

---

# Text-Clustern

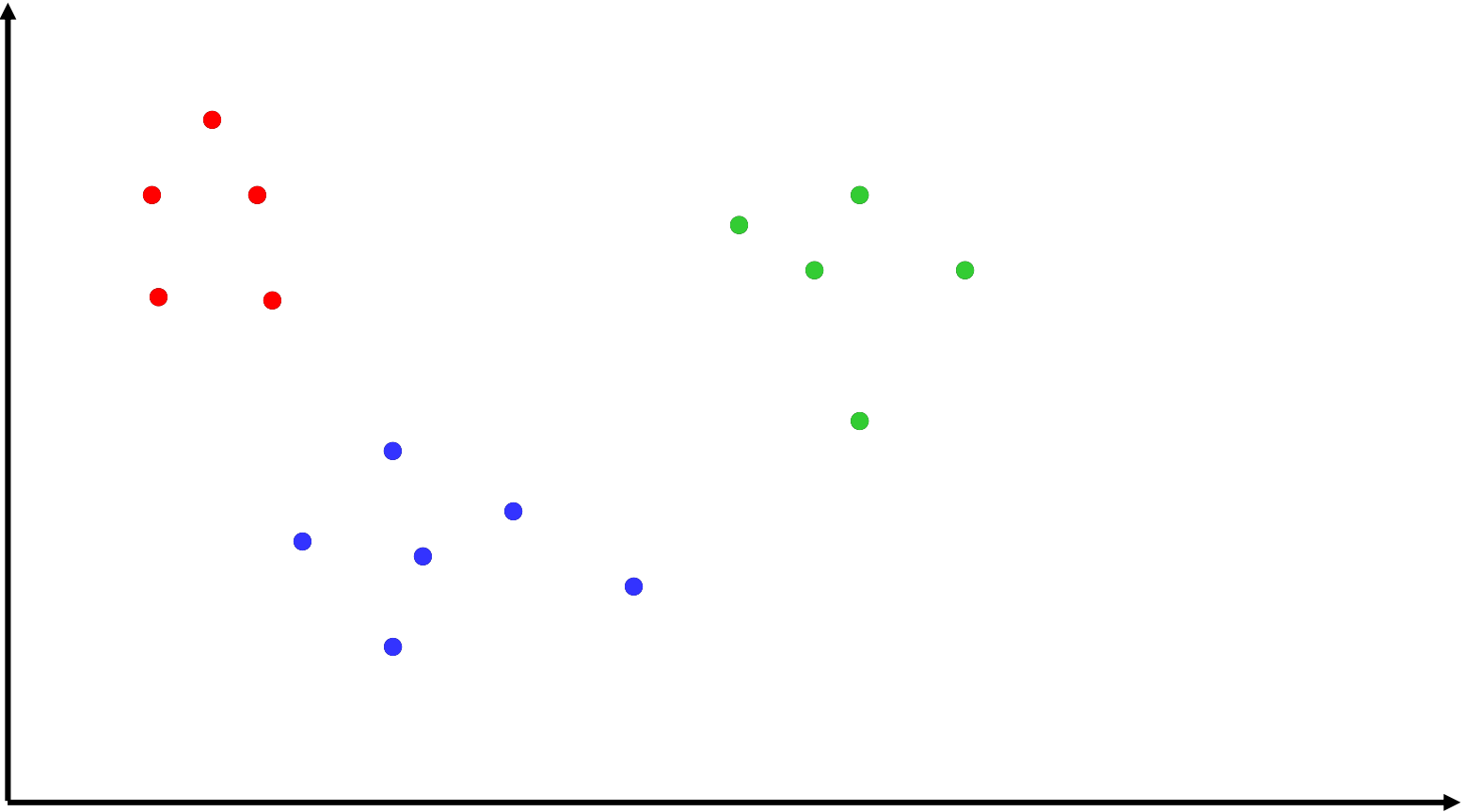
# Clustern

---

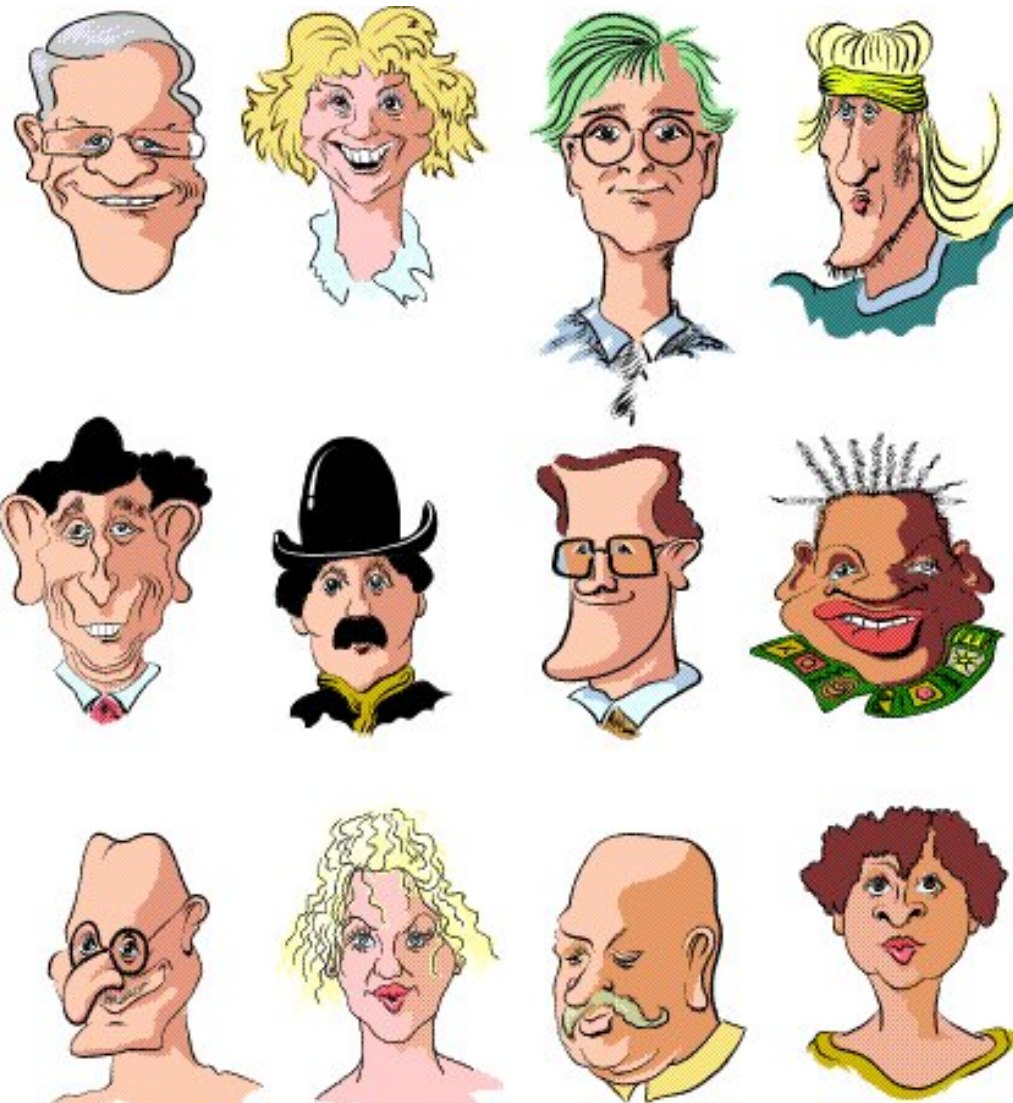
- Teile nicht kategorisierte Beispiele in disjunkte Untermengen, so genannte *Cluster*, ein, so dass:
  - Beispiele innerhalb eines Clusters sich sehr ähnlich
  - Beispiele in verschiedenen Clustern möglichst unterschiedlich sind.
- Entdeckt neue Kategorien in einer *unüberwachten* Weise: es werden keine Schlagwörter für die Kategorien vorab zur Verfügung gestellt (im Gegensatz zu Kategorisierung)

