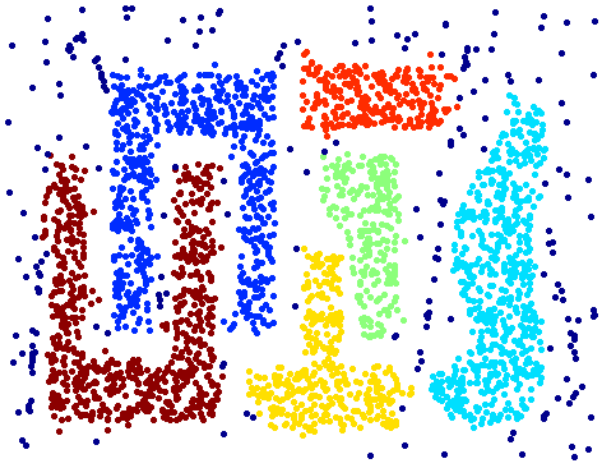
Clustering III
Agglomerative Clustering and DBSCAN



Tamás Horváth

Dept. of Computer Science III, University of Bonn

Fraunhofer IAIS, Schloß Birlinghoven, Sankt Augustin

tamas.horvath@iais.fraunhofer.de

many of the slides: Stefan Wrobel

universitätbonn

RHEINISCHE FRIEDRICH-WILHELMS-UNIVERSITÄT

Fraunhofer IAIS

K-Means Clustering Revisited

Clustering: Problem Definition (Recap)

Given:

- a set of instances $I = \{x_1, \ldots, x_n\}$ from an instance space X
- a quality measure q on sets of sets of instances, i.e.

$$q : 2^{2^X} \to \mathbb{R}$$

find a set $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ of *clusters* such that

- $C_i \subseteq X$ for all $i \in \{1, \ldots, k\}$,
- $I \subseteq \bigcup_{i=1,\ldots,k} C_i$,
- $q(\mathcal{C})$ is maximal

remark: often, $\mathcal{C}$ is required to be a partition, i.e., $C_i \cap C_j = \emptyset$ $(1 \leq i < j \leq k)$

Euclidean Sum-of-Squares Clustering

for $I = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ and integer $k > 0$, define the *distortion measure* by

$$J = \sum_{i=1}^{n} \sum_{j=1}^{k} z_{ij} \|x_i - \mu_j\|^2$$

where

- $z_{ij} \in \{0, 1\}$ defines whether or not x_i belongs to cluster C_j
  - $\sum_{j=1}^{k} z_{ij} = 1$ for all $i = 1, \ldots, n$
- $\mu_j \in \mathbb{R}^d$ is a prototype associated with cluster C_j,
- $\|\cdot\|$ is the Euclidean norm

Goal: find values for the $\{z_{ij}\}$ and the $\{\mu_j\}$ so as to *minimize* J

Complexity of Sum-of-Squares Clustering

Problem: *Given* a set $S \subset \mathbb{R}^d$ of points and a positive integer k, *find* a clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$ that *minimizes*

$$J = \sum_{i=1}^{n} \sum_{j=1}^{k} z_{ij} \|x_i - \mu_j\|^2$$

Thm. (Aloise, Deshpande, Hansen & Popat, 2009):
NP-hard even for $k = 2$ if d is arbitrary

Thm. (Mahajan, Nimbhorkar & Varadarajan, 2009):
NP-hard even for $d = 2$ (i.e., in the plane) if k is arbitrary

proofs: *omitted*

K-means Clustering Revisited

k**-means** clustering algorithm (LLoyd, 1957)

- choose some initial value for μ_j for all $j = 1, \ldots, k$
- repeat until convergence
  1. (E-step) minimize J w.r.t. z_{ij}, keeping μ_j fixed
  2. (M-step) minimize J w.r.t. μ_j, keeping z_{ij} fixed

K-means Clustering Revisited

E-step: minimize

$$J = \sum_{i=1}^{n} \sum_{j=1}^{k} z_{ij} \|x_i - \mu_j\|^2$$

with respect to z_{ij}, keeping μ_j fixed

solution: since J is a linear function of z_{ij},

$$z_{ij} = \begin{cases} 1 & \text{if } j = \operatorname{argmin}_l \|x_i - \mu_l\|^2 \\ 0 & \text{o/w} \end{cases}$$

i.e., we assign x_i to the closest cluster center

K-means Clustering Revisited

M-step: minimize

$$J = \sum_{i=1}^{n} \sum_{j=1}^{k} z_{ij} \|x_i - \mu_j\|^2$$

with respect to μ_j, keeping z_{ij} fixed

solution: as J is a quadratic function of μ_j, minimize J by setting $\frac{\partial J}{\partial \mu_j} = 0$, i.e.,

$$\begin{aligned} -2 \sum_{i=1}^{n} z_{ij}(x_i - \mu_j) &= 0 \\ \mu_j &= \frac{\sum_i z_{ij} x_i}{\sum_i z_{ij}} = \frac{1}{|C_j|} \sum_{x \in C_j} x = m(C_j) \end{aligned}$$

i.e., μ_j is set to the mean of all instances in cluster C_j

Hierarchial Clustering

Hierarchical Clustering

hierarchical clustering of a set $I = \{x_1, \ldots, x_n\}$ of instances:

a sequence $\mathcal{C}_1, \ldots, \mathcal{C}_n$ of nested partitions of I, where

- $\mathcal{C}_1$ contains n clusters, i.e., $\mathcal{C}_1 = \{\{x_1\}, \ldots, \{x_n\}\}$

  $\mathcal{C}_2$ contains $n-1$ clusters

  $\vdots$

  $\mathcal{C}_n$ contains 1 cluster, i.e., $\mathcal{C}_n = \{\{x_1, \ldots, x_n\}\}$

- if x, x' belong to the same cluster in $\mathcal{C}_k$ then they belong to the same cluster in $\mathcal{C}_l$ for all $l = k, k+1, \ldots, n$

dendrogram: natural (binary) tree representation of a hierarchial clustering

Dendograms

set of nested partitions can be visualized as a dendrogram

height: proportional to the distance/similarity at which two clusters have been merged

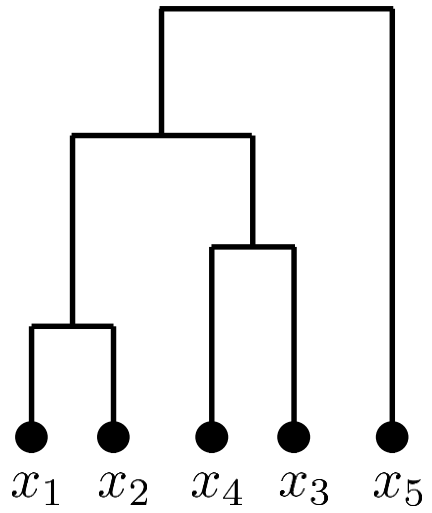
$$\mathcal{C}_5 = \{\{x_1, x_2, x_3, x_4, x_5\}\}$$

$$\mathcal{C}_4 = \{\{x_1, x_2, x_3, x_4\}, \{x_5\}\}$$

$$\mathcal{C}_3 = \{\{x_1, x_2\}, \{x_3, x_4\}, \{x_5\}\}$$

$$\mathcal{C}_2 = \{\{x_1, x_2\}, \{x_3\}, \{x_4\}, \{x_5\}\}$$

$$\mathcal{C}_1 = \{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\}\}$$



disjoint

conjoint

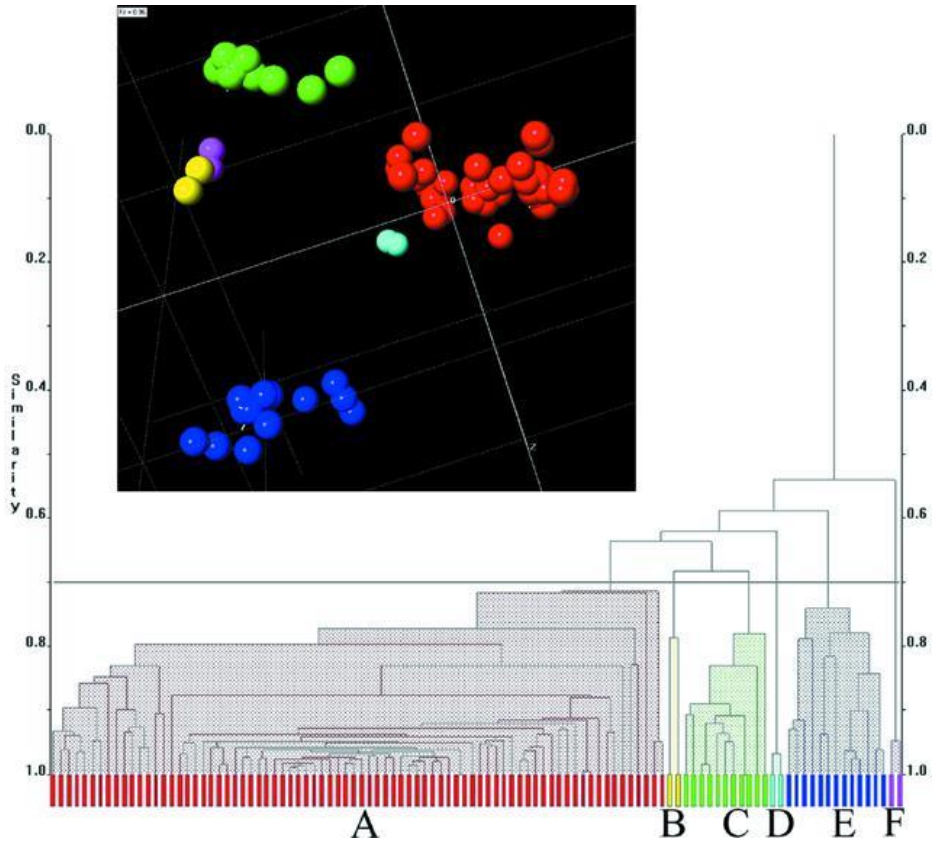
universitätbonn
RHEINISCHE FRIEDRICH-WILHELMS-UNIVERSITÄT

Fraunhofer IAIS

Dendograms

set of nested partitions can be visualized as a dendrogram

height: proportional to the distance/similarity at which two clusters have been merged



universitätbonn
RHEINISCHE FRIEDRICH-WILHELMS-UNIVERSITÄT

Fraunhofer IAIS

Strengths of Hierarchical Clustering

- no assumptions on the number of clusters (in contrast to e.g. k-Means)
  - any desired number of clusters can be obtained by “cutting” the dendogram at the proper level
- hierarchical clusterings may correspond to meaningful taxonomies
  - for example in biological sciences (e.g., phylogeny reconstruction, etc), web (e.g., product catalogs) etc

Hierarchical Clustering

- two main types of hierarchical clustering
  - **agglomerative (bottom-up, clumping):**
    - start with the instances as singleton clusters
    - at each step, merge the closest pair of clusters until only one cluster left
  - **divisive (top-down, splitting):**
    - start with one, all-inclusive cluster
    - at each step, split a cluster until each cluster becomes a singleton
- traditional hierarchical algorithms use a similarity or distance matrix to merge or split one cluster at a time

Agglomerative Clustering

algorithm:

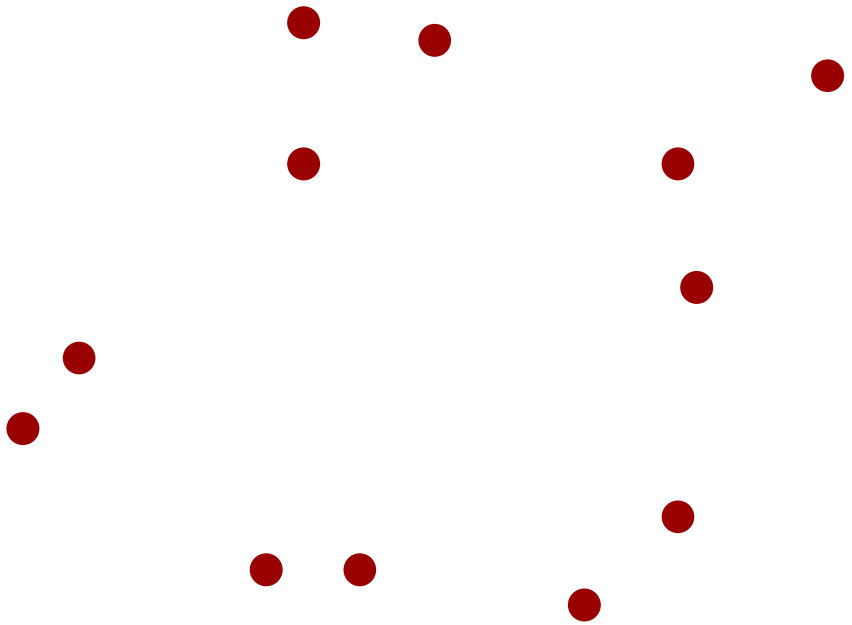
1. $\mathcal{C}_1 = \{\{x_1\}, \ldots, \{x_n\}\}$
2. $k := 1$
3. **for** $(k = 1;\ |\mathcal{C}_k| > 1;\ k = k + 1)$ **do**
4. find “closest" pair $C_i, C_j \in \mathcal{C}_k$ with $i \neq j$
5. $\mathcal{C}_{k+1} := \mathcal{C}_k \setminus \{C_i, C_j\}$
6. $\mathcal{C}_{k+1} := \mathcal{C}_{k+1} \cup \{C_i \cup C_j\}$
7. **return** $\mathcal{C}_1, \ldots, \mathcal{C}_n$

key operation: computing the distance between two clusters

- different distance definitions between clusters lead to different algorithms

Input/Initial setting

- start with clusters of individual points and a distance/proximity matrix



	p_1	p_2	p_3	p_4	p_5	$\cdots$
p_1						
p_2						
p_3						
p_4						
p_5						
$\vdots$						

distance/proximity matrix

universitätbonn
RHEINISCHE FRIEDRICH-WILHELMS-UNIVERSITÄT

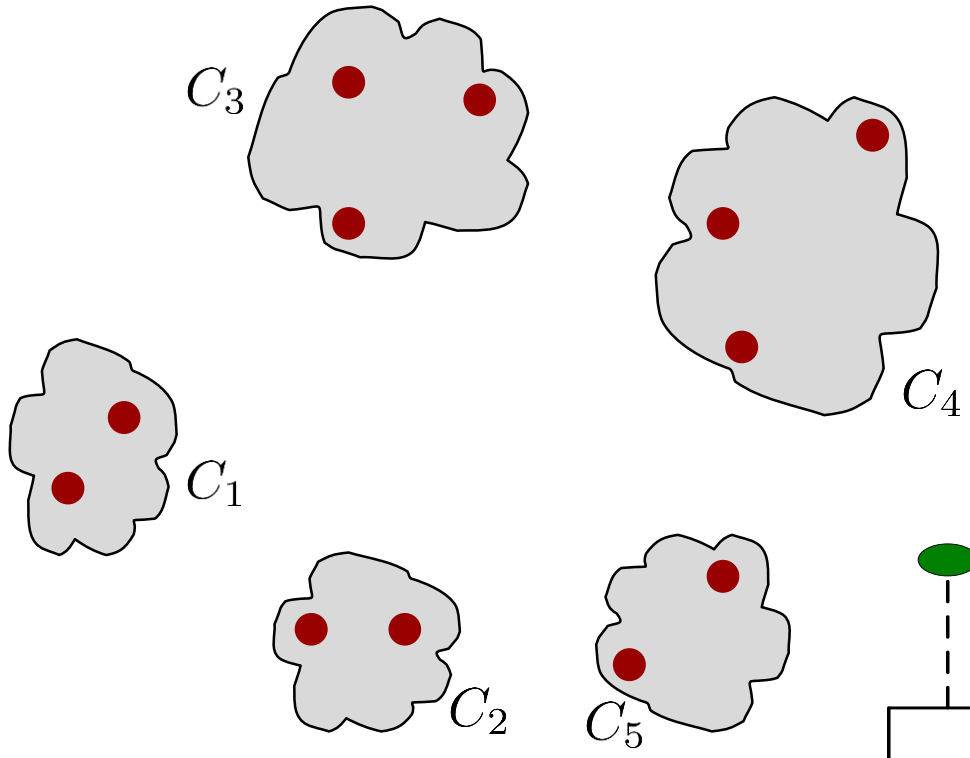
Fraunhofer IAIS

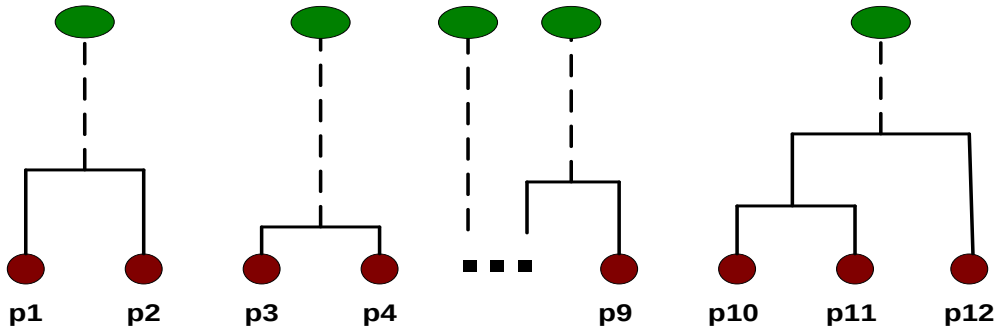
Intermediate State

- after some merging steps, we have some clusters

	C_1	C_2	C_3	C_4	C_5
C_1					
C_2					
C_3					
C_4					
C_5					

distance/proximity matrix



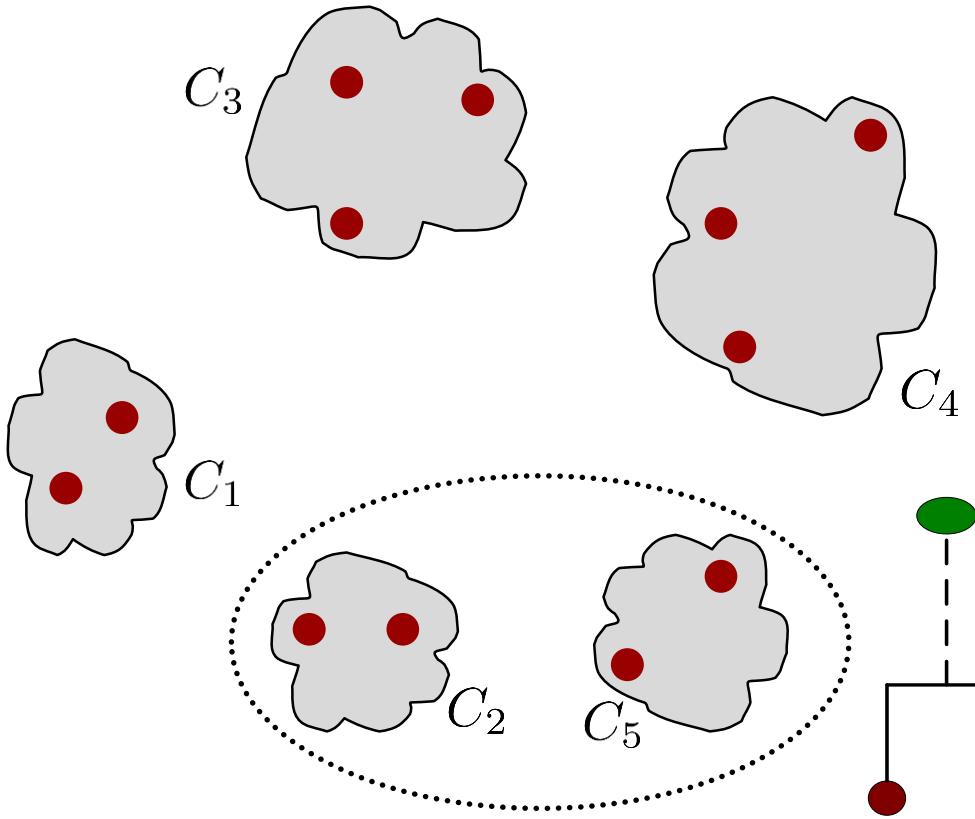


universitätbonn
RHEINISCHE FRIEDRICH-WILHELMS-UNIVERSITÄT

Fraunhofer IAIS

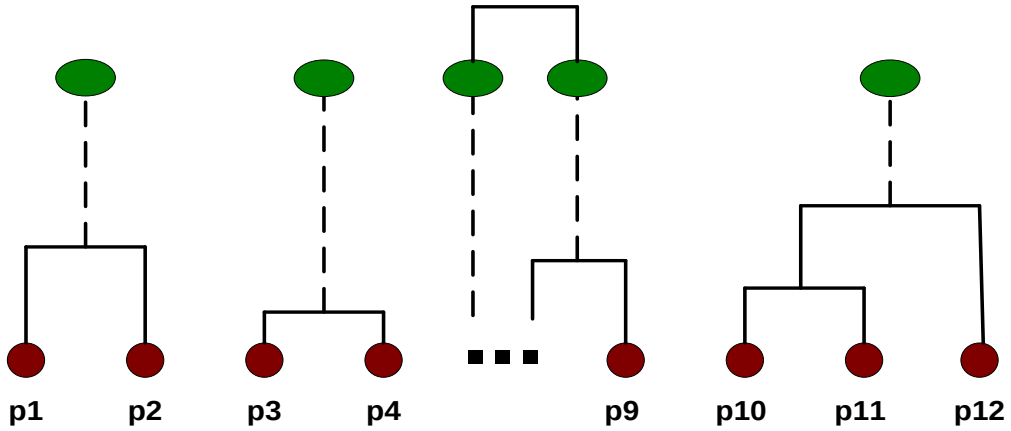
Intermediate State

- merge the two closest clusters (C_2 and C_5) and update the distance/proximity matrix



	C_1	C_2	C_3	C_4	C_5
C_1					
C_2					
C_3					
C_4					
C_5					

distance/proximity matrix



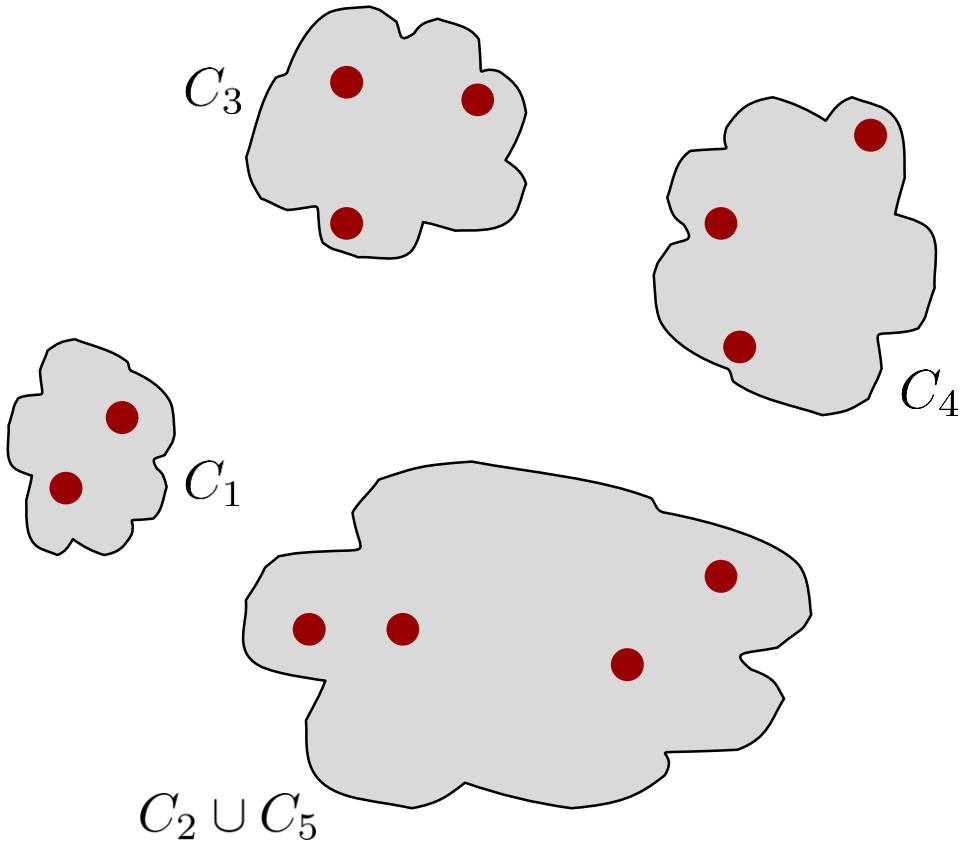
universitätbonn
RHEINISCHE FRIEDRICH-WILHELMS-UNIVERSITÄT

Fraunhofer IAIS

After Merging

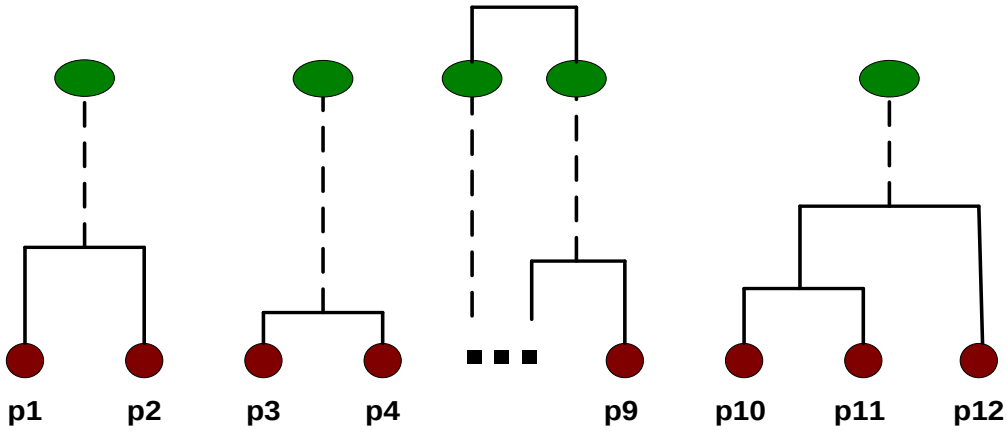
"How do we update the distance matrix?"

