

Text Clustern

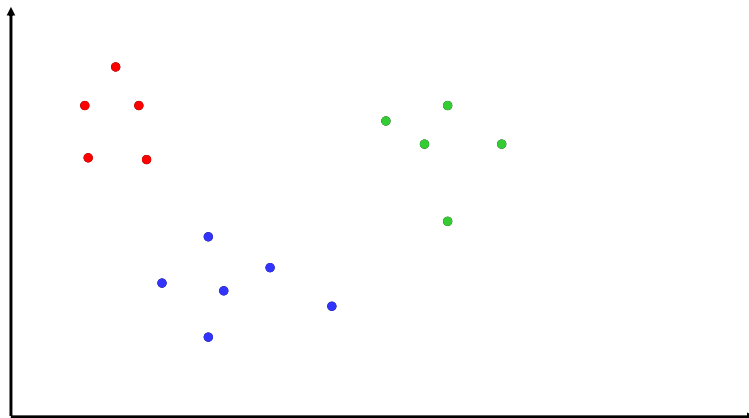
1

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 sehr unterschiedlich sind.
- Entdecke neue Kategorien in einer *unüberwachten* Art
- es werden keine Schlagwort für die Kategorien vorab zur Verfügung gestellt

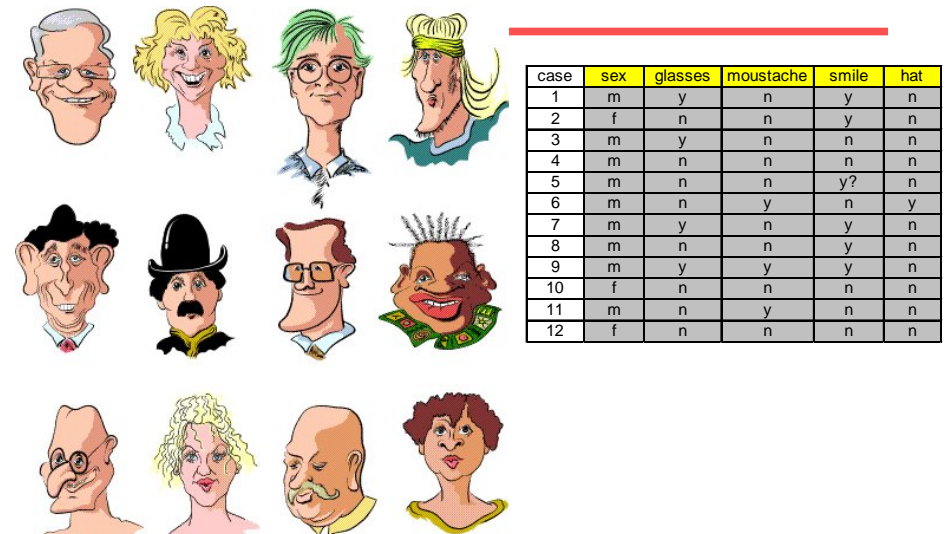
2

Einführung Clustern



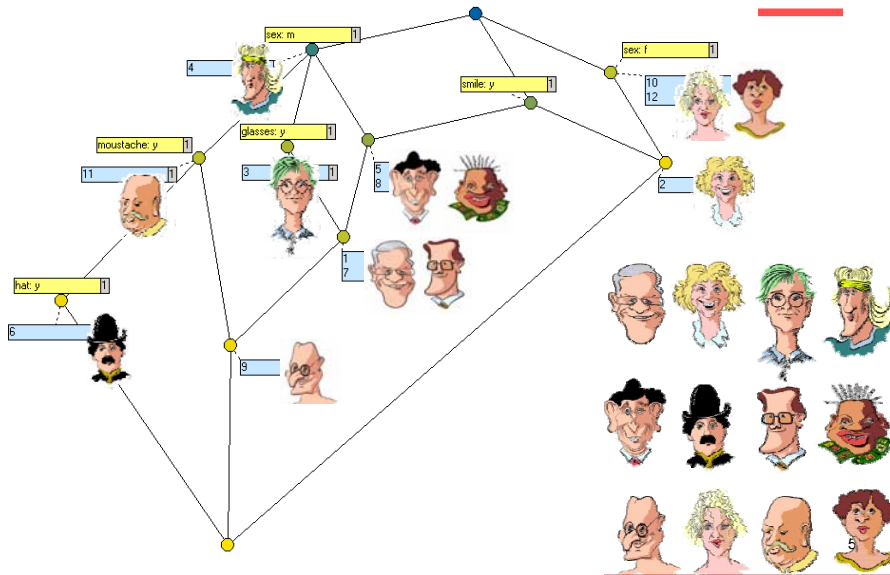
3

Einführung Clustern



4

Einführung Clustern



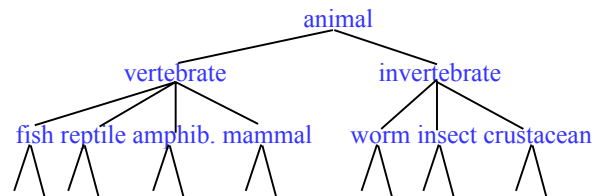
Typen von Clustering-Verfahren

- **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.

6

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).

7

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.

8

Hierarchisches Agglomeratives Clustern (HAC)

- Gegeben ist eine *Ähnlichkeitsfunktion* zur Bestimmung der Ähnlichkeit von zwei Objekten.
- Beginnt mit allen Objekten in einem separaten 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.

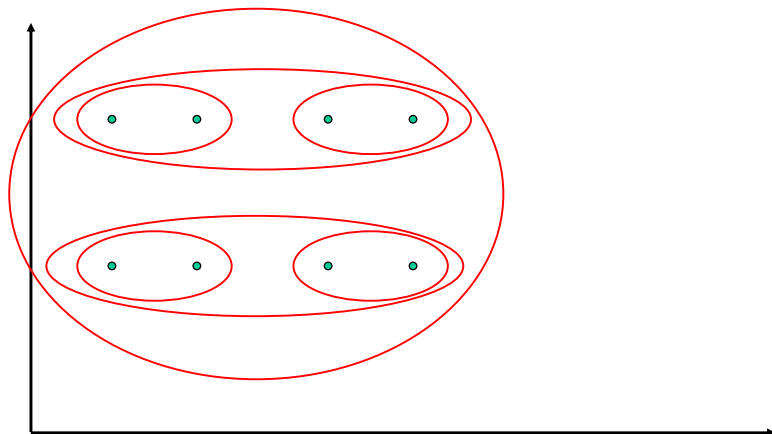
9

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.

10

Single Link am Beispiel



