



Skript zur Vorlesung
Datenbanksysteme II
Sommersemester 2006

Kapitel 5: Einführung in Multimedia-Datenbanken

Vorlesung: Christian Böhm

Übungen: Elke Achtert, Peter Kunath, Alexey Pryakhin

Skript © 2006 Christian Böhm

<http://www.dbs.informatik.uni-muenchen.de/Lehre/DBSII>

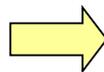


Multimedia-Datenbanken

- **Persistente Speicherung** von Mediendaten, z.B.
 - Text-Dokumente
 - Vektorgraphik, CAD
 - Bilder, Audio, Video
- Unterstützung von effizientem **Information-Retrieval**

Beispiel Bildsuche

„Finde alle zum Anfrageobjekt ähnlichen Bilder“



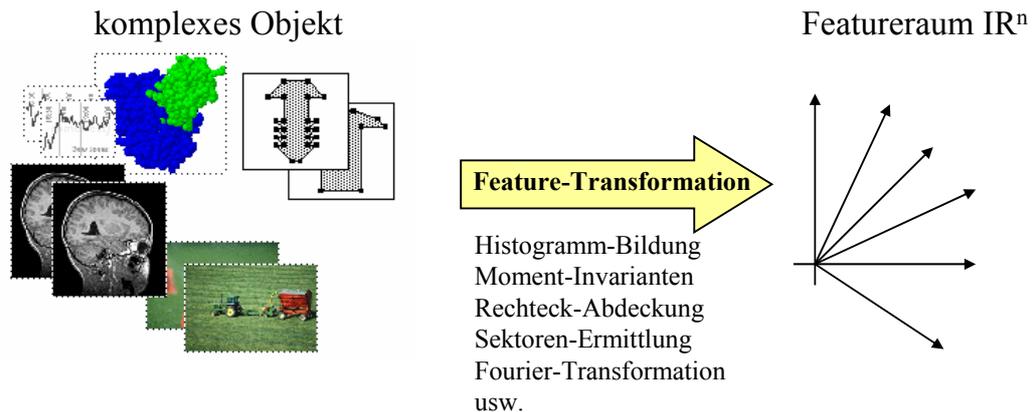
Was genau
zeichnet diese
Bilder aus?



Featurebasierte Ähnlichkeit (1)

Feature-Transformation

- Definition bzw. Auswahl geeigneter numerischer Merkmale (*Features*), die für die Unterscheidung (Klassifikation, Ähnlichkeit) der Multimedia-Objekte relevant sind
- Wichtigste Eigenschaft: Ähnlichkeit der Objekte entspricht geringem Abstand der Feature-Vektoren

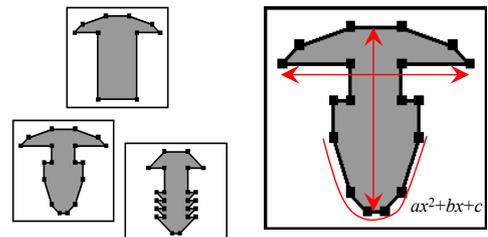


Featurebasierte Ähnlichkeit (2)

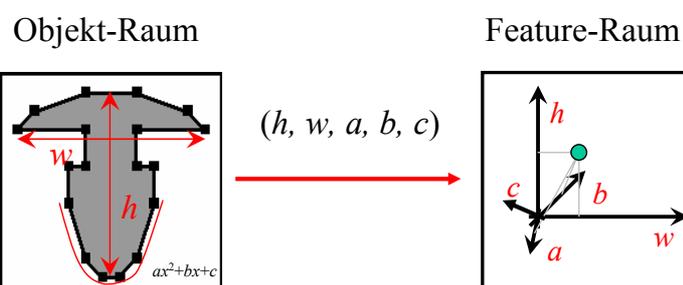
- Beispiel: CAD-Zeichnungen

- Mögliche Merkmale:

- Höhe h
- Breite w
- Krümmungs-Parameter (a, b, c)



