Lecture 5.
Algebraic IR Models

Generalized Vector Model

Latent Semantic Indexing

Generalized Vector Model

Generalized Vector Model

- Classic models enforce independence of index terms
- For instance, in the Vector model
  - A set of term vectors $\{\vec{k}_1, \vec{k}_2, \ldots, \vec{k}_t\}$ are linearly independent
  - Frequently, this is interpreted as $\forall_{i,j} \Rightarrow \vec{k}_i \bullet \vec{k}_j = 0$
- In the generalized vector space model, two index term vectors might be non-orthogonal

	K_1	K_2	K_3	$q \bullet d_j$
d_1	2	0	1	5
d_2	1	0	0	1
d_3	0	1	3	11
d_4	2	0	0	2
d_5	1	2	4	17
d_6	1	2	0	5
d_7	0	5	0	10
q	1	2	3	

Key Idea

- As before, let $w_{i,j}$ be the weight associated with $[k_i, d_j]$ and V = {k_1, k_2, ..., k_t} be the set of all terms
- If the $w_{i,j}$ weights are binary, all patterns of occurrence of terms within docs can be represented by minterms:

$$
\begin{array}{rcl}
 & & (k_1, k_2, k_3, \ldots, k_t) \\
m_1 & = & (0,\ 0,\ 0,\ \ \ldots,\ 0) \\
m_2 & = & (1,\ 0,\ 0,\ \ \ldots,\ 0) \\
m_3 & = & (0,\ 1,\ 0,\ \ \ldots,\ 0) \\
m_4 & = & (1,\ 1,\ 0,\ \ \ldots,\ 0) \\
 & \vdots & \\
m_{2^t} & = & (1,\ 1,\ 1,\ \ \ldots,\ 1)
\end{array}
$$

For instance, m_2 indicates documents in which solely the term k_1 occurs

Key Idea

- For any document d_j, there is a minterm m_r that includes exactly the terms that occur in the document
- Let us define the following set of minterm vectors $\vec{m}_r$,

$$\begin{array}{rcl} & & 1, 2, \ldots, 2^t \\ \vec{m}_1 & = & (1, 0, \ldots, 0) \\ \vec{m}_2 & = & (0, 1, \ldots, 0) \\ & \vdots & \\ \vec{m}_{2^t} & = & (0, 0, \ldots, 1) \end{array}$$

Notice that we can associate each unit vector $\vec{m}_r$ with a minterm m_r, and that $\vec{m}_i \bullet \vec{m}_j = 0$ for all $i \neq j$

Key Idea

- Pairwise orthogonality among the $\vec{m}_r$ vectors does not imply independence among the index terms
- On the contrary, index terms are now correlated by the $\vec{m}_r$ vectors
  - For instance, the vector $\vec{m}_4$ is associated with the minterm $m_4 = (1, 1, \ldots, 0)$
  - This minterm induces a dependency between terms k_1 and k_2
  - Thus, if such document exists in a collection, we say that the minterm m_4 is active
- The model adopts the idea that co-occurrence of terms induces dependencies among these terms

Forming the Term Vectors

- Let $on(i, m_r)$ return the weight $\{0, 1\}$ of the index term k_i in the minterm m_r
- The vector associated with the term k_i is computed as:

$$\vec{k}_i = \frac{\sum_{\forall r} on(i, m_r)\ c_{i,r}\ \vec{m}_r}{\sqrt{\sum_{\forall r} on(i, m_r)\ c_{i,r}^2}}$$

$$c_{i,r} = \sum_{d_j \mid c(d_j)=m_r} w_{i,j}$$

- Notice that for a collection of size N, only N minterms affect the ranking (and not 2^t)

