Chapter IR:V

V. Retrieval Models

- Overview of Retrieval Models
- Empirical Models
- Boolean Retrieval
- Vector Space Model
- Probabilistic Models
- Binary Independence Model
- Okapi BM25
- Hidden Variable Models
- Latent Semantic Indexing
- Explicit Semantic Analysis
- Generative Models
- Language Models
- Combining Evidence
- Web Search
- Learning to Rank

Hidden Variable Models

Obviously, the terms found in a document $d \in D$ are somehow related to the semantics of d. Hidden variable models do not require this relation to be explicit and directly quantifiable.

The terms of a document $d \in D$ are *a manifestation* of its semantics, which actually relate to underlying concepts, ideas, or metaphors. This relation results from a common context and cultural background of author and reader.

Hidden Variable Models

[Empirical Models] [Probabilistic Models] [Generative Models]

Obviously, the terms found in a document $d \in D$ are somehow related to the semantics of d. Hidden variable models do not require this relation to be explicit and directly quantifiable.

The terms of a document $d \in D$ are *a manifestation* of its semantics, which actually relate to underlying concepts, ideas, or metaphors. This relation results from a common context and cultural background of author and reader.

Discriminating factors of hidden variable models:

1. What a hidden variable represents (e.g., concept, aspect, topic).
2. How hidden variables relate to document d.
3. Extent of assumptions about independence.
4. Computation method for hidden variables.
5. Computation method of the relevance function $\rho(\mathbf{q}, \mathbf{d})$.

Hidden Variable Models

Term-Document Matrix

Consideration:

In an $m \times n$ term-document matrix, correlations can be observed because of synonymy, co-occurrence, repeated phrases, and n-grams.

Arguably, the m-dimensional representations of the documents can be mapped to lower-dimensional vector representations through a coordinate transformation, approximating the original vector space.

Idea:

Transform the high-dimensional vector representations to a low-dimensional space, approximating the original information as accurately as possible.

The resulting linear combinations of terms may be interpreted as hidden concepts.

Hidden Variable Models

Term-Document Matrix

Term-document matrix:

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

Hidden Variable Models

Term-Document Matrix

Term-document matrix:

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

Co-occurrence

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

$t_1 \sim t_3$

Hidden Variable Models

Term-Document Matrix

Term-document 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}

Co-occurrence

	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}

$t_1 \sim t_3$

Repeated 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

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

Hidden Variable Models

Term-Document Matrix

Term-document 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}

Co-occurrence

	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}

$t_1 \sim t_3$

Repeated 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

$(t_1) \sim t_3 + t_4$

Remarks:

- Co-occurrence: t_1 and t_3 occur (almost) always simultaneously.
- Repeated phrase: A phrase exists, where t_1 (almost) always occurs with $2 \cdot t_3$ and one t_4.
- Synonym: For a given concept (here represented as (t_1)) holds that it can be described by either t_3 or t_4.

Latent Semantic Indexing

Singular Value Decomposition

From linear algebra:

(1) Let A denote an $n \times n$ matrix, λ an eigenvalue of A with eigenvector $\mathbf{x}$. Then:

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

Latent Semantic Indexing

Singular Value Decomposition

From linear algebra:

(1) Let A denote an $n \times n$ matrix, λ an eigenvalue of A with eigenvector $\mathbf{x}$. Then:

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

(2) Let A denote a symmetric $n \times n$ matrix of rank r. Then A can be presented as follows:

$$A = U\Lambda U^T$$

Λ is an $r \times r$ diagonal matrix occupied with the eigenvalues of A
U is an $n \times r$ column orthonormal matrix: $U^T U = I$

Latent Semantic Indexing

Singular Value Decomposition

From linear algebra:

(1) Let A denote an $n \times n$ matrix, λ an eigenvalue of A with eigenvector $\mathbf{x}$. Then:

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

(2) Let A denote a symmetric $n \times n$ matrix of rank r. Then A can be presented as follows:

$$A = U \Lambda U^T$$

Λ is an $r \times r$ diagonal matrix occupied with the eigenvalues of A
U is an $n \times r$ column orthonormal matrix: $U^T U = I$

(3) Let A denote an $m \times n$ matrix of rank r. Then A can be presented as follows:

$$A = U S V^T$$

U is an $m \times r$ column orthonormal matrix
S is an $r \times r$ diagonal matrix occupied by the singular values of A
V is an $n \times r$ column orthonormal matrix

Latent Semantic Indexing

Singular Value Decomposition

From linear algebra (continued):

(4) With $A = USV^T$ holds:

$$A^T A = (U S V^T)^T (U S V^T) = V S U^T U S V^T = V S^2 V^T$$

The columns of V are eigenvectors of $A^T A$.
The singular values of A correspond to the square root of the eigenvalues of $A^T A$.

Latent Semantic Indexing

Singular Value Decomposition

From linear algebra (continued):

(4) With $A = USV^T$ holds:

$$A^TA = (USV^T)^T(USV^T) = VSU^TUSV^T = VS^2V^T$$

The columns of V are eigenvectors of A^TA.
The singular values of A correspond to the square root of the eigenvalues of A^TA.

(5) and moreover:

$$AA^T = (USV^T)(USV^T)^T = USV^TVSU^T = US^2U^T$$

The columns of U are eigenvectors of AA^T.
The singular values of A correspond to the square root of the eigenvalues of AA^T

Latent Semantic Indexing

Singular Value Decomposition

From linear algebra (continued):

(4) With $A = USV^T$ holds:

$$A^TA = (USV^T)^T(USV^T) = VSU^TUSV^T = VS^2V^T$$

The columns of V are eigenvectors of A^TA.
The singular values of A correspond to the square root of the eigenvalues of A^TA.

(5) and moreover:

$$AA^T = (USV^T)(USV^T)^T = USV^TVSU^T = US^2U^T$$

The columns of U are eigenvectors of AA^T.
The singular values of A correspond to the square root of the eigenvalues of AA^T

(6) $A = USV^T$ can be written as sum of (dyadic) vector products:

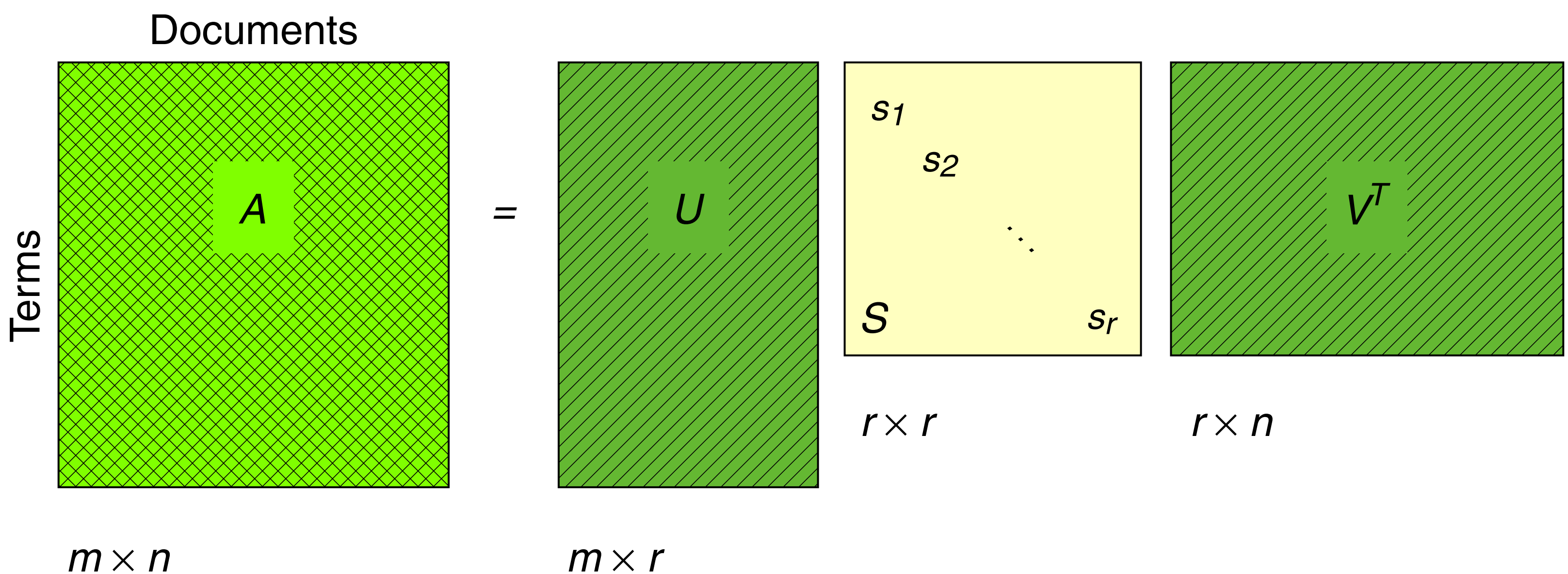
$$A = s_1(\mathbf{u}_1\mathbf{v}_1^T) + s_2(\mathbf{u}_2\mathbf{v}_2^T) + \ldots + s_r(\mathbf{u}_r\mathbf{v}_r^T)$$

Approximation of A by omission of summands with smallest singular values.

Latent Semantic Indexing

Singular Value Decomposition

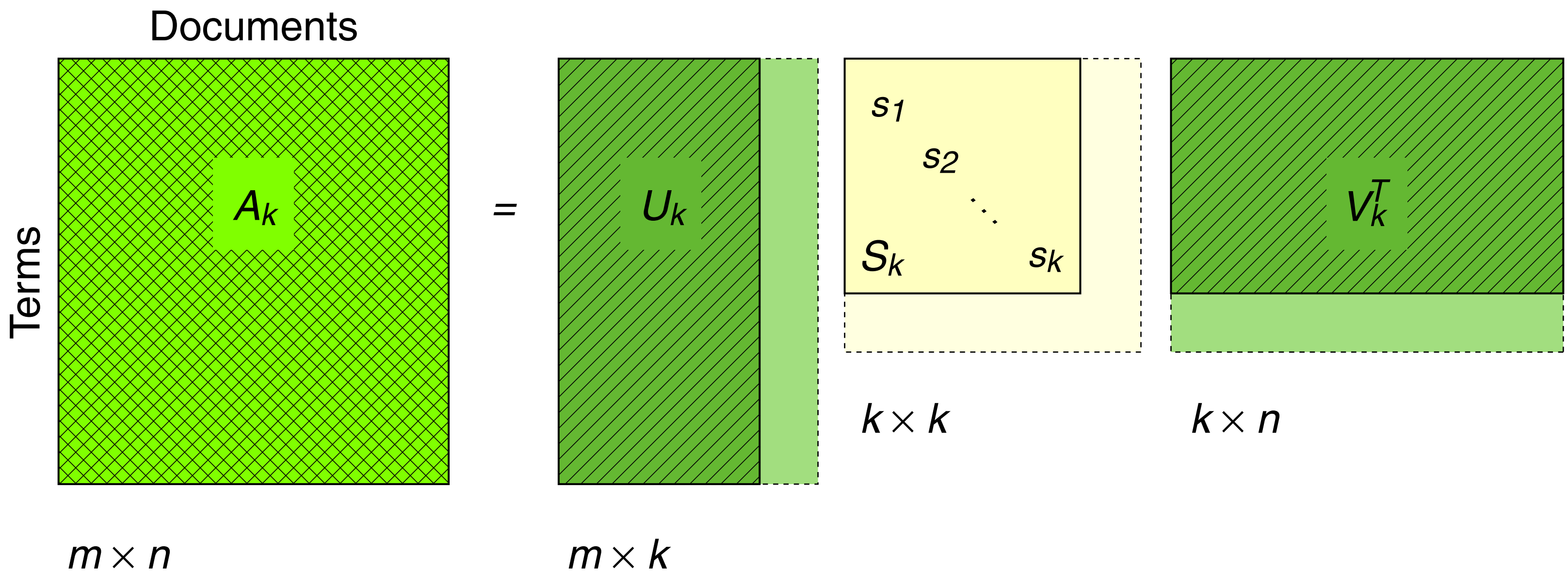
Singular value decomposition $A = USV^T$:



U is column orthonormal
S is diagonal, $r \leq \min\{m, n\}$
V^T is row orthonormal

Latent Semantic Indexing
Singular Value Decomposition

Dimensionality reduction $A_k = U_k S_k V_k^T$:



U_k is column orthonormal
S_k is diagonal, $k < r$
V_k^T is row orthonormal

Remarks:

- The eigenvalues of A result from the equation $\det(A - \lambda I) = 0$. This equation defines a polynomial of n-th degree that has n roots, which can be real or complex and repeated. The corresponding eigenvectors are orthogonal.
- A symmetric matrix has real eigenvalues. A positive-definite matrix has only positive eigenvalues.
- The singular value decomposition generalizes the eigen decomposition to rectangular matrices.
- Matrix multiplication and transposition: $(AB)^T = B^T A^T$
- Matrix diagonalization or eigen decomposition of a square matrix A: $A = PDP^{-1}$, where D is a diagonal matrix with the eigenvalues of A, and P contains the eigenvectors of A. A is diagonalizable, iff it has n linearly independent eigenvectors.
- $U^T = U^{-1}$, if U is an orthogonal matrix.
- $U^T U = I$, if U is a column orthonormal matrix.
- $U^T = U$, if U is a symmetric matrix.
- Reducing the $r \times r$ diagonal matrix S to the smaller $k \times k$ diagonal matrix S_k is done by omitting the smallest diagonal elements, presuming the column vectors of U_k and V_k are ordered accordingly.
- Typically, for a term-document matrix with rank of several thousands, k is chosen in the low hundreds.

Latent Semantic Indexing

Retrieval Model $\mathcal{R} = \langle \mathbf{D}, \mathbf{Q}, \rho \rangle$ [Generic Model] [Boolean Retrieval] [VSM] [BIM] [BM25] [ESA] [LM]

Document representations $\mathbf{D}$.

1. The document representations of the vector space model are combined to form an $m \times n$ term-document matrix A.
2. By dimensionality reduction, A is turned into a concept-document matrix $\mathbf{D} = V_k^T$. $\mathbf{D}$ represents the documents in a concept space (latent semantic space).

Query representations $\mathbf{Q}$.

Starting from a query q's vector space model representation $\mathbf{q}$, the following operation transforms $\mathbf{q}$ into the concept space:

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

Relevance function ρ.

ρ is applied directly on the representations of documents and queries in concept space. The retrieval functions of the vector space model can be directly applied (e.g., cosine similarity).