- Zusammenfassen der ausgewählten Merkmale zu Feature-Vektoren:

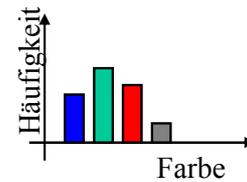




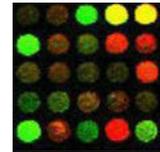
Featurebasierte Ähnlichkeit (3)

- Weitere Beispiele für Features

- Bilddatenbanken:
Farbhistogramme



- Gen-Datenbanken:
Expressionslevel



- Text-Datenbanken:
Begriffs-Häufigkeiten



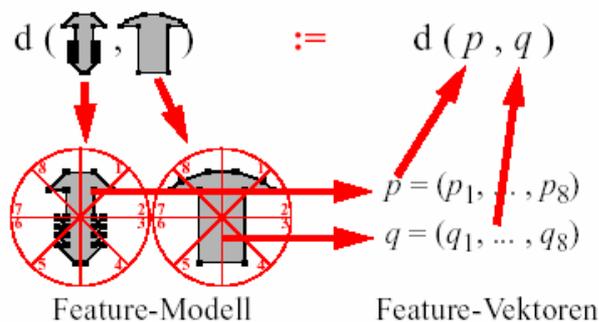
| | |
|---------|----|
| Data | 25 |
| Mining | 15 |
| Feature | 12 |
| Object | 7 |
| ... | |



Featurebasierte Ähnlichkeit (4)

Distanzbasiertes Ähnlichkeitsmaß

- Ähnlichkeit der Objekte ist durch die Distanz der zugeordneten Feature-Vektoren definiert
- Distanz im Feature-Raum steht für Unähnlichkeit der Objekte





Klassen von Distanzmaßen

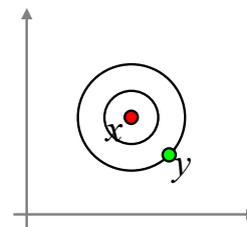
| $d : \mathcal{R}^n \times \mathcal{R}^n \rightarrow \mathcal{R}_0^+$, $x, y, z \in \mathcal{R}^n$ | Distanz- funktion | Semi- Metrik | Metrik |
|---|----------------------|-----------------|----------|
| reflexiv $x = y \Rightarrow d(x, y) = 0$ | X | X | X |
| symmetrisch $d(x, y) = d(y, x)$ | X | X | X |
| strikt $d(x, y) = 0 \Rightarrow x = y$ | | X | X |
| Dreiecksungleichung $d(x, z) \leq d(x, y) + d(y, z)$ | | | X |



Beispiele für Distanzfunktionen (1)

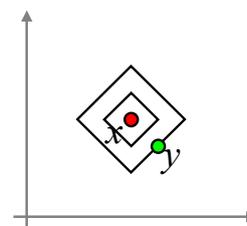
- **Euklidische Distanz** (L_2 -Metrik):
Natürliches Distanzmaß

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$



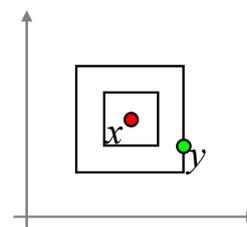
- **Manhattan-Distanz** (L_1 -Metrik):
Die Unähnlichkeiten der einzelnen Merkmale werden direkt addiert

$$d(x, y) = \sum_{i=1}^n |x_i - y_i|$$



- **Maximums-Distanz** (L_∞ -Metrik):
Die Unähnlichkeit des am wenigsten ähnlichen Merkmals zählt

$$d(x, y) = \max \{ |x_i - y_i|, i = 1 \dots n \}$$





Beispiele für Distanzfunktionen (2)