11

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.

12

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.

13

K-Means

- Gegeben: Objekte werden durch reelle-wertige Vektoren repräsentiert.
- Die Cluster basieren auf dem *Schwerpunkt* (Centroid) - dem Mittelwert von Punkten eines Cluster, c :

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

- Die Neuuzuordnung von Objekten zu Clustern basiert auf dem Abstand zu den aktuellen Cluster-Centroiden.

14

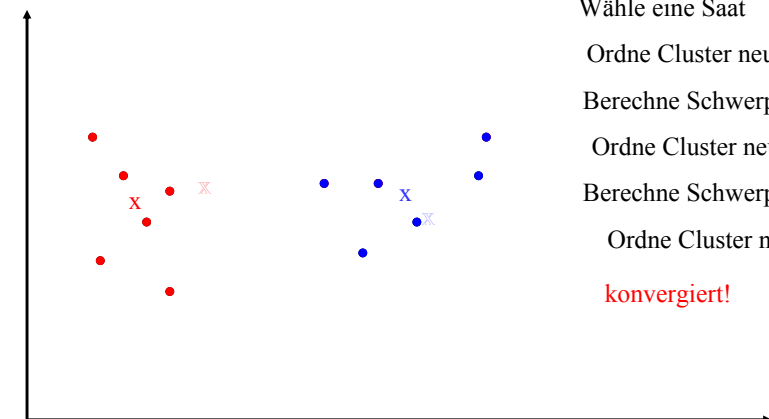
Abstandsmaße

- Euklidischer Abstand (L_2 Norm):
$$L_2(\vec{x}, \vec{y}) = \sum_{i=1}^m (x_i - y_i)^2$$
- L_1 Norm:
$$L_1(\vec{x}, \vec{y}) = \sum_{i=1}^m |x_i - y_i|$$
- Kosinus Ähnlichkeit (\rightarrow Abstandsmaß durch Subtraktion von 1):

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

15

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!**

16

Zeit Komplexität

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

17

Text Clustering

- HAC und K-Means wurden 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).

18