Ludwig-Maximilians-Universität München
Lehrstuhl für Datenbanksysteme und Data Mining
Prof. Dr. Thomas Seidl

Knowledge Discovery and Data Mining 1

(Data Mining Algorithms 1)

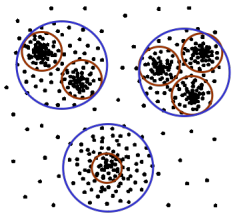
Winter Semester 2019/20

Agenda

Agenda

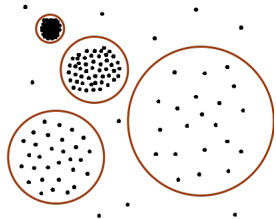
From Partitioning to Hierarchical Clustering

Global parameters to separate all clusters with a partitioning clustering method may not exist:



hierarchical cluster structure

and/or

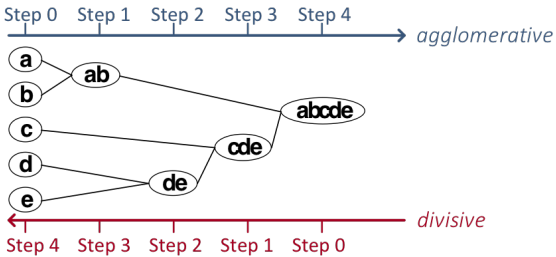


largely differing densities and sizes

Need a hierarchical clustering algorithm in these situations

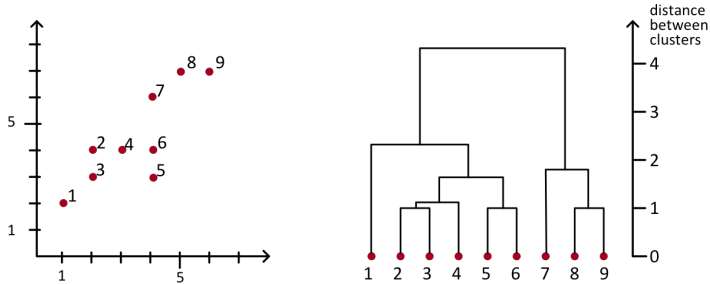
Hierarchical Clustering: Basic Notions

- Hierarchical decomposition of the data set (with respect to a given similarity measure) into a set of nested clusters
- Result represented by a so called *dendrogram* (greek $\delta\epsilon\nu\delta\rho o$ = tree)
  - Nodes in the dendrogram represent possible clusters
  - Dendrogram can be constructed bottom-up (agglomerative approach) or top down (divisive approach)



Hierarchical Clustering: Example

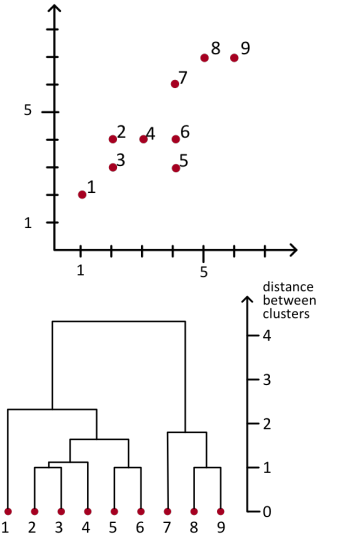
- Interpretation of the dendrogram
  - The root represents the whole data set
  - A leaf represents a single object in the data set
  - An internal node represents the union of all objects in its sub-tree
  - The height of an internal node represents the distance between its two child nodes



Agglomerative Hierarchical Clustering

Generic Algorithm

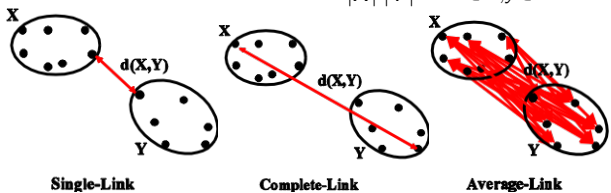
1. Initially, each object forms its own cluster
2. Consider all pairwise distances between the initial clusters (objects)
3. Merge the closest pair (A, B) in the set of the current clusters into a new cluster $C = A \cup B$
4. Remove A and B from the set of current clusters; insert C into the set of current clusters
5. If the set of current clusters contains only C (i.e., if C represents all objects from the database): STOP
6. Else: determine the distance between the new cluster C and all other clusters in the set of current clusters and go to step 3.