- depends on the distance between clusters



	C_1	$C_2 \cup C_5$	C_3	C_4
C_1		?		
$C_2 \cup C_5$	?	?	?	?
C_3		?		
C_4		?		

distance/proximity matrix



universitätbonn
RHEINISCHE FRIEDRICH-WILHELMS-UNIVERSITÄT

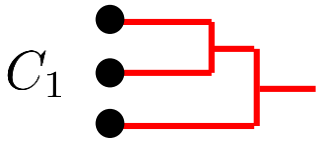
Fraunhofer IAIS

Distance between Two Clusters

each cluster is a set of points

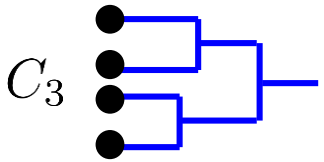
Q: How do we define distance between two sets of points?

A: lots of alternatives . . .



Merge which pair of clusters?

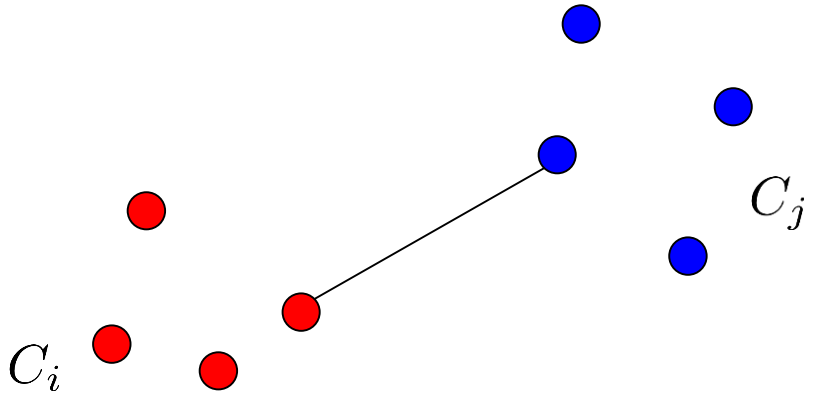
- Depends on the choice of the inter-cluster distance!



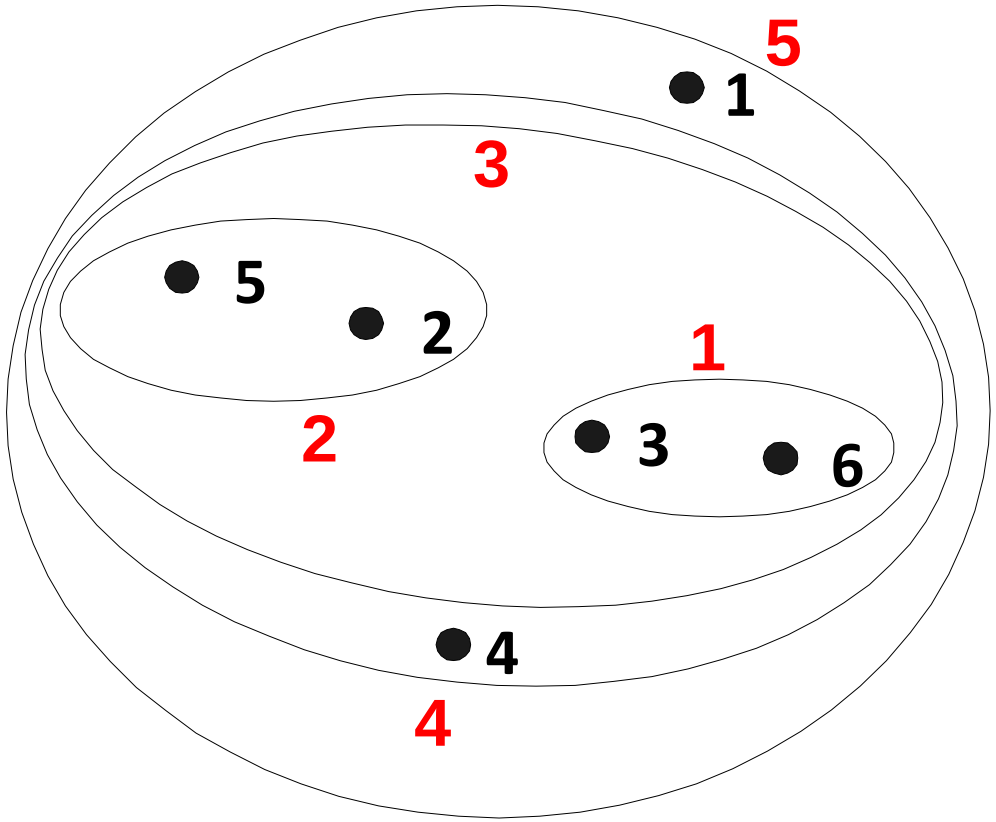
1. Inter-Cluster Distances: Single-Link Distance

single-link distance between two clusters: **minimum** distance between the members of two cluster, i.e.,

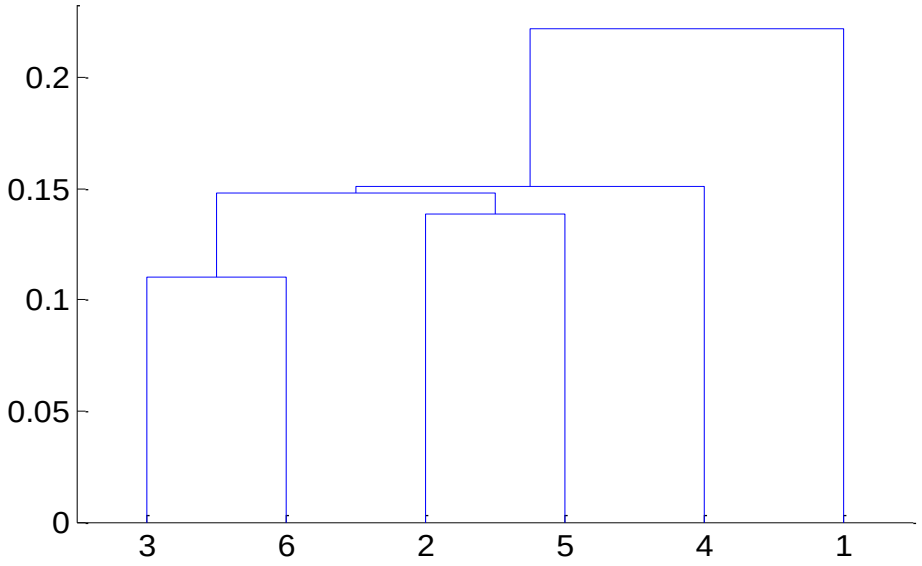
$$d_{\mathrm{SL}}(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y)$$



Single-Link Clustering: Example



nested clusters

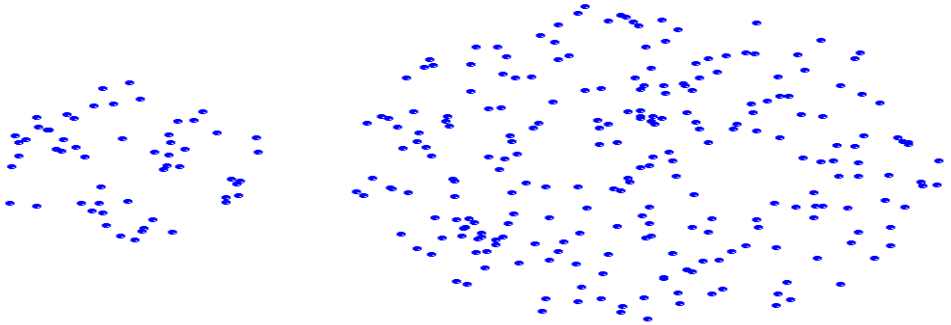


dendrogram

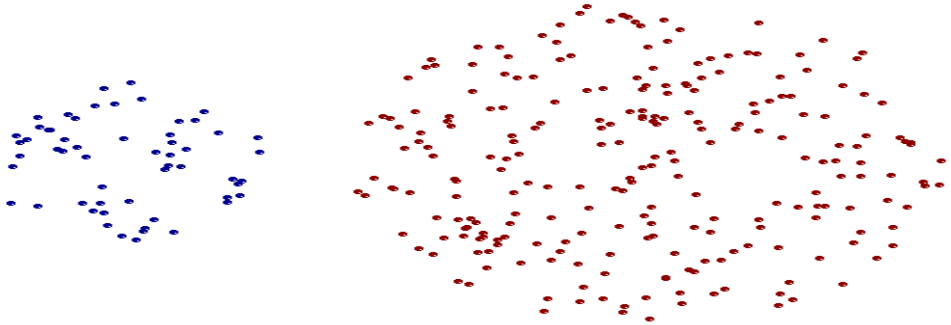
universitätbonn
RHEINISCHE FRIEDRICH-WILHELMS-UNIVERSITÄT

Fraunhofer IAIS

Strengths of Single-Link Clustering



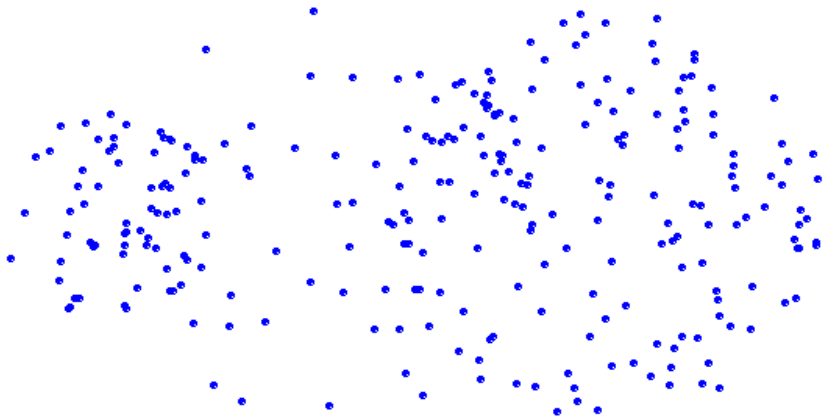
original points



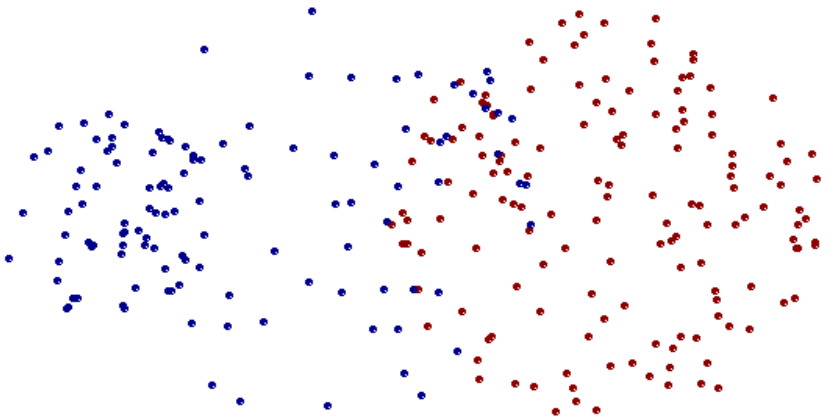
two clusters

☺ Can handle non-elliptical shapes!