Latent Semantic Indexing

Document representations $\mathbf{D}$.

1. The document representations of the vector space model are combined to form an $m \times n$ term-document matrix A.
2. By dimensionality reduction, A is turned into a concept-document matrix $\mathbf{D} = V_k^T$. $\mathbf{D}$ represents the documents in a concept space (latent semantic space).

Query representations $\mathbf{Q}$.

Starting from a query q's vector space model representation $\mathbf{q}$, the following operation transforms $\mathbf{q}$ into the concept space:

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

Relevance function ρ.

ρ is applied directly on the representations of documents and queries in concept space. The retrieval functions of the vector space model can be directly applied (e.g., cosine similarity).

Latent Semantic Indexing

Example [Schek 2001]

Document collection D:

d_1	Human machine interface for Lab ABC computer applications.
d_2	A survey of user opinion of computer system response time.
d_3	The EPS user interface management system.
d_4	System and human system engineering testing of EPS.
d_5	Relation of user-perceived response time to error measurement.
d_6	The generation of random, binary, unordered trees.
d_7	The intersection graph of paths in trees.
d_8	Graph minors IV: Widths of trees and well-quasi-ordering.
d_9	Graph minors: A survey

Latent Semantic Indexing

Example [Schek 2001]

Document collection D:

d_1	Human machine interface for Lab ABC computer applications.
d_2	A survey of user opinion of computer system response time.
d_3	The EPS user interface management system.
d_4	System and human system engineering testing of EPS.
d_5	Relation of user-perceived response time to error measurement.
d_6	The generation of random, binary, unordered trees.
d_7	The intersection graph of paths in trees.
d_8	Graph minors IV: Widths of trees and well-quasi-ordering.
d_9	Graph minors: A survey

Query $\mathbf{q}$ = { human, computer, interaction }

Latent Semantic Indexing

Example [Schek 2001]

Document collection D:

d_1	Human machine interface for Lab ABC computer applications.
d_2	A survey of user opinion of computer system response time.
d_3	The EPS user interface management system.
d_4	System and human system engineering testing of EPS.
d_5	Relation of user-perceived response time to error measurement.
d_6	The generation of random, binary, unordered trees.
d_7	The intersection graph of paths in trees.
d_8	Graph minors IV: Widths of trees and well-quasi-ordering.
d_9	Graph minors: A survey

Query $\mathbf{q}$ = { human, computer, interaction }

The documents have many relations, transitivley relating the query to them.

Remarks:

- Retrieval in term-document space under the Boolean retrieval model with $\wedge$-connected terms in $\mathbf{q}$: result set $R = \emptyset$.
- Retrieval in term-document space under the Boolean retrieval model with $\vee$-connected terms in $\mathbf{q}$: result set $R = \{d_1, d_2, d_4\}$.
- Retrieval in term-document space under the vector space model: result set $R = \{d_1, d_2, d_4\}$.

Latent Semantic Indexing

Example: Term-Document Matrix A

	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

Terms occurring in only one document, and stop words are omitted.

Latent Semantic Indexing

Example: Singular Value Decomposition $A = U S V^T$

$U =$

0.2214	-0.1132	0.2890	-0.4148	-0.1063	-0.3410	0.5227	-0.0605	-0.4067
0.1976	-0.0721	0.1350	-0.5522	0.2818	0.4959	-0.0704	-0.0099	-0.1089
0.2405	0.0432	-0.1644	-0.5950	-0.1068	-0.2550	-0.3022	0.0623	0.4924
0.4036	0.0571	-0.3378	0.0991	0.3317	0.3848	0.0029	-0.0004	0.0123
0.6445	-0.1673	0.3611	0.3335	-0.1590	-0.2065	-0.1658	0.0343	0.2707
0.2650	0.1072	-0.4260	0.0738	0.0803	-0.1697	0.2829	-0.0161	-0.0539
0.2650	0.1072	-0.4260	0.0738	0.0803	-0.1697	0.2829	-0.0161	-0.0539
0.3008	-0.1413	0.3303	0.1881	0.1148	0.2722	0.0330	-0.0190	-0.1653
0.2059	0.2736	-0.1776	-0.0324	-0.5372	0.0809	-0.4669	-0.0363	-0.5794
0.0127	0.4902	0.2311	0.0248	0.5942	-0.3921	-0.2883	0.2546	-0.2254
0.0361	0.6228	0.2231	0.0007	-0.0683	0.1149	0.1596	-0.6811	0.2320
0.0318	0.4505	0.1411	-0.0087	-0.3005	0.2773	0.3395	0.6784	0.1825

Latent Semantic Indexing

Example: Singular Value Decomposition $A = USV^T$

$$U = \begin{array}{|rrrrrrrrr|}
\hline
0.2214 & -0.1132 & 0.2890 & -0.4148 & -0.1063 & -0.3410 & 0.5227 & -0.0605 & -0.4067 \\
0.1976 & -0.0721 & 0.1350 & -0.5522 & 0.2818 & 0.4959 & -0.0704 & -0.0099 & -0.1089 \\
0.2405 & 0.0432 & -0.1644 & -0.5950 & -0.1068 & -0.2550 & -0.3022 & 0.0623 & 0.4924 \\
0.4036 & 0.0571 & -0.3378 & 0.0991 & 0.3317 & 0.3848 & 0.0029 & -0.0004 & 0.0123 \\
0.6445 & -0.1673 & 0.3611 & 0.3335 & -0.1590 & -0.2065 & -0.1658 & 0.0343 & 0.2707 \\
0.2650 & 0.1072 & -0.4260 & 0.0738 & 0.0803 & -0.1697 & 0.2829 & -0.0161 & -0.0539 \\
0.2650 & 0.1072 & -0.4260 & 0.0738 & 0.0803 & -0.1697 & 0.2829 & -0.0161 & -0.0539 \\
0.3008 & -0.1413 & 0.3303 & 0.1881 & 0.1148 & 0.2722 & 0.0330 & -0.0190 & -0.1653 \\
0.2059 & 0.2736 & -0.1776 & -0.0324 & -0.5372 & 0.0809 & -0.4669 & -0.0363 & -0.5794 \\
0.0127 & 0.4902 & 0.2311 & 0.0248 & 0.5942 & -0.3921 & -0.2883 & 0.2546 & -0.2254 \\
0.0361 & 0.6228 & 0.2231 & 0.0007 & -0.0683 & 0.1149 & 0.1596 & -0.6811 & 0.2320 \\
0.0318 & 0.4505 & 0.1411 & -0.0087 & -0.3005 & 0.2773 & 0.3395 & 0.6784 & 0.1825 \\
\hline
\end{array}$$

$$S = \begin{array}{ccccccccc}
3.3409 & & & & & & & & \\
 & 2.5417 & & & & & & & \\
 & & 2.3539 & & & & & & \\
 & & & 1.6445 & & & & & \\
 & & & & 1.5048 & & & & \\
 & & & & & 1.3064 & & & \\
 & & & & & & 0.8459 & & \\
 & & & & & & & 0.5601 & \\
 & & & & & & & & 0.3637
\end{array}$$

