

Retrievalmodelle

Algebraische Modelle

Algebraisches Modell

In der $m \times n$ Term-Dokument-Matrix bestehen **Korrelationen** durch **Synonyme**, **Kookkurrenzen**, sich wiederholende **Phrasen** und n-Gramme.

Man könnte also die Dokumente des m -dimensionalen Vektorraums auch in einem Teilraum niedriger Dimension darstellen, aber trotzdem die ursprüngliche Information ausreichend genau approximieren.

Idee:

Transformation der **hochdimensionalen Dokumentvektoren** in einen **niedrigdimensionalen Raum** bei möglichst genauer Erhaltung der Information.

Die hierbei entstehenden Linearkombinationen der Terme lassen sich als **verborgene Konzepte** interpretieren.

Algebraisches Modell

Term-Dokument-Matrix

	d_1	d_2	\dots	d_n
t_1	w_{1_1}	w_{1_2}	\dots	w_{1_n}
t_2	w_{2_1}	w_{2_2}	\dots	w_{2_n}
\vdots				
t_m	w_{m_1}	w_{m_2}	\dots	w_{m_n}

Kookkurrenz

	d_1	d_2	d_3	d_4
t_1	2	7	4	0
t_2	w_{2_1}	w_{2_2}	w_{2_3}	w_{2_4}
t_3	2	6	3	0
t_4	w_{4_1}	w_{4_2}	w_{4_4}	w_{4_4}

$t_1 \sim t_3$

wiederholte Phrase

	d_1	d_2	d_3	d_4
t_1	1	2	4	0
t_2	w_{2_1}	w_{2_2}	w_{2_3}	w_{2_4}
t_3	2	4	7	0
t_4	1	2	3	0

$t_1 \sim 2 \cdot t_3 \wedge 1 \cdot t_4$

Synonym

	d_1	d_2	d_3	d_4
t_1	2	4	3	0
t_2	w_{2_1}	w_{2_2}	w_{2_3}	w_{2_4}
t_3	2	0	1	0
t_4	0	4	2	0

$(t_1) \sim t_3 + t_4$

Wiederholung: Lineare Algebra

- Sei A eine $n \times n$ -Matrix, λ ein Eigenwert von A mit Eigenvektor \mathbf{x} . Dann gilt:

$$A\mathbf{x} = \lambda\mathbf{x}$$

- Sei A eine symmetrische $n \times n$ -Matrix mit Rang r . Dann können wir A in folgender Form darstellen:

$$A = U\Delta U^T$$

Δ ist eine mit den Eigenwerten von A besetzte $r \times r$ -Diagonalmatrix

U ist eine $n \times r$ -spaltenorthonormale Matrix: $U^t U = I$

- Sei A eine $m \times n$ -Matrix vom Rang r . Dann können wir A in folgender Form darstellen:

$$A = USV^T$$

S ist eine $r \times r$ -Diagonalmatrix

U ist eine spaltenorthonormale $m \times r$ -Matrix

V ist eine spaltenorthonormale $r \times n$ -Matrix

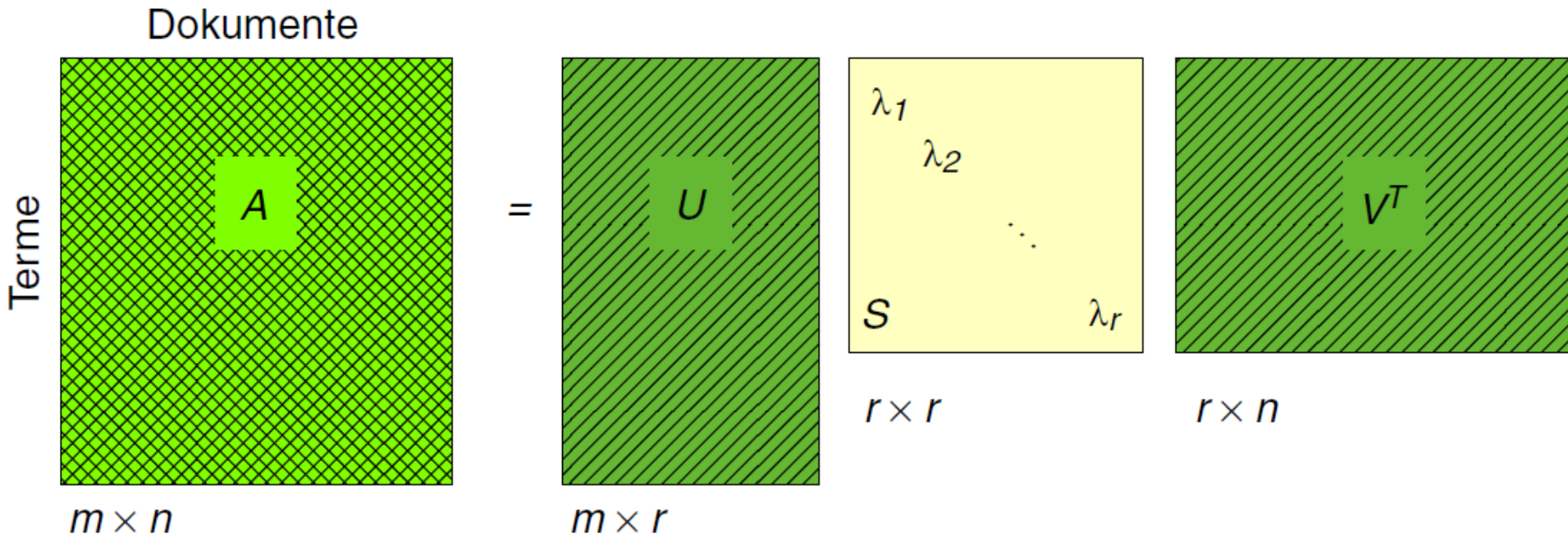
Wiederholung: Lineare Algebra

$A=USV^T$ lässt sich als Summe von Vektorprodukten schreiben:

$$A = s_1(\mathbf{u}_1 \mathbf{v}_1^T) + \dots + s_r(\mathbf{u}_r \mathbf{v}_r^T)$$

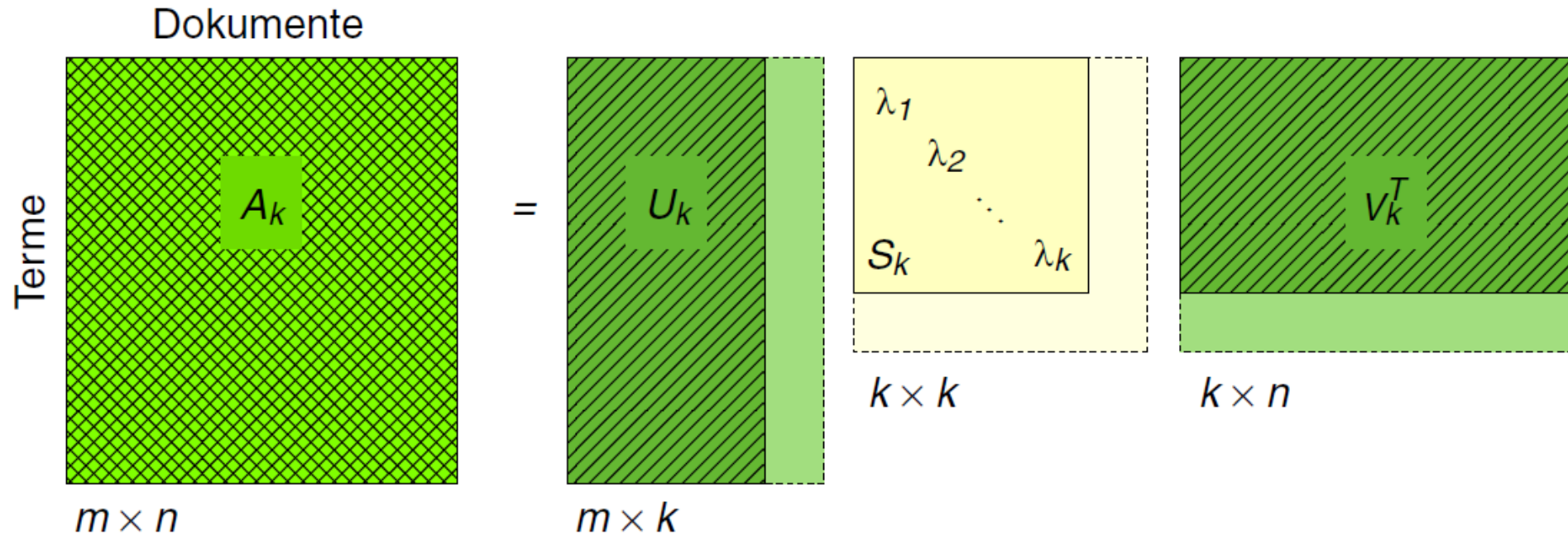
Wir können A **approximieren**, indem wir die Summanden mit den kleinsten Singulärwerten weglassen