Dependency between Index Terms

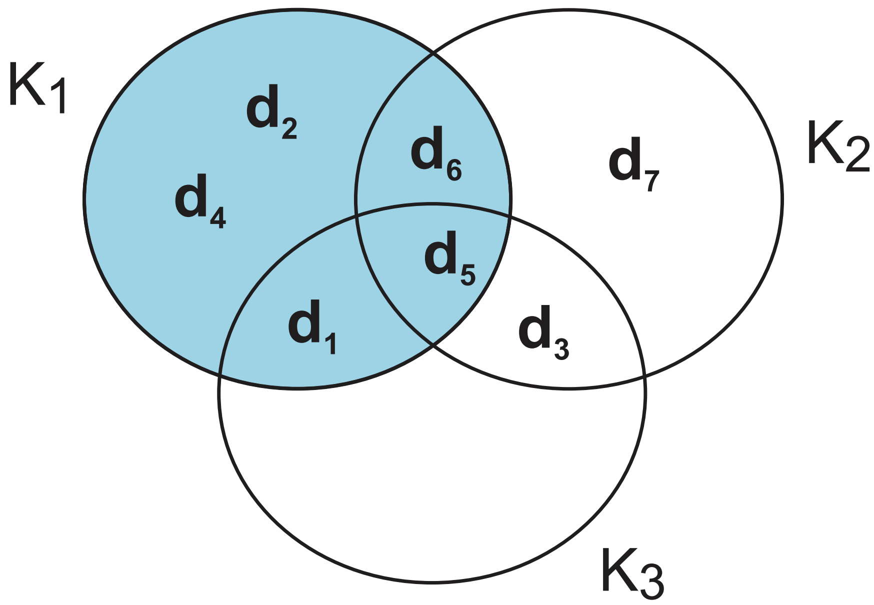
- A degree of correlation between the terms k_i and k_j can now be computed as:

$$\vec{k}_i \bullet \vec{k}_j = \sum_{\forall r} on(i, m_r) \times c_{i,r} \times on(j, m_r) \times c_{j,r}$$

- This degree of correlation sums up the dependencies between k_i and k_j induced by the docs in the collection

The Generalized Vector Model

- An Example



	K_1	K_2	K_3
d_1	2	0	1
d_2	1	0	0
d_3	0	1	3
d_4	2	0	0
d_5	1	2	4
d_6	1	2	0
d_7	0	5	0
q	1	2	3

Computation of $c_{i,r}$

	K_1	K_2	K_3
d_1	2	0	1
d_2	1	0	0
d_3	0	1	3
d_4	2	0	0
d_5	1	2	4
d_6	0	2	2
d_7	0	5	0
q	1	2	3

	K_1	K_2	K_3
$d_1 = m_6$	1	0	1
$d_2 = m_2$	1	0	0
$d_3 = m_7$	0	1	1
$d_4 = m_2$	1	0	0
$d_5 = m_8$	1	1	1
$d_6 = m_7$	0	1	1
$d_7 = m_3$	0	1	0
$q = m_8$	1	1	1

	$c_{1,r}$	$c_{2,r}$	$c_{3,r}$
m_1	0	0	0
m_2	3	0	0
m_3	0	5	0
m_4	0	0	0
m_5	0	0	0
m_6	2	0	1
m_7	0	3	5
m_8	1	2	4

Assume wij=fij in this example.

m1	0	0	0
m2	1	0	0
m3	0	1	0
m4	1	1	0
m5	0	0	1
m6	1	0	1
m7	0	1	1
m8	1	1	1

$$\vec{k}_i = \frac{\sum_{\forall r} on(i, m_r)\ c_{i,r}\ \vec{m}_r}{\sqrt{\sum_{\forall r} on(i, m_r)\ c_{i,r}^2}}$$

$$c_{i,r} = \sum_{d_j \mid c(d_j)=m_r} w_{i,j}$$

Computation of $\overrightarrow{k_i}$

- $\overrightarrow{k_1} = \frac{(3\vec{m}_2+2\vec{m}_6+\vec{m}_8)}{\sqrt{3^2+2^2+1^2}}$
- $\overrightarrow{k_2} = \frac{(5\vec{m}_3+3\vec{m}_7+2\vec{m}_8)}{\sqrt{5+3+2}}$
- $\overrightarrow{k_3} = \frac{(1\vec{m}_6+5\vec{m}_7+4\vec{m}_8)}{\sqrt{1+5+4}}$

