Cluster Analysis

Machine Learning I

Prof. K-R Müller
(Dr. S. Lemm)

TU Berlin

WS 2009/10

Outline

Unüberwachtes Lernen

Überwachtes Lernen

- gegeben: $(x_i, y_i)_{i=1...N}$
  Menge von Input-Output Relationen
- gesucht: $f : x \mapsto y$
  Funktion, welche x auf y abbildet
- Bsp: Klassifikation $y \in \{\pm 1\}$
  Regression $y \in \mathbb{R}$

Unüberwachtes Lernen

- gegeben $(x_i)_{i=1...N}$
  Menge von ungelabelten Datenpunkten
- gesucht: Eigenschaften der Verteilung $p(x)$ der Daten
- Bsp: Principle Component Analysis, Independent Component Analysis

Warum unüberwachtes Lernen

- Inputdaten $(x_i)_{i=1...N}$ leicht verfügbar – Label $(y_i)_{i=1...N}$ aufwendig zu erheben
- erste Analysen über inherente Strukturen der Daten, d.h. Explorative Datenanalyse
- $x \in \mathbb{R}^n$ hoch-dimensional: ¿Exisitieren niedrig-dimensionale Feature (Merkmale) welche x hinreichend erklären?
- $N \to \infty$: Wahl geeigneter Repräsentanten zur Reduktion der Komplexität, Kompression
- Cluster Analysen sind eine Methode des unüberwachten Lernens

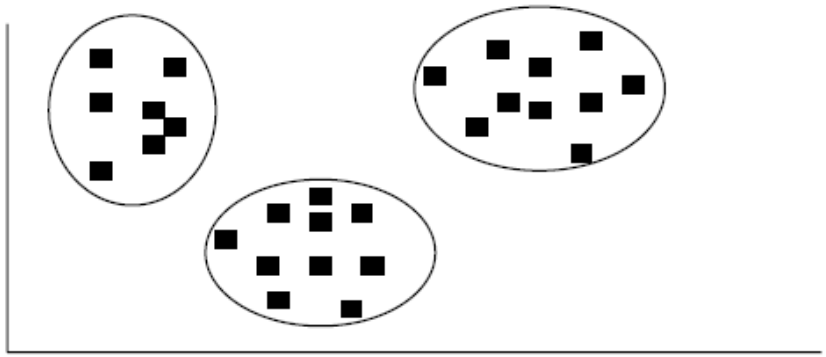
Was ist Cluster Analyse?

Gegeben:

- Menge von Objekten (nicht notwendigerweise metrisch -> Texte, Bilder, ...)
- Relation zwischen Objekten (Ähnlichkeits-, Abstandsmaß, Nachbarschaftsrelation, ...)

Ziel (intuitiv): Finde Gruppierung der Objekte, so dass

- Objekte einer Gruppe ähnlich sind
- Objekte verschiedener Gruppe unterschiedlich sind