# Einführung Clustern

---



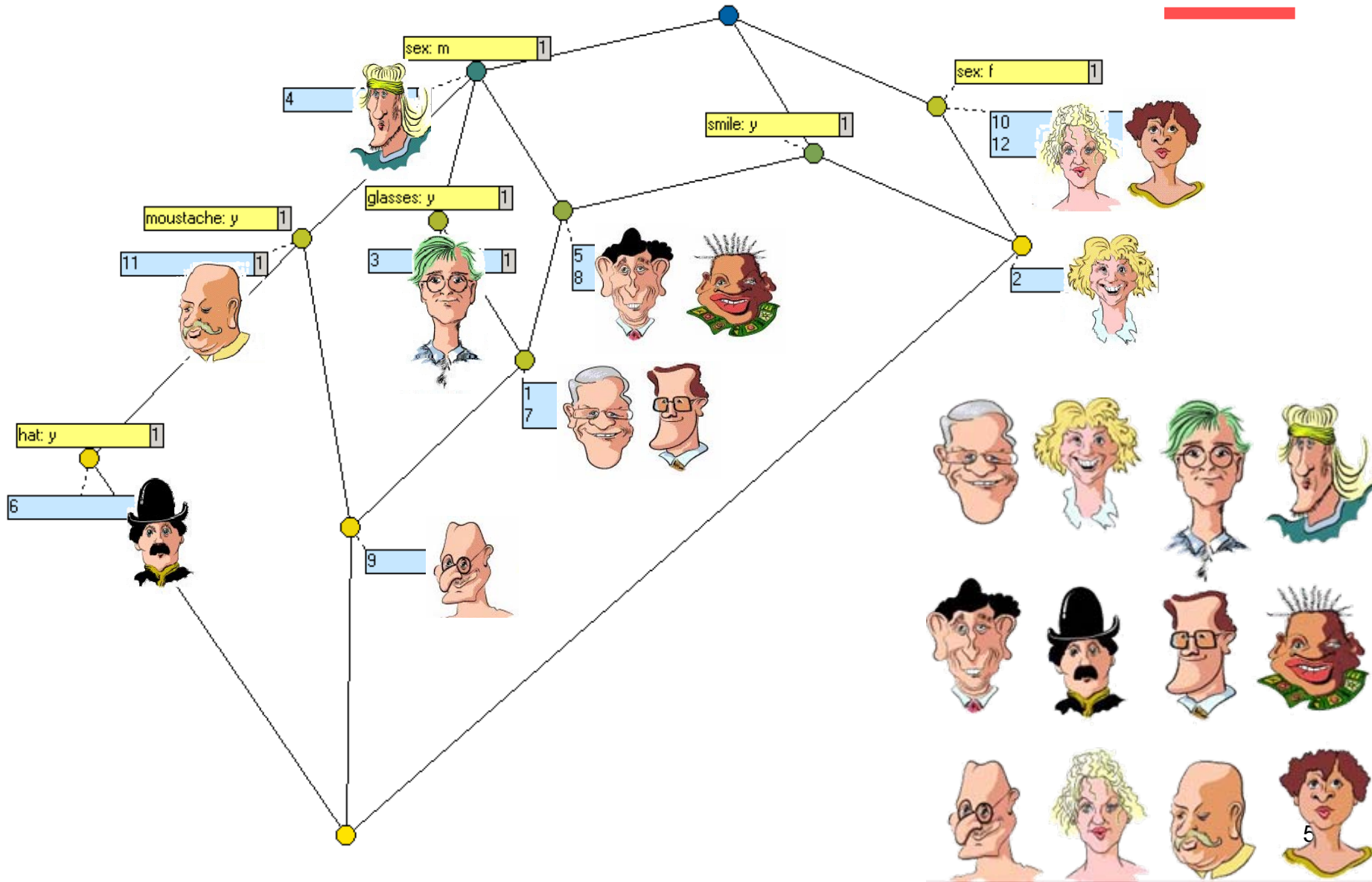
# Einführung Clustern



---

case	sex	glasses	moustache	smile	hat
1	m	y	n	y	n
2	f	n	n	y	n
3	m	y	n	n	n
4	m	n	n	n	n
5	m	n	n	y?	n
6	m	n	y	n	y
7	m	y	n	y	n
8	m	n	n	y	n
9	m	y	y	y	n
10	f	n	n	n	n
11	m	n	y	n	n
12	f	n	n	n	n

