

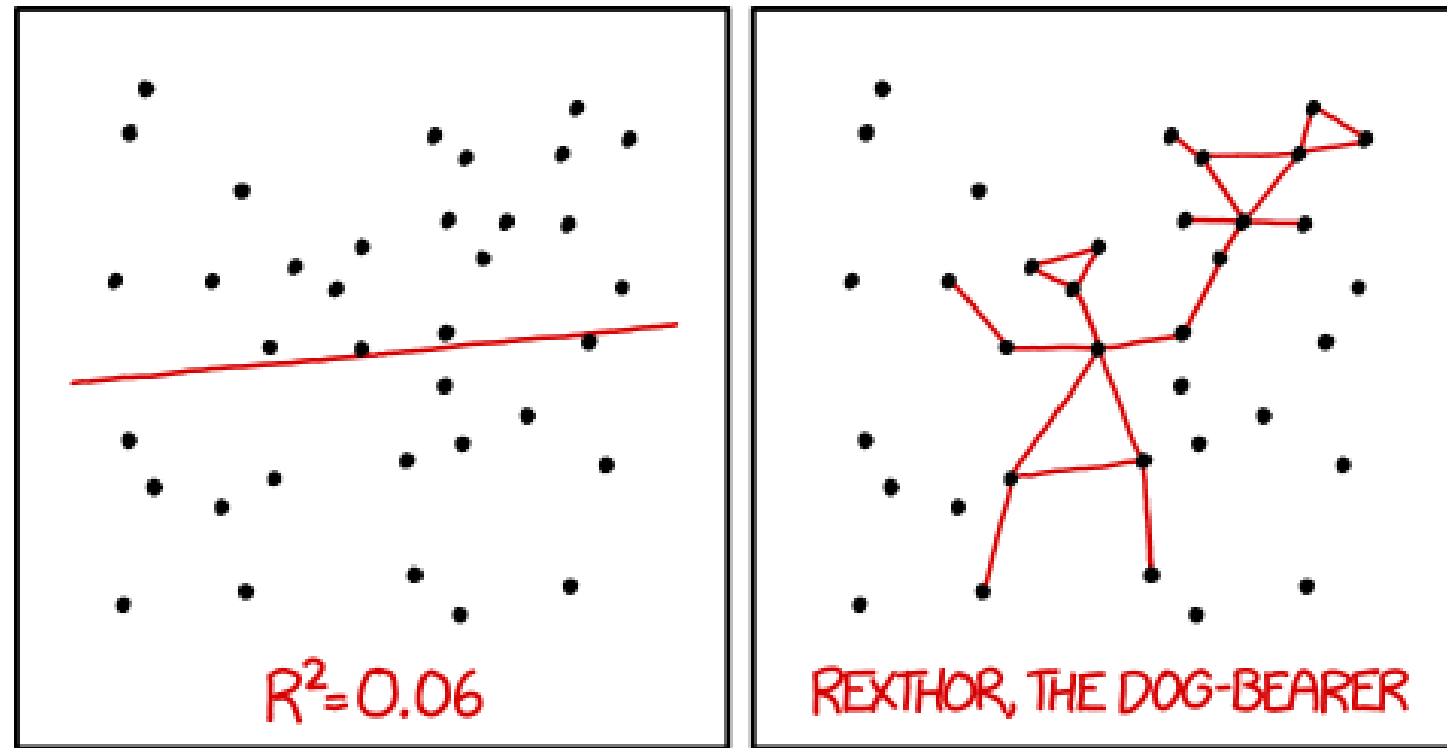


UNIVERSITÄT
LEIPZIG

Data Mining

INTERACTIVE VISUAL DATA MINING

Lineare Regression



I DON'T TRUST LINEAR REGRESSIONS WHEN IT'S HARDER
TO GUESS THE DIRECTION OF THE CORRELATION FROM THE
SCATTER PLOT THAN TO FIND NEW CONSTELLATIONS ON IT.

"The 95% confidence interval suggests Rexthor's dog could also be a cat, or possibly a teapot."

Lineare Regression

- Lineare Modelle
 - Funktionieren gut mit numerischen Werten
 - Numerische Vorhersage: Linear regression
- Vorbedingung: Ergebnis und alle Attribute sind numerisch
- Das Ergebnis kann als Kombination der Attribute ausgedrückt werden:
 - $x = \sum_{i=0}^k w_i a_i$
 - x : Ergebnis
 - a_0 : 1
 - a_i : Wert des Attributs i
 - w_i : Vorberechnete Gewichte

Lineare Regression

- Ziel: Minimiere die folgende Gleichung:

- $\sum_{i=1}^n (x^{(i)} - \sum_{j=0}^k w_j a_j^{(i)})$

- $x^{(i)}$: Ergebnis, Klasse der Instanz i

- $a_0^{(i)} = 1, a_j^{(i)}$: Wert des Attributs j der Instanz i

- w_j : gesuchte Gewichte

- Die Anzahl der Trainingsinstanzen n sollte deutlich größer als die Anzahl der Attribute k sein

- Ergebnis:

- Gewichte, die auf den Trainingsdaten basieren

- Diese Gewichte werden dann für die Vorhersage der Instanz unbekannter Daten genutzt

Lineare Regression

- Eigenschaften:
 - Einfache Methode
 - Wird oft in statistischen Analysen genutzt
 - Nur für Daten mit linearen Abhängigkeiten geeignet
 - Wenn die Daten nicht linear abhängig sind, funktioniert das Verfahren nicht!

Lineare Regression

- Multiresponse linear regression
- Nutzung der linearen Regression auch hier möglich
- Nutzung anderer Regressionstechniken aber auch möglich
- Erstelle eine Regression für jede Instanz:
 - Setze das Ergebnis auf 1 für Trainingsinstanzen, die zu der Klasse gehören
 - Setze das Ergebnis auf 0 für alle anderen
- Das Ergebnis ist ein linearer Ausdruck für die Klasse
- Für eine unbekannte Instanz:
 - Berechne die Werte für alle lineare Ausdrücke
 - Wähle den größten

Lineare Regression

- Multiresponse linear regression
 - Kann als numerische Funktion für Mitgliedschaften zu jeder Klasse aufgefasst werden
 - Bringt gute Ergebnisse in der Praxis
 - Hat zwei Nachteile:
 - Mitgliedschaftswerte können außerhalb des $[0, 1]$ Intervalls liegen und sind somit keine “ordentlichen” Wahrscheinlichkeiten
- Least-squares Regression erwartet, dass die Fehler
 - Statistisch unabhängig sind
 - Normalverteilt sind mit der gleichen Standardabweichung
- Diese Annahme wird aber verletzt, wenn man sie für Klassifikationsprobleme verwendet
 - Beobachtungswerte können nur zwischen 0 und 1 liegen

Lineare Regression

- Logistische Regression

- Baut ein lineares Modell auf einer transformierten Zielvariable

- Ersetze $\Pr[1 | a_1, \dots, a_k]$ mit $\frac{\log(\Pr[1 | a_1, \dots, a_k])}{1 - \Pr[1 | a_1, \dots, a_k]}$

- Die Ergebniswerte liegen zwischen $[-\infty, \infty]$

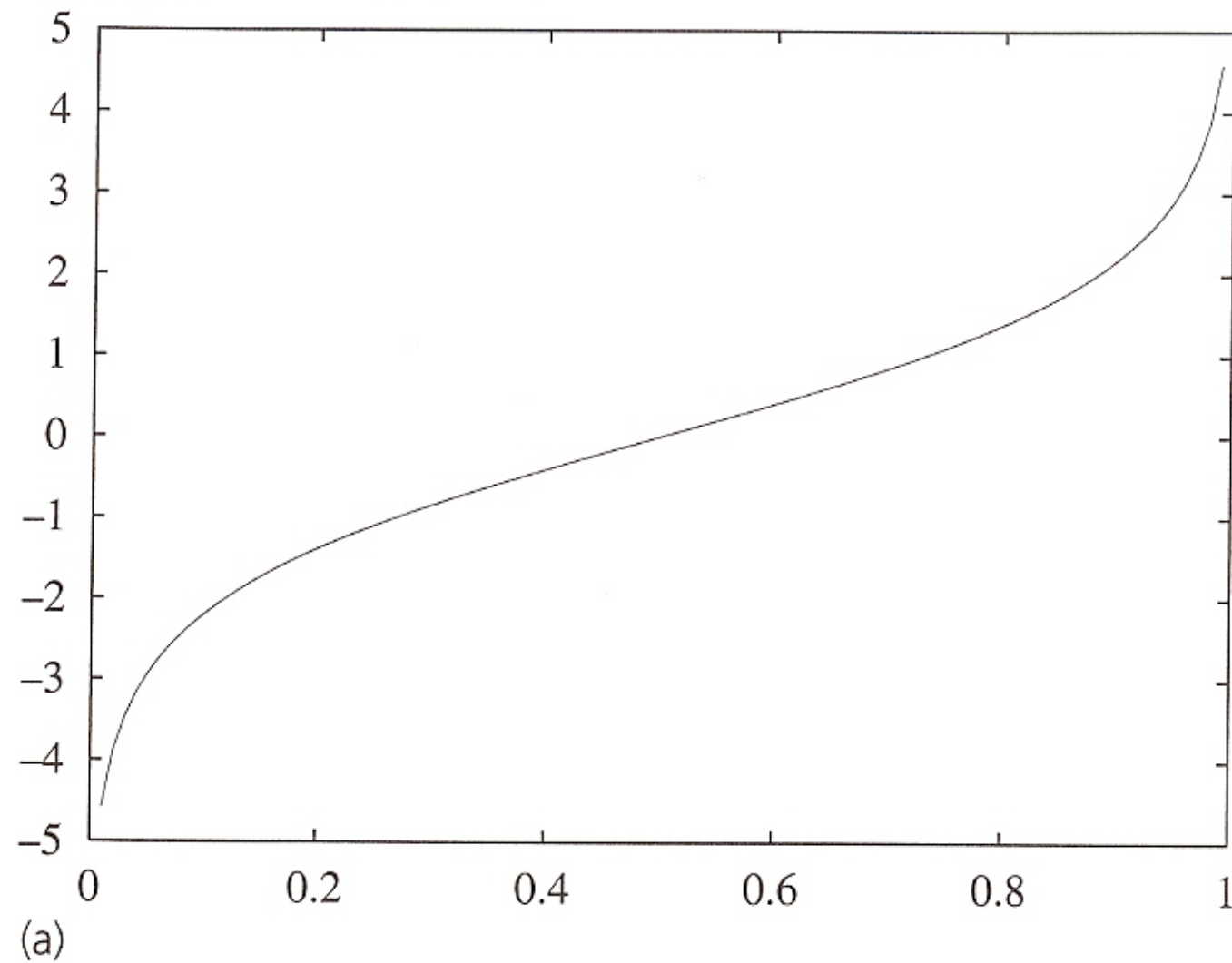
- Transformation wird auch logit transformation genannt (siehe Bild auf der nächsten Folie)

- Die transformierte Variable wird wie zuvor durch eine lineare Funktion approximiert

- Resultierendes Modell:

$$\Pr[1 | a_1, \dots, a_k] = \frac{1}{1 + e^{-\sum w_i \cdot a_i}}$$

Lineare Regression



Lineare Regression

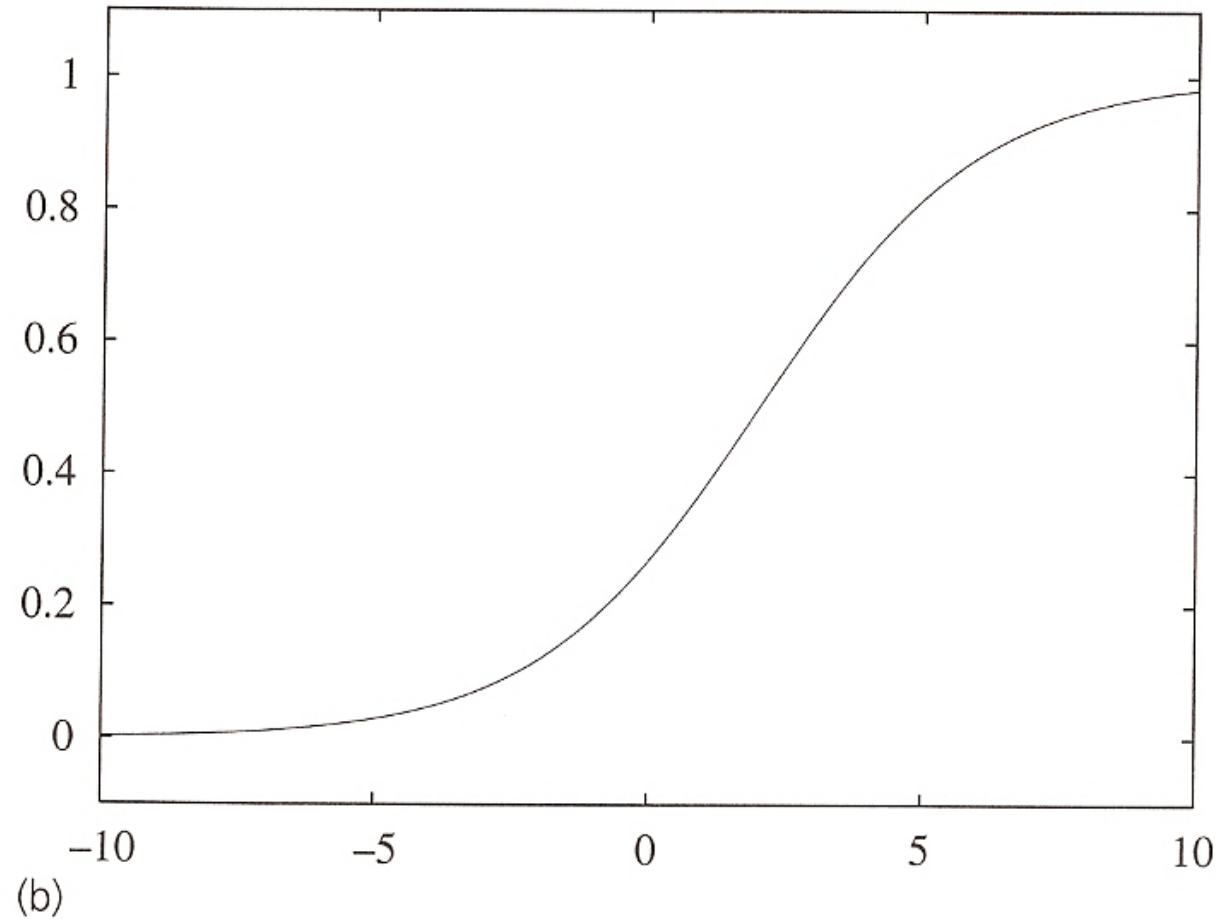


Figure 4.9 Logistic regression: (a) the logit transform and (b) an example logistic regression function.

Lineare Regression

- Logistische Regression

- Finde Gewichte , die zu den Trainingsdaten passen

- Nutzt die *log-likelihood* des Modells

- $\sum_{i=1}^n (1 - x^{(i)}) \log(1 - \Pr[1|a_1^{(i)}, \dots, a_k^{(i)}]) + x^{(i)} \log(\Pr[1|a_1^{(i)}, \dots, a_k^{(i)}])$

- $x^{(i)} \in \{0, 1\}$

- Die Gewichte w_i werden so gewählt, dass die log-likelihood maximiert wird

- Simple Methode:

- Löse iterativ eine Folge von gewichteten least-square Regressionsproblemen, bis die log-likelihood in einem Maximum konvergiert

- Konvergiert normalerweise nach ein paar Iterationen

Lineare Regression

- Paarweise Klassifikation:
 - Ein Klassifikator wird für jedes Paar von Klassen gebaut
 - Nur Instanzen dieser Klassen werden benutzt
 - Ergebnis für einen unbekannten Datensatz basiert auf der Klasse, die am besten passt
- Bringt gute Ergebnisse unter Beachtung des Klassifikationsfehlers
- Kann man auch zur Berechnung von Wahrscheinlichkeiten nutzen:
 - Paarweise koppeln
 - Kalibriert die individuellen Wahrscheinlichkeiten der verschiedenen Klassifikatoren

Lineare Regression

- Paarweise Klassifikation

- Für k Klassen werden $\frac{k(k-1)}{2}$ Klassifikatoren erstellt
- Ist dennoch genauso schnell wie die anderen Multiklassenmethoden
- Nur die Instanzen, die zu den zwei Klassen gehören, werden benutzt

- Gleichmäßige Verteilung von n Instanzen über k Klassen: $\frac{2n}{k}$ Instanzen pro Problem

- Laufzeit ist proportional zu

$$\frac{k(k-1)}{2} \cdot \frac{2n}{k} = (k-1) \cdot n$$

- Methode skaliert linear mit der Anzahl der Klassen

Nearest Neighbor

- Klassifiziere Datenpunkte aufgrund ihrer Nachbarschaft
- Attribute werden als Vektoren aufgefasst und in einen Raum gelegt
- Mehrheitsentscheidung der Trainingsdaten in der Nachbarschaft
- Algorithmus kann k Nachbarn in die Klassifikation einbeziehen
 - Zu wenig/ zu viele Nachbarn verändern die Ergebnisse
- Vektoren werden durch Distanzmaße auf „Nähe“/ „Ähnlichkeit“ geprüft
- Probleme:
 - Rechenzeit
 - Speicherbedarf
 - Verteilung der Daten im Raum

Nearest Neighbor

- Ähnlichkeitsmaß:
 - Ähnlichkeitsfunktion
 - Ähnlichkeitskoeffizient
- Ähnlichkeitsmatrix:
 - $S := (s_{ik})$
 - Symmetrisch
- Zusätzliche Bedingungen
 - $s_{ik} \geq 0$
 - $s_{ii} = 1$
- Abstandsmaß:
 - Abstandsfunction
- Abstandsmatrix:
 - $T := (t_{ik})$
 - Symmetrisch

Nearest Neighbor

- Minkowski Metriken (L_p Norm):

$$t_p(d_i, d_k) := \left(\sum_{j=1}^m |a_{ij} - a_{kj}|^p \right)^{1/p}$$

- Meist: Euklid'scher Abstand

$$\sqrt{\sum_{j=1}^m (a_{ij} - a_{kj})^2}$$

- Alternative: Manhattan (city-block) Metrik

$$\sum_{k=1}^m |a_{ij} - a_{kj}|$$

- Bemerkungen

- Für Vergleiche kann die Wurzel weggelassen werden
- Invariant bezüglich Verschiebungen
- Nicht invariant bezüglich Skalierung

Nearest Neighbor

- Mahalanobis Abstand:

$$t_M(d_i, d_k) := (d_i - d_k)^T M^{-1} (d_i - d_k)$$

$$M := \frac{1}{n} \cdot \sum_{i=1}^n (d_i - \bar{d})(d_i - \bar{d})^T$$

$$\bar{d} := \frac{1}{n} \cdot \sum_{i=1}^n d_i$$

- Bemerkungen

- M : Co-Varianz Matrix
- Invariant bezüglich Verschiebungen
- Invariant bezüglich Skalierung

Nearest Neighbor

- Algorithmus Aufbau k-d-Baum

- Initialisierung

1. Sortiere Punkte für jedes Attribut
2. $i \leftarrow 1$
3. Erzeuge Knoten
4. Starte Rekursion

- Rekursion

1. Teile alle Datenpunkte bezüglich Attribut i in zwei gleich große Teilmengen (verwende Median)
2. Trage Unterteilungswert t in aktuellen Knoten ein

3. Erzeuge linken Kindknoten

1. Weise ihm seine Datenpunkte zu
2. Falls Anzahl der Datenpunkte größer als Schwellwert
 1. $i \leftarrow i + 1$
 2. Rekursiver Aufruf

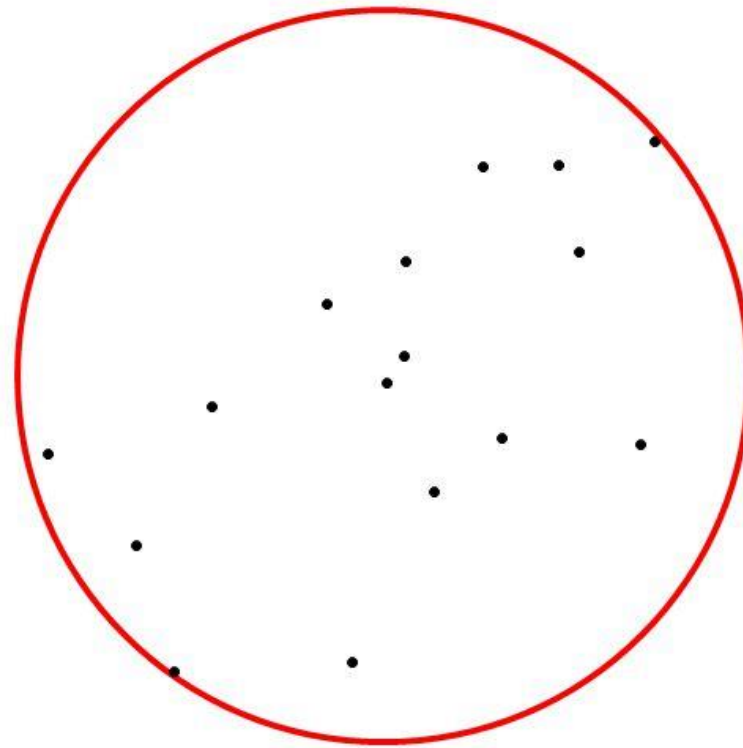
4. Erzeuge rechten Kindknoten

1. Weise ihm seine Datenpunkte zu
2. Falls Anzahl der Datenpunkte größer als Schwellwert
 1. $i \leftarrow i + 1$
 2. Rekursiver Aufruf

Ball Tree

- k-d-Bäume sind aufgrund ihrer Ecken nicht ideal
- Daten sind in der Fläche nicht gleichmäßig verteilt
- Nutzung von Sphären:
 - Zerlege den Raum in rekursiv kleiner werdende überlappende Sphären
 - Vergleichbar wie k-d-Baum
- Aufbau:
 - Wähle einen Punkt möglichst entfernt vom Mittelpunkt
 - Wähle einen zweiten möglichst weit entfernt vom ersten
 - Ordne alle Punkte dem nächsten der beiden Punkte zu
 - Berechne den Zentroid der beiden Mengen
 - Nutze den Zentroid als Sphärenmittelpunkt und die Distanz zum entferntesten Punkt als Radius
 - Zerlege die Teilmenge rekursiv weiter

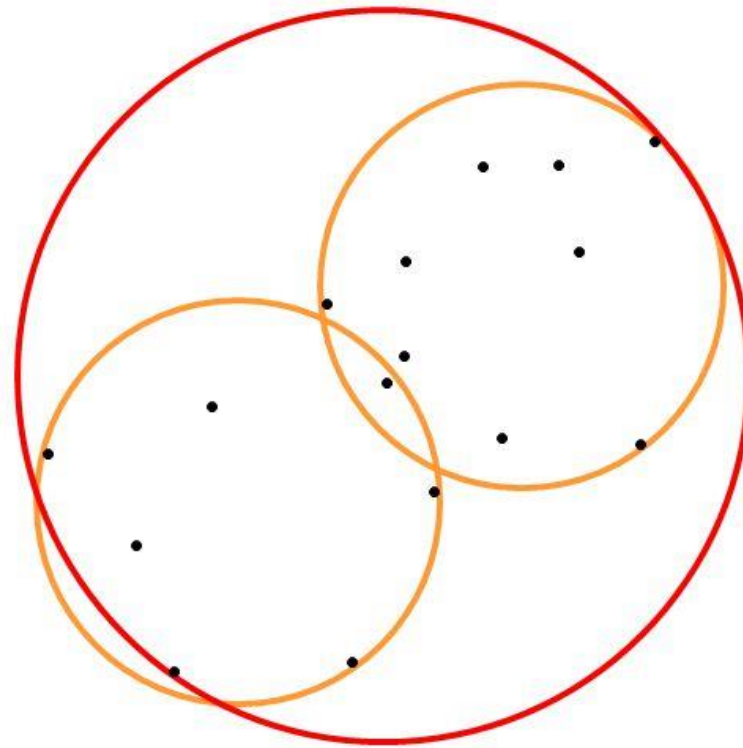
Ball Tree



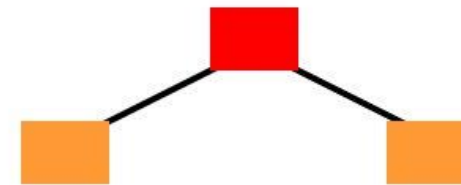
A ball-tree: level 1



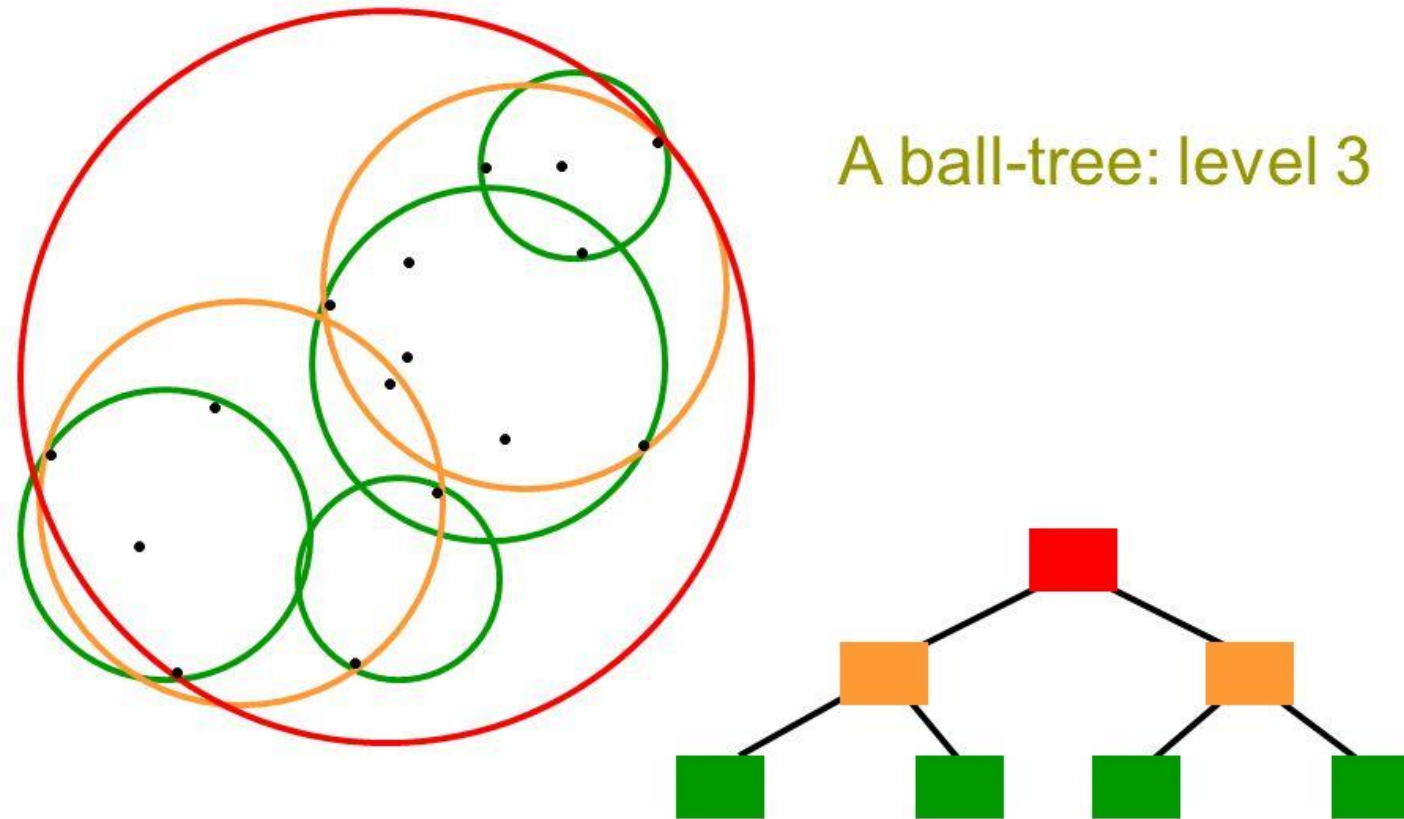
Ball Tree



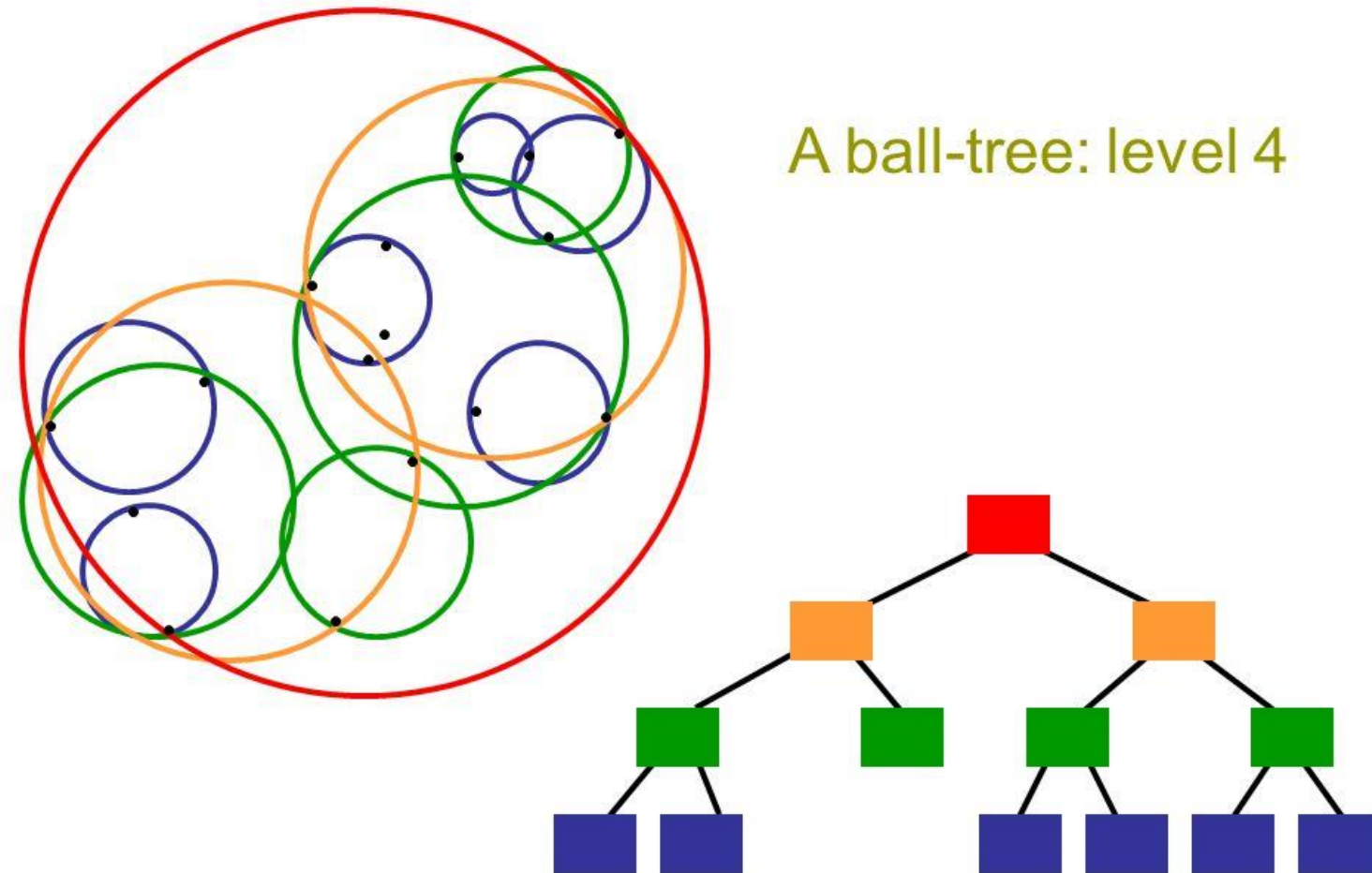
A ball-tree: level 2



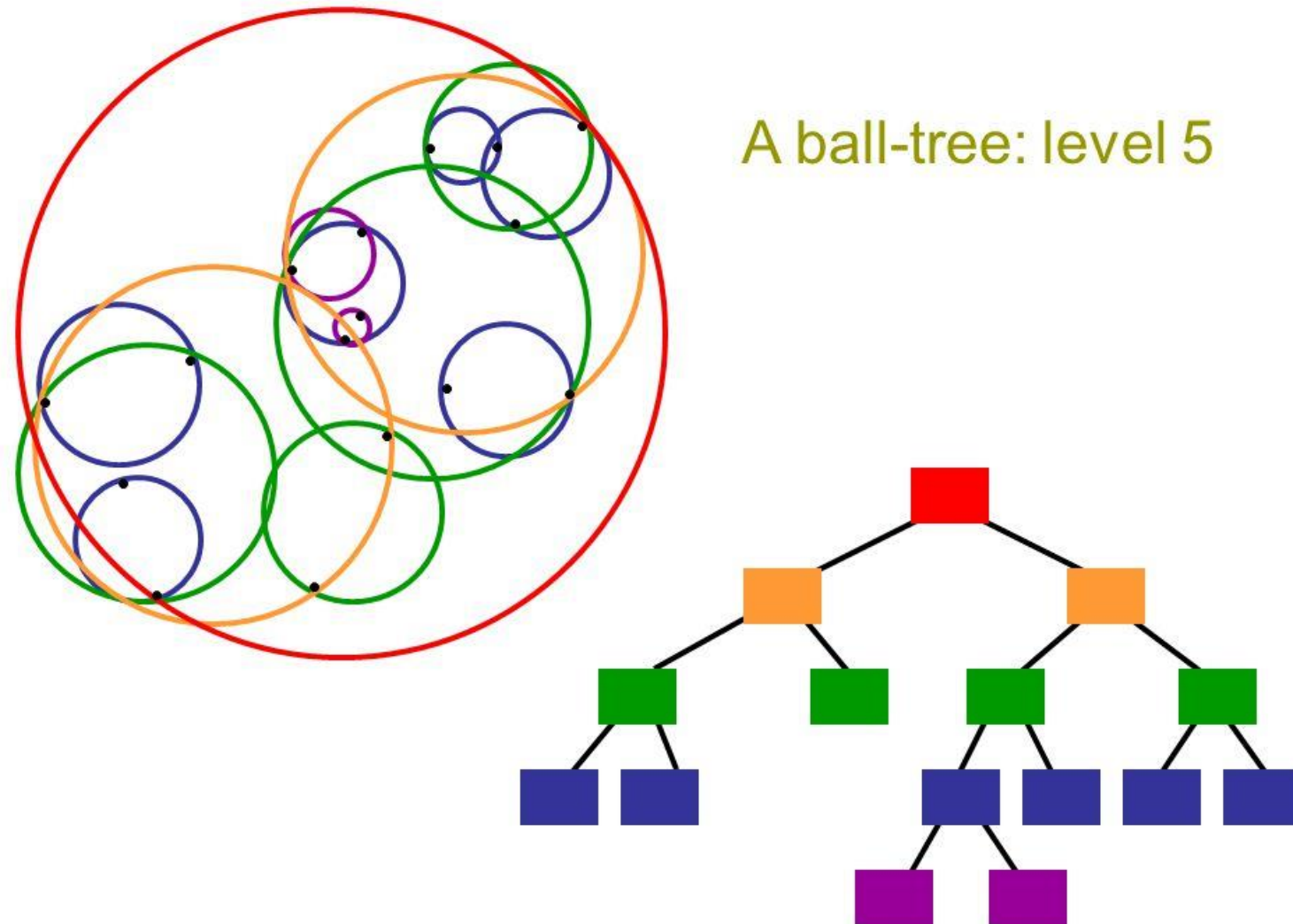
Ball Tree



Ball Tree



Ball Tree



Erzeugung von Clustern

- Clustering

- Gegeben

Sei $D = \{d_1, \dots, d_n\}$ eine Menge von Objekten.

- Gesucht

Klassen $K = \{C_1, \dots, C_k\}$, $k \ll n$

- Bedingungen

- $D = \bigcup_{i=1}^k C_i$

- $\forall i \neq j: C_i \cap C_j = \emptyset$ (Partitionierung)

- Motivation

- Suche nach Gemeinsamkeiten und Unterschieden

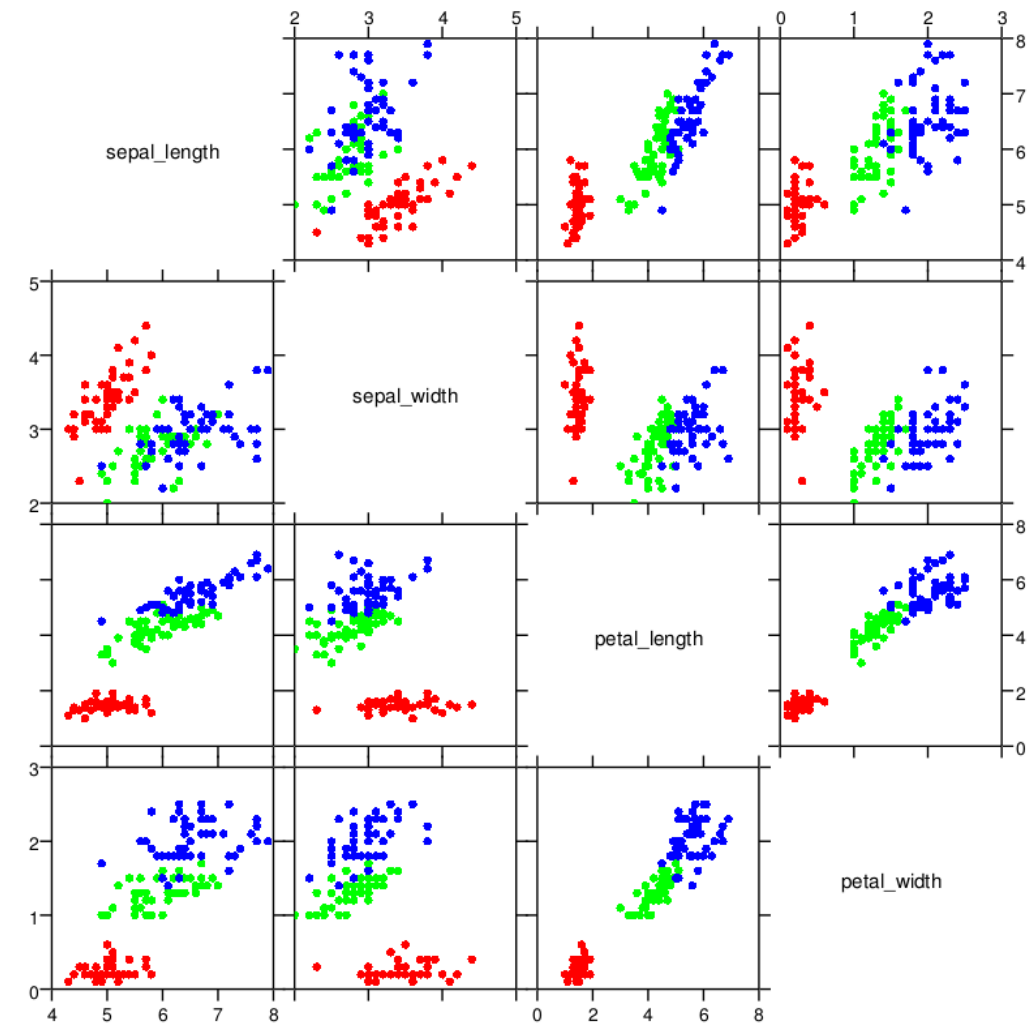
- Suche nach zusätzlicher Struktur in den Daten

- Ersetzen von Mengen von Datenpunkten (die Cluster) durch ihre Repräsentanten

- Analyse der Cluster anstelle der Datenpunkte

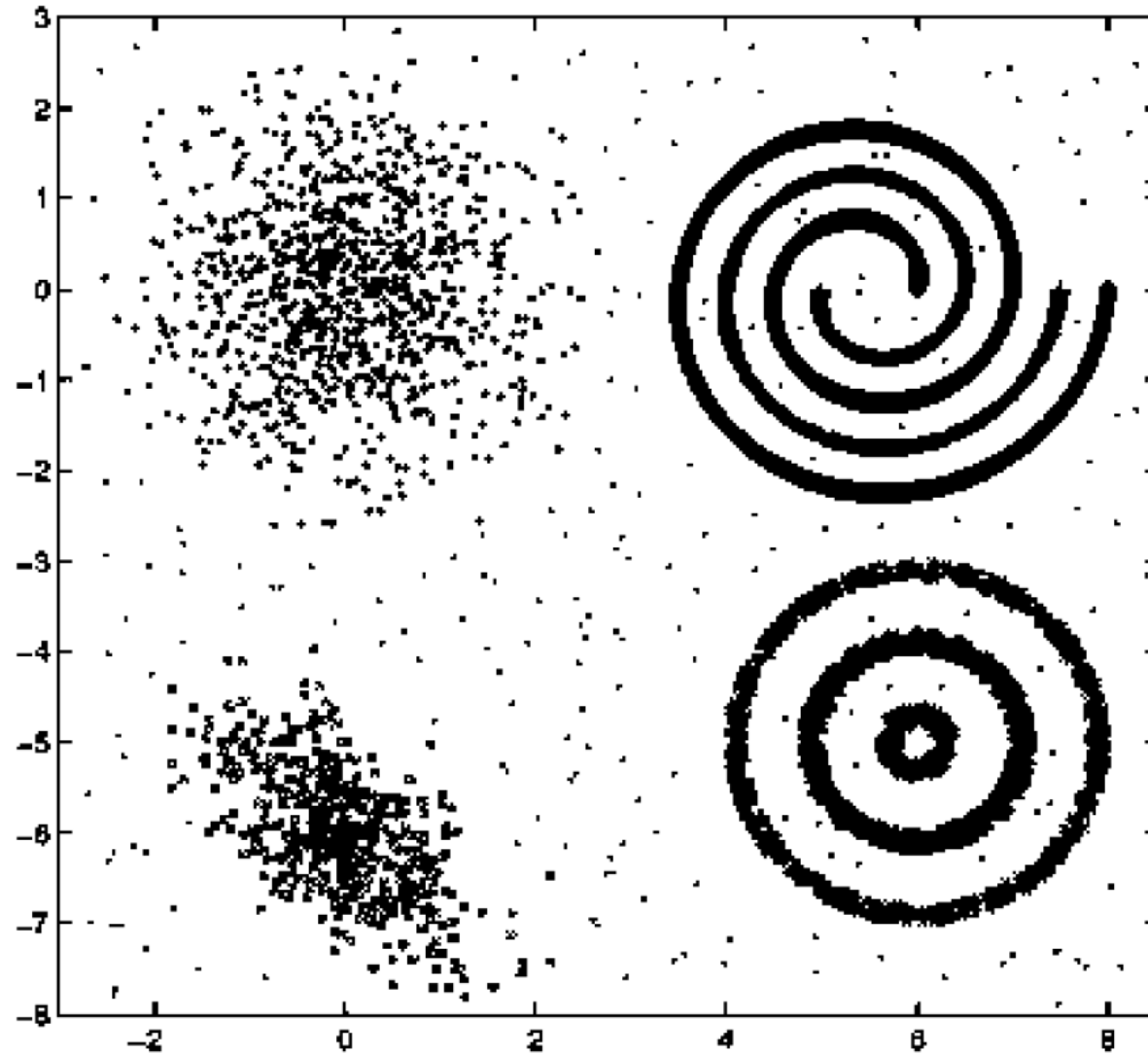
Erzeugung von Clustern

- Beispiel Iris Data Set
- <https://archive.ics.uci.edu/ml/datasets/Iris>

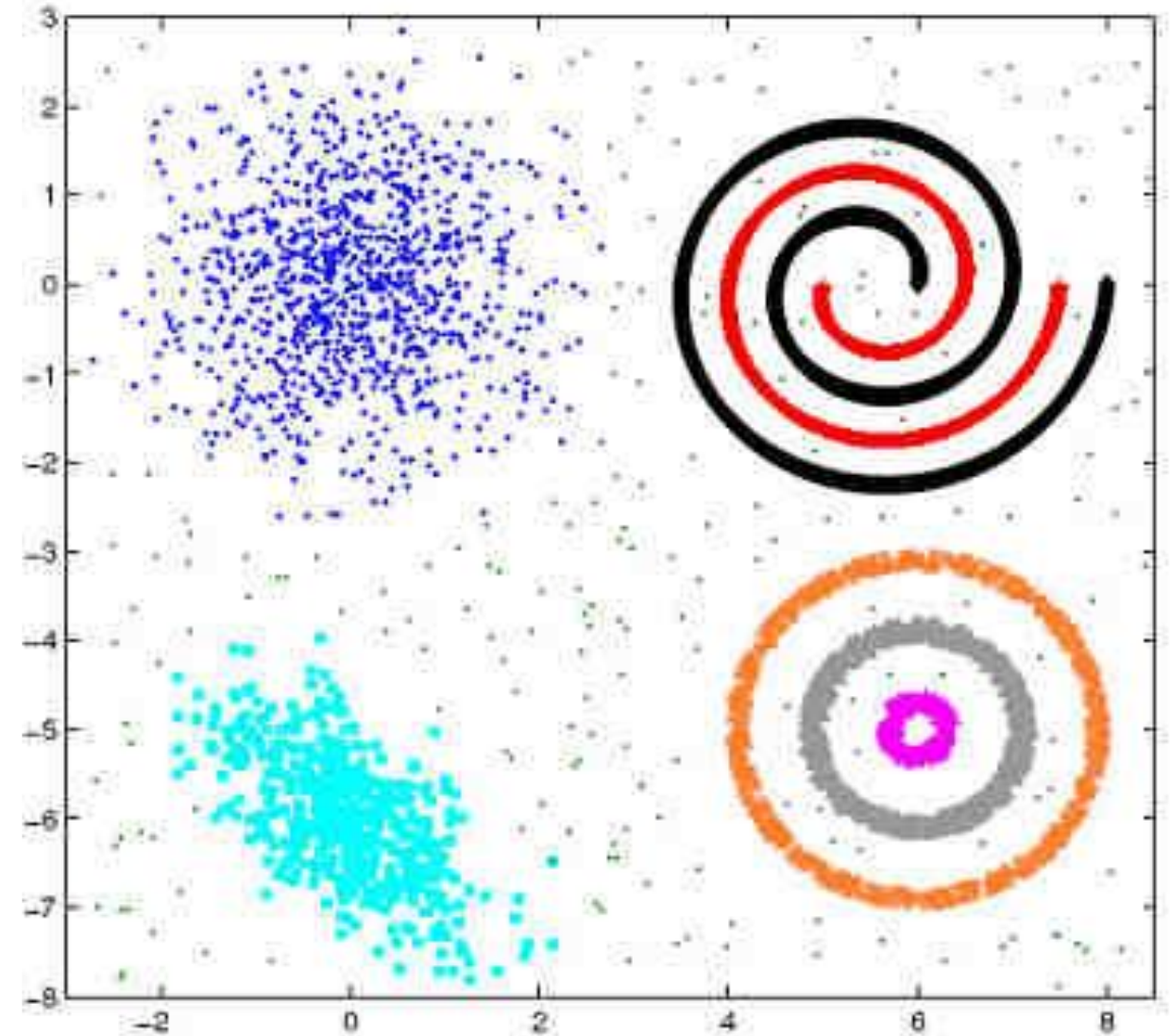
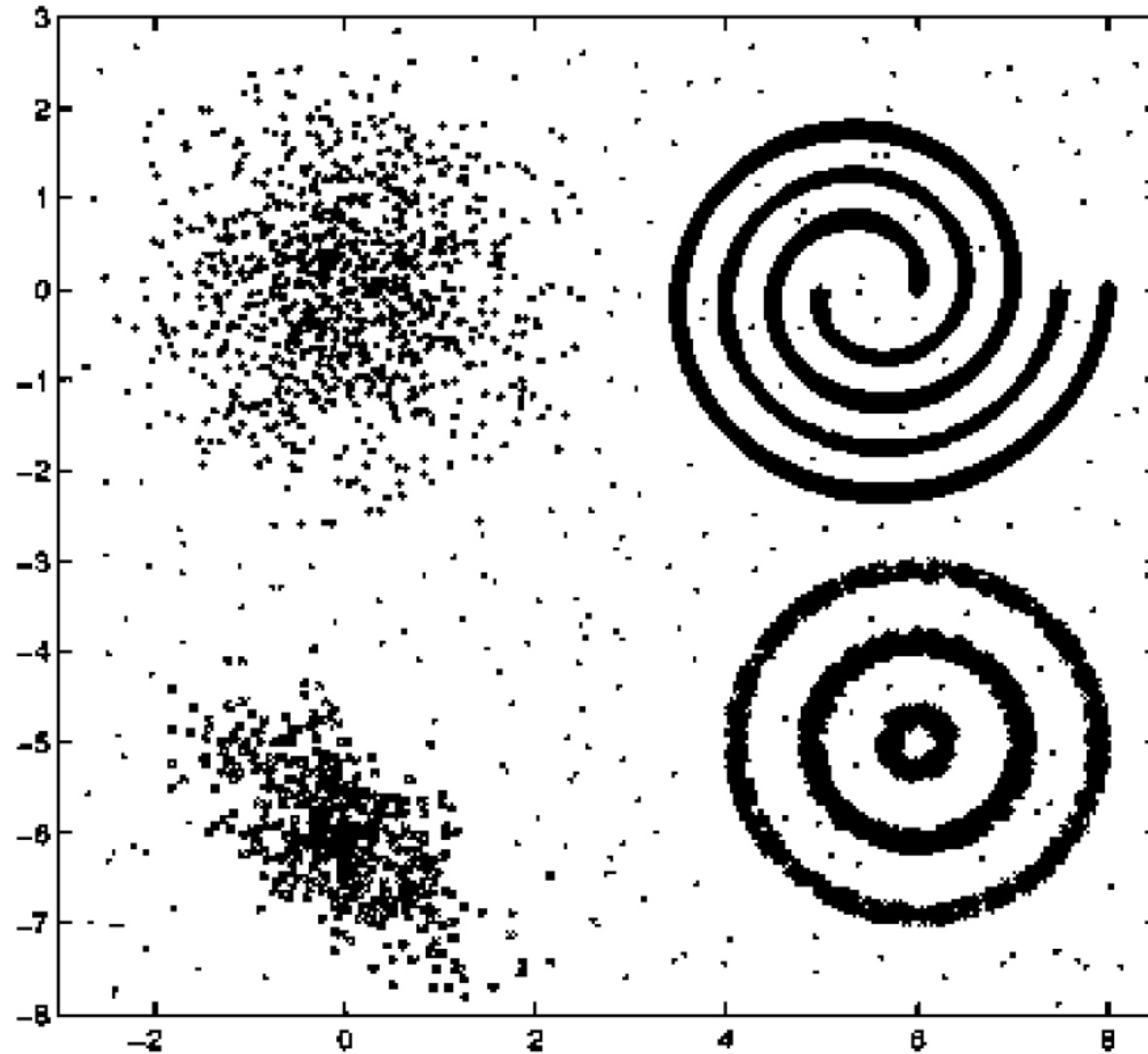


<http://cxc.harvard.edu/chips/gallery/multi.html>

Erzeugung von Clustern

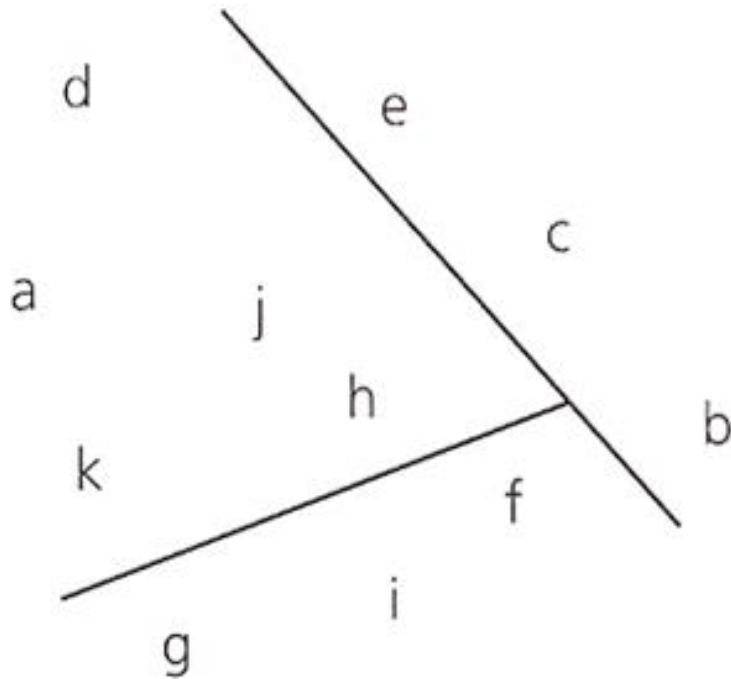


Erzeugung von Clustern



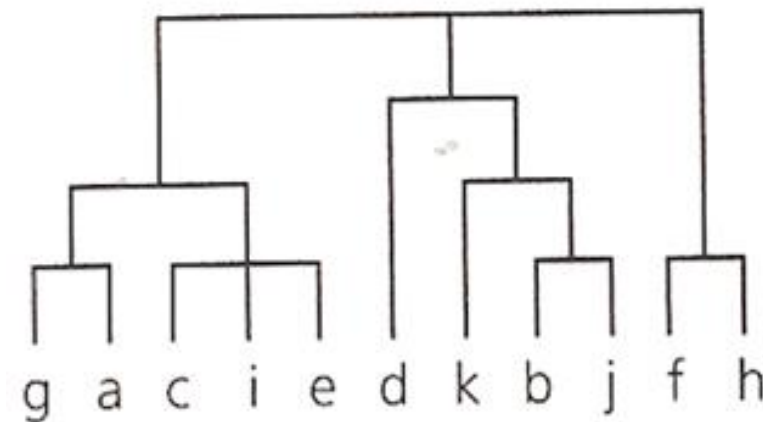
Erzeugung von Clustern

- Divisive Clustering
 - Unterteilung des Raumes
 - Zeichne Grenzen zwischen Clustern
 - Beispiel: k-means clustering



[WFH2011], Figure 3.9 (a)

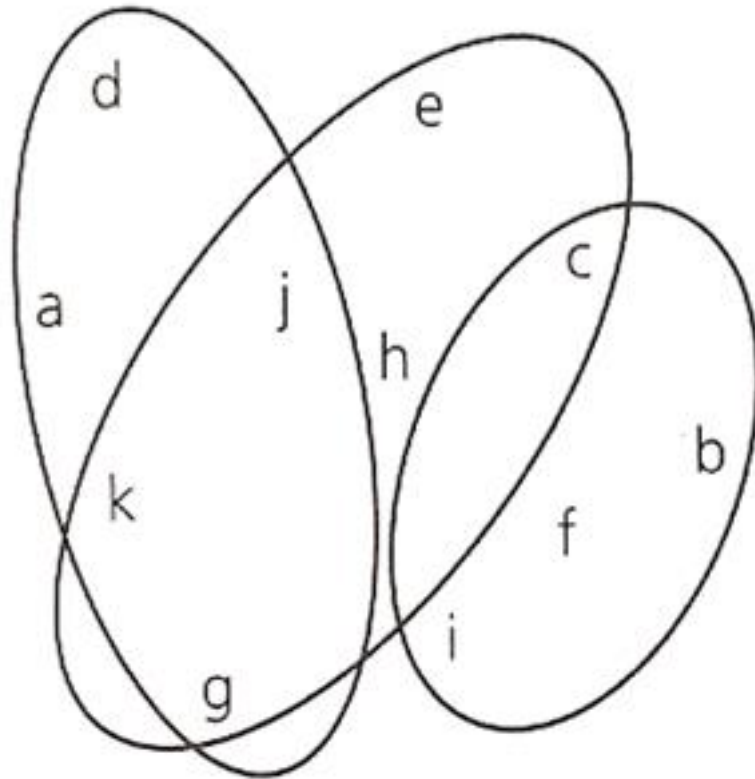
- Hierarchisches Clustering
 - Erzeugt einen Baum
 - Verwende Dendrogramme zur Darstellung
 - Beispiele:
 - Hierarchical Agglomerative Clustering
 - Divisive Hierarchical Clustering



[WFH2011], Figure 3.9 (d)

Erzeugung von Clustern

- Überlappendes Clustering
 - Darstellung: Venn-Diagramm



[WFH2011], Figure 3.9 (b)

- Probabilistisches Clustering
 - Weise jedem Element eine Wahrscheinlichkeit zu, mit der es zu den Clustern gehört
 - Die Wahrscheinlichkeiten summieren sich zu 1

	1	2	3
a	0,4	0,1	0,5
b	0,1	0,8	0,1
c	0,3	0,3	0,4
d	0,1	0,1	0,8
e	0,4	0,2	0,4
f	0,1	0,4	0,5
g	0,7	0,2	0,1
h	0,5	0,4	0,1

nach [WFH2011],
Figure 3.9 (c)

Erzeugung von Clustern

- k-means Clustering Algorithmus

1. Spezifiziere die Anzahl der Cluster: k
2. Wähle zufällig k Punkte z_1, \dots, z_k : Cluster-Zentren
3. Weise alle Instanzen d_i ihrem nächsten Cluster-Zentrum zu
4. Metrik: z.B. Euklid'sche Distanz
5. Berechne das neue Zentrum (mean) aller Cluster aus den zugehörigen Instanzen
6. Falls Cluster nicht stabil \rightarrow Schritt 3

- Cluster sind stabil
- Die Zuweisung der Instanzen zu den Clustern bleibt unverändert
- Komplexität Schritt 3: $O(k \cdot n \cdot m)$
- Komplexität Schritt 5: $O(k \cdot n \cdot m)$

Erzeugung von Clustern

- k-means Clustering

- Qualität

- Berechne

$$q = \sum_{i=1}^k \sum_{j=1}^{n_i} (d_{i,j} - z_i)^2$$

- z_i : Zentrum des i -ten Clusters

- $d_{i,j}$: j -ter Punkt des i -ten Clusters

- $n_i = |C_i|$

- Problematik

- Berechnet ein lokales Minimum q , aber nicht notwendigerweise das globale Minimum

- Ergebnis ist abhängig von

- k

- den gewählten Startzentren der Cluster

- der „empty-Cluster“ Strategie

- Splitte „größten Cluster“

- größte Anzahl

- größte Varianz

- Nehme vom Zentroiden des Clusters am weitesten entfernten Punkt

Erzeugung von Clustern

- k-means Clustering

- Verbesserungsmöglichkeit
 - Lasse k-means mehrfach mit verschiedenen Startzentren laufen
 - Wähle Ergebnis mit geringstem q
- Optimierung der Wahl der Startzentren: k-means++
 - Wähle ersten Startpunkt zufällig
 - Wähle nächsten Punkt aus den verbleibenden Punkten mit Wahrscheinlichkeit proportional zur Entfernung zu allen bisherigen Punkten
 - Erhöht
 - Konvergenzgeschwindigkeit
 - Genauigkeit

Erzeugung von Clustern

- k-means Clustering

- Wie findet man optimales k ?

1. Schätzung der Anzahl der Cluster [MKB1979]:

$$k = \sqrt{\frac{n}{2}}$$

2. Nutze Domänenwissen zur Abschätzung der Anzahl der Cluster (ist vorzuziehen)
3. Variiere Anzahl der Cluster k , ausgehend von initialem Wert (z.B. $k - 2, \dots, k + 2$)

- Problem: Evaluation des Ergebnisses

- Optimal: $k = n \rightarrow q = 0$
- im Allgemeinen werden größere k bevorzugt

- Mögliche Lösung: Cluster Indices

- Relative Indices
 - Davies-Bouldin Index [DB1979]
 - Modified Hubert Γ [HA1985]
- Dunn Index [BP1998]
- Silhouette Coefficient [Rou1987]

Erzeugung von Clustern

- Davies Bouldin Index

- Kohäsion

$$S_i = \left(\frac{1}{n_i} \sum_{j=1}^{n_i} |d_{i,j} - z_i|^p \right)^{\frac{1}{p}}$$

- Trennung

$$M_{i,j} = \|z_i - z_j\|_p = \left(\sum_{k=1}^m |z_{i,k} - z_{j,k}|^p \right)^{\frac{1}{p}}$$

- Maß für Qualität des Clusterings für i und j

$$R_{i,j} = \frac{S_i + S_j}{M_{i,j}}$$

- Schlechtester Wert für i

$$D_i = \max_{j \neq i} \{R_{i,j}\}$$

- Davies-Bouldin Index

$$DB := \frac{1}{k} \sum_{i=1}^k D_i$$

Erzeugung von Clustern

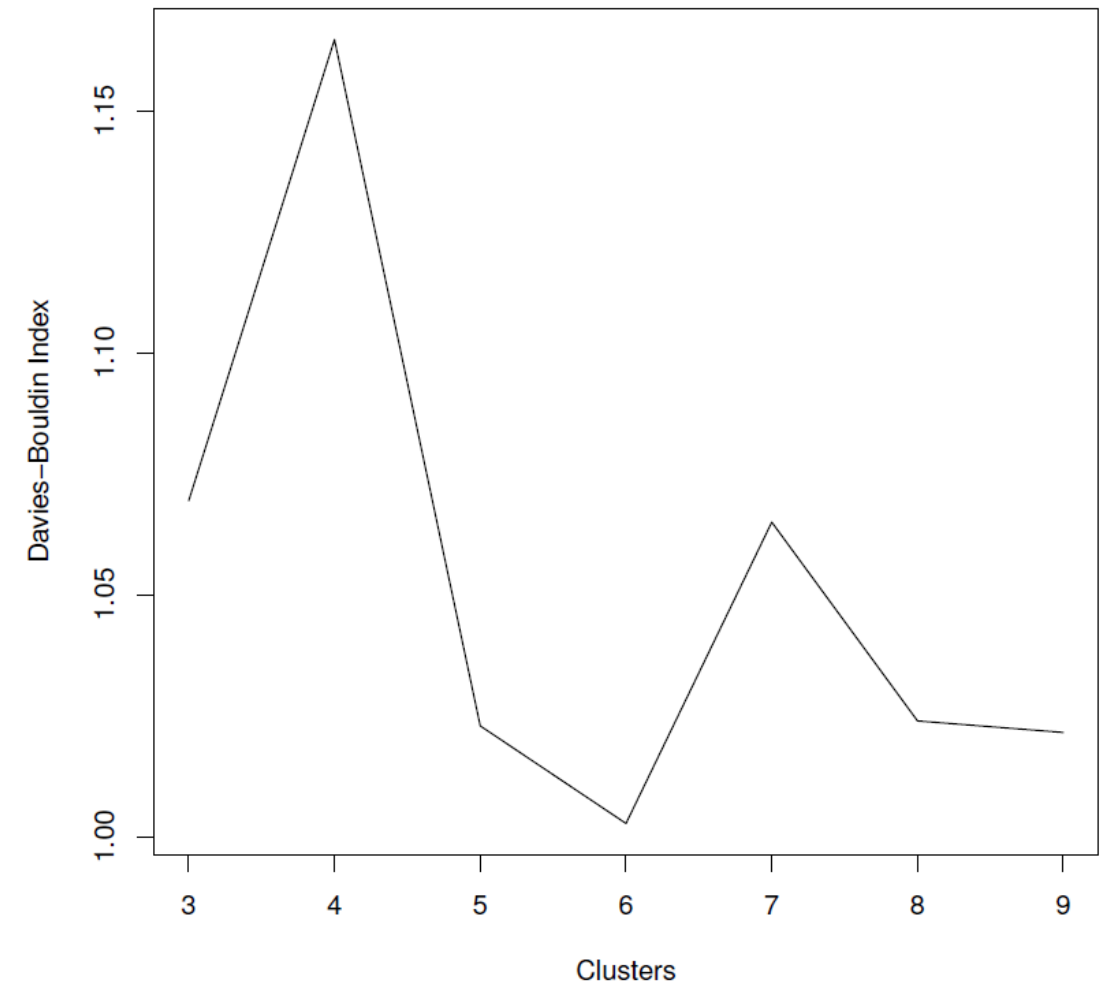
- Davies Bouldin Index
 - Komplexität
 - S_i : $O(n \cdot m)$
 - $M_{i,j}$: $O(m)$
 - $R_{i,j}$: $O(n \cdot m + n \cdot m + m) = O(n \cdot m)$
 - D_i : $O(k \cdot n \cdot m)$
 - DB : $O(k \cdot k \cdot n \cdot m) = O(k^2 \cdot n \cdot m)$

Erzeugung von Clustern

- Davies Bouldin Index

k	Davies Bouldin Index
3	1,06957
4	1,16492
5	1,02300
6	1,00282
7	1,06513
8	1,02402
9	1,02164

— Wähle k mit kleinstem Index: 6



Erzeugung von Clustern

- Modified Hubert Γ

- Clusterindikator

$$L(i) = k$$

Objekt i gehört zu Cluster k

- Matrizen

$$X(i, j) = |d_i - d_j|$$

$$Y(i, j) = |z_{L(i)} - z_{L(j)}|$$

- Modified Hubert Γ

$$\Gamma = \sum_{i=1}^{n-1} \sum_{j=i+1}^n (X(i, j) \cdot Y(i, j))$$

- Komplexität

- $O(n \cdot n \cdot m \cdot m) = O(n^2 \cdot m^2)$

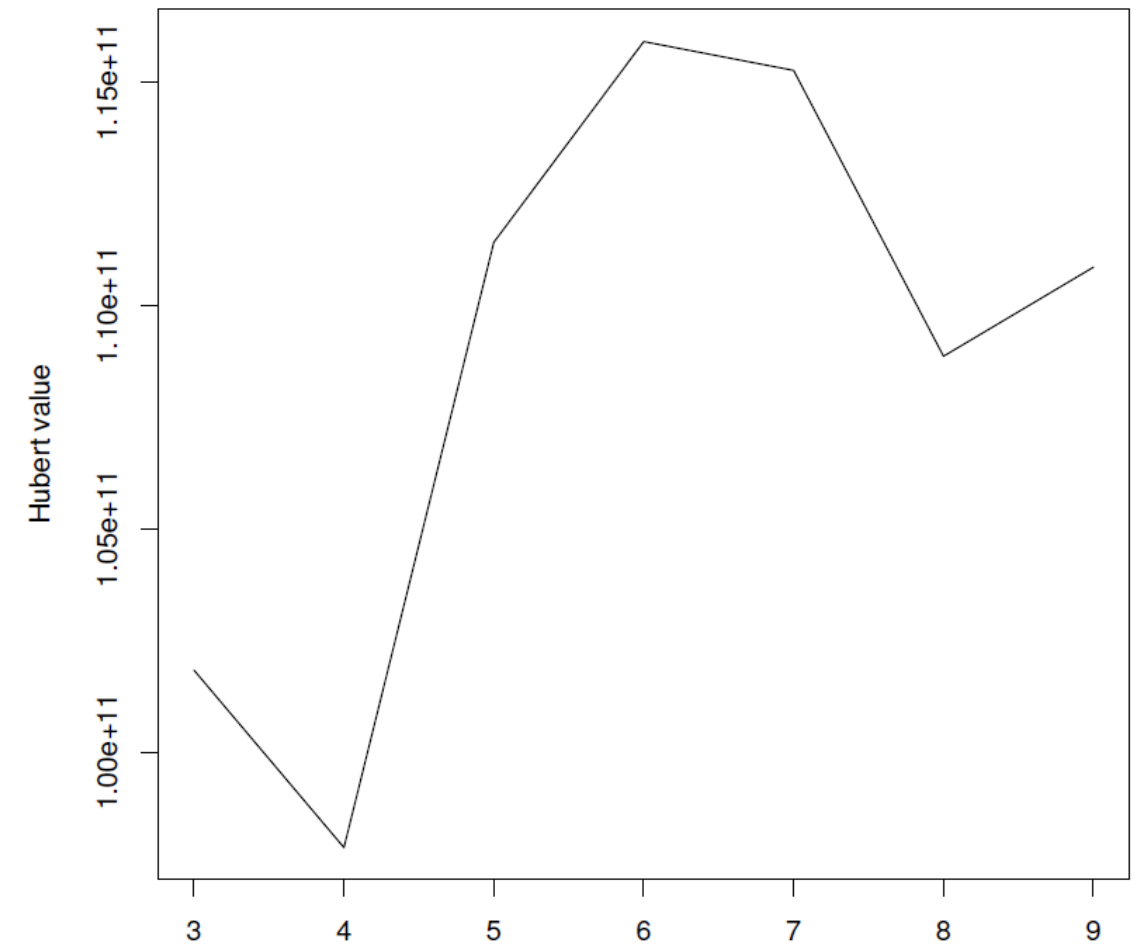
- unabhängig von k

Erzeugung von Clustern

- Modified Hubert Γ

k	Modified Hubert Γ
3	$1,01844 \cdot 10^{11}$
4	$0,97873 \cdot 10^{11}$
5	$1,11428 \cdot 10^{11}$
6	$1,15922 \cdot 10^{11}$
7	$1,15276 \cdot 10^{11}$
8	$1,08876 \cdot 10^{11}$
9	$1,10869 \cdot 10^{11}$

- Wähle k nach dem größten Anstieg: 5



Erzeugung von Clustern

- Dunn Index

- Berechnung

$$DI_k = \frac{\min_{1 \leq i < j \leq k} \delta(C_i, C_j)}{\max_{1 \leq l \leq k} \Delta_l}$$

- $\delta(C_i, C_j)$: Abstand zwischen Cluster i und j
- Δ_l : Kohäsion des Clusters l
- Komplexität: abhängig von $\delta(C_i, C_j)$ und Δ_l

- Δ_l

- Maximale Distanz zweier Elemente des Clusters l

$$\Delta_l = \max_{x, y \in C_l} d(x, y)$$

- Durchschnittliche Distanz zweier Elemente des Clusters l

$$\Delta_l = \frac{1}{|C_l| \cdot (|C_l| - 1)} \cdot \sum_{x, y \in C_l, x \neq y} d(x, y)$$

- Abstände aller Punkte vom Zentrum des Clusters l

$$\Delta_l = \frac{\sum_{x \in C_l} d(x, z_l)}{|C_l|}$$

Erzeugung von Clustern

- Dunn Index

- $\delta(C_i, C_j)$

- Minimaler Abstand zwischen zwei Punkten

$$\delta(C_i, C_j) = \min_{x \in C_i, y \in C_j} \{d(x, y)\}$$

- Maximaler Abstand zwischen zwei Punkten

$$\delta(C_i, C_j) = \max_{x \in C_i, y \in C_j} \{d(x, y)\}$$

- Durchschnitt aller Abstände zwischen zwei Punkten

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

- $\delta(C_i, C_j)$

- Abstand der Cluster-Zentren

$$\delta(C_i, C_j) = d(z_i, z_j)$$

- Durchschnitt der Abstände zum anderen Cluster-Zentrum

$$\delta(C_i, C_j) = \frac{1}{|C_i| + |C_j|} \cdot \left(\sum_{x \in C_i} d(x, z_j) + \sum_{x \in C_j} d(x, z_i) \right)$$

- Maximum der minimalen Abstände von Punkten zweier Cluster

$$\delta(C_i, C_j) = \max\{\delta'(C_i, C_j), \delta'(C_j, C_i)\}$$

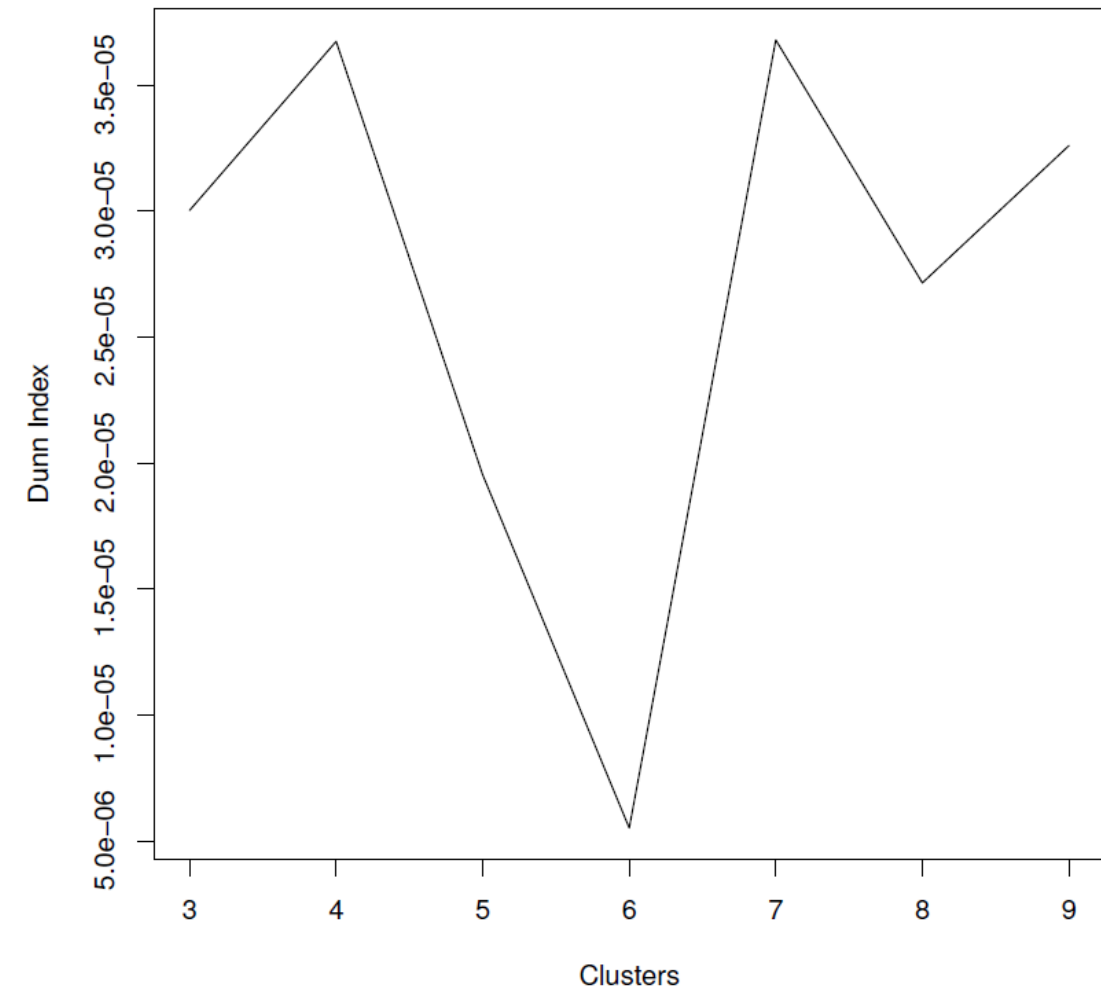
$$\delta'(C_i, C_j) = \max_{x \in C_i} \min_{y \in C_j} \{d(x, y)\}$$

Erzeugung von Clustern

- Dunn Index

k	Dunn Index
3	$3,00291 \cdot 10^{-5}$
4	$3,67348 \cdot 10^{-5}$
5	$1,95366 \cdot 10^{-5}$
6	$0,55217 \cdot 10^{-5}$
7	$3,67917 \cdot 10^{-5}$
8	$2,71530 \cdot 10^{-5}$
9	$3,25993 \cdot 10^{-5}$

– Wähle k mit dem kleinsten Index: 6



Nearest Neighbor

- Algorithmus Aufbau k-d-Baum

- Initialisierung

1. Sortiere Punkte für jedes Attribut
2. $i \leftarrow 1$
3. Erzeuge Knoten
4. Starte Rekursion

- Rekursion

1. Teile alle Datenpunkte bezüglich Attribut i in zwei gleich große Teilmengen (verwende Median)
2. Trage Unterteilungswert t in aktuellen Knoten ein

3. Erzeuge linken Kindknoten

1. Weise ihm seine Datenpunkte zu
2. Falls Anzahl der Datenpunkte größer als Schwellwert
 1. $i \leftarrow i + 1$
 2. Rekursiver Aufruf

4. Erzeuge rechten Kindknoten

1. Weise ihm seine Datenpunkte zu
2. Falls Anzahl der Datenpunkte größer als Schwellwert
 1. $i \leftarrow i + 1$
 2. Rekursiver Aufruf

Erzeugung von Clustern

- Silhouette Coefficient

- Berechnung

$$s(d_i) = \frac{d(d_i, C_n) - d(d_i, C_l)}{\max\{d(d_i, C_l), d(d_i, C_n)\}}$$

- Abstand zwischen d_i und einem Cluster C

$$d(d_i, C) = \frac{1}{n_C} \cdot \sum_{d_j \in C} d(d_i, d_j)$$

- C_l : Cluster von d_i

- $C_n = \operatorname{argmin}_{C \neq C_l} d(d_i, C)$: Cluster, der Element mit kleinstem Abstand zu d_i enthält

- Komplexität

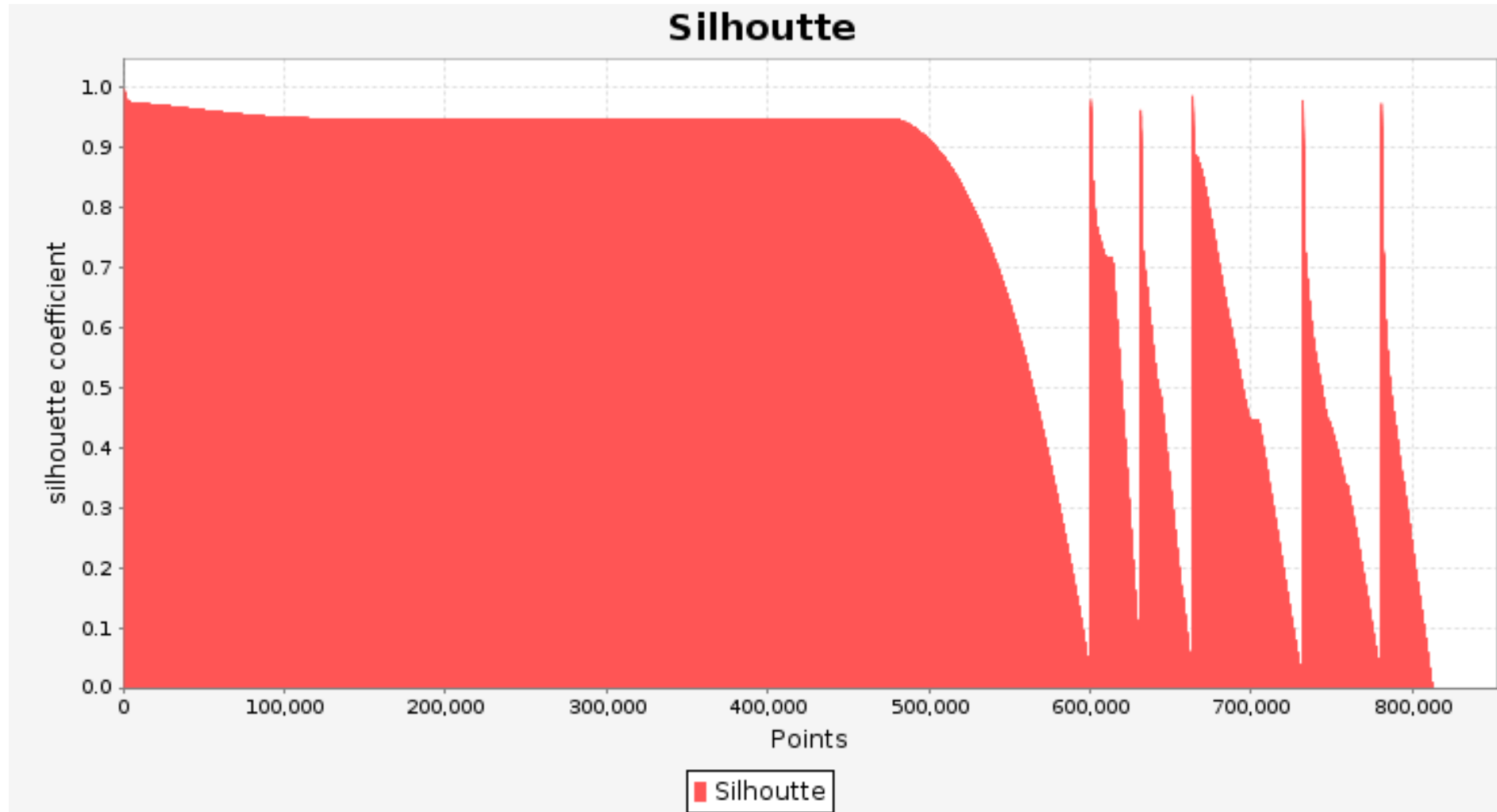
$$O(n^2 \cdot m \cdot k)$$

- Auswertung

	relationship
$0,75 < s(i) \leq 1$	strong
$0,5 < s(i) \leq 0,75$	normal
$0,25 < s(i) \leq 0,5$	weak
$0 < s(i) \leq 0,25$	no

Erzeugung von Clustern

- Silhouette Coefficient



Erzeugung von Clustern

- k-means Clustering
- Beschleunigung [HD2015]:
 - Raumteilungsverfahren
 - k-d-Bäume (k-dimensionale Bäume)
 - Ball-Trees [WFH2011]
 - Parallelisierung

Erzeugung von Clustern

- k-means Clustering
 - k-d-Bäume (k-dimensionale Bäume)
 - k : Anzahl der Dimensionen
 - entspricht der Anzahl der Attribute m
 - Aufbau: $O(k \cdot n \cdot \log n)$
 - Sortiere n Punkte nach k Attributen
 - Nearest-Neighbor-Suche: $O(n \cdot \log n)$
 - Sinnvoll bei
 - Vielen Datenpunkten
 - Wenigen Dimensionen
(Langsam für $m > 8$ [HD2015])

Erzeugung von Clustern

- Frage: Welches Clustering ist die richtige?
 - Im Allgemeinen nicht entscheidbar
- Zusatzfragen
 - Welches Verfahren soll gewählt werden?
 - Welche Parameter sollen für das Verfahren gewählt werden?
 - Wie viele Cluster sollen gewählt werden?

Erzeugung von Clustern

- Evidence Accumulation [FJ2002]

- Unterteile (Split)

- Unterteile die Daten in mehrere kugelförmige Cluster
- Berechne N -mal k-means mit unterschiedlichen Initialisierungen

- Verbinde (Combine)

- Berechne

$$co_assoc(i, j) = \frac{votes_{i,j}}{N}$$

- $votes_{i,j}$: Anzahl der Durchläufe bei denen i und j im gleichen Cluster liegen

- Vereinige (Merge)

- Verwende Minimalen Spannbaum mit Schwellwert t um Cluster zu bestimmen

Erzeugung von Clustern

- Evidence Accumulation [FJ2002]
 - Algorithmus
 - N -mal
 - Wähle k Cluster-Zentren
 - Berechne Clustering mit k-means
 - Aktualisiere die Co-Assoziationsmatrix
$$co_assoc(i, j) \leftarrow co_assoc(i, j) + \frac{1}{N}$$
 - Für alle Paare (i, j)
 - Falls $t \leq co_assoc(i, j)$ vereinige die Cluster in denen i und j liegen
 - Für alle Objekte, die nicht in einem Cluster liegen
 - Bilde ein-elementigen Cluster mit diesem Objekt

Erzeugung von Clustern

- Evidence Accumulation [FJ2002]
 - Ergebnis 4-mal k-means mit $k = 2$

Objekt	Lauf 1	Lauf 2	Lauf 3	Lauf 4
d_1	1	2	2	1
d_2	1	2	2	1
d_3	1	2	1	1
d_4	1	2	1	1
d_5	2	1	2	2
d_6	2	1	2	2
d_7	2	1	2	2

- Assoziationsmatrix

	d_1	d_2	d_3	d_4	d_5	d_6	d_7
d_1	1	1	0,75	0,75	0,25	0,25	0,25
d_2	1	1	0,75	0,75	0,25	0,25	0,25
d_3	0,75	0,75	1	1	0	0	0
d_4	0,75	0,75	1	1	0	0	0
d_5	0,25	0,25	0	0	1	1	1
d_6	0,25	0,25	0	0	1	1	1
d_7	0,25	0,25	0	0	1	1	1

Erzeugung von Clustern

- Evidence Accumulation [FJ2002]
 - Ergebnis 4-mal k-means mit $k = 2$

	d_1	d_2	d_3	d_4	d_5	d_6	d_7
d_1	1	1	0,75	0,75	0,25	0,25	0,25
d_2	1	1	0,75	0,75	0,25	0,25	0,25
d_3	0,75	0,75	1	1	0	0	0
d_4	0,75	0,75	1	1	0	0	0
d_5	0,25	0,25	0	0	1	1	1
d_6	0,25	0,25	0	0	1	1	1
d_7	0,25	0,25	0	0	1	1	1

- Ergebnis
 - $0 \leq t \leq 0,25$:
 $\{d_1, d_2, d_3, d_4, d_5, d_6, d_7\}$
 - $0,25 < t \leq 0,75$:
 $\{d_1, d_2, d_3, d_4\}, \{d_5, d_6, d_7\}$
 - $0,75 < t \leq 1$:
 $\{d_1, d_2\}, \{d_3, d_4\}, \{d_5, d_6, d_7\}$

Erzeugung von Clustern

- Evidence Accumulation [FJ2002]
 - Bemerkungen
 - Zur Unterteilung können beliebige Cluster-Verfahren benutzt werden
 - Die Matrix benötigt $O(n^2)$ Speicher
→ ungünstig bei großem n
 - Alternative
 - Speichere für jeden Durchlauf in welchem Cluster das Objekt liegt
(Speicher: $O(n \cdot k)$)
 - Berechne Hamming-Distanz der Clustervektoren
 - Hamming Distanz 1 gruppiert alle Objekte, welche in allen Durchläufen im jeweils gleichen Cluster waren

Erzeugung von Clustern

- Consensus Clustering

Objekt	Lauf 1	Lauf 2	Lauf 3	Lauf 4
d_1	1	2	2	1
d_2	1	2	2	1
d_3	1	2	1	1
d_4	1	2	1	1
d_5	2	1	2	2
d_6	2	1	2	2
d_7	2	1	2	2

- k-means

- 4 Läufe

- $k = 2$

- Ergebnis

- 3 Cluster

- $C_1 = \{d_1, d_2\}$

- $C_2 = \{d_3, d_4\}$

- $C_3 = \{d_5, d_6, d_7\}$

Erzeugung von Clustern

- Hierarchical Agglomerative Clustering
 - Bottom-Up Technik
 - Anfangszustand:
 - Jedes Objekt ist in einer eigenen Klasse
 - $g = m$
 - $\forall i: C_i^0 = \{d_i\}$
 - $G^0 := \{C_1^0, \dots, C_g^0\}$

- Divisive Hierarchical Clustering
 - Top-Down Technik
 - Anfangszustand:
 - Alle Objekte sind in einer Klasse
 - $g = 1$
 - $G^0 = \{C_1^0\} = \{\{d_1, \dots, d_m\}\}$

Erzeugung von Clustern

- Hierarchical Agglomerative Clustering

- Wähle ein Ähnlichkeitsmaß s
- Vereinige die beiden ähnlichsten Klassen:
 - Seien $j \neq k: s(C_j^0, C_k^0)$ maximal
 - $C_j^1 = C_j^0 \cup C_k^0$
 - $G^1 = G^0 \setminus \{C_j^0, C_k^0\} \cup C_j^1$

- Divisive Hierarchical Clustering

- Wähle ein Abstandsmaß t
- Unterteile C_1^0 in zwei Klassen:
 - $G^1 := \{C_1^1, C_2^1\}$
 - $C_1^1 \cup C_2^1 = D$
 - $C_1^1 \cap C_2^1 = \emptyset$
 - $t(C_1^1, C_2^1)$ ist maximal

Erzeugung von Clustern

- Hierarchical Agglomerative Clustering

- Wiederhole bis nur noch eine Klasse vorhanden ist:

$$G^m = \{C_1^m\} = \{\{d_1, \dots, d_m\}\}$$

- Endzustand entspricht Ausgangszustand des „Divisive Hierarchical Clustering“

- Divisive Hierarchical Clustering

- Wiederhole bis jede Klasse genau ein Objekt enthält:

- $g = m$

- $\forall i: C_i^m = \{d_i\}$

- $G^m := \{C_1^m, \dots, C_g^m\}$

- Endzustand entspricht Ausgangszustand des „Hierarchical Agglomerative Clustering“

Erzeugung von Clustern

- Hierarchical Agglomerative Clustering
 - Oft: Euklid'sche Distanz
 - Beachte: nicht invariant bezüglich Skalierung

- Maße:

- Single Linkage:

$$t_{SL}(C_k, C_l) := \min_{d_i \in C_k, d_j \in C_l} \{ |d_i - d_j|_2 \}$$

- Complete Linkage:

$$t_{CL}(C_k, C_l) := \max_{d_i \in C_k, d_j \in C_l} \{ |d_i - d_j|_2 \}$$

Erzeugung von Clustern

- Darstellung der erzeugtem Clusterhierarchie als Dendrogramm [Mar1986]