Limitations of Single-Link Clustering



original points



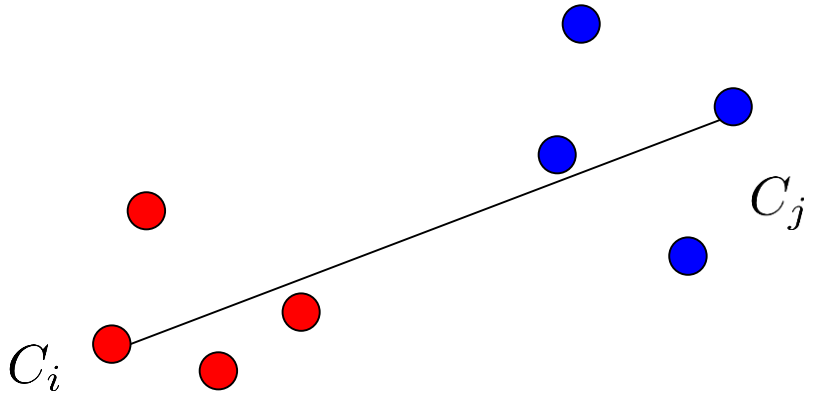
two clusters

- ☹ sensitive to noise and outliers
- ☹ elongated clusters

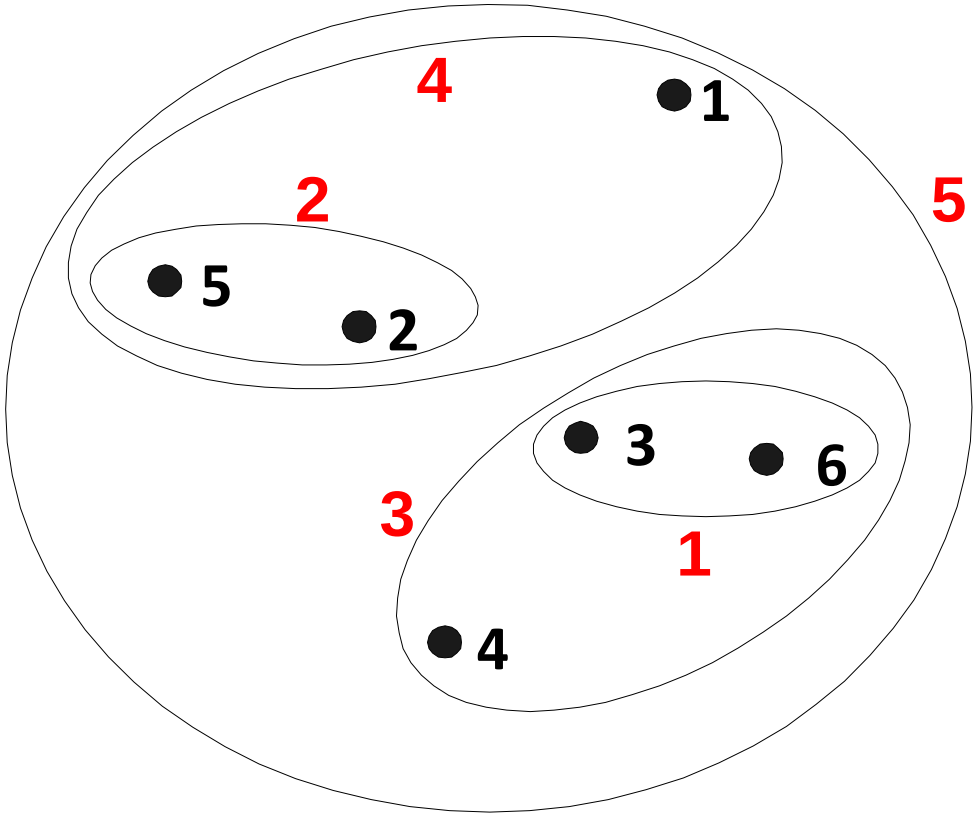
2. Inter-Cluster Distances: Complete-Link Distance

complete-link distance between two clusters: **maximum** distance between the members of two cluster, i.e.,

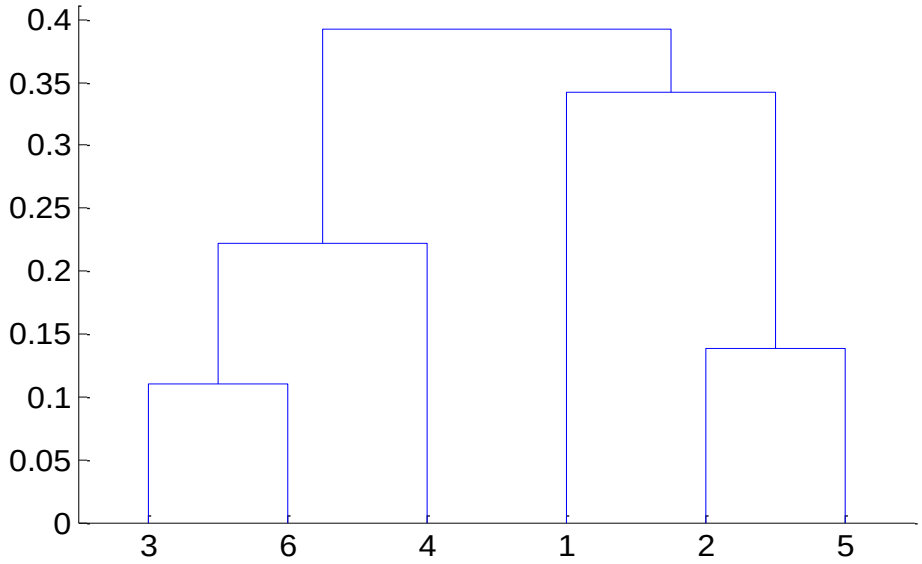
$$d_{\mathrm{CL}}(C_i, C_j) = \max_{x \in C_i, y \in C_j} d(x, y)$$



Complete-Link Clustering: Example

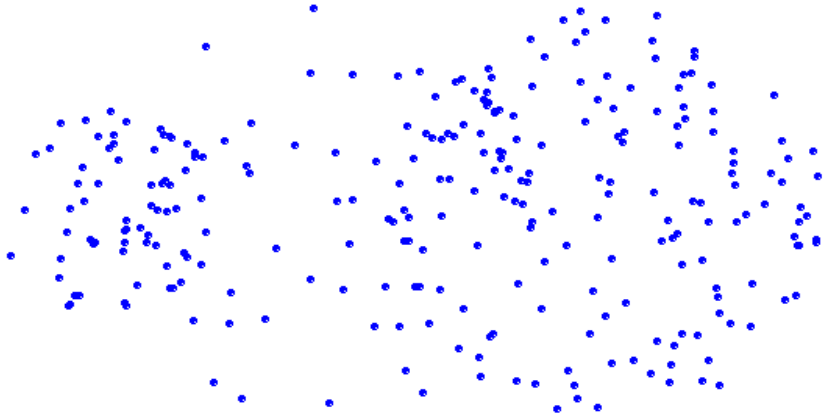


nested clusters

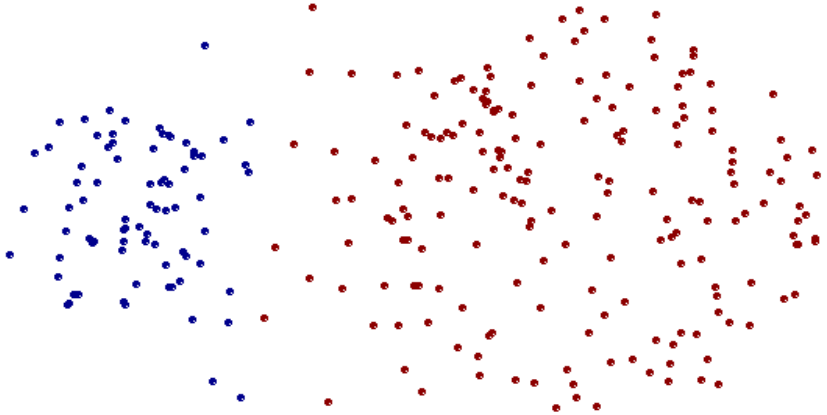


dendrogram

Strengths of Complete-Link Clustering



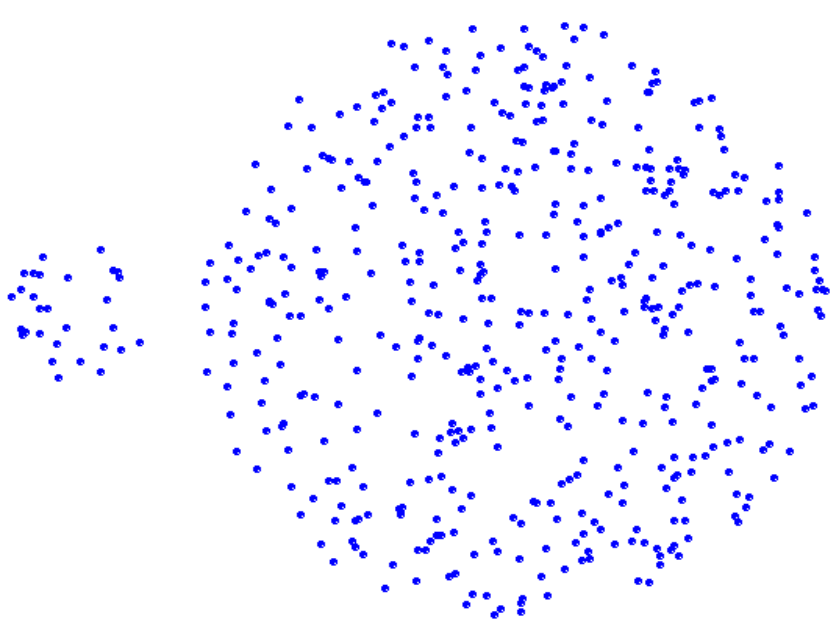
original points



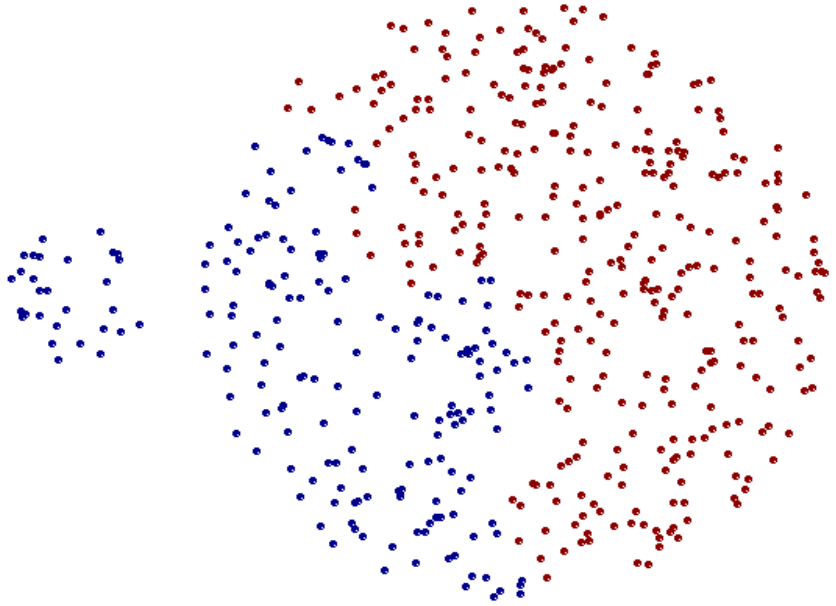
two clusters

- ☺ more balanced clusters (with equal diameter)
- ☺ less sensitive to noise

Limitations of Complete-Link Clustering



original points



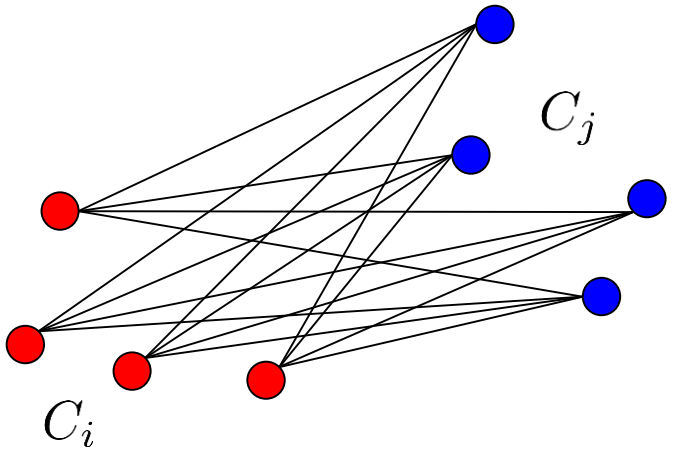
two clusters

- ☹ tends to break large clusters
- ☹ all clusters tend to have the same diameter
  - small clusters are merged with larger ones