# Einführung Clustern



# Typen von Clustering-Verfahren

---

- Hierarchische Verfahren

- Parameter: Distanz- oder Ähnlichkeitsfunktion für Punkte und für Cluster
- bestimmt eine Hierarchie von Clustern, indem die jeweils ähnlichsten Cluster verschmolzen werden

- Partitionierende Verfahren

- Parameter: Anzahl  $k$  der Cluster, Distanzfunktion
- sucht ein „flaches“ Clustering in  $k$  Cluster mit minimalen Kosten

- Dichtebasierte Verfahren

- Parameter: minimale Dichte in einem Cluster, Distanzfunktion
- erweitert Punkte um ihre Nachbarn solange Dichte groß genug

- Andere Clustering-Verfahren

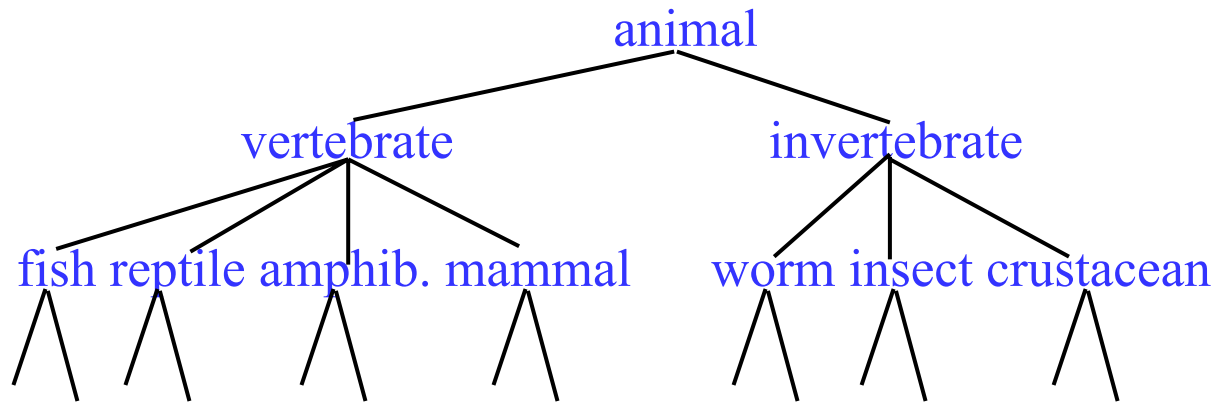
- Fuzzy Clustering
- Soft Clustering (EM)
- Graph-theoretische Verfahren
- neuronale Netze

Mehr Details zu Clusterverfahren gibt es in der KDD Vorlesung.

# Hierarchisches Clustern

---

- Bilde eine baum-basierte hierarchische Taxonomy (*Dendrogram*) aus einer Menge von Beispielen.



- Die rekursive Anwendung eines Standard-Cluster-Algorithmus führt auch zu hierarchischen Clustern (z.B. Bi-Sec-KMeans).

# Agglomeratives vs. divisives Clustern

---

- *Agglomerative* (bottom-up) Methoden beginnen mit je einem Beispiel als eigener Cluster und verbinden diese iterativ, um größere Cluster zu bilden.
- *Divisive* (*partitionierende, top-down*) trennen die Menge aller Beispiele in eine gegebene Anzahl von Cluster und wiederholen dies für jeden Cluster solange bis jeder Cluster nur noch ein Beispiel enthält.



# Hierarchisches Agglomeratives Clustern (HAC)

---

- Gegeben ist eine *Ähnlichkeitsfunktion* zur Bestimmung der Ähnlichkeit von zwei Objekten.
- Beginnt mit allen Objekten in einem separatem Cluster und verbindet dann mehrmals die beiden Cluster, die am ähnlichsten sind, bis nur noch ein Cluster da ist.
- Die Historie des Zusammenlegens bildet einen binären Baum oder eine Hierarchie.

