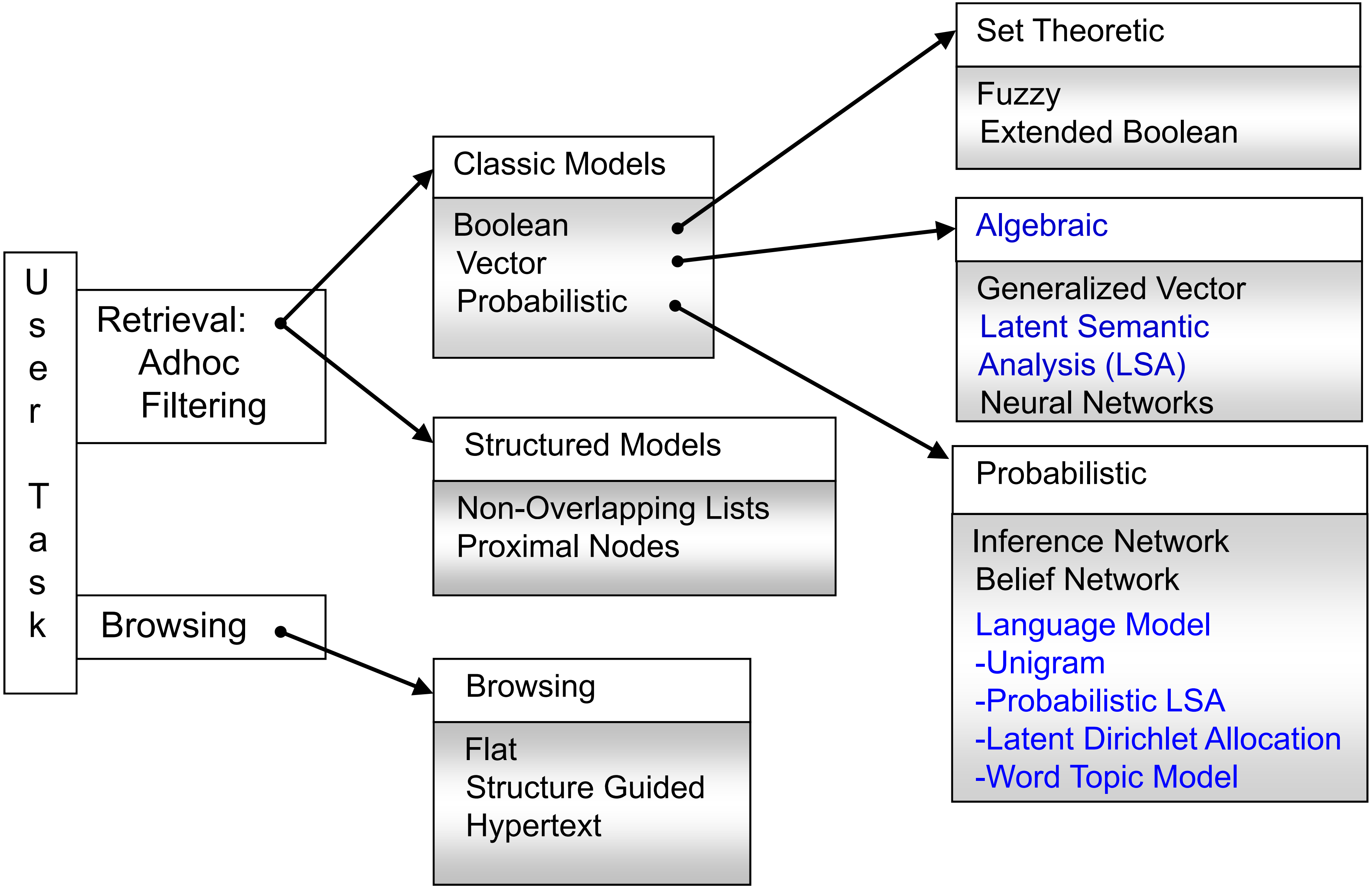
Latent Semantic Approaches for Information Retrieval and Language Modeling

Berlin Chen
Department of Computer Science & Information Engineering
National Taiwan Normal University

References

- G.W.Furnas, S. Deerwester, S.T. Dumais, T.K. Landauer, R. Harshman, L.A. Streeter, K.E. Lochbaum, “*Information Retrieval using a Singular Value Decomposition Model of Latent Semantic Structure*,” ACM SIGIR Conference on R&D in Information Retrieval , 1988
- J.R. Bellegarda, ”*Latent semantic mapping*,” IEEE Signal Processing Magazine, September 2005
- J.R. Bellegarda. *Latent Semantic Mapping: Principles and Applications.* Morgan and Claypool, 2007
- C.X. Zhai, "Statistical Language Models for Information Retrieval (Synthesis Lectures Series on Human Language Technologies)," Morgan & Claypool Publishers, 2008
- T. Hofmann, “*Unsupervised learning by probabilistic latent semantic analysis*,” Machine Learning 42, 2001
- M. Steyvers, T. Griffiths, "*Probabilistic topic models*," In T. K. Landauer, D. S. McNamara, S. Dennis, W. Kintsch (eds.). Handbook of Latent Semantic Analysis, Mahwah NJ: Lawrence Erlbaum, 2007
- B. Chen, “*Word topic models for spoken document retrieval and transcription*,” ACM Transactions on Asian Language Information Processing 8(1), pp. 2:1-2:27 2009
- B. Chen, "*Latent topic modeling of word co-occurrence information for spoken document retrieval*," ICASSP 2009
- H.-S. Chiu, B. Chen, “*Word topical mixture models for dynamic language model adaptation*,” ICASSP 2007
- D.M. Blei, A.Y.Ng, M. I. Jordan, “*Latent Dirichlet allocation,”* Journal of Machine Learning Research, 2003
- W. Kim, S. Khudanpur, “*Lexical triggers and latent semantic analysis for cross-lingual language model adaptation*,” ACM Transactions on Asian Language Information Processing 3(2), 2004
- D. Gildea, T. Hofmann, “Topic-based language models using EM,” Eurospeech1999
- L. K. Saul and F. C. N. Pereira, “*Aggregate and mixed-order Markov models for statistical language processing*,” EMNLP1997

Taxonomy of Classic IR Models

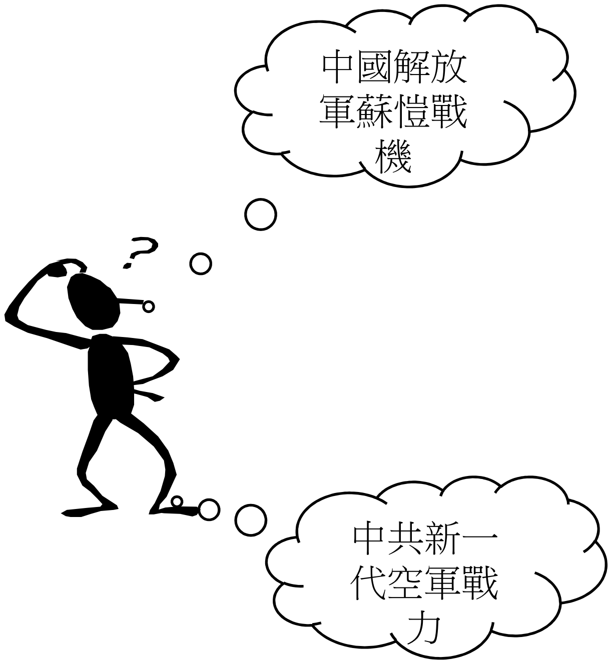


Classification of IR Models Along Two Axes

- Matching Strategy
  - Literal term matching
    - E.g., Vector Space Model (VSM), Hidden Markov Model (HMM), Language Model (LM)
  - Concept matching
    - E.g., Latent Semantic Analysis (LSA), Probabilistic Latent Semantic Analysis (PLSA), Word Topic Model (WTM)
- Learning Capability
  - Heuristic approaches for term weighting, query expansion, document expansion, etc.
    - E.g., Vector Space Model, Latent Semantic Analysis
    - Most approaches are based on linear algebra operations
  - Solid statistical foundations (optimization algorithms)
    - E.g., Unigram or Hidden Markov Model (HMM), Probabilistic Latent Semantic Analysis, Latent Dirichlet Allocation (LDA), Word Topic Model (WTM)
    - Most models belong to the language modeling approach

Two Perspectives for IR Models (cont.)

- Literal Term Matching vs. Concept Matching



香港星島日報篇報導引述軍事觀察家的話表示，到二零零五年台灣將完全喪失空中優勢，原因是中國大陸戰機不論是數量或是性能上都將超越台灣，報導指出中國在大量引進俄羅斯先進武器的同時也得加快研發自製武器系統，目前西安飛機製造廠任職的改進型飛豹戰機即將部署尚未與蘇愷三十通道地對地攻擊住宅飛機，以督促遇到挫折的監控其戰機目前也已經取得了重大階段性的認知成果。根據日本媒體報導在台海戰爭隨時可能爆發情況之下北京方面的基本方針，使用高科技答應局部戰爭。因此，解放軍打算在二零零四年前又有包括蘇愷三十二期在內的兩百架蘇霍伊戰鬥機。

  - There are usually many ways to express a given concept (an information need), so literal terms in a user's query may not match those of a relevant document

Latent Semantic Analysis (LSA)

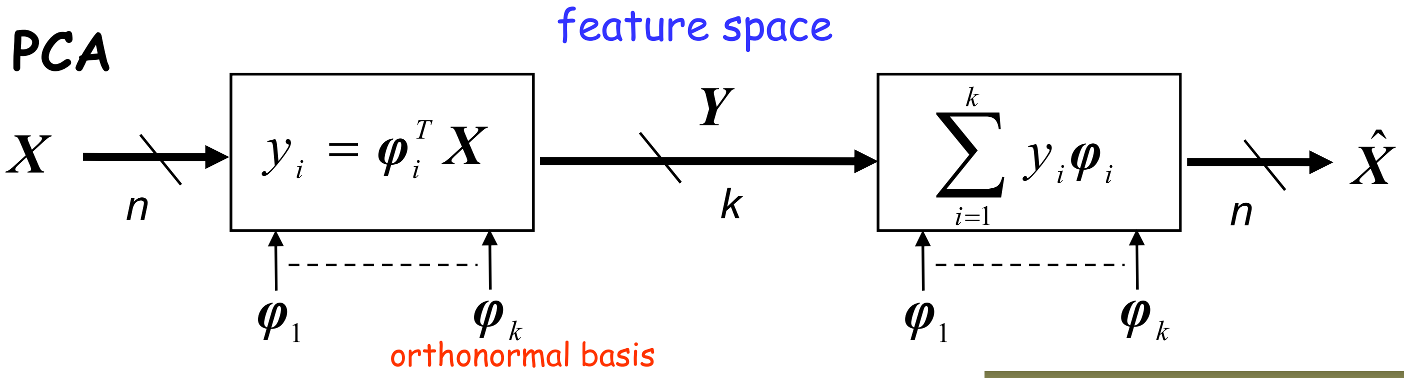
- Also called Latent Semantic Indexing (LSI), Latent Semantic Mapping (LSM), or Two-Mode Factor Analysis
  - Original formulated in the context of information retrieval
    - Users tend to retrieve documents on the basis of conceptual content
    - Individual terms (units) provide unreliable evidence about the conceptual topic or meaning of a document (composition)
    - There are many ways to express a given concept
  - LSA attempts to explore some underlying latent semantic structure in the data (documents) which is partially obscured by the randomness of word choices
  - LSA results in a parsimonious description of terms and documents
    - Contextual or positional information for words in documents is discarded (the so-called bag-of-words assumption)

Applications of LSA

- Information Retrieval
- Word/document/Topic Clustering
- Language Modeling
- Automatic Call Routing
- Language Identification
- Pronunciation Modeling
- Speaker Verification (Prosody Analysis)
- Utterance Verification
- Text/Speech Summarization
- Automatic Image Annotation
- ....

