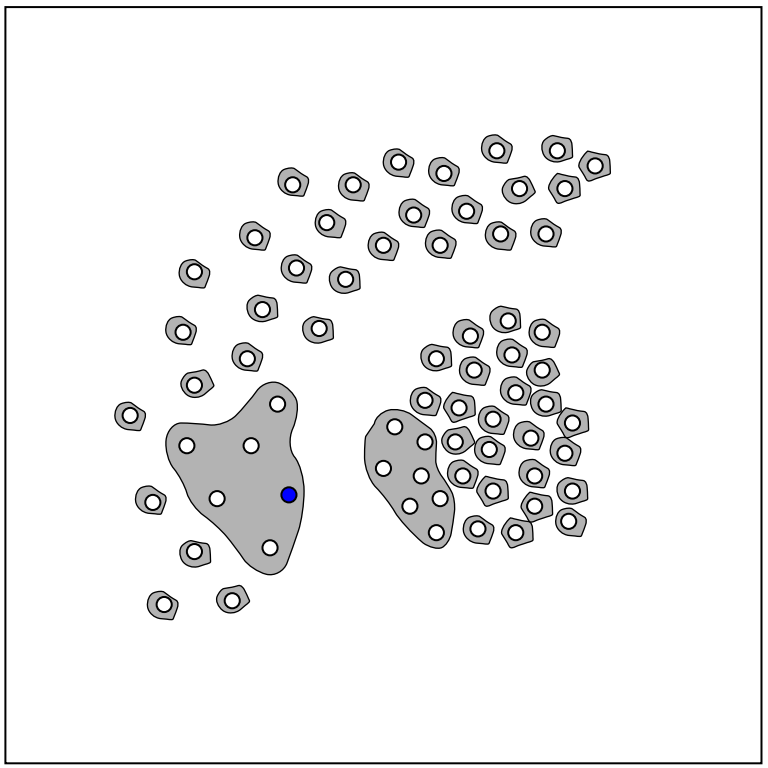
Density-Based Cluster Analysis

MajorClust



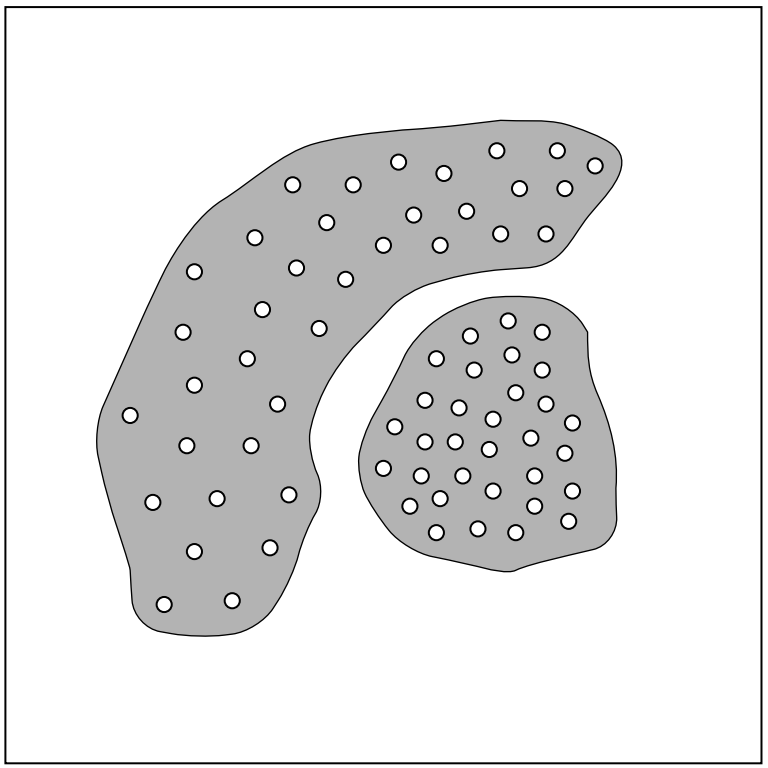
Density-Based Cluster Analysis

MajorClust



Density-Based Cluster Analysis

MajorClust



Density-Based Cluster Analysis

MajorClust: Density Estimation Principle (continued)

Each clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$ induces k subgraphs within $G = \langle V, E, w \rangle$.
MajorClust is a heuristic to maximize the *weighted partial edge connectivity*, $\Lambda(\mathcal{C})$.

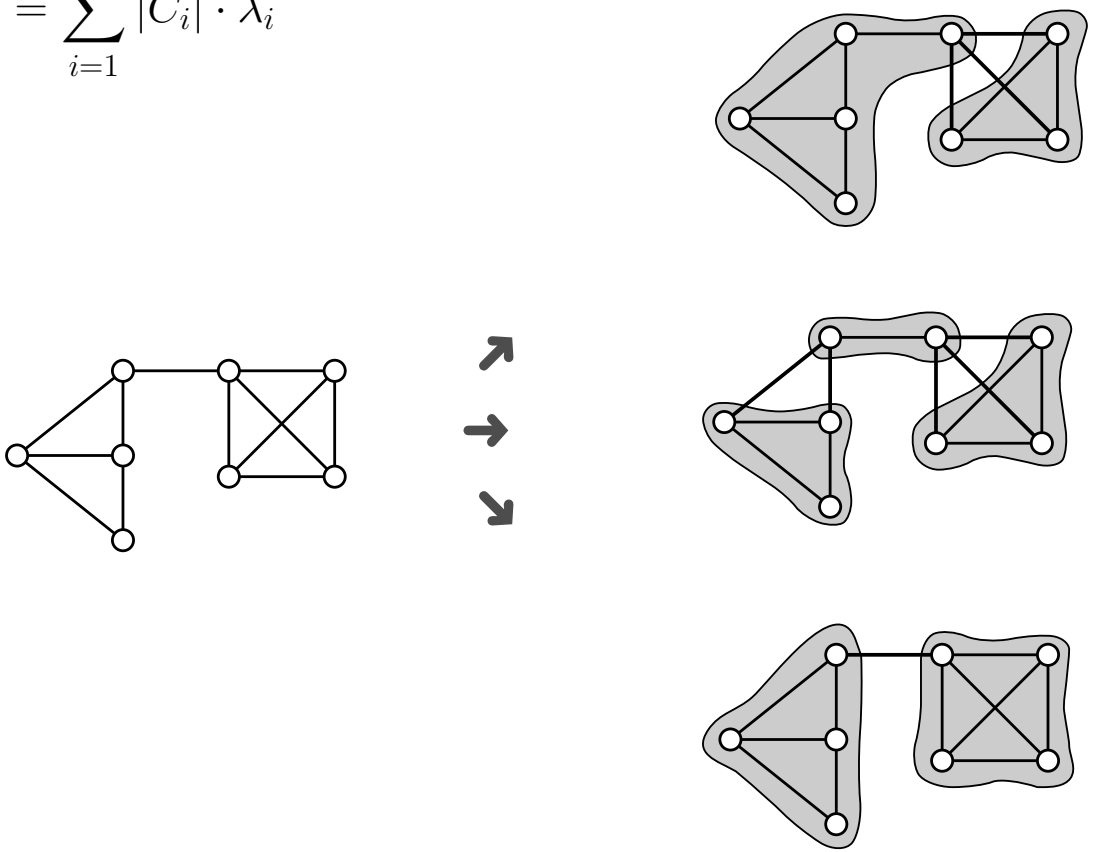
$$\Lambda(\mathcal{C}) = \sum_{i=1}^{k} |C_i| \cdot \lambda_i$$

Density-Based Cluster Analysis

MajorClust: Density Estimation Principle (continued)

Each clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$ induces k subgraphs within $G = \langle V, E, w \rangle$.
MajorClust is a heuristic to maximize the *weighted partial edge connectivity*, $\Lambda(\mathcal{C})$.

$$\Lambda(\mathcal{C}) = \sum_{i=1}^{k} |C_i| \cdot \lambda_i$$

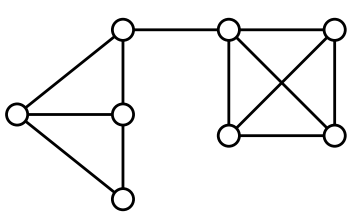


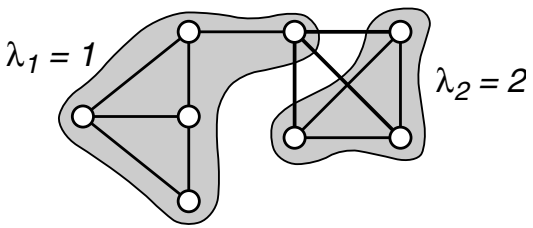
Density-Based Cluster Analysis

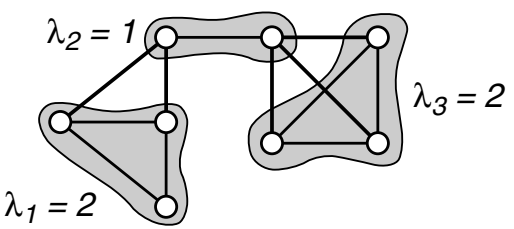
MajorClust: Density Estimation Principle (continued)

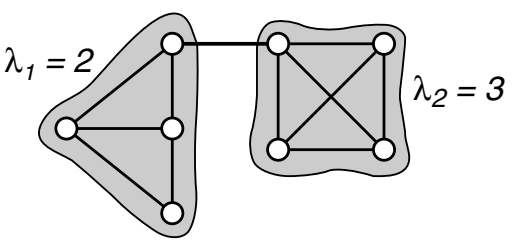
Each clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$ induces k subgraphs within $G = \langle V, E, w \rangle$.
MajorClust is a heuristic to maximize the *weighted partial edge connectivity*, $\Lambda(\mathcal{C})$.

$$\Lambda(\mathcal{C}) = \sum_{i=1}^{k} |C_i| \cdot \lambda_i$$







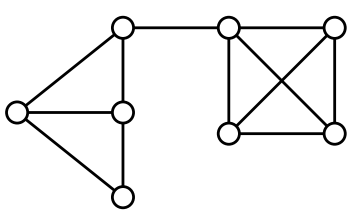


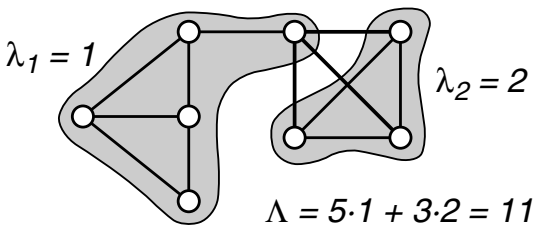
Density-Based Cluster Analysis

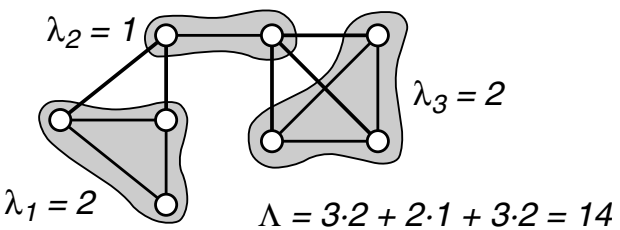
MajorClust: Density Estimation Principle (continued)

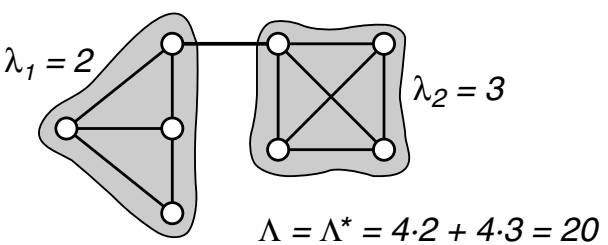
Each clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$ induces k subgraphs within $G = \langle V, E, w \rangle$.
MajorClust is a heuristic to maximize the *weighted partial edge connectivity*, $\Lambda(\mathcal{C})$.

$$\Lambda(\mathcal{C}) = \sum_{i=1}^{k} |C_i| \cdot \lambda_i$$



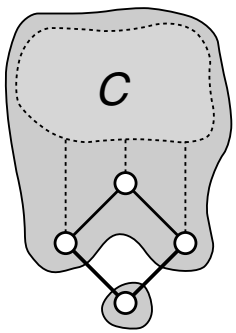




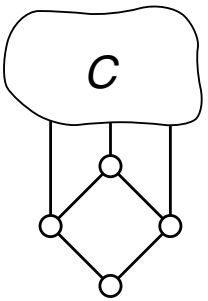


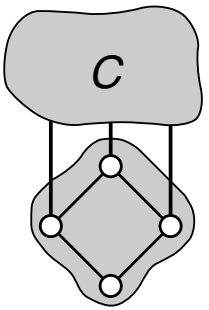
Density-Based Cluster Analysis

MajorClust: Density Estimation Principle (continued)



Mincut clustering

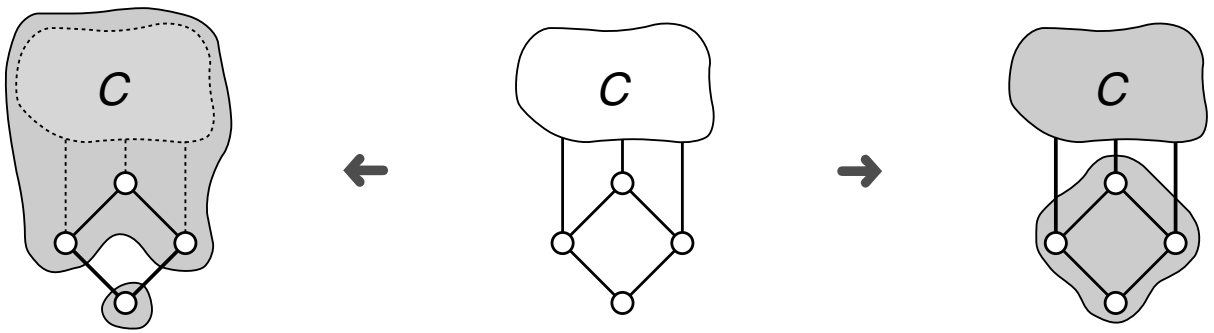




Λ maximization

Density-Based Cluster Analysis

MajorClust: Density Estimation Principle (continued)



Mincut clustering

Λ maximization

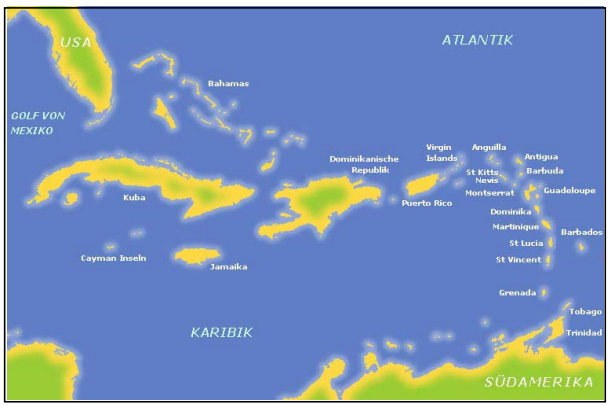
Theorem 1 (Strong Splitting Condition [Stein/Niggemann 1999]**)**
Let $\mathcal{C} = \{C_1, \ldots, C_k\}$ be a partitioning of a graph $G = \langle V, E, w \rangle$. Moreover, let $\lambda(G)$ denote the edge connectivity of G, and let $\lambda_1, \ldots, \lambda_k$ denote the edge connectivity values of the k subgraphs that are induced by $C_1, \ldots, C_k$.

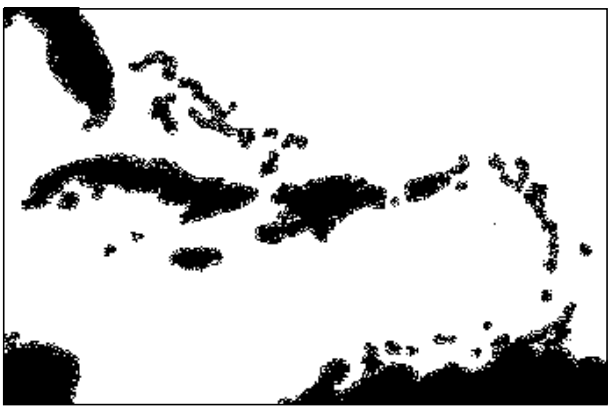
If the inequality $\lambda(G) < \min\{\lambda_1, \ldots, \lambda_k\}$ holds, then the partitioning defined by Λ-maximization corresponds to the minimum cut splitting of G. The inequality is called “Strong Splitting Condition”.

Density-Based Cluster Analysis

DBSCAN versus MajorClust: Low-Dimensional Data

Caribbean Islands, about 20.000 points:

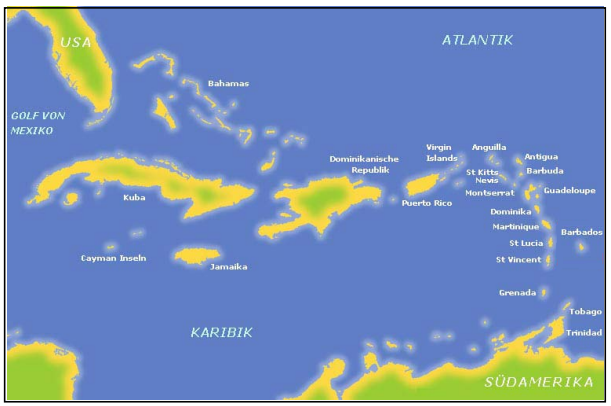


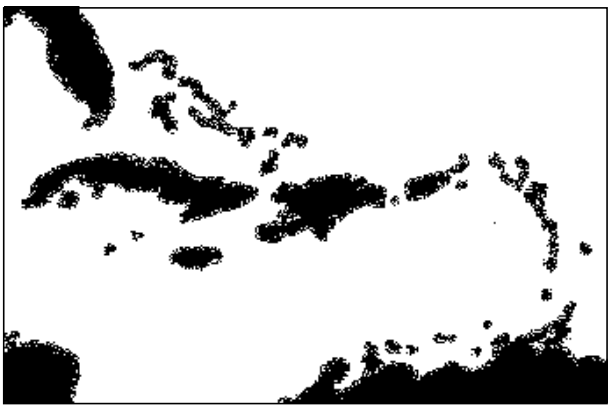


Density-Based Cluster Analysis

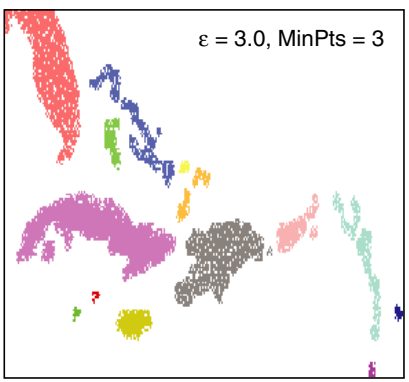
DBSCAN versus MajorClust: Low-Dimensional Data (continued)

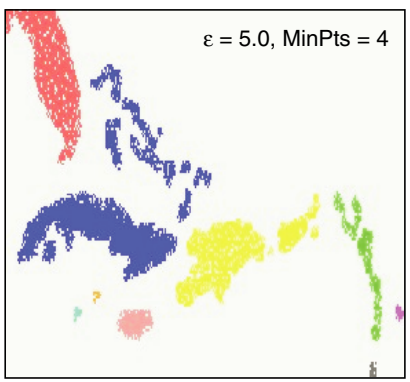
Caribbean Islands, about 20.000 points:

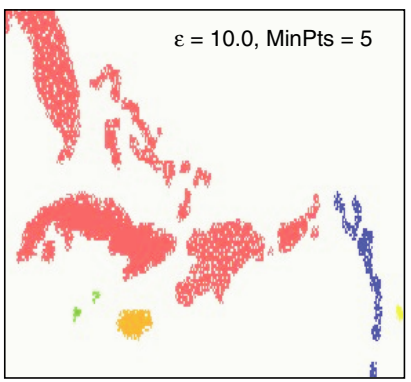




Cluster analysis by DBSCAN:



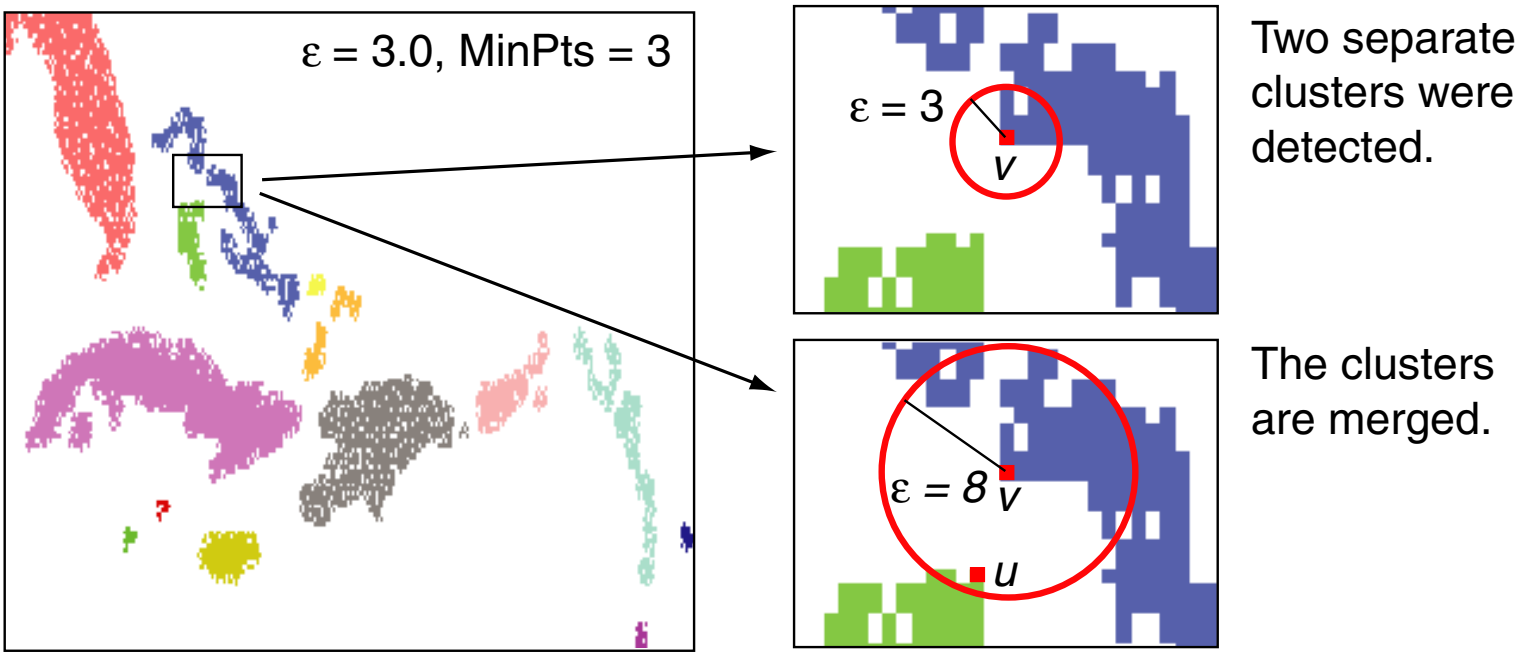




Density-Based Cluster Analysis

DBSCAN versus MajorClust: Low-Dimensional Data (continued)

The problem of finding useful ε-values for DBSCAN:



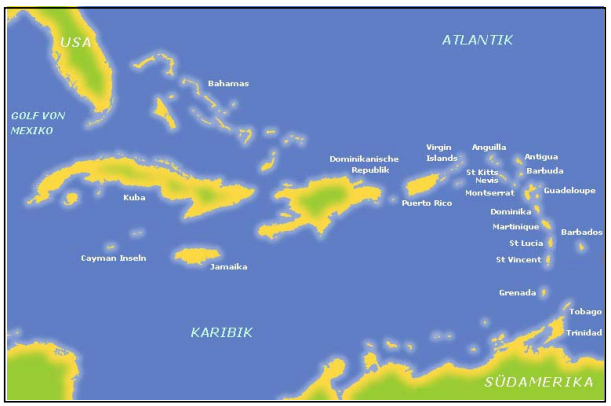
Two separate clusters were detected.

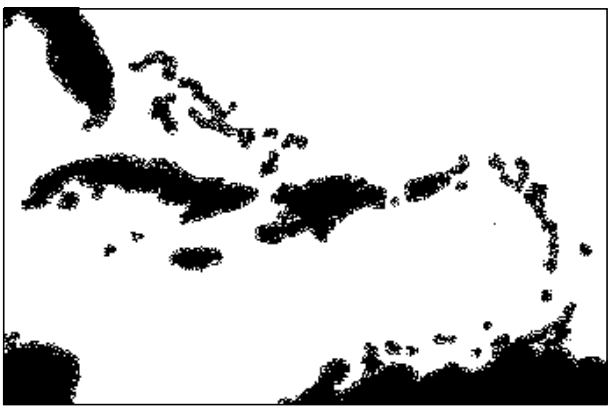
The clusters are merged.

Density-Based Cluster Analysis

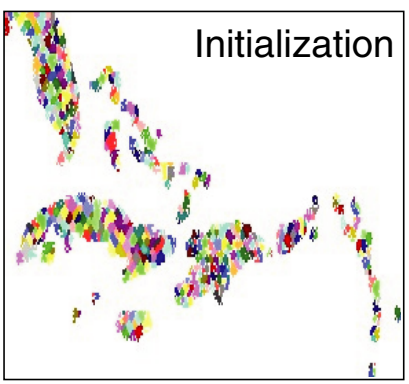
DBSCAN versus MajorClust: Low-Dimensional Data (continued)

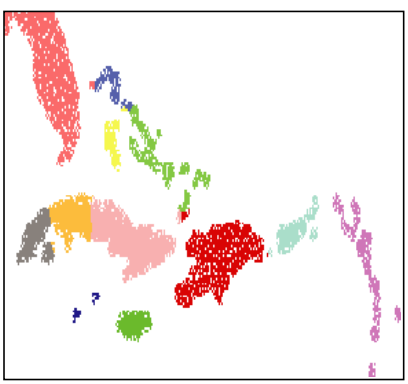
Caribbean Islands, about 20.000 points:





Cluster analysis by MajorClust:

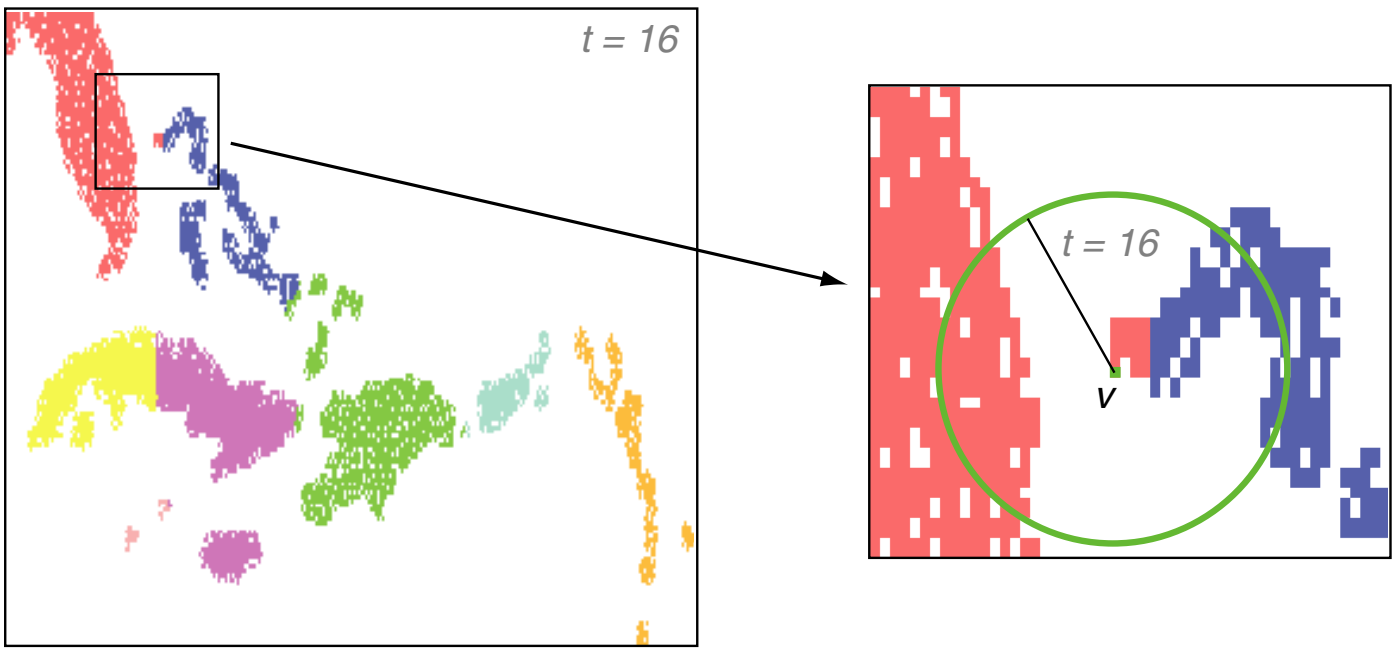




Density-Based Cluster Analysis

DBSCAN versus MajorClust: Low-Dimensional Data (continued)

The problem of the global analysis approach (no restriction of an ε-neighborhood) of MajorClust:



Density-Based Cluster Analysis

DBSCAN versus MajorClust: High-Dimensional Data

Document categorization setting using the Reuters corpus:

- 1 000 documents
- 10 categories: politics, culture, economics, etc.
- the documents are equally distributed and belong to exactly one category
- dimension of the feature space: $> 10\,000$