	$c_{1,r}$	$c_{2,r}$	$c_{3,r}$
m_1	0	0	0
m_2	3	0	0
m_3	0	5	0
m_4	0	0	0
m_5	0	0	0
m_6	2	0	1
m_7	0	3	5
m_8	1	2	4

Computation of Document Vectors

- $\overrightarrow{d_1} = 2\overrightarrow{k_1} + \overrightarrow{k_3}$
- $\overrightarrow{d_2} = \overrightarrow{k_1}$
- $\overrightarrow{d_3} = \overrightarrow{k_2} + 3\overrightarrow{k_3}$
- $\overrightarrow{d_4} = 2\overrightarrow{k_1}$
- $\overrightarrow{d_5} = \overrightarrow{k_1} + 2\overrightarrow{k_2} + 4\overrightarrow{k_3}$
- $\overrightarrow{d_6} = 2\overrightarrow{k_2} + 2\overrightarrow{k_3}$
- $\overrightarrow{d_7} = 5\overrightarrow{k_2}$
- $\overrightarrow{q} = \overrightarrow{k_1} + 2\overrightarrow{k_2} + 3\overrightarrow{k_3}$

	K_1	K_2	K_3
d_1	2	0	1
d_2	1	0	0
d_3	0	1	3
d_4	2	0	0
d_5	1	2	4
d_6	0	2	2
d_7	0	5	0
q	1	2	3

Conclusions

- Model considers correlations among index terms
- Not clear in which situations it is superior to the standard Vector model
- Computation costs are higher
- Model does introduce interesting new ideas

Latent Semantic Indexing

Latent Semantic Indexing

- Classic IR might lead to poor retrieval due to:
  - unrelated documents might be included in the answer set Index Match 並不一定相關
  - relevant documents that do not contain at least one index term are not retrieved Index不match不一定不相關
  - **Reasoning**: retrieval based on index terms is vague and noisy
- The user information need is more related to concepts and ideas than to index terms
- A document that shares concepts with another document known to be relevant might be of interest

Latent Semantic Indexing

- The idea here is to map documents and queries into a dimensional space composed of concepts
- Let
  - t: total number of index terms
  - N: number of documents
  - $\mathbf{M} = [m_{ij}]$: term-document matrix $t \times N$
- To each element of $\mathbf{M}$ is assigned a weight $w_{i,j}$ associated with the term-document pair $[k_i, d_j]$
  - The weight $w_{i,j}$ can be based on a *tf-idf* weighting scheme

Latent Semantic Indexing

- The matrix $\mathbf{M} = [m_{ij}]$ can be decomposed into three components using singular value decomposition

$$\mathbf{M} = \mathbf{K} \cdot \mathbf{S} \cdot \mathbf{D}^T$$

- were
  - $\mathbf{K}$ is the matrix of eigenvectors derived from $\mathbf{C} \overset{\text{term matrix}}{=} \mathbf{M} \cdot \mathbf{M}^T$
  - $\mathbf{D}^T$ is the matrix of eigenvectors derived from $\mathbf{M}^T \cdot \mathbf{M}$ $\quad \text{document matrix}$
  - $\mathbf{S}$ is an $r \times r$ diagonal matrix of singular values where $r = \min(t, N)$ is the rank of $\mathbf{M}$

Computing an Example

- Let $\mathbf{M}^T = [m_{ij}]$ be given by

	K_1	K_2	K_3	$q \bullet d_j$
d_1	2	0	1	5
d_2	1	0	0	1
d_3	0	1	3	11
d_4	2	0	0	2
d_5	1	2	4	17
d_6	1	2	0	5
d_7	0	5	0	10
q	1	2	3	