Singulärwertzerlegung: $A=USV^T$



U ist spalten-orthonormal
S ist diagonal, $r \leq \min\{m, n\}$
V ist spalten-diagonal

Approximation: $A_k = U_k S_k V_k^T$



U_k ist spalten-orthonormal
 S_k ist diagonal, $r \leq \min\{m, n\}$
 V_k ist spalten-diagonal

Dokumentmodell

- **Dokumentrepräsentationen D.**

- Die Dokumentrepräsentationen des Vektorraummodells werden zu einer $m \times n$ Term-Dokument-Matrix A zusammengefasst.
- A wird durch Dimensionsreduktion zur **Konzept-Dokument-Matrix** $D = V_k^T$.
 D repräsentiert die Dokumente im **Konzeptraum** (latent semantic space).

- **Formalisierte Anfragemenge Q.**

Ausgangspunkt einer formalen Anfrage ist ihre Vektorraumrepräsentation \mathbf{q} . Durch folgende Operation wird \mathbf{q} in den Konzeptraum transformiert:

$$\mathbf{q}' := S_k^{-1} U_k^T \mathbf{q}$$

- **Retrieval-Funktion $\rho_{\mathcal{R}}$.**

$\rho_{\mathcal{R}}$ wird unmittelbar auf die Darstellungen der Dokumente und Anfragen im Konzeptraum angewandt. Dabei kommen Retrieval-Funktionen wie beim Vektorraummodell zum Einsatz.

Beispiel (aus dem LSI-Paper)

Human-
Computer
Interaction

- d1 Human machine interface for Lab ABC computer applications
- d2 A survey of user opinion of computer system response time
- d3 The EPS user interface management system
- d4 System and human system engineering testing of EPS
- d5 Relation of user-perceived response time to error measurement
- d6 The generation of random, binary, unordered trees
- d7 The intersection graph of paths in trees
- d8 Graph minors IV: Widths of trees and well-quasi-ordering
- d9 Graph minors: A survey

Graphen

Anfrage $q = \{\text{human, computer, interaction}\}$

Ergebnisse für diese Anfrage im Booleschen Modell mit AND bzw. OR der Terme?
Ergebnisse für diese Anfrage im Vektorraummodell?