Latent Semantic Indexing

Example: Singular Value Decomposition $A = U S V^T$

$$U = \begin{array}{|rrrrrrrrr|}
\hline
0.2214 & -0.1132 & 0.2890 & -0.4148 & -0.1063 & -0.3410 & 0.5227 & -0.0605 & -0.4067 \\
0.1976 & -0.0721 & 0.1350 & -0.5522 & 0.2818 & 0.4959 & -0.0704 & -0.0099 & -0.1089 \\
0.2405 & 0.0432 & -0.1644 & -0.5950 & -0.1068 & -0.2550 & -0.3022 & 0.0623 & 0.4924 \\
0.4036 & 0.0571 & -0.3378 & 0.0991 & 0.3317 & 0.3848 & 0.0029 & -0.0004 & 0.0123 \\
0.6445 & -0.1673 & 0.3611 & 0.3335 & -0.1590 & -0.2065 & -0.1658 & 0.0343 & 0.2707 \\
0.2650 & 0.1072 & -0.4260 & 0.0738 & 0.0803 & -0.1697 & 0.2829 & -0.0161 & -0.0539 \\
0.2650 & 0.1072 & -0.4260 & 0.0738 & 0.0803 & -0.1697 & 0.2829 & -0.0161 & -0.0539 \\
0.3008 & -0.1413 & 0.3303 & 0.1881 & 0.1148 & 0.2722 & 0.0330 & -0.0190 & -0.1653 \\
0.2059 & 0.2736 & -0.1776 & -0.0324 & -0.5372 & 0.0809 & -0.4669 & -0.0363 & -0.5794 \\
0.0127 & 0.4902 & 0.2311 & 0.0248 & 0.5942 & -0.3921 & -0.2883 & 0.2546 & -0.2254 \\
0.0361 & 0.6228 & 0.2231 & 0.0007 & -0.0683 & 0.1149 & 0.1596 & -0.6811 & 0.2320 \\
0.0318 & 0.4505 & 0.1411 & -0.0087 & -0.3005 & 0.2773 & 0.3395 & 0.6784 & 0.1825 \\
\hline
\end{array}$$

$$S = \begin{array}{ccccccccc}
3.3409 & & & & & & & & \\
& 2.5417 & & & & & & & \\
& & 2.3539 & & & & & & \\
& & & 1.6445 & & & & & \\
& & & & 1.5048 & & & & \\
& & & & & 1.3064 & & & \\
& & & & & & 0.8459 & & \\
& & & & & & & 0.5601 & \\
& & & & & & & & 0.3637
\end{array}$$

$$V^T = \begin{array}{|rrrrrrrrr|}
\hline
0.1974 & 0.6060 & 0.4629 & 0.5421 & 0.2795 & 0.0038 & 0.0146 & 0.0241 & 0.0820 \\
-0.0559 & 0.1656 & -0.1273 & -0.2318 & 0.1068 & 0.1928 & 0.4379 & 0.6151 & 0.5299 \\
0.1103 & -0.4973 & 0.2076 & 0.5699 & -0.5054 & 0.0982 & 0.1930 & 0.2529 & 0.0793 \\
-0.9498 & -0.0286 & 0.0416 & 0.2677 & 0.1500 & 0.0151 & 0.0155 & 0.0102 & -0.0246 \\
0.0457 & -0.2063 & 0.3783 & -0.2056 & 0.3272 & 0.3948 & 0.3495 & 0.1498 & -0.6020 \\
-0.0766 & -0.2565 & 0.7244 & -0.3689 & 0.0348 & -0.3002 & -0.2122 & 0.0001 & 0.3622 \\
0.1773 & -0.4330 & -0.2369 & 0.2648 & 0.6723 & -0.3408 & -0.1522 & 0.2491 & 0.0380 \\
-0.0144 & 0.0493 & 0.0088 & -0.0195 & -0.0583 & 0.4545 & -0.7615 & 0.4496 & -0.0696 \\
-0.0637 & 0.2428 & 0.0241 & -0.0842 & -0.2624 & -0.6198 & 0.0180 & 0.5199 & -0.4535 \\
\hline
\end{array}$$

Latent Semantic Indexing

Example: Dimensionality Reduction $A_k = U_k S_k V_k^T$

U_k

0.2214	-0.1132
0.1976	-0.0721
0.2405	0.0432
0.4036	0.0571
0.6445	-0.1673
0.2650	0.1072
0.2650	0.1072
0.3008	-0.1413
0.2059	0.2736
0.0127	0.4902
0.0361	0.6228
0.0318	0.4505

S_k

3.3409	
	2.5417

V_k^T

0.1974	0.6060	0.4629	0.5421	0.2795	0.0038	0.0146	0.0241	0.0820
-0.0559	0.1656	-0.1273	-0.2318	0.1068	0.1928	0.4379	0.6151	0.5299

Latent Semantic Indexing

Example: Dimensionality Reduction $A_k = U_k S_k V_k^T$

U_k

0.2214	-0.1132
0.1976	-0.0721
0.2405	0.0432
0.4036	0.0571
0.6445	-0.1673
0.2650	0.1072
0.2650	0.1072
0.3008	-0.1413
0.2059	0.2736
0.0127	0.4902
0.0361	0.6228
0.0318	0.4505

S_k

3.3409	
	2.5417

V_k^T

0.1974	0.6060	0.4629	0.5421	0.2795	0.0038	0.0146	0.0241	0.0820
-0.0559	0.1656	-0.1273	-0.2318	0.1068	0.1928	0.4379	0.6151	0.5299

A_k

0.1621	0.4005	0.3790	0.4676	0.1760	-0.0527	-0.1151	-0.1591	-0.0918
0.1406	0.3698	0.3290	0.4004	0.1650	-0.0328	-0.0706	-0.0968	-0.0430
0.1524	0.5050	0.3579	0.4101	0.2362	0.0242	0.0598	0.0869	0.1240
0.2580	0.8411	0.6057	0.6974	0.3923	0.0331	0.0832	0.1218	0.1874
0.4488	1.2344	1.0509	1.2658	0.5563	-0.0738	-0.1547	-0.2096	-0.0489
0.1596	0.5817	0.3752	0.4169	0.2765	0.0559	0.1322	0.1889	0.2169
0.1596	0.5817	0.3752	0.4169	0.2765	0.0559	0.1322	0.1889	0.2169
0.2185	0.5496	0.5110	0.6281	0.2425	-0.0654	-0.1425	-0.1966	-0.1079
0.0969	0.5321	0.2299	0.2118	0.2665	0.1368	0.3146	0.4444	0.4250
-0.0613	0.2321	-0.1389	-0.2656	0.1449	0.2404	0.5461	0.7674	0.6637
-0.0647	0.3353	-0.1456	-0.3014	0.2028	0.3057	0.6949	0.9766	0.8487
-0.0431	0.2539	-0.0967	-0.2079	0.1519	0.2212	0.5029	0.7069	0.6155

Latent Semantic Indexing

Example: Dimensionality Reduction $A_k = U_k S_k V_k^T$

U_k

0.2214	-0.1132
0.1976	-0.0721
0.2405	0.0432
0.4036	0.0571
0.6445	-0.1673
0.2650	0.1072
0.2650	0.1072
0.3008	-0.1413
0.2059	0.2736
0.0127	0.4902
0.0361	0.6228
0.0318	0.4505

S_k

3.3409	
	2.5417

V_k^T

0.1974	0.6060	0.4629	0.5421	0.2795	0.0038	0.0146	0.0241	0.0820
-0.0559	0.1656	-0.1273	-0.2318	0.1068	0.1928	0.4379	0.6151	0.5299