Single-Link Method and Variants

- Agglomerative hierarchical clustering requires a distance function for clusters
- Given: a distance function $dist(p, q)$ for database objects
- The following distance functions for clusters (i.e., sets of objects) X and Y are commonly used for hierarchical clustering:

$$\begin{aligned}
&\text{Single-Link:} && dist_{sl}(X, Y) = \min_{x \in X, y \in Y} dist(x, y) \\
&\text{Complete-Link:} && dist_{cl}(X, Y) = \max_{x \in X, y \in Y} dist(x, y) \\
&\text{Average-Link:} && dist_{al}(X, Y) = \frac{1}{|X| \cdot |Y|} \sum_{x \in X, y \in Y} dist(x, y)
\end{aligned}$$



Divisive Hierarchical Clustering

General Approach: Top Down

- Initially, all objects form one cluster
- Repeat until all clusters are singletons
  - Choose a cluster to split $\rightarrow$ *how?*
  - Replace the chosen cluster with the sub-clusters and split into two $\rightarrow$ *how to split?*

Example solution: DIANA

- Select the cluster C with largest diameter for splitting
- Search the most disparate object o in C (highest average dissimilarity)
  - Splinter group $S = \{o\}$
  - Iteratively assign the $o' \notin S$ with the highest $D(o') > 0$ to the splinter group until $D(o') \leq 0$ for all $o' \notin S$, where

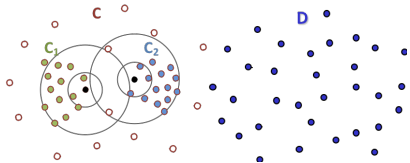
$$D(o') = \sum_{o_j \in C \setminus S} \frac{d(o', o_j)}{|C \setminus S|} - \sum_{o_i \in S} \frac{d(o', o_i)}{|S|}$$

Discussion Agglomerative vs. Divisive HC

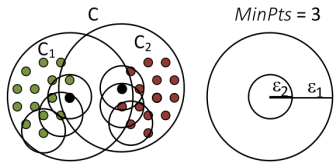
- Divisive and Agglomerative HC need $n-1$ steps
  - Agglomerative HC has to consider $\frac{n(n-1)}{2} = \binom{n}{2}$ combinations in the first step
  - Divisive HC potentially has $2^{n-1}-1$ many possibilities to split the data in its first step. Not every possibility has to be considered (DIANA)
- Divisive HC is conceptually more complex since it needs a second "flat" clustering algorithm (splitting procedure)
- Agglomerative HC decides based on local patterns
- Divisive HC uses complete information about the global data distribution $\rightsquigarrow$ able to provide better clusterings than Agglomerative HC?

Density-Based Hierarchical Clustering

- *Observation:* Dense clusters are completely contained by less dense clusters



- *Idea:* Process objects in the "right" order and keep track of point density in their neighborhood



Core Distance and Reachability Distance

Parameters: "generating" distance ϵ, fixed value *MinPts*

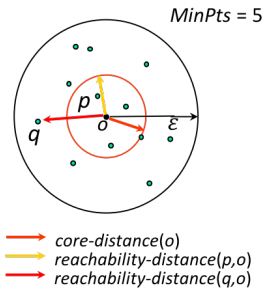
core-dist$_{\epsilon,MinPts}(o)$

- "smallest distance such that o is a core object"
- if core-dist $> \epsilon$: *undefined*

reach-dist$_{\epsilon,MinPts}(p, o)$

- "smallest dist. s.t. p is directly density-reachable from o"
- if reach-dist $> \epsilon$: ∞

$$\text{reach-dist}(p,o) = \begin{cases} dist(p,o) & , dist(p,o) \geq \text{core-dist}(o) \\ \text{core-dist}(o) & , dist(p,o) < \text{core-dist}(o) \\ \infty & , dist(p,o) > \epsilon \end{cases}$$

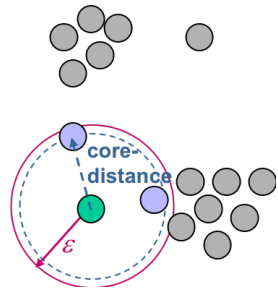


The Algorithm OPTICS

OPTICS[1]: Main Idea

"Ordering Points To Identify the Clustering Structure"

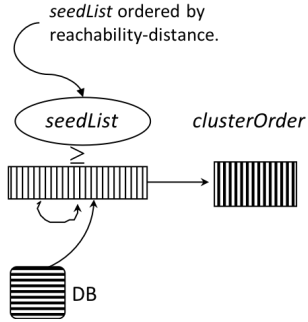
- Maintain two data structures
  - *seedList*: Stores all objects with shortest reachability distance seen so far ("distance of a jump to that point") in ascending order; organized as a heap
  - *clusterOrder*: Resulting cluster order is constructed sequentially (order of objects + reachability-distances)
- Visit each point
  - Always make a shortest jump