Beispiel: Term-Dokument-Matrix

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9
human	1			1					
interface	1		1						
computer	1	1							
user		1	1		1				
system		1	1	2					
response		1			1				
time		1			1				
EPS			1	1					
survey		1							1
trees						1	1	1	
graph							1	1	1
minors								1	1

Beispiel: Term-Dokument-Matrix

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9
human	1			1					
interface	1		1						
computer	1	1							
user		1	1						
system		1	1						
response		1							
time		1							
EPS			1						
survey		1							1
trees									
graph									1
minors									1

Eingabe in Mathesoftware
(hier: Scilab)

```
Scilab 5.5.2 Console
-->A=[
-->1 0 0 1 0 0 0 0 0
-->1 0 1 0 0 0 0 0 0
-->1 1 0 0 0 0 0 0 0
-->0 1 1 0 1 0 0 0 0
-->0 1 1 2 0 0 0 0 0
-->0 1 0 0 1 0 0 0 0
-->0 1 0 0 1 0 0 0 0
-->0 0 1 1 0 0 0 0 0
-->0 1 0 0 0 0 0 0 1
-->0 0 0 0 0 1 1 1 0
-->0 0 0 0 0 0 1 1 1
-->0 0 0 0 0 0 0 1 1]
A =
    1.   0.   0.   1.   0.   0.   0.   0.   0.
    1.   0.   1.   0.   0.   0.   0.   0.   0.
    1.   1.   0.   0.   0.   0.   0.   0.   0.
    0.   1.   1.   0.   1.   0.   0.   0.   0.
    0.   1.   1.   2.   0.   0.   0.   0.   0.
    0.   1.   0.   0.   1.   0.   0.   0.   0.
    0.   1.   0.   0.   1.   0.   0.   0.   0.
    0.   0.   1.   1.   0.   0.   0.   0.   0.
```

Scilab 5.5.2 Console

```
-->A=[
-->1 0 0 1 0 0 0 0 0
-->1 0 1 0 0 0 0 0 0
-->1 1 0 0 0 0 0 0 0
-->0 1 1 0 1 0 0 0 0
-->0 1 1 2 0 0 0 0 0
-->0 1 0 0 1 0 0 0 0
-->0 1 0 0 1 0 0 0 0
-->0 0 1 1 0 0 0 0 0
-->0 1 0 0 0 0 0 0 1
-->0 0 0 0 0 1 1 1 0
-->0 0 0 0 0 0 1 1 1
-->0 0 0 0 0 0 0 1 1]
```

$$\mathbf{A} =$$

1.	0.	0.	1.	0.	0.	0.	0.	0.
1.	0.	1.	0.	0.	0.	0.	0.	0.
1.	1.	0.	0.	0.	0.	0.	0.	0.
0.	1.	1.	0.	1.	0.	0.	0.	0.
0.	1.	1.	2.	0.	0.	0.	0.	0.
0.	1.	0.	0.	1.	0.	0.	0.	0.
0.	1.	0.	0.	1.	0.	0.	0.	0.
0.	0.	1.	1.	0.	0.	0.	0.	0.
0.	1.	0.	0.	0.	0.	0.	0.	1.
0.	0.	0.	0.	0.	1.	1.	1.	0.
0.	0.	0.	0.	0.	0.	1.	1.	1.
0.	0.	0.	0.	0.	0.	0.	1.	1.