- Compute the matrices $\mathbf{K}$, $\mathbf{S}$, and $\mathbf{D}^t$

Latent Semantic Indexing

- In the matrix $\mathbf{S}$, consider that only the s largest singular values are selected
- Keep the corresponding columns in $\mathbf{K}$ and $\mathbf{D}^T$
- The resultant matrix is called $\mathbf{M}_s$ and is given by

$$\mathbf{M}_s = \mathbf{K}_s \cdot \mathbf{S}_s \cdot \mathbf{D}_s^T$$

- where s, $s < r$, is the dimensionality of a reduced concept space
- The parameter s should be
  - large enough to allow fitting the characteristics of the data
  - small enough to filter out the non-relevant representational details

Latent Ranking

- The relationship between any two documents in s can be obtained from the $\mathrm{M}_s^T \cdot \mathrm{M}_s$ matrix given by

$$
\begin{aligned}
\mathrm{M}_s^T \cdot \mathrm{M}_s &= (\mathbf{K}_s \cdot \mathbf{S}_s \cdot \mathbf{D}_s^T)^T \cdot \mathbf{K}_s \cdot \mathbf{S}_s \cdot \mathbf{D}_s^T \\
&= \mathbf{D}_s \cdot \mathbf{S}_s \cdot \mathbf{K}_s^T \cdot \mathbf{K}_s \cdot \mathbf{S}_s \cdot \mathbf{D}_s^T \\
&= \mathbf{D}_s \cdot \mathbf{S}_s \cdot \mathbf{S}_s \cdot \mathbf{D}_s^T \\
&= (\mathbf{D}_s \cdot \mathbf{S}_s) \cdot (\mathbf{D}_s \cdot \mathbf{S}_s)^T
\end{aligned}
$$

- In the above matrix, the (i, j) element quantifies the relationship between documents d_i and d_j

Latent Ranking

- The user query can be modelled as a pseudo-document in the original $\mathbf{M}$ matrix
- Assume the query is modelled as the document numbered 0 in the $\mathbf{M}$ matrix
- The matrix $\mathbf{M}_s^T \cdot \mathbf{M}_s$ quantifies the relationship between any two documents in the reduced concept space
- The first row of this matrix provides the rank of all the documents with regard to the user query

Information Retrieval using a Singular Value Decomposition Model

George W. Furnas (Bellcore)
Scott Deerwester (Universiry of Chicago)
Susan T. Dumais (Bellcore) Thomas K. Landauer (Bellcore)
Richard A. Xarshman (University of Western Ontario)
Lynn A. Streeter (Bellcore) Karen E. Lochbaum (Bellcore)