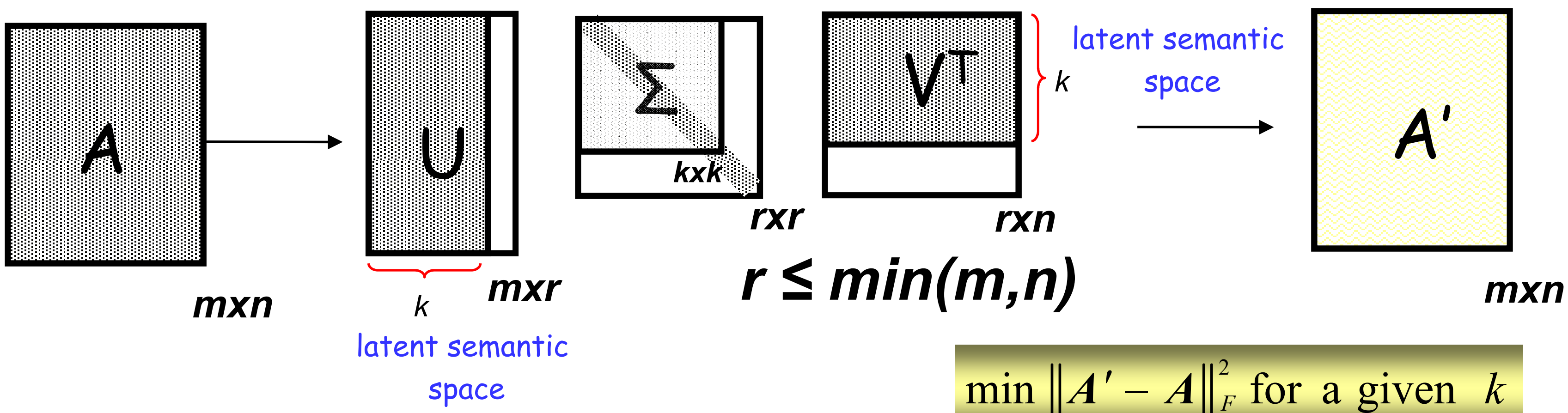
LSA : Schematic Depiction

- Dimension Reduction and Feature Extraction
  - PCA



$$\min \left\| \hat{X} - X \right\|^2 \quad \text{for a given } k$$

  - SVD (in LSA)



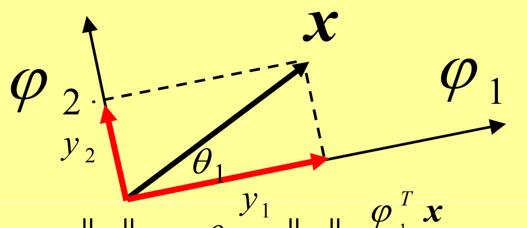
$$\min \left\| A' - A \right\|_F^2 \quad \text{for a given } k$$

LSA: An Example

– Singular Value Decomposition (SVD) used for the word-document matrix
  - *A least-squares method for dimension reduction*

	Term 1	Term 2	Term 3	Term 4
Query	user	interface		
Document 1	user	interface	HCI	interaction
Document 2			HCI	interaction

Projection of a Vector $\boldsymbol{x}$:

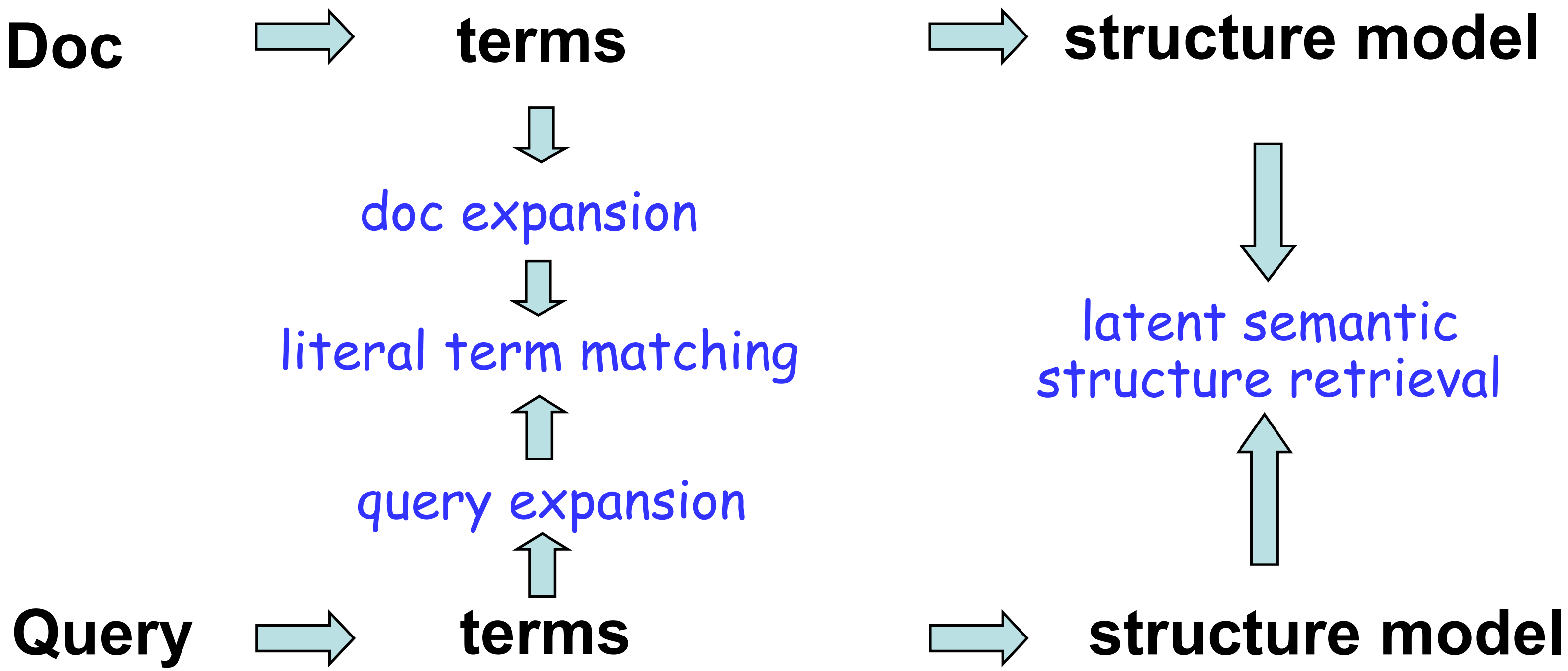


$$y_1 = \|\boldsymbol{x}\| \cos \theta_1 = \|\boldsymbol{x}\| \frac{\varphi_1^T \boldsymbol{x}}{\|\boldsymbol{x}\| \|\boldsymbol{\varphi}_1\|} = \varphi_1^T \boldsymbol{x}$$

$$\text{, where } \|\boldsymbol{\varphi}_1\| = 1$$

LSA: Latent Structure Space

- Two alternative frameworks to circumvent vocabulary mismatch



LSA: Another Example (1/2)

Titles

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
1.	*human*	1	0	0	1	0	0	0	0	0
2.	*interface*	1	0	1	0	0	0	0	0	0
3.	*computer*	1	1	0	0	0	0	0	0	0
4.	*user*	0	1	1	0	1	0	0	0	0
5.	*system*	0	1	1	2	0	0	0	0	0
6.	*response*	0	1	0	0	1	0	0	0	0
7.	*time*	0	1	0	0	1	0	0	0	0
8.	*EPS*	0	0	1	1	0	0	0	0	0
9.	*survey*	0	1	0	0	0	0	0	0	1
10.	*trees*	0	0	0	0	0	1	1	1	0
11.	*graph*	0	0	0	0	0	0	1	1	1
12.	*minors*	0	0	0	0	0	0	0	1	1

LSA: Another Example (2/2)