[1] Ankerst M., Breunig M., Kriegel H.-P., Sander J. "OPTICS: Ordering Points To Identify the Clustering Structure". SIGMOD (1999)

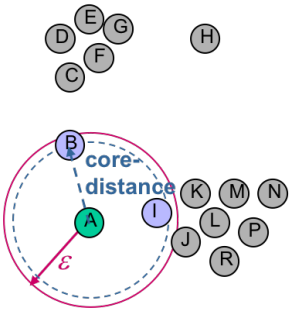
The Algorithm OPTICS

1: $seedList = \emptyset$
2: **while** there are unprocessed objects in DB **do**
3: **if** $seedList = \emptyset$ **then**
4: insert arbitrary unprocessed object into *clusterOrder* with reach-dist $= \infty$
5: **else**
6: remove first object from *seedList* and insert into *clusterOrder* with its current reach-dist
7: // *Let o be the last object inserted into clusterOrder*
8: mark o as processed
9: **for** $p \in range(o, \epsilon)$ **do**
10: // Insert/update p in *seedList*
11: compute reach-dist(p, o)
12: *seedList*.update$(p,$ reach-dist$(p, o))$



OPTICS: Example

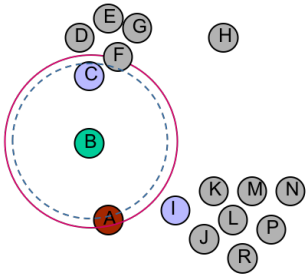
$\epsilon = 44, MinPts = 3$

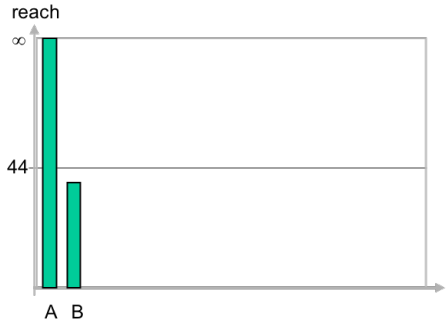


seed list: (B,40) (I, 40)

OPTICS: Example

$\epsilon = 44, MinPts = 3$

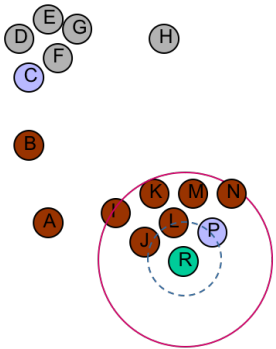


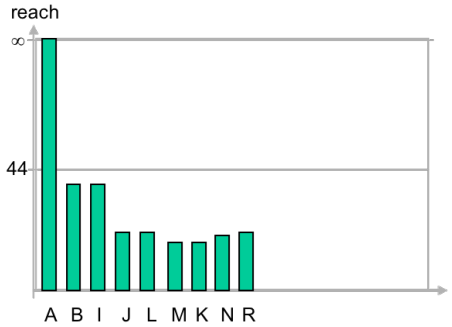


seed list: (I, 40) (C, 40)

OPTICS: Example

$\epsilon = 44, MinPts = 3$

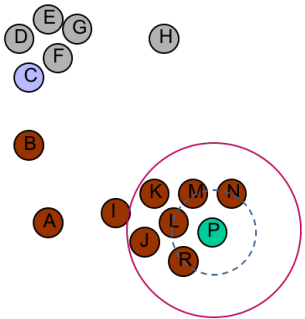




seed list: (P, 21) (C, 40)

OPTICS: Example

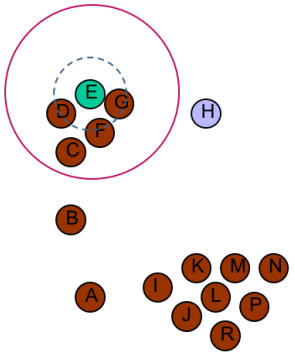
$\epsilon = 44, MinPts = 3$



seed list: (C, 40)

OPTICS: Example

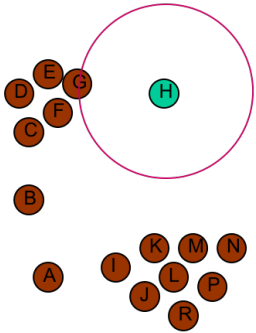
$\epsilon = 44, MinPts = 3$

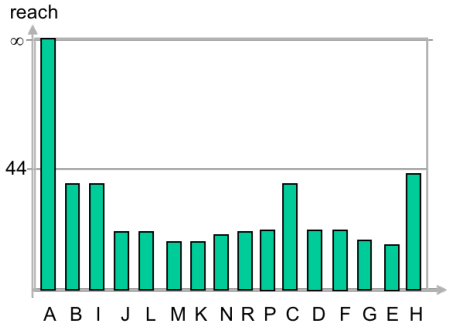


seed list: (H, 43)