→

Singulärwertzerlegung dieser Matrix

latente Topics/Themen

U

Terme

-0.2213508	-0.1131796	0.2889582	-0.4147507	-0.1062751	-0.3409833	-0.5226578	0.0604501	0.4066775
-0.1976454	-0.0720878	0.1350396	-0.5522396	0.2817689	0.4958780	0.0704234	0.0099400	0.1089303
-0.2404702	0.0431520	-0.1644291	-0.5949618	-0.1067553	-0.2549551	0.3022402	-0.0623280	-0.4924444
-0.4035989	0.0570703	-0.3378035	0.0991137	0.3317337	0.3848319	-0.0028722	0.0003905	-0.0123293
-0.6444812	-0.1673012	0.3611482	0.3334616	-0.1589550	-0.2065226	0.1658286	-0.0342720	-0.2706963
-0.2650375	0.1071596	-0.4259985	0.0738122	0.0803194	-0.1696764	-0.2829157	0.0161465	0.0538747
-0.2650375	0.1071596	-0.4259985	0.0738122	0.0803194	-0.1696764	-0.2829157	0.0161465	0.0538747
-0.3008282	-0.1412705	0.3303084	0.1880919	0.1147846	0.2721553	-0.0329941	0.0189980	0.1653392
-0.2059179	0.2736474	-0.1775970	-0.0323519	-0.53715	0.0809440	0.4668975	0.0362988	0.5794261
-0.0127462	0.4901618	0.2311202	0.0248020	0.5941695	-0.3921251	0.2883175	-0.2545679	0.2254241
-0.0361358	0.6227852	0.2230864	0.0007001	-0.0682529	0.1149090	-0.1595755	0.6811254	-0.2319612
-0.0317563	0.4505089	0.1411152	-0.0087295	-0.3004951	0.2773434	-0.3394953	-0.6784179	-0.1825350

S

3.3409

2.5417

2.3539

1.6445

1.5048

1.3064

0.8459

0.5601

0.3637

V^T

latente Topics

-0.1973928	-0.6059903	-0.4629175	-0.5421144	-0.2794691	-0.0038152	-0.0146315	-0.0241368	-0.0819574
-0.0559135	0.1655929	-0.1273121	-0.2317552	0.1067747	0.1928479	0.4378749	0.6151219	0.5299371
0.1102697	-0.4973265	0.2076060	0.5699214	-0.5054499	0.0981842	0.1929556	0.2529040	0.0792731
-0.9497850	-0.0286489	0.0416092	0.2677140	0.1500354	0.0150815	0.0155072	0.0101990	-0.0245549
0.0456786	-0.2063273	0.3783362	-0.2056047	0.3271944	0.3948412	0.3494853	0.1497985	-0.6019930
-0.0765936	-0.2564752	0.7243996	-0.3688609	0.0348130	-0.3001611	-0.2122014	0.0000974	0.3622190
-0.1773183	0.4329842	0.2368897	-0.2647995	-0.6723035	0.3408398	0.1521947	-0.2491459	-0.0380342
0.0143933	-0.0493053	-0.0088255	0.0194669	0.0583496	-0.4544765	0.7615270	-0.4496428	0.0696375
0.0636923	-0.2427829	-0.0240769	0.0842069	0.2623759	0.6198472	-0.0179752	-0.5198905	0.4535068

Dokumente

Dimensionsreduktion: 2 Dimensionen

U₂

-0.2213508	-0.1131796
-0.1976454	-0.0720878
-0.2404702	0.0431520
-0.4035989	0.0570703
-0.6444812	-0.1673012
-0.2650375	0.1071596
-0.2650375	0.1071596
-0.3008282	-0.1412705
-0.2059179	0.2736474
-0.0127462	0.4901618
-0.0361358	0.6227852
-0.0317563	0.4505089

q	1
	0
	1
	0
	0
	0
	0
	0
	0
	0
	0
	0

$$q' := S_k^{-1}U_k^Tq$$

-0.1382332
-0.0275515

$$qprime=inv(S)*U' *q$$

S₂

3.3409
2.5417

V₂^T

-0.1973928	-0.6059903	-0.4629175	-0.5421144	-0.2794691	-0.0038152	-0.0146315	-0.0241368	-0.0819574
-0.0559135	0.1655929	-0.1273121	-0.2317552	0.1067747	0.1928479	0.4378749	0.6151219	0.5299371