- Verallgemeinerung: **Allgemeine L_p -Metrik**

$$d(x, y) = \sqrt[p]{\sum_{i=1}^n |x_i - y_i|^p}$$

- **Gewichtete Euklidische-Distanz**

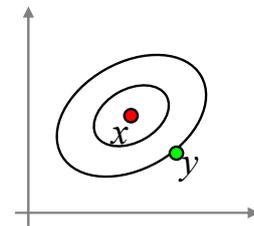
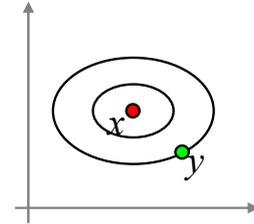
Benutzer kann Gewichte für einzelne Dimensionen vorgeben

$$d(x, y) = \sqrt{\sum_{i=1}^n w_i \cdot (x_i - y_i)^2}$$

- **Quadratische Formen**

Gewichtungsmatrix A erlaubt gemeinsame Gewichtung der verschiedenen Merkmale

$$d(x, y) = \sqrt{(x - y) \cdot A \cdot (x - y)^T}$$

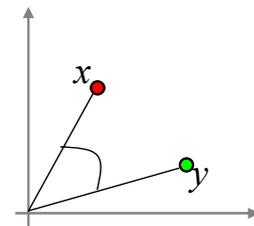


Beispiele für Distanzfunktionen (3)

- **Cosinus-Distanz**

$$d(x, y) = 1 - \frac{\langle x, y \rangle}{\|x\| \cdot \|y\|}$$

- Misst den Cosinus des Winkels zwischen x und y
- **Keine** (Semi-)Metrik im \mathbb{R}^n
- Repräsentiert strukturelle Unähnlichkeit oder strukturelle Distanz
- Sinnvoll z.B. für den Abstand von Wortvektoren im Textmining, dann gilt: $d(x, y) \in [0, 1]$ (nur positive Anzahl von Wörtern)



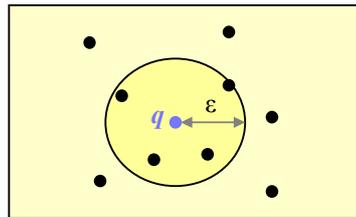


Typen von Ähnlichkeitsanfragen (1)

Basis : Objektmenge O , Distanzfunktion $dist : O \times O \rightarrow \mathfrak{R}_0^+$, Datenbank $DB \subseteq O$

• Bereichsanfrage

- Anfrageparameter: Anfrageobjekt q , max. Ähnlichkeitsabstand ε
- Ergebnismenge: $sim_\varepsilon(q) = \{o \in DB \mid d(o, q) \leq \varepsilon\}$
- Anzahl Ergebnisse: im vorhinein unbekannt, zwischen 0 und $|DB|$
- Ergebnisbereich: spezifizierter Bereich ε



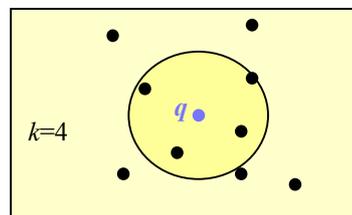
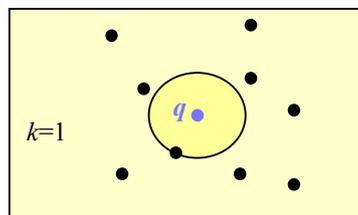
Typen von Ähnlichkeitsanfragen (2)

• Nächste-Nachbarn-Anfrage

- Anfrageparameter: Anfrageobjekt q
- Ergebnismenge: $NN(q) = \{o \mid \forall o' \in DB : d(o, q) \leq d(o', q)\}$
- Anzahl Ergebnisse: mind. 1
- Ergebnisbereich: im vorhinein unbekannt, $\varepsilon_1 = \min \{d(o, q) \mid o \in DB\}$

• k -Nächste-Nachbarn-Anfrage

- Anfrageparameter: Anfrageobjekt q ,
Anzahl gewünschter Ergebnisse k
- Ergebnismenge: kleinste Menge $NN_k(q) \subseteq DB$ mit $|NN_k(q)| \geq k$ für die gilt:
 $\forall o \in NN_k(q) : \forall o' \in DB - NN_k(q) : d(o, q) < d(o', q)$
- Anzahl Ergebnisse: mind. k
- Ergebnisbereich: im vorhinein unbekannt, $\varepsilon_k = \max \{d(o, q) \mid o \in NN_k(q)\}$