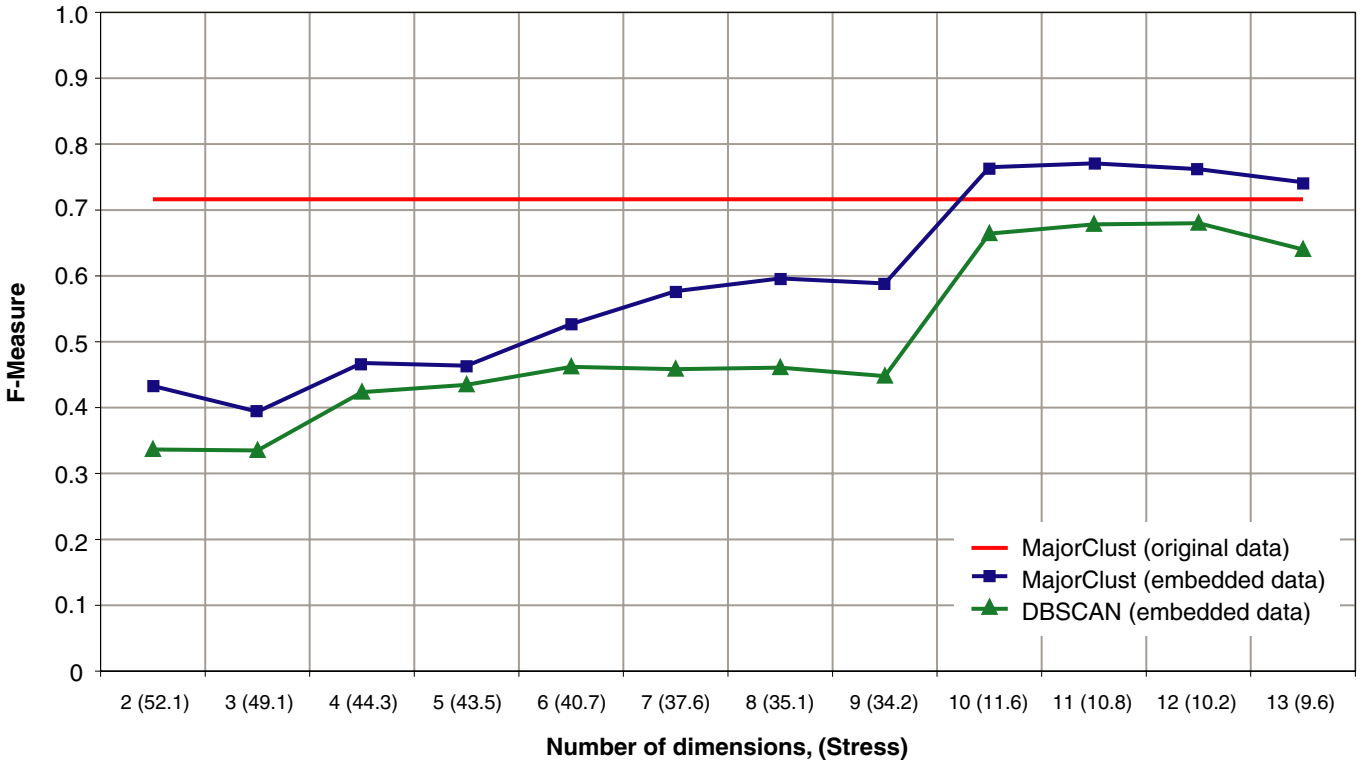
DBSCAN:

- degenerates with increasing number of dimensions
- the degeneration is rooted in the computation of the ε-neighborhood
- dimension reduction provides a way out, e.g. by embedding the data with multi-dimensional scaling, MDS

Density-Based Cluster Analysis

DBSCAN versus MajorClust: High-Dimensional Data (continued)

Classification effectiveness (F measure) over dimension number:



[Stein/Busch 2005]

Remarks:

- Usually, a neighborhood search in high-dimensional spaces cannot be solved efficiently: From a dimension number of 10-20 a linear scan of all feature vectors will be more efficient than the application of a highly specialized space partitioning data structure such as R-tree, X-tree, quadtree, KD-tree, etc.
- DBSCAN employs the R-tree data structure to determine ε-neighborhoods. This data structure accomplishes the major part of the DBSCAN cluster analysis approach and is ideally suited for treating low-dimensional data efficiently. The application of DBSCAN to high-dimensional data either requires an embedding into a low-dimensional space or to accept the runtime for a naive construction of ε-neighborhoods.
- The outlined "curse of dimensionality" can be addressed with approximative neighborhood search approaches such as locality sensitive hashing, LSH, or Fuzzy fingerprinting.
  [Weber 1999] [Gionis/Indyk/Motwani 1999-2004] [Stein 2005] [Stein/SMZE 2005]

Chapter ML:XI (continued)

XI. Cluster Analysis

- Data Mining Overview
- Cluster Analysis Basics
- Hierarchical Cluster Analysis
- Iterative Cluster Analysis
- Density-Based Cluster Analysis
- Cluster Evaluation
- Constrained Cluster Analysis

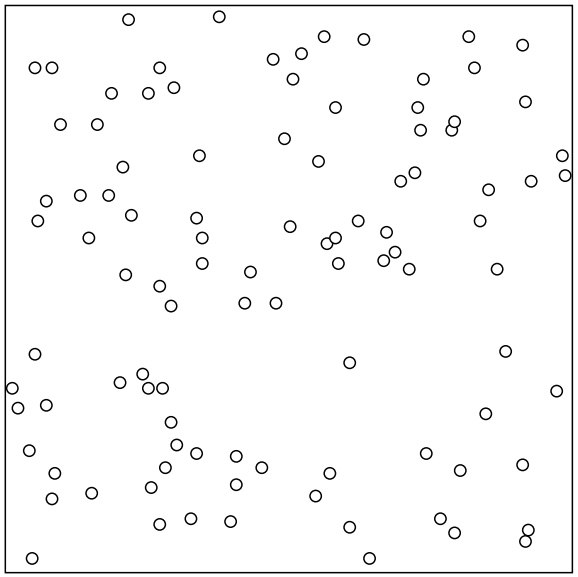
Cluster Evaluation

Overview

"The validation of clustering structures is the most difficult and frustrating part of cluster analysis. Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage."

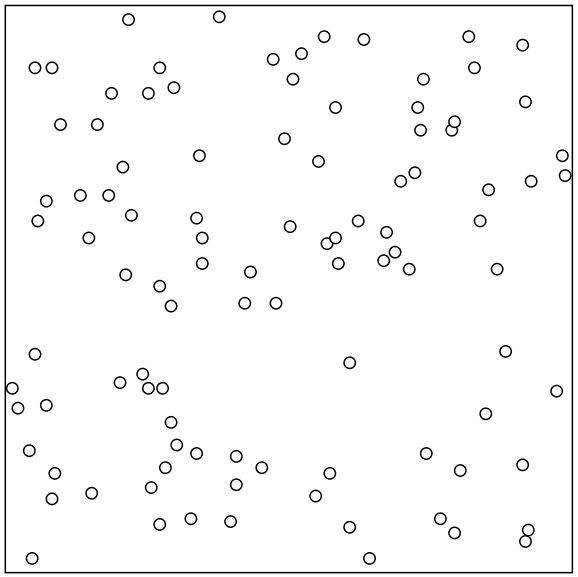
[Jain/Dubes 1990]

Cluster Evaluation [Tan/Steinbach/Kumar 2005]

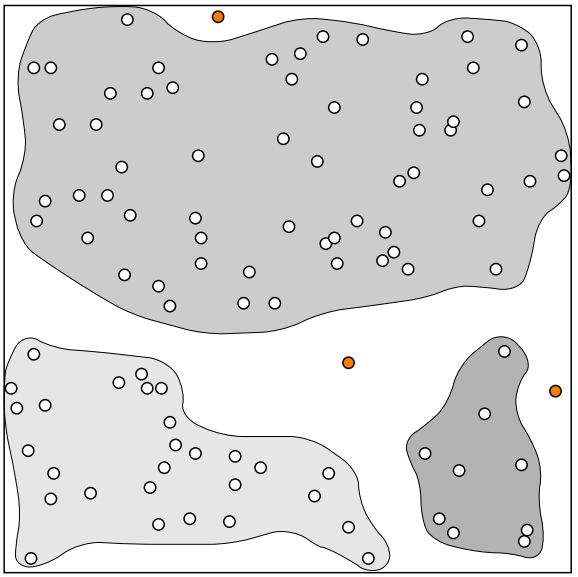


Random points

Cluster Evaluation [Tan/Steinbach/Kumar 2005]

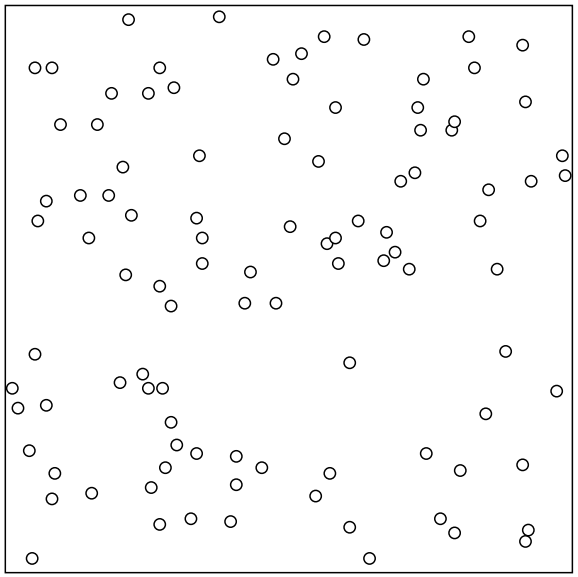


Random points

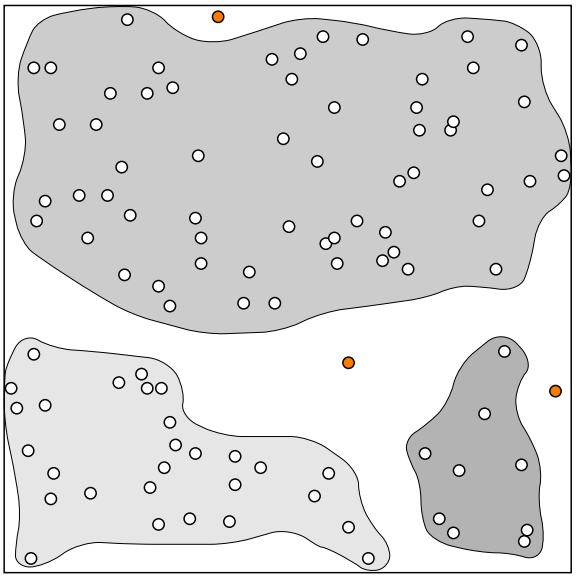


DBSCAN

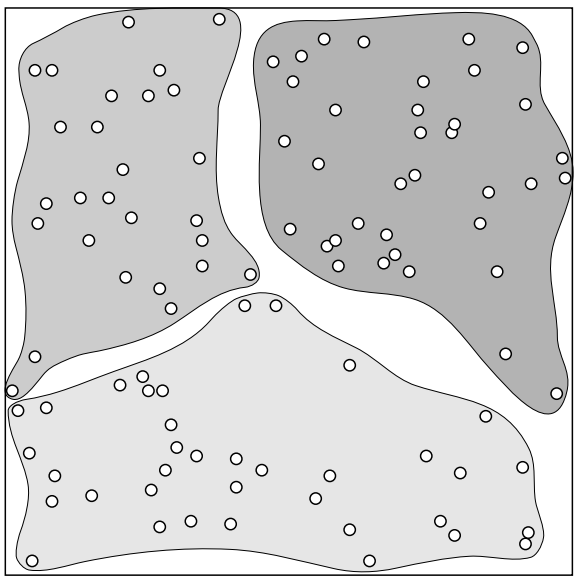
Cluster Evaluation [Tan/Steinbach/Kumar 2005]



Random points

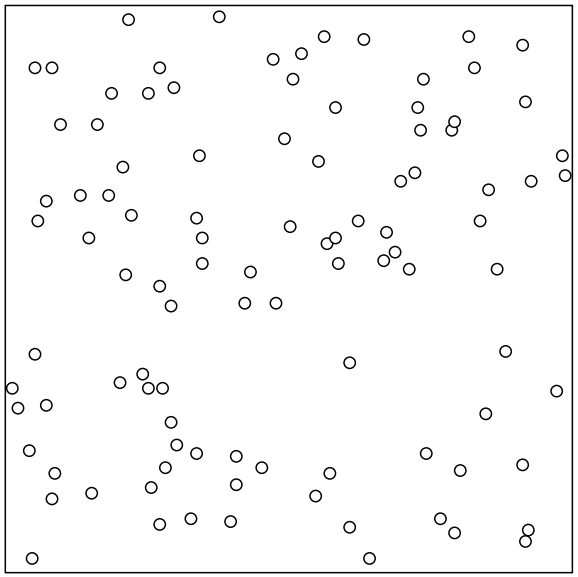


DBSCAN

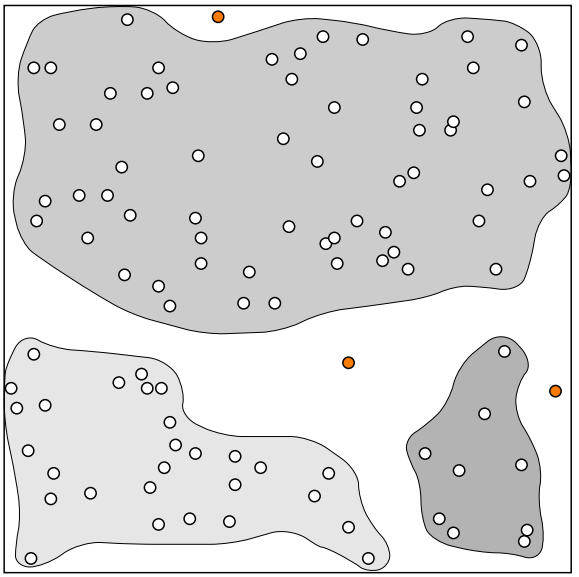


k-means

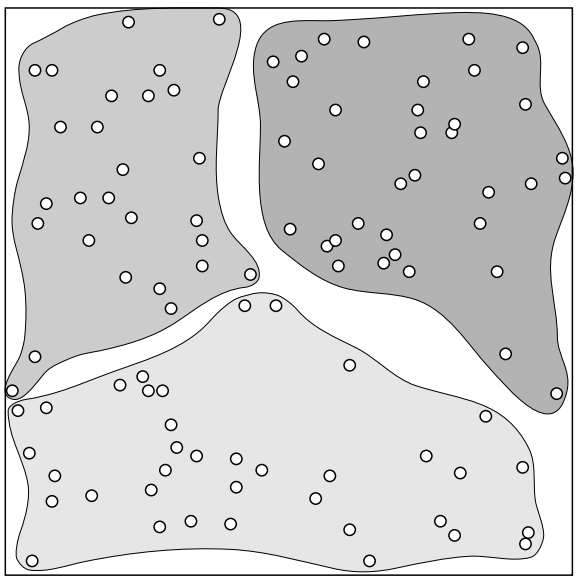
Cluster Evaluation [Tan/Steinbach/Kumar 2005]



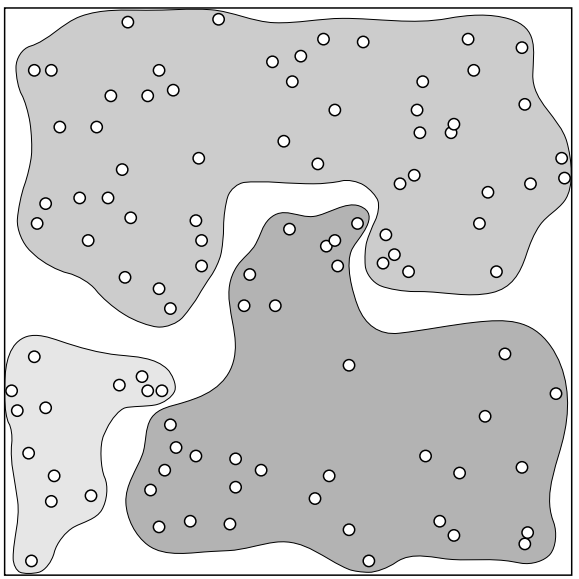
Random points



DBSCAN



k-means



Complete link

Cluster Evaluation

Overview

Cluster evaluation can address different issues:

- Provide evidence whether data contains non-random structures.
- Relate found structures in the data to externally provided class information.
- Rank alternative clusterings with regard to their quality.
- Determine the ideal number of clusters.
- Provide information to choose a suited clustering approach.

Cluster Evaluation

Overview

Cluster evaluation can address different issues:

- Provide evidence whether data contains non-random structures.
- Relate found structures in the data to externally provided class information.
- Rank alternative clusterings with regard to their quality.
- Determine the ideal number of clusters.
- Provide information to choose a suited clustering approach.

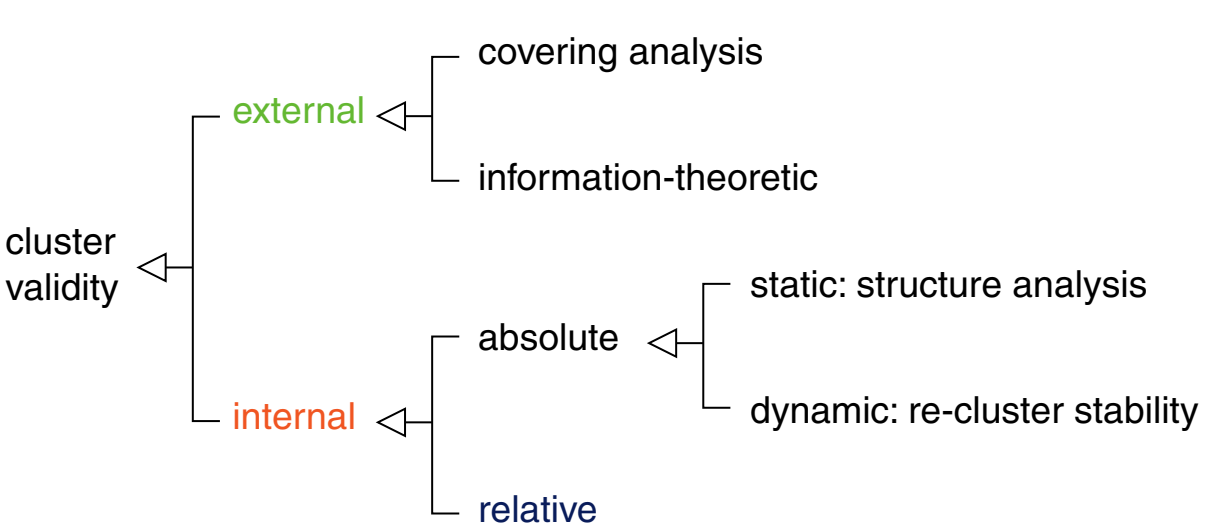
(1) External validity measures:
Analyze how close is a clustering to a reference.

(2) Internal validity measures:
Analyze intrinsic characteristics of a clustering.

(3) Relative validity measures:
Analyze the sensitivity (of internal measures) during clustering generation.

Cluster Evaluation

Overview



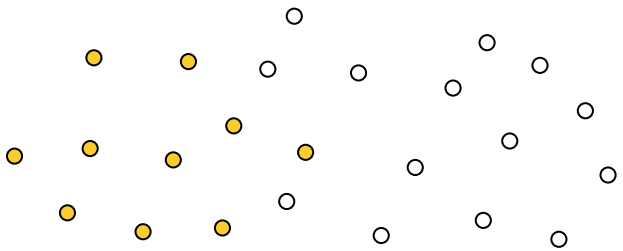
(1) External validity measures:
Analyze how close is a clustering to a reference.

(2) Internal validity measures:
Analyze intrinsic characteristics of a clustering.

(3) Relative validity measures:
Analyze the sensitivity (of internal measures) during clustering generation.

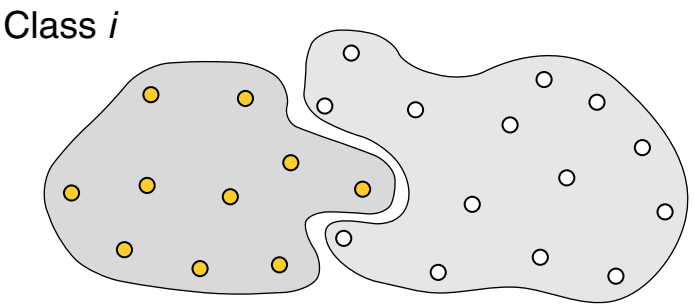
Cluster Evaluation

(1) External Validity Measures: F-Measure



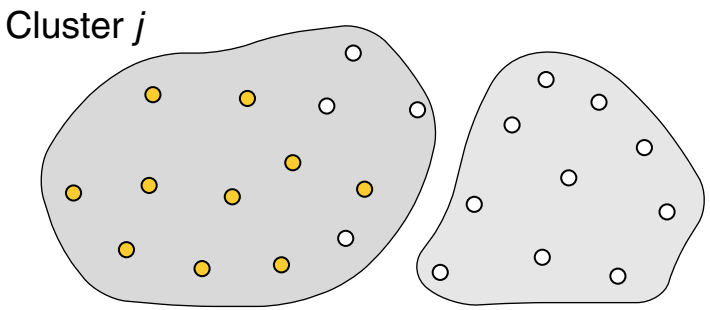
Cluster Evaluation

(1) External Validity Measures: F-Measure



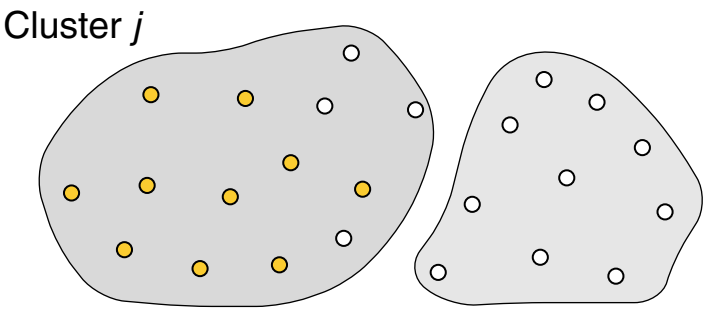
Cluster Evaluation

(1) External Validity Measures: F-Measure



Cluster Evaluation

(1) External Validity Measures: F-Measure

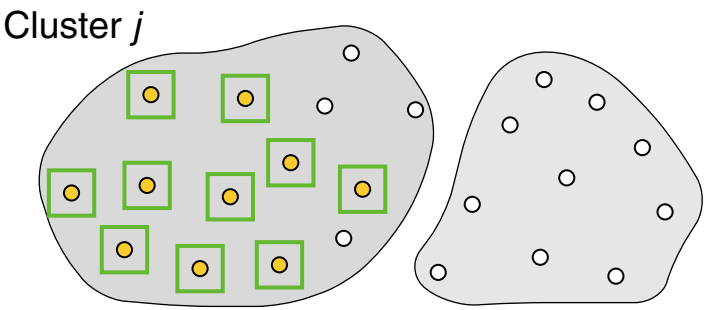


(node-based analysis)

		Truth	
		P	N
Hypothesis	P		
	N		

Cluster Evaluation

(1) External Validity Measures: F-Measure

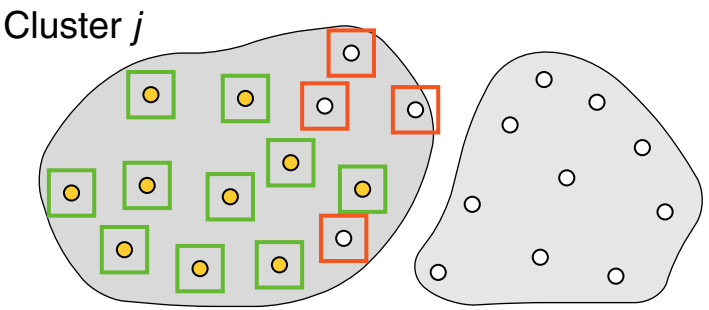


(node-based analysis)

		Truth	
		P	N
Hypothesis	P	TP (a)	
	N		

Cluster Evaluation

(1) External Validity Measures: F-Measure

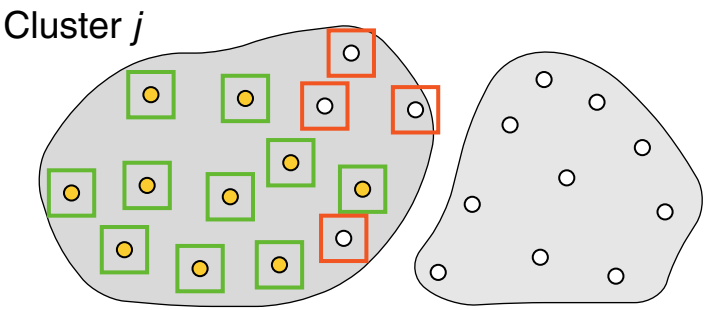


(node-based analysis)

		Truth	
		P	N
Hypothesis	P	TP (a)	FP (b)
	N		

Cluster Evaluation

(1) External Validity Measures: F-Measure

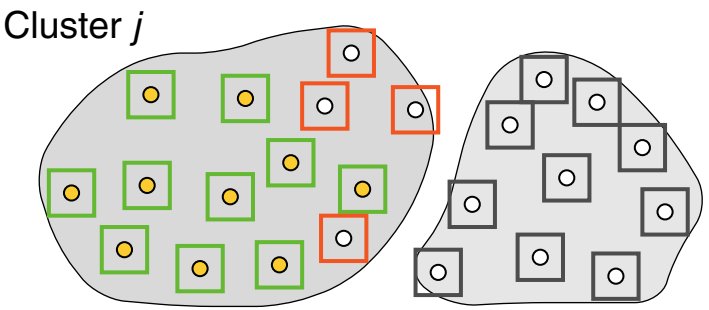


(node-based analysis)

		Truth	
		P	N
Hypothesis	P	TP (a)	FP (b)
	N	FN (c)	

Cluster Evaluation

(1) External Validity Measures: F-Measure

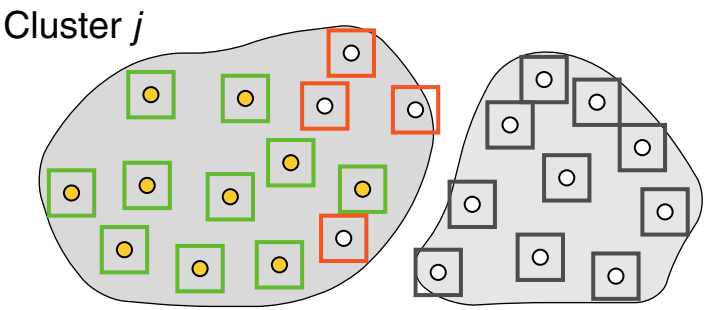


(node-based analysis)

		Truth	
		P	N
Hypothesis	P	TP (a)	FP (b)
	N	FN (c)	TN (d)

Cluster Evaluation

(1) External Validity Measures: F-Measure



(node-based analysis)

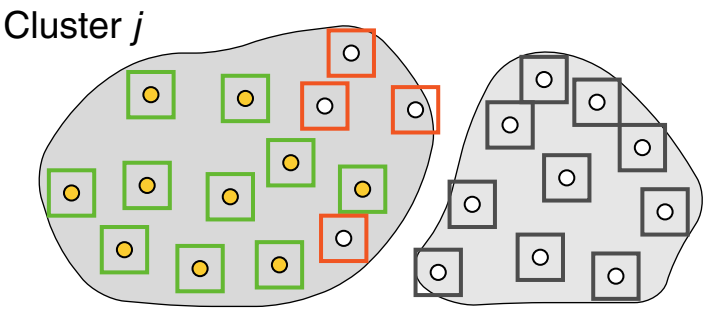
		Truth	
		P	N
Hypothesis	P	TP (a)	FP (b)
	N	FN (c)	TN (d)

Precision: $\frac{a}{a+b}$

Recall: $\frac{a}{a+c}$

Cluster Evaluation

(1) External Validity Measures: F-Measure



(node-based analysis)

		Truth	
		P	N
Hypothesis	P	TP (a)	FP (b)
	N	FN (c)	TN (d)

Precision: $\frac{a}{a+b}$

Recall: $\frac{a}{a+c}$

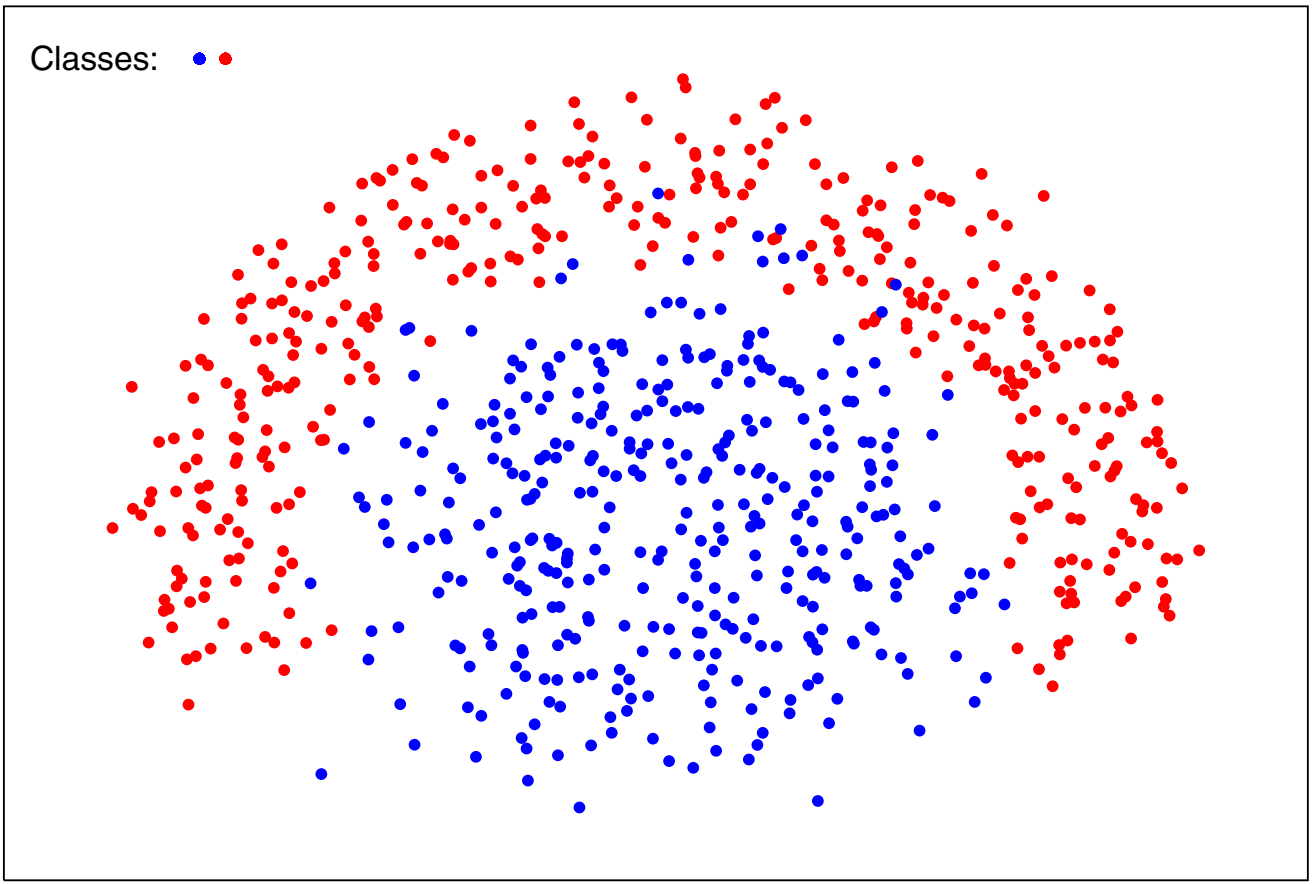
F-measure:

$$F_\alpha = \frac{1+\alpha}{\frac{1}{\textit{precision}} + \frac{\alpha}{\textit{recall}}}$$