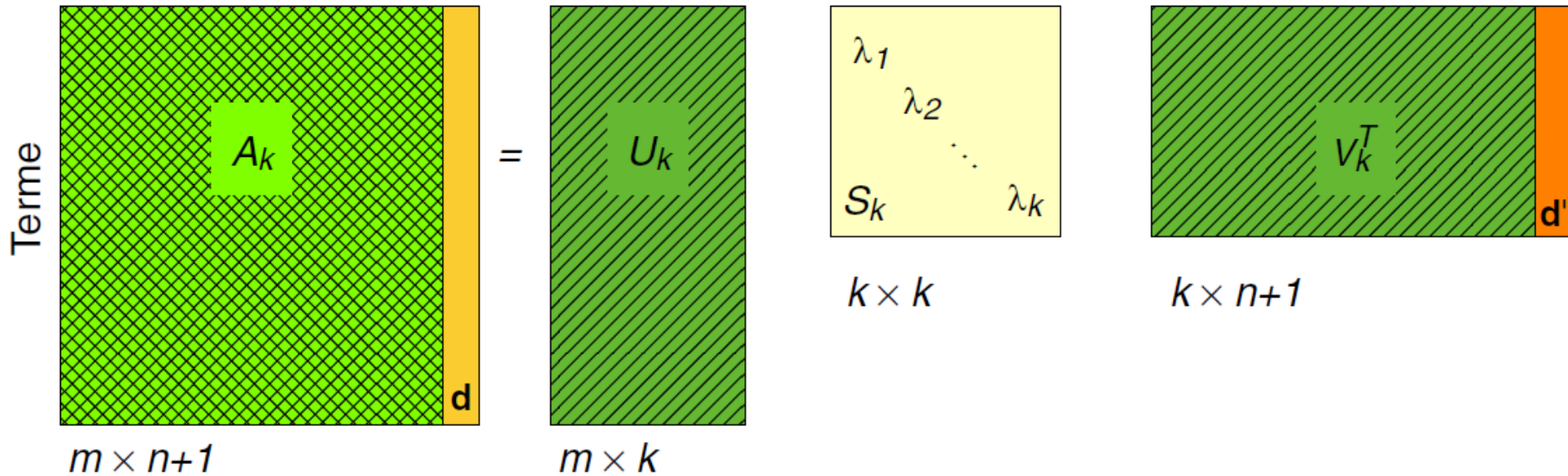
$$A_2=U_2S_2V_2^T\approx A$$

$$U*S*V'$$

0.1620580	0.4004983	0.3789545	0.4675663	0.1759537	-0.0526549	-0.1151428	-0.1591020	-0.0918383
0.1405853	0.3698008	0.3289960	0.4004272	0.1649725	-0.0328155	-0.0705686	-0.0967683	-0.0429807
0.1524495	0.5050044	0.3579366	0.4101068	0.2362317	0.0242165	0.0597805	0.0868573	0.1239663
0.2580493	0.8411234	0.6057199	0.6973572	0.3923179	0.0331180	0.0832449	0.1217724	0.1873797
0.4487898	1.2343648	1.0508615	1.2657956	0.5563314	-0.0737900	-0.1546938	-0.2095982	-0.0488795
0.1595543	0.5816819	0.3752190	0.4168977	0.2765405	0.0559037	0.1322185	0.1889115	0.2169076
0.1595543	0.5816819	0.3752190	0.4168977	0.2765405	0.0559037	0.1322185	0.1889115	0.2169076
0.2184628	0.5495806	0.5109605	0.6280580	0.2425361	-0.0654110	-0.1425215	-0.1966119	-0.1079133
0.0969064	0.5320644	0.2299137	0.2117536	0.2665251	0.1367562	0.3146208	0.4444406	0.4249695
-0.0612539	0.2321082	-0.1388984	-0.2656459	0.1449255	0.2404210	0.5461472	0.7673742	0.6637093
-0.0646770	0.3352812	-0.1456405	-0.3014061	0.2027564	0.3057261	0.6948934	0.9766112	0.8487497
-0.0430820	0.2539057	-0.0966670	-0.2078582	0.1519134	0.2212270	0.5029449	0.7069116	0.6155044