OPTICS: Example

$\epsilon = 44, MinPts = 3$

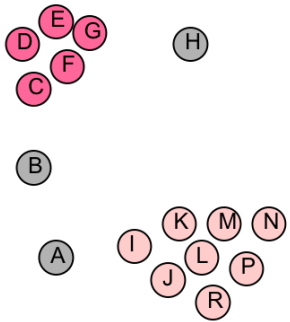


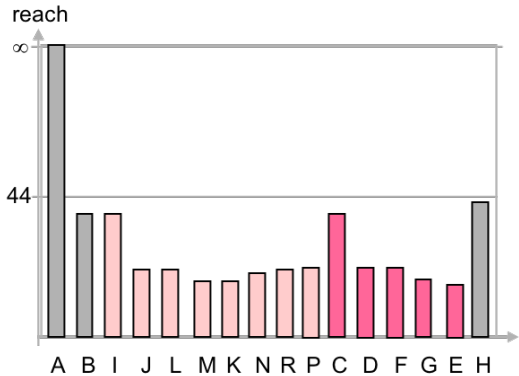


seed list: -

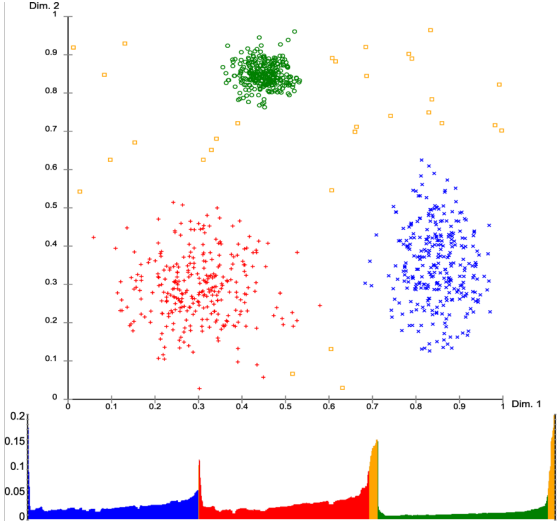
OPTICS: Example

$\epsilon = 44, MinPts = 3$



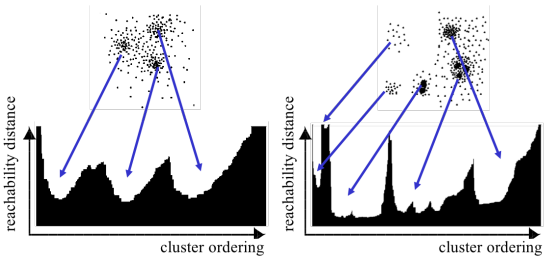


OPTICS: The Reachability Plot



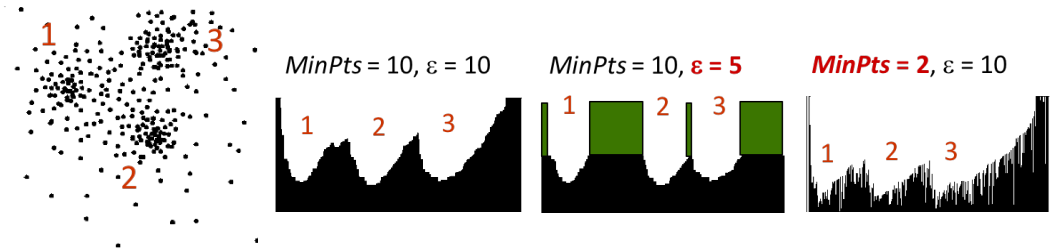
OPTICS: The Reachability Plot

- Plot the points together with their reachability-distances. Use the order in which they where returned by the algorithm
  - Represents the density-based clustering structure
  - Easy to analyze
  - Independent of the dimensionality of the data



OPTICS: Parameter Sensitivity

- Relatively insensitive to parameter settings
- Good result if parameters are just "large enough"



Hierarchical Clustering: Discussion

Advantages

- Does not require the number of clusters to be known in advance
- No (standard methods) or very robust parameters (OPTICS)
- Computes a complete hierarchy of clusters
- Good result visualizations integrated into the methods
- A "flat" partition can be derived afterwards (e.g. via a cut through the dendrogram or the reachability plot)