$\alpha = 1$	harmonic mean
$\alpha \in (0;1)$	favor precision over recall
$\alpha > 1$	favor recall over precision

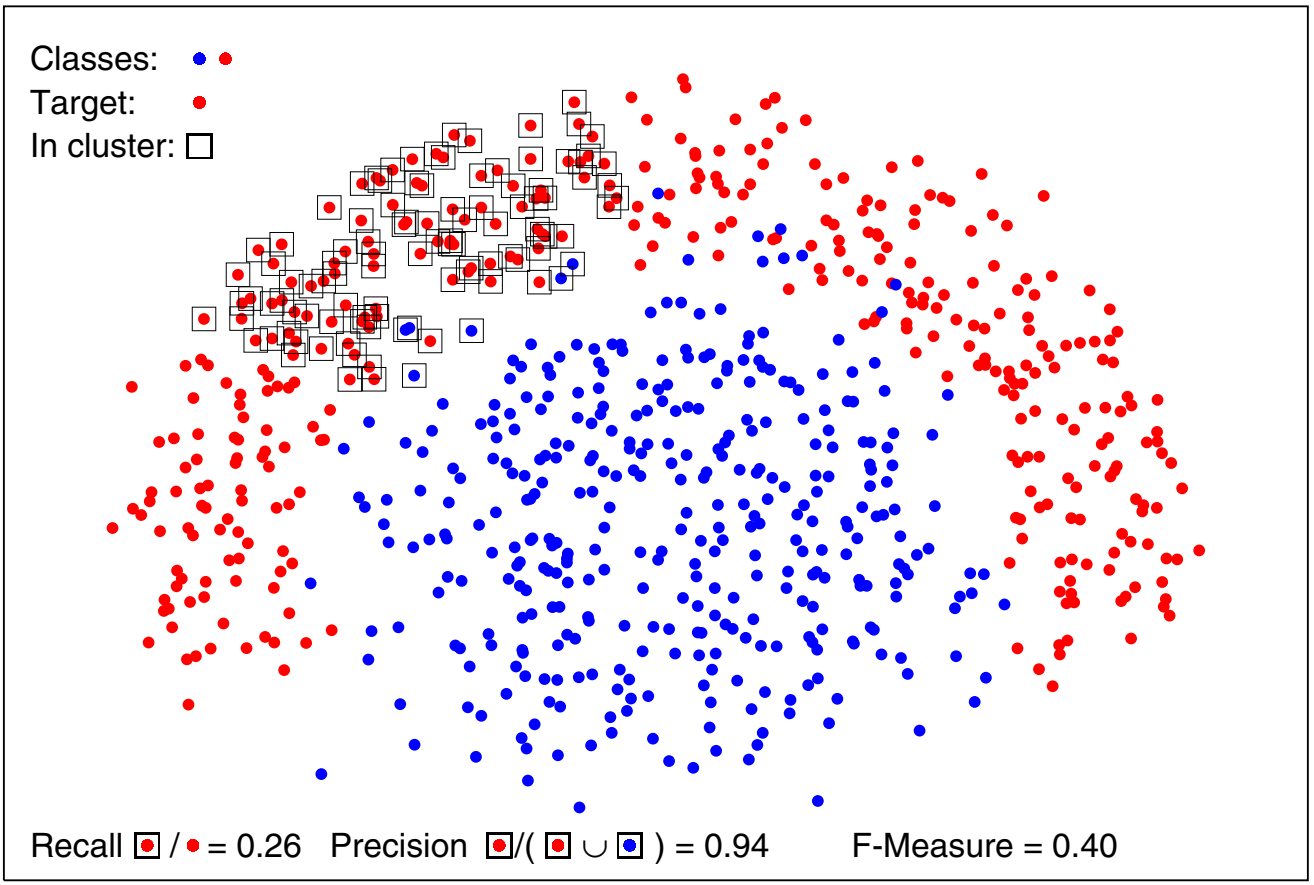
Cluster Evaluation

(1) External Validity Measures: F-Measure



Cluster Evaluation

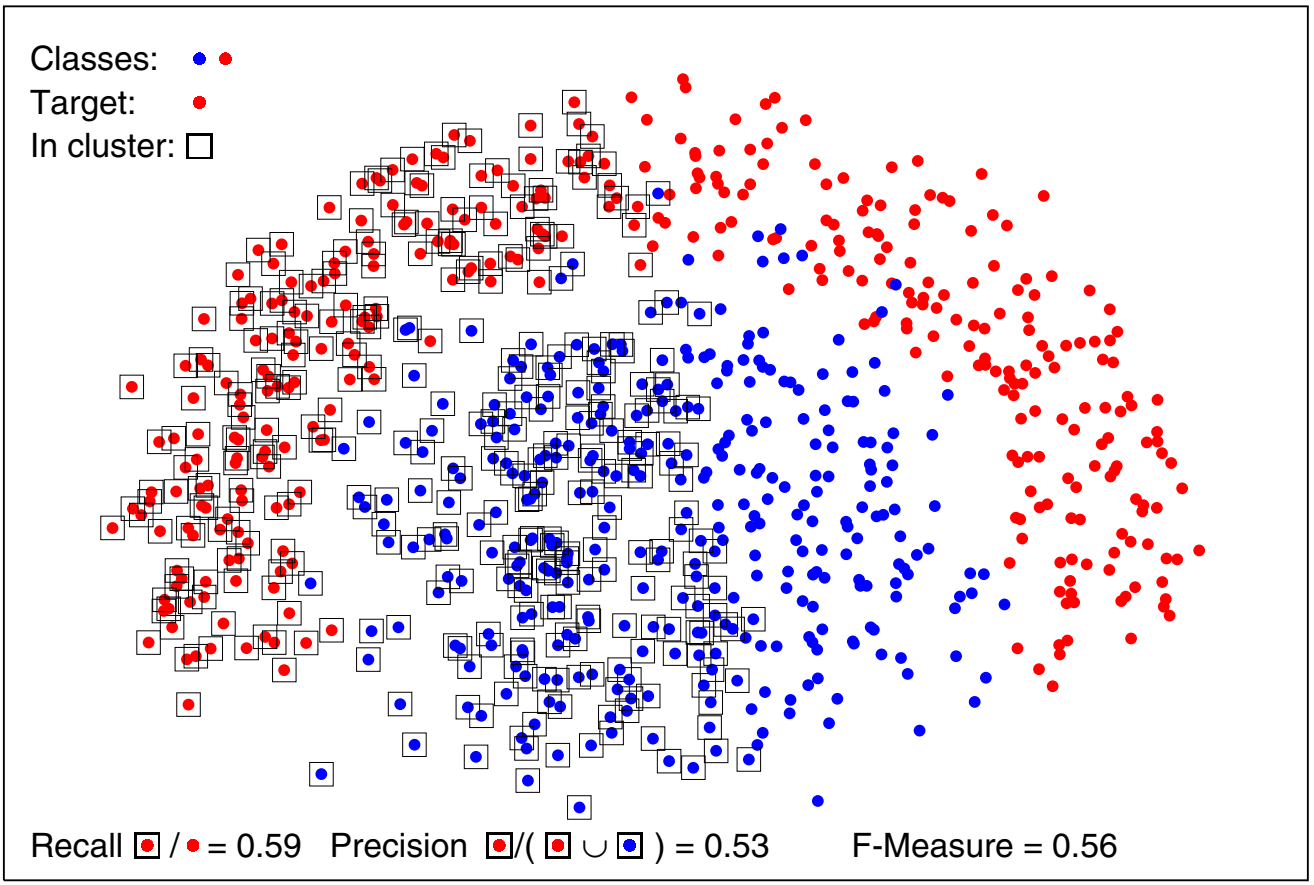
(1) External Validity Measures: F-Measure



High precision, low recall $\Rightarrow$ low F-measure.

Cluster Evaluation

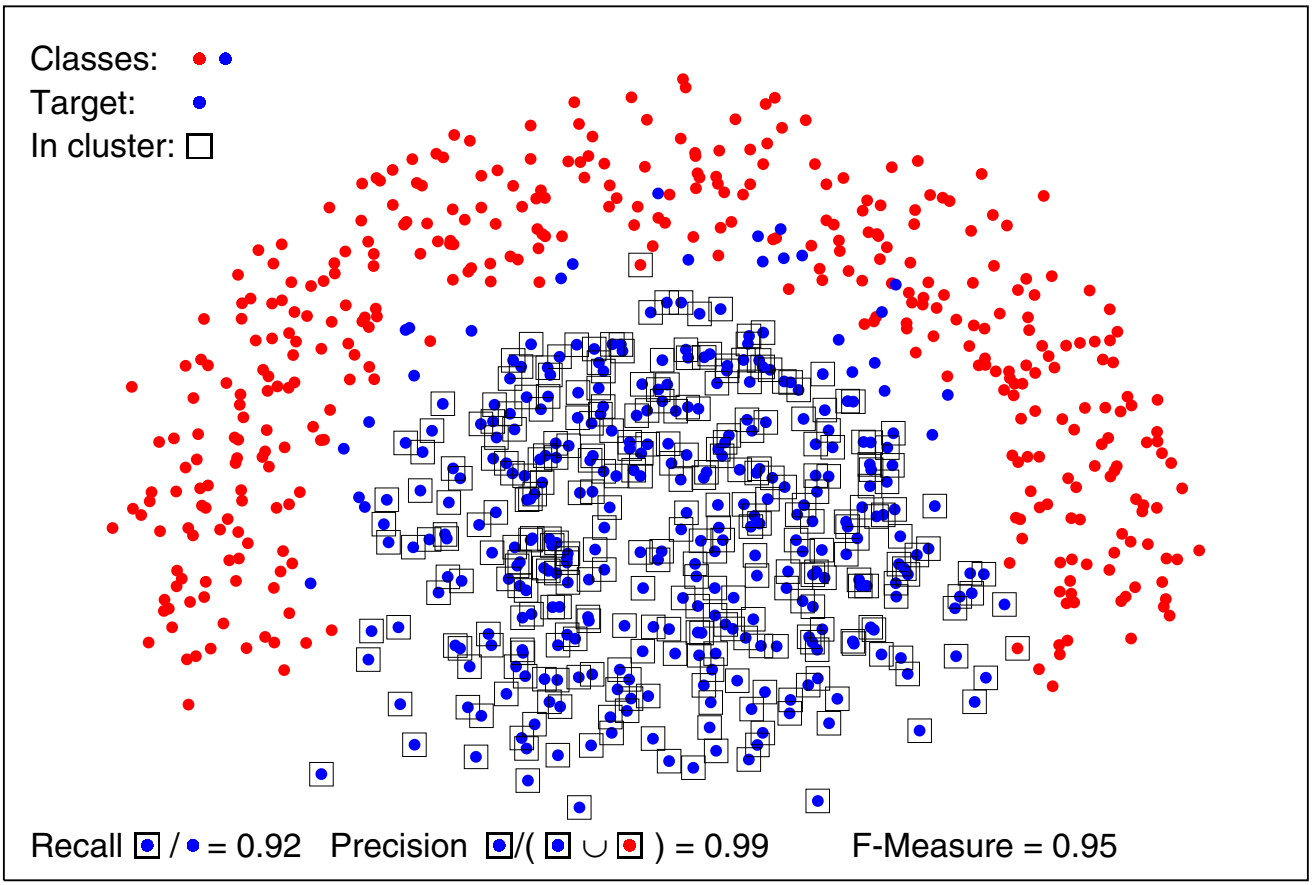
(1) External Validity Measures: F-Measure



Low precision, low recall $\Rightarrow$ low F-measure.

Cluster Evaluation

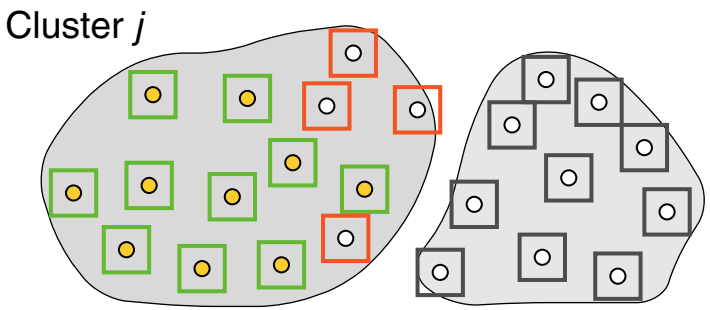
(1) External Validity Measures: F-Measure



High precision, high recall $\Rightarrow$ high F-measure.

Cluster Evaluation

(1) External Validity Measures: F-Measure



(node-based analysis)

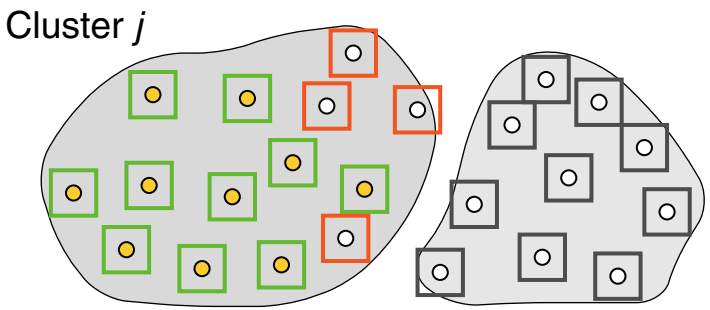
- Clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$ and classification $\mathcal{C}^* = \{C_1^*, \ldots, C_l^*\}$ of D.
- $F_{i,j}$ is the F-measure of a cluster j computed *with respect to a class* i.

  Recall of cluster j with respect to class i is $|C_j \cap C_i^*|/|C_i^*|$ (here: $Rec_{i,j} = 1.0$)

  Precision of cluster j with respect to class i is $|C_j \cap C_i^*|/|C_j|$ (here: $Prec_{i,j} = 0.71$)

Cluster Evaluation

(1) External Validity Measures: F-Measure



(node-based analysis)

- Clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$ and classification $\mathcal{C}^* = \{C_1^*, \ldots, C_l^*\}$ of D.
- $F_{i,j}$ is the F-measure of a cluster j computed *with respect to a class* i.

  Recall of cluster j with respect to class i is $|C_j \cap C_i^*|/|C_i^*|$ (here: $\mathit{Rec}_{i,j} = 1.0$)

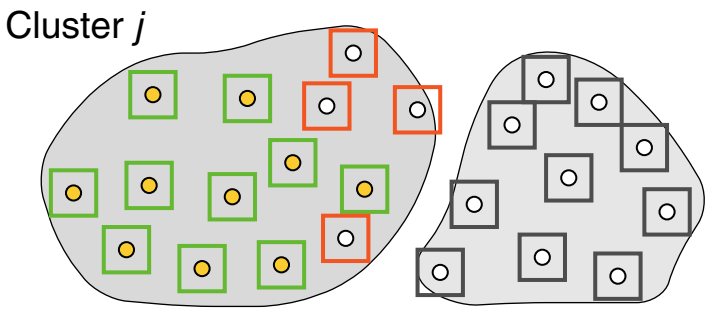
  Precision of cluster j with respect to class i is $|C_j \cap C_i^*|/|C_j|$ (here: $\mathit{Prec}_{i,j} = 0.71$)

- Micro-averaged F-measure for $\langle D, \mathcal{C}, \mathcal{C}^* \rangle$:

$$F = \sum_{i=1}^{l} \frac{|C_i^*|}{|D|} \cdot \max_{j=1,\ldots,k} \{F_{i,j}\}$$

Cluster Evaluation

(1) External Validity Measures: F-Measure



(node-based analysis)

- Clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$ and classification $\mathcal{C}^* = \{C_1^*, \ldots, C_l^*\}$ of D.
- $F_{i,j}$ is the F-measure of a cluster j computed *with respect to a class* i.

  Recall of cluster j with respect to class i is $|C_j \cap C_i^*|/|C_i^*|$ (here: $\mathit{Rec}_{i,j} = 1.0$)

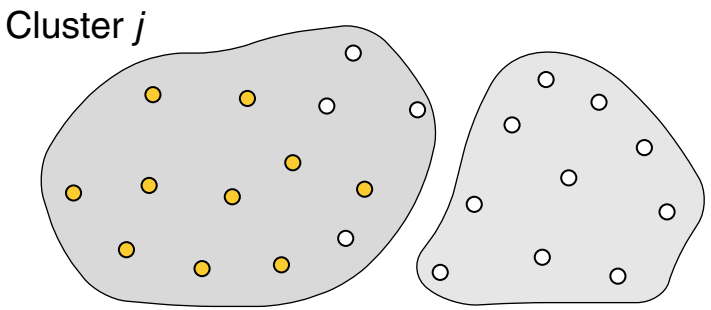
  Precision of cluster j with respect to class i is $|C_j \cap C_i^*|/|C_j|$ (here: $\mathit{Prec}_{i,j} = 0.71$)

- Macro-averaged F-measure for $\langle D, \mathcal{C}, \mathcal{C}^* \rangle$:

$$F = \frac{1}{l} \sum_{i=1}^{l} \max_{j=1,\ldots,k} \{F_{i,j}\}$$

Cluster Evaluation

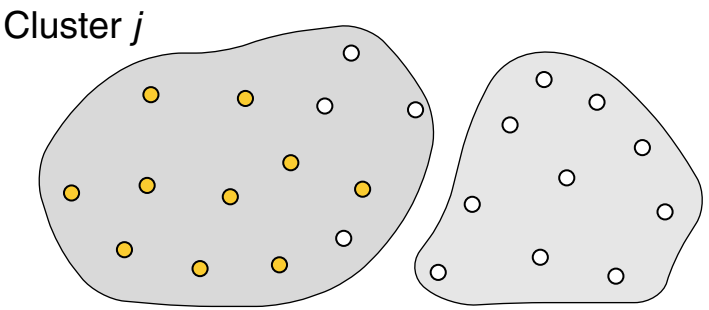
(1) External Validity Measures: Entropy



(node-based analysis)

Cluster Evaluation

(1) External Validity Measures: Entropy

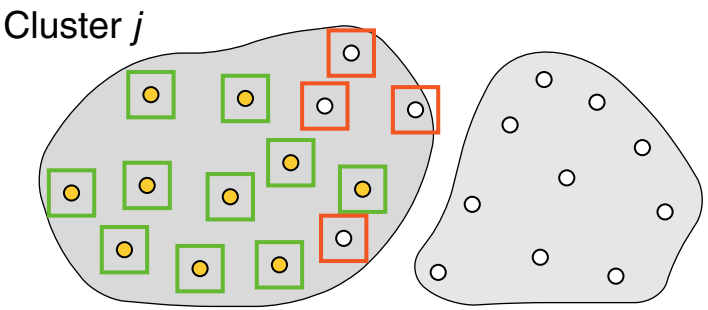


(node-based analysis)

- A cluster C acts as information source $\mathcal{L}$.
  $\mathcal{L}$ emits cluster labels $L_1, \ldots, L_l$ with probabilities $P(L_1), \ldots, P(L_l)$.

Cluster Evaluation

(1) External Validity Measures: Entropy



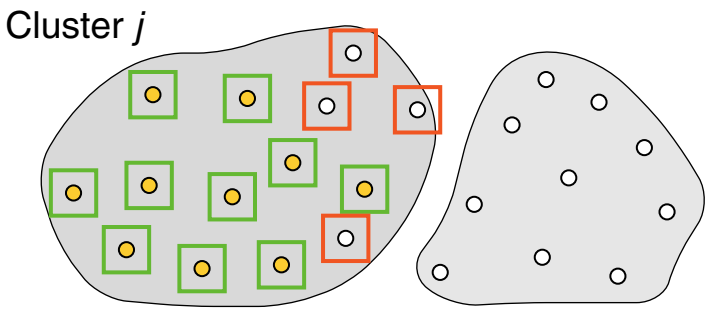
(node-based analysis)

- A cluster C acts as information source $\mathcal{L}$.
  $\mathcal{L}$ emits cluster labels $L_1, \ldots, L_l$ with probabilities $P(L_1), \ldots, P(L_l)$.

  $\hat{P}(\square) = 10/14, \quad \hat{P}(\square) = 4/14$

Cluster Evaluation

(1) External Validity Measures: Entropy



(node-based analysis)

- A cluster C acts as information source $\mathcal{L}$.
  $\mathcal{L}$ emits cluster labels $L_1, \ldots, L_l$ with probabilities $P(L_1), \ldots, P(L_l)$.

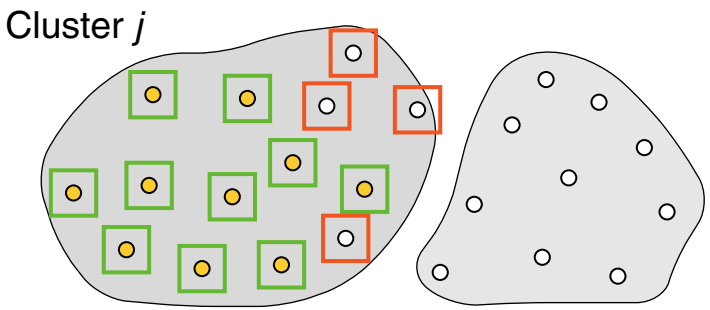
  $\hat{P}(\square) = 10/14, \quad \hat{P}(\square) = 4/14$

- Entropy of $\mathcal{L}$: $H(\mathcal{L}) = -\sum_{i=1}^{l} P(L_i) \cdot \log_2(P(L_i))$

  Entropy of C_j wrt. $\mathcal{C}^*$: $H(C_j) = -\sum_{C_j \cap C_i^* \neq \emptyset} |C_j \cap C_i^*|/|C_j| \cdot \log_2(|C_j \cap C_i^*|/|C_j|)$

Cluster Evaluation

(1) External Validity Measures: Entropy



(node-based analysis)

- A cluster C acts as information source $\mathcal{L}$.
  $\mathcal{L}$ emits cluster labels $L_1, \ldots, L_l$ with probabilities $P(L_1), \ldots, P(L_l)$.

  $\hat{P}(\square) = 10/14, \quad \hat{P}(\square) = 4/14$

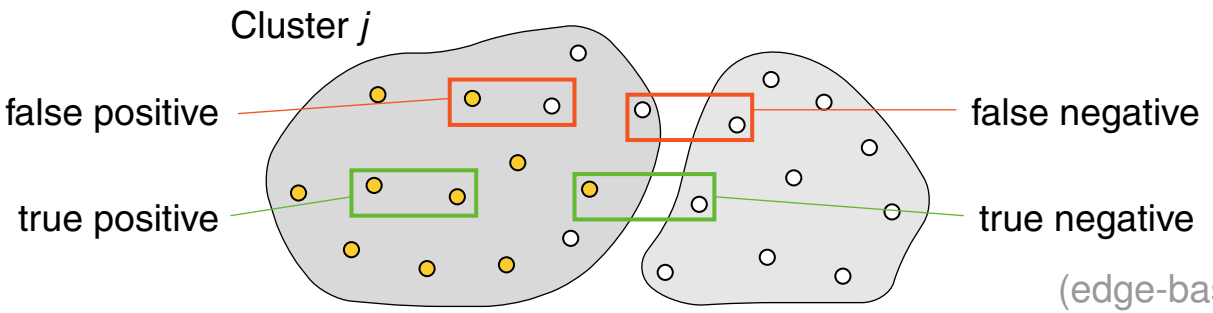
- Entropy of $\mathcal{L}$: $H(\mathcal{L}) = -\sum_{i=1}^{l} P(L_i) \cdot \log_2(P(L_i))$

  Entropy of C_j wrt. $\mathcal{C}^*$: $H(C_j) = -\sum_{C_j \cap C_i^* \neq \emptyset} |C_j \cap C_i^*|/|C_j| \cdot \log_2(|C_j \cap C_i^*|/|C_j|)$

- Entropy of $\mathcal{C}$ wrt. $\mathcal{C}^*$: $H(\mathcal{C}) = \sum_{C_j \in \mathcal{C}} |C_j|/|D| \cdot H(C_j)$

Cluster Evaluation

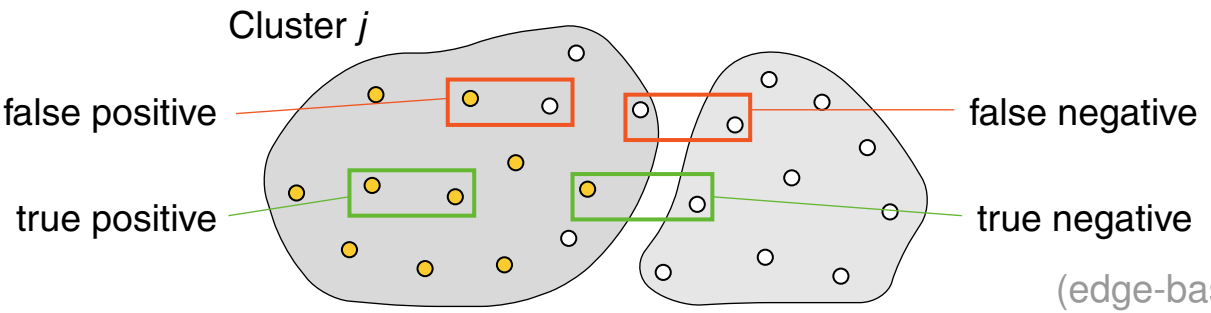
(1) External Validity Measures: Rand, Jaccard



(edge-based analysis)

Cluster Evaluation

(1) External Validity Measures: Rand, Jaccard



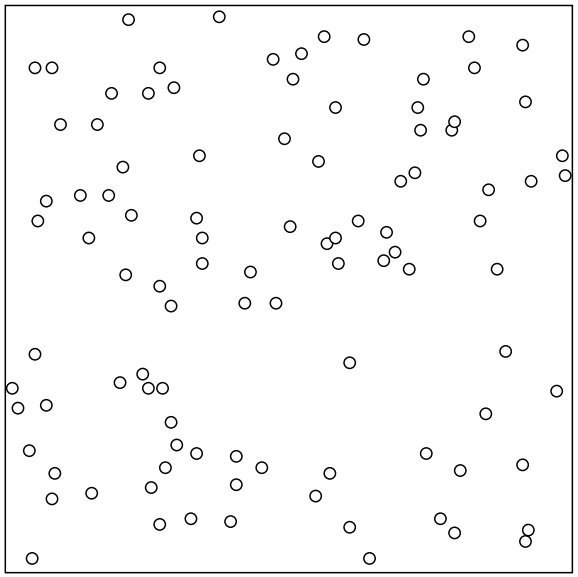
(edge-based analysis)

- $R(\mathcal{C}) = \dfrac{|TP| + |TN|}{|TP| + |TN| + |FP| + |FN|} = \dfrac{|TP| + |TN|}{n(n-1)/2}$, with $n = |D|$

- $J(\mathcal{C}) = \dfrac{|TP|}{|TP| + |FP| + |FN|}$

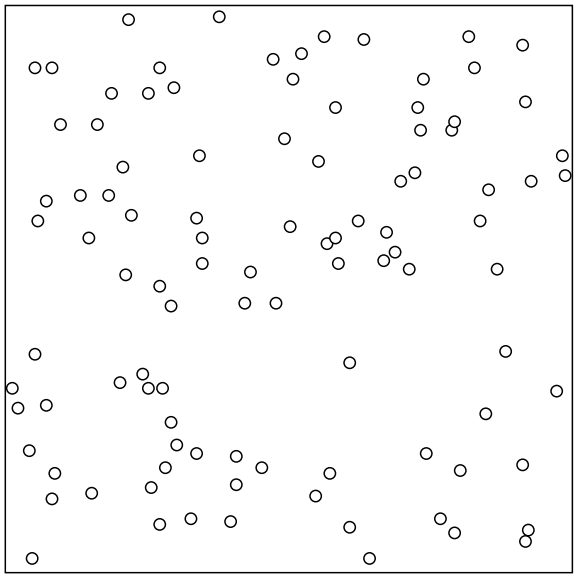
Cluster Evaluation

(2) Internal Validity Measures: Edge Correlation [Tan/Steinbach/Kumar 2005]



Cluster Evaluation

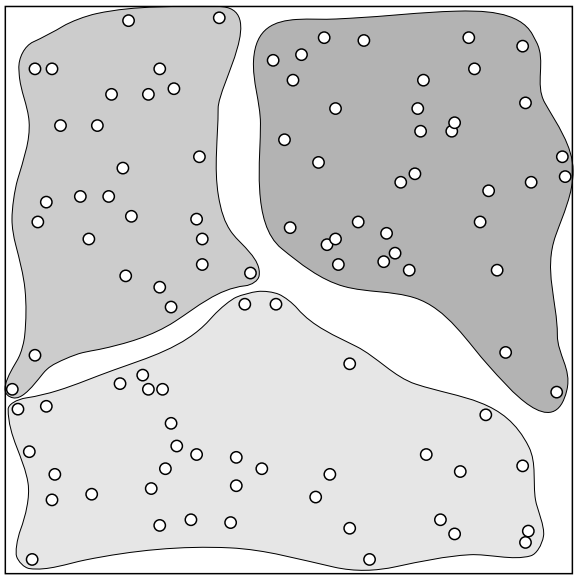
(2) Internal Validity Measures: Edge Correlation [Tan/Steinbach/Kumar 2005]



$$\begin{pmatrix} 1.0 & 0.2 & 0.1 & 0.3 & \dots & 0.1 & 0.0 \\ - & 1.0 & 0.1 & 0.0 & \dots & 0.0 & 0.2 \\ & & & & \vdots & & \\ - & - & - & - & - & 1.0 & 0.6 \\ - & - & - & - & - & - & 1.0 \end{pmatrix}$$

Cluster Evaluation

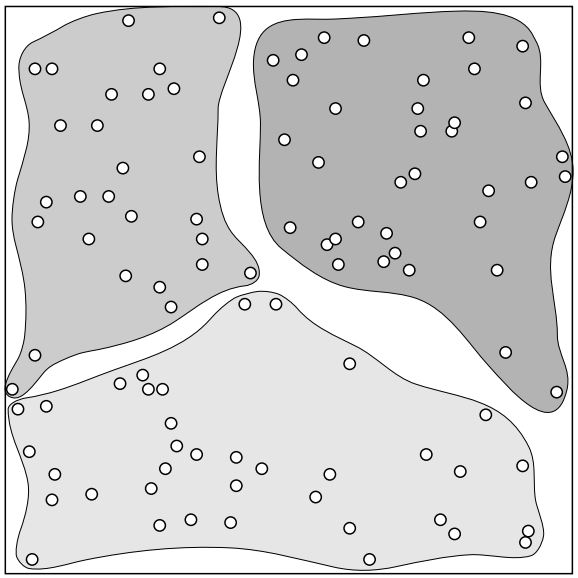
(2) Internal Validity Measures: Edge Correlation [Tan/Steinbach/Kumar 2005]



$$\begin{pmatrix} 1.0 & 0.2 & 0.1 & 0.3 & \dots & 0.1 & 0.0 \\ - & 1.0 & 0.1 & 0.0 & \dots & 0.0 & 0.2 \\ & & & & \vdots & & \\ - & - & - & - & - & 1.0 & 0.6 \\ - & - & - & - & - & - & 1.0 \end{pmatrix}$$