SIGIR1988

The Singular Value Decomposition (SVD) Model

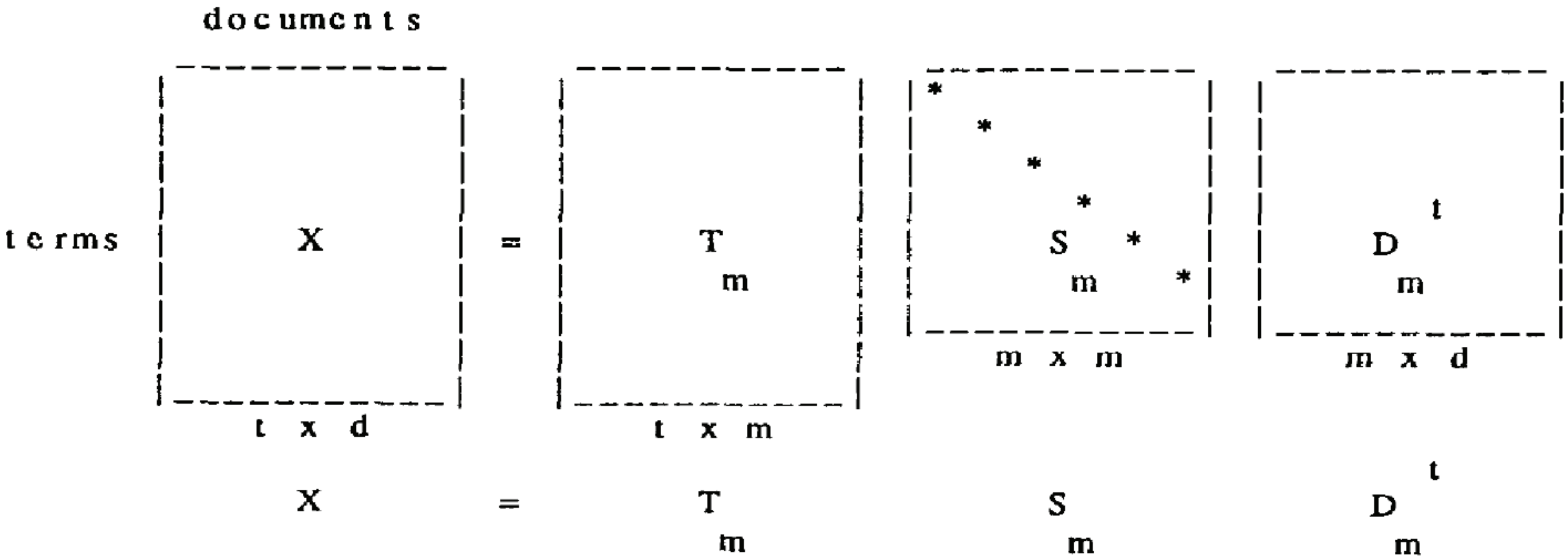


Figure 3. Singular value decomposition of the term x document matrix, **X**. Where :

$\mathbf{T}_m$ has orthogonal, unit-length columns ($\mathbf{T}_m{}^t\mathbf{T}_m = \mathbf{I}$)
$\mathbf{D}_m$ has orthogonal, unit-length columns ($\mathbf{D}_m{}^t\mathbf{D}_m = \mathbf{I}$)
$\mathbf{S}_m$ is the diagonal matrix of singular values
t is the number of rows of **X**
d is the number of columns of **X**

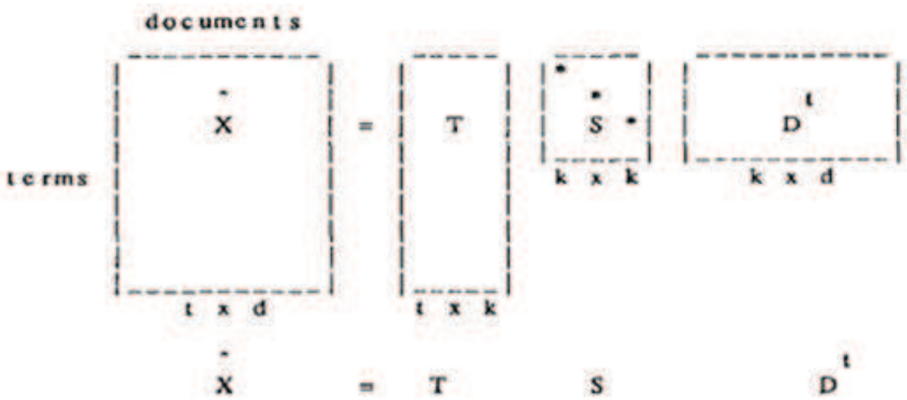
T_m is the matrix of eigenvectors of the square symmetric matrix $Y = XX^t$, and D_m is the matrix of cigenvectors of $Y = X^tX$.