Disadvantages

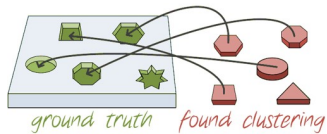
- May not scale well
  - Runtime for the standard methods: $\mathcal{O}(n^2 \log n^2)$
  - Runtime for OPTICS: without index support $\mathcal{O}(n^2)$
- User has to choose the final clustering

Agenda

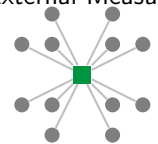
Evaluation of Clustering Results

Type	Positive	Negative
Expert's Opinion	may reveal new insight into the data	very expensive, results are not comparable
External Measures	objective evaluation	needs "ground truth"
Internal Measures	no additional information needed	approaches optimizing the evaluation criteria will always be preferred

Expert's Opinion



External Measure



Internal Measure

External Measures

Notation

Given a data set D, a clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$ and ground truth $\mathcal{G} = \{G_1, \ldots, G_l\}$.

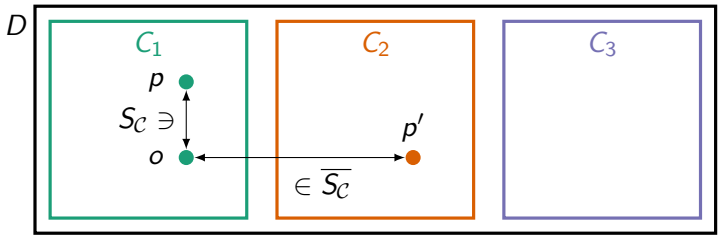
Problem

Since the cluster labels are "artificial", permuting them should not change the score.

Solution

Instead of comparing cluster and ground truth labels directly, consider all pairs of objects. Check whether they have the same label in $\mathcal{G}$ and if they have the same in $\mathcal{C}$.

Formalisation as Retrieval Problem for Clustering



With $P = \{(o, p) \in D \times D \mid o \neq p\}$ define:

- *Same* cluster label: $S_C = \{(o, p) \in P \mid \exists C_i \in C : \{o, p\} \subseteq C_i\}$
- *Different* cluster label: $\overline{S_C} = P \setminus S_C$

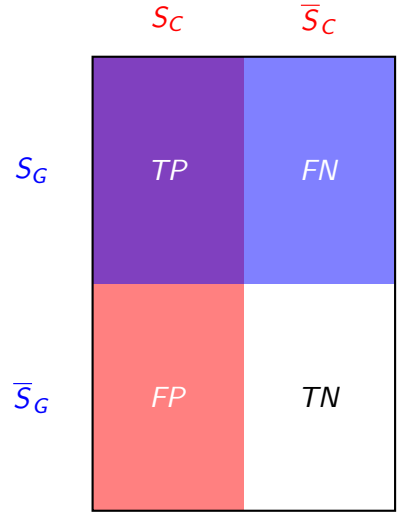
and analogously for $\mathcal{G}$.

Formalisation as Retrieval Problem for Clustering

Define

- $TP = |S_{\mathcal{C}} \cap S_{\mathcal{G}}|$
  (same cluster in both, "true positives")
- $FP = |S_{\mathcal{C}} \cap \overline{S_{\mathcal{G}}}|$
  (same cluster in $\mathcal{C}$, different cluster in $\mathcal{G}$, "false positives")
- $TN = |\overline{S_{\mathcal{C}}} \cap \overline{S_{\mathcal{G}}}|$
  (different cluster in both, "true negatives")
- $FN = |\overline{S_{\mathcal{C}}} \cap S_{\mathcal{G}}|$
  (different cluster in $\mathcal{C}$, same cluster in $\mathcal{G}$, "false negatives")

Note the difference to the definitions in classification!



External Measures - Retrieval Problem

- **Recall** ($0 \leq rec \leq 1$, larger is better)

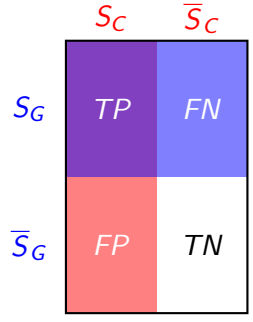
$$rec = \frac{TP}{TP + FN} = \frac{|S_C \cap S_G|}{|S_G|}$$

- **Precision** ($0 \leq prec \leq 1$, larger is better)

$$prec = \frac{TP}{TP + FP} = \frac{|S_C \cap S_G|}{|S_C|}$$

- F_1**-Measure** ($0 \leq F_1 \leq 1$, larger is better)

$$F_1 = \frac{2 \cdot rec \cdot prec}{rec + prec} = \frac{2|S_C \cap S_G|}{|S_C| + |S_G|}$$