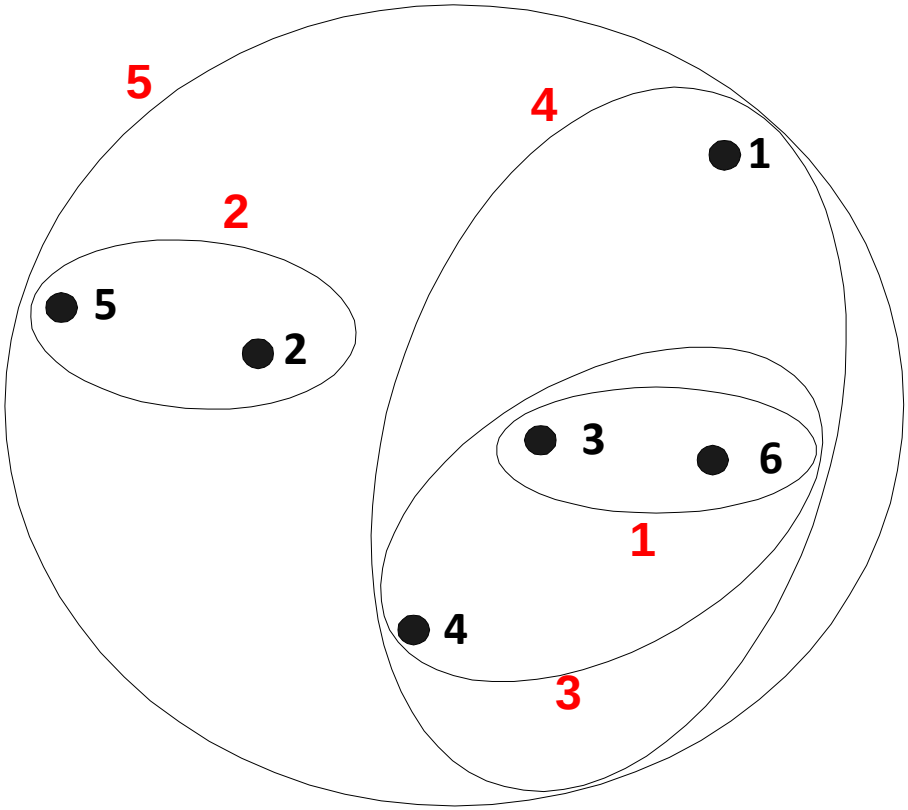
3. Inter-Cluster Distances: Average-Link Distance

average-link distance between two clusters: **averaged** distances of all pairs of objects (one from each cluster), i.e.,

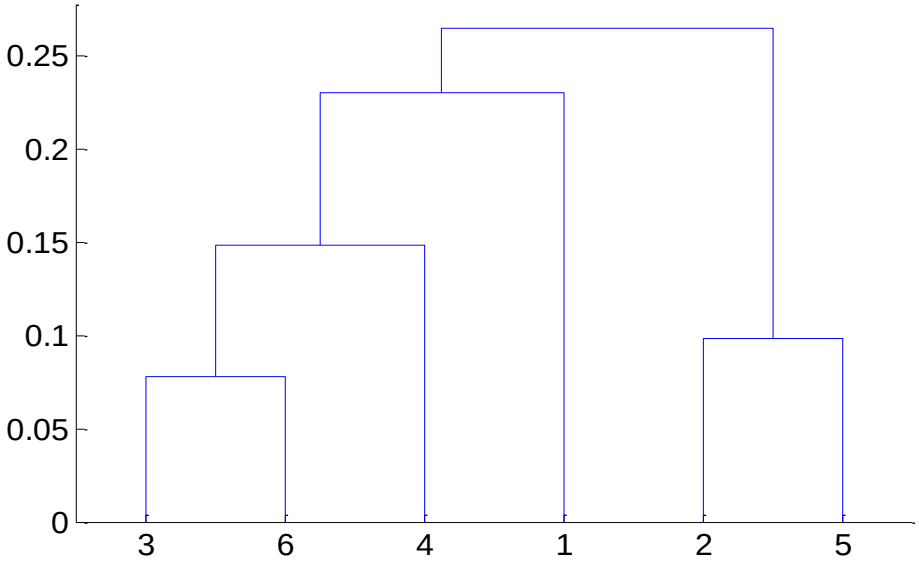
$$d_{\mathrm{AL}}(C_i, C_j) = \frac{1}{|C_i| \cdot |C_j|} \sum_{x \in C_i, y \in C_j} d(x, y)$$



Average-Link Clustering: Example



nested clusters



dendrogram

universitätbonn
RHEINISCHE FRIEDRICH-WILHELMS-UNIVERSITÄT

Fraunhofer IAIS

Average-link Clustering: Discussion

- compromise between single and complete link

☺ strengths

- less susceptible to noise and outliers

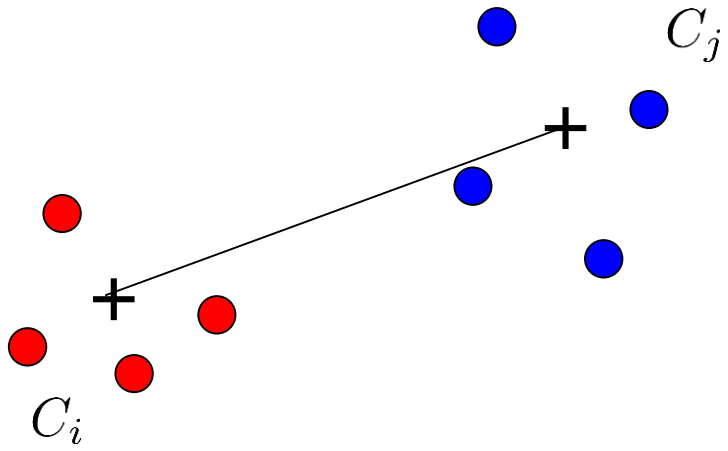
☹ limitations

- may cause elongated clusters to split and for portions of neighboring elongated clusters to merge

4. Inter-Cluster Distances: Centroid Distance

centroid distance between two clusters: distance between two cluster centroids, i.e.,

$$d_{\mathrm{C}}(C_i, C_j) = d\left(\frac{\sum_{x \in C_i} x}{|C_i|}, \frac{\sum_{x \in C_j} x}{|C_j|}\right)$$



5. Inter-Cluster Distances: Ward's Distance

Ward's distance between clusters C_i and C_j:

difference between the total within cluster sum of squares for the two clusters separately, and the within cluster sum of squares resulting from merging the two clusters in cluster $C_{ij} = C_i \cup C_j$, i.e.,

$$d_W(C_i, C_j) = \sum_{x \in C_i} d^2(x - r_i) + \sum_{x \in C_j} d^2(x - r_j) - \sum_{x \in C_{ij}} d^2(x - r_{ij})$$

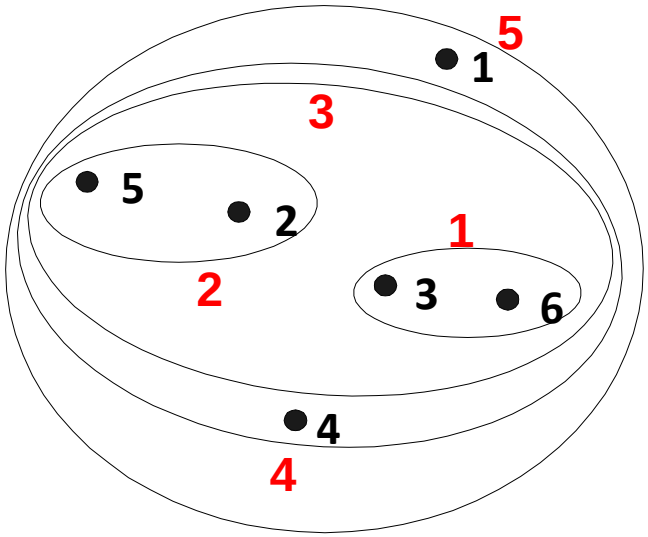
where

- r_i: centroid of C_i
- r_j: centroid of C_j
- r_{ij}: centroid of C_{ij}

Ward's distance for clusters

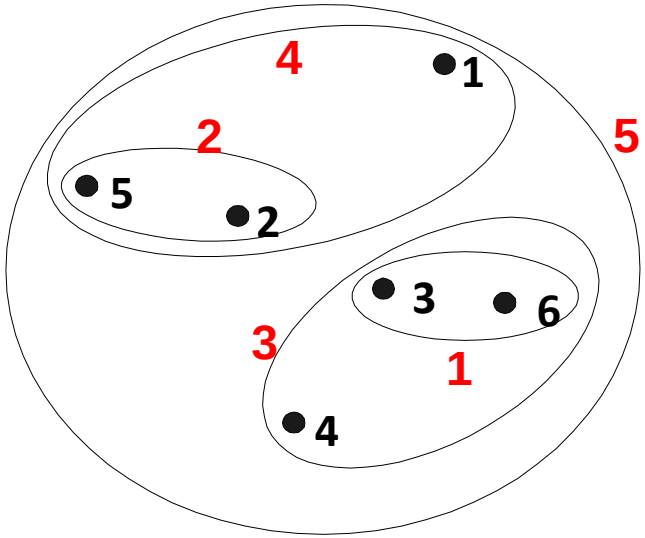
- similar to group average and centroid distance
- less susceptible to noise and outliers
- hierarchical analogue of k-means
  - can be used to initialize k-means

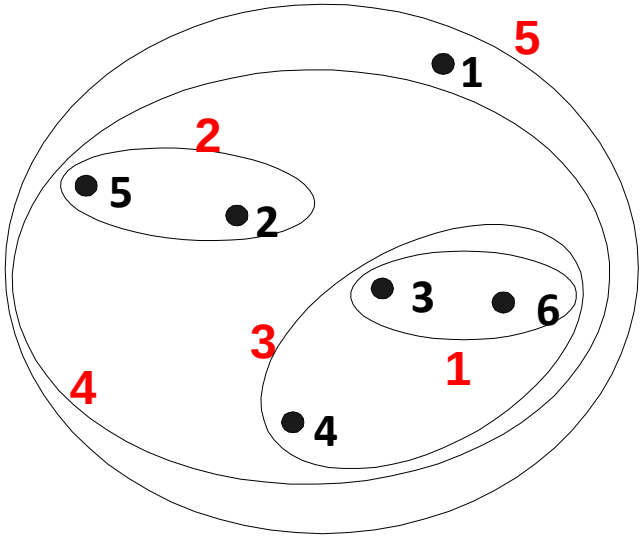
Hierarchical Clustering: Comparison



single-link

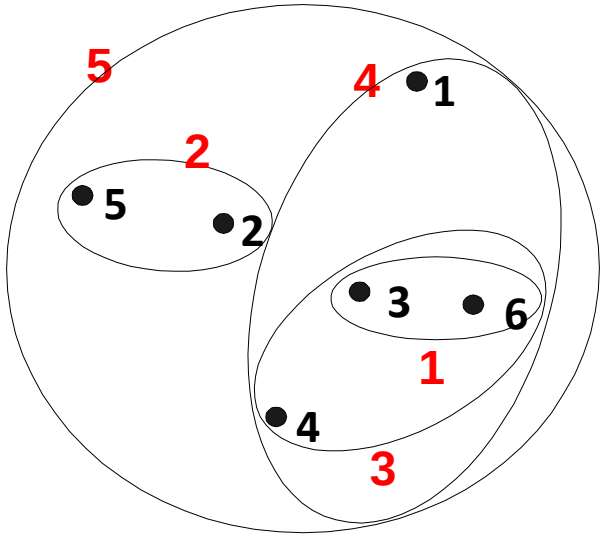
complete-link





average-link

Ward's



universitätbonn
RHEINISCHE FRIEDRICH-WILHELMS-UNIVERSITÄT

Fraunhofer IAIS

Inter-Cluster Distances

general case: various other inter-cluster distance definitions can be used

general agglomerative clustering algorithm: iteratively update a distance matrix

- for distance function d on the instances and inter-cluster distance f, start the agglomerative clustering with the distance matrix

$$D = (d(x_i, x_j))_{1 \leq i \leq, j \leq n}$$

  - set of instances: $I = \{x_1, \ldots, x_n\}$
  - upper triangular, as distances are symmetric

- merging clusters C_i and C_j: merge the corresponding rows and columns in the matrix and update the values according to f

Matrix Update: Example

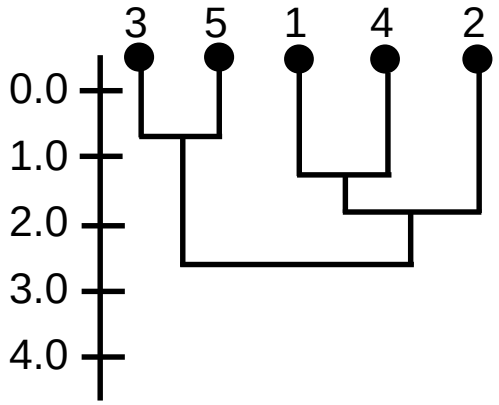
Single link

	1,4	2	3,5
1,4	0	(1,8)	3,4
2		0	2,6
3,5			0

D^3

	1,2,4	3,5
1,2,4	0	2,6
3,5		0

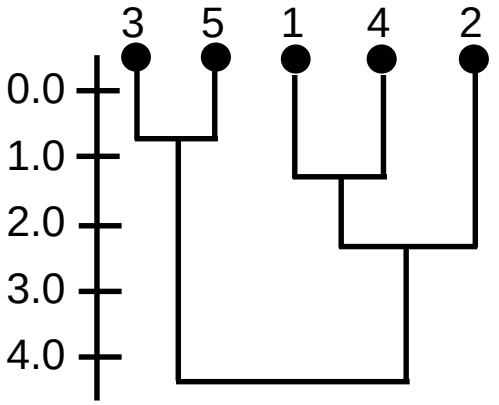
D^4



Complete link

	1,4	2	3,5
1,4	0	(2,3)	4,4
2		0	4,6
3,5			0

	1,2,4	3,5
1,2,4	0	4,6
3,5		0



Hierarchical Clustering: Time and Space Requirements

for a dataset I consisting of n instances (points)