Reduced Model

In general, if $\mathbf{X}=\mathbf{T}_m\mathbf{S}_m\mathbf{D}_m{}^t$ is of full rank, then the matrices $\mathbf{T}_m$, $\mathbf{D}_m$, and $\mathbf{S}_m$ must be also. However, if only the k largest singular values of $\mathbf{S}_m$ are kept along with their corresponding columns in the $\mathbf{T}_m$ and $\mathbf{D}_m$ matrices, and the rest deleted (yielding matrices $\mathbf{S}_k$, $\mathbf{T}_k$ and $\mathbf{D}_k$), the resulting matrix, $\hat{\mathbf{X}}$, is the unique matrix of rank k which is closest in the least squares sense to $\mathbf{X}$:

$$(2) \qquad \hat{\mathbf{X}} = \mathbf{T}_k\mathbf{S}_k\mathbf{D}_k{}^t \approx \mathbf{X}$$

contain only the k largest independent linear components of X
capture the major associational structure of the data and throw out much of the noise
use the value of k which maximizes the retrieval



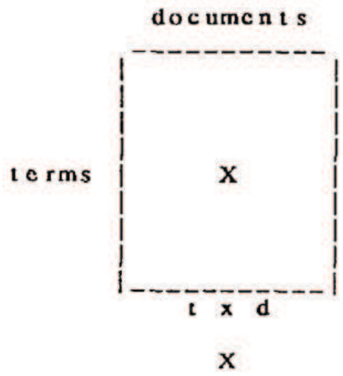
Interpretation of the row vectors of the SVD matrices, T and D:

coordinates in a k-dimensional space,
terms and documents are points in a vector space

The diagonal matrix, S, serves to stretch or shrink the orthogonal axes of this space

Term Matching Paradigm

- The simiIarity of two documents is obtained by using an cosine measure of the corresponding two column vectors of the raw data matrix X.
- A query is represented as a sort of pseudo-document, i.e., a column vector of term frequencies, $X_{\bullet q}$, which is similarly compared against columns of X, and the best matches found.
- The appropriate “cleaned up” version of the query column-vector, $X_{\bullet q}$, is given by $\sim X_{\bullet q} = TT^{t}X_{\bullet q}$.



Latent Structure Paradigm

- similarities for all pairs of documents

$$\hat{X}^t\hat{X} = (TSD^t)^t TSD^t = DST^t TSD^t = DSSD^t = (DS)(DS)^t.$$

comparison of document i and document j :
the inner product of rows i and j of the matrix DS

- similarities for all pairs of terms

$$\hat{X}\hat{X}^t = TSD^t(TSD^t)^t = TSD^t DST^t = TSST^t = (TS)(TS)^t.$$

comparison of term i and term j:
inner product of rows i and j of the matrix TS

- association between term i and document j

$$\hat{X} = TSD^t = (TS^{1/2})(DS^{1/2})^t,$$

the association between term i and document j:
the inner product of row i of the matrix, $TS^{1/2}$, and
row j of the matrix, $DS^{1/2}$

Geometric Interpretation

- If the axes of the space are resealed by the associated diagonal values of S, the inner-product between term points or document points can be used to make the algebraic comparisons of interest.
- The axes must be resealed by the associated diagonal values of $S^{1/2}$ for comparisons between a term and a document.

The Procedure

- A collection of documents has its content terms tabulated to give a frequency matrix, which is taken as X.
- A *k*-dimensional SVD decomposition of X is computed yielding matrices T, S, and D.
- The rows of T and D are taken as index vectors for corresponding terms and documents, respectively.
- The diagonal elements of S (or $S^{1/2}$, as needed), are taken as component-wise weights in ensuing similarity calculations.
- A query, treated as vector of term frequencies (albeit very sparse), is converted to a pseudo-document, $D_{q\bullet}$, in the factor space following equation.

$$\mathbf{D}_{q\bullet} = \mathbf{X}_{\bullet q}^{\mathbf{t}} \mathbf{TS}^{-1}$$

row column

- This query factor-vector is then compared to the factor-vectors of all the documents, and the documents ordered according to the results.