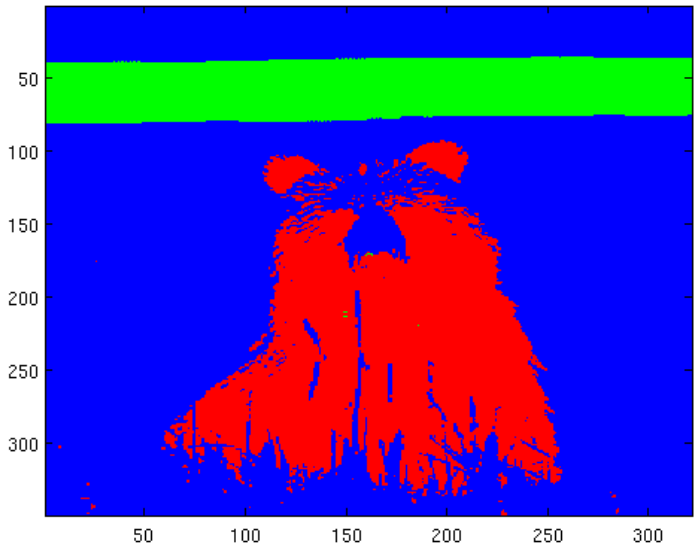


Anwendung von Cluster Analysen

Dokumente, z.B. Search Results

Anwendung von Cluster Analysen

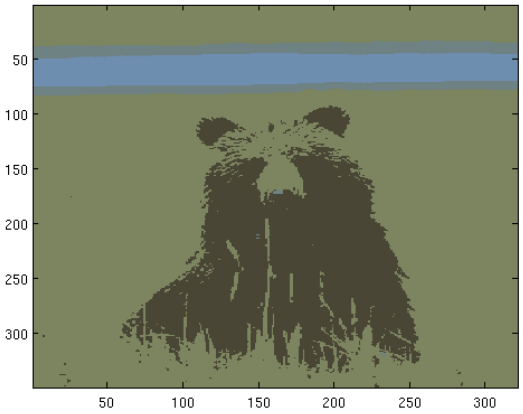
Image Segmentation



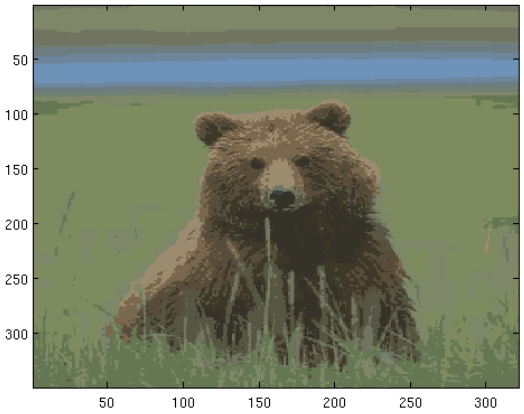
Anwendung von Cluster Analysen

Image Kompression - Vektorquantisierung

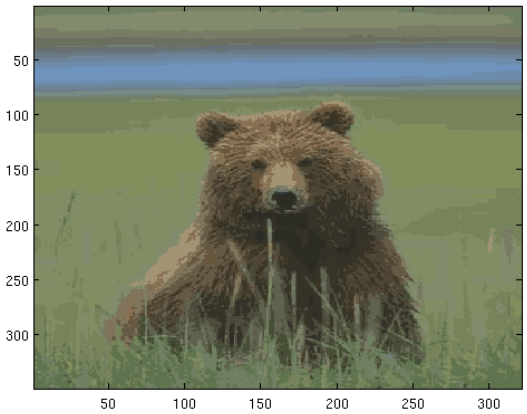
2 Bit



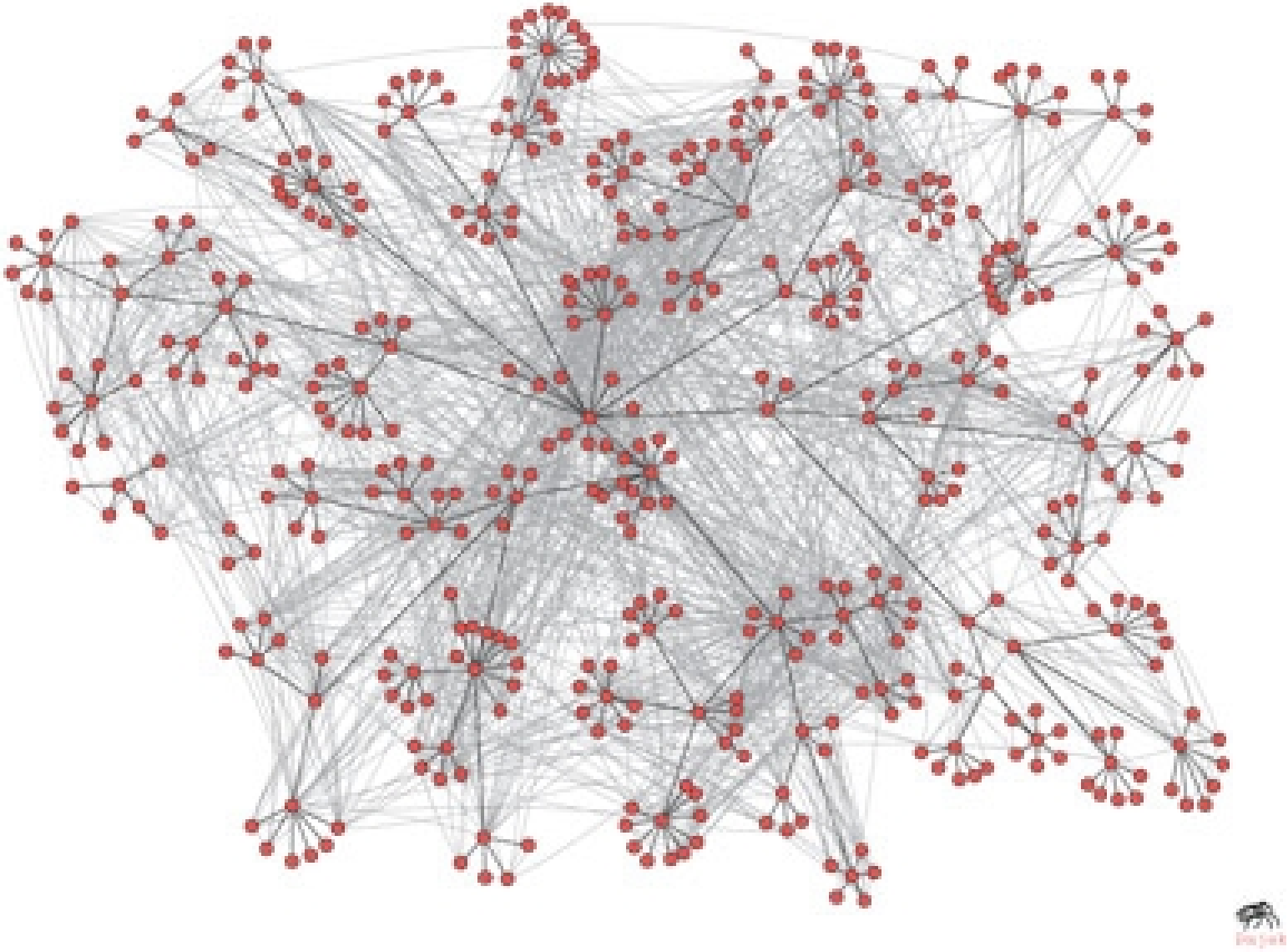
4 Bit



5 Bit



Soziale Netzwerke



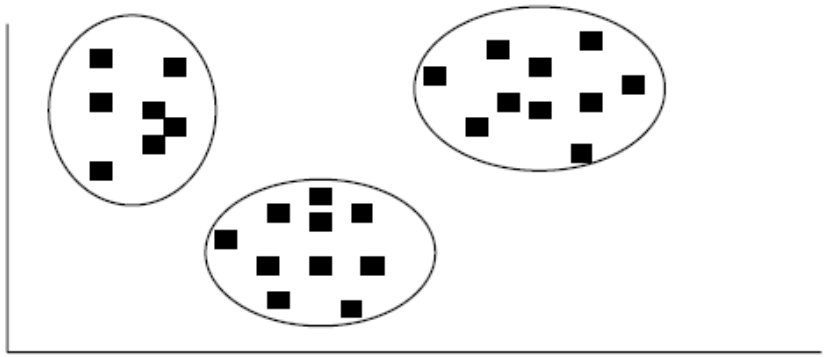
In a nutshell

Cluster Analyse

= Ähnlichkeits-, bzw Abstandsmaß,

\+ Kostenfunktion zur Evaluation einer Gruppierung

\+ Algorithmus, welcher die Kostenfunktion minimiert



Ähnlichkeit von Objekten

Essenzielle Frage: Wie messe ich den Abstand zweier Objekte ?

- Domain- und Problemabhängig
- sollte den Typ der Merkmale berücksichtigen:
  - quantitative, ordinale, kategorielle Daten
- kann zusätzlich Struktur der Daten berücksichtigen:
  - zeitliche, räumliche Position
- Wahl des Abstandsmaß beeinflusst Ergebnis der Cluster Analyse