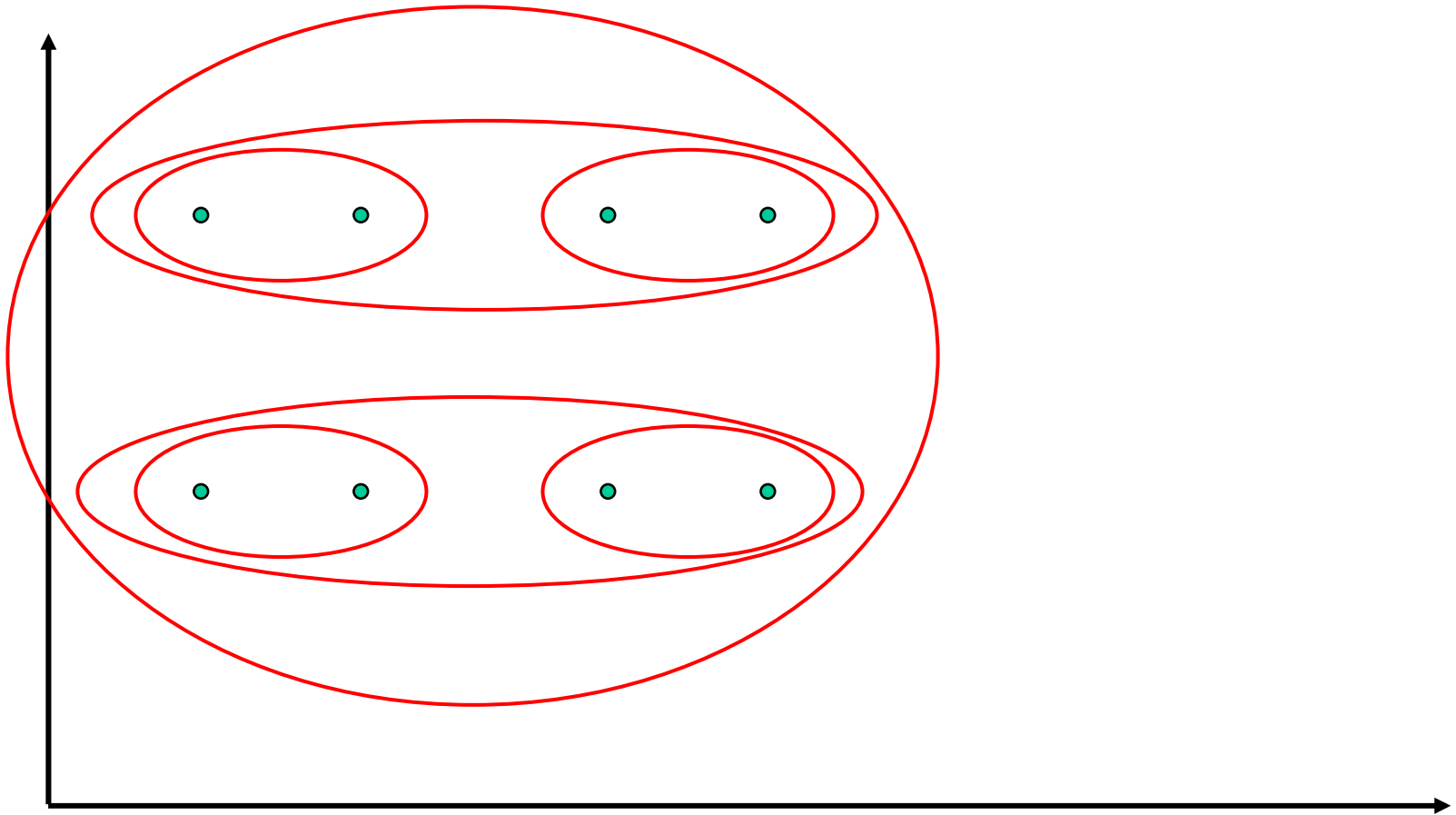
# Cluster-Ähnlichkeit

---

- Ähnlichkeitsfunktion, die die Ähnlichkeit von zwei Objekten bestimmt :  $sim(x,y)$ .
  - Kosinus Ähnlichkeit von Dokumentvektoren.
- Wie berechnet man die Ähnlichkeit von zwei Clustern, von denen jeder möglicherweise eine Vielzahl von Objekten enthält?
  - **Single Link**: Ähnlichkeit der zwei ähnlichsten Mitglieder.
  - **Complete Link**: Ähnlichkeit der zwei am wenigstens ähnlichen Mitglieder.
  - **Average Link**: Durchschnittsähnlichkeit zwischen allen Mitgliedern.