Cluster Evaluation

(2) Internal Validity Measures: Edge Correlation [Tan/Steinbach/Kumar 2005]

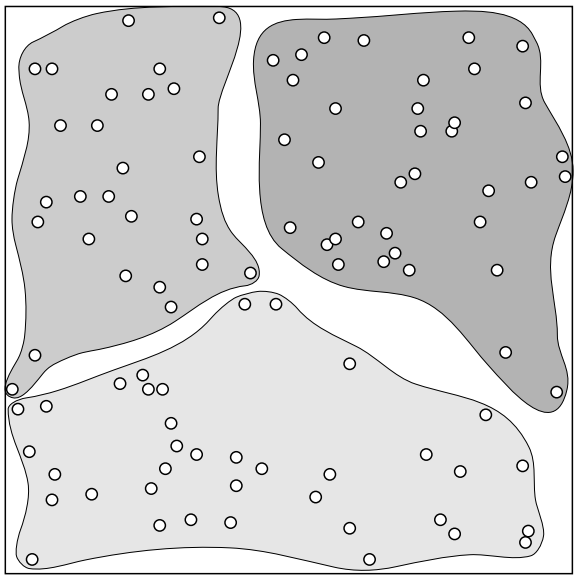


$$\begin{pmatrix} 1.0 & 0.2 & 0.1 & 0.3 & \dots & 0.1 & 0.0 \\ - & 1.0 & 0.1 & 0.0 & \dots & 0.0 & 0.2 \\ & & & & \vdots & & \\ - & - & - & - & - & 1.0 & 0.6 \\ - & - & - & - & - & - & 1.0 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0 & 1 & \dots & 0 & 0 \\ - & 1 & 0 & 0 & \dots & 0 & 1 \\ & & & & \vdots & & \\ - & - & - & - & - & 1 & 1 \\ - & - & - & - & - & - & 1 \end{pmatrix}$$

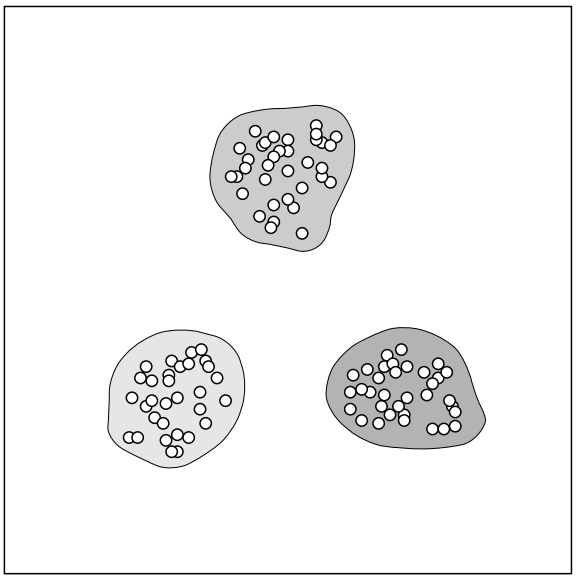
- Construct occurrence matrix based on cluster analysis.
- Compare similarity matrix to occurrence matrix: correlation τ

Cluster Evaluation

(2) Internal Validity Measures: Edge Correlation [Tan/Steinbach/Kumar 2005]



k-means
$\tau = 0.58$



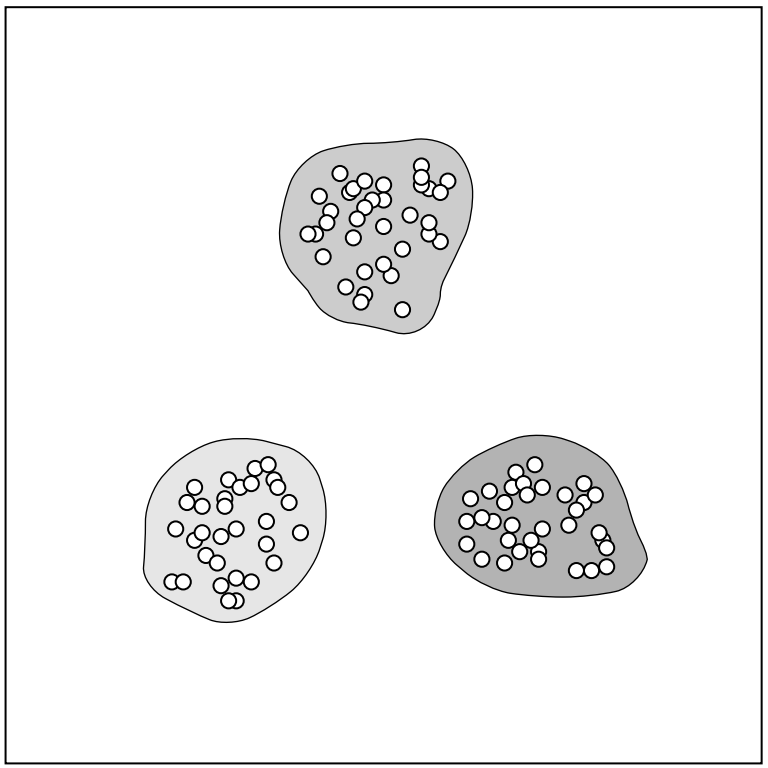
k-means
$\tau = 0.92$

$$\begin{pmatrix} 1.0 & 0.2 & 0.1 & 0.3 & \dots & 0.1 & 0.0 \\ - & 1.0 & 0.1 & 0.0 & \dots & 0.0 & 0.2 \\ & & & & \vdots & & \\ - & - & - & - & - & 1.0 & 0.6 \\ - & - & - & - & - & - & 1.0 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0 & 1 & \dots & 0 & 0 \\ - & 1 & 0 & 0 & \dots & 0 & 1 \\ & & & & \vdots & & \\ - & - & - & - & - & 1 & 1 \\ - & - & - & - & - & - & 1 \end{pmatrix}$$

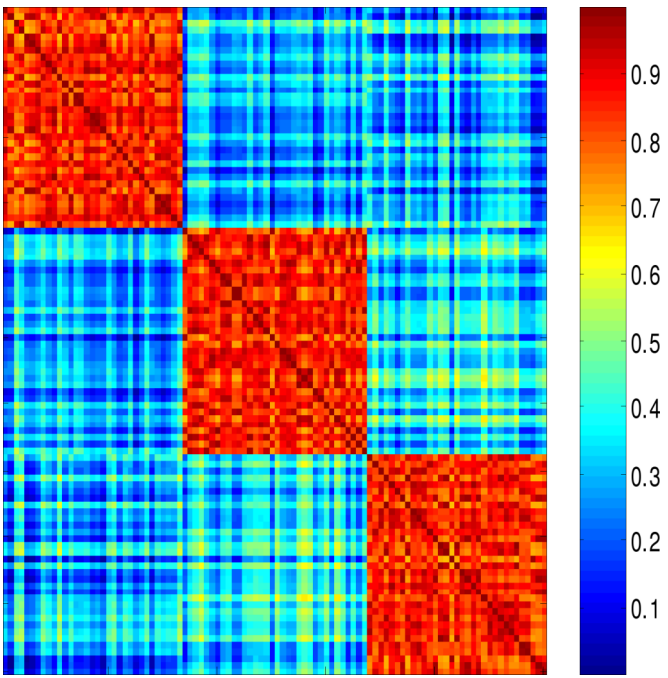
- Construct occurrence matrix based on cluster analysis.
- Compare similarity matrix to occurrence matrix: correlation τ

Cluster Evaluation

(2) Internal Validity Measures: Edge Correlation [Tan/Steinbach/Kumar 2005]



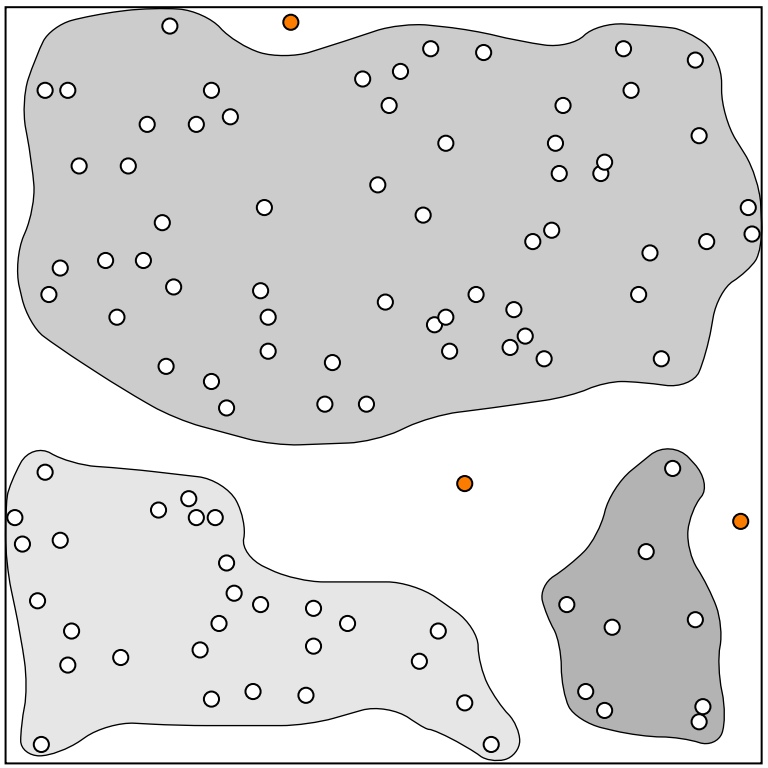
k-means at structured data.



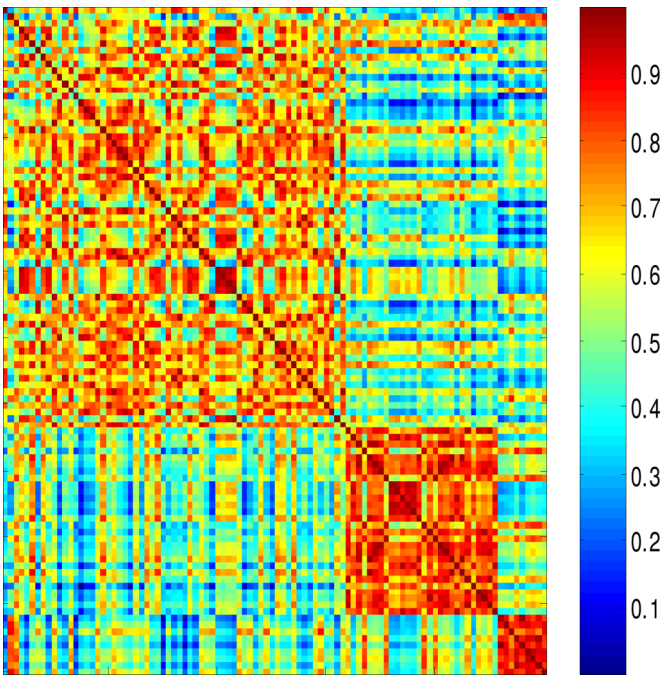
Similarity matrix sorted by cluster label.

Cluster Evaluation

(2) Internal Validity Measures: Edge Correlation [Tan/Steinbach/Kumar 2005]



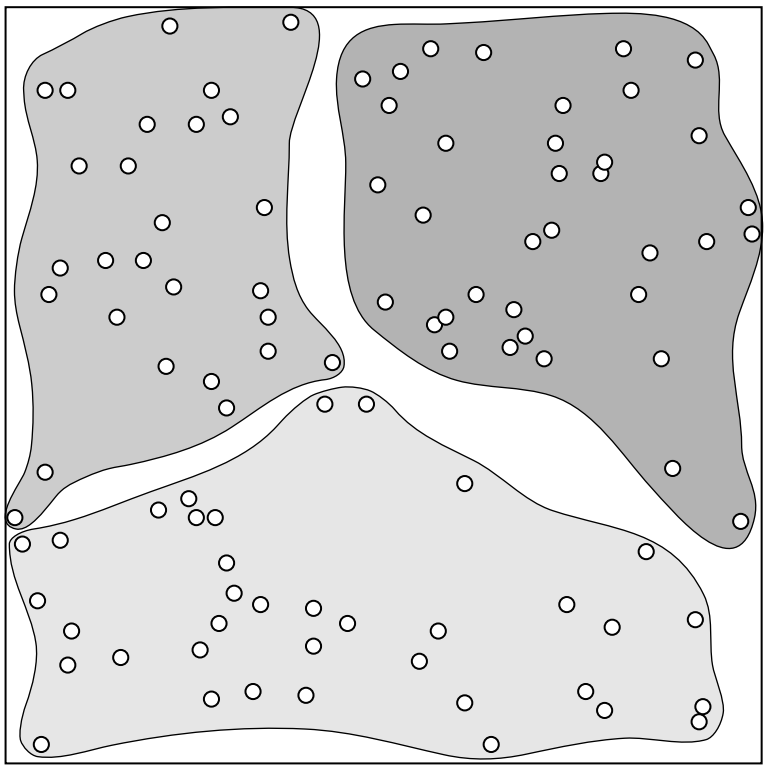
DBSCAN at random data.



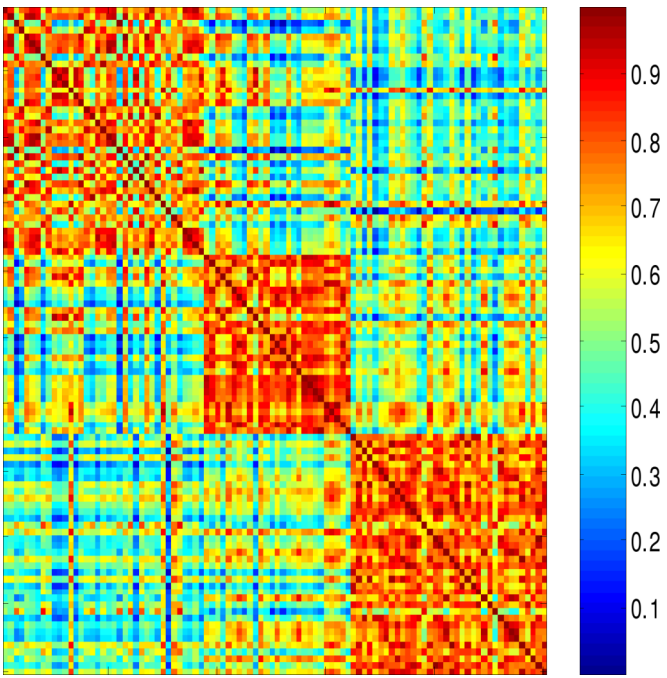
Similarity matrix sorted by cluster label.

Cluster Evaluation

(2) Internal Validity Measures: Edge Correlation [Tan/Steinbach/Kumar 2005]



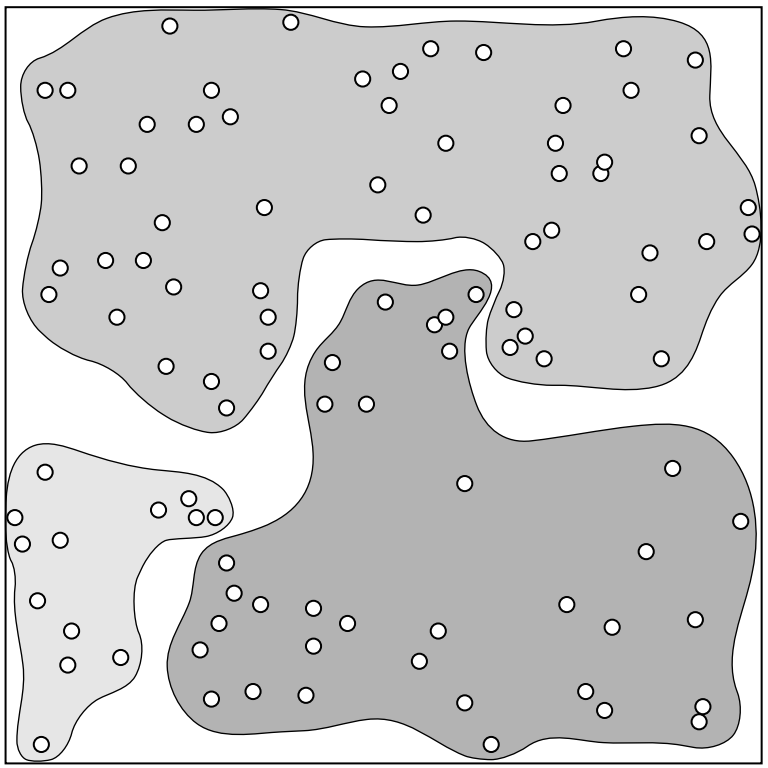
k-means at random data.



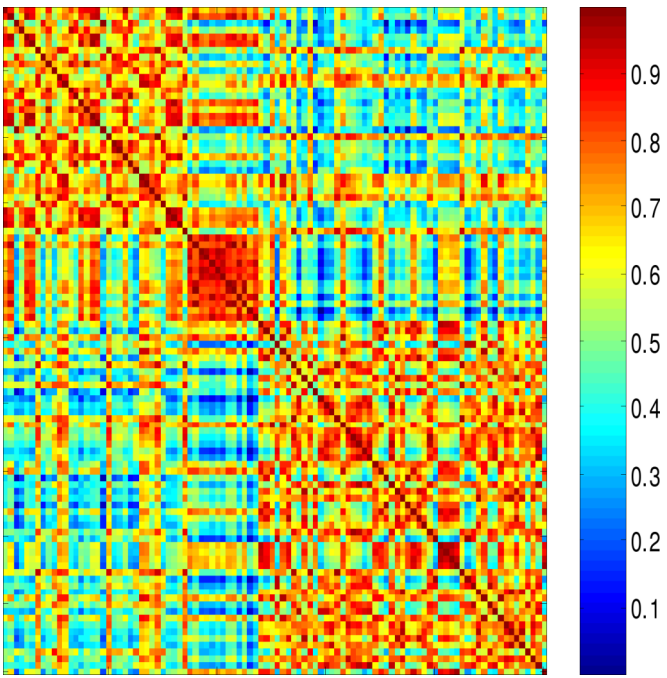
Similarity matrix sorted by cluster label.

Cluster Evaluation

(2) Internal Validity Measures: Edge Correlation [Tan/Steinbach/Kumar 2005]



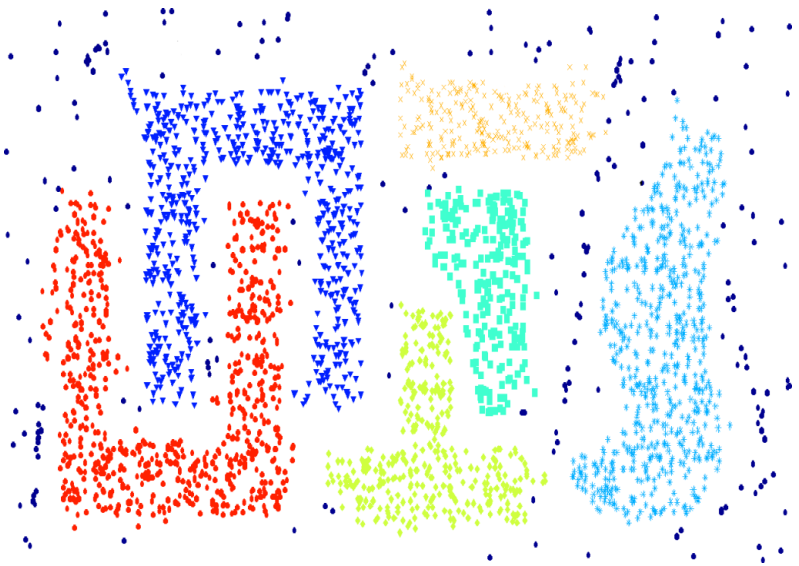
Complete link at random data.



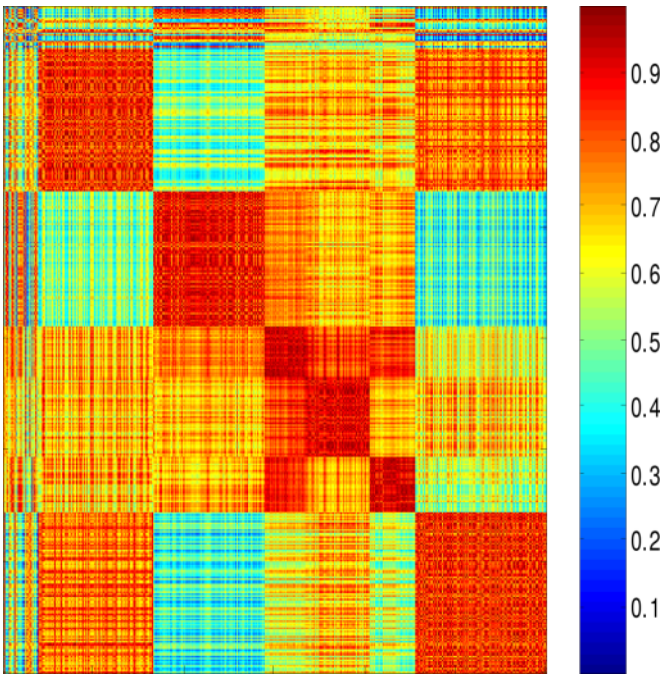
Similarity matrix sorted by cluster label.

Cluster Evaluation

(2) Internal Validity Measures: Edge Correlation [Tan/Steinbach/Kumar 2005]



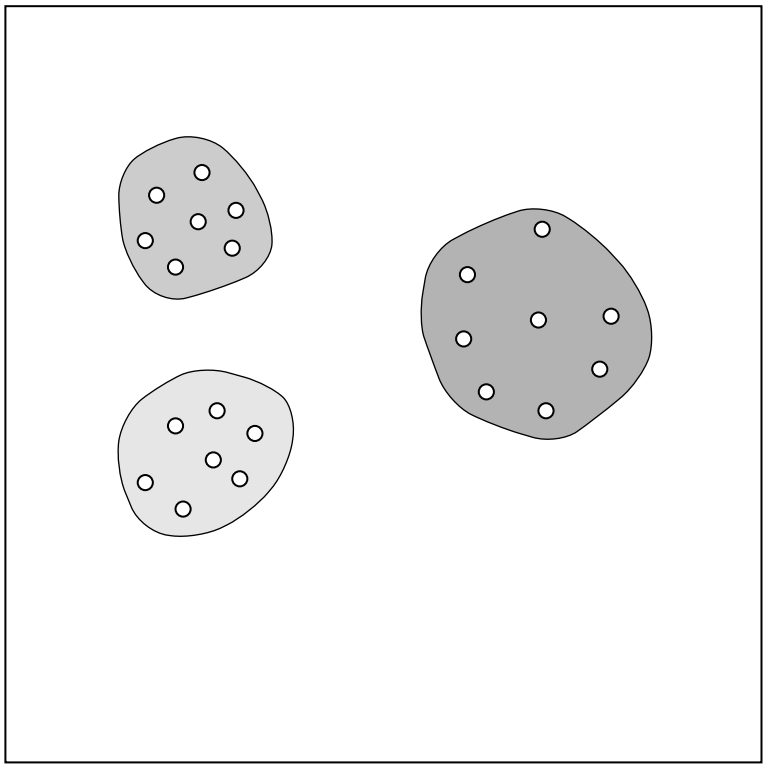
DBSCAN at structured data.



Similarity matrix sorted by cluster label.

Cluster Evaluation

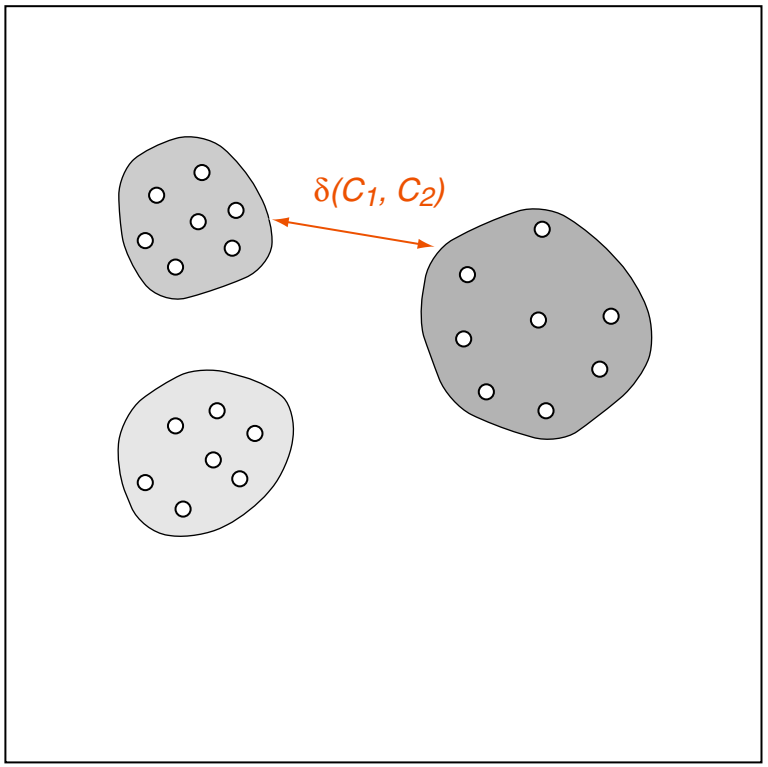
(2) Internal Validity Measures: Structural Analysis



- Distance for two clusters, $\delta(C_1, C_2)$.
- Diameter of a cluster, $\Delta(C)$.
- Scatter within a cluster, $\sigma^2(C)$, SSE.

Cluster Evaluation

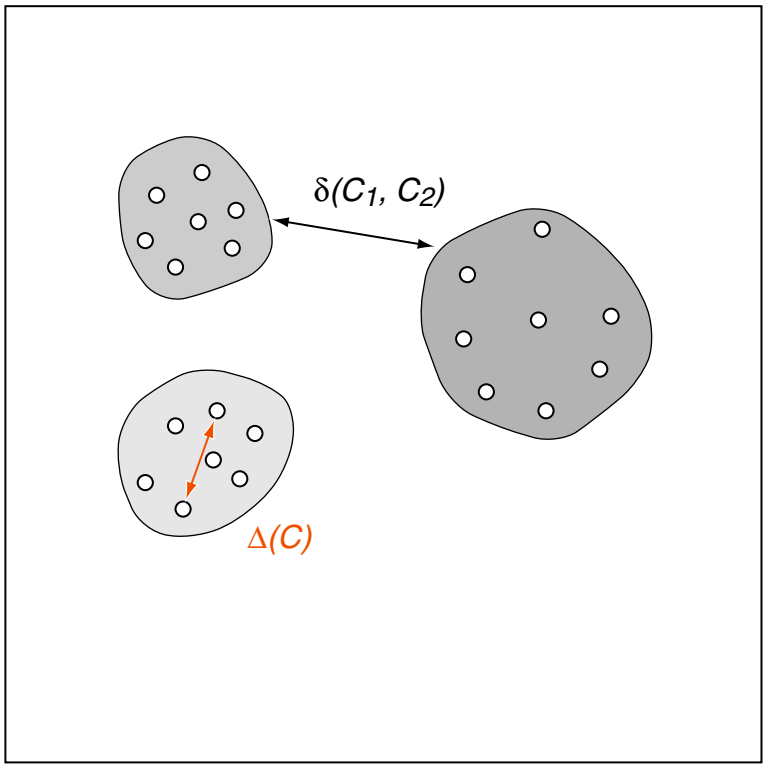
(2) Internal Validity Measures: Structural Analysis



- Distance for two clusters, $\delta(C_1, C_2)$.
- Diameter of a cluster, $\Delta(C)$.
- Scatter within a cluster, $\sigma^2(C)$, SSE.

Cluster Evaluation

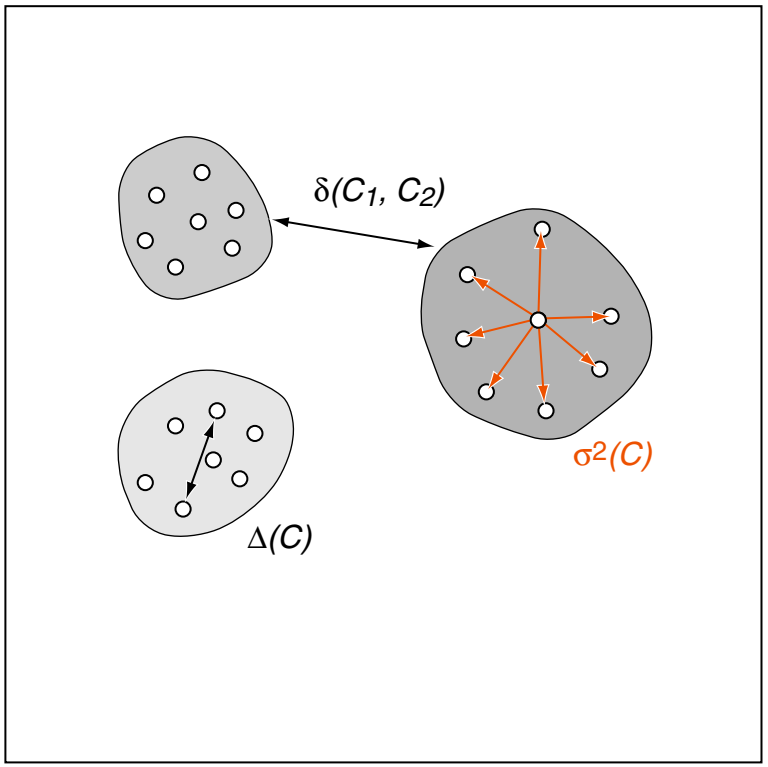
(2) Internal Validity Measures: Structural Analysis



- Distance for two clusters, $\delta(C_1, C_2)$.
- Diameter of a cluster, $\Delta(C)$.
- Scatter within a cluster, $\sigma^2(C)$, SSE.