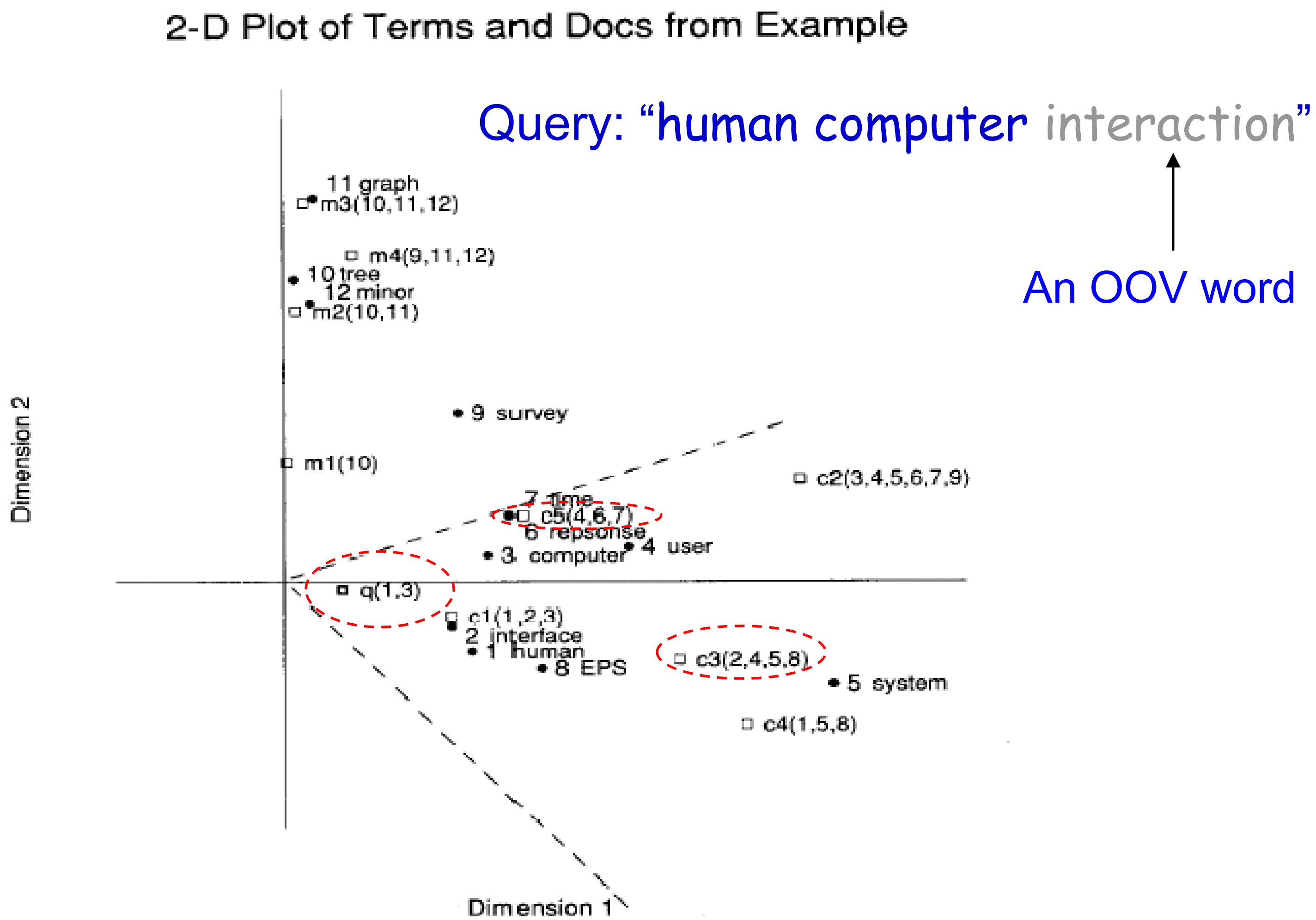


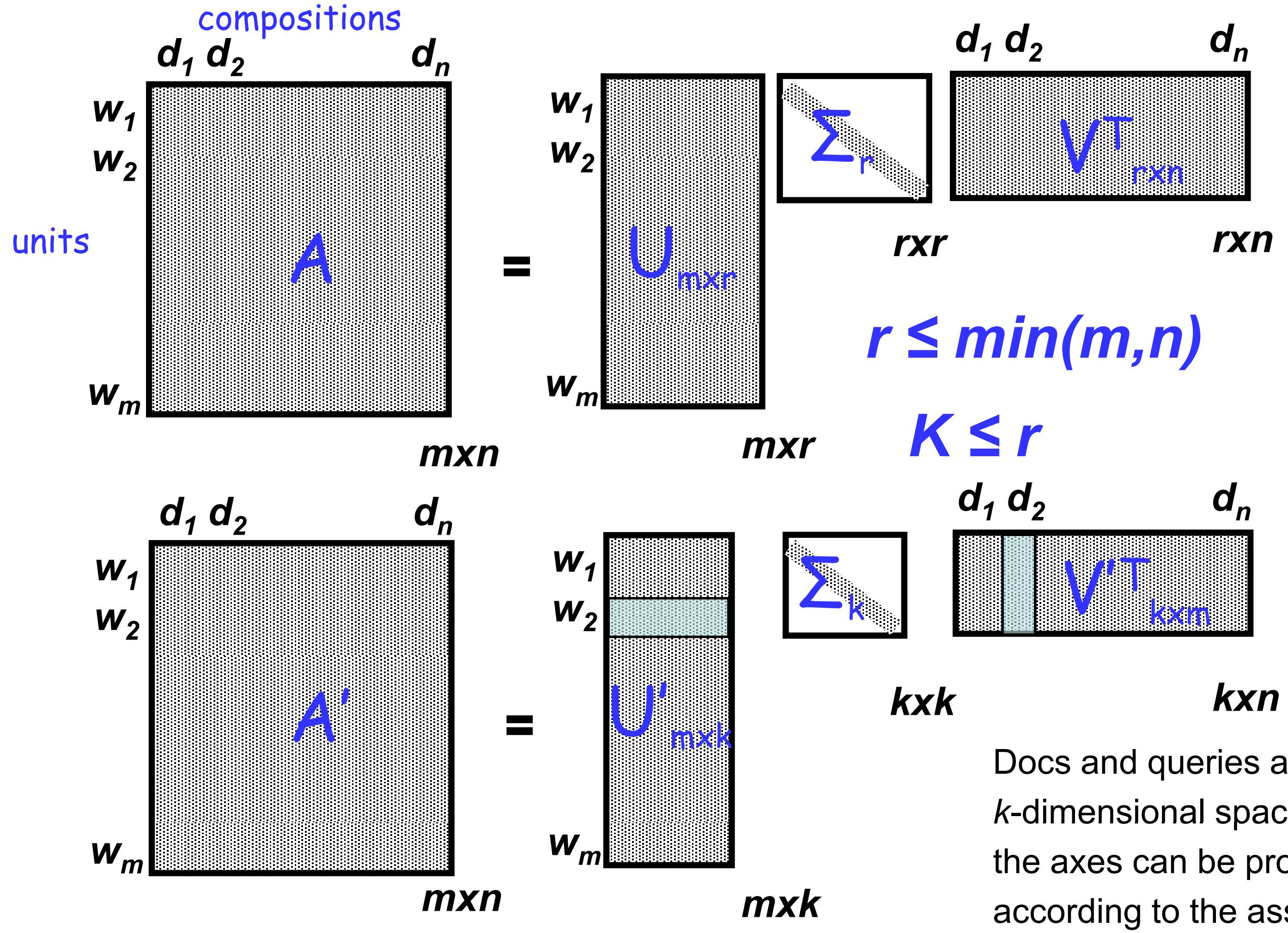
FIG. 1. A two-dimensional plot of 12 Terms and 9 Documents from the sampe TM set. Terms are represented by filled circles. Documents are shown as open squares, and component terms are indicated parenthetically. The query ("human computer interaction") is represented as a pseudo-document at point q. Axes are scaled for Document-Document or Term-Term comparisons. The dotted cone represents the region whose points are within a cosine of .9 from the query q. All documents about human-computer (c1–c5) are "near" the query (i.e., within this cone), but none of the graph theory documents (m1–m4) are nearby. In this reduced space, even documents c3 and c5 which share no terms with the query are near it.

LSA: Theoretical Foundation

- Singular Value Decomposition (SVD)

Row A $\in R^n$

Col A $\in R^m$



Both U and V has orthonormal column vectors

$U^TU=I_{rXr}$

$V^TV=I_{rXr}$

$r \le min(m,n)$

$K \le r$

$||A||_F^2 \ge ||A'||_F^2$

$$\|A\|_F^2 = \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2$$

Docs and queries are represented in a *k*-dimensional space. The quantities of the axes can be properly weighted according to the associated diagonal values of Σ_k

LSA: Theoretical Foundation

- "term-document" matrix *A* has to do with the co-occurrences between terms (units) and documents (compositions)
  - Contextual or positional information for words in documents is discarded
    - "bag-of-words" modeling

- **Feature extraction** for the entities $a_{i,j}$ of matrix *A*

  1. Conventional *tf-idf* statistics

  2. Or, $a_{i,j}$:occurrence frequency weighted by negative entropy

occurrence count of term *i* in document *j*

$$a_{i,j} = \frac{f_{i,j}}{\left|d_j\right|} \times (1 - \varepsilon_i), \quad \left|d_j\right| = \sum_{i=1}^{m} f_{i,j}$$

negative normalized entropy

document length

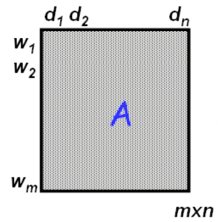
normalized entropy of term *i*

occurrence count of term *i* in the collection

$$\varepsilon_i = -\frac{1}{\log n} \sum_{j=1}^{n} \left(\frac{f_{i,j}}{\tau_i} \log \frac{f_{i,j}}{\tau_i} \right), \quad \tau_i = \sum_{j=1}^{n} f_{i,j}$$

$0 \le \varepsilon_i \le 1$

LSA: Theoretical Foundation



- Singular Value Decomposition (SVD)
  - $A^T A$ is symmetric $n \times n$ matrix
    - All eigenvalues λ_j are nonnegative real numbers

$$\lambda_1 \geq \lambda_2 \geq .... \geq \lambda_n \geq 0 \qquad \Sigma^2 = diag(\lambda_1, \lambda_1, \ldots, \lambda_n)$$

    - All eigenvectors v_j are orthonormal ($\in R^n$)

$$V = [v_1 v_2 \ldots v_n] \qquad v_j^T v_j = 1 \qquad (V^T V = I_{nxn})$$

    - Define **singular values:** sigma $\sigma_j = \sqrt{\lambda_j}, \; j = 1, \ldots, n$
      - As the square roots of the eigenvalues of $A^T A$
      - As the lengths of the vectors $Av_1, Av_2, \ldots, Av_n$

For $\lambda_i \neq 0$, $i=1,\ldots r$,
$\{Av_1, Av_2, \ldots, Av_r\}$ is an orthogonal basis of Col A

$$\sigma_1 = \|Av_1\|$$
$$\sigma_2 = \|Av_2\|$$
.....

$$\|Av_i\|^2 = v_i^T A^T A v_i = v_i^T \lambda_i v_i = \lambda_i$$
$$\Rightarrow \|Av_i\| = \sigma_i$$

LSA: Theoretical Foundation

- $\{Av_1, Av_2, \ldots, Av_r\}$ is an orthogonal basis of Col A $(\in R^m)$

$$Av_i \bullet Av_j = (Av_i)^T Av_j = v_i^T A^T Av_j = \lambda_j v_i^T v_j = 0$$

  - Suppose that A (or A^TA) has rank $r \leq n$

$$\lambda_1 \geq \lambda_2 \geq \ldots. \geq \lambda_r > 0, \quad \lambda_{r+1} = \lambda_{r+2} = \ldots. = \lambda_n = 0$$

  - Define an orthonormal basis $\{u_1, u_2, \ldots, u_r\}$ for Col A

$$u_i = \frac{1}{\|Av_i\|} Av_i = \frac{1}{\sigma_i} Av_i \Rightarrow \sigma_i u_i = Av_i$$

U is also an orthonormal matrix (mxr)

V: an orthonormal matrix

$$\Rightarrow [u_1\ u_2 \ldots u_r]\Sigma_{r\times r} = A\boxed{[v_1\ v_2 \quad v_r]}$$

Known in advance

- Extend to an orthonormal basis $\{u_1, u_2, \ldots, u_m\}$ of R^m

$$\Rightarrow [u_1\ u_2 \ldots u_r \ldots u_m]\Sigma_{m\times n} = A[v_1\ v_2 \ldots v_r \ldots v_n]$$

$$\Rightarrow U\Sigma = AV \Rightarrow U\Sigma V^T = A\underbrace{VV^T}_{I_{nxn}}\ ?$$

$$\Rightarrow A = U\Sigma V^T$$

$$\Sigma_{m\times n} = \begin{pmatrix} \Sigma_r & 0_{r\times(n-r)} \\ 0_{(m-r)\times r} & 0_{(m-r)\times(n-r)} \end{pmatrix}$$

$$\|A\|_F^2 = \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2$$

$$\|A\|_F^2 = \sigma_1^2 + \sigma_2^2 + \ldots + \sigma_r^2 \quad ?$$

LSA: Theoretical Foundation

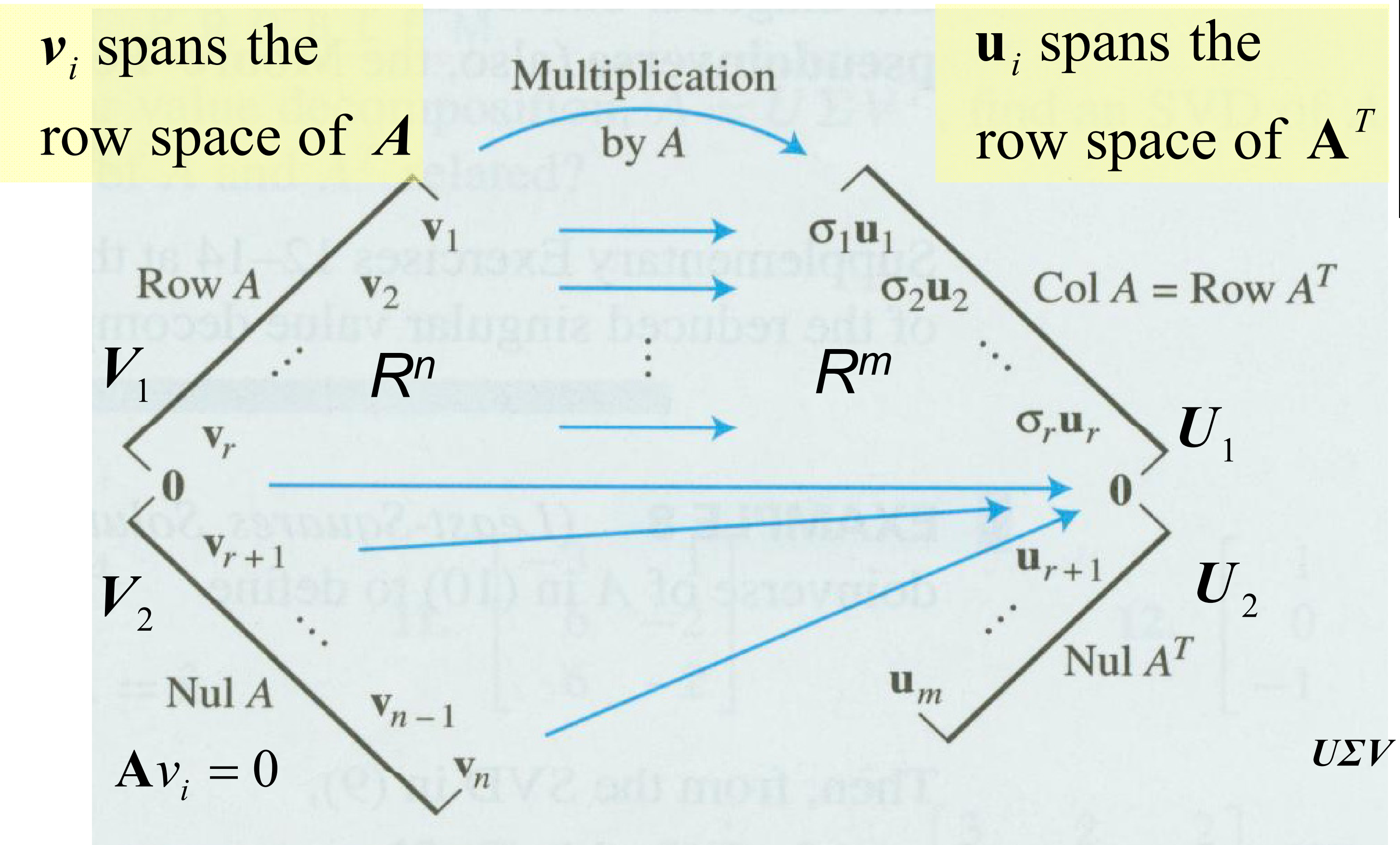
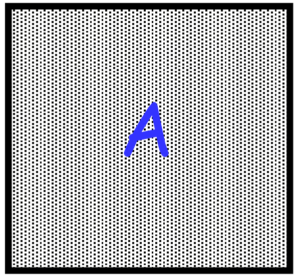
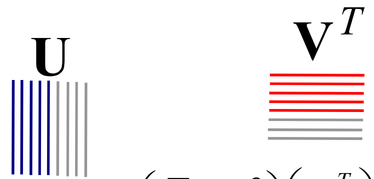


