Boole'sches Modell

basiert auf Mengentheorie und Boole'scher Algebra
sehr einfaches Modell mit klarer Semantik

Dokumente als Mengen von Indextermen → Termgewichte
sind binär: im Dokument enthalten oder nicht enthalten

Test auf Enthaltensein als Vergleichsfunktion

Verknüpfung von Enthaltenseinsbedingungen mittels Boole'scher Junktoren

- *and* (Mengendurchschnitt)
- *or* (Mengenvereinigung)
- *not* (Mengendifferenz)

Boole'sches Modell: Beispiel

Indexvokabular {Korsika, Sardinien, Strand, Ferienwohnung, Gebirge}

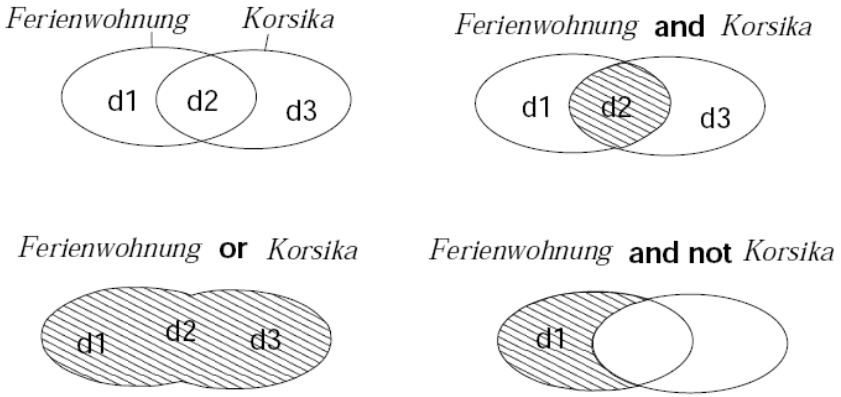
Dokument d1:{Sardinien, Strand, Ferienwohnung}
Dokument d2:{Korsika, Strand, Ferienwohnung}
Dokument d3:{Korsika, Gebirge}

Boole'sches Modell: Beispiel (2)

Anfrage

Korsika
liefert {d2, d3}
Ferienwohnung
liefert {d1, d2}
Ferienwohnung *and Korsika* liefert *{d2}*
Ferienwohnung or Korsika liefert *{d1, d2, d3}*
Ferienwohnung and not Korsika liefert *{d1}*

Boole'sches Modell: Beispiel (3)



Anfragenormalisierung in DNF bzw. KNF

Anfrage:

Ferienwohnung and ((Sardinien and Strand) or Korsika)

in disjunktiver Normalform (DNF):

(Ferienwohnung and Sardinien and Strand) or
(Ferienwohnung and Korsika)

in konjunktiver Normalform (KNF):

Ferienwohnung and (Sardinien or Korsika) and (Strand or Korsika)

Anfrageauswertung

jeder Term liefert Menge von Dokumenten, die diesen Term enthalten

komplexe Anfrage: Kombination der Dokumentenmengen durch entsprechende Mengenoperationen

kleine Zwischenergebnisse durch DNF (zuerst Durchschnitt, dann Vereinigung)

Nachteile des Boole'schen Modells

exaktes Modell aufgrund binärer Gewichte

- eher Daten-Retrieval
- keine Ähnlichkeitssuche durch zu scharfe Suche

Größe des Ergebnisses

- alle Dokumente sind bzgl. Anfrage gleichrangig → Präsentation der gesamten Ergebnismenge
- Ergebnismenge in Abhängigkeit von Anfrage oft zu groß oder zu klein bzw. leer

Nachteile des Boole'schen Modells (2)

Boole'sche Junktoren

- schwierige Anwendung Boole'scher Junktoren
- Verwechslung mit „und“, „oder“ und „nicht“

**⇒ Impliziter weiterer Gegensatz:
IR wird stärker von Laien benutzt als SQL**

Milderung Problem “exaktes Modell”

verschiedene Stufen der Relevanz durch Überführung Konjunktion in Disjunktion

Präsentation der Ergebnisse sortiert nach Relevanzstufen

Beispiel: Umwandlung von

„Korsika and Strand“ → „Korsika or Strand“

Ergebnis: zuerst die and-({d2}), dann restliche or-Dokumente ({d1, d3})

Milderung Problem “Größe des Ergebnisses”

faceted query

zweistufiges Suchverfahren

1. Formulierung und Verfeinerung der Anfrage anhand benannter Anfragen und Ergebnisanzahl
2. Ergebnis zur finalen Anfrage anzeigen

Milderung Problem “Größe des Ergebnisses (2)”

Beispiel:

1	Korsika	liefert	Q1: 1345
1	Q1 and Strand	liefert	Q2: 13
2	Anzeige Q2	liefert	die 13 Ergebnisdokumente

Beispiel Problem Boole'sche Junktoren

Anwender sucht Dokumente über Korsika und Dokumente über Sardinien

“falsche” d.h. nicht intendierte Anfrage:

Korsika and Sardinien

liefert Dokumente, in denen beide Terme gemeinsam auftreten

“richtige” bzw. intendierte Anfrage:

Korsika or Sardinien

Milderung Problem Boole'sche Junktoren

<is web>

Verwendung besserer Junktorenbegriffe, etwa Ersetzen von

„and“ durch „all“

„or“ durch „any“

Fuzzy-Modell

<is web>

Erweiterung des Boole'schen Modells um Unschärfe (fuzzy)

Verallgemeinerung Boole'scher Junktoren

Unschärfe durch graduelle Zugehörigkeit von Dokumenten zu Termen

Definition:
Eine Fuzzy-Menge $A = \{\langle u; \mu_A(u)\rangle\}$ über einem Universum U ist durch eine Zugehörigkeitsfunktion $\mu_A : U \rightarrow [0,1]$ charakterisiert, welche jedem Element u des Universums U einen Wert $\mu_A(u)$ aus dem Intervall $[0,1]$ zuordnet.

<is web>

Fuzzy-Mengen beim Information Retrieval

Universum ist Menge aller gespeicherten Dokumente

Term definiert Fuzzy-Menge

Zugehörigkeit (Fuzzy-Wert) des Dokuments *d* zu Term *t* *durch Wert*

- ***0 für keine Relevanz***
- ***1 für maximale Relevanz*** $\mu_t(d)$
- ***Wert zwischen 0 und 1 für graduelle Relevanz***

Beispiel

<is web>

Universum umfasst 3 Dokumente {d1, d2, d3}

Fuzzy-Mengen Korsika bzw. Strand drücken Zugehörigkeit zu Term „Korsika“ bzw. „Strand“ aus

$$\begin{aligned} \texttt{Korsika} &= \{\langle d1;\, 0{,}1\rangle, \langle d2;\, 0{,}6\rangle, \langle d3; 1\rangle\} \\ \texttt{Strand} &= \{\langle d1;\, 0{,}3\rangle, \langle d2;\, 0{,}2\rangle, \langle d3;\, 0{,}8\rangle\} \end{aligned}$$

μ	$d1$	$d2$	$d3$
$\mu_{\texttt{Korsika}}$	0,1	0,6	1
$\mu_{\texttt{Strand}}$	0,3	0,2	0,8

Fuzzy-Junktoren

jedes Dokument in jeder Fuzzy-Menge
→ übliche Mengenoperationen nicht anwendbar

Junktoren ermitteln neue Zugehörigkeitswerte

and durch Min-Funktion

or durch Max-Funktion

Fuzzy-Junktoren (2)

not durch Subtraktion von 1:

$$\begin{array}{rlcl} \text{Konjunktion } \mathbf{and}: & \mu_{A\cap B}(u) & = & min(\mu_A(u), \mu_B(u)) \\ \text{Disjunktion } \mathbf{or}: & \mu_{A\cup B}(u) & = & max(\mu_A(u), \mu_B(u)) \\ \text{Negation } \mathbf{not}: & \mu_{\bar{A}}(u) & = & 1 - \mu_A(u) \end{array}$$

Beispiel

Korsika and Strand

Korsika or Strand

not Korsika

Anfrage	μ	$d1$	$d2$	$d3$
	$\mu_{\texttt{Korsika}}$	0,1	0,6	1
	$\mu_{\texttt{Strand}}$	0,3	0,2	0,8
1	$\mu_{\texttt{Korsika} \cap \texttt{Strand}}$	0,1	0,2	0,8
2	$\mu_{\texttt{Korsika} \cup \texttt{Strand}}$	0,3	0,6	1
3	$\mu_{\overline{\texttt{Korsika}}}$	0,9	0,4	0

Fuzzy-Anfragen und Anfragebearbeitung

Überführung Anfrage in disjunktive Normalform

jeder Suchterm induziert eine Fuzzy-Menge

Anwendung entsprechender Fuzzy-Operationen auf Fuzzy-Mengen

Ergebnis: Dokumente absteigend sortiert nach Zugehörigkeitsgrad ausgeben

Begrenzung Fuzzy-Ähnlichkeitsanfrage

Ergebnis umfasst alle Dokumente des Universums

Begrenzung der Anzahl durch

- Schwellwerte für Zugehörigkeitswerte
- vorgegebene Anzahl von Ergebnisdokumenten

Begrenzung Fuzzy-Ähnlichkeitsanfrage (2)

Beispiel:

Korsika and Strand

Schwellwert 0,5 liefert	d3
Anzahl 2 liefert	d2, d3

Zugehörigkeitswert: Dokument zu Term

viele Möglichkeiten

Beispiel: Ansatz von Ogawa, Morita, Kobayashi mittels Term-zu-Term-Korrelationsmatrix

- ***Zeile entspricht Term i und Spalte entspricht Term j***
- ***$n_{i,j}$ ist Anzahl Dokumente, welche Terme t_i, t_j enthalten***

Zugehörigkeitswert

Dokument zu Term (2)

n_i ist Anzahl Dokumente, welche Term t_i enthalten

Zugehörigkeitsgrad: $$c_{i,j} = \frac{n_{i,j}}{n_i + n_j - n_{i,j}}$$

$$\mu_{t_i}(d_j) = 1 - \prod_{t_k \in d_j} (1 - c_{i,k})$$

Beispiel: Term-zu-Term-Korrelationsmatrix

Dokument d1: {Sardinien, Strand, Ferienwohnung}
Dokument d2: {Korsika, Strand, Ferienwohnung}
Dokument d3: {Korsika, Gebirge}

	$t_{Sardinien}$	t_{Strand}	$t_{Ferienw.}$	$t_{Korsika}$	$t_{Gebirge}$
$t_{Sardinien}$	1	0,5	0,5	0	0
t_{Strand}	0,5	1	1	0,333	0
$t_{Ferienw.}$	0,5	1	1	0,333	0
$t_{Korsika}$	0	0,333	0,333	1	0,5
$t_{Gebirge}$	0	0	0	0,5	1

Beispielrechnung

Dokument d1: {**Sar**dinien, **Str**and, **Fer**ienwohnung}
Dokument d2: {**Kor**sika, **Str**and, **Fer**ienwohnung}
Dokument d3: {**Kor**sika, **Geb**irge}

$\mu_{Sar}(d_1)=1-\Pi_{t_k 2\{Sar,Str,Fer\}} (1-c_{i,k})=1-(1-1)(1-0,5)(1-0,5)=1$

$\mu_{Kor}(d_1)=1-\Pi_{t_k 2\{Sar,Str,Fer\}} (1-c_{i,k})=1-(1-0)(1-0,33)(1-0,33)=5/9$

Zugehörigkeitswert

Dokument zu Anfrageterm

Fuzzy-Modell für komplexe Ähnlichkeitsanfragen:
Ähnlichkeitswert bzgl. atomarer Anfrage als Zugehörigkeitswert
anwendbar für jeden Medien-Typ
Beispiel: Suche nach roter Morgensonne

Farbe = *rot* and *Gestalt* = *Kreis*

Vektorraummodelle

UNIVERSITÄT KOBLENZ · LANDAU

Vektorraummodell

<is web>

weit verbreitetes Retrieval-Modell

Dokumente als Vektoren eines Vektorraums
→ Überführung Retrieval-Problem in Gebiet der linearen Algebra

einsetzbar, wenn jedes Medien-Objekt darstellbar durch feste Anzahl von numerischen Merkmalswerten

Beispiel Bild-Retrieval: Vektorwerte etwa anhand der Farbverteilung

Vektorraummodell (2)

<is web>

Beispiel Text-Retrieval:

- jeder Indexterm eine eigene Dimension
- Termgewicht (meist Häufigkeiten) als Vektorwert einer Dimension

Ähnlichkeit im Vektorraum

<is web>

Anfrage selbst als Vektor

Ähnlichkeit zwischen Anfrage *q und Dokument d über deren Vektoren*

- *Kosinusmaß (Kosinus des eingeschlossenen Winkels als Ähnlichkeitsmaß)*

$$sim_{cos}(d,q) = \frac{\langle d,q \rangle}{|d| * |q|}$$

- *Distanzmaß (Abstand zwischen Vektoren als Unähnlichkeitsmaß) z.B. Euklid'sche Distanz:*

$$dissim_{L_2}(d,q) = \sqrt{(d-q)^T(d-q)} = \sqrt{\sum_i (d[i]-q[i])^2}$$

Beispiel

<is web>

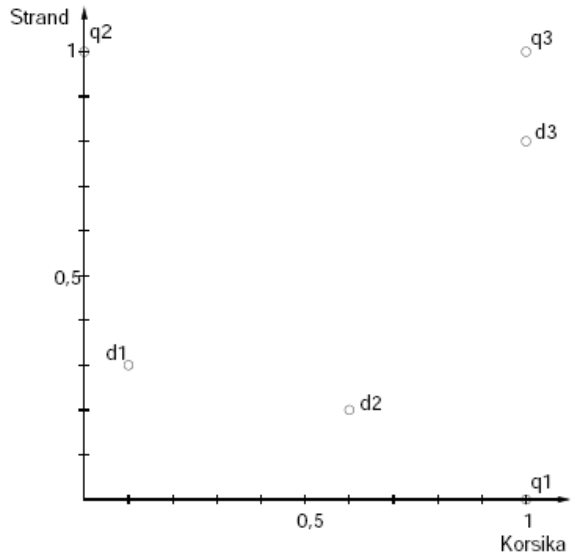
Dokumente:

Dimension	$d1$	$d2$	$d3$
Korsika	0,1	0,6	1
Strand	0,3	0,2	0,8

Anfragen:

Dimension	$q1$	$q2$	$q3$
Korsika	1	0	1
Strand	0	1	1

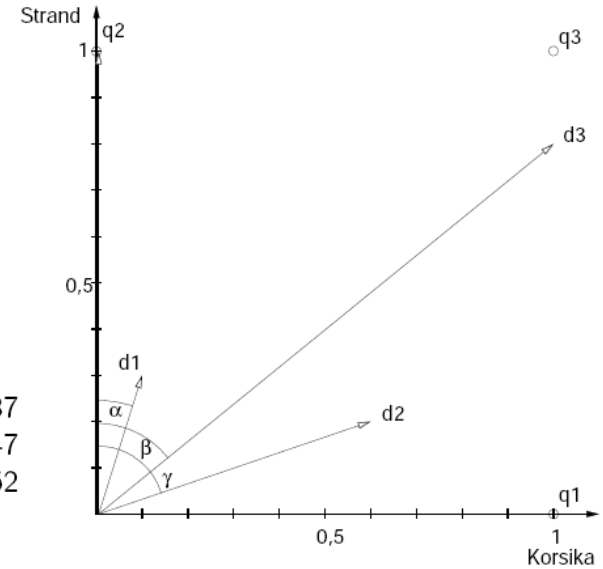
Darstellung im Vektorraum



Ähnlichkeitswerte nach Kosinusmaß

sim_{cos}	$d1$	$d2$	$d3$
$q1$	0,3162	0,9487	0,7809
$q2$	0,9487	0,3162	0,6247
$q3$	0,8944	0,8944	0,9939

Eingeschlossene Winkel

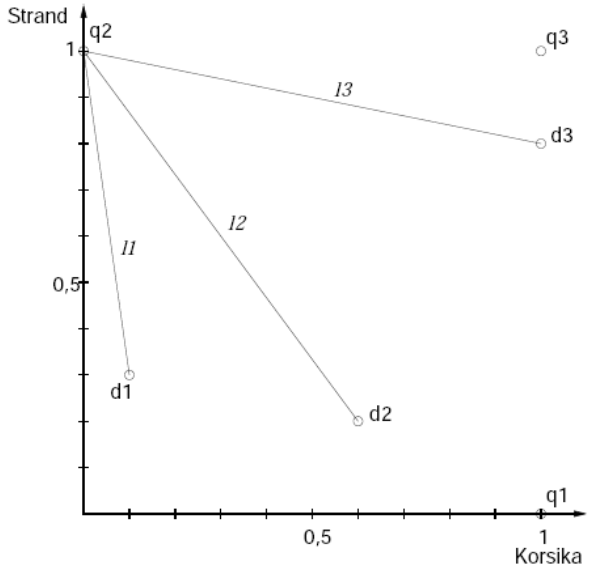


$$\cos\alpha = 0{,}9487$$
$$\cos\beta = 0{,}6247$$
$$\cos\gamma = 0{,}3162$$

Unähnlichkeitswerte Euklid'scher Distanz

$dissim_{L_2}$	$d1$	$d2$	$d3$
$q1$	0,9487	0,4472	0,8
$q2$	0,7071	1	1,0198
$q3$	1,1402	0,8944	0,2

Abstände



$$
\begin{aligned}
l1 &= 0{,}7071 \\
l2 &= 1 \\
l3 &= 1{,}0198
\end{aligned}
$$

Kosinusmaß und Euklid'sche Distanz

erzeugen unterschiedliche Ergebnisse (Sortierungen)

- Kosinusmaß erzeugt <d1, d3, d2>
- Euklid'sche Distanz erzeugt <d1, d2, d3>

Wahl der geeigneten Ähnlichkeitsfunktion abhängig von

- subjektivem Ähnlichkeitsempfinden
- Anwendungsszenario

Zusammenfassung Vektorraummodell

Vektorraummodell sehr weit verbreitet

setzt feste Anzahl von numerischen Merkmalswerten pro Dokument voraus

Zusammenfassung Vektorraummodell (2)

Probleme:

- Merkmale als orthogonale Dimensionen aufgefasst (unrealistisch)
  → Orthogonalisierung
- Problem bei hoher Anzahl von Merkmalswerten bzgl. Effektivität und Effizient
  → Dimensionsreduktion
- Anfrage ist Vektor, also keine Junktoren
  → Boole'sche Junktoren durch Kombination mit dem Fuzzy-Modell