# Single Link am Beispiel

---



# Rechnerische Komplexität

---

- Bei der ersten Iteration müssen alle HAC-Methoden die Ähnlichkeit aller Paare von  $n$  Objekten berechnen. Aufwand:  $O(n^2)$
- Vor jedem der nachfolgenden  $n-2$  Zusammenlegungsschritte muss der Abstand zwischen dem neu erzeugten Cluster und allen anderen noch existierenden Clustern berechnet werden.
- Um bei einem Gesamtaufwand von  $O(n^2)$  zu bleiben, muss die Berechnung der Ähnlichkeit mit jedem Cluster in konstanter Zeit erfolgen.

# Nicht-Hierarchisches Clustern

---

- Typischerweise muß man bei den meisten Verfahren die Anzahl der gewünschten Cluster  $k$  angeben.
- Wähle willkürlich  $k$  Objekte als *Saat* (Ausgangspunkt) der Clusterung (einen pro Cluster).
- Bilde anfängliche Cluster, die auf dieser Saat basieren.
- Iteriere mehrfach und ordne Objekte Clustern neu zu, mit dem Ziel das Gesamtclusterergebnis zu verbessern.
- Stoppe, wenn das Clustern konvergiert oder nach einer festen Anzahl von Iterationen.

# K-Means

---

- Gegeben: Objekte werden durch reellwertige Vektoren repräsentiert.
- Die Cluster werden durch ihre *Schwerpunkte* (Zentroide) repräsentiert – dies ist der Mittelwert der Punkte eines Clusters,  $c$ :

$$\bar{\mu}(c) = \frac{1}{|c|} \sum_{\vec{x} \in c} \vec{x}$$

- Die Neuordnung von Objekten zu Clustern basiert auf dem Abstand zu den aktuellen Cluster-Zentroiden.

# Abstandsmaße

---

- Euklidischer Abstand ( $L_2$ -Norm):

$$L_2(\vec{x}, \vec{y}) = \sum_{i=1}^m (x_i - y_i)^2$$

- $L_1$ -Norm (Manhattan-Distanz):

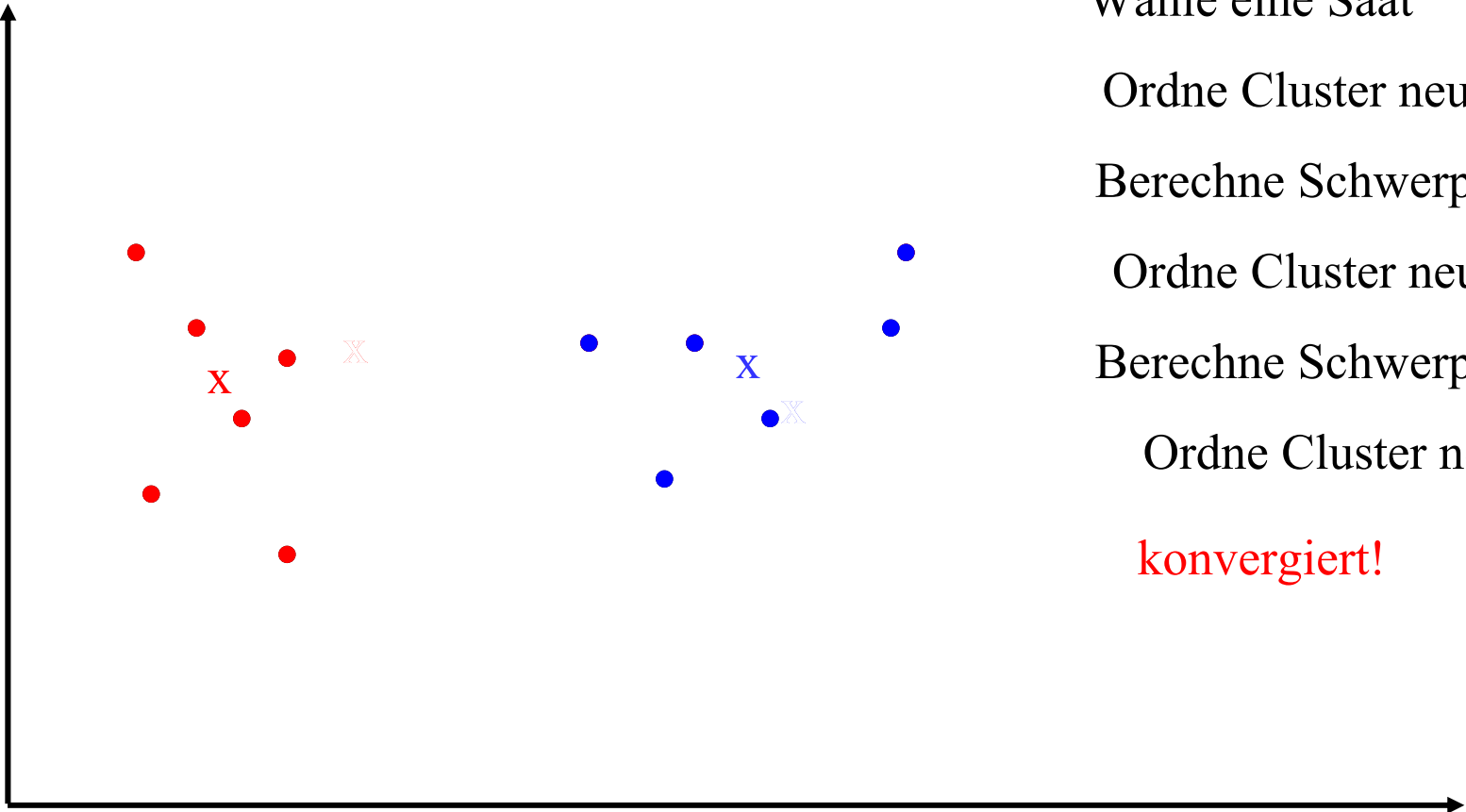
$$L_1(\vec{x}, \vec{y}) = \sum_{i=1}^m |x_i - y_i|$$