FIGURE 4 The four fundamental subspaces and the action of A.



mxn



$$U\Sigma V^T = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \begin{pmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix}$$

$$= U_1 \Sigma_1 V_1^T$$

$$= A V_1 V_1^T$$

$$= A$$

$U\Sigma = AV$

LSA: Theoretical Foundation

- Additional Explanations
  - Each row of U is related to the projection of a corresponding row of A onto the basis formed by columns of V

$$A = U\Sigma V^T$$

$$\Rightarrow AV = U\Sigma V^T V = U\Sigma \quad \Rightarrow \quad U\Sigma = AV$$

    - the *i*-th entry of a row of U is related to the projection of a corresponding row of A onto the *i*-th column of V

  - Each row of V is related to the projection of a corresponding row of A^T onto the basis formed by U

$$A = U\Sigma V^T$$

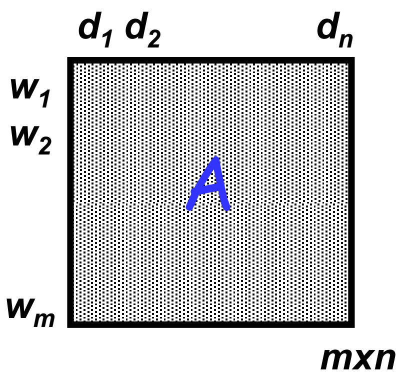
$$\Rightarrow A^T U = \left(U\Sigma V^T\right)^T U = V\Sigma U^T U = V\Sigma$$

$$\Rightarrow V\Sigma = A^T U$$

    - the *i*-th entry of a row of V is related to the projection of a corresponding row of A^T onto the *i*-th column of U

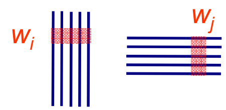
LSA: Theoretical Foundation

- Fundamental comparisons based on SVD
  - The original word-document matrix (A)



- compare two terms → dot product of two rows of A
  - or an entry in AA^T
- compare two docs → dot product of two columns of A
  - or an entry in A^TA
- compare a term and a doc → each individual entry of A

  - The new word-document matrix (A')



$U'=U_{m\times k}$
$\Sigma'=\Sigma_k$
$V'=V_{n\times k}$

- compare two terms
  → dot product of two rows of $U'\Sigma'$

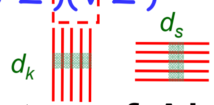
$$A'A'^T=(U'\Sigma'V'^T)(U'\Sigma'V'^T)^T=U'\Sigma'V'^TV'\Sigma'^TU'^T=(U'\Sigma')(U'\Sigma')^T$$

$I_{r\times r}$

For stretching or shrinking

- compare two docs
  → dot product of two rows of $V'\Sigma'$

$$A'^TA'=(U'\Sigma'V'^T)^T(U'\Sigma'V'^T)=V'\Sigma'^TU'^TU'\Sigma'V'^T=(V'\Sigma')(V'\Sigma')^T$$



- compare a query word and a doc → each individual entry of A'

LSA: Fold-in

- Find representations for pesudo-docs
  - For objects (new queries or docs) that did not appear in the original analysis
    - Fold-in a new mx1 query (or doc) vector

See Figure A in next page

$$\hat{q}_{1\times k} = \boxed{\left(q^T\right)_{1\times m} U_{m\times k}} \boxed{\Sigma^{-1}_{k\times k}}$$

The separate dimensions are differentially weighted

Just like a row of V

Query represented by the weighted sum of it constituent term vectors

  - Represented as the weighted sum of its component word (or term) vectors
  - Cosine measure between the query and doc vectors in the latent semantic space

$$sim\left(\hat{q}, \hat{d}\right) = coine\ (\hat{q}\Sigma, \hat{d}\Sigma) = \frac{\hat{q}\Sigma^2\hat{d}^T}{|\hat{q}\Sigma||\hat{d}\Sigma|}$$

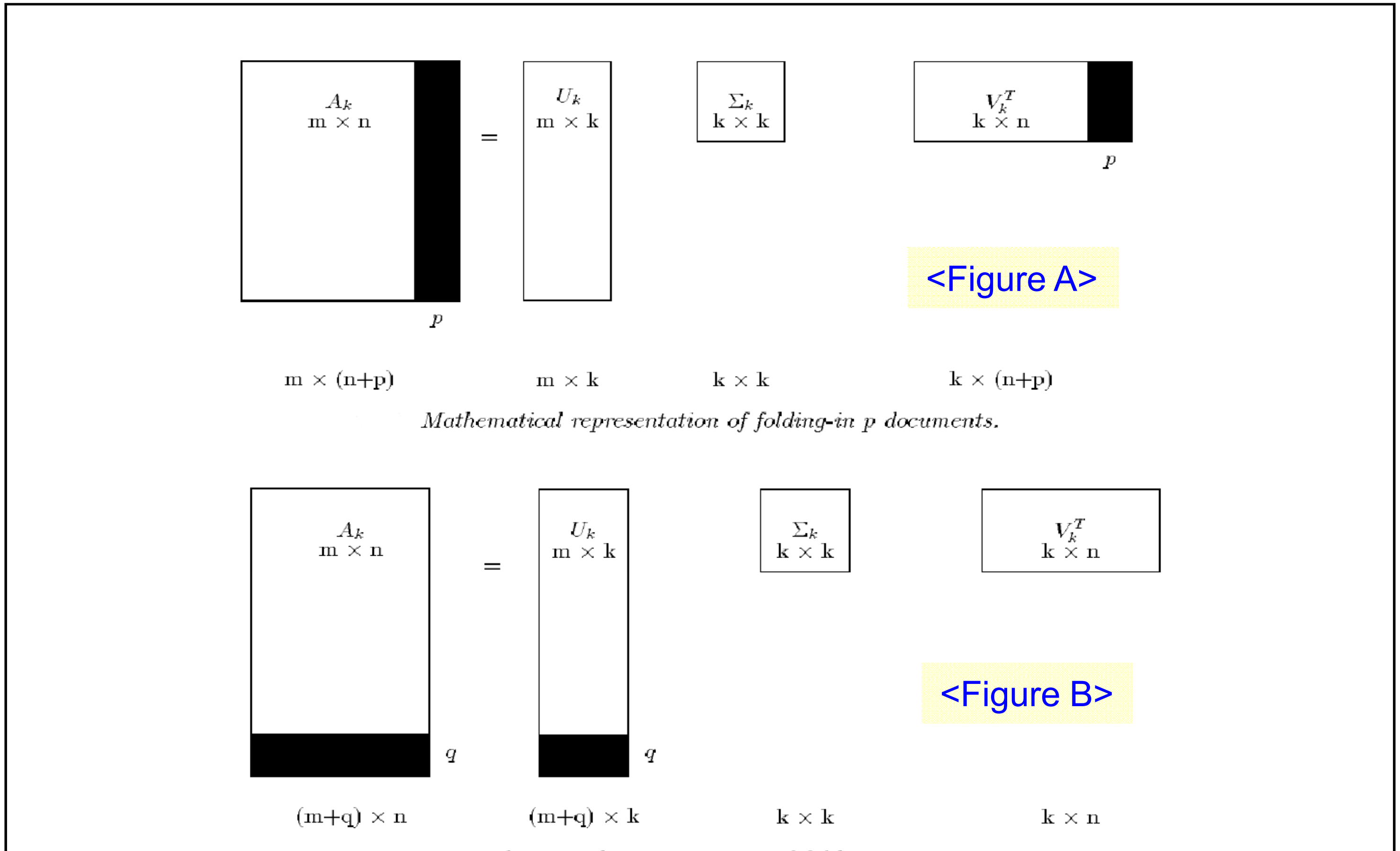
row vectors

LSA: Theoretical Foundation

- Fold-in a new 1 x n term vector

$$\hat{t}_{1 \times k} = t_{1 \times n} V_{n \times k} \Sigma^{-1}_{k \times k}$$

See Figure B below

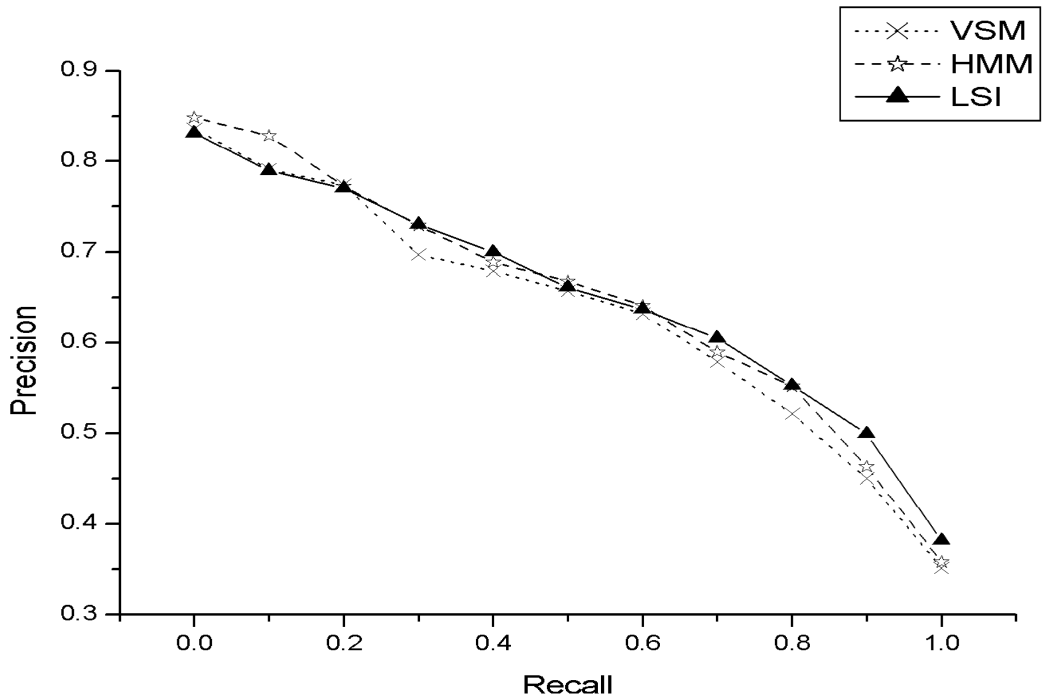


Mathematical representation of folding-in p documents.

Mathematical representation of folding-in q terms.