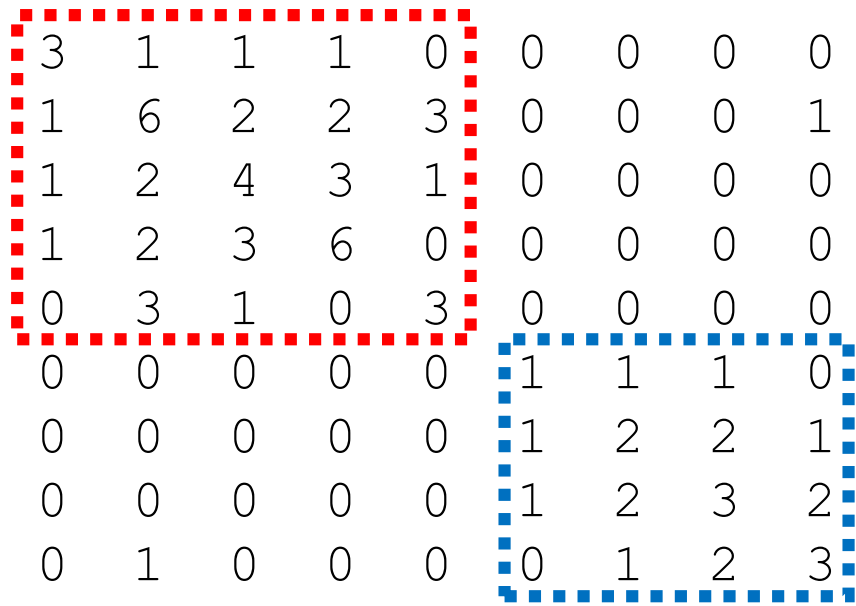
Einfügen neuer Dokumente

Dokumente



- originalen Dokumentvektor \mathbf{d} an A_k hängen
- reduzierten Dokumentvektor $\mathbf{d}' := S_k^{-1} U_k^T \mathbf{d}$ berechnen (vgl. formalisierte Anfrage)
- reduzierten Dokumentvektor \mathbf{d}' an V_k^T

Dokumentähnlichkeitsmatrix $A^T A$



3	1	1	1	0	0	0	0	0	0
1	6	2	2	3	0	0	0	0	1
1	2	4	3	1	0	0	0	0	0
1	2	3	6	0	0	0	0	0	0
0	3	1	0	3	0	0	0	0	0
0	0	0	0	0	1	1	1	0	0
0	0	0	0	0	1	2	2	1	0
0	0	0	0	0	1	2	3	2	0
0	1	0	0	0	0	1	2	3	0
0	0	0	0	0	0	0	0	0	0

Interpretation: $A^T A$ zeigt die Dokument-Cluster; ggf.
Umsortierung der Dokumente nötig

Termähnlichkeitsmatrix AA^T

2	1	1	0	2	0	0	1	0	0	0	0
1	2	1	1	1	0	0	1	0	0	0	0
1	1	2	1	1	1	1	0	1	0	0	0
0	1	1	3	2	2	2	1	1	0	0	0
2	1	1	2	6	1	1	3	1	0	0	0
0	0	1	2	1	2	2	0	1	0	0	0
0	0	1	2	1	2	2	0	1	0	0	0
1	1	0	1	3	0	0	2	0	0	0	0
0	0	1	1	1	1	1	0	2	0	1	1
0	0	0	0	0	0	0	0	0	3	2	1
0	0	0	0	0	0	0	0	1	2	3	2
0	0	0	0	0	0	0	0	1	1	2	2

Interpretation: AA^T zeigt die Term-Cluster bzw. Konzepte, evtl. Synonyme; ggf. Umsortierung der Terme nötig

Diskussion: Algebraisches Modell

Vorteile:

- automatische Entdeckung verborgener **Konzepte**
- syntaktische Erkennung von **Synonymen**
- **semantische Erweiterung** von Anfragen aufgrund syntaktischer Analyse – und nicht durch Relevanz-Feedback oder die Bemühung von Thesauri

Nachteile:

- die Wirkungsweise von LSI ist nicht vollständig verstanden; eine theoretisch fundierte Brücke zur Linguistik ist nur ansatzweise vorhanden
- LSI entfaltet die volle Wirkung nur in einer **geschlossenen Retrieval-Situation**: die Kollektion ist bekannt, gegeben und ändert sich nur wenig
- die Singulärwertzerlegung ist **rechenaufwendig**, $O(n^3)$

Retrievalmodelle

Kombination mehrerer Modelle

In der Praxis: Komplexe Kombinationen von Features

In der Praxis verwendet man in den großen Suchmaschinen nicht die reinen Scores, die wir bisher kennengelernt haben, sondern kombiniert sie mit Relevanzsignalen verschiedener Art zu einem **Gesamtscore**. Der Einfluss der einzelnen Komponenten (“die Google-Formel”) wird dabei entweder von Hand getunt oder von automatischen Verfahren bestimmt.

Übliche Klassen von **Relevanzsignalen** (oder **Features**) sind:

- **Dynamische Signale**, die von der Anfrage und vom Dokument abhängen, z.B. die verschiedenen Scores (tfidf, BM25, LM, etc.) in verschiedenen Dokumentzonen und für das ganze Dokument
- **Statische Signale**, die nur vom Dokument abhängen, z.B. Qualitätsscores wie PageRank und HITS, Dokumentlänge, Spamwahrscheinlichkeit, Änderungsfrequenz, Sprache, etc.
- **Anfrageeigenschaften**, z.B. relative Häufigkeit, mittlere IDF-Werte der Terme, etc.

In der Praxis: Komplexe Kombinationen von Features

Orthogonal und teilweise ergänzend kann man die Features auch nach ihrer Quelle gruppieren:

- **Inhaltssignale**, die den Inhalt eines einzelnen Dokuments betrachten (z.B. Retrievalscores, aber auch Dokumentlänge, Eigenschaften der HTML-Quellen, Änderungsfrequenz etc.)
- **Struktursignale**, die die Verlinkung von Seiten im Web ausnutzen (z.B. Anchortexte, aber auch Pagerank)
- **Verhaltens- und Benutzersignale**, die das Clickverhalten von Benutzern berücksichtigen (z.B. Clickrate auf dieses Dokument in einer Ergebnisliste, sowohl global als auch pro Anfrage) sowie Eigenschaften des Benutzers selbst (z.B. aktuelle Position, Sprache, Anfragehistorie, etc.)

Alle Signale werden in den gleichen numerischen Raum abgebildet, z.B. auf die Menge \mathbb{R} der reellen Zahlen.

Kombination der Einzelsignale

Die einzelnen Signale aus einer Menge F werden zu einem **gewichteten Gesamtscore** kombiniert, z.B. linear:

$$s(q, d) := \sum_{f \in F} \omega_f \cdot s_f(q, d)$$

Die Gewichte ω_f werden auf Basis **großer Trainingsdaten** automatisch oder manuell bestimmt. Dazu sammelt man eine große Menge Q von Anfragen sowie, für jede Anfrage $q \in Q$, ihre wahrscheinlich relevanten Dokumente $R(q)$, z.B. aus Clicks in Suchmaschinen, und **minimiert dann die Zahl der “Fehlentscheidungen”** (mit D als Menge aller Dokumente)

$$\sum_{q \in Q} |\{(d, r) \in (D \setminus R(q)) \times R(q) : s(q, d) \geq s(q, r)\}|$$

Entsprechende automatische Verfahren sind als **“Learning to Rank”** bekannt. Dabei werden auch komplexere Kombinations- und Optimierungsmethoden als die hier gezeigte angewendet.