A_k

0.1621	0.4005	0.3790	0.4676	0.1760	-0.0527	-0.1151	-0.1591	-0.0918
0.1406	0.3698	0.3290	0.4004	0.1650	-0.0328	-0.0706	-0.0968	-0.0430
0.1524	0.5050	0.3579	0.4101	0.2362	0.0242	0.0598	0.0869	0.1240
0.2580	0.8411	0.6057	0.6974	0.3923	0.0331	0.0832	0.1218	0.1874
0.4488	1.2344	1.0509	1.2658	0.5563	-0.0738	-0.1547	-0.2096	-0.0489
0.1596	0.5817	0.3752	0.4169	0.2765	0.0559	0.1322	0.1889	0.2169
0.1596	0.5817	0.3752	0.4169	0.2765	0.0559	0.1322	0.1889	0.2169
0.2185	0.5496	0.5110	0.6281	0.2425	-0.0654	-0.1425	-0.1966	-0.1079
0.0969	0.5321	0.2299	0.2118	0.2665	0.1368	0.3146	0.4444	0.4250
-0.0613	0.2321	-0.1389	-0.2656	0.1449	0.2404	0.5461	0.7674	0.6637
-0.0647	0.3353	-0.1456	-0.3014	0.2028	0.3057	0.6949	0.9766	0.8487
-0.0431	0.2539	-0.0967	-0.2079	0.1519	0.2212	0.5029	0.7069	0.6155