LSA: A Simple IR Evaluation

- Experimental results
  - HMM is consistently better than VSM at all recall levels
  - LSA is better than VSM at higher recall levels



Recall-Precision curve at 11 standard recall levels evaluated on TDT-3 SD collection. (Using word-level indexing terms)

LSA: Pro and Con (1/2)

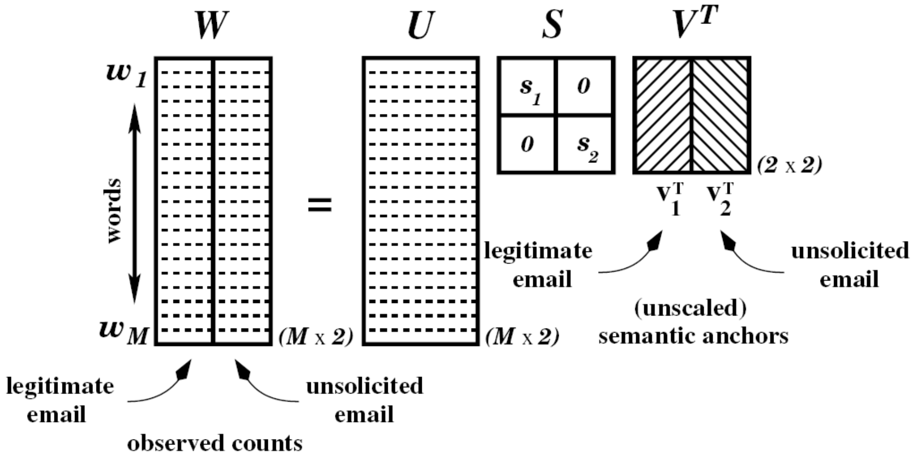
- Pro (Advantages)
  - A clean formal framework and a clearly defined optimization criterion (least-squares)
    - Conceptual simplicity and clarity
  - Handle synonymy problems (“heterogeneous vocabulary”)
    - Replace individual terms as the descriptors of documents by independent “artificial concepts” that can specified by any one of several terms (or documents) or combinations
  - Good results for high-recall search
    - Take term co-occurrence into account

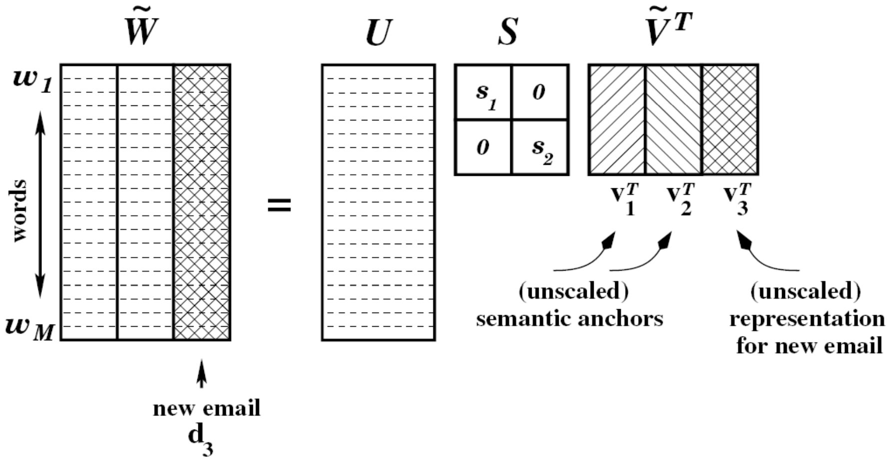
LSA: Pro and Con (2/2)

- Disadvantages
  - High computational complexity (e.g., SVD decomposition)
  - Exhaustive comparison of a query against all stored documents is needed (cannot make use of inverted files ?)
  - LSA offers only a partial solution to polysemy (e.g. bank, bass,…)
    - Every term is represented as just one point in the latent space (represented as weighted average of different meanings of a term)

LSA: Junk E-mail Filtering

- One vector represents the centriod of all e-mails that are of interest to the user, while the other the centriod of all e-mails that are not of interest





LSA: Dynamic Language Model Adaptation (1/4)

- Let w_q denote the word about to be predicted, and $H_{q\text{-}1}$ the admissible LSA history (context) for this particular word
  - The vector representation of $H_{q\text{-}1}$ is expressed by $\tilde{d}_{q-1}$
    - Which can be then projected into the latent semantic space

LSA representation

$$\tilde{\tilde{v}}_{q-1} = \tilde{v}_{q-1} S = \tilde{d}_{q-1}^{T} U \quad [\text{change of notation} : S = \Sigma]$$

    - Iteratively update $\tilde{d}_{q-1}$ and $\tilde{\tilde{v}}_{q-1}$ as the decoding evolves

VSM representation

$$\tilde{d}_q = \frac{n_q - 1}{n_q} \tilde{d}_{q-1} + \frac{1 - \varepsilon_i}{n_q} [0 \ldots 1 \ldots 0]^T$$

LSA representation

$$\tilde{\tilde{v}}_q = \tilde{v}_q S = d_{q-1}^T U = \frac{1}{n_q} \left[(n_q - 1) \tilde{\tilde{v}}_{q-1} + \underline{(1 - \varepsilon_i)} u_i \right]$$

$$\text{or} \quad = \frac{1}{n_q} \left[\boxed{\lambda} \cdot (n_q - 1) \tilde{\tilde{v}}_{q-1} + (1 - \varepsilon_i) u_i \right]$$

with exponential decay

LSA: Dynamic Language Model Adaptation (2/4)

- Integration of LSA with N-grams

$$\Pr(w_q \mid H_{q-1}^{(n+l)}) = \Pr(w_q \mid H_{q-1}^{(n)}, H_{q-1}^{(l)})$$

where H_{q-1} denotes some suitable history for word w_q,

and the superscripts $^{(n)}$ *and* $^{(l)}$ refer to the n - gram

component ($w_{q-1} w_{q-2} \ldots w_{q-n+1}$, with $n > 1$), the LSA

component ($\tilde{d}_{q-1}$):

This expression can be rewritten as:

$$\Pr(w_q \mid H_{q-1}^{(n+l)}) = \frac{\Pr(w_q, H_{q-1}^{(l)} \mid H_{q-1}^{(n)})}{\sum_{w_i \in V} \Pr(w_i, H_{q-1}^{(l)} \mid H_{q-1}^{(n)})}$$

LSA: Dynamic Language Model Adaptation (3/4)

- Integration of LSA with N-grams (cont.)

$$\Pr(w_q, H_{q-1}^{(l)} \mid H_{q-1}^{(n)}) =$$

$$\Pr(w_q \mid H_{q-1}^{(n)}) \cdot \Pr(H_{q-1}^{(l)} \mid w_q, H_{q-1}^{(n)})$$

Assume the probability of the document history given the current word is not affected by the immediate context preceding it

$$= \Pr(w_q \mid w_{q-1} w_{q-2} \cdots w_{q-n+1}) \cdot \Pr(\tilde{d}_{q-1} \mid w_q \underline{w_{q-1} w_{q-2} \cdots w_{q-n+1}})$$

$$= \Pr(w_q \mid w_{q-1} w_{q-2} \cdots w_{q-n+1}) \cdot \Pr(\tilde{d}_{q-1} \mid w_q)$$

$$= \Pr(w_q \mid w_{q-1} w_{q-2} \cdots w_{q-n+1}) \cdot \frac{\Pr(w_q \mid \tilde{d}_{q-1}) \Pr(\tilde{d}_{q-1})}{\Pr(w_q)}$$

$\Rightarrow$ $$\Pr(w_q \mid H_{q-1}^{(n+l)}) =$$

$$\frac{\Pr(w_q \mid w_{q-1} w_{q-2} \cdots w_{q-n+1}) \cdot \dfrac{\Pr(w_q \mid \tilde{d}_{q-1})}{\Pr(w_q)}}{\sum_{w_i \in V} \Pr(w_i \mid w_{q-1} w_{q-2} \cdots w_{q-n+1}) \cdot \dfrac{\Pr(w_i \mid \tilde{d}_{q-1})}{\Pr(w_i)}}$$

LSA: Dynamic Language Model Adaptation (4/4)

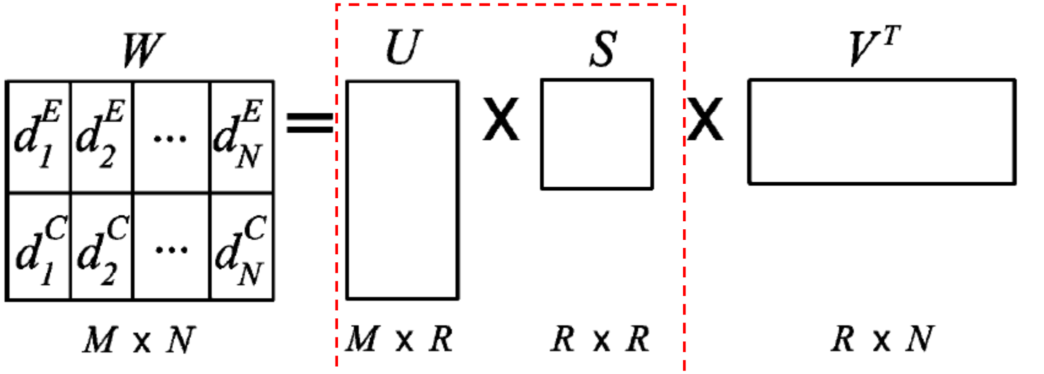
Intuitively, $\Pr(w_q \mid \tilde{d}_{q-1})$ reflects the "relevance" of word w_q to the admissible history, as observed through $\tilde{d}_{q-1}$:

$$
\begin{aligned}
&\Pr(w_q \mid \tilde{d}_{q-1}) \\
&\approx K(w_q \mid \tilde{d}_{q-1}) \\
&= \cos(u_q S^{1/2}, \tilde{v}_{q-1} S^{1/2}) = \frac{u_q S \tilde{v}_{q-1}^T}{\left\| u_q S^{1/2} \right\| \left\| \tilde{v}_{q-1} S^{1/2} \right\|}
\end{aligned}
$$

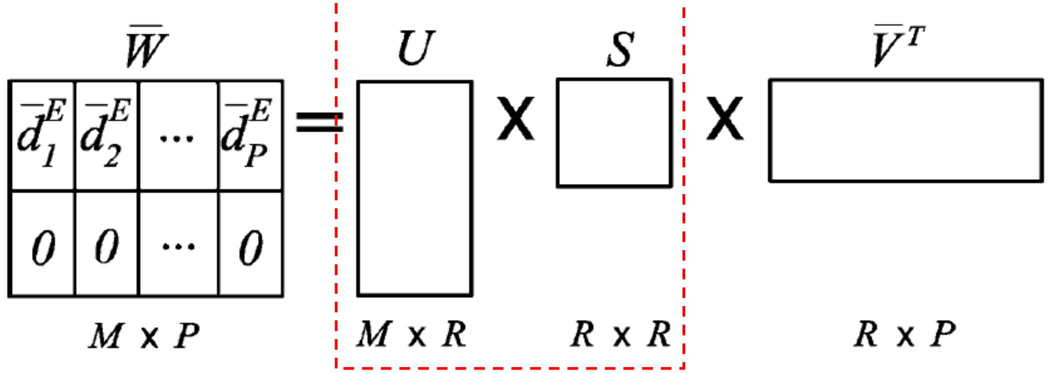
As such, it will be highest for words whose meaning aligns most closely with the semantic favric of $\tilde{d}_{q-1}$ (i.e., relevant "content" words), and lowest for words which do not convey any particular information about this fabric (e.g., "function" works like "*the*").

LSA: Cross-lingual Language Model Adaptation (1/2)

- Assume that a document-aligned (instead of sentence-aligned) Chinese-English bilingual corpus is provided



SVD of a word-document matrix for CL-LSA.



Folding-in a monolingual corpus into LSA.