- Kosinus-Ähnlichkeit ( $\rightarrow$  wird zu einem Abstandsmaß durch Subtraktion von 1):

$$1 - \frac{\vec{x} \cdot \vec{y}}{|\vec{x}| \cdot |\vec{y}|}$$

# K-Means-Beispiel (K=2)

---



Wähle eine Saat

Ordne Cluster neu zu

Berechne Schwerpunkte

Ordne Cluster neu zu

Berechne Schwerpunkte

Ordne Cluster neu zu

**konvergiert!**



# Zeit-Komplexität

---

- Der Aufwand zur Berechnung des Abstandes zwischen zwei Objekten sei  $O(m)$ , wobei  $m$  die Dimensionalität der Vektoren ist.
- Neuzuordnung von Clustern:  $O(kn)$  Abstandsberechnungen, oder  $O(knm)$ .
- Zentroidberechnung: Jeder Objektvektor wird einmal zu seinem Schwerpunkt addiert:  $O(nm)$ .
- Die letzten zwei Schritte werden jeweils einmal pro Iteration durchgeführt:  $O(Iknm)$ ,  $I$  ist die Anzahl von Iterationen.
- Linear in allen relevanten Faktoren, wobei von einer festen Anzahl von Iterationen ausgegangen wird,
- KMeans ist effizienter als HAC ( $O(n^2)$ ).

# Text Clustering

---

- HAC und K-Means werden zum Clustern von Texten wie folgt angewendet:
- Typischer Weise nutzt man einen *normalisierten*, TF/IDF-gewichteten Vektor und die Kosinus-Ähnlichkeit.
- Die Berechnung wird für dünn besetzte Vektoren optimiert.
- Anwendungen:
  - Bei einer typischen Suchanfrage werden Dokumente des gleichen Clusters passend zu der ursprünglichen Antwortmenge zurückgeliefert, um so den Recall zu erhöhen.
  - Clustern der Suchergebnisse um besser organisierte Ergebnisse dem Anwender anbieten zu können. (z.B. <http://de.vivisimo.com/>).
  - Die automatische Erstellung von taxonomischen Hierarchien für eine Menge von Dokumenten mit dem Ziel das Browsing zu erleichtern. (z.B. Yahoo & DMOZ).