- $O(n^2)$ **space**
  - it requires storing the distance matrix
- $O(n^3)$ **time** in most of the cases
  - complexity can be reduced to $O(n^2 \log n)$ time for some approaches by using appropriate data structures

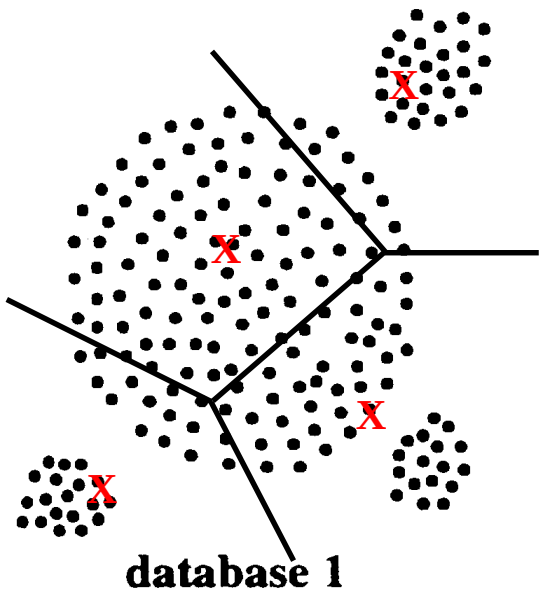
Discussion

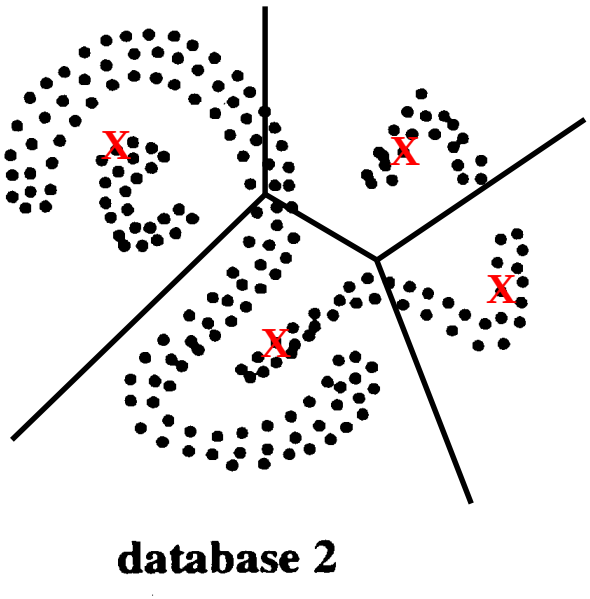
- weighted/unweighted:
  - **unweighted:** each instance counted equally
  - **weighted:** each cluster counted equally, i.e. instances weighted by cluster size
- only needs distance matrix
  - can work with specialized distances
- user can and has to pick granularity
- nice visualization (for small datasets)
- not very suitable for very large instance sets
- shapes of clusters very much depend on chosen method, no clear cluster "model" as in K-means

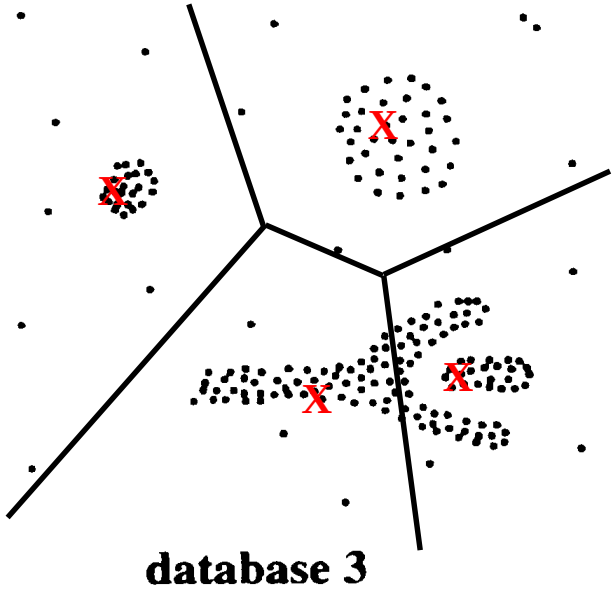
Density-Based Clustering

Clustering spatial data

- distance-based clustering is inherently spatial
- assumption of convex clusters is inappropriate for many practical problems, e.g., for “geographical” tasks







Density-based Clustering

- DBSCAN – **D**ensity-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise
  - (M. Ester, H.P. Kriegel, J. Sander and X. Xu., 1996)
  - density-based clustering *algorithm*

idea:

locate regions of high density that are separated from one another by regions of low density

- **density** = number of points within a specified radius

features:

- easily allows noisy points to be ignored
- requires definition of “density” in the “neighborhood” of a point or cluster

DBSCAN – Concepts: Core Points

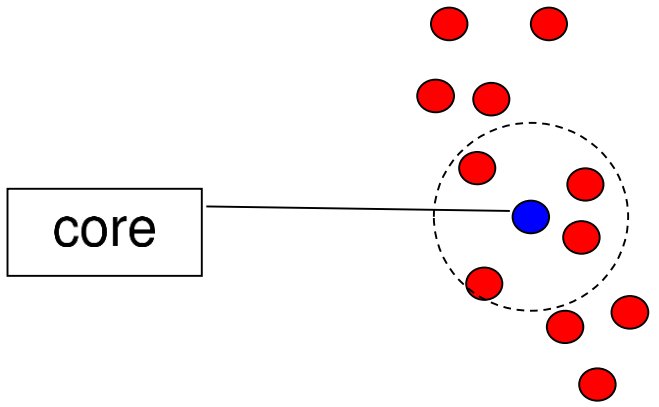
$I = \{x_1, \ldots, x_n\}$: set of instances; two parameters $\epsilon \in \mathbb{R}^+$ and MinPts $\in \mathbb{N}$

Def.: the ϵ**-neighborhood** of a point $x \in I$ is the set

$$N_\epsilon(x) = \{y \in I : d(x, y) \leq \epsilon\}$$

Def.: $x \in I$ is a **core point** if it has at least MinPts instances within its ϵ-neighborhood, i.e.,

$$|N_\epsilon(x)| \geq \text{MinPts}$$



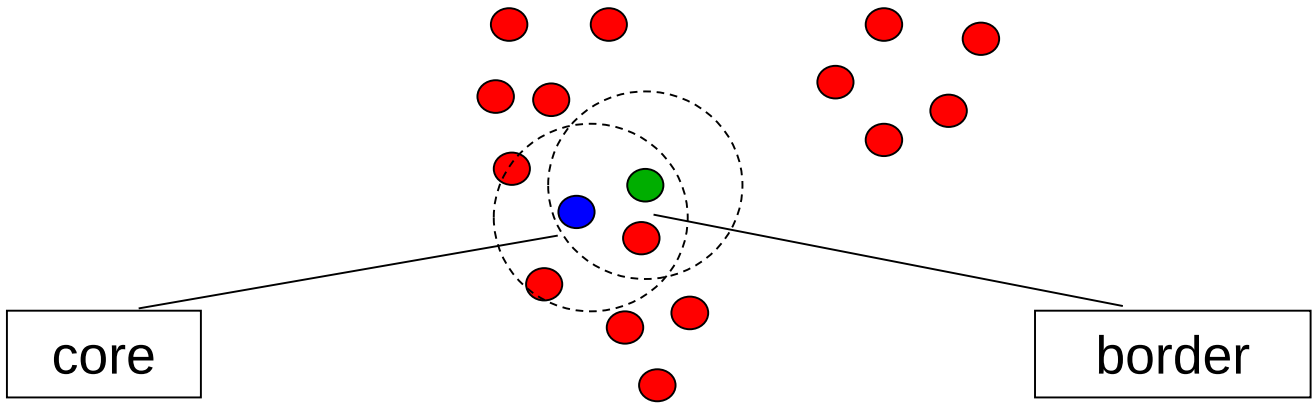
$\epsilon = 1$ unit; MinPts $= 5$

DBSCAN – Concepts: Border Points

$I = \{x_1, \ldots, x_n\}$: set of instances; two parameters $\epsilon \in \mathbb{R}^+$ and MinPts $\in \mathbb{N}$

Def.: $x \in I$ is a **border point** if its number of ϵ-neighbors is less than MinPts and it is the neighbor of a core point, i.e.,

$|N_\epsilon(x)| <$ MinPts and there exists a core point c such that $x \in N_\epsilon(c)$



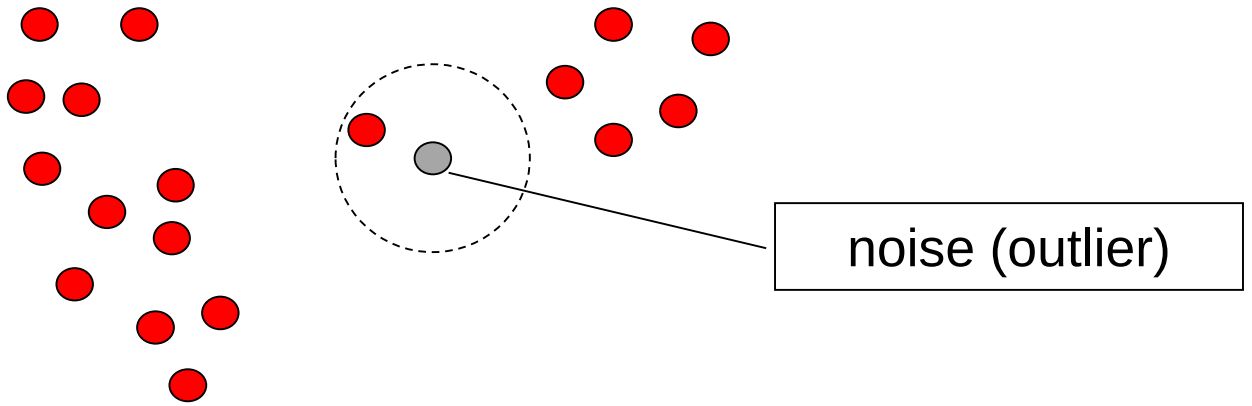
$\epsilon = 1$ unit; MinPts $= 5$

DBSCAN – Concepts: Noise Points

$I = \{x_1, \ldots, x_n\}$: set of instances; two parameters $\epsilon \in \mathbb{R}^+$ and MinPts $\in \mathbb{N}$

Def.: $x \in I$ is a **noise point** if it is neither a core nor a border point, i.e.,

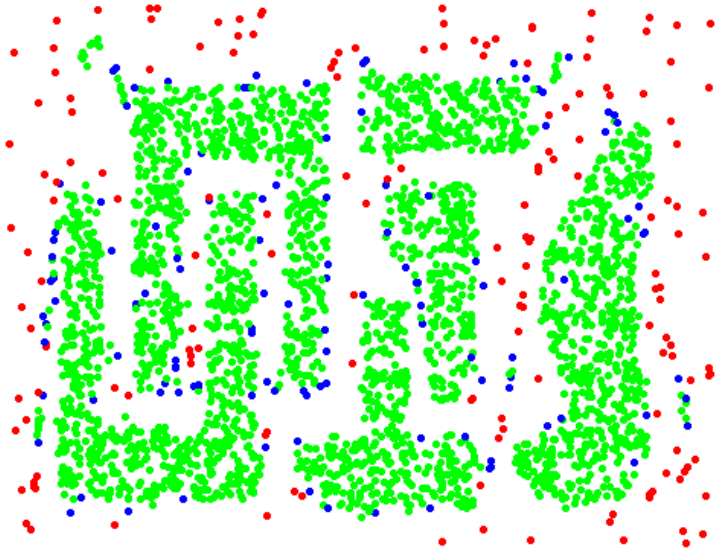
$|N_\epsilon(x)| <$ MinPts and there is **no** core point c such that $x \in N_\epsilon(c)$



$\epsilon = 1$ unit; MinPts $= 5$

Example: Core, Border, and Noise Points

original points



point types: core, border and noise

$$\epsilon = 10; \text{MinPts} = 4$$

DBSCAN: Cluster Assignment

- core and border points: lie in the **interioir** of a cluster
- noise points: do not belong to any of the clusters

DBSCAN:

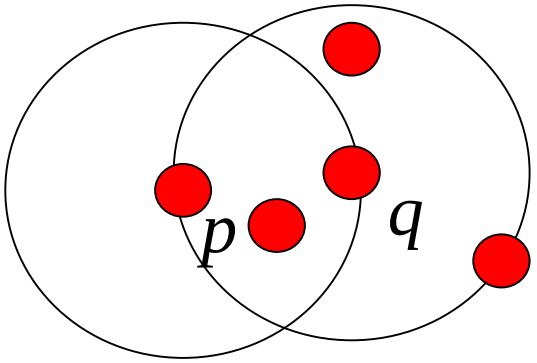
- if any two **core** points are within the ϵ-neighborhood of one another then put them in the same cluster
- if a **border** point is in the ϵ-neighborhood of a core point then put it in the same cluster as the core point
  - not necessarily unique assignment
- discard all **noise** points

DBSCAN – Concepts: Direct Density-Reachability

Def.: an instance $p \in I$ is **directly density-reachable** from $q \in I$ w.r.t. ϵ and MinPts if p is within the ϵ-neighborhood of q and q is a core object, i.e.,

- $p \in N_\epsilon(q)$
- $|N_\epsilon(q)| \geq$ MinPst

Not symmetric!