Lexical triggers and latent semantic analysis for cross-lingual language model adaptation, TALIP 2004, 3(2)

LSA: Cross-lingual Language Model Adaptation (2/2)

- CL-LSA adapted Language Model

d_i^E is a relevant English doc of the Mandarin d_i^C doc being transcribed, obtained by CL-IR

$$P_{\text{Adapt}}\left(c_k \middle| c_{k-1}, c_{k-2}, d_i^E\right)$$
$$= \lambda \cdot PP_{\text{CL-LCA-Unigram}}\left(c_k \middle| d_i^E\right) + P_{\text{BG-Trigram}}\left(c_k \middle| c_{k-1}, c_{k-2}\right)$$

$$P_{\text{CL-LCA-Unigram}}\left(c \middle| d_i^E\right) = \sum_e P_T(c|e) P\left(e \middle| d_i^E\right)$$

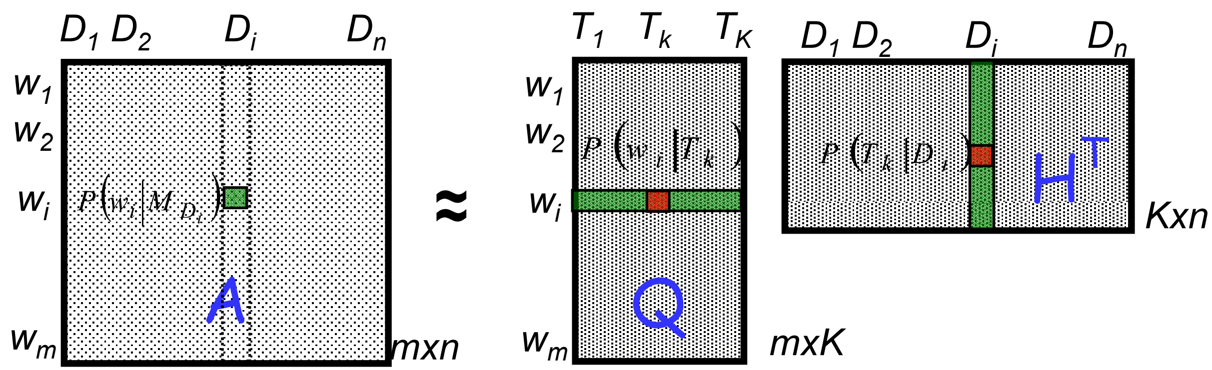
$$P_T(c|e) \approx \frac{\text{sim}(\vec{c}, \vec{e})^\gamma}{\sum_{c'} \text{sim}(\vec{c}', \vec{e})^\gamma} \qquad (\gamma >> 1)$$

Probabilistic Latent Semantic Analysis (PLSA)

- PLSA models the co-occurrence of word and documents and evaluates the relevance in a low dimensional semantic/topic space
  - Each document D is treated as a document model M_D

$$P_{\mathrm{PLSA}}(w_i \mid M_D) = \sum_{k=1}^{K} P(w_i \mid T_k) P(T_k \mid M_D)$$

- PLSA can be viewed as a nonnegative factorization of a "word-document" matrix consisting probability entries
  - A procedure similar to the SVD performed by its algebraic counterpart- LSA

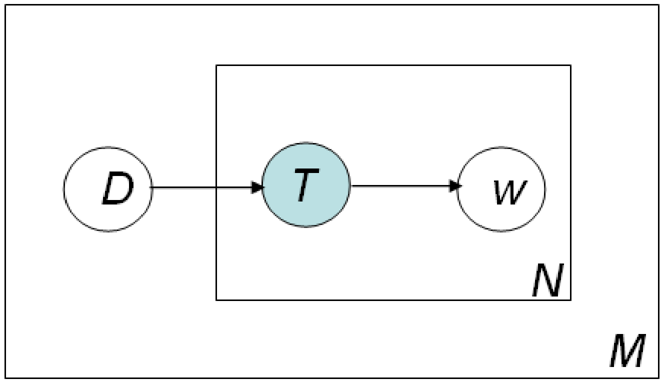


PLSA: Information Retrieval (1/3)

- The relevance measure between a query and a document can be expressed by

$$P_{\text{PLSA}}\left(Q \middle| M_D\right) = \prod_{w_i \in Q} \left[\sum_{k=1}^{K} P\left(w_i \middle| T_k\right) P\left(T_k \middle| M_D\right) \right]^{c(w_i, Q)}$$

  - Relevance measure is not obtained based on the frequency of a respective query term occurring in a document, but instead based on the frequency of the term and document in the latent topics
  - A query and a document thus may have a high relevance score even if they do not share any terms in common



N: number of distinct in the vocabulary

M: number of documents in the collection

○ : observed variable

● : latent variable

PLSA: Information Retrieval (2/3)

- Unsupervised training: The model parameters are trained beforehand using a set of text documents
  - Maximize the log-likelihood of entire collection $\mathbf{D}$

$$log\, L_{\mathbf{D}} = \sum_{D \in \mathbf{D}} log\, P_{PLSA}(D \mid M_D) = \sum_{D \in \mathbf{D}} \sum_{w_n \in D} c(w_i, D)\, log\, P_{PLSA}(w_i \mid M_D)$$

- Supervised training: The model parameters are trained using a training set of query exemplars and the associated query-document relevance information
  - Maximize the log-likelihood of the training set of query exemplars generated by their relevant documents

$$log\, L_{\mathbf{Q}_{TrainSet}} = \sum_{Q \in \mathbf{Q}_{TrainSet}} \sum_{D \in \mathbf{D}_{R \text{ to } Q}} log\, P_{PLSA}(Q \mid M_D)$$

$$= \sum_{Q \in \mathbf{Q}_{TrainSet}} \sum_{D \in \mathbf{D}_{R \text{ to } Q}} \sum_{w_i \in Q} c(w_i, Q)\, log\, P(w_i \mid M_D)$$

PLSA: Information Retrieval (3/3)

- Example: most probable words form 4 latent topics

aviation	space missions	family love	Hollywood love
Aspect 1	Aspect 2	Aspect 3	Aspect 4
plane	space	home	film
airport	shuttle	family	movie
crash	mission	like	music
flight	astronauts	love	new
safety	launch	kids	best
aircraft	station	mother	hollywood
air	crew	life	love
passenger	nasa	happy	actor
board	satellite	friends	entertainment
airline	earth	cnn	star

The 2 aspects to most likely generate the word 'flight' (left) and 'love' (right), derived from a $K = 128$ aspect model of the TDT1 document collection. The displayed terms are the most probable words in the class-conditional distribution $P(w_j | z_k)$, from top to bottom in descending order.

PLSA vs. LSA

- Decomposition/Approximation
  - **LSA**: least-squares criterion measured on the L2- or Frobenius norms of the word-doc matrices
  - **PLSA**: maximization of the likelihoods functions based on the cross entropy or Kullback-Leibler divergence between the empirical distribution and the model
- Computational complexity
  - LSA: SVD decomposition
  - PLSA: EM training, is time-consuming for iterations ?
  - The model complexity of both LSA and PLSA grows linearly with the number of training documents
    - There is no general way to estimate or predict the vector representation (of LSA) or the model parameters (of PLSA) for a newly observed document
- LSA and PLSA both assume "bag-of-words" representations of documents (how to distinguish "street market" from market street ?)

PLSA: Dynamic Language Model Adaptation

- The search history can be treated as a pseudo-document which is varying during the speech recognition process

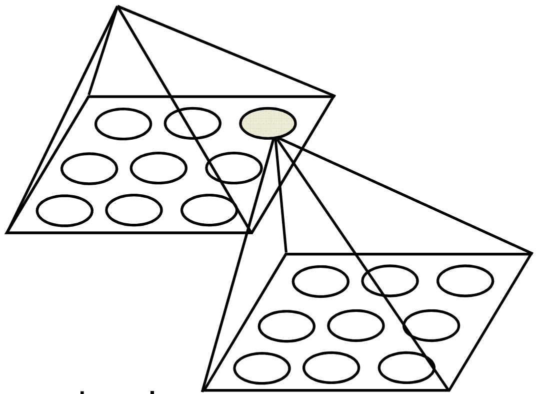
$$P_{\text{PLSA}}\left(w_i \mid H_{w_i}\right) = \sum_{k=1}^{K} P\left(w_i \mid T_k\right) P\left(T_k \mid H_{w_i}\right)$$

  - The topic unigrams $P\left(w_i \mid T_k\right)$ are kept unchanged
  - The history's probability distribution over the latent topics is gradually updated
  - The topic mixture weights $P\left(T_k \mid H_{w_i}\right)$ are estimated on the fly
    - It would be time-consuming

PLSA: Document Organization (1/3)

- Each document is viewed as a document model to generate itself
  - Additional transitions between topical mixtures have to do with the topological relationships between topical classes on a 2-D map

$$P_{\text{PLSA}}\left(w_i \middle| M_D\right) = \sum_{k=1}^{K} P\left(T_k \middle| \mathrm{M}_D\right)\left[\sum_{l=1}^{K} P\left(T_l \middle| T_k\right) P\left(w_i \middle| T_l\right)\right]$$



Two-dimensional Tree Structure for Organized Topics

$$E\left(T_l, T_k\right) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{dist\left(T_k, T_l\right)^2}{2\sigma^2}\right]$$

$$P\left(T_l \middle| T_k\right) = \frac{E\left(T_l, T_k\right)}{\sum_{s=1}^{K} E\left(T_s, T_k\right)}$$

PLSA: Document Organization (2/3)

- Document models can be trained in an unsupervised way by maximizing the total log-likelihood of the document collection

$$L_T = \sum_{j=1}^{n} \sum_{i=1}^{V} c\left(w_i, D_j\right) \log P\left(w_i \middle| D_j\right)$$

- Each topical class can be labeled by words selected using the following criterion

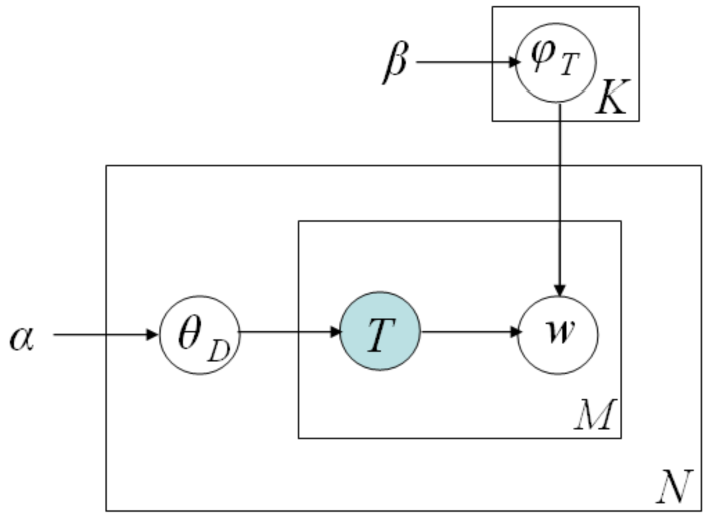
$$Sig\left(w_i, T_k\right) = \frac{\sum_{j=1}^{n} c\left(w_i, D_j\right) P\left(T_k \middle| D_j\right)}{\sum_{i=1}^{n} c\left(w_i, D_j\right)\left[1 - P\left(T_k \middle| D_j\right)\right]}$$

PLSA: Document Organization (3/3)

- Spoken Document Retrieval and Browsing System developed by NTU (Prof. Lin-shan Lee)

Latent Dirichlet Allocation (LDA) (1/2)

- The basic generative process of LDA closely resembles PLSA; however,
  - In PLSA, the topic mixture $P(T_k|D)$ is conditioned on each document ($P(T_k|D)$ is fixed, unknown)
  - While in LDA, the topic mixture $P(T_k|D)$ is drawn from a Dirichlet distribution, so-called the conjugate prior, ($P(T_k|D)$ is unknown and follows a probability distribution)



Process of generating a corpus with LDA

1) Pick a multinomial distribution φ_T for each topic T from a Dirichlet distribution with parameter β

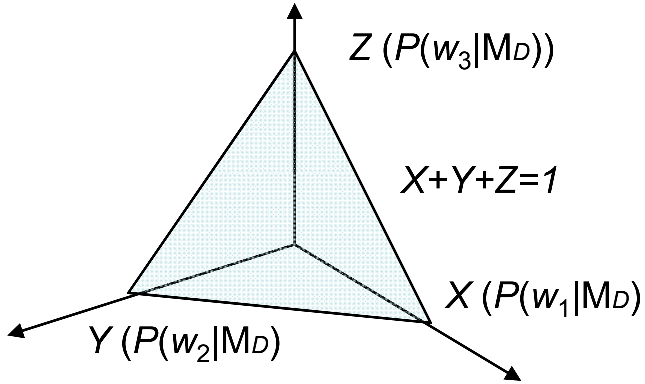
2) Pick a multinomial distribution θ_D for each docu D from a Dirichlet distribution with parameter α