Cluster Evaluation

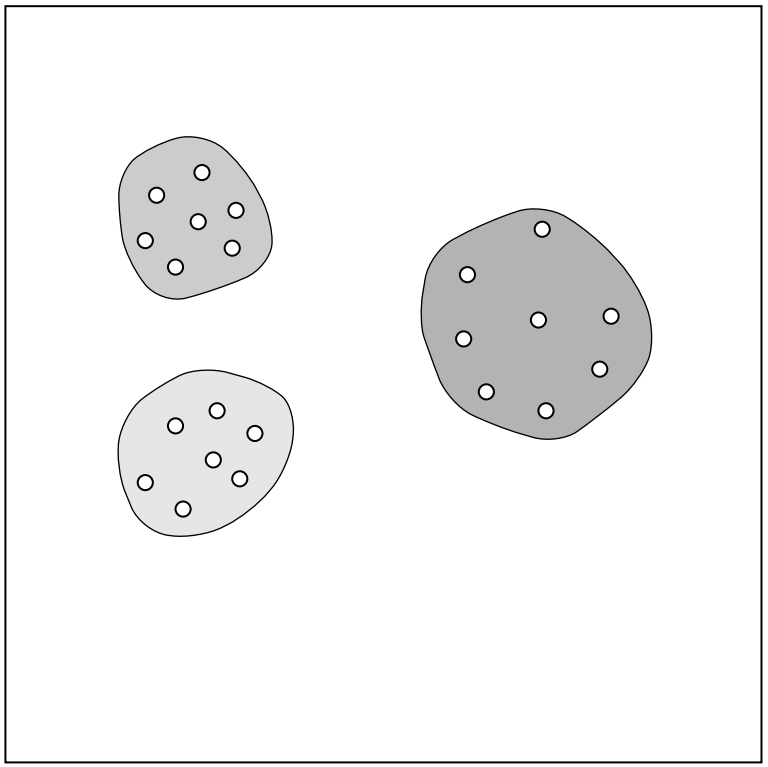
(2) Internal Validity Measures: Structural Analysis



- Distance for two clusters, $\delta(C_1, C_2)$.
- Diameter of a cluster, $\Delta(C)$.
- Scatter within a cluster, $\sigma^2(C)$, SSE.

Cluster Evaluation

(2) Internal Validity Measures: Dunn Index

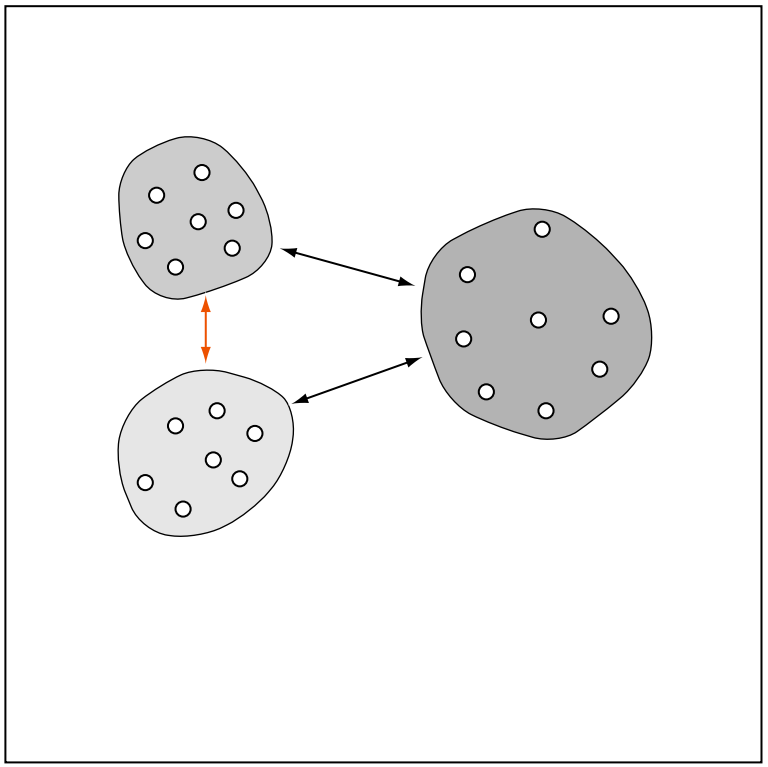


$$I(\mathcal{C}) = \frac{\min_{i \neq j}\{\delta(C_i, C_j)\}}{\max_{1 \leq l \leq k}\{\Delta(C_l)\}},$$

$$I(\mathcal{C}) \rightarrow \max$$

Cluster Evaluation

(2) Internal Validity Measures: Dunn Index

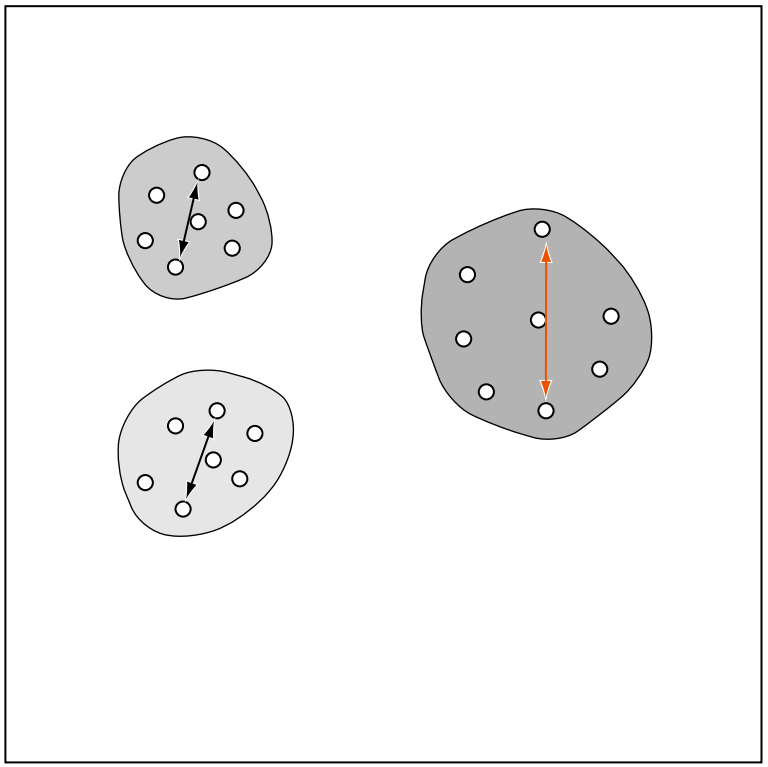


$$I(\mathcal{C}) = \frac{\min_{i \neq j}\{\delta(C_i, C_j)\}}{\max_{1 \leq l \leq k}\{\Delta(C_l)\}},$$

$$I(\mathcal{C}) \rightarrow \max$$

Cluster Evaluation

(2) Internal Validity Measures: Dunn Index

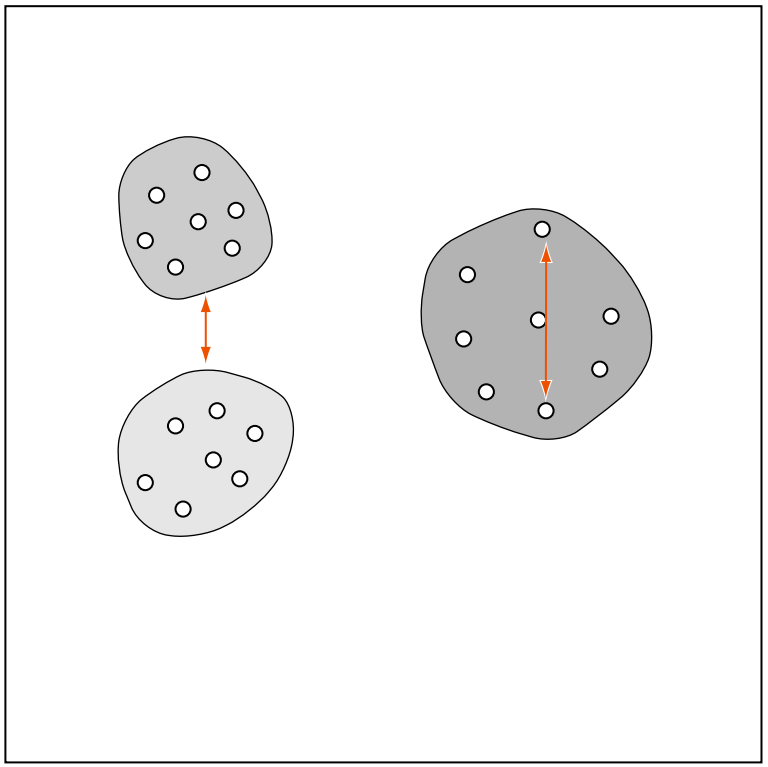


$$I(\mathcal{C}) = \frac{\min_{i \neq j}\{\delta(C_i, C_j)\}}{\max_{1 \leq l \leq k}\{\Delta(C_l)\}},$$

$$I(\mathcal{C}) \rightarrow \max$$

Cluster Evaluation

(2) Internal Validity Measures: Dunn Index



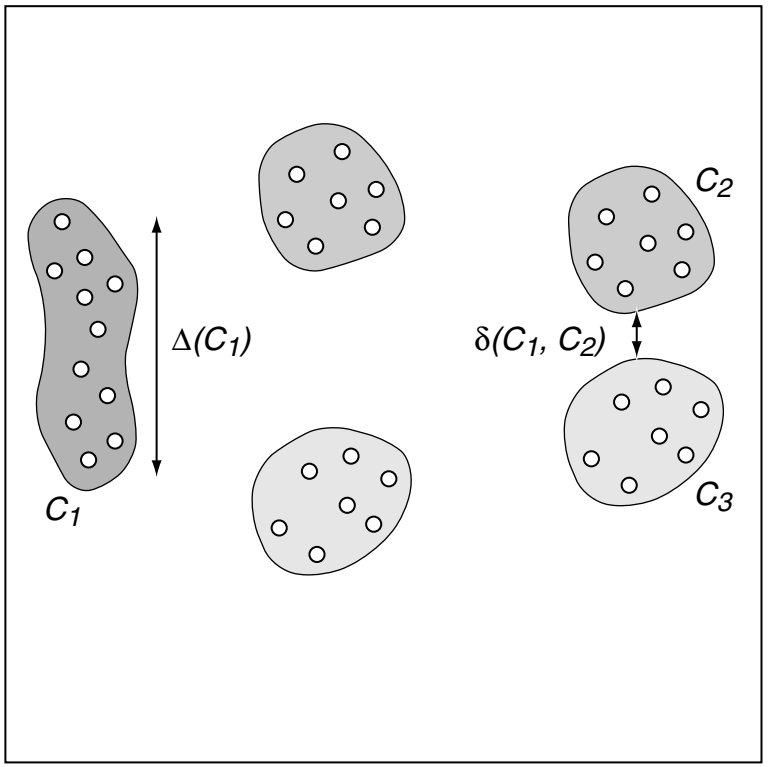
$$I(\mathcal{C}) = \frac{\min_{i \neq j}\{\delta(C_i, C_j)\}}{\max_{1 \leq l \leq k}\{\Delta(C_l)\}},$$

$$I(\mathcal{C}) \rightarrow \max$$

- Dunn is susceptible to noise.
- Dunn is biased towards the worst substructure in a clustering (cf. min)
- Dunn cannot put distances and diameters into relation.

Cluster Evaluation

(2) Internal Validity Measures: Dunn Index



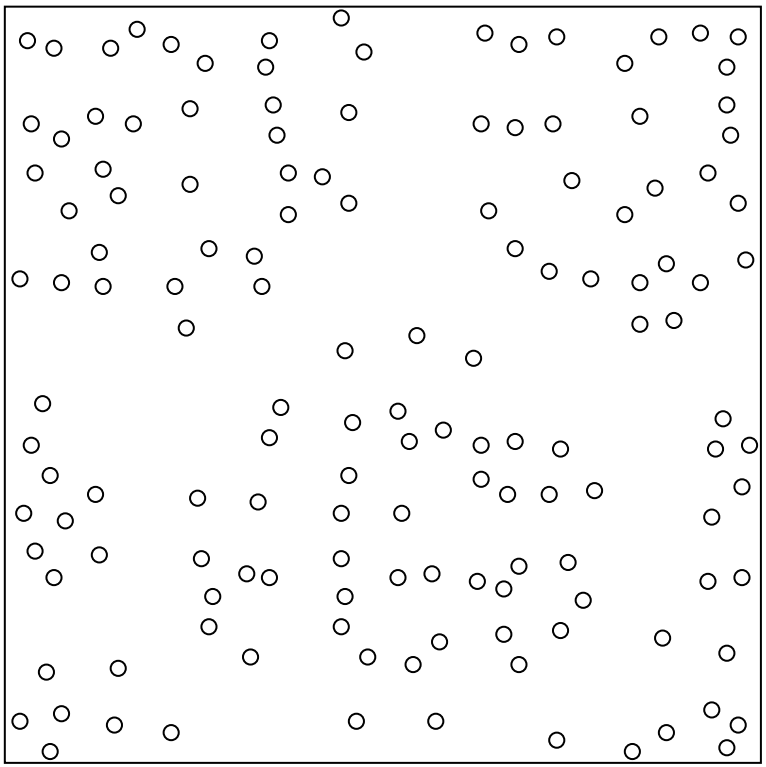
$$I(\mathcal{C}) = \frac{\min_{i \neq j}\{\delta(C_i, C_j)\}}{\max_{1 \leq l \leq k}\{\Delta(C_l)\}},$$

$$I(\mathcal{C}) \rightarrow \max$$

- Dunn is susceptible to noise.
- Dunn is biased towards the worst substructure in a clustering (cf. min)
- Dunn cannot put distances and diameters into relation.

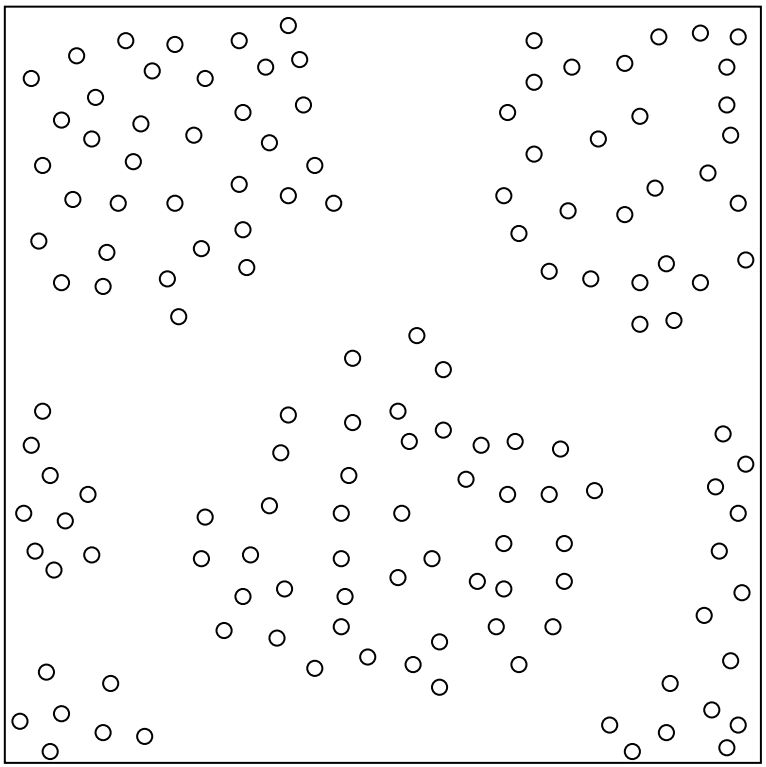
Cluster Evaluation

(2) Internal Validity Measures: Expected Density ρ [Stein/Meyer zu Eissen 2007]



Cluster Evaluation

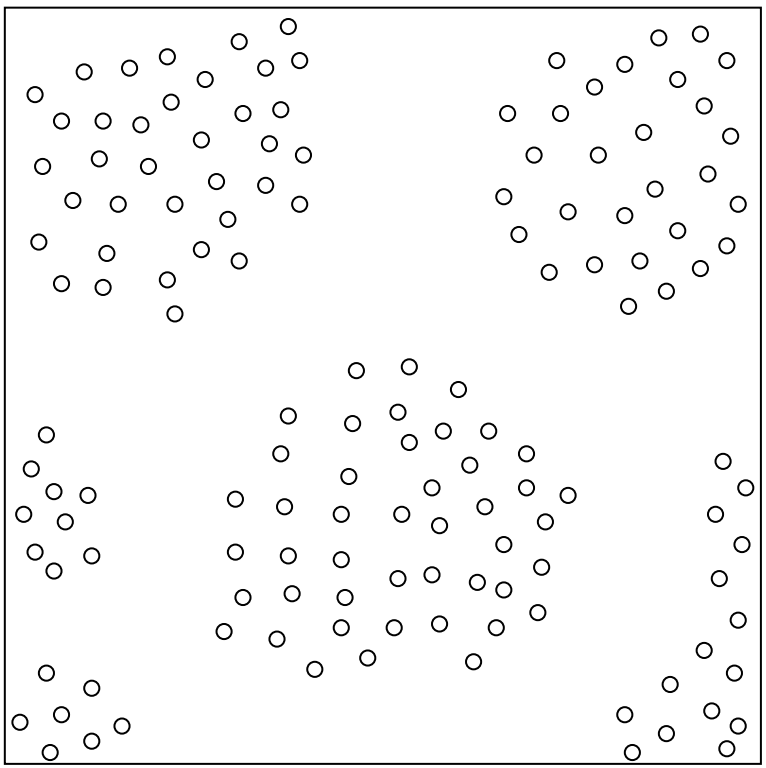
(2) Internal Validity Measures: Expected Density ρ [Stein/Meyer zu Eissen 2007]



Different retrieval models yield different similarity graphs.

Cluster Evaluation

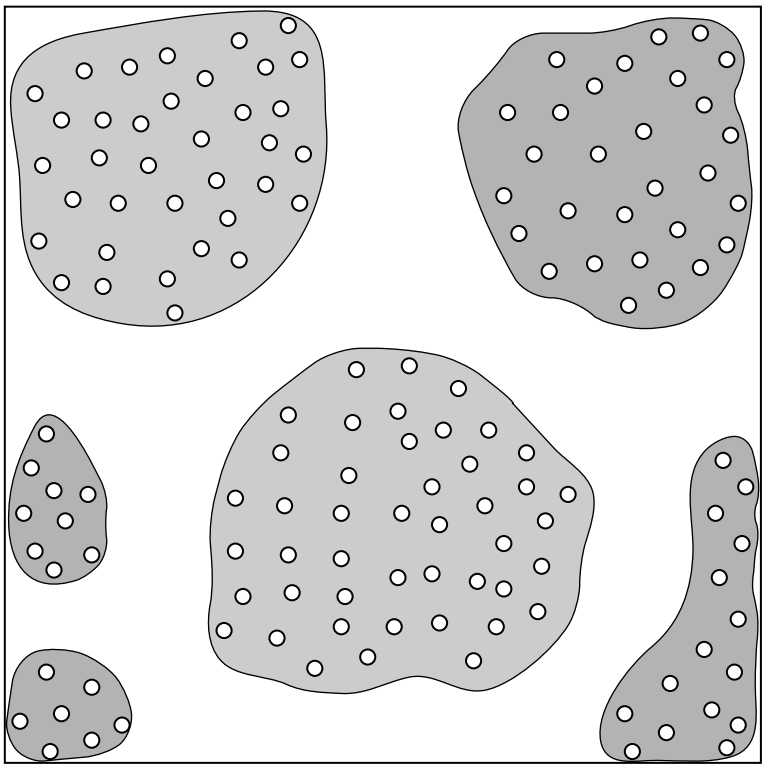
(2) Internal Validity Measures: Expected Density ρ [Stein/Meyer zu Eissen 2007]



Different retrieval models yield different similarity graphs.

Cluster Evaluation

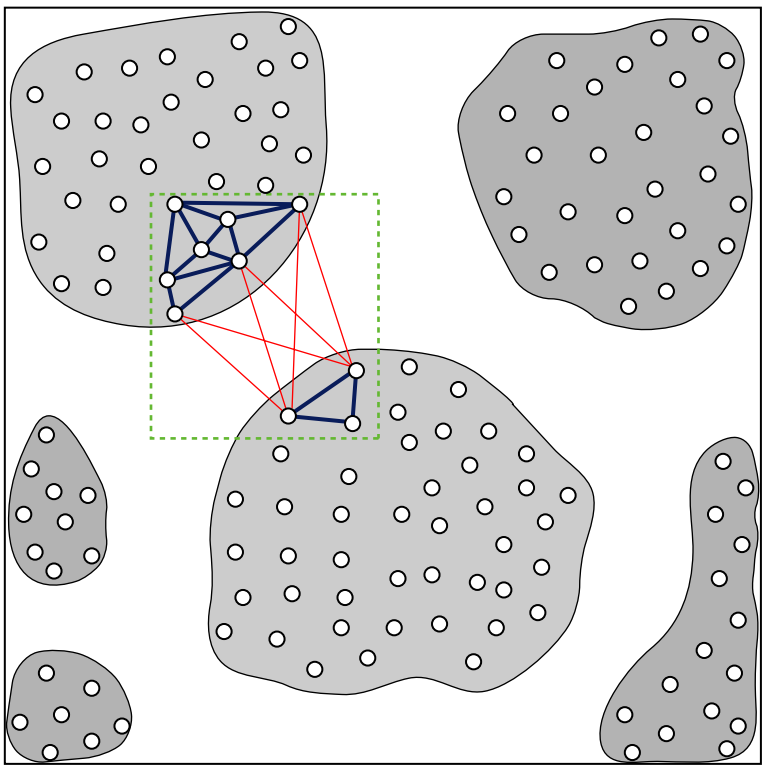
(2) Internal Validity Measures: Expected Density ρ



Compare (for alternative clusterings) the similarity density within the clusters to the average similarity of the entire graph.

Cluster Evaluation

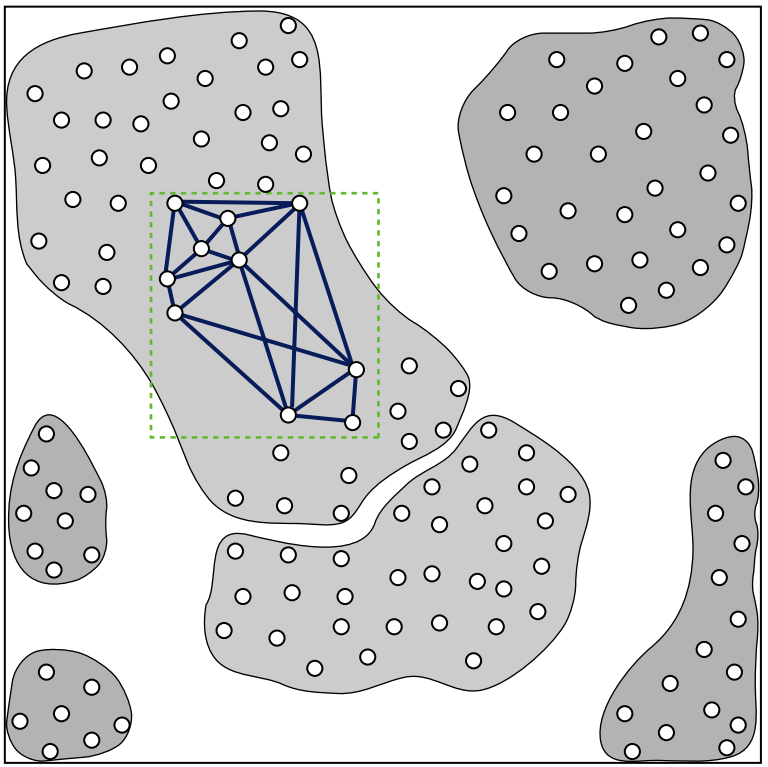
(2) Internal Validity Measures: Expected Density ρ



Compare (for alternative clusterings) the similarity density within the clusters to the average similarity of the entire graph.

Cluster Evaluation

(2) Internal Validity Measures: Expected Density ρ



Compare (for alternative clusterings) the similarity density within the clusters to the average similarity of the entire graph.

Cluster Evaluation

(2) Internal Validity Measures: Expected Density ρ

Graph $G = \langle V, E \rangle$

- G is called sparse [dense] if $|E| = O(|V|)$ $[O(|V|^2)]$
- the density θ computes from the equation $|E| = |V|^\theta$

Cluster Evaluation

(2) Internal Validity Measures: Expected Density ρ

Graph $G = \langle V, E \rangle$

- G is called sparse [dense] if $|E| = O(|V|)$ $[O(|V|^2)]$
- the density θ computes from the equation $|E| = |V|^\theta$

Similarity graph $G = \langle V, E, w \rangle$, $\quad |E| \sim w(G) := \sum_{e \in E} w(e)$

- the density θ computes from the equation $w(G) = |V|^\theta$

Cluster Evaluation

(2) Internal Validity Measures: Expected Density ρ

Graph $G = \langle V, E \rangle$

- G is called sparse [dense] if $|E| = O(|V|)$ $[O(|V|^2)]$
- the density θ computes from the equation $|E| = |V|^\theta$

Similarity graph $G = \langle V, E, w \rangle$, $\quad |E| \sim w(G) := \sum_{e \in E} w(e)$

- the density θ computes from the equation $w(G) = |V|^\theta$

Induced subgraph G_i for class C_i

- the expected density ρ compares class C_i to the density average in D

$$\rho(G_i) = \frac{w(G_i)}{|V_i|^\theta}$$

Cluster Evaluation

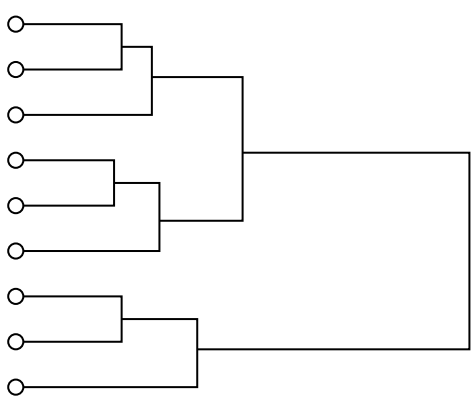
(3) Relative Validity Measures: Elbow Criterion

1. Hyperparameters of a clustering algorithm: $p_1, \ldots, p_m$
   - number of centroids for k-means
   - stopping level for hierarchical algorithms
   - neighborhood size for DBSCAN
2. Clusterings $\mathcal{C} = \{\mathcal{C}_{p_1}, \ldots, \mathcal{C}_{p_m}\}$ associated with $p_1, \ldots, p_m$.
3. Points of an error curve $\{(p_i, e(\mathcal{C}_{p_i})) \mid i = 1, \ldots, m\}$.

Cluster Evaluation

(3) Relative Validity Measures: Elbow Criterion

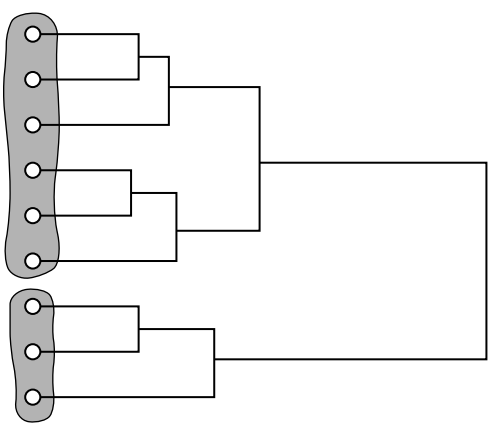
1. Hyperparameters of a clustering algorithm: $p_1, \ldots, p_m$
   - number of centroids for k-means
   - stopping level for hierarchical algorithms
   - neighborhood size for DBSCAN
2. Clusterings $\mathcal{C} = \{\mathcal{C}_{p_1}, \ldots, \mathcal{C}_{p_m}\}$ associated with $p_1, \ldots, p_m$.
3. Points of an error curve $\{(p_i, e(\mathcal{C}_{p_i})) \mid i = 1, \ldots, m\}$.

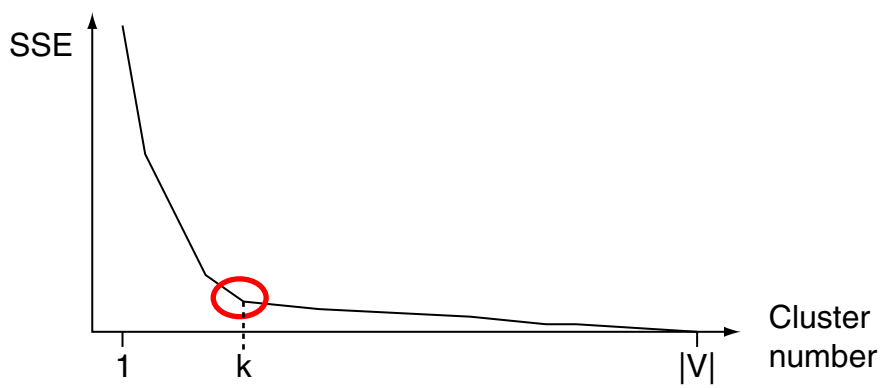


Cluster Evaluation

(3) Relative Validity Measures: Elbow Criterion

1. Hyperparameters of a clustering algorithm: $p_1, \ldots, p_m$
   - number of centroids for k-means
   - stopping level for hierarchical algorithms
   - neighborhood size for DBSCAN
2. Clusterings $\mathcal{C} = \{\mathcal{C}_{p_1}, \ldots, \mathcal{C}_{p_m}\}$ associated with $p_1, \ldots, p_m$.
3. Points of an error curve $\{(p_i, e(\mathcal{C}_{p_i})) \mid i = 1, \ldots, m\}$.