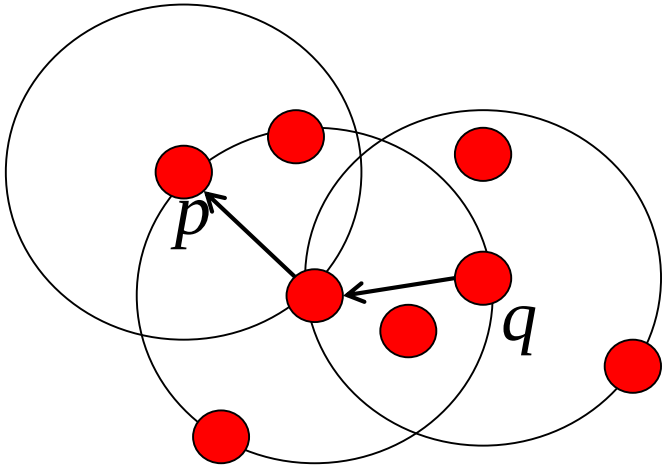


for $\epsilon =$ unit length and MinPts $= 4$:

- p is directly density-reachable from q
- q is **not** directly density-reachable from p

DBSCAN – Concepts: Density-Reachability

Def. (density-reachability): transitive closure of direct densitive-reachability, i.e., $p \in I$ is **density-reachable** from $q \in I$ w.r.t ϵ and MinPts if there is a chain $q_1, \ldots, q_n \in I$ with $q_1 = q$, $q_n = p$ such that $q_{i+1}$is directly density-reachable from q_i w.r.t ϵ and MinPts for all $i = 1, \ldots, n-1$

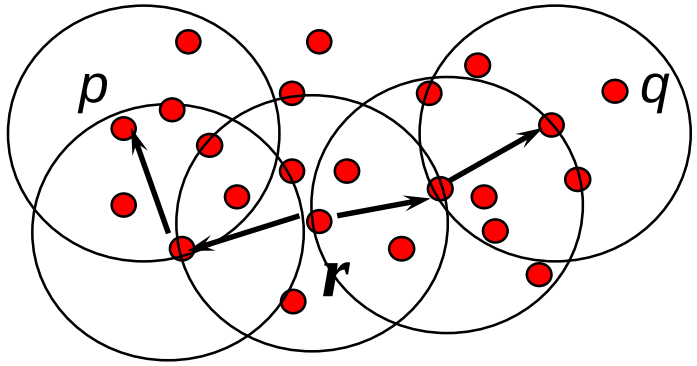


for $\epsilon =$ unit length and MinPts $= 4$:

- p is density-reachable from q
- q is **not** density-reachable from p

DBSCAN – Concepts: Density-Connectivity

Def.: $p \in I$ is **density-connected** to $q \in I$ w.r.t ϵ and MinPts if there is an instance $r \in I$ such that both p and q are density-reachable from r w.r.t ϵ and MinPts



for $\epsilon =$ unit length and MinPts $= 4$:

- p and q are density-connected to each other by r
- density-connectivity is symmetric

DBSCAN – Concepts: Cluster and Noise

Def. (density-connected sets): $C \subseteq I$ is density-connected w.r.t ϵ and MinPts if C is non-empty and it satisfies

(i) **maximality:** for all $p, q \in I$, if $q \in C$ and p is density-reachable from q w.r.t ϵ and MinPts, then $p \in C$

(ii) **connectivity:** for all $p, q \in C$, p is density-connected to q w.r.t ϵ and MinPts in I

Def. (cluster): density-connected set

- cluster contains core as well as border points

Def. (noise): non-core points that are not directly density-reachable from at least one core point

Properties

(1) Let $p \in I$ be a core point w.r.t. ϵ and MinPts. Then

$$\{o \in I | o \text{ is density reachable from } p \text{ w.r.t. } \epsilon \text{ and MinPts}\}$$

is a density-connected set.

(2) Let C be a density-connected set and $p \in C$ be a core point, both w.r.t. ϵ and MinPts. Then

$$C = \{o \in I | o \text{ density reachable from } p \text{ w.r.t. } \epsilon \text{ and MinPts}\}$$

proof: *exercise*

$\Rightarrow$ Density-connected sets can be constructed from any core object by going to all reachable objects!

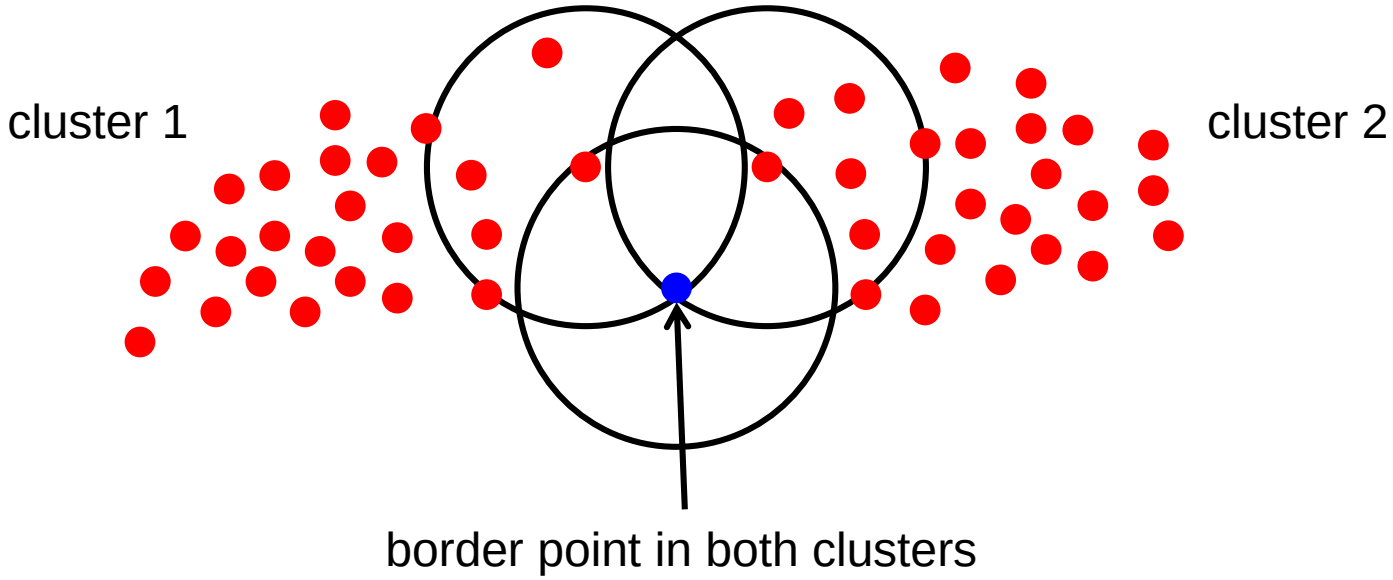
$\Rightarrow$ give rise to algorithm

Density-Connected Clustering

a density-connected clustering of a set I of instances w.r.t. ϵ and MinPts is the set of all density-connected sets w.r.t. ϵ and MinPts in I

- non-covered points are noise (noise$_{CL}$)
- clusters can overlap only in border points
- exactly one clustering, i.e., deterministic (up to shared border points)
- hard cluster membership constraint
  - ambiguous border points are assigned to the first cluster during the algorithm

Overlapping Clusters



Algorithm DBSCAN

input: set I of instances, ϵ, and MinPts

1. $\mathcal{C} = \emptyset$
2. **for** each unvisited point $p \in I$ **do**
3. mark p as visited
4. **if** $|N_\epsilon(p)| <$ MinPts **then** mark p as noise point
5. **else**
6. $C = \emptyset$ // next cluster
7. EXPANDCLUSTER$(p, N_\epsilon(p), C, \epsilon,$ MinPts$)$ // next slide
8. add C to $\mathcal{C}$
9. **return** $\mathcal{C}$

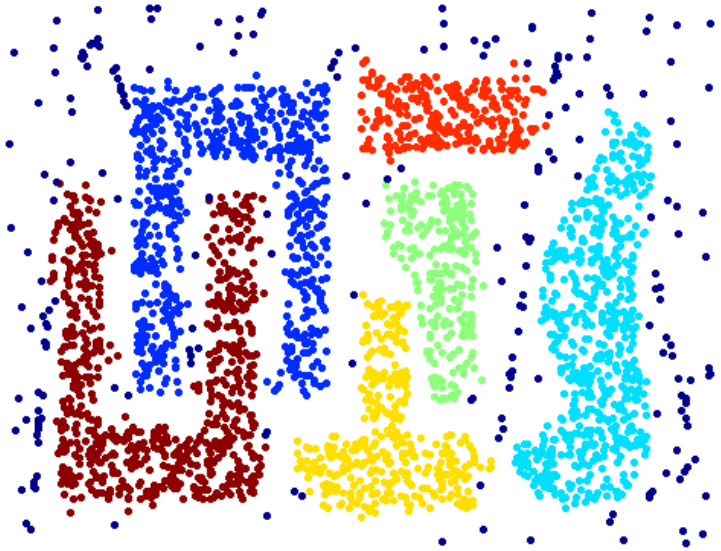
Procedure ExpandCluster

input: $p \in I$, $N \subseteq I$, $C = \emptyset$, ϵ, and MinPts

1. add p to cluster C
2. **forall** $q \in N$ **do**
3. **if** q is unmarked **then**
4. mark q
5. **if** $|N_\epsilon(q)| \geq$ MinPts **then** $N = N \cup N_\epsilon(q)$
6. **if** q has not yet been added to any cluster
7. add q to cluster C

When DBSCAN Works Well

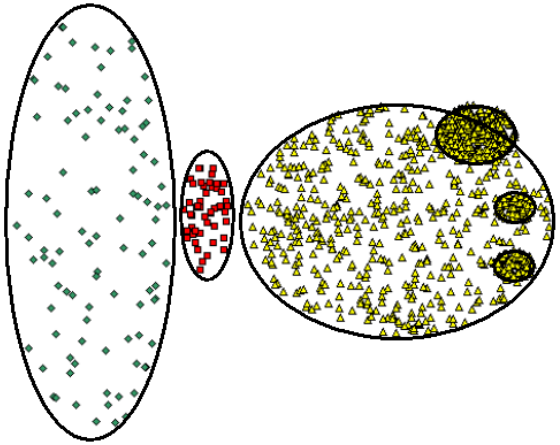
original points



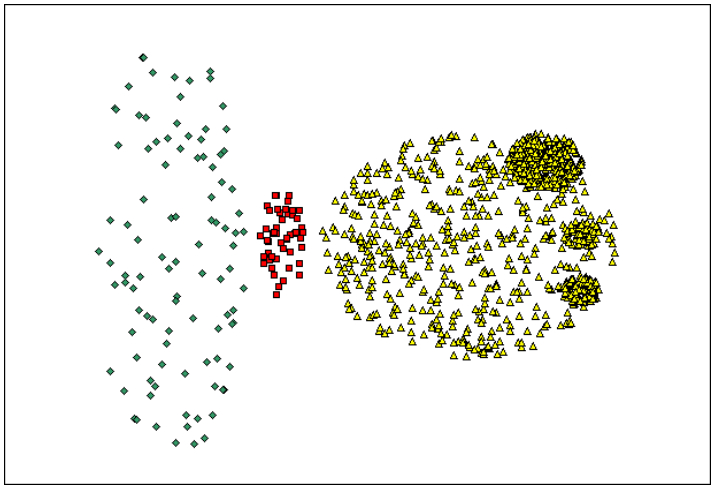
clusters

- resistant to noise
- can handle clusters of different shapes and sizes

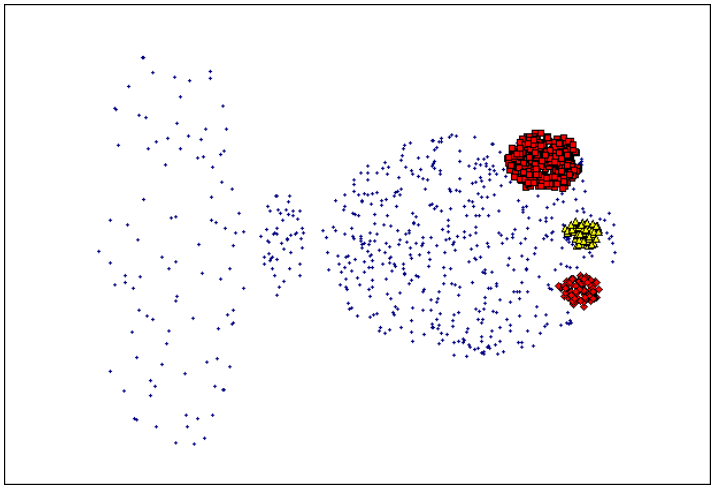
When DBSCAN Does Not Work Well



original points



MinPts = 4; ϵ = large value



MinPts = 4; ϵ = small value
min density increases

- varying densities
- high-dimensional data
  - holds for K-means as well

DBSCAN: Determining the Parameters ε and MinPts

idea: for $k = 1, 2, \ldots$:

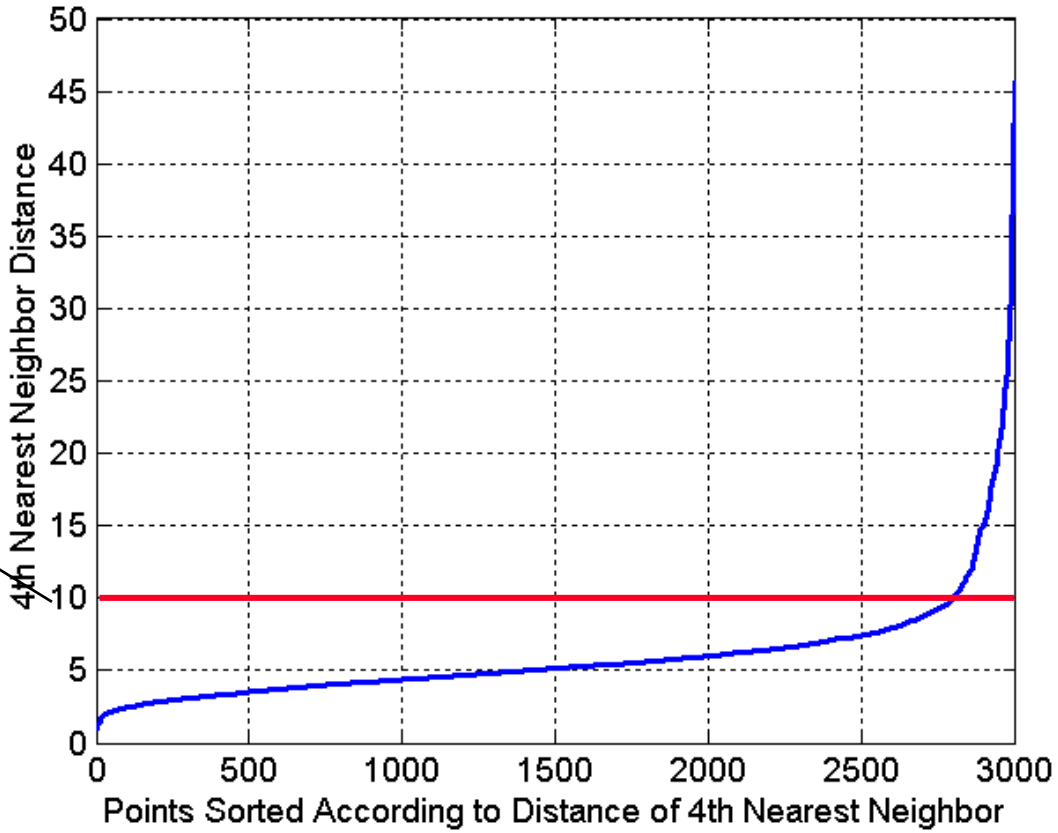
- k**-distance**: distance between a point and its k-th nearest neighbor
  - k-distance for points within a cluster: roughly the same
  - k-distance for noise points is **relatively large**

heuristic:

1. compute k-distance for all points for some k
2. sort them in increasing order and plot sorted values
3. a sharp change at the value of k-distance that corresponds to suitable value of ϵ for MinPts = k

DBSCAN: Determining ε

$$\Rightarrow \epsilon = 10$$



universitätbonn
RHEINISCHE FRIEDRICH-WILHELMS-UNIVERSITÄT

Fraunhofer IAIS

Complexity

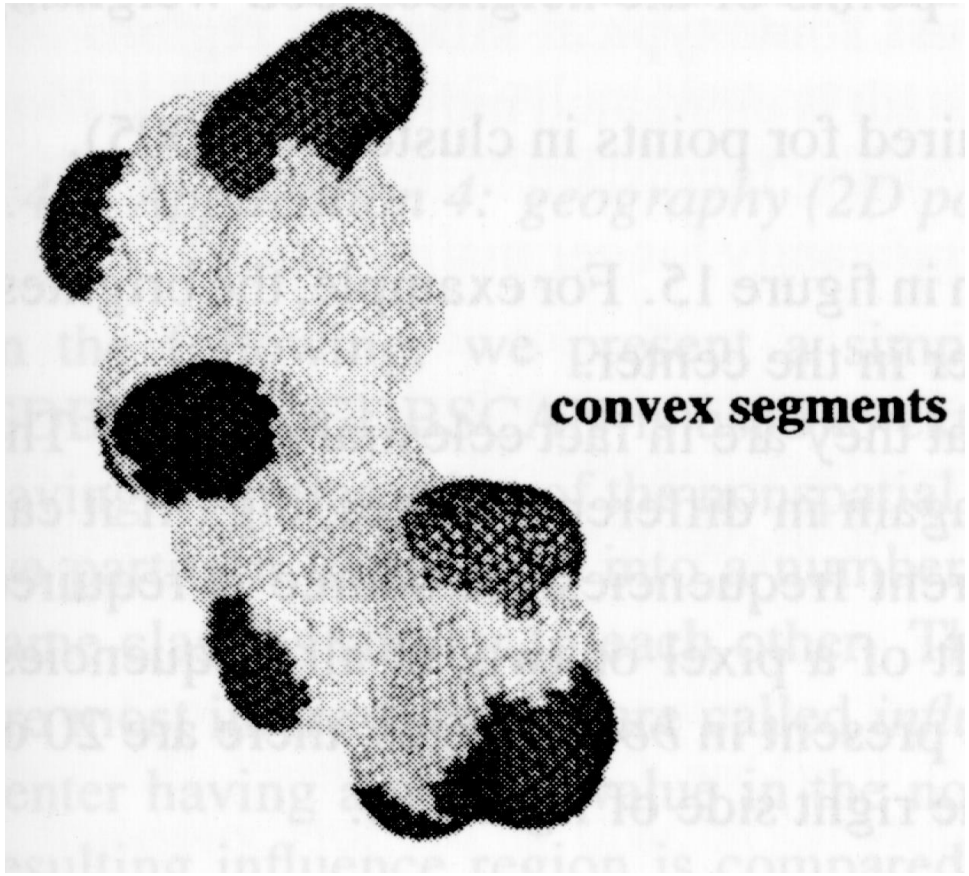
- **complexity:** $O(n \cdot T(N_\epsilon))$
  - $T(N_\epsilon)$: runtime of computing neighbors
- efficient neighborhood queries in SDBS (spatial DBS)
  - use data structure R-tree (multidimensional B-tree for spatial data)
  - $\log n$ access

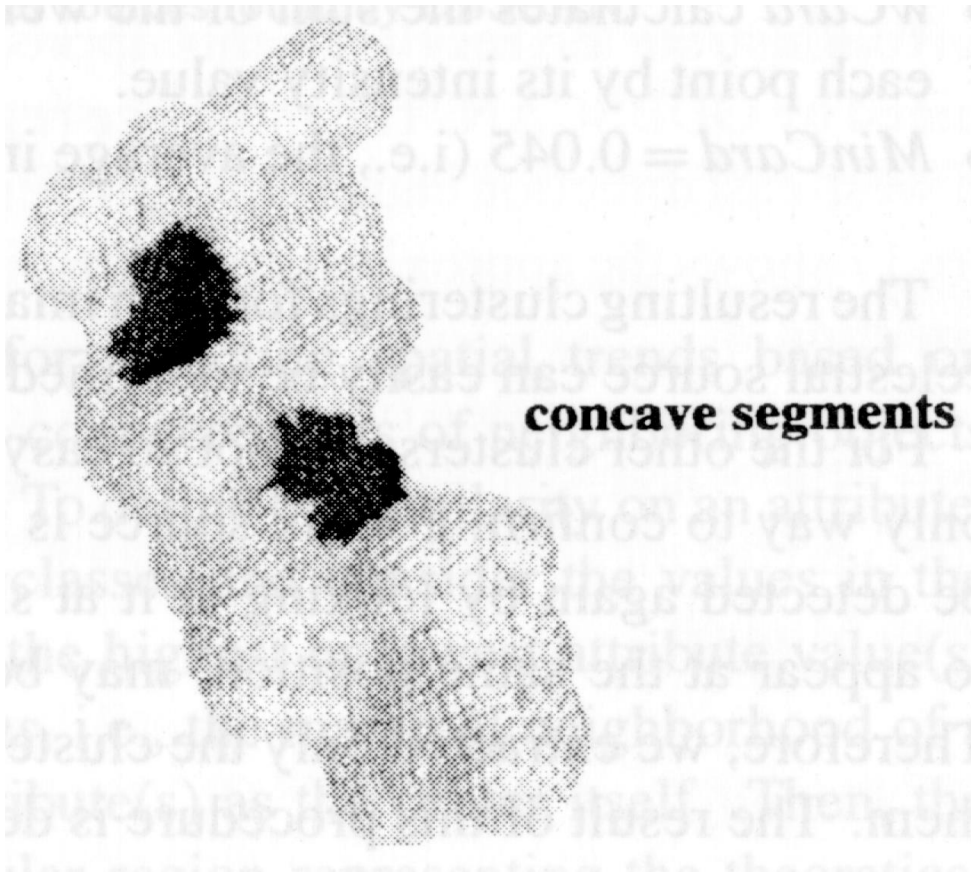
  $\Rightarrow$ $n \log n$

Applications

- 5-D multispectral satellite images of California
- molecular biology: atom coordinates of proteins (special wCard function!)
- astronomy: detecting radio sources in 2D intensity images
- geography: influence regions of cities

Proteins





DBSCAN: Summary

- density-based
- able to work with concave clusters
- robust to noisy data
- output clustering is deterministically determined by the two input parameters
- input parameters can be estimated by a simple heuristic
- different practical applications

Clustering: Summary

- partitional clustering
  - finds single partitioning, larger datasets
  - works only where center can be computed, local minima
  - K-means for n-dimensional Euclidean space
  - EM, GMM
- hierarchical distance-based clustering
  - only distance needed, so powerful or custom functions possible
- DBSCAN
  - deterministic, resistant to noise, non-convex shapes
- **no** clustering satisfying all desirable properties at the same time
  - see Appendix

Appendix:

Kleinberg's Impossibility Theorem

(won't be asked in the exam)

The Impossibility Theorem

clustering in general: can be regarded as a function F mapping a set I of points and a distance d over I to a partitioning (clustering) $\mathcal{C}$ of I

three natural properties one would require F to satisfy:

1. scale-invariance
2. richness
3. consistency

1. Scale-Invariance

scale-invariance: F is *not* sensitive to changes in the units of distance measurement, i.e., for any distance function d and any $\alpha > 0$,

$$F(I, d) = F(I, \alpha d)$$

2. Richness

richness: the range of F should be *rich*, i.e., for any partition $\mathcal{C}$ of I, there exists a distance function d such that

$$F(I, d) = \mathcal{C}$$

3. Consistency

consistency: let d, d' be distance functions satisfying the following property:

if

- $\forall C \in F(S,d), \forall x,y \in C$: $d'(x,y) \leq d(x,y)$ and
- $\forall C_i, C_j \in F(S,d)$ with $C_i \neq C_j, \forall x \in C_i, \forall y \in C_j$: $d'(x,y) \geq d(x,y)$

then

$$F(S,d) = F(S,d')$$

- i.e., d' equals d, except for shrinking distances within clusters of $F(S,d)$ or stretching between-cluster distances in $F(S,d)$

The Impossibility Theorem

Thm. (Kleinberg, J., 2002): For each $n \geq 2$, there is **no** clustering function F that satises scale-invariance, richness, and consistency.

- n: cardinality of I