Euklidische Abstand

Meist $x_i \in \mathbb{R}^n$

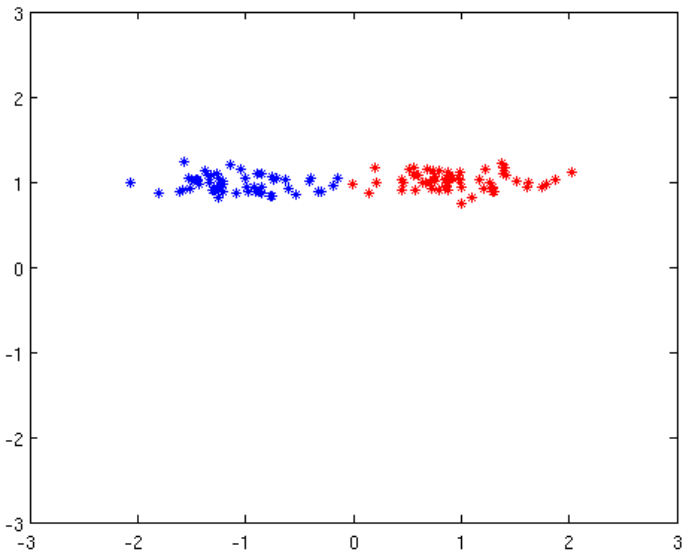
- kanonische Abstandsmaß ist Euklidische Abstand
  $D_1(x, x') = \|x - x'\| = \sqrt{\sum_{k=1}^{D}(x_k - x'_k)^2}$
- oder auch quadratischer Euklidischer Abstand
  $D_2(x, x') = \|x - x'\|^2 =< x - x', x - x' >$
- invariant gegenüber Verschiebung und Rotation aber nicht invariant gegenüber Skalierung

Beachte!!! Normalisierung der Daten (Mittelwertfrei, Einheitsvarianz) beinhaltet Verschiebung und Skalierung

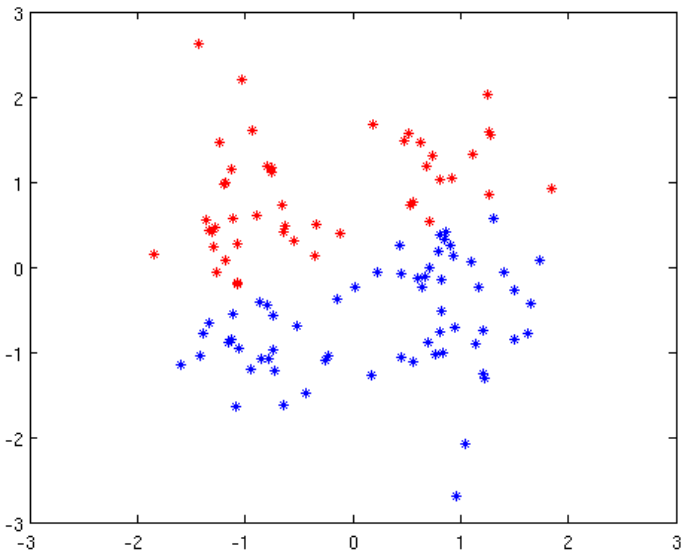
Normalisierung nicht immer hilfreich

Clusterlösung

vor Normalisierung



nach Normalisierung



verallgemeinerte Metrik

$x_i \in \mathbb{R}^n$ und sei $A \in GL(n)$ symmetrisch, positive definit

- allgemeine Metrik bzgl A

  $D_3(x, x') = \|x - x'\|_A = \sqrt{(x_k - x'_k)^T A (x_k - x'_k)}$

- A symm., pos.def. $\Rightarrow A = E \Lambda E^T$

  $\tilde{x} = \Lambda^{\frac{1}{2}} E^T x \Rightarrow D_3(x, x') = \|x - x'\|_A = \|\tilde{x} - \tilde{x}'\|$

- $A = cov(X)^{-1} \Rightarrow$ Mahalanobis-Distanz

  Translations- und Skalierungsinvariant

Beispiel Distanzen für Strings

$x_i \in \mathcal{A}^n$, Strings der Länge n

- Hamming Distanz: $D_4(x, x') = n - \sum_{k=1}^{n} \delta_{x_k, x'_k}$
- basierend auf Worthäufigkeit in Menge von Dokumenten $\mathcal{D}$

$$D_5(x, x') = 1 - \frac{\#\{D \in \mathcal{D} : x \in D \wedge x' \in D\}}{\#\{D \in \mathcal{D} : x \in D \vee x' \in D\}}$$

Beispiel: x_1 =’worte’, x_2 =’sorte’, x_3 =’sprache’

$$D_4(x_1, x_2) = 1 \text{ und } D_4(x_1, x_3) = 6$$

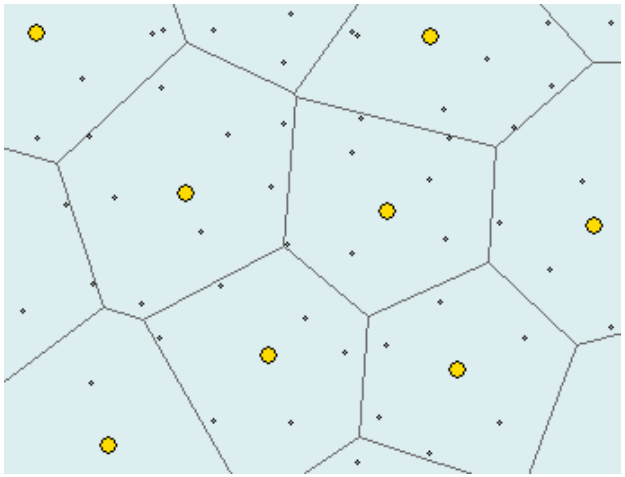
$$\text{aber } D_5(x_1, x_2) >> D_5(x_1, x_3)$$

Für Cluster Analysen ist die Wahl der Metrik ebenso entscheidend wie die Wahl des Algorithmus !!!

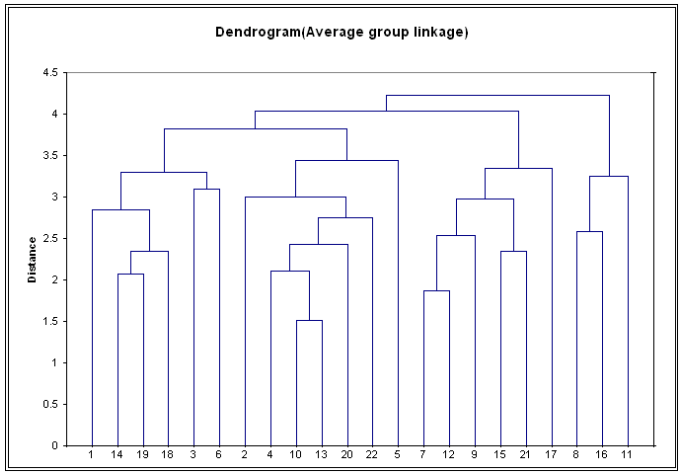
Cluster Algorithmen

Zwei Klassen von Cluster Algorithmen

- ’Flat’ - flache
  Konstruktion einer optimalen Partitionierung des Raumes bei gegebener Zahl von Partitionen