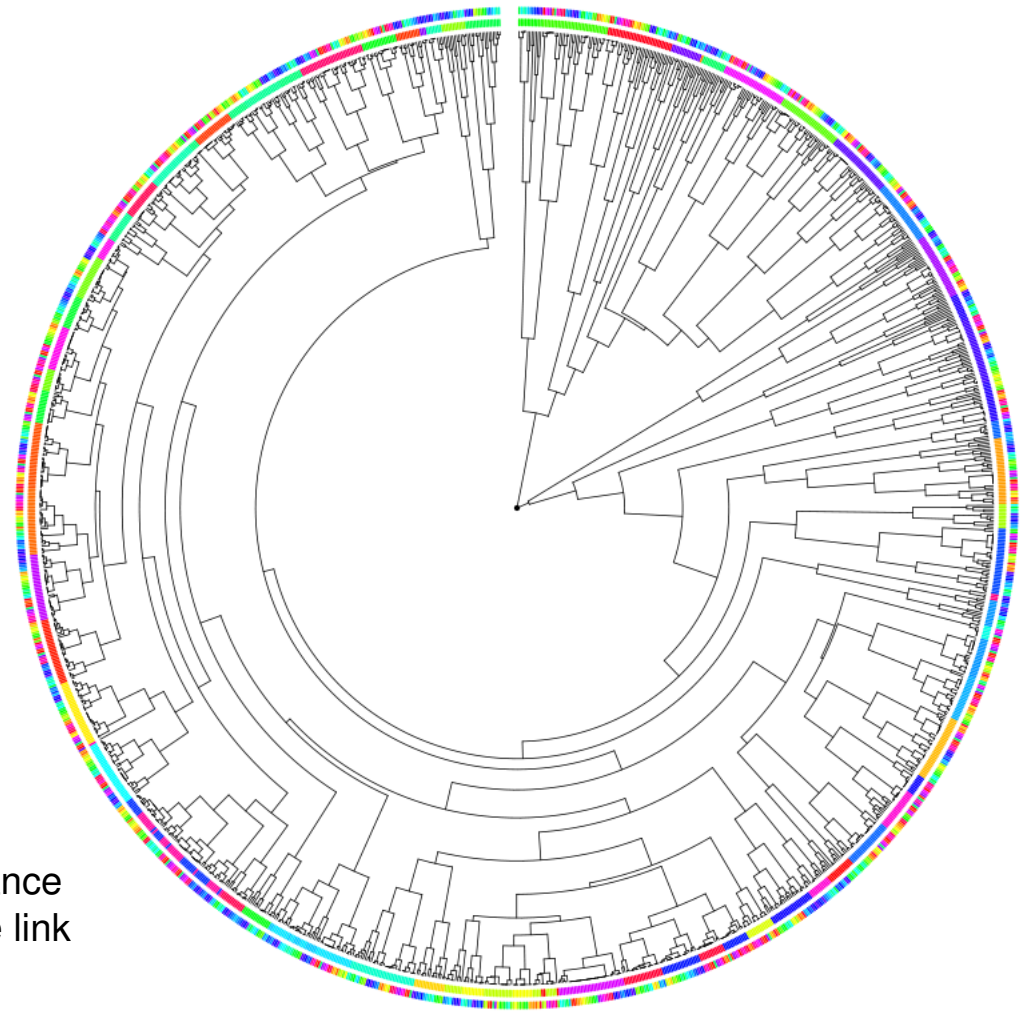




4. Find point that maximizes error drop with respect to its predecessor.

Cluster Evaluation

(3) Relative Validity Measures: Elbow Criterion



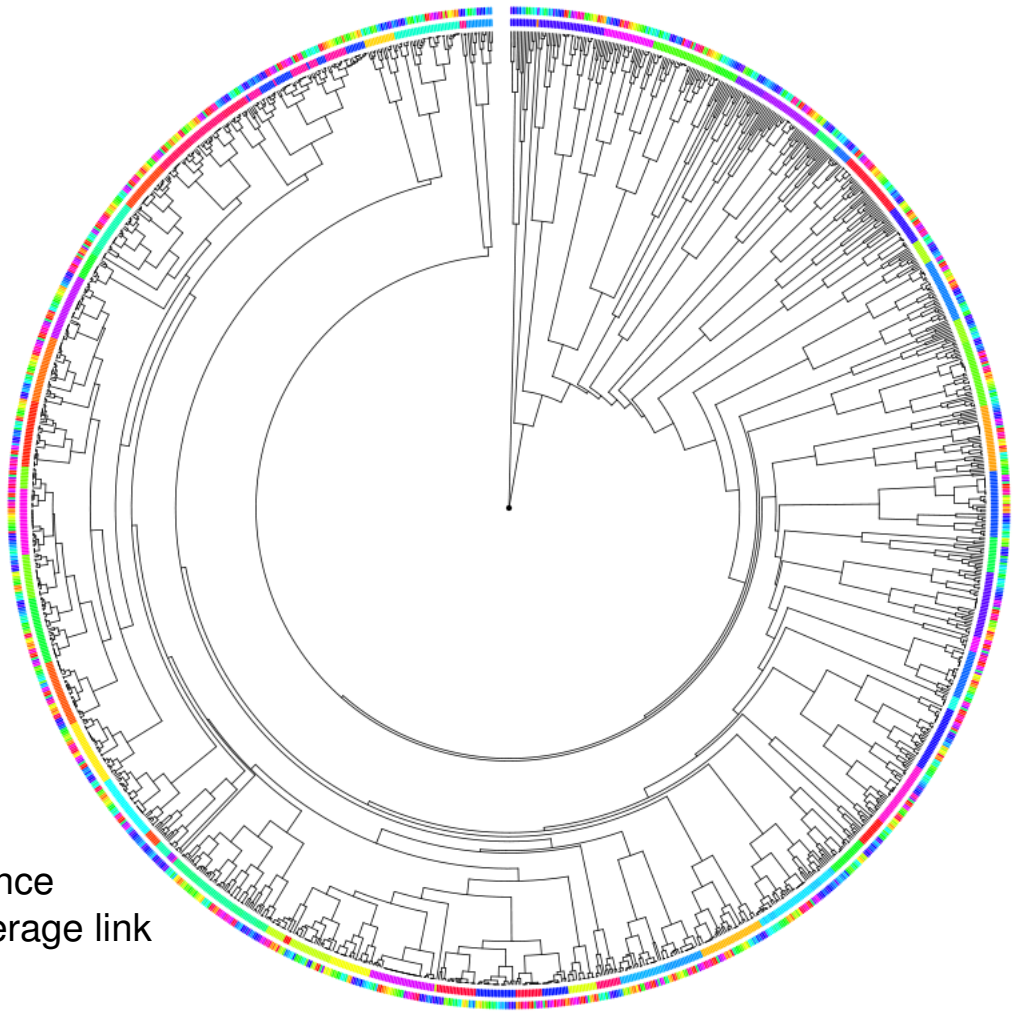
$d_{\mathcal{C}}$: Hamming distance
Merging: complete link

http://cs.jhu.edu/~razvanm/fs-expedition/2.6.x.html

Relations between 1377 file systems for Linux Kernel 2.6.0. [Razvan Musaloiu 2009]

Cluster Evaluation

(3) Relative Validity Measures: Elbow Criterion



$d_{\mathcal{C}}$: Hamming distance
Merging: group average link

http://cs.jhu.edu/~razvanm/fs-expedition/2.6.x.html

Relations between 1377 file systems for Linux Kernel 2.6.0. [Razvan Musaloiu 2009]

Cluster Evaluation

Correlation between External and Internal Measures

In the wild, we are not given a reference classification.

- ➜ An external evaluation is not possible.
  (though many papers report on such experiments)
- ➜ Resort to an internal evaluation.
  (connectivity, squared error sums, distance-diameter heuristics, etc.)

Cluster Evaluation

Correlation between External and Internal Measures

In the wild, we are not given a reference classification.

- ➜ An external evaluation is not possible.
  (though many papers report on such experiments)
- ➜ Resort to an internal evaluation.
  (connectivity, squared error sums, distance-diameter heuristics, etc.)

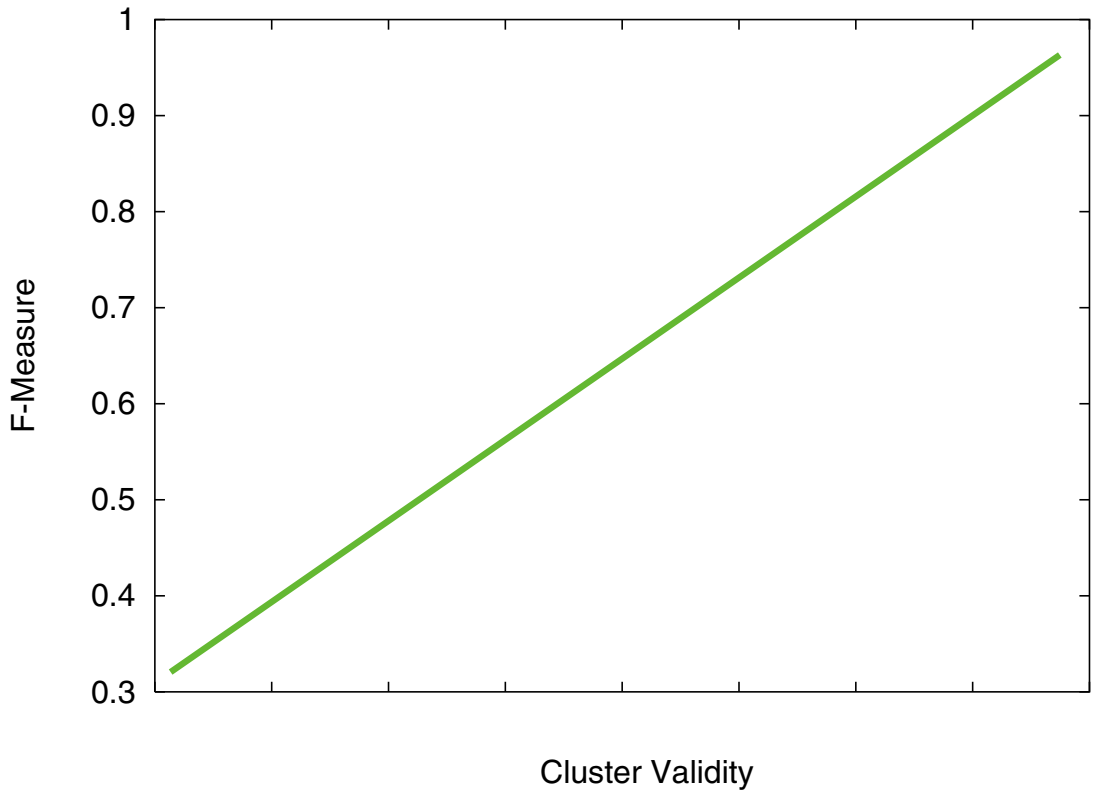
"To which extent can an internal evaluation ϕ be used to predict for a clustering its distance from the best reference classification—say, to predict the F-measure?"

$$\operatorname*{argmax}_{\phi} \{\tau\langle X, Y\rangle \mid x = F(\mathcal{C}),\ y = \phi(\mathcal{C}),\ \mathcal{C} \in \boldsymbol{\mathcal{C}}\}$$

[Stein/Meyer zu Eissen 2007]

Cluster Evaluation

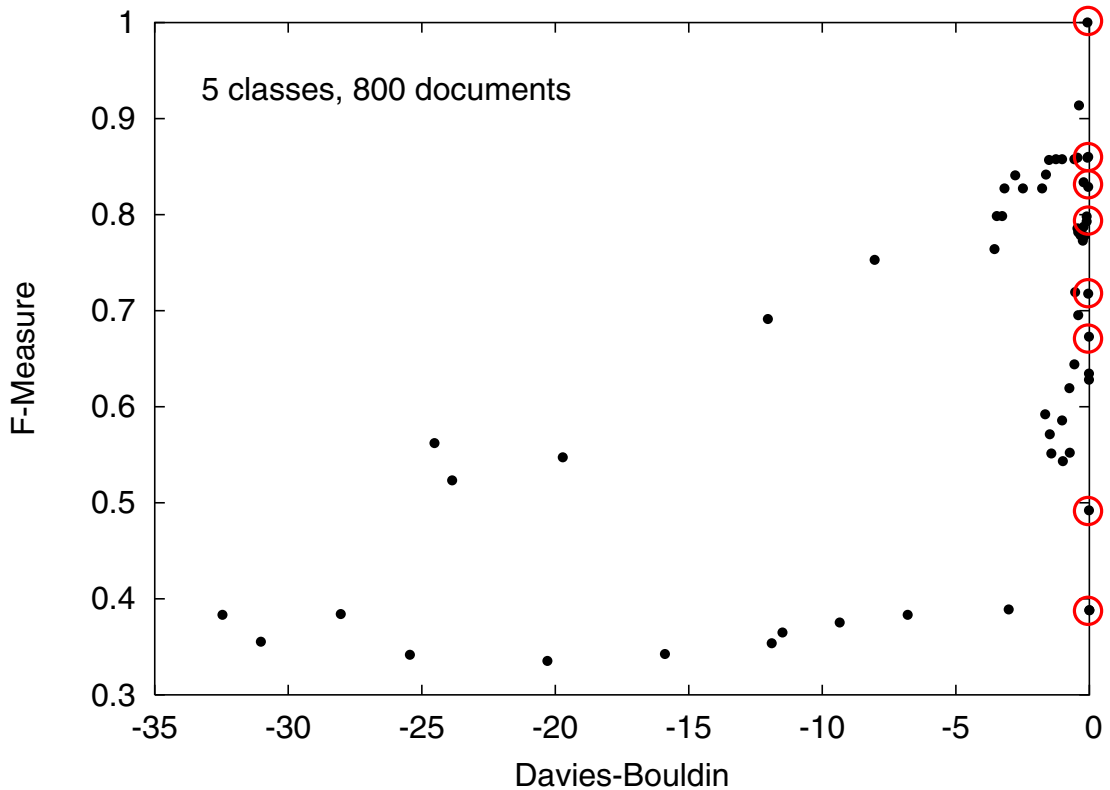
Correlation between External and Internal Measures



Perfect correlation (desired).

Cluster Evaluation

Correlation between External and Internal Measures

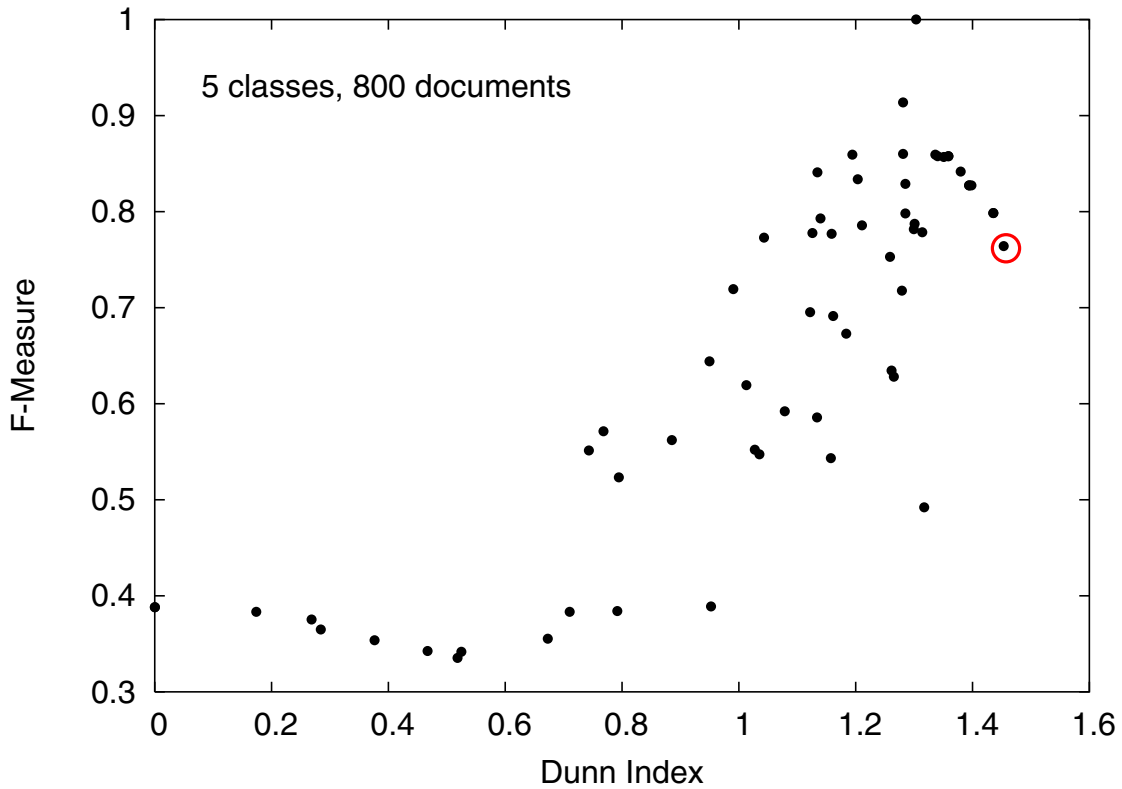


Davies-Bouldin: $$\frac{1}{k} \cdot \sum_{i=1}^{k} \max_{j} \frac{s(C_i) + s(C_j)}{\delta(C_i, C_j)}$$

Prefers spherical clusters.

Cluster Evaluation

Correlation between External and Internal Measures

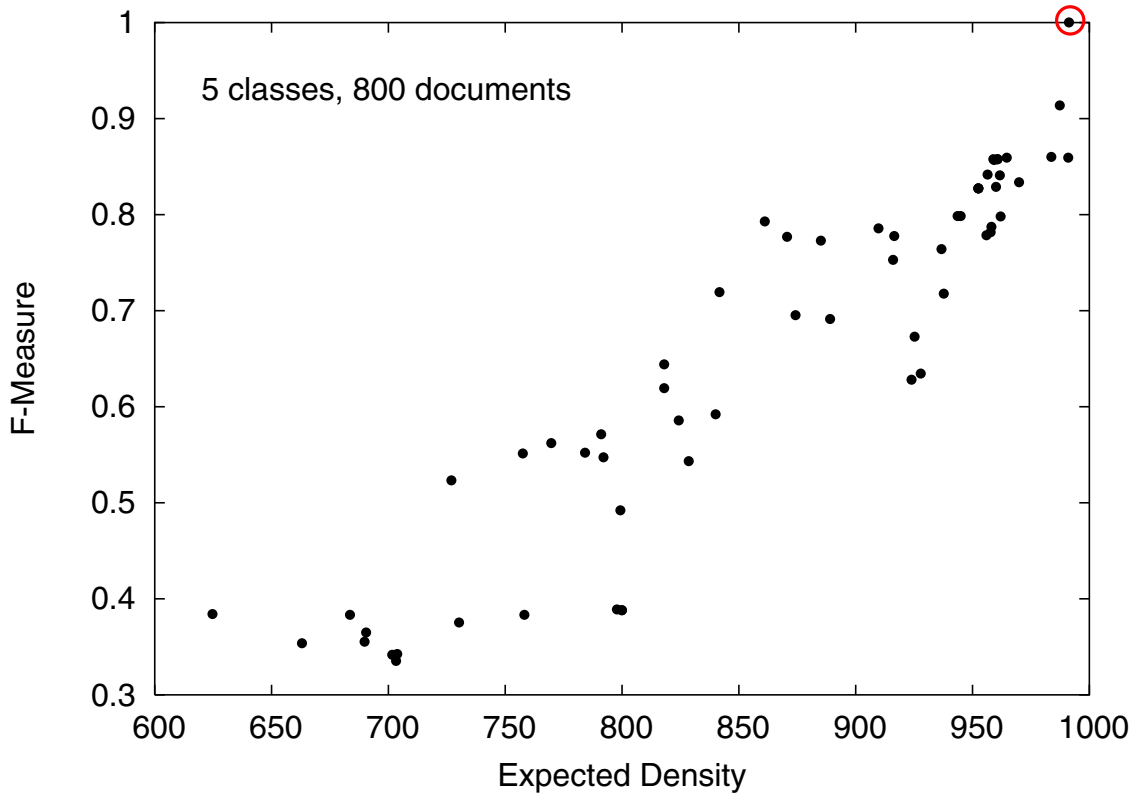


Dunn Index: $\frac{\min_{i \neq j}\{\delta(C_i, C_j)\}}{\max_{1 \leq l \leq k}\{\Delta(C_l)\}}$

Maximizes dilatation = inter/intra-cluster-diameter.

Cluster Evaluation

Correlation between External and Internal Measures



Expected Density: $\bar{\rho} = \sum_{i=1}^{k} \frac{|V_i|}{|V|} \cdot \frac{w(G_i)}{|V_i|^{\theta}}$

Independent of cluster forms and sizes.

Chapter ML:XI (continued)

XI. Cluster Analysis

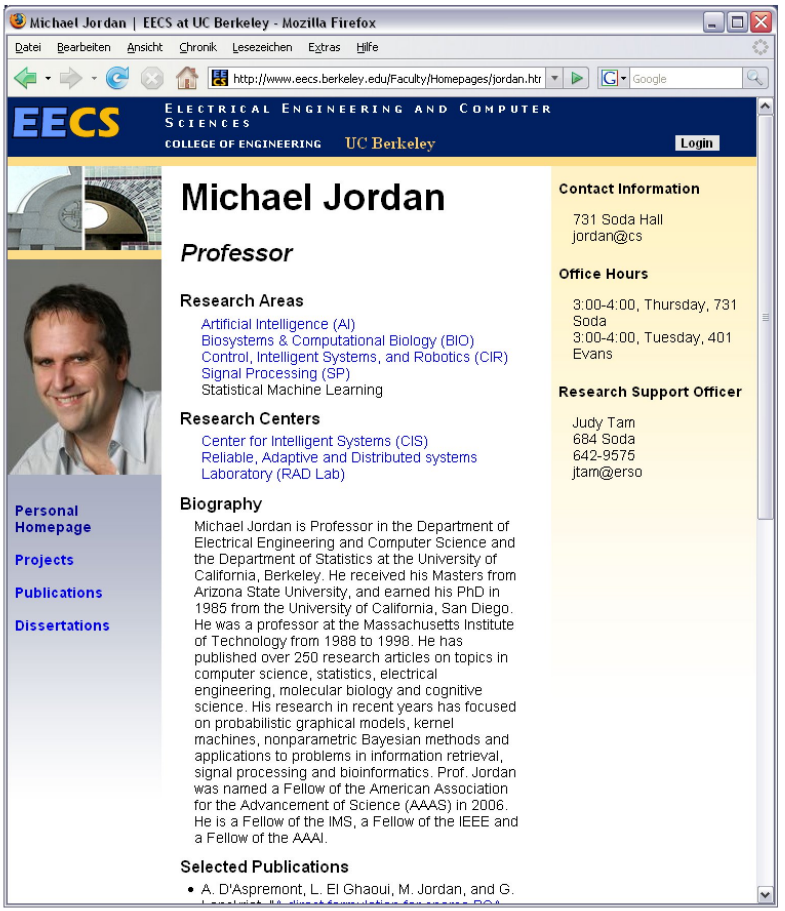
- Data Mining Overview
- Cluster Analysis Basics
- Hierarchical Cluster Analysis
- Iterative Cluster Analysis
- Density-Based Cluster Analysis
- Cluster Evaluation
- Constrained Cluster Analysis

Constrained Cluster Analysis

Person Resolution Task

Constrained Cluster Analysis

Person Resolution Task



Constrained Cluster Analysis

Person Resolution Task

Constrained Cluster Analysis

Person Resolution Task

The basket ball player.

...

The statistician.

Constrained Cluster Analysis

Person Resolution Task

The basket ball player.

...

The statistician.

- **Multi-document resolution task:**

Names, Target names:	$N = \{n_1, \ldots, n_l\}$,	$T \subset N$
Referents:	$R = \{r_1, \ldots, r_m\}$,	$\tau : R \to T$, $\lvert R \rvert \gg \lvert T \rvert$
Documents:	$D = \{d_1, \ldots, d_n\}$,	$\nu : D \to \mathcal{P}(N)$, $\lvert \nu(d_i) \cap T \rvert = 1$
A solution:	$\gamma : D \to R$,	s.t. $\tau(\gamma(d_i)) \in \nu(d_i)$

Constrained Cluster Analysis

Person Resolution Task

The basket ball player.

...

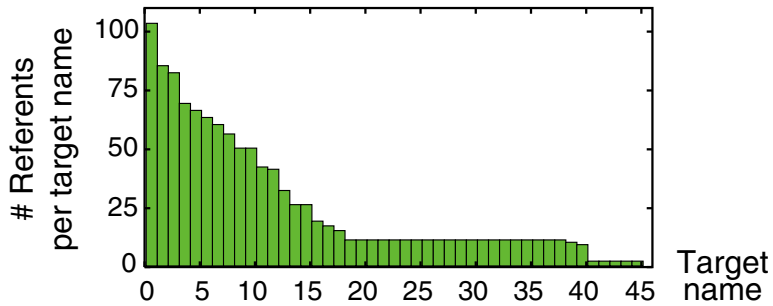
The statistician.

- **Facts about the Spock data mining challenge:**

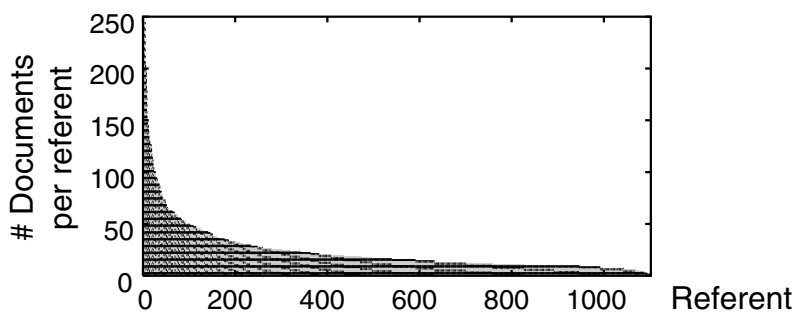
Target names:	$\lvert T \rvert = 44$
Referents:	$\lvert R \rvert = 1\,101$
Documents:	$\lvert D_{train} \rvert = 27\,000$ (labeled $\approx$ 2.3GB)
	$\lvert D_{test} \rvert = 75\,000$ (unlabeled $\approx$ 7.8GB)

Constrained Cluster Analysis

Person Resolution Task



- up to 105 referents for a single target name
- about 25 referents on average per target name



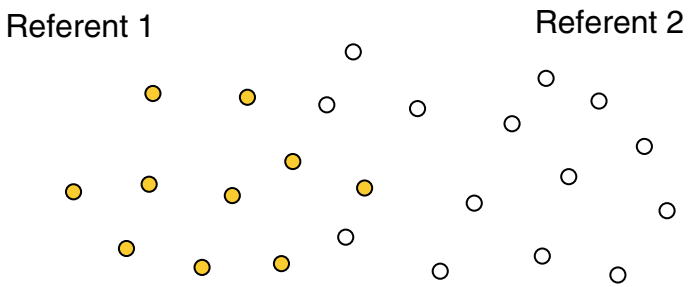
- about 23 documents on average per referent

- **Facts about the Spock data mining challenge:**

Target names:	$\lvert T \rvert = 44$
Referents:	$\lvert R \rvert = 1\,101$
Documents:	$\lvert D_{train} \rvert = 27\,000$ (labeled $\approx$ 2.3GB)
	$\lvert D_{test} \rvert = 75\,000$ (unlabeled $\approx$ 7.8GB)

Constrained Cluster Analysis

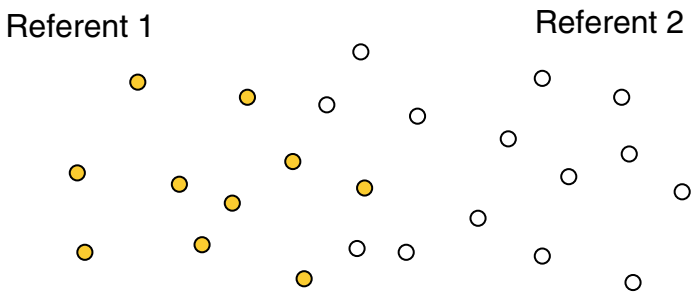
Applied to Multi-Document Resolution



1. Model similarities ➜ new and established retrieval models:
   - global and context-based vector space models
   - explicit semantic analysis
   - ontology alignment
2. Learn class memberships (supervised) ➜ logistic regression
3. Find equivalence classes (unsupervised) ➜ cluster analysis:
   (a) adaptive graph thinning
   (b) multiple, density-based cluster analysis
   (c) clustering selection by expected density maximization

Constrained Cluster Analysis

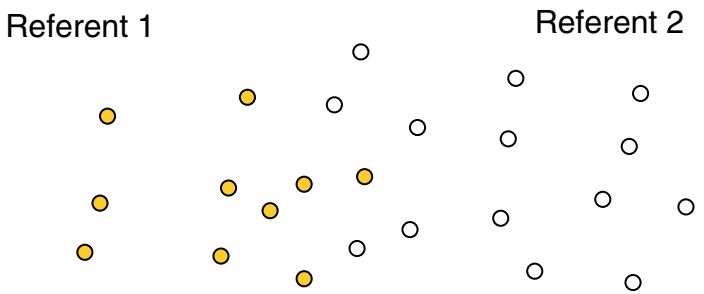
Applied to Multi-Document Resolution



1. Model similarities ➜ new and established retrieval models:
   - ❑ global and context-based vector space models
   - ❑ explicit semantic analysis
   - ❑ ontology alignment
2. Learn class memberships (supervised) ➜ logistic regression
3. Find equivalence classes (unsupervised) ➜ cluster analysis:
   (a) adaptive graph thinning
   (b) multiple, density-based cluster analysis
   (c) clustering selection by expected density maximization

Constrained Cluster Analysis

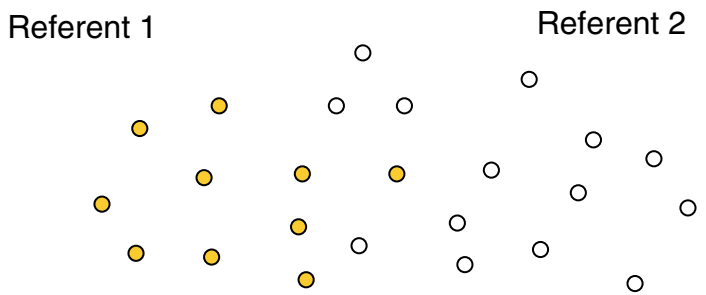
Applied to Multi-Document Resolution



1. Model similarities ➜ new and established retrieval models:
   - ❑ global and context-based vector space models
   - ❑ explicit semantic analysis
   - ❑ ontology alignment
2. Learn class memberships (supervised) ➜ logistic regression
3. Find equivalence classes (unsupervised) ➜ cluster analysis:
   (a) adaptive graph thinning
   (b) multiple, density-based cluster analysis
   (c) clustering selection by expected density maximization

Constrained Cluster Analysis

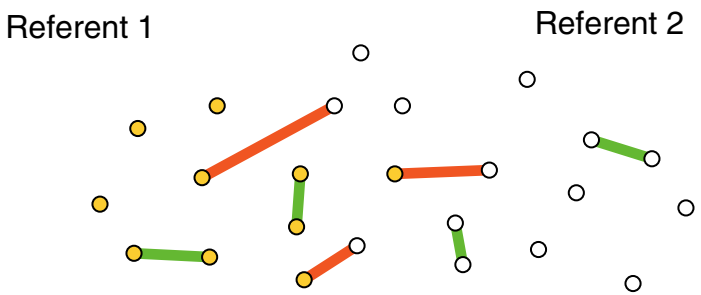
Applied to Multi-Document Resolution



1. Model similarities ➜ new and established retrieval models:
   - ❑ global and context-based vector space models
   - ❑ explicit semantic analysis
   - ❑ ontology alignment
2. Learn class memberships (supervised) ➜ logistic regression
3. Find equivalence classes (unsupervised) ➜ cluster analysis:
   - (a) adaptive graph thinning
   - (b) multiple, density-based cluster analysis
   - (c) clustering selection by expected density maximization

Constrained Cluster Analysis

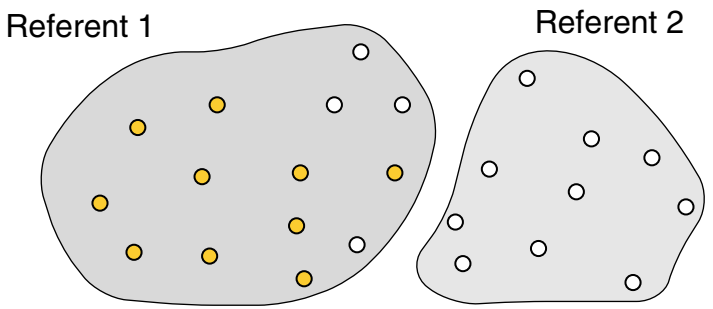
Applied to Multi-Document Resolution



1. Model similarities ➜ new and established retrieval models:
   - ❑ global and context-based vector space models
   - ❑ explicit semantic analysis
   - ❑ ontology alignment
2. Learn class memberships (supervised) ➜ logistic regression
3. Find equivalence classes (unsupervised) ➜ cluster analysis:
   - (a) adaptive graph thinning
   - (b) multiple, density-based cluster analysis
   - (c) clustering selection by expected density maximization