$X=TSD^t \rightarrow X^t=DS^tT^t$

$S^t=S$ and $T^tT=I$

$X^t=DS^tT^t \rightarrow D=X^tTS^{-1}$

$D_{q\bullet}=X_{\bullet q}{}^tTS^{-1}$

c1: *Human* machine *interface* for Lab ABC *computer* applications
c2: A *survey* of *user* opinion of *computer system response time*
c3: The *EPS user interface* management *system*
c4: *System* and *human system* engineering testing of *EPS*
c5: Relation of *user*-perceived *response time* to error measurement

m1: The generation of random, binary, unordered *trees*
m2: The intersection *graph* of paths in *trees*
m3: *Graph minors* IV: Widths of *trees* and well-quasi-ordering
m4: *Graph minors*: A *survey*

Terms	**Documents**								
	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

T_m =

0.22	-0.11	0.29	-0.41	-0.11	-0.34	0.52	-0.06	-0.41
0.20	-0.07	0.14	-0.55	0.28	0.50	-0.07	-0.01	-0.11
0.24	0.04	-0.16	-0.59	-0.11	-0.25	-0.30	0.06	0.49
0.40	0.06	-0.34	0.10	0.33	0.38	0.00	0.00	0.01
0.64	-0.17	0.36	0.33	-0.16	-0.21	-0.17	0.03	0.27
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.30	-0.14	0.33	0.19	0.11	0.27	0.03	-0.02	-0.17
0.21	0.27	-0.18	-0.03	-0.54	0.08	-0.47	-0.04	-0.58
0.01	0.49	0.23	0.03	0.59	-0.39	-0.29	0.25	-0.23
0.04	0.62	0.22	0.00	-0.07	0.11	0.16	-0.68	0.23
0.03	0.45	0.14	-0.01	-0.30	0.28	0.34	0.68	0.18

S_m =

3.34								
	2.54							
		2.35						
			1.64					
				1.50				
					1.31			
						0.85		
							0.56	
								0.36

D_m =

0.20	-0.06	0.11	-0.95	0.05	-0.08	0.18	-0.01	-0.06
0.61	0.17	-0.50	-0.03	-0.21	-0.26	-0.43	0.05	0.24
0.46	-0.13	0.21	0.04	0.38	0.72	-0.24	0.01	0.02
0.54	-0.23	0.57	0.27	-0.21	-0.37	0.26	-0.02	-0.08
0.28	0.11	-0.51	0.15	0.33	0.03	0.67	-0.06	-0.26
0.00	0.19	0.10	0.02	0.39	-0.30	-0.34	0.45	-0.62
0.01	0.44	0.19	0.02	0.35	-0.21	-0.15	-0.76	0.02
0.02	0.62	0.25	0.01	0.15	0.00	0.25	0.45	0.52
0.08	0.53	0.08	-0.03	-0.60	0.36	0.04	-0.07	-0.45

$\hat{X} =$

T

0.22	-0.11
0.20	-0.07
0.24	0.04
0.40	0.06
0.64	-0.17
0.27	0.11
0.27	0.11
0.30	-0.14
0.21	0.27
0.01	0.49
0.04	0.62
0.03	0.45