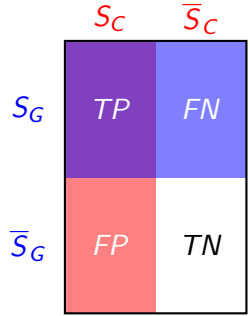
External Measures - Retrieval Problem

- **Rand Index** ($0 \leq RI \leq 1$, larger is better):

$$RI(\mathcal{C} \mid \mathcal{G}) = \frac{TP + TN}{TP + TN + FP + FN} = \frac{|S_{\mathcal{C}} \cap S_{\mathcal{G}}| + |\overline{S_{\mathcal{C}}} \cap \overline{S_{\mathcal{G}}}|}{|P|}$$

- **Adjusted Rand Index (ARI)**: Compares $RI(\mathcal{C}, \mathcal{G})$ against expected $(\mathcal{R}, \mathcal{G})$ of random cluster assignment $\mathcal{R}$.
- **Jaccard Coefficient** ($0 \leq JC \leq 1$, larger is better):

$$JC = \frac{TP}{TP + FP + FN} = \frac{|S_{\mathcal{C}} \cap S_{\mathcal{G}}|}{|P| - |\overline{S_{\mathcal{C}}} \cap \overline{S_{\mathcal{G}}}|}$$



External Measures - Retrieval Problem

- **Confusion Matrix / Contingency Table** $N \in \mathbb{N}^{k \times l}$ with $N_{ij} = |C_i \cap G_j|$

	G_1	$\dots$	G_l
C_1	$\lvert C_1 \cap G_1 \rvert$	$\dots$	$\lvert C_1 \cap G_l \rvert$
$\vdots$	$\vdots$	$\ddots$	
C_k	$\lvert C_k \cap G_1 \rvert$		$\lvert C_k \cap G_l \rvert$

- Define $N_i = \sum_{j=1}^{l} N_{ij}$ (i.e. $N_i = |C_i|$)
- Define $N = \sum_{i=1}^{k} N_i$ (i.e. $N = |D|$)

External Measures - Information Theory

- **(Shannon) Entropy**:

$$H(\mathcal{C}) = -\sum_{C_i \in \mathcal{C}} p(C_i) \log p(C_i) = -\sum_{C_i \in \mathcal{C}} \frac{|C_i|}{|D|} \log \frac{|C_i|}{|D|} = -\sum_{i=1}^{k} \frac{N_i}{N} \log \frac{N_i}{N}$$

- **Mutual Entropy**:

$$\begin{aligned} H(\mathcal{C} \mid \mathcal{G}) &= -\sum_{C_i \in \mathcal{C}} p(C_i) \sum_{G_j \in \mathcal{G}} p(G_j \mid C_i) \log p(G_j \mid C_i) \\ &= -\sum_{C_i \in \mathcal{C}} \frac{|C_i|}{|D|} \sum_{G_j \in \mathcal{G}} \frac{|C_i \cap G_j|}{|C_i|} \log \frac{|C_i \cap G_j|}{|C_i|} \\ &= -\sum_{i=1}^{k} \frac{N_i}{N} \sum_{j=1}^{l} \frac{N_{ij}}{N_i} \log \frac{N_{ij}}{N_i} \end{aligned}$$

External Measures - Information Theory

- **Mutual Information**:

$$I(\mathcal{C}, \mathcal{G}) = H(\mathcal{C}) - H(\mathcal{C} \mid \mathcal{G}) = H(\mathcal{G}) - H(\mathcal{G} \mid \mathcal{C})$$

- **Normalized Mutual Information (NMI)** ($0 \leq NMI \leq 1$, larger is better):

$$NMI(\mathcal{C}, \mathcal{G}) = \frac{I(\mathcal{C}, \mathcal{G})}{\sqrt{H(\mathcal{C})H(\mathcal{G})}}$$

- **Adjusted Mutual Information (AMI)**: Compares $MI(\mathcal{C}, \mathcal{G})$ against expected $MI(\mathcal{R}, \mathcal{G})$ of random cluster assignment $\mathcal{R}$.

Internal Measures: Cohesion

Notation

Let D be a set of size $n = |D|$, and let $\mathcal{C} = \{C_1, \ldots, C_k\}$ be a partitioning of D.