$\mathbf{q}$

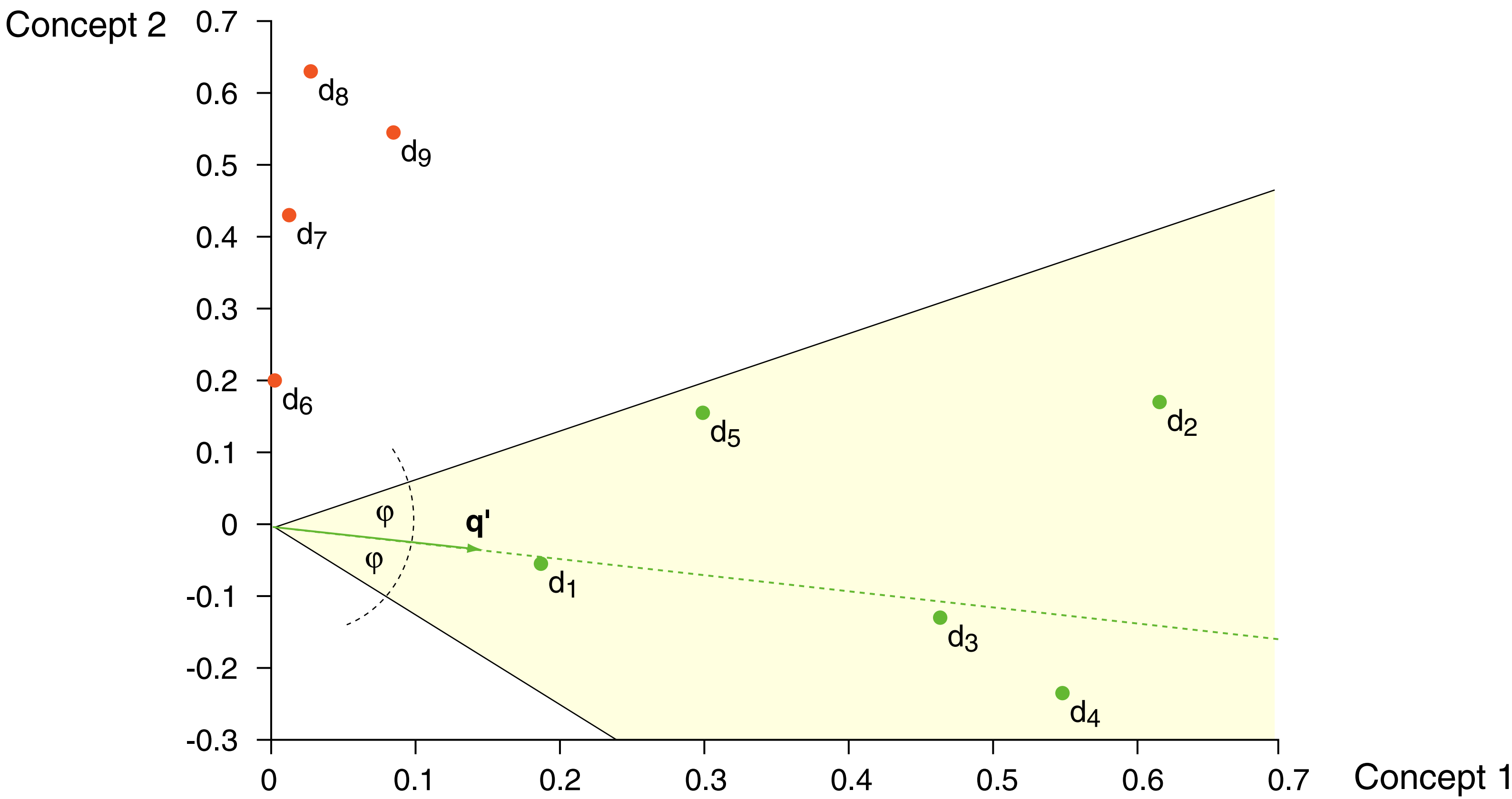
1
0
1
0
0
0
0
0
0
0
0
0

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

0.1382
-0.0276

Latent Semantic Indexing

Example: Retrieval in Concept Space

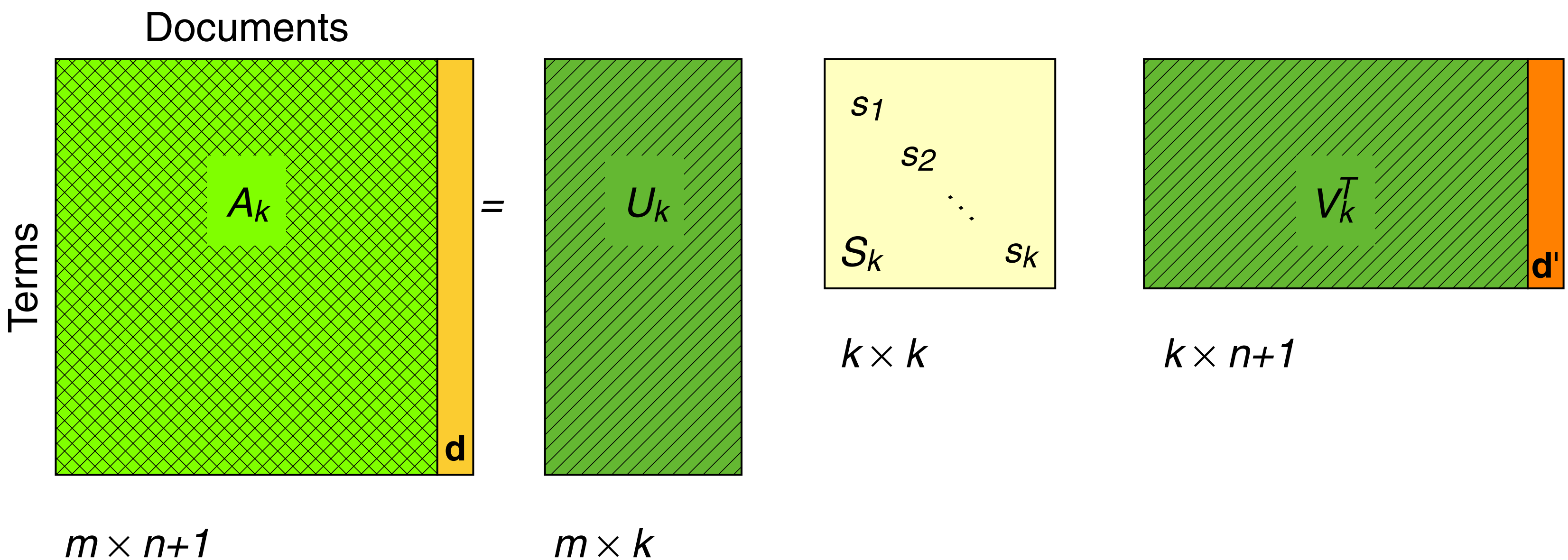


$\varphi = 30°$ ➜ Documents must have a cosine similarity of >0.87 to the query vector $\mathbf{q}'$.

Latent Semantic Indexing

Retrieval Model $\mathcal{R} = \langle \mathbf{D}, \mathbf{Q}, \rho \rangle$ (continued)

Adding new documents:

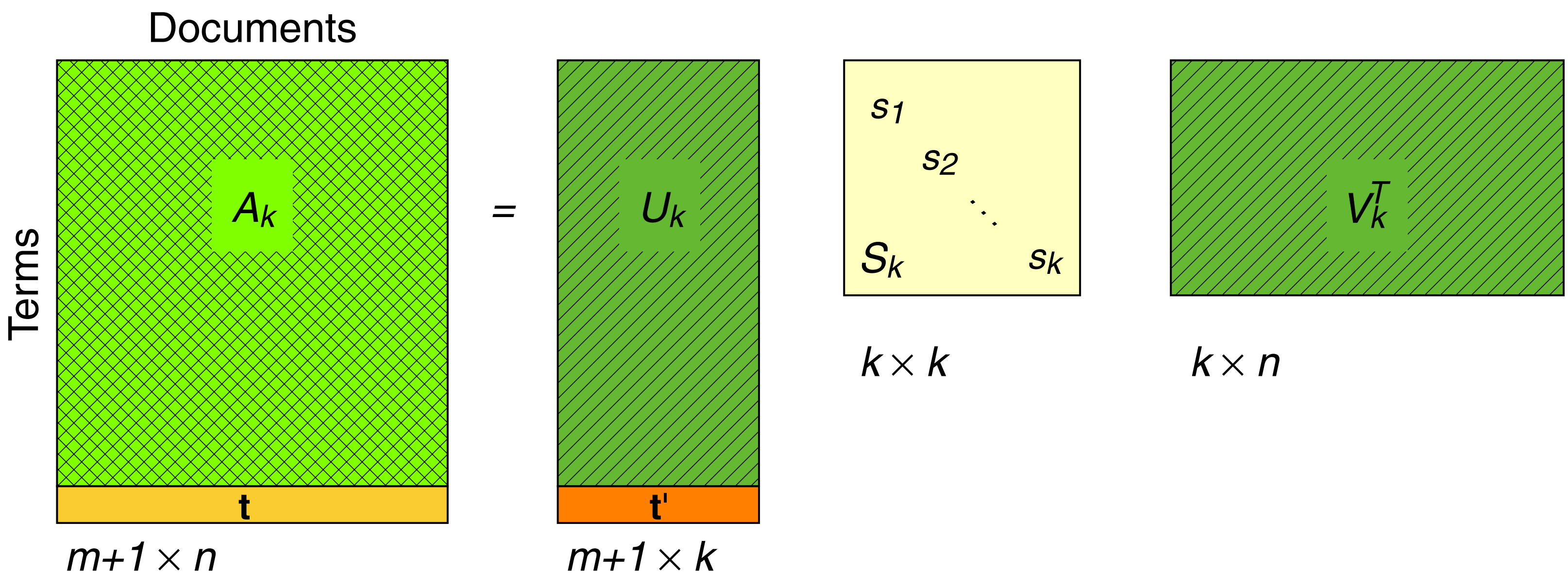


1. add original document vector $\mathbf{d}$ as column to A_k
2. compute reduced document vector $\mathbf{d}' = \mathbf{d}^T U_k S_k^{-1}$ (compare with query representation)
3. add reduced document vector $\mathbf{d}'$ to V_k^T

Latent Semantic Indexing

Retrieval Model $\mathcal{R} = \langle \mathbf{D}, \mathbf{Q}, \rho \rangle$ (continued)

Adding new terms:



1. add original term vector $\mathbf{t}$ as row to A_k
2. compute reduced term vector $\mathbf{t}' = \mathbf{t}^T V_k S_k^{-1}$
3. add reduced term vector $\mathbf{t}'$ as row to U_k

Latent Semantic Indexing

Example 2 [Schek 2001]

	d_1	d_2	d_3	d_4	d_5	d_6	d_7
data	1	2	1	5	0	0	0
information	1	2	1	5	0	0	0
retrieval	1	2	1	5	0	0	0
brain	0	0	0	0	2	3	1
lung	0	0	0	0	2	3	1

Latent Semantic Indexing

Example 2 [Schek 2001]

	d_1	d_2	d_3	d_4	d_5	d_6	d_7
data	1	2	1	5	0	0	0
information	1	2	1	5	0	0	0
retrieval	1	2	1	5	0	0	0
brain	0	0	0	0	2	3	1
lung	0	0	0	0	2	3	1

$A = U S V^T$, approximates: $A_k = U_k S_k V_k^T$

Rank$(A) = $ 2, so that with $k = 2$, it follows that $A_2 = A,\ U_2 = U,\ S_2 = S,\ V_2^T = V^T$:

Latent Semantic Indexing

Example 2 [Schek 2001]

	d_1	d_2	d_3	d_4	d_5	d_6	d_7
data	1	2	1	5	0	0	0
information	1	2	1	5	0	0	0
retrieval	1	2	1	5	0	0	0
brain	0	0	0	0	2	3	1
lung	0	0	0	0	2	3	1

$A = USV^T$, approximates: $A_k = U_k S_k V_k^T$

Rank$(A) = 2$, so that with $k = 2$, it follows that $A_2 = A,\ U_2 = U,\ S_2 = S,\ V_2^T = V^T$:

$$A = \begin{pmatrix} 0.58 & 0 \\ 0.58 & 0 \\ 0.58 & 0 \\ 0 & 0.71 \\ 0 & 0.71 \\ 0 & 0.71 \end{pmatrix} \times \begin{pmatrix} 9.64 & 0 \\ 0 & 5.29 \end{pmatrix} \times \begin{pmatrix} 0.18 & 0.36 & 0.18 & 0.9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.53 & 0.8 & 0.27 \end{pmatrix}$$

Remarks:

- There are two concepts; the computer science concept {data, information, retrieval} and the medicine concept {brain, lung}.