S

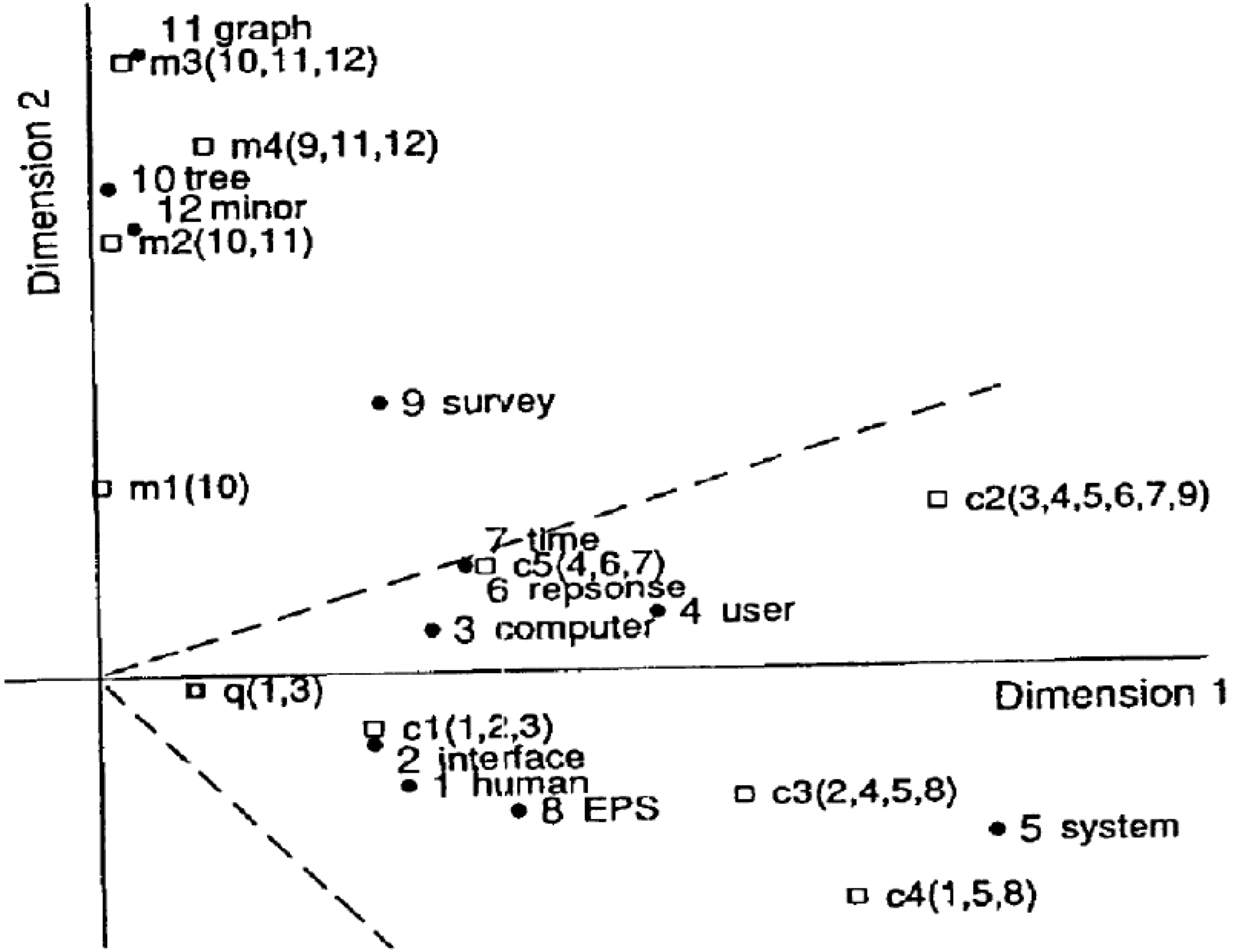
3.34	
	2.54

D′

0.20	0.61	0.46	0.54	0.28	0.00	0.02	0.02	0.08
-0.06	0.17	-0.13	-0.23	0.11	0.19	0.44	0.62	0.53

$\hat{X} =$

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	0.16	0.40	0.38	0.47	0.18	-0.05	-0.12	-0.16	-0.09
interface	0.14	0.37	0.33	0.40	0.16	-0.03	-0.07	-0.10	-0.04
computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
user	0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
system	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
survey	0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
trees	-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
graph	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
minors	-0.04	0.25	-0.10	-0.21	0.15	0.22	0.50	0.71	0.62



Dimension 2
Dimension 1
11 graph
m3(10,11,12)
m4(9,11,12)
10 tree
12 minor
m2(10,11)
9 survey
m1(10)
c2(3,4,5,6,7,9)
7 time
c5(4,6,7)
6 repsonse
4 user
3 computer
q(1,3)
c1(1,2,3)
2 interface
1 human
8 EPS
c3(2,4,5,8)
5 system
c4(1,5,8)

Conclusions

- Latent semantic indexing provides an interesting conceptualization of the IR problem
- Thus, it has its value as a new theoretical framework
- From a practical point of view, the latent semantic indexing model has not yielded encouraging results