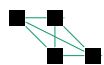
Cohesion

Average distance between objects of the same cluster.

$$coh(C_i) = \binom{|C_i|}{2}^{-1} \sum_{o,p \in C_i, o \neq p} d(o,p)$$

Cohesion of clustering is equal to weighted mean of the clusters' cohesions.

$$coh(\mathcal{C}) = \sum_{i=1}^{k} \frac{|C_i|}{n} coh(C_i)$$



Internal Measures: Separation

Separation

Separation between to clusters: Average distance between pairs

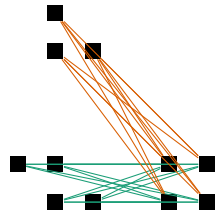
$$sep(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{o \in C_i, p \in C_j} d(o, p)$$

Separation of one cluster: Minimum separation to another cluster:

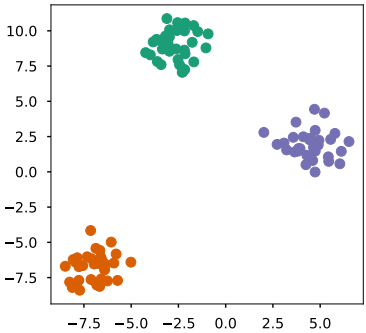
$$sep(C_i) = \min_{j \neq i} sep(C_i, C_j)$$

Separation of clustering is equal to weighted mean of the clusters' separations.

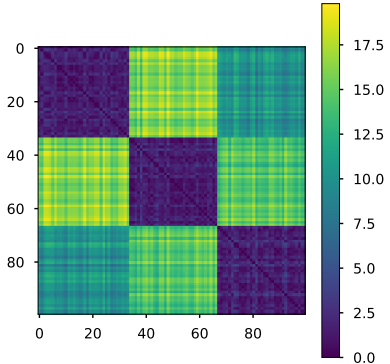
$$sep(\mathcal{C}) = \sum_{i=1}^{k} \frac{|C_i|}{n} sep(C_i)$$



Evaluating the Distance Matrix



dataset
(well separated)

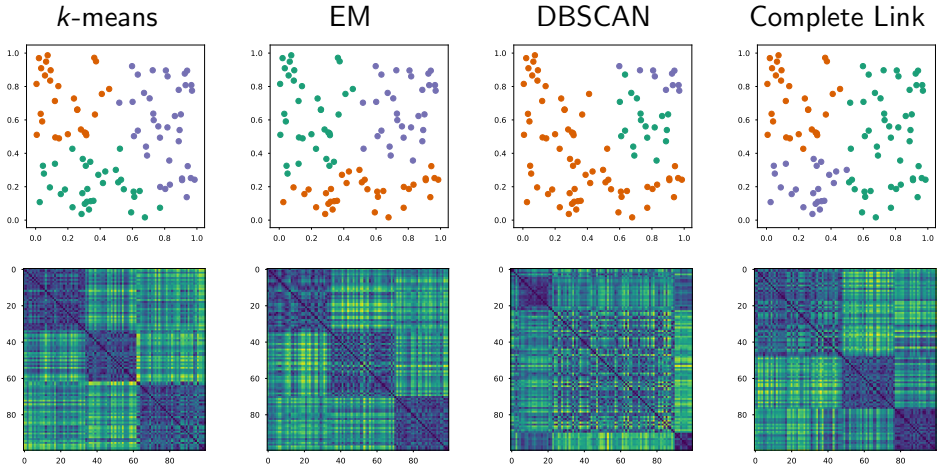


Distance matrix
(sorted by k-means cluster label)

after: Tan, Steinbach, Kumar: Introduction to Data Mining (Pearson, 2006)

Evaluating the Distance Matrix

Distance matrices differ for different clustering approaches (here on random data)

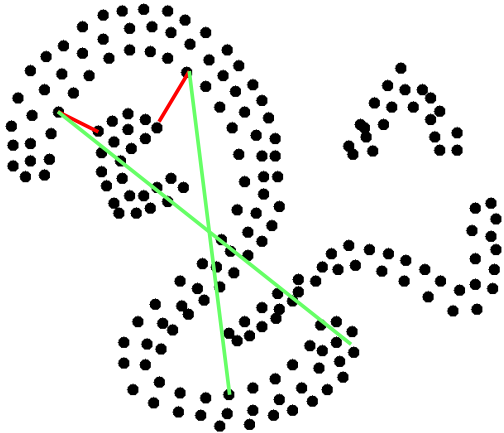


after: Tan, Steinbach, Kumar: Introduction to Data Mining (Pearson, 2006)