Latent Semantic Indexing

Example 2: Document Similarity Matrix $A^T A$

$$A^T A = \begin{pmatrix} \mathbf{3} & \mathbf{6} & \mathbf{6} & \mathbf{15} & 0 & 0 & 0 \\ \mathbf{6} & \mathbf{12} & \mathbf{6} & \mathbf{30} & 0 & 0 & 0 \\ \mathbf{3} & \mathbf{6} & \mathbf{6} & \mathbf{15} & 0 & 0 & 0 \\ \mathbf{15} & \mathbf{37} & \mathbf{15} & \mathbf{75} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{8} & \mathbf{12} & \mathbf{4} \\ 0 & 0 & 0 & 0 & \mathbf{12} & \mathbf{18} & \mathbf{6} \\ 0 & 0 & 0 & 0 & \mathbf{4} & \mathbf{6} & \mathbf{2} \end{pmatrix}$$

Latent Semantic Indexing

Example 2: Document Similarity Matrix $A^T A$

$$A^T A = \begin{pmatrix} \mathbf{3} & \mathbf{6} & \mathbf{6} & \mathbf{15} & 0 & 0 & 0 \\ \mathbf{6} & \mathbf{12} & \mathbf{6} & \mathbf{30} & 0 & 0 & 0 \\ \mathbf{3} & \mathbf{6} & \mathbf{6} & \mathbf{15} & 0 & 0 & 0 \\ \mathbf{15} & \mathbf{37} & \mathbf{15} & \mathbf{75} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{8} & \mathbf{12} & \mathbf{4} \\ 0 & 0 & 0 & 0 & \mathbf{12} & \mathbf{18} & \mathbf{6} \\ 0 & 0 & 0 & 0 & \mathbf{4} & \mathbf{6} & \mathbf{2} \end{pmatrix}$$

Interpretation. $A^T A$ shows document clusters.

Latent Semantic Indexing

Example 2: Document Similarity Matrix $A^T A$

$$A^T A = \begin{pmatrix} \mathbf{3} & \mathbf{6} & \mathbf{6} & \mathbf{15} & 0 & 0 & 0 \\ \mathbf{6} & \mathbf{12} & \mathbf{6} & \mathbf{30} & 0 & 0 & 0 \\ \mathbf{3} & \mathbf{6} & \mathbf{6} & \mathbf{15} & 0 & 0 & 0 \\ \mathbf{15} & \mathbf{37} & \mathbf{15} & \mathbf{75} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{8} & \mathbf{12} & \mathbf{4} \\ 0 & 0 & 0 & 0 & \mathbf{12} & \mathbf{18} & \mathbf{6} \\ 0 & 0 & 0 & 0 & \mathbf{4} & \mathbf{6} & \mathbf{2} \end{pmatrix}$$

Interpretation. $A^T A$ shows document clusters.

Explanation. Since $A^T A = V S^2 V^T$, the rows of V_k^T are eigenvectors of $A^T A$, which denote uncorrelated principal directions of documents clusters:

$$V_2^T = \begin{pmatrix} 0.18 & 0.36 & 0.18 & 0.9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.53 & 0.8 & 0.27 \end{pmatrix}$$

➜ If d_1 is relevant, so are d_2, d_3, d_4, but not d_5, d_6, d_7.

Latent Semantic Indexing

Example 2: Term Similarity Matrix AA^T

$$AA^T = \begin{pmatrix} \mathbf{31} & \mathbf{31} & \mathbf{31} & 0 & 0 \\ \mathbf{31} & \mathbf{31} & \mathbf{31} & 0 & 0 \\ \mathbf{31} & \mathbf{31} & \mathbf{31} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{14} & \mathbf{14} \\ 0 & 0 & 0 & \mathbf{14} & \mathbf{14} \end{pmatrix}$$

Latent Semantic Indexing

Example 2: Term Similarity Matrix AA^T

$$AA^T = \begin{pmatrix} \mathbf{31} & \mathbf{31} & \mathbf{31} & 0 & 0 \\ \mathbf{31} & \mathbf{31} & \mathbf{31} & 0 & 0 \\ \mathbf{31} & \mathbf{31} & \mathbf{31} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{14} & \mathbf{14} \\ 0 & 0 & 0 & \mathbf{14} & \mathbf{14} \end{pmatrix}$$

Interpretation. AA^T shows term clusters, i.e., concepts, possibly synonyms.

Latent Semantic Indexing

Example 2: Term Similarity Matrix AA^T

$$AA^T = \begin{pmatrix} \mathbf{31} & \mathbf{31} & \mathbf{31} & 0 & 0 \\ \mathbf{31} & \mathbf{31} & \mathbf{31} & 0 & 0 \\ \mathbf{31} & \mathbf{31} & \mathbf{31} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{14} & \mathbf{14} \\ 0 & 0 & 0 & \mathbf{14} & \mathbf{14} \end{pmatrix}$$

Interpretation. AA^T shows term clusters, i.e., concepts, possibly synonyms.

Explanation. Since $AA^T = US^2U^T$, the columns of U_k are the eigenvectors of AA^T, which denote uncorrelated principal directions for concepts:

$$U_2 = \begin{pmatrix} 0.58 & 0 \\ 0.58 & 0 \\ 0.58 & 0 \\ 0 & 0.71 \\ 0 & 0.71 \\ 0 & 0.71 \end{pmatrix}$$

Latent Semantic Indexing

Discussion

Advantages:

- automatic discovery of hidden concepts
- syntactic detection of synonyms
- semantic query expansion based on syntactical analysis—not based on relevance feedback

Disadvantages:

- the effect of LSI in this domain is not fully understood; a theoretical connection to linguistics is only partially available
- LSI works best in a closed-set retrieval situation: the document collection is known, available, and does not change a lot
- the singular value decomposition is computationally expensive, $O(n^3)$

Explicit Semantic Analysis

Concept Hypothesis

Consideration:

An explicit manifestation of a concept is a document talking about it. However, most documents cover more than one concept at a time, and hardly any in depth.

Arguably, a (long) Wikipedia article covers exactly one concept in depth.

Idea:

Given a set D^* of Wikipedia articles, interpret their normalized representations $\mathbf{D}^*$ under the vector space model as explicit concepts, spanning a concept space.

Then a document can be embedded into the concept space, e.g., by computing its similarity under the vector space model to the concept representations in $\mathbf{D}^*$.

Caveat:

This concept hypothesis has been falsified. Other kinds of documents work, too.

➜ We say that a document in D^* represents a pseudo-concept.

Explicit Semantic Analysis

Retrieval Model $\mathcal{R} = \langle \mathbf{D}, \mathbf{Q}, \rho \rangle$ [Generic Model] [Boolean Retrieval] [VSM] [BIM] [BM25] [LSI] [LM]

Document representations $\mathbf{D}$.

1. Given a collection D^* of index documents, let A_{D^*} denote an $m \times n$ term-document matrix of the combined, normalized index document representations under the vector space model.
2. Starting from a normalized document d's vector space model representation $\mathbf{d}$, its ESA representation is computed as follows:

$$\mathbf{d}' = A_{D^*}^T \cdot \mathbf{d}$$

   $\mathbf{D}$ represents the documents in a pseudo-concept space, where each document $d^* \in D^*$ is interpreted as manifestation of one (orthogonal) pseudo-concept.

Query representations $\mathbf{Q}$.

Query representations $\mathbf{q}'$ are computed like document representations.

Relevance function ρ.