Constrained Cluster Analysis

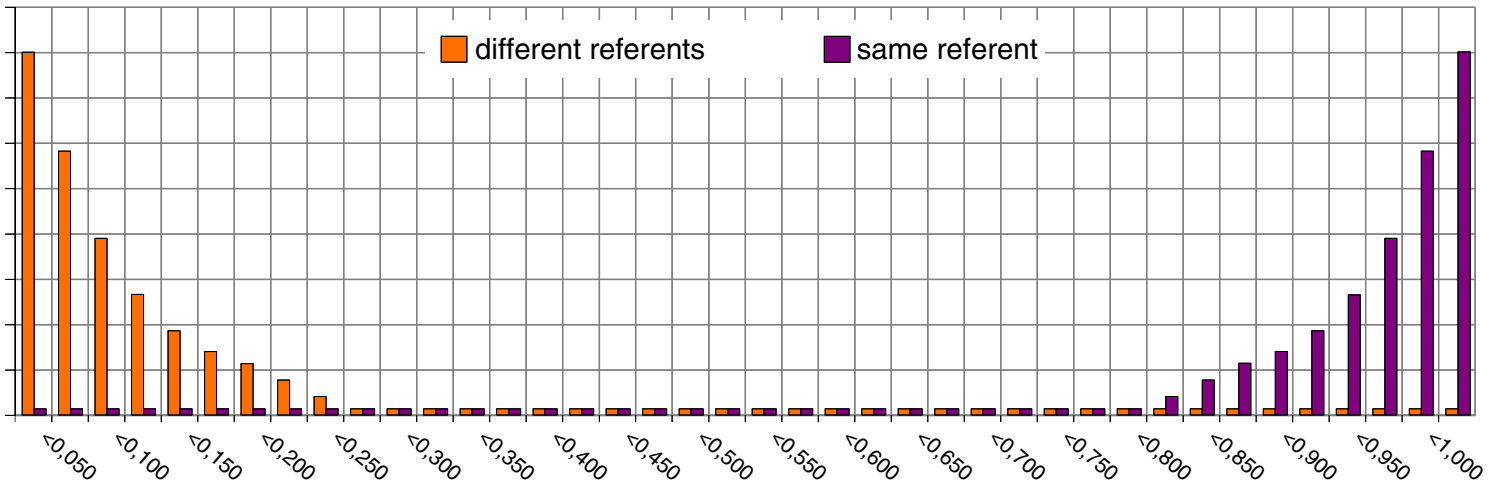
Applied to Multi-Document Resolution



1. Model similarities ➜ new and established retrieval models:
   - global and context-based vector space models
   - explicit semantic analysis
   - ontology alignment
2. Learn class memberships (supervised) ➜ logistic regression
3. Find equivalence classes (unsupervised) ➜ cluster analysis:
   (a) adaptive graph thinning
   (b) multiple, density-based cluster analysis
   (c) clustering selection by expected density maximization

Constrained Cluster Analysis

Idealized Class Membership Distribution over Similarities



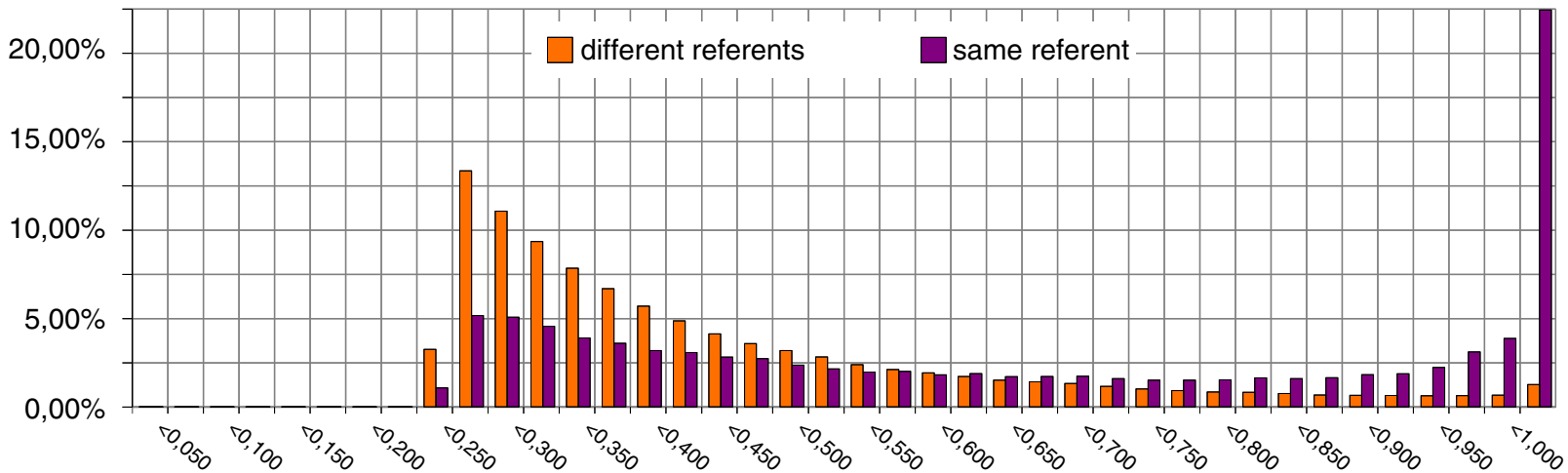
Similarity distributions for document pairs from different referents and same referent.

Logistic regression task:

- sample size: 400 000
- classes imbalance: non-target class : target class ≈ 25:1
- items are drawn uniformly distributed wrt. non-targets and targets
- items are uniformly distributed over the groups of target names

Constrained Cluster Analysis

Membership Distribution under *tf·idf* Vector Space Model

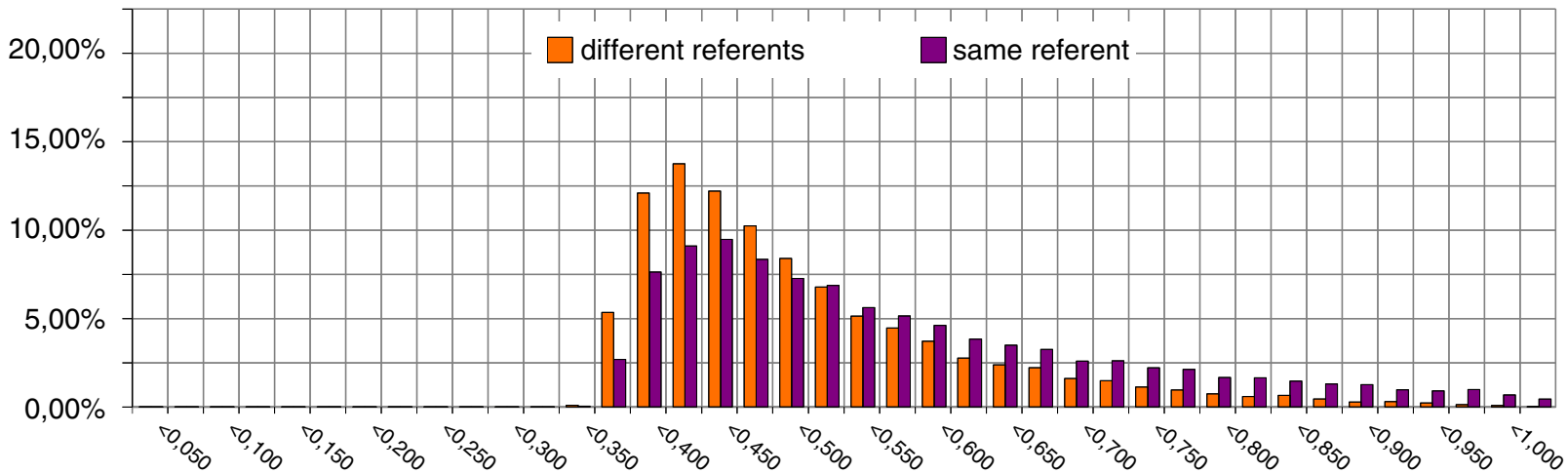


Model details:

- corpus size: 25 000 documents
- dictionary size: 1,2 Mio terms
- stopwords number: 850
- stopword volume: 36%

Constrained Cluster Analysis

Membership Distribution under Context-Based Vector Space Model

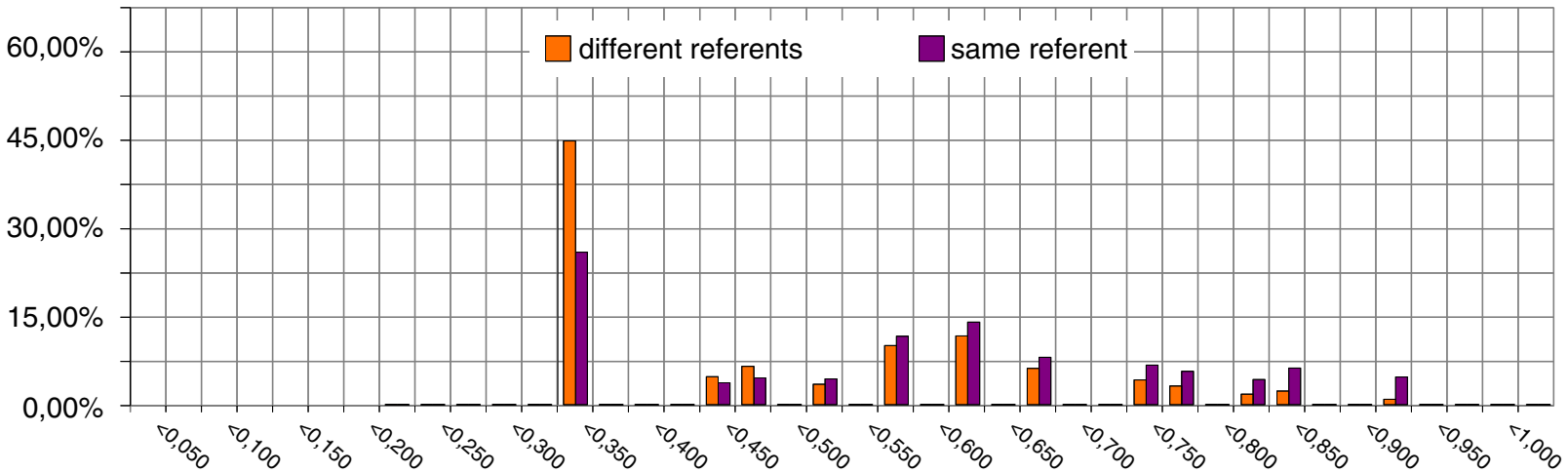


Model details:

- corpus size: 25 000 documents
- dictionary size: 1,2 Mio terms
- stopwords number: 850
- stopword volume: 36%

Constrained Cluster Analysis

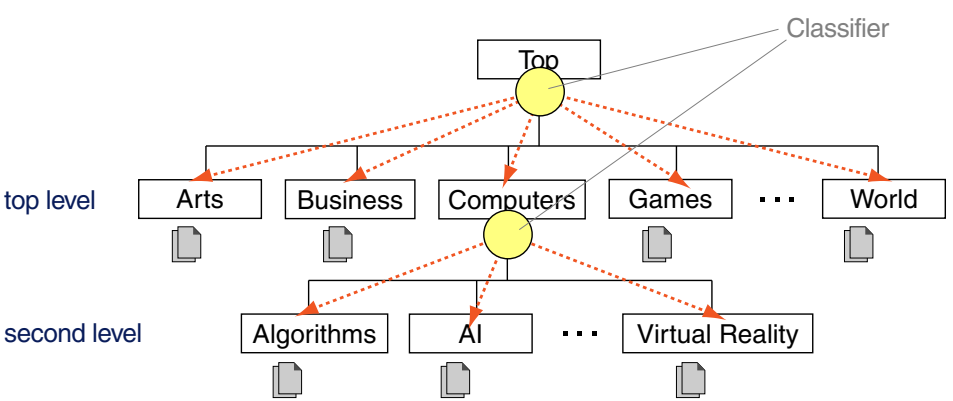
Membership Distribution under Ontology Alignment Model



Model details:

- DMOZ open directory project
- > 5 million documents

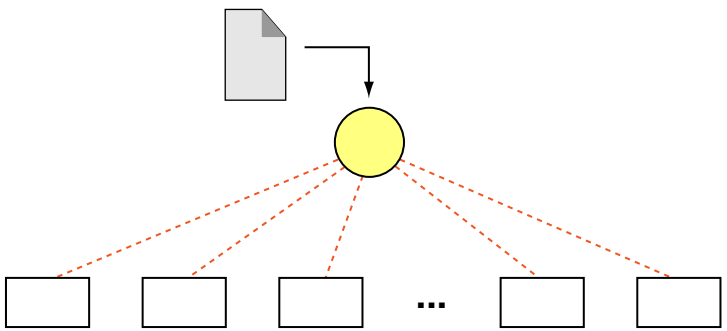
- 12 top-level categories
- 31 second level categories
- ML: hierarchical Bayes
- training set: 100 000 pages



Constrained Cluster Analysis

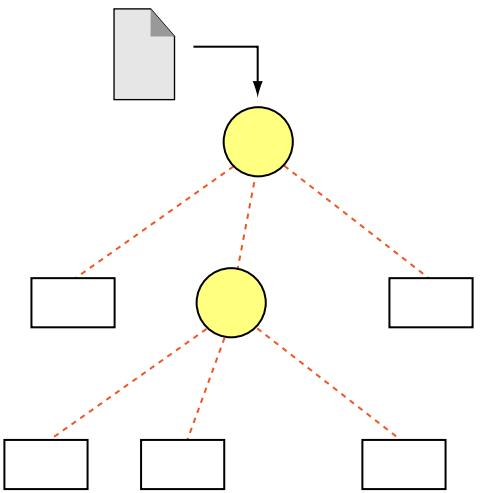
In-Depth: Multi-Class Hierarchical Classification

Flat (big-bang) classification



+ simple realization
– loss of discriminative power with increasing number of categories

Hierarchical (top-down) classification



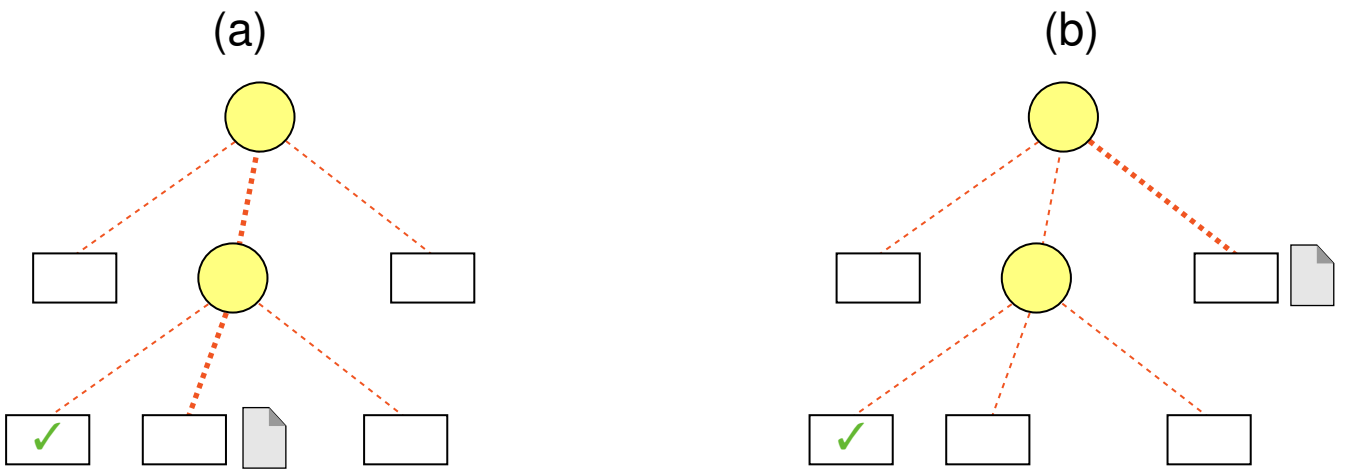
+ specialized classifiers (divide and conquer)
– misclassification at higher levels can never become repaired

Constrained Cluster Analysis

In-Depth: Multi-Class Hierarchical Classification

State of the art of effectiveness analyses:

1. independence assumption between categories
2. neglection of both hierarchical structure and degree of misclassification



Improvements:

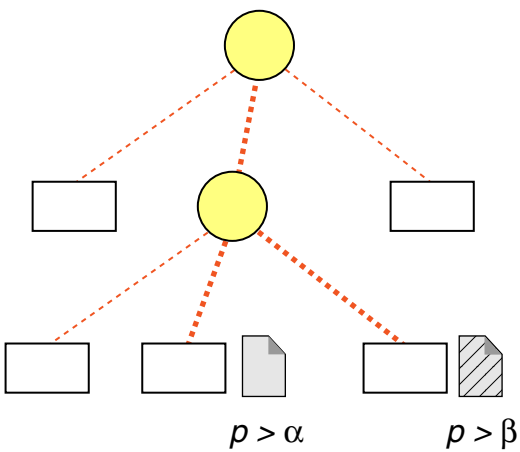
- Consider similarity $\varphi(C_i, C_j)$ between correct and wrong category.
- Consider graph distance $d(C_i, C_j)$ between correct and wrong category.

Constrained Cluster Analysis

In-Depth: Multi-Class Hierarchical Classification

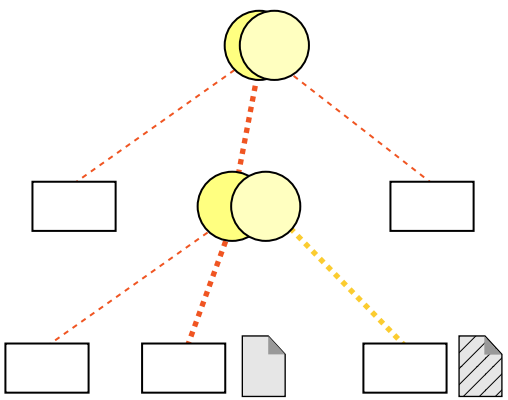
Improvements continued:

Multi-label (multi path) classification



- traverse more than one path and return all labels
- employ probabilistic classifiers with a threshold: split a path or not

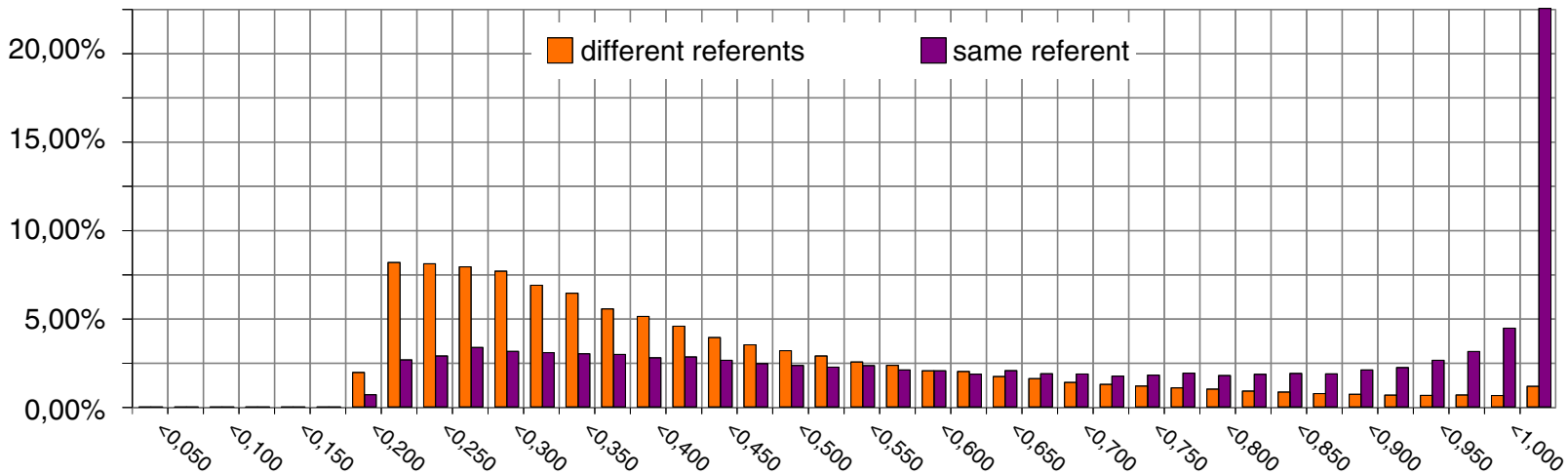
Multi-classifier (ensemble) classification



- classification result is a majority decision
- employ different classifier (different types or differently parameterized)

Constrained Cluster Analysis

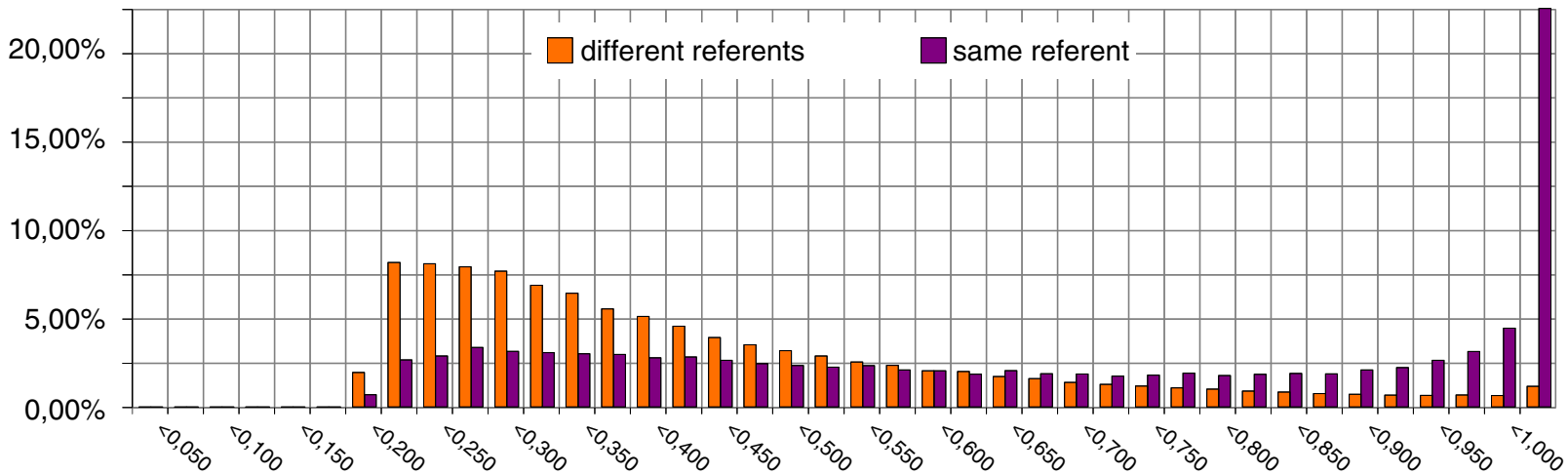
Membership Distribution under Optimized Retrieval Model Combination



Retrieval Model	$F_{1/3}$ -Measure
tf-idf vector space	0.39
context-based vector space	0.32
ESA Wikipedia persons	0.30
phrase structure grammar	0.17
ontology alignment	0.15
optimized combination	**0.42**

Constrained Cluster Analysis

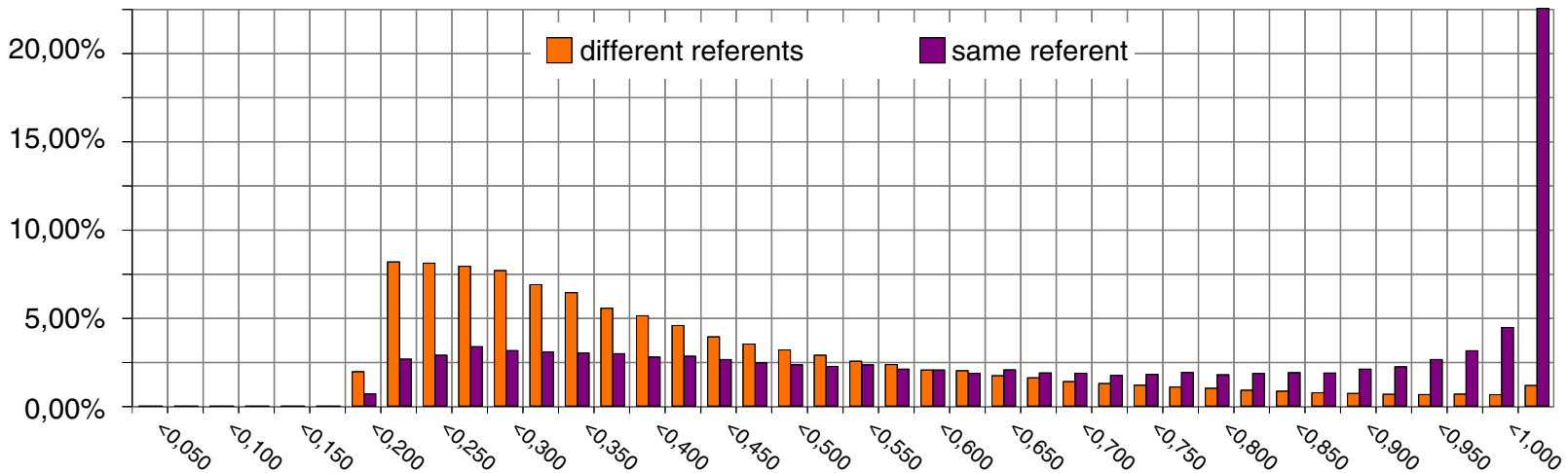
Membership Distribution under Optimized Retrieval Model Combination



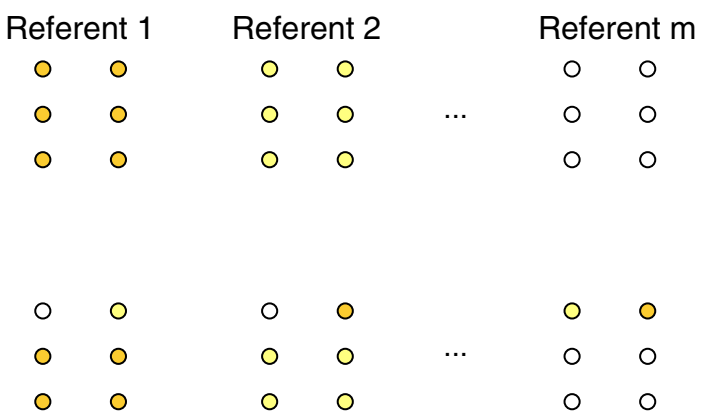
Retrieval Model	$F_{1/3}$ -Measure
tf-idf vector space	0.39
context-based vector space	0.32
ESA Wikipedia persons	0.30
phrase structure grammar	0.17
ontology alignment	0.15
optimized combination	**0.42**

Constrained Cluster Analysis

Membership Distribution under Optimized Retrieval Model Combination

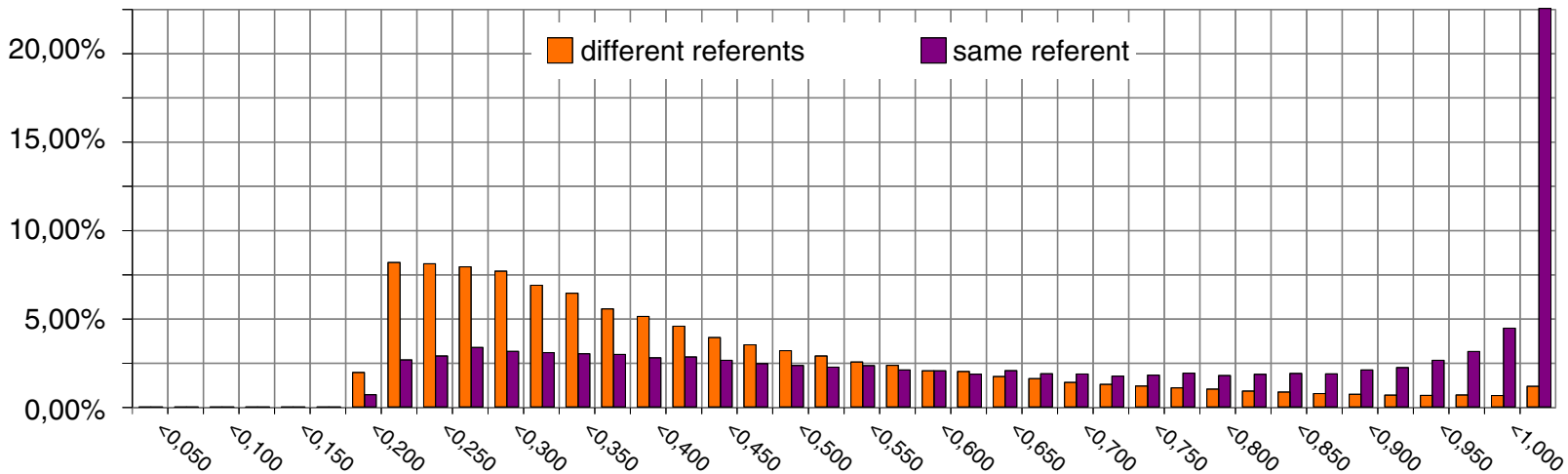


Retrieval Model	$F_{1/3}$ -Measure
tf-idf vector space	0.39
context-based vector space	0.32
ESA Wikipedia persons	0.30
phrase structure grammar	0.17
ontology alignment	0.15
optimized combination	**0.42**