Cohesion and Separation

Problem

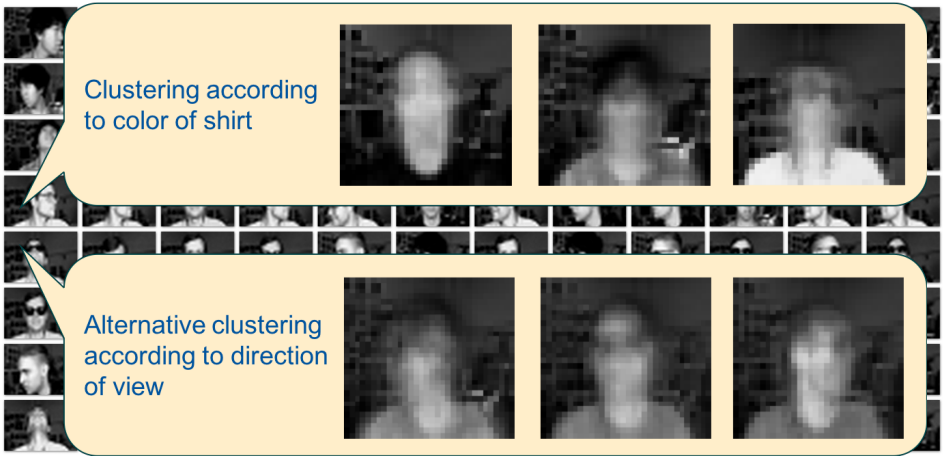
Suitable for convex cluster, but not for stretched clusters (cf. silhouette coefficient).



Ambiguity of Clusterings

- Clustering according to: Color of shirt, direction of view, glasses, ...

Ambiguity of Clusterings



- Clustering according to: Color of shirt, direction of view, glasses, ...

Ambiguity of Clusterings

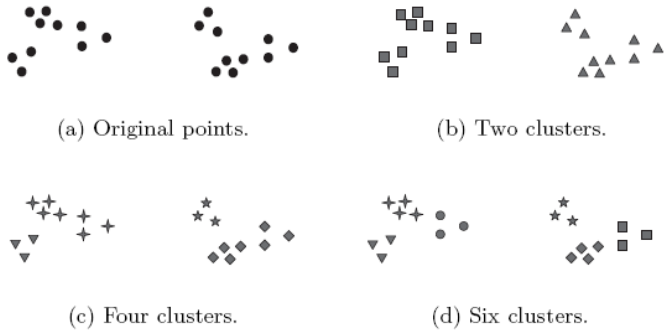


Figure 8.1. Different ways of clustering the same set of points.

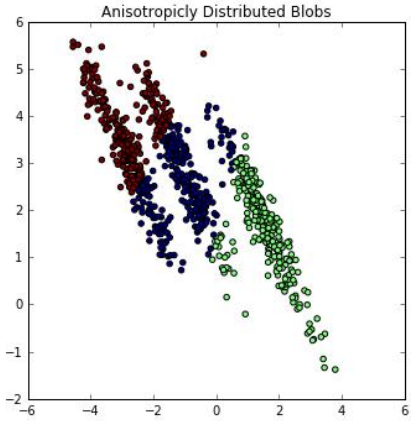
from: Tan, Steinbach, Kumar: Introduction to Data Mining (Pearson, 2006)

Ambiguity of Clusterings

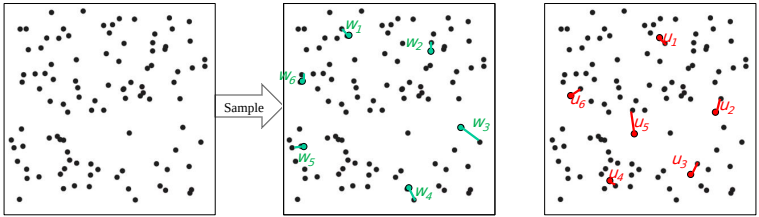
"Philosophical" Problem

"What is a correct clustering?"

- Most approaches find clusters in every dataset, even in uniformly distributed objects
- Are there clusters?
  - Apply clustering algorithm
  - Check for reasonability of clusters
- Problem: No clusters found $\neq$ no clusters existing
  - Maybe clusters exists only in certain models, but can not be found by used clustering approach



Hopkins Statistics



$$H = \frac{\sum_{i=1}^{m} u_i}{\sum_{i=1}^{m} u_i + \sum_{i=1}^{m} w_i}$$

- w_i: distance of selected objects to the next neighbor in dataset
- u_i: distances of uniformly distributed objects to next neighbor in dataset
- $0 \leq H \leq 1$;
  - $H \approx 0$: very regular data (e.g. grid);
  - $H \approx 0.5$: uniformly distributed data;
  - $H \approx 1$: strongly clustered,

Recap: Observed Clustering Methods

- Partitioning Methods: Find k partitions, minimizing some objective function
- Probabilistic Model-Based Clustering (EM)
- Density-based Methods: Find clusters based on connectivity and density functions
- Mean-Shift: Find modes in the point density
- Spectral Clustering: Find global minimum cut
- Hierarchical Methods: Create a hierarchical decomposition of the set of objects
- Evaluation: External and internal measures