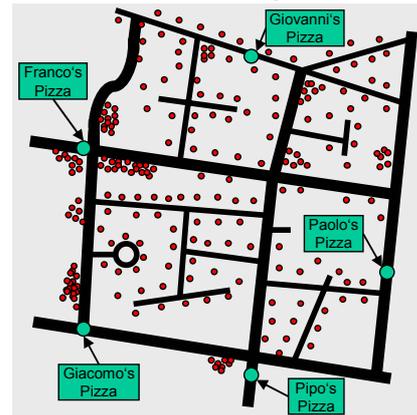
Typen von Ähnlichkeitsanfragen (3)

- **Reverse k -Nächste-Nachbarn-Anfrage**

- Anfrageparameter: Anfrageobjekt q , Parameter k
- Ergebnismenge: $RNN_k(q) = \{o \in DB \mid q \in NN_k(o)\}$
- Anzahl Ergebnisse: unbekannt, max. $|DB|$
- Anwendungen der Reverse k -Nächste-Nachbarn-Anfrage:



Planung von Spielzügen



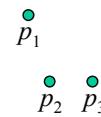
Werbung



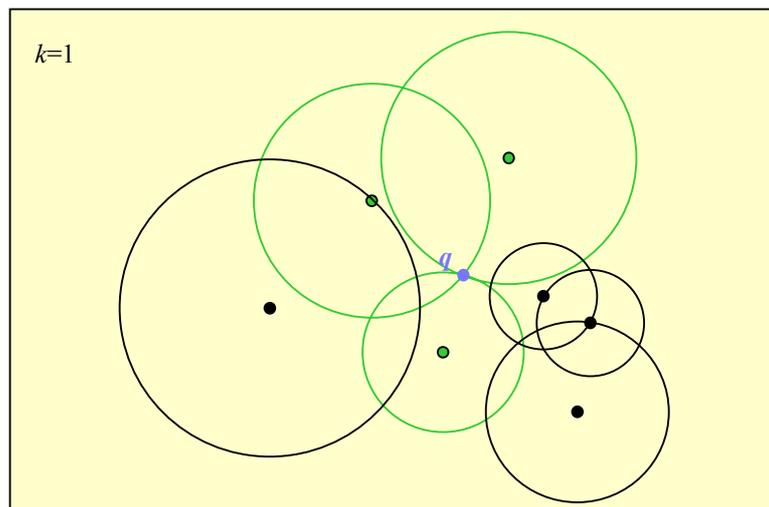
Typen von Ähnlichkeitsanfragen (4)

- Zusammenhang zwischen NN und RNN

- NN ist keine symmetrische Relation
 - $y \in NN(x) \not\Rightarrow x \in NN(y)$
 - $y \in NN(x) \not\Rightarrow y \in RNN(x)$
- RNN ist ein „eigenständiges“ Problem



| | NN | RNN |
|-------|-----------|----------------|
| p_1 | $\{p_2\}$ | $\{\}$ |
| p_2 | $\{p_3\}$ | $\{p_3, p_1\}$ |
| p_3 | $\{p_2\}$ | $\{p_2\}$ |





Typen von Ähnlichkeitsanfragen (5)

• Inkrementelles Ranking (*Give-me-more Query*)

– Ablauf

- Spezifikation eines Anfrageobjektes q beim Start.
- Wiederholte Aufrufe der Funktion $getnext(k_i)$, die jeweils die nächsten k_i Ergebnisse liefern, bis die gewünschte Ergebnismenge erreicht ist.
- Der Inhalt der DB wird also (partiell) aufgezählt, und zwar aufsteigend nach dem Abstand zum Anfrageobjekt, d.h. für zwei Objekte o_i und o_j gilt:

$$\forall i, j \in \{1, \dots, N\} : i < j \Rightarrow d(o_i, q) \leq d(o_j, q)$$