3) Pick a topic $T \in \{1,2,\cdots,K\}$ from a multinomial distribution with parameter θ_D

4) Pick a *w*ord from a multinomial distribution with parameter φ_T

Blei et al. Latent Dirichlet allocation. Journal of Machine Learning Research, 2003

Latent Dirichlet Allocation (2/2)



$$P_{\mathrm{LDA}}(w_i \mid M_D) = \sum_{k=1}^{K} P(w_i \mid T_k) P(T_k \mid M_D)$$

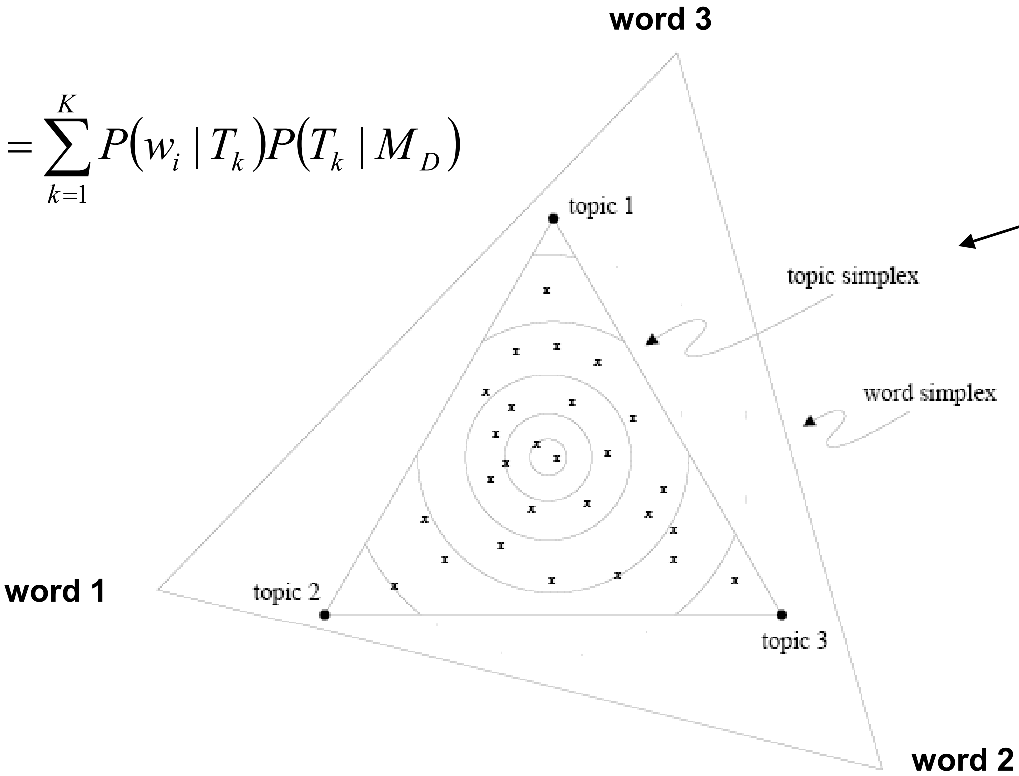


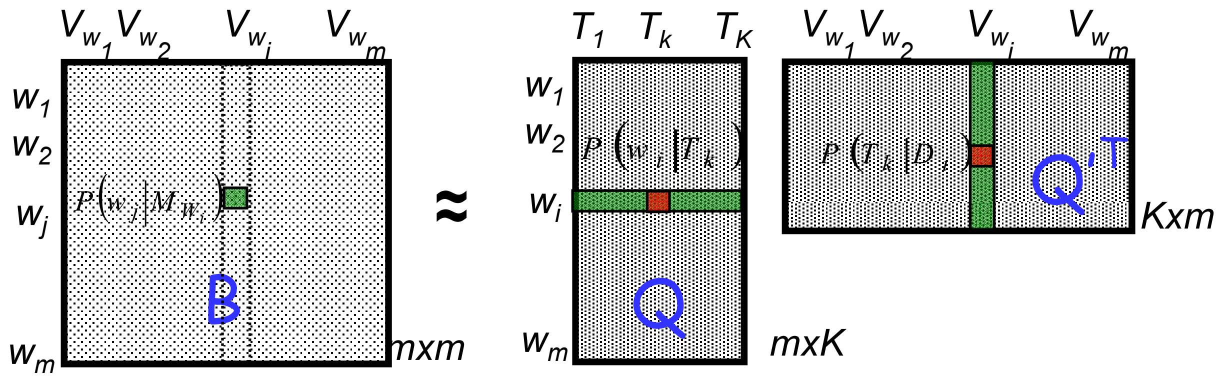
Figure 4: The topic simplex for three topics embedded in the word simplex for three words. The corners of the word simplex correspond to the three distributions where each word (respectively) has probability one. The three points of the topic simplex correspond to three different distributions over words. The mixture of unigrams places each document at one of the corners of the topic simplex. The pLSI model induces an empirical distribution on the topic simplex denoted by x. LDA places a smooth distribution on the topic simplex denoted by the contour lines.

Word Topic Models (WTM)

- Each word of language are treated as a word topical mixture model for predicting the occurrences of other words

$$P_{\mathrm{WTM}}\left(w_i \mid M_{w_j}\right) = \sum_{k=1}^{K} P\left(w_i \mid T_k\right) P\left(T_k \mid M_{w_j}\right)$$

- WTM also can be viewed as a nonnegative factorization of a "word-word" matrix consisting probability entries
  - Each column encodes the vicinity information of all occurrences of a certain type of word



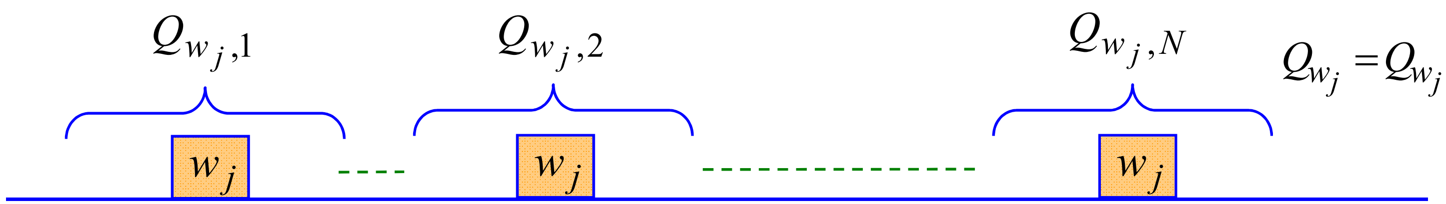
WTM: Information Retrieval (1/3)

- The relevance measure between a query and a document can be expressed by

$$P_{\mathrm{WTM}}(Q|D)=\prod_{w_i\in Q}\left[\sum_{w_j\in D}\alpha_{j,D}\sum_{k=1}^{K}P(w_i|T_k)P(T_k|\mathrm{M}_{w_j})\right]^{c(w_i,Q)}$$

- Unsupervised training
  - The WTM of each word can be trained by concatenating those words occurring within a context window of size around each occurrence of the word, which are postulated to be relevant to the word

$$\log L_{\mathbf{w}}=\sum_{w_j\in\mathbf{w}}\log P_{\mathrm{WTM}}\left(Q_{w_j}|\mathrm{M}_{w_j}\right)=\sum_{w_j\in\mathbf{w}}\sum_{w_i\in Q_{w_j}}c\left(w_i,Q_{w_j}\right)\log P_{\mathrm{WTM}}\left(w_i|\mathrm{M}_{w_j}\right)$$



$$Q_{w_j}=Q_{w_j,1},Q_{w_j,2},\cdots,Q_{w_j,N}$$

WTM: Information Retrieval (2/3)

- Supervised training: The model parameters are trained using a training set of query exemplars and the associated query-document relevance information
  - Maximize the log-likelihood of the training set of query exemplars generated by their relevant documents

$$\log L_{\mathbf{Q}_{TrainSet}} = \sum_{Q \in \mathbf{Q}_{TrainSet}} \sum_{D \in \mathbf{D}_{R\ \mathrm{to}\ Q}} \log P_{\mathrm{WTM}}(Q|D)$$

WTM: Information Retrieval (3/3)

- Formulas for Supervised Training

$$\hat{P}(w \mid T_k) = \frac{\sum_{Q \in [TrainSetQ]} \sum_{D_i \in [Doc]_{R \text{ to } Q}} n(w,Q) P(T_k \mid w, D_i)}{\sum_{Q' \in [TrainSetQ]} \sum_{D'_i \in [Doc]_{R \text{ to } Q'}} \sum_{w_n \in Q'} n(w_n, Q') P(T_k \mid w_n, D'_i)}$$

where

$$P(T_k \mid w, D_i) = \frac{P(w|T_k) \left[\sum_{w_j \in D_i} \alpha_{j,i} P\left(T_k \middle| M_{w_j}\right) \right]}{\sum_{l=1}^{K} \left[P(w|T_l) \sum_{w_j \in D_i} \alpha_{j,i} P\left(T_l \middle| M_{w_j}\right) \right]}$$

$$\hat{P}(T_k \mid M_{w_j}) = \frac{\sum_{Q \in [TrainSetQ]} \sum_{D_i \in [Doc]_{R \text{ to } Q}} \sum_{w \in Q} n(w,Q) P(M_{w_j} \mid w, M_{D_i}) P(T_k \mid w, M_{w_j})}{\sum_{Q' \in [TrainSetQ]} \sum_{D'_i \in [Doc]_{R \text{ to } Q'}} \sum_{w' \in Q'} n(w',Q') P(M_{w_j} \mid w', M_{D'_i})}$$

where

$$P(M_{w_j} \mid w, M_{D_i}) = \frac{\alpha_{j,i} \cdot P\left(w \middle| M_{w_j}\right)}{\sum_{w_l \in D_i} \alpha_{l,i} \cdot P\left(w \middle| M_{w_l}\right)}$$

and $P(T_k \mid w, M_{w_j}) = \dfrac{P(w|T_k) P\left(T_k \middle| M_{w_j}\right)}{\sum_{z=1}^{K} P(w|T_z) P\left(T_z \middle| M_{w_j}\right)}$

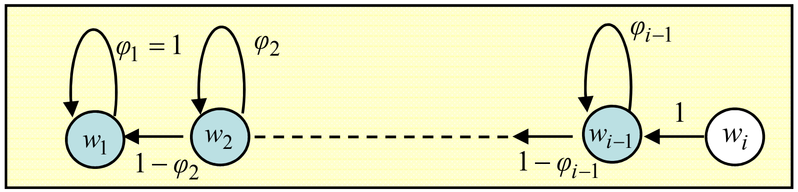
$$\begin{aligned}
& P(M_{w_j} \mid w, M_{D_i}) P(T_k \mid w, M_{w_j}) \\
&= \frac{\alpha_{j,i} P\left(w \middle| M_{w_j}\right)}{\sum_{w_l \in D_i} \alpha_{l,i} P\left(w \middle| M_{w_l}\right)} \cdot \frac{P(w|T_k) P\left(T_k \middle| M_{w_j}\right)}{\sum_{z=1}^{K} P(w|T_z) P\left(T_z \middle| M_{w_j}\right)} \\
&= \frac{\alpha_{j,i} P\left(w \middle| M_{w_j}\right)}{P\left(w \middle| M_{D_i}\right)} \cdot \frac{P(w|T_k) P\left(T_k \middle| M_{w_j}\right)}{P\left(w \middle| M_{w_j}\right)} \\
&= \frac{\alpha_{j,i} P(w|T_k) P\left(T_k \middle| M_{w_j}\right)}{P\left(w \middle| M_{D_i}\right)}
\end{aligned}$$