ρ is applied directly on the representations of documents and queries in concept space. The retrieval functions of the vector space model can be directly applied (e.g., cosine similarity).

Explicit Semantic Analysis

Retrieval Model $\mathcal{R} = \langle \mathbf{D}, \mathbf{Q}, \rho \rangle$ [Generic Model] [Boolean Retrieval] [VSM] [BIM] [BM25] [LSI] [LM]

Document representations $\mathbf{D}$.

1. Given a collection D^* of index documents, let A_{D^*} denote an $m \times n$ term-document matrix of the combined, normalized index document representations under the vector space model.
2. Starting from a normalized document d's vector space model representation $\mathbf{d}$, its ESA representation is computed as follows:

$$\mathbf{d}' = A_{D^*}^T \cdot \mathbf{d}$$

$\mathbf{D}$ represents the documents in a pseudo-concept space, where each document $d^* \in D^*$ is interpreted as manifestation of one (orthogonal) pseudo-concept.

Query representations $\mathbf{Q}$.

Query representations $\mathbf{q}'$ are computed like document representations.

Relevance function ρ.

ρ is applied directly on the representations of documents and queries in concept space. The retrieval functions of the vector space model can be directly applied (e.g., cosine similarity).

Explicit Semantic Analysis

Document Representation

Let $D^* = \{d_1,\ \ldots,\ d_m\}$ denote a collection of documents, called index documents, and let $\mathbf{D}^*$ be the set of document representations under the vector space model.

Under explicit semantic analysis, a document d is represented by its vector space model similarities to D^*:

$$\mathbf{d}' \quad = \quad (\rho_{\text{VSM}}(\mathbf{d}_1, \mathbf{d}),\ \ldots,\ \rho_{\text{VSM}}(\mathbf{d}_m, \mathbf{d}))^T$$

Let ρ_{VSM} be the cosine similarity measure, and let $||\mathbf{d}_i|| = ||\mathbf{d}|| = 1$:

$$\mathbf{d}' \quad = \quad (\mathbf{d}_1^T \cdot \mathbf{d},\ \ldots,\ \mathbf{d}_m^T \cdot \mathbf{d})^T \quad = \quad A_{D^*}^T \cdot \mathbf{d},$$

where A_{D^*} is the term-document matrix of D^*.

Explicit Semantic Analysis

Relevance Function ρ

Given a query q and a document d, and an index collection D^*, let $\mathbf{q}'$ and $\mathbf{d}'$ denote the representations of q and d under the explicit semantic analysis model.

The relevance of document d to query q is computed using the cosine similarity:

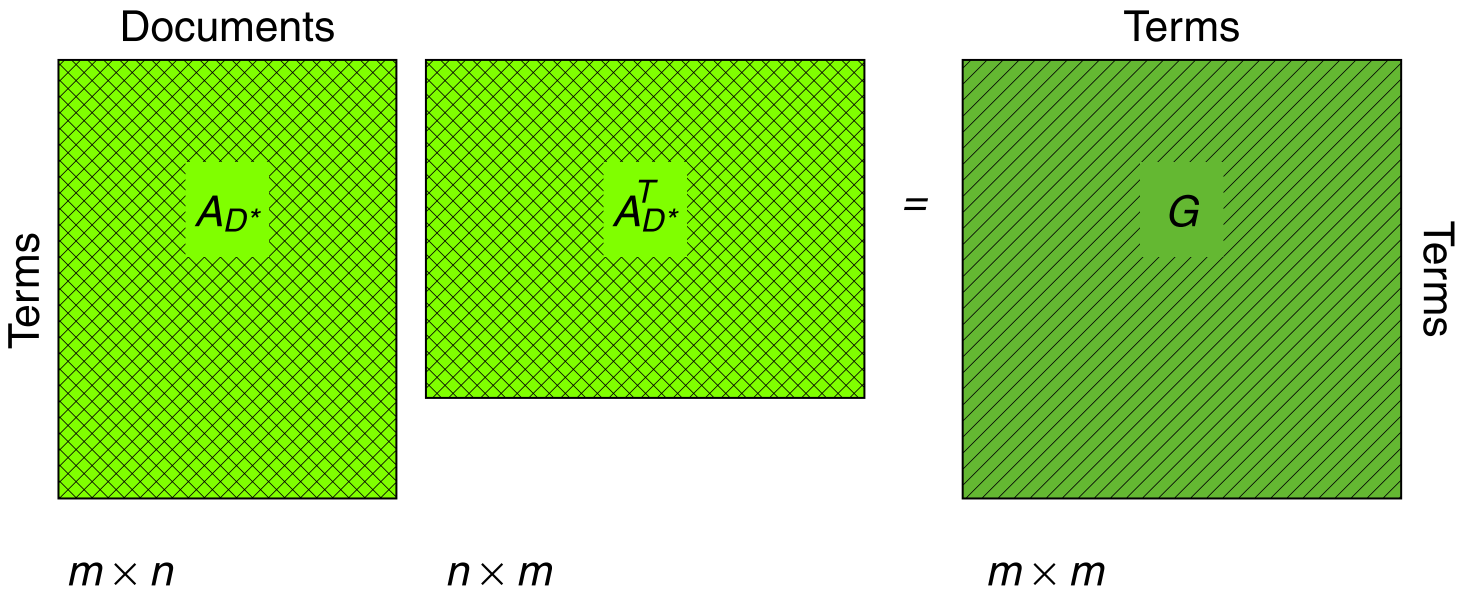
$$
\begin{aligned}
\rho(\mathbf{q}', \mathbf{d}') &= \frac{\mathbf{q}'^T \cdot \mathbf{d}'}{||\mathbf{q}'|| \cdot ||\mathbf{d}'||} && \mathcal{O}(|q| \cdot |D^*|) \\
&= \frac{(A_{D^*}^T \cdot \mathbf{q})^T \cdot A_{D^*}^T \cdot \mathbf{d}}{||\mathbf{q}'|| \cdot ||\mathbf{d}'||} \\
&= \frac{\mathbf{q}^T \cdot A_{D^*} \cdot A_{D^*}^T \cdot \mathbf{d}}{\sqrt{\mathbf{q}^T \cdot A_{D^*} \cdot A_{D^*}^T \cdot \mathbf{q}} \cdot ||\mathbf{d}'||} && \mathcal{O}(|q|)
\end{aligned}
$$

The majority of the computations can be done offline.

Explicit Semantic Analysis

Relevance Function ρ

The multiplication $A_{D^*} \cdot A_{D^*}^T$ yields a term co-occurrence matrix G:



Given term t_i and t_j from T, the matrix G has a non-zero value in its i-th row and its j-th value iff a document $d \in D^*$ exists that contains both t_i and t_j. Thus:

$$\rho(\mathbf{q}, \mathbf{d}) = \frac{\mathbf{q}^T \cdot G \cdot \mathbf{d}}{\sqrt{\mathbf{q}^T \cdot G \cdot \mathbf{q}} \cdot \sqrt{\mathbf{d}^T \cdot G \cdot \mathbf{d}}}$$

Explicit Semantic Analysis

Discussion

Advantages:

- simple model
- better retrieval performance than basic models
- can be improved by using a tailored index collection

Disadvantages:

- concept hypothesis is weak; has been shown to also work with random documents
- requires high-dimensional representations >10.000 index documents
- computationally expensive