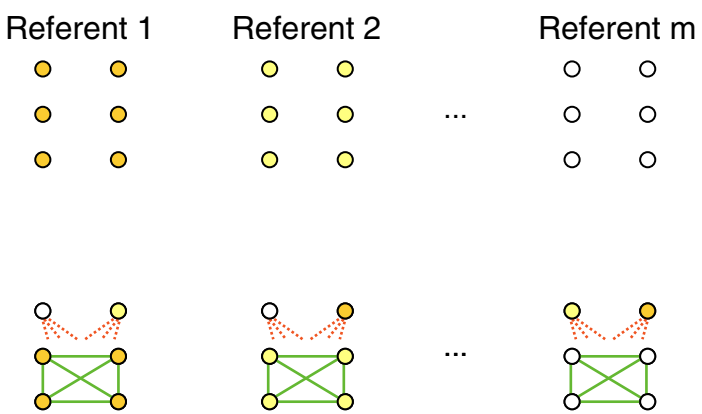


Constrained Cluster Analysis

Membership Distribution under Optimized Retrieval Model Combination

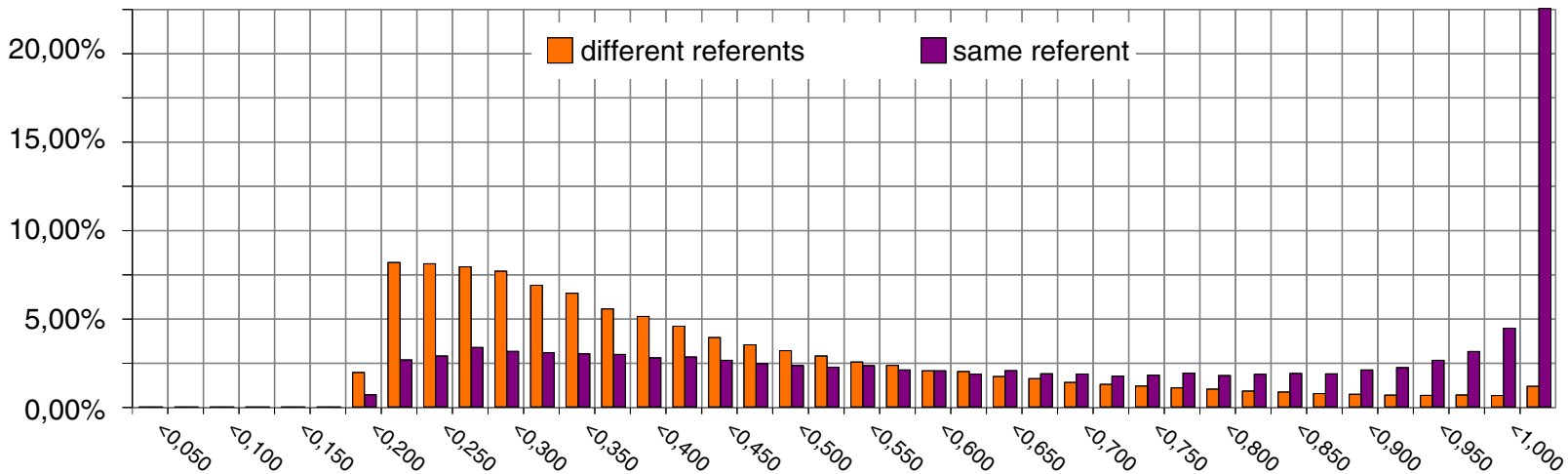


Retrieval Model	$F_{1/3}$ -Measure
tf-idf vector space	0.39
context-based vector space	0.32
ESA Wikipedia persons	0.30
phrase structure grammar	0.17
ontology alignment	0.15
optimized combination	**0.42**



Constrained Cluster Analysis

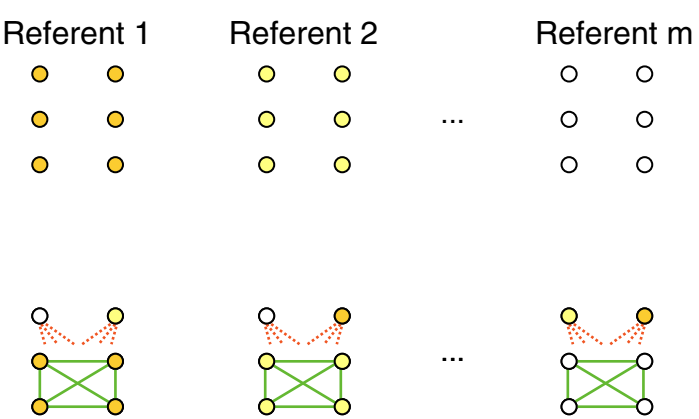
Membership Distribution under Optimized Retrieval Model Combination



In the example:

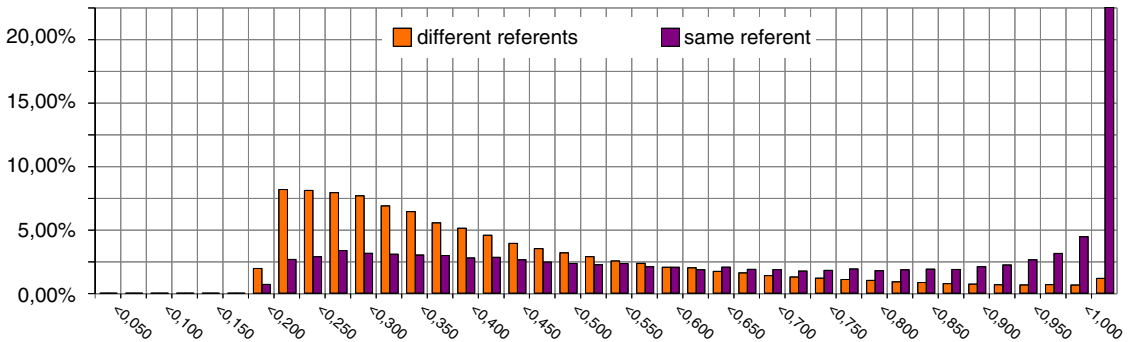
- precision = 0.4
- recall = 0.43
- $F_{1/3}$ = 0.41

(if false negatives are uniformly distributed)

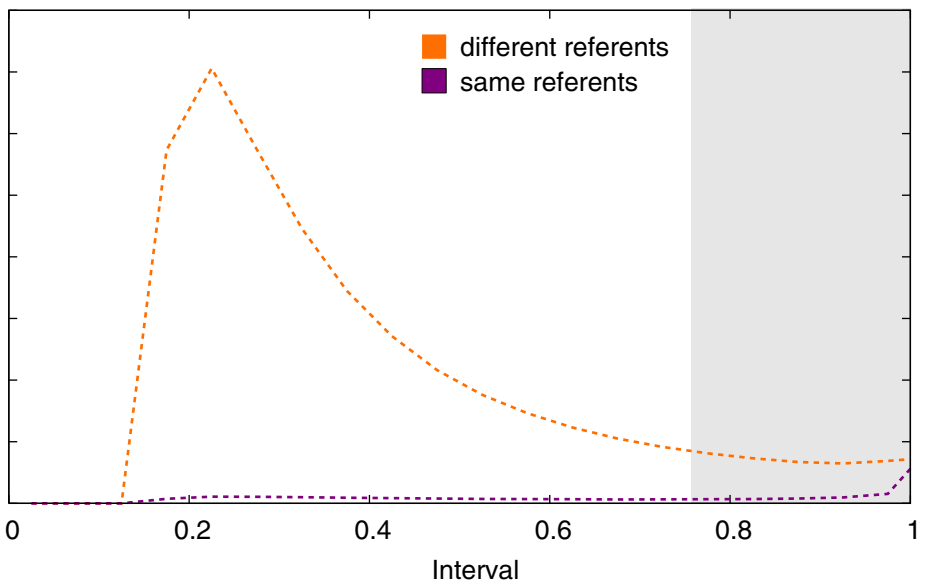


Constrained Cluster Analysis

In-Depth: Analysis of Classifier Effectiveness



Consideration of imbalance:



Constrained Cluster Analysis

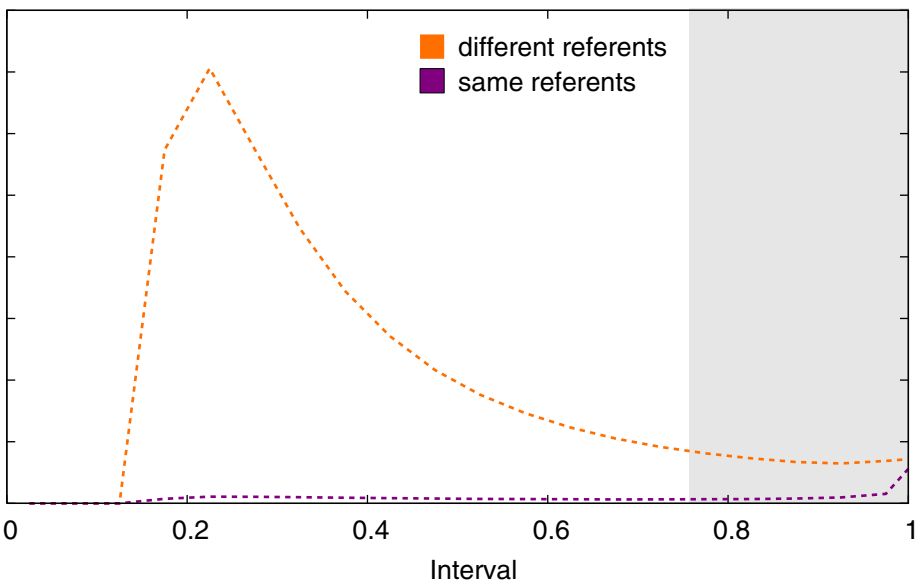
In-Depth: Analysis of Classifier Effectiveness

- ❑ class imbalance factor (*CIF*) of 25

⇒ precision in interval [0.725; 1] for edges between same referents: ≈ 0.17

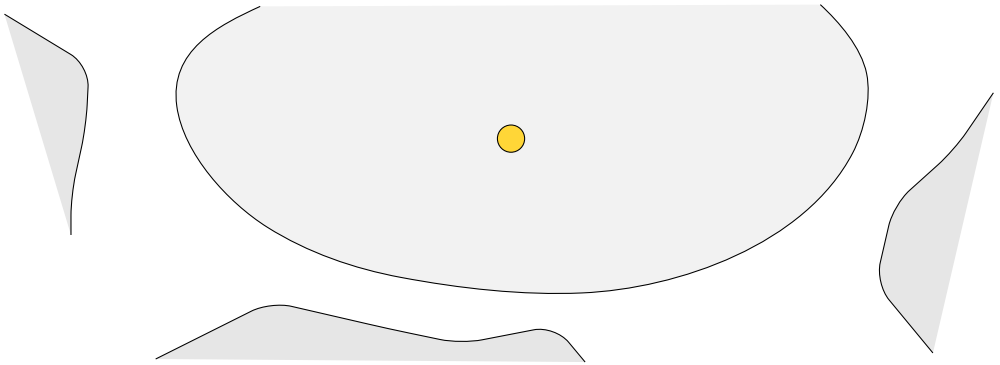
How can $F_{1/3} = 0.42$ be achieved via cluster analysis?

Consideration of imbalance:



Constrained Cluster Analysis

In-Depth: Analysis of Classifier Effectiveness



Assumption: uniform distribution of referents over documents (here: 25 clusters with $|C| = 23$)

- $\Rightarrow$ $|TP|$ true 1-similarities per cluster (here: 130 @ threshold 0.725)
- $\Rightarrow$ $\frac{|TP|}{|C|}$ degree of true positives per node (here: 11)
- $\Rightarrow$ $|TP|(\frac{1}{precision} - 1)$ false 1-similarities per cluster (here: 760)

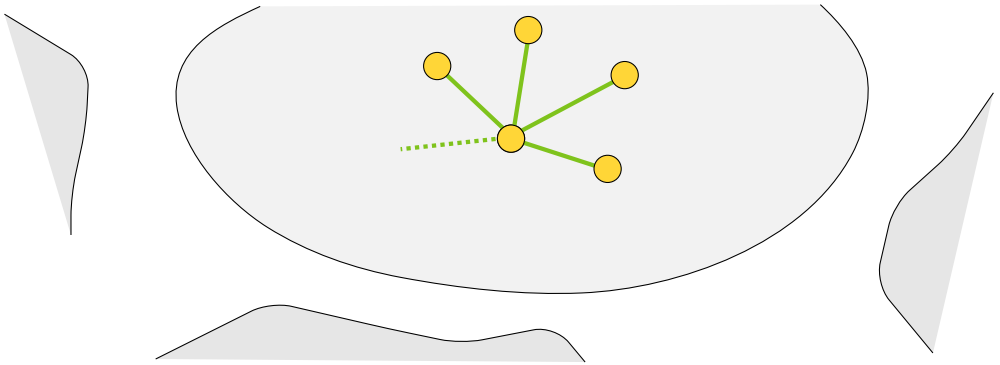
Density-based cluster analysis: effective false positives, FP^*, connect to same cluster

- $\Rightarrow$ analyze $P(|FP^*| > k \mid D, R_{iid})$ (here: $E(|FP^*|) = 2.7$)
- $\Rightarrow$ edge tie factor (ETF) specifies the excess of true positives until tie (here: $3 \ldots 5$)

$$ETF = \frac{|TP|}{|C| \cdot E(|FP^*|)}, \qquad \text{effective precision} = precision \cdot \frac{CIF}{ETF}$$

Constrained Cluster Analysis

In-Depth: Analysis of Classifier Effectiveness



Assumption: uniform distribution of referents over documents (here: 25 clusters with $|C| = 23$)

- $\Rightarrow$ $|TP|$ true 1-similarities per cluster (here: 130 @ threshold 0.725)
- $\Rightarrow$ $\frac{|TP|}{|C|}$ degree of true positives per node (here: 11)
- $\Rightarrow$ $|TP|(\frac{1}{precision} - 1)$ false 1-similarities per cluster (here: 760)

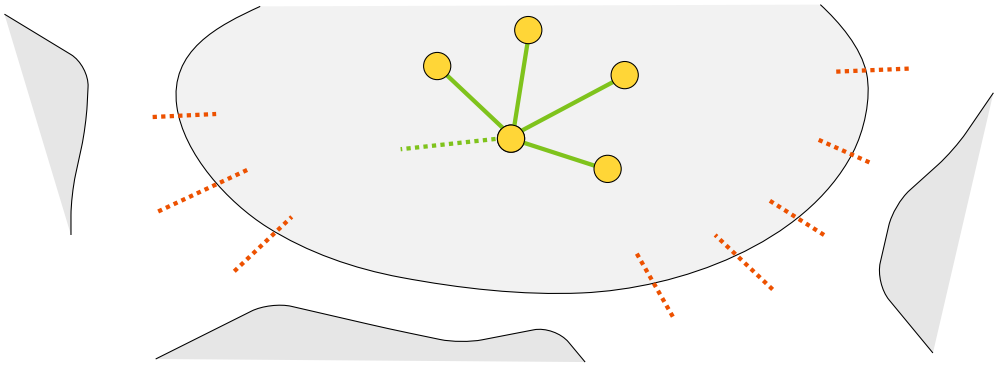
Density-based cluster analysis: effective false positives, FP^*, connect to same cluster

- $\Rightarrow$ analyze $P(|FP^*| > k \mid D, R_{iid})$ (here: $E(|FP^*|) = 2.7$)
- $\Rightarrow$ edge tie factor (ETF) specifies the excess of true positives until tie (here: $3 \ldots 5$)

$$ETF = \frac{|TP|}{|C| \cdot E(|FP^*|)}, \qquad \text{effective precision} = precision \cdot \frac{CIF}{ETF}$$

Constrained Cluster Analysis

In-Depth: Analysis of Classifier Effectiveness



Assumption: uniform distribution of referents over documents (here: 25 clusters with $|C| = 23$)

- $\Rightarrow$ $|TP|$ true 1-similarities per cluster (here: 130 @ threshold 0.725)
- $\Rightarrow$ $\frac{|TP|}{|C|}$ degree of true positives per node (here: 11)
- $\Rightarrow$ $|TP|(\frac{1}{precision} - 1)$ false 1-similarities per cluster (here: 760)

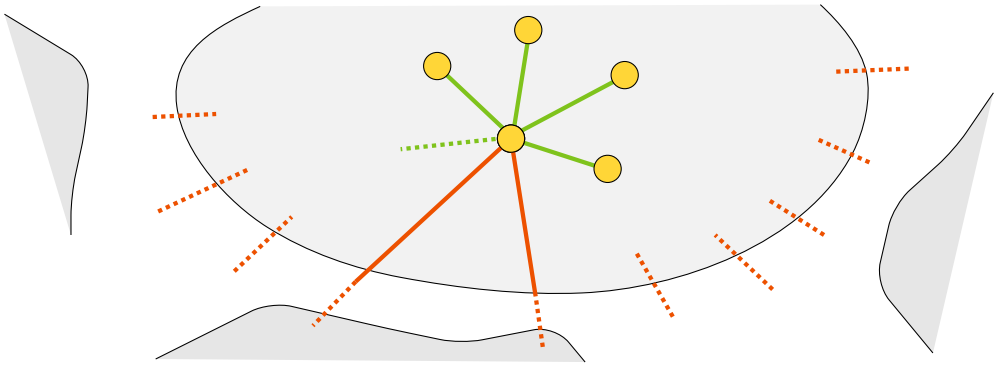
Density-based cluster analysis: effective false positives, FP^*, connect to same cluster

- $\Rightarrow$ analyze $P(|FP^*| > k \mid D, R_{iid})$ (here: $E(|FP^*|) = 2.7$)
- $\Rightarrow$ edge tie factor (ETF) specifies the excess of true positives until tie (here: $3 \ldots 5$)

$$ETF = \frac{|TP|}{|C| \cdot E(|FP^*|)}, \qquad \text{effective precision} = precision \cdot \frac{CIF}{ETF}$$

Constrained Cluster Analysis

In-Depth: Analysis of Classifier Effectiveness



Assumption: uniform distribution of referents over documents (here: 25 clusters with $|C| = 23$)

- $\Rightarrow$ $|TP|$ true 1-similarities per cluster (here: 130 @ threshold 0.725)
- $\Rightarrow$ $\frac{|TP|}{|C|}$ degree of true positives per node (here: 11)
- $\Rightarrow$ $|TP|(\frac{1}{precision} - 1)$ false 1-similarities per cluster (here: 760)

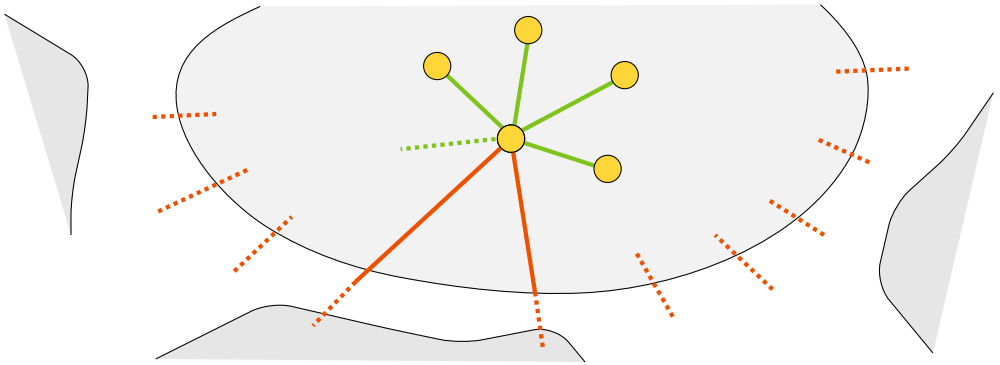
Density-based cluster analysis: effective false positives, FP^*, connect to same cluster

- $\Rightarrow$ analyze $P(|FP^*| > k \mid D, R_{iid})$ (here: $E(|FP^*|) = 2.7$)
- $\Rightarrow$ edge tie factor (ETF) specifies the excess of true positives until tie (here: $3 \ldots 5$)

$$ETF = \frac{|TP|}{|C| \cdot E(|FP^*|)}, \qquad \text{effective precision} = precision \cdot \frac{CIF}{ETF}$$

Constrained Cluster Analysis

In-Depth: Analysis of Classifier Effectiveness



Assumption: uniform distribution of referents over documents (here: 25 clusters with $|C| = 23$)

- $\Rightarrow$ $|TP|$ true 1-similarities per cluster (here: 130 @ threshold 0.725)
- $\Rightarrow$ $\frac{|TP|}{|C|}$ degree of true positives per node (here: 11)
- $\Rightarrow$ $|TP|(\frac{1}{precision} - 1)$ false 1-similarities per cluster (here: 760)

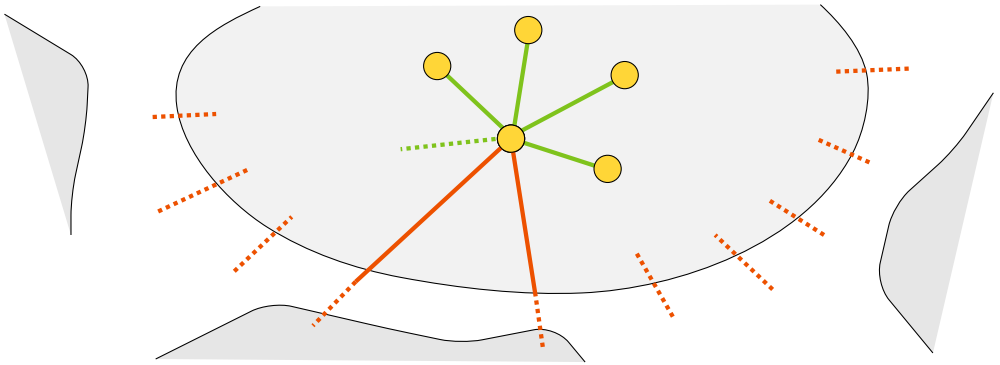
Density-based cluster analysis: effective false positives, FP^*, connect to same cluster

- $\Rightarrow$ analyze $P(|FP^*| > k \mid D, R_{iid})$ (here: $E(|FP^*|) = 2.7$)
- $\Rightarrow$ edge tie factor (*ETF*) specifies the excess of true positives until tie (here: $3 \ldots 5$)

$$ETF = \frac{|TP|}{|C| \cdot E(|FP^*|)}, \qquad \text{effective precision} = precision \cdot \frac{CIF}{ETF}$$

Constrained Cluster Analysis

In-Depth: Analysis of Classifier Effectiveness



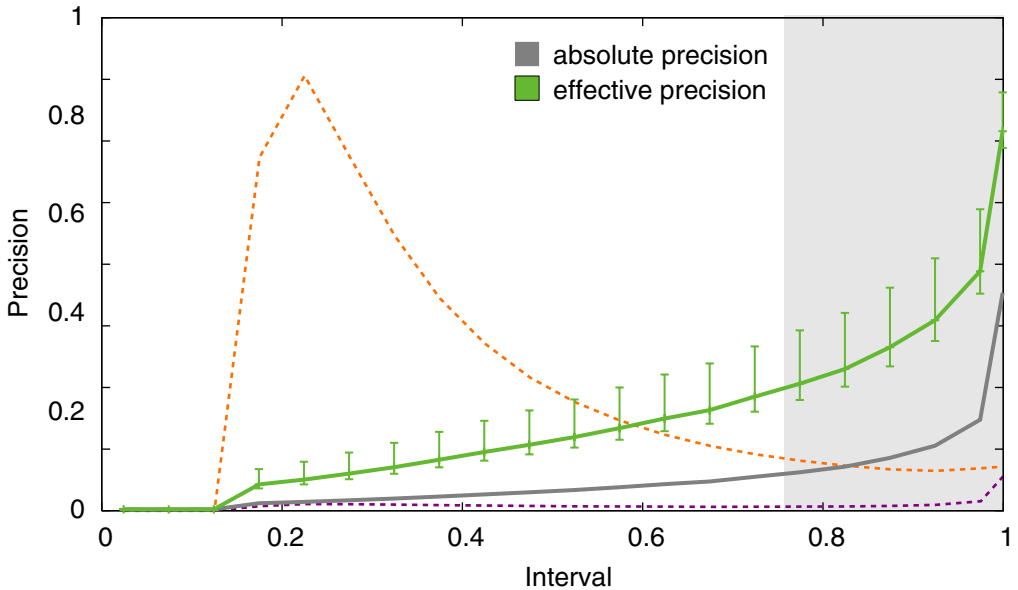
Assumption: uniform distribution of referents over documents (here: 25 clusters with $|C| = 23$)

- $\Rightarrow$ $|TP|$ true 1-similarities per cluster (here: 130 @ threshold 0.725)
- $\Rightarrow$ $\frac{|TP|}{|C|}$ degree of true positives per node (here: 11)
- $\Rightarrow$ $|TP|(\frac{1}{precision} - 1)$ false 1-similarities per cluster (here: 760)

Density-based cluster analysis: effective false positives, FP^*, connect to same cluster

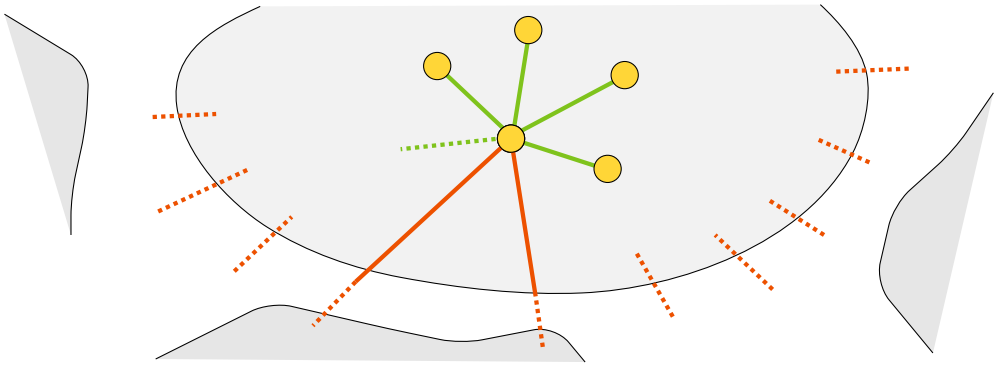
- $\Rightarrow$ analyze $P(|FP^*| > k \mid D, R_{iid})$ (here: $E(|FP^*|) = 2.7$)
- $\Rightarrow$ edge tie factor (*ETF*) specifies the excess of true positives until tie (here: $3 \ldots 5$)

$$ETF = \frac{|TP|}{|C| \cdot E(|FP^*|)}, \qquad \text{effective precision} = precision \cdot \frac{CIF}{ETF}$$



Constrained Cluster Analysis

In-Depth: Analysis of Classifier Effectiveness



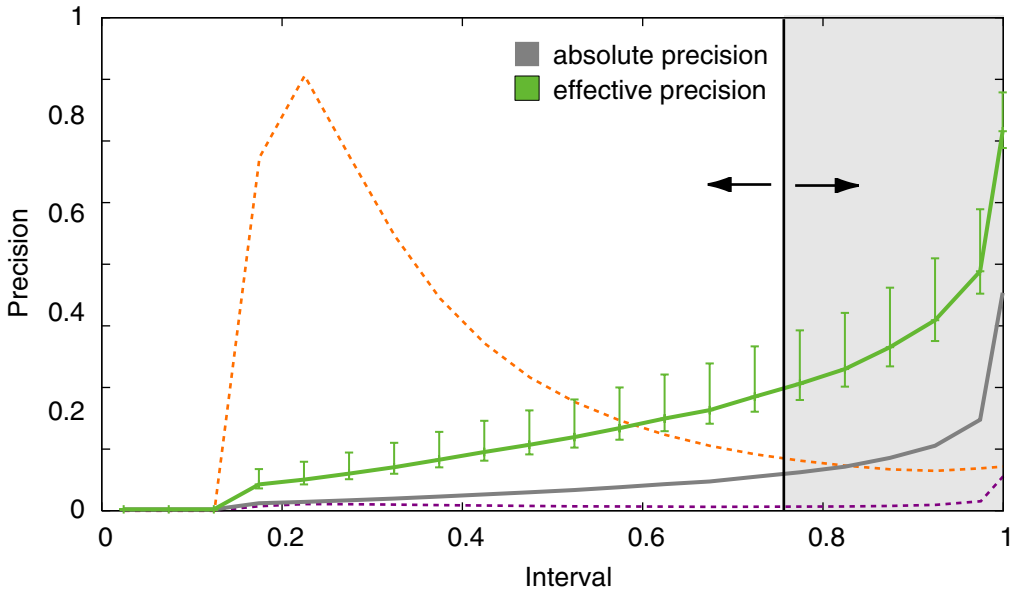
Assumption: uniform distribution of referents over documents (here: 25 clusters with $|C| = 23$)

- $\Rightarrow$ $|TP|$ true 1-similarities per cluster (here: 130 @ threshold 0.725)
- $\Rightarrow$ $\frac{|TP|}{|C|}$ degree of true positives per node (here: 11)
- $\Rightarrow$ $|TP|(\frac{1}{precision} - 1)$ false 1-similarities per cluster (here: 760)

Density-based cluster analysis: effective false positives, FP^*, connect to same cluster

- $\Rightarrow$ analyze $P(|FP^*| > k \mid D, R_{iid})$ (here: $E(|FP^*|) = 2.7$)
- $\Rightarrow$ edge tie factor (*ETF*) specifies the excess of true positives until tie (here: $3 \ldots 5$)

$$ETF = \frac{|TP|}{|C| \cdot E(|FP^*|)}, \qquad \text{effective precision} = precision \cdot \frac{CIF}{ETF}$$



Determine optimum similarity threshold for class-membership function:

$$\theta^* = \underset{\theta \in [0;1]}{\text{argmax}} \left\{ \frac{1+\alpha}{\frac{ETF}{precision_\theta \cdot CIF} + \frac{\alpha}{recall_\theta}} \right\}$$

θ^* considers co-variate shift, introduces model formation bias and sample selection bias.

Constrained Cluster Analysis

Model Selection: Our Risk Minimization Strategy

Retrieval Model	$F_{1/3}$ -Measure
tf-idf vector space	0.39
context-based vector space	0.32
ESA Wikipedia persons	0.30
phrase structure grammar	0.17
ontology alignment	0.15
optimized combination	**0.42**
Ensemble cluster analysis	**0.40**

Ensemble cluster analysis: higher bias, better generalization.

(1) Do we speculate on a better fit for D_{test}?

(2) Do we expect a significant covariate shift, more noise, etc. in D_{test}?

Constrained Cluster Analysis

Recap

1. Multi-document resolution can be tackled with constrained cluster analysis.
2. Constraints are derived from labeled examples.
3. Class membership function ties constraints to multiple retrieval models.
4. Advanced density-based clustering technology is key.

Constrained Cluster Analysis

References

- Disambiguating Web Appearances of People in a Social Network.
  [R. Bekkerman, A. McCallum. WWW 2005]
- A Bayesian Model for Supervised Clustering with the Dirichlet Process Prior.
  [H. Daumé III, D. Marcu. Journal MLR 2005]
- Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis.
  [E. Gabrilovich, S. Markovitch. IJCAI 2007]
- Unsupervised Discrimination of Person Names in Web Contexts.
  [T. Pedersen, A. Kulkarni. CICLing 2007]
- On Information Need and Categorizing Search.
  [S. Meyer zu Eissen. Dissertation, Paderborn University, 2007]
- Weighted Experts: A Solution for the Spock Data Mining Challenge.
  [B. Stein, S. Meyer zu Eissen. I-KNOW 2008]
- GRAPE: A System for Disambiguating and Tagging People Names in Web Search.
  [L. Jiang, W. Shen, J. Wang, N. An. WWW 2010]