WTM: Dynamic Language Model Adaptation (1/2)

- For a decoded word w_i , we can again interpret it as a (single-word) query; while for each of its search histories, expressed by $H_{w_i} = w_1, w_2, \ldots, w_{i-1}$, we can linearly combine the associated WTM models of the words occurring in H_{w_i} to form a composite WTM model

$$P_{\mathrm{WTM}}\left(w_i \middle| \mathrm{M}_{H_{w_i}}\right) = \sum_{j=1}^{i-1} \beta_j P_{\mathrm{WTM}}\left(w_i \middle| \mathrm{M}_{w_j}\right) = \sum_{j=1}^{i-1} \beta_j \sum_{k=1}^{K} P\left(w_i \middle| T_k\right) P\left(T_k \middle| \mathrm{M}_{w_j}\right)$$

$$\beta_j = \varphi_j \prod_{s=1}^{i-j-1} \left(1 - \varphi_{j+s}\right)$$



  - $\beta_j = \varphi_j$ are nonnegative weighting coefficients which empirically set to be exponentially decayed as the word is being apart from w_i
  - ϕ_j is set to a fixed value (between 0 and 1) for $j = 2, \cdots, i-1$, and set to 1 for $j = 1$

WTM: Dynamic Language Model Adaptation (2/2)

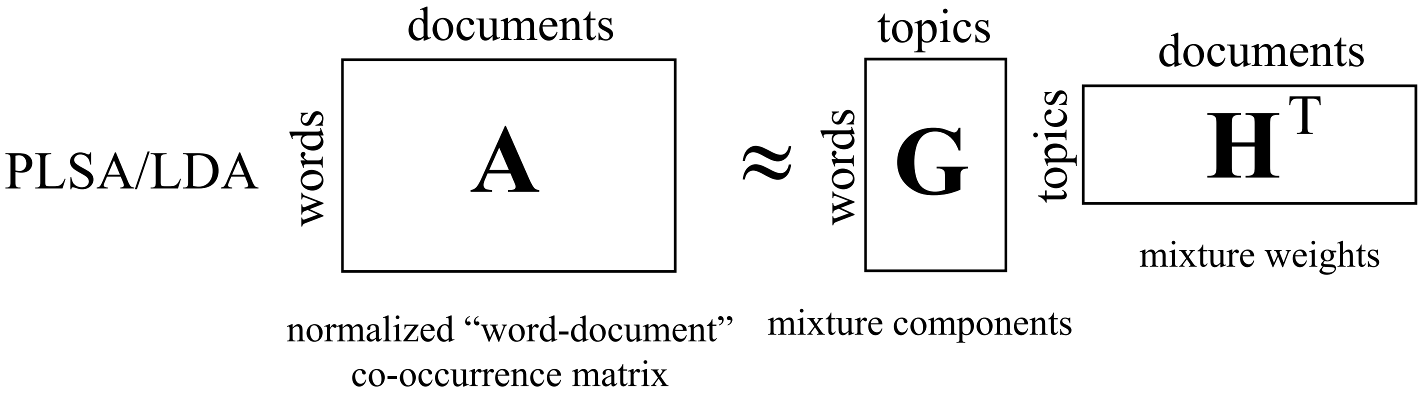
- For our speech recognition test data, it was experimentally observed that the language model access time of WTM was approximately 1/30 of that of PLSA for language model adaptation, as the iteration number of the online EM estimation of $P(T_k \mid H_{w_i})$ for PLSA was set to 5

$$P_{\mathrm{Adapt}}(w_i | w_{i-2} w_{i-1}) = \lambda \cdot P_{\mathrm{WTM}}(w_i | \mathrm{M}_{H_{w_i}}) + (1-\lambda) \cdot P_{\mathrm{BG}}(w_i | w_{i-2} w_{i-1})$$

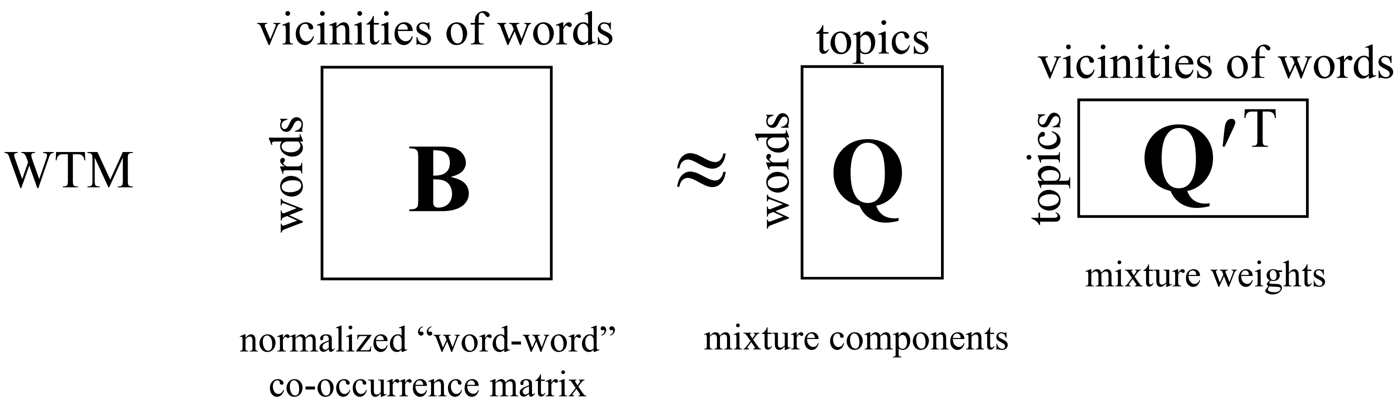
$\mathrm{BG}:$ background n-gram model

Comparison of WTMM and PLSA/LDA

- A schematic comparison for the matrix factorizations of PLSA/LDA and WTM

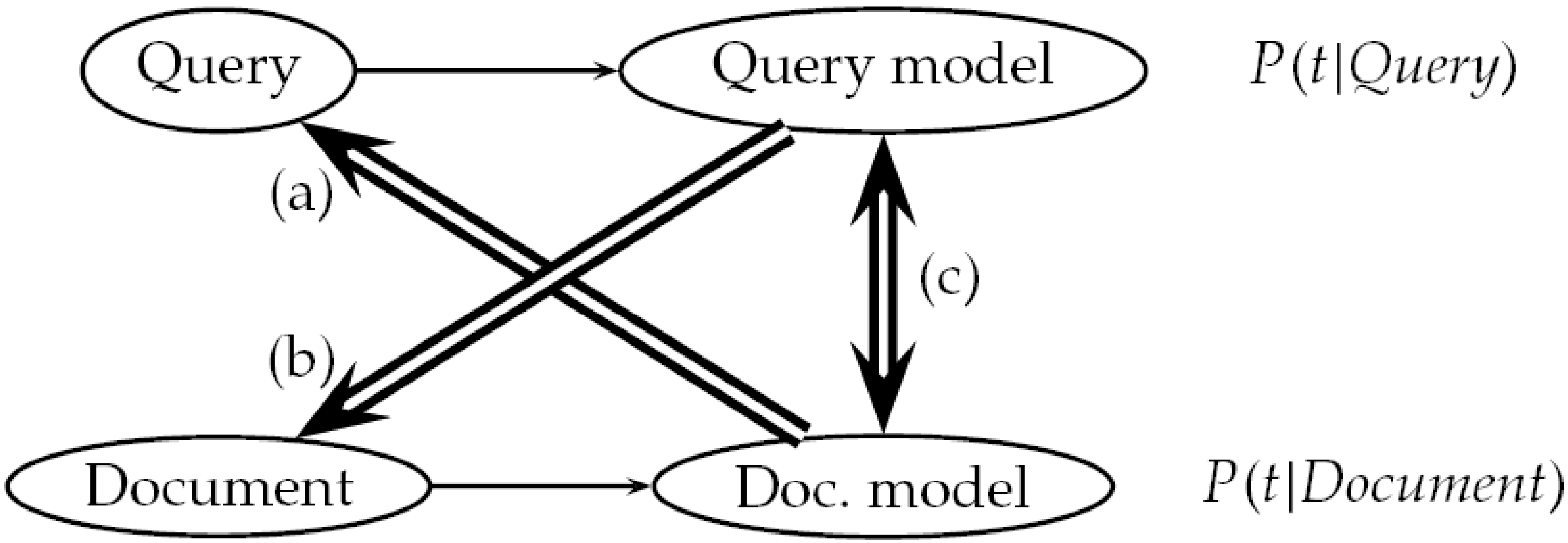


$$P_{\text{PLSA/LDA}}\left(w_i \mid \mathrm{M}_D\right) = \sum_{k=1}^{K} P\left(w_i \mid T_k\right) P\left(T_k \mid \mathrm{M}_D\right)$$



$$P_{\text{WTM}}\left(w_i \mid \mathrm{M}_{w_j}\right) = \sum_{k=1}^{K} P\left(w_i \mid T_k\right) P\left(T_k \mid \mathrm{M}_{w_j}\right)$$

Summary: Three Ways of Developing LM Approaches for IR



(a) Query likelihood
(b) Document likelihood
(c) Model comparison

literal term matching
or concept matching

LSA: SVDLIBC

- Doug Rohde's SVD C Library version 1.3 is based on the SVDPACKC library
- Download it at http://tedlab.mit.edu/~dr/

LSA: Exercise (1/4)

- Given a sparse term-document matrix
  - E.g., 4 terms and 3 docs

$$
\begin{array}{cc}
 & \text{Doc} \\
\text{Term} & \begin{array}{ccc} 2.3 & 0.0 & 4.2 \\ 0.0 & 1.3 & 2.2 \\ 3.8 & 0.0 & 0.5 \\ 0.0 & 0.0 & 0.0 \end{array}
\end{array}
$$

Row #Tem	Col. # Doc	Nonzero entries	
4	3	6	
2			2 nonzero entries at **Col 0**
0	2.3		Col 0, Row 0
2	3.8		Col 0, Row 2
1			1 nonzero entry at **Col 1**
1	1.3		Col 1, Row 1
3			3 nonzero entry at **Col 2**
0	4.2		Col 2, Row 0
1	2.2		Col 2, Row 1
2	0.5		Col 2, Row 2

  - Each entry can be weighted by *TF*x*IDF* score

- Perform SVD to obtain term and document vectors represented in the latent semantic space
- Evaluate the information retrieval capability of the LSA approach by using varying sizes (e.g., 100, 200,...,600 etc.) of LSA dimensionality

LSA: Exercise (2/4)

- Example: term-document matrix

Indexing Term no. | Doc no. | Nonzero entries

51253 2265 218852
77
508 7.725771
596 16.213399
612 13.080868
709 7.725771
713 7.725771
744 7.725771
1190 7.725771
1200 16.213399
1259 7.725771
……

- SVD command (IR_svd.bat)

svd -r st -o LSA100 -d 100 Term-Doc-Matrix

sparse matrix input | prefix of output files | No. of reserved eigenvectors | name of sparse matrix input

output → **LSA100-Ut** **LSA100-S** **LSA100-Vt**

LSA: Exercise (3/4)

- **LSA100-Ut**

51253 words

100 51253

0.003 0.001 ……..

0.002 0.002 …….

word vector (u^T): 1x100

- **LSA100-S**

100

2686.18

829.941

559.59

….

100 eigenvalues

- **LSA100-Vt**

2265 docs

100 2265

0.021 0.035 ……..

0.012 0.022 …….

doc vector (v^T): 1x100

LSA: Exercise (4/4)

- Fold-in a new *m*x1 query vector

$$\hat{q}_{1\times k} = \boxed{\left(q^T\right)_{1\times m} U_{m\times k}}\boxed{\Sigma^{-1}_{k\times k}}$$

Just like a row of V

Query represented by the weighted sum of it constituent term vectors

The separate dimensions are differentially weighted

- Cosine measure between the query and doc vectors in the latent semantic space

$$sim\left(\hat{q},\hat{d}\right) = coine\ (\hat{q