– Anfrageparameter: Anfrageobjekt q , Aufrufe von $getnext(k_i)$

– Ergebnismenge: $NN_q(k)$ mit $k = \sum_{i=1}^n k_i$ für n Aufrufe von $getnext(k_i)$

– Anzahl Ergebnisse: $k = \sum_{i=1}^n k_i$ für n Aufrufe von $getnext(k_i)$

– Ergebnisbereich: im vorhinein unbekannt, $\epsilon_k = \max \{d(o, q) \mid o \in NN_q(k)\}$



Bewertung von Methoden zur Ähnlichkeitssuche

| | | |
|-----------------------|------------------|--------------------|
| | erwünscht | unerwünscht |
| gefunden | richtig positive | falsch positive |
| nicht gefunden | falsch negative | richtig negative |

• **Recall:** Wie viele der erwünschten Objekte wurden gefunden?

$$\frac{rp}{rp + fn} = \frac{\text{gefundene erwünschte Objekte}}{\text{alle erwünschten Objekte}}$$

• **Precision:** Wie viele der gefundenen Objekte sind erwünscht?

$$\frac{rp}{rp + fp} = \frac{\text{gefundene erwünschte Objekte}}{\text{alle gefundenen Objekte}}$$

• **Sensitivität:** WS, dass Test für gewünschtes Objekt positiv verläuft (=Recall)

$$\frac{rp}{rp + fn} = \frac{\text{richtig positiv}}{\text{alle erwünschten Objekte}}$$

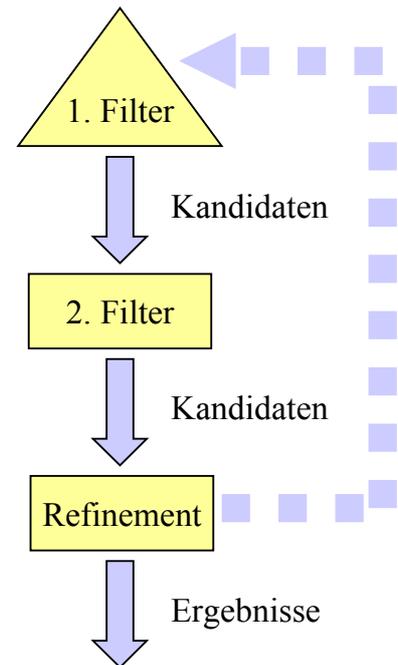
• **Spezifität:** WS, dass Test für unerwünschtes Objekt negativ verläuft

$$\frac{rn}{rn + fp} = \frac{\text{richtig negativ}}{\text{alle unerwünschten Objekte}}$$



Mehrstufige Anfragebearbeitung (1)

- **Filterschritt(e):**
 - Filter ermittelt Kandidatenmenge (*mögliche* Ergebnisse) mit Hilfe einer *Filterdistanz*
 - Filterdistanz sollte billiger sein als exakte Distanz
 - Weitere Filter schränken Kandidatenmenge möglichst weiter ein
- **Verfeinerungsschritt (Refinement)**
 - Ermittelt korrektes Ergebnis aus der Kandidatenmenge
 - Berechnet eigentliches Distanzmaß für die Kandidaten



Mehrstufige Anfragebearbeitung (2)

Qualitätskriterien für Filter

- **Effizienz**
 - Indextauglichkeit
 - schnelle Auswertung
- **Vollständigkeit**
 - Kandidaten müssen alle Ergebnisse enthalten
 - Beweis durch Lower-Bounding-Lemma:
 $d_{\text{filter}}(p,q) \leq d_{\text{exact}}(p,q)$
 - Analog gilt bei mehreren Filtern:
 $d_{\text{filter1}}(p,q) \leq d_{\text{filter2}}(p,q) \leq \dots$
- **Gute Selektivität**
 - Möglichste wenig Kandidaten für Verfeinerung
 - datenabhängig → empirischer Nachweis