- Hierarchische
  Konstruktion einer geschachtelten Hierarchie von Partitionen



Der Standard Algorithmus

für flat clustering:

K-means

K-means: Idee

Grundlegende Idee von K-means:

- Partitioniere die Daten $x_1, ..., x_n \in \mathbb{R}^d$ als K Cluster
- repräsentiere jeden Cluster durch Prototypen μ_k
- Kostenfunktion : Summe der euklidischen Distanz jedes Punktes von seinem assoziierten Prototypen

$$J = \sum_{k=1}^{K} \sum_{x \in C_k} \|x - \mu_k\|^2$$

Minimiere der Kostenfunktion

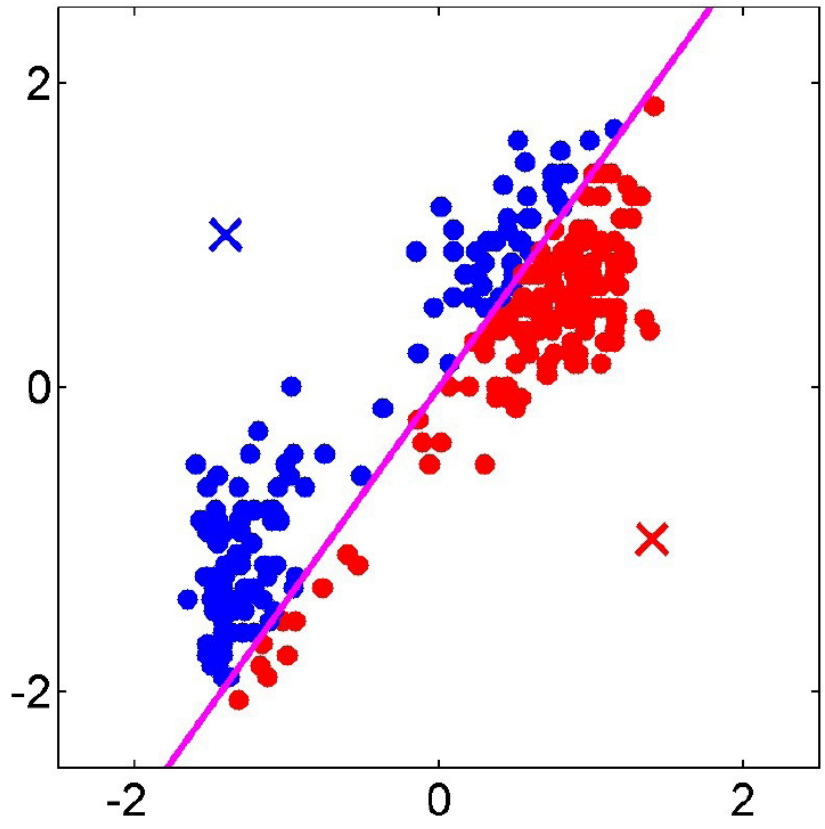
- Henne und Ei Problem
  - Wenn Prototypen μ_k bekannt, optimale Cluster-Zugehörigkeit berechenbar.
  - Wenn Zugehörigkeiten bekannt, optimale Prototypen berechenbar.
- *K*-means optimiert Kostenfunktion iterativ.
  - E-step: fixiere μ_k und minimiere *J* bzgl der Clusterzuordnung
  - M-step: fixiere Clusterzuordnung und minimiere *J* bzgl μ_k

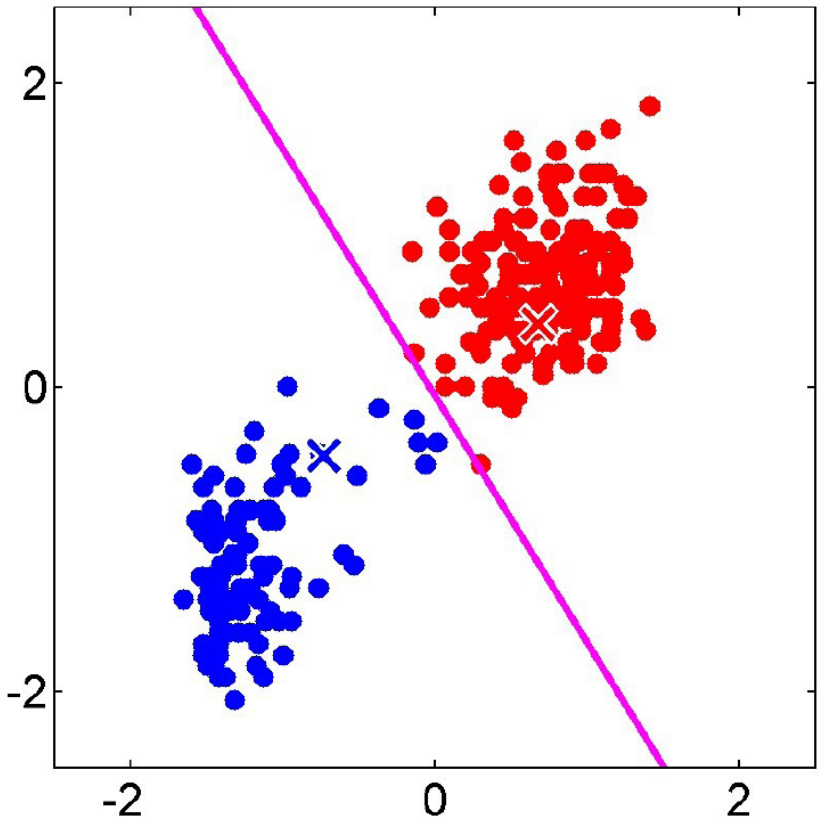
K-means: Algorithmus

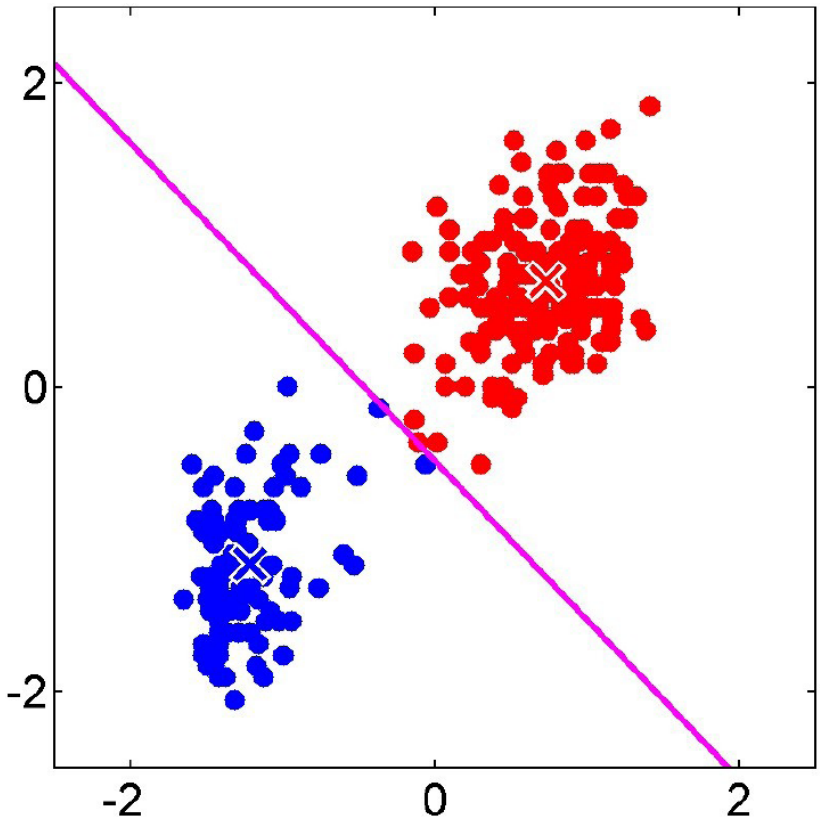
Input: Daten $X_1, \ldots, X_n \in \mathbb{R}^d$, Anzahl K der Cluster

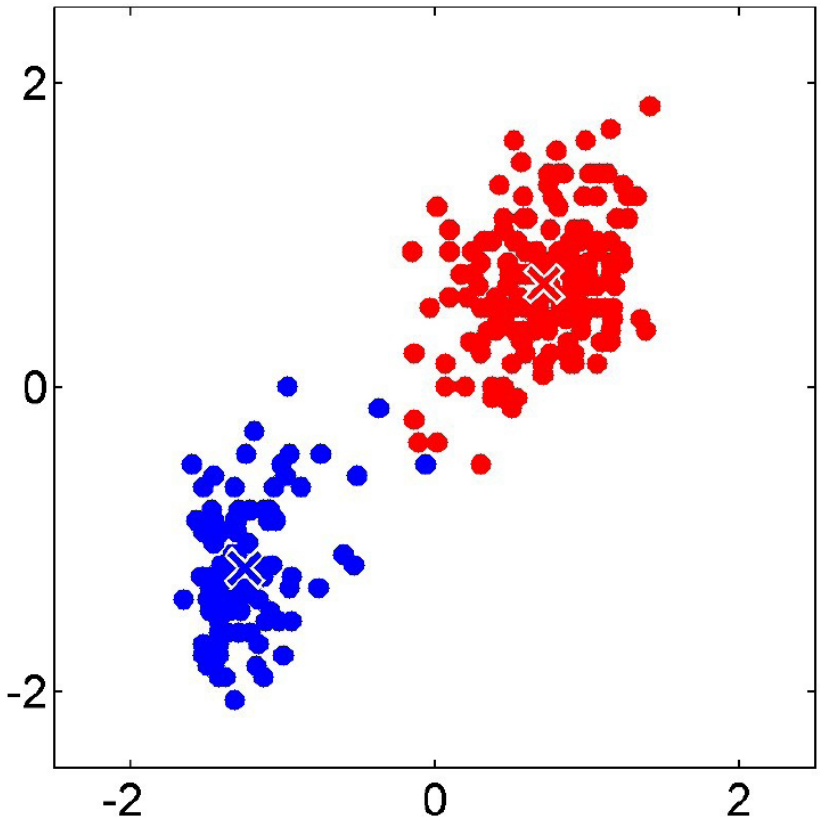
1. Initialisiere Prototypen $\mu_1^{(0)}, \ldots, \mu_K^{(0)}$ zufällig.
2. Wiederhole bis konvergiert:
   - 2.1 ordne jeden Datenpunkt dem nahesten Prototypen zu, dies definiert die Cluster $C_1^{(j+1)}, \ldots, C_K^{(j+1)}$
     $X_i \in C_k^{(j+1)} \Leftrightarrow \|X_i - \mu_k^{(j)}\|^2 \leq \|X_i - \mu_s^{(j)}\|^2, \ \forall s = 1, \ldots K$
   - 2.2 Berechne neue Prototypen als Clusterzentren
     $\mu_k^{(j+1)} = \frac{1}{|C_k^{(j+1)}|} \sum_{i \in C_k^{(j+1)}} X_i$

Output: Cluster $C_1, \ldots, C_K$.


2
0
-2
-2
0
2

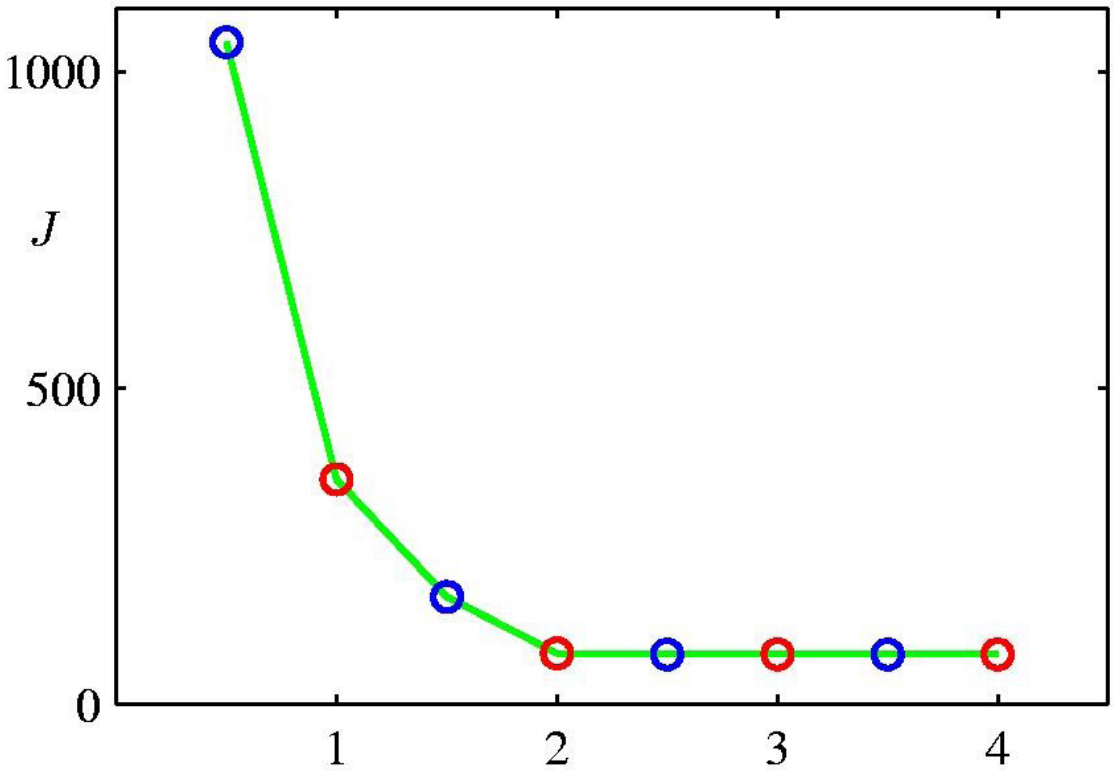

2
0
-2
-2
0
2


2
0
-2
-2
0
2


2
0
-2
-2
0
2

K-means: Trainingsfehler

Kostenfunktion nach jedem E und M step



K-means: Implementation (1)

Kostenfunktion nicht konvex $\Rightarrow$ lokale Minima

Heuristiken zur Verbesserung der Lösung:

- Restart mit unterschiedlichen Initialisierungen
- Verschmelze Cluster und füge neue hinzu
- Spalte Cluster mit hohen Kosten auf, entferne zufällig einen anderen Cluster
- Vertausche einzelne Punkte zwischen Clustern

K-means: Implementation (2)

Heuristiken für Zufällige Initialisierung

- Meist: wähle zufällig Datenpunkte $X_{i_k}, k = 1, ..., K$
- Ziehe zufällige Startpunkte aus $\mathbb{R}^d$
- Verwende Vorwissen, dass bestimmte Punkte zu verschiedenen Gruppen gehören
- *furthest first*-Algorithmus
  - Initialisiere 1. Zentrum zufällig
  - 2. Zentrum: Punkt am entferntesten vom 1. Zentrum
  - 3. Zentrum: entferntesten Punkt vom 1.& 2. Zentrum
  - Allgemein: Nächste Zentrum ist

$$\mu_{k+1} = \underset{X_i}{\operatorname{argmax}} \min_{s=1,...,k} d(X_i, \mu_s)$$

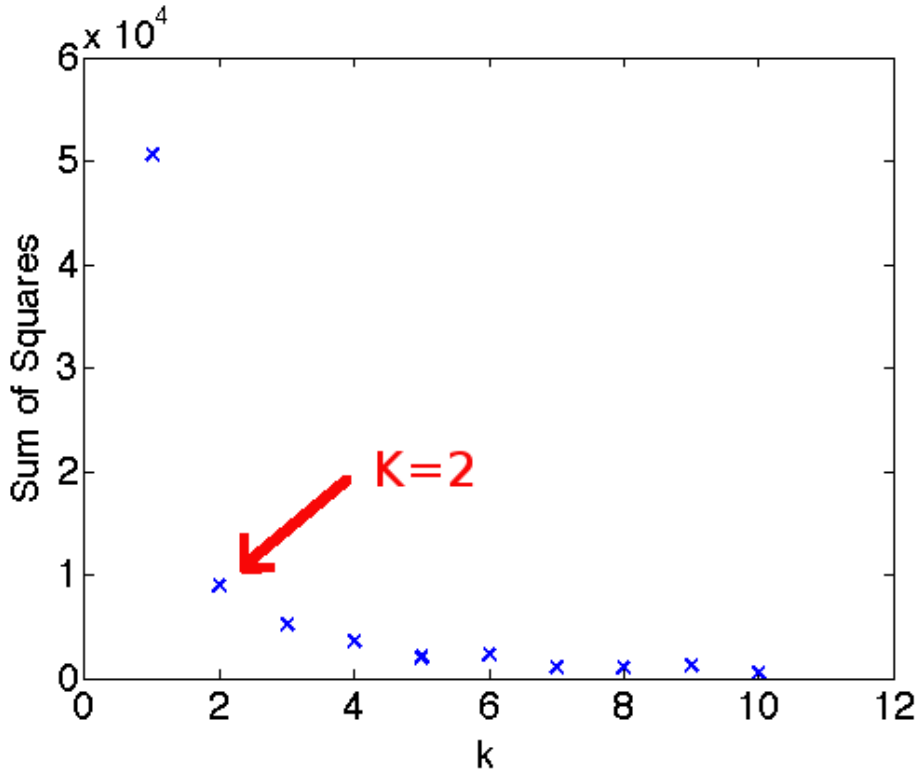
K-means: Implementation (3)

Wie viele Cluster?

- In seltenen Fällen apriori bekannt
- Wahl datengetrieben, aber die Kostenfunktion J wird allgemein kleiner mit zunehmender Zahl an Zentren
- Idee:
  - Sei K^* die korrekte Anzahl Cluster
  - $K < K^*$: Jeder Cluster enthält Punkte verschiedener natürlicher Gruppen
  - $K > K^*$: Einige natürliche Gruppen müssen aufgespalten worden sein
  - $\Rightarrow$ für $K < K^*$ verbessert sich die Kostenfunktion substanziell; für $K > K^*$ Verbesserung gering

K-means: Gap-Statistik

Kostenfunktion als Funktion von *K*



K-means: Eigenschaften

- Sensitiv gegenüber Outlier in den Daten
  Lösung: Outlier removal; Anpassung der Metrik
- Zuordnung der Datenpunkte zu genau einem Cluster
  Lösung: soft Assignment
  $\Rightarrow$ Gaussian Mixture Models (GMM)
- Annahme Cluster sind konvex $\Rightarrow$ schlechte Performanz bei nicht-konvexen Strukturen
  Alternativen: Hierarchisches Clustern, Spectral clustering

K-means: Kostenfunktion

Summe der quadratischen Fehler

$$J_1 = \sum_{k=1}^{k} \sum_{x \in C_k} \|x - m_k\|^2,$$

Minimum Varianz

$$J_2 = \frac{1}{2} \sum_{k=1}^{k} \frac{1}{|C_k|} \sum_{x \in C_k} \sum_{x' \in C_k} D(x, x')$$

Beide Kriterien sind identisch für $m_k = \frac{1}{|C_k|} \sum_{x \in C_k} x$

K-means: Scatter Kriterium (1)

$$m = \frac{1}{n}\sum_{i=1}^{n} x_i,\ m_k = \frac{1}{|C_k|}\sum_{x \in C_k} x,\ S_k = \sum_{x \in C_k}(x - m_k)(x - m_k)^T \tag{1}$$

$$\text{total scatter} \quad S_T = \sum_{x \in}(x - m)(x - m)^T$$

$$\text{between-cluster} \quad S_B = \sum_{k=1}^{K}|C_k|(m_k - m)(m_k - m)^T$$

$$\text{within-cluster} \quad S_W = \sum_{k=1}^{K} S_k$$

K-means: Scatter Kriterium (2)

$$tr(S_W) = \sum_{k=1}^{K} tr(S_k) = \sum_{k=1}^{K} \sum_{x \in C_k} \|x - m_k\|^2$$

Identisch zu *quadratischem Fehler* Kriterium

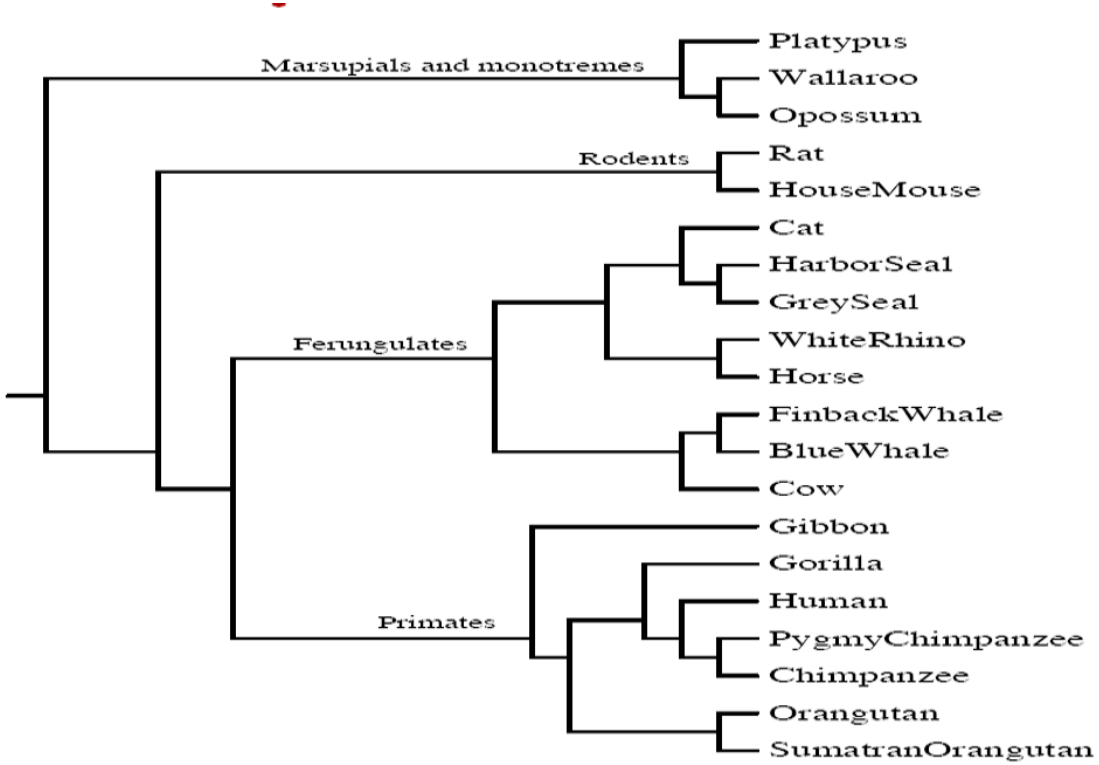
$$\begin{aligned} S_T &= S_W + S_B \\ tr(S_T) &= tr(S_W) + tr(S_B) \end{aligned}$$

$\Rightarrow$ Minimierung des *within-cluster* Kriterium äquivalent zu Maximierung des *between-cluster* Kriteriums

$$tr(S_B) = \sum_{k=1}^{K} |C_k| \cdot \|m_k - m\|^2$$

Hierarchisches Clustern

Ziel: bestimmen einer vollständigen Hierarchie, von Cluster und Teil-Cluster in Form eines Dendrogrammes



Hierarchisches Clustern: Strategien

Agglomerative (bottom-up) Strategie:

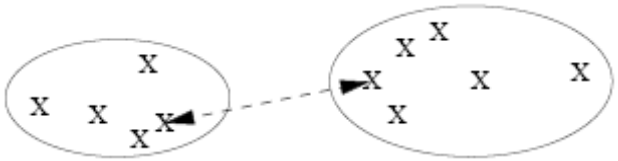
1. Start: jeder Punkt ist ein Cluster
2. Vereinige die zwei Cluster mit dem kleinsten Abstand
3. Wiederhole 2. bis ein Cluster übrig

'Divisive' aufteilende (top-down) Strategie:

1. Start: alle Punkte formen einen Cluster
2. Teile den am wenigsten zusammenhängenden Cluster in zwei Cluster
3. Wiederhole 2. bis alle Cluster unteilbar ($|C_k| = 1$)

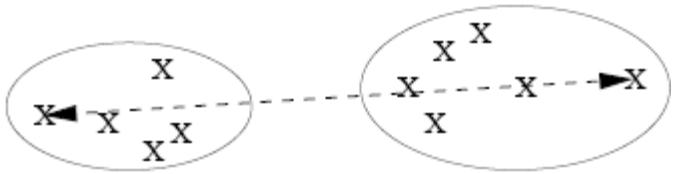
Abstand zweier Cluster

single linkage:



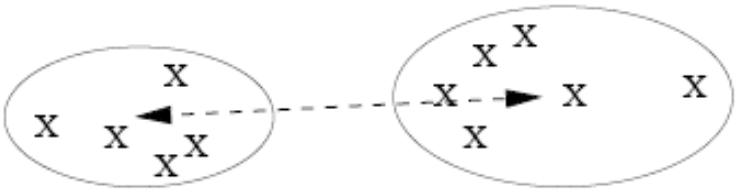
$$D(C, C') = \min_{x \in C, x' \in C'} D(x, x')$$

complete linkage:



$$D(C, C') = \max_{x \in C, x' \in C'} D(x, x')$$

averaged linkage:



$$D(C, C') = \frac{\sum_{x \in C, x' \in C'} D(x, x')}{|C| \cdot |C'|}$$

Linkage Algorithmus

Input:

- Matrix $D_{i,j}$ - Distanzen zwischen den Datenpunkte
- Funktion *dist* zur Berechnung des Abstandes zwischen Clustern

Initialisierung: $\mathcal{C}^{(0)} = \{C_1^{(0)}, \ldots, C_n^{(0)}\}$, mit $C_i^{(0)} = \{i\}$.
While Anzahl Cluster > 1:

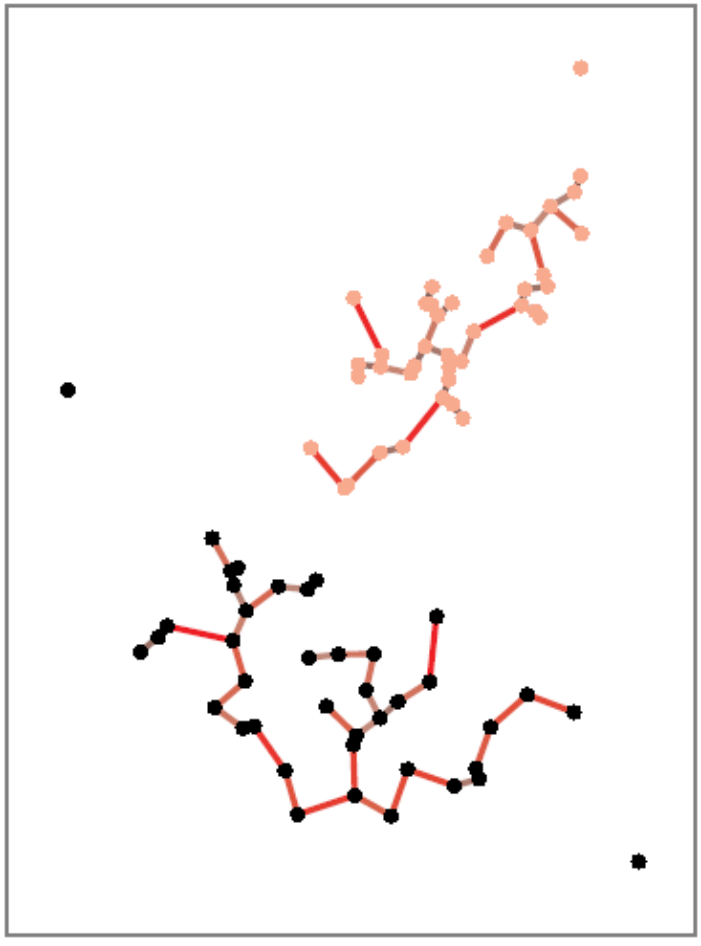
- bestimme die beiden Cluster mit kleinster Distanz
- vereinige beide Cluster zu einem

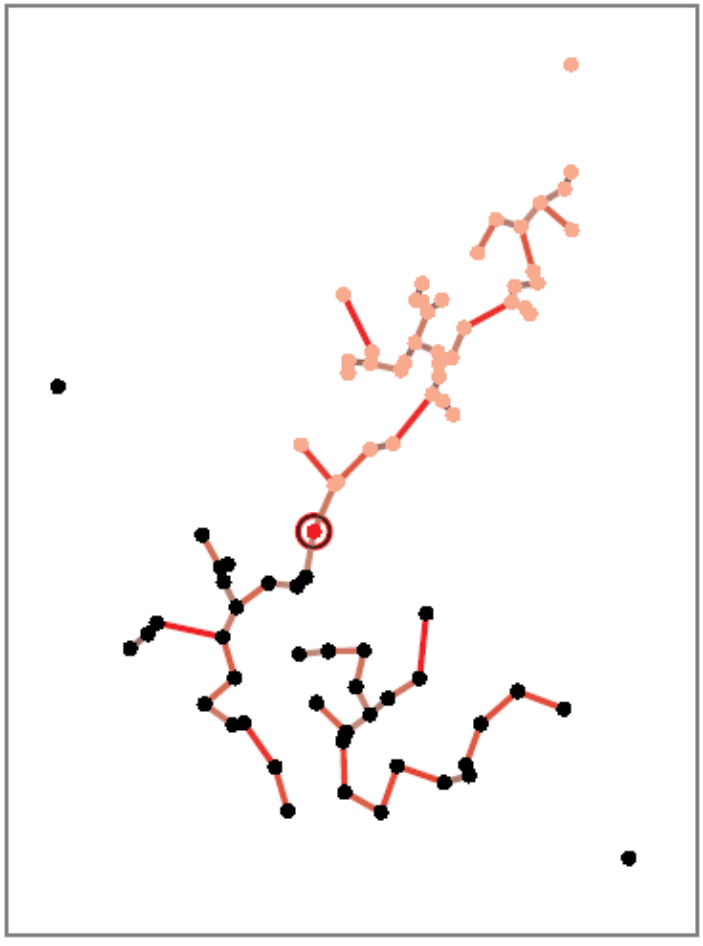
Output: Dendrogramm

Linkage Algorithmus: Eigenschaften

- *Single linkage* erzeugt tendenziell langgestreckte Cluster (Ketten)
- *Complete linkage* erzeugt tendenziell kompakte Cluster
- *Single linkage* erzeugt Minimal Spanning Tree
- Linkage ist sensitiv gegenüber einzelnen Datenpunkten

Sensitivität gegenüber einzelnen Datenpunkten





[Duda and Hart]

Literatur

- Clustering allgemein
  - Duda, Hart, and Stork, Pattern Classification, Chapter 10
  - Hastie, Tibshirani, and Friedman, The Elements of Statistical Learning, Chapter 14,
  - Bishop, Pattern Recognition and Machine Learning, Chapter 9
- Hierarchisches Clustern
  - Jardine and Sibson. Mathematical taxonomy. Wiley, London, 1971.
- K-means
  - Charles Elkans website http://cseweb.ucsd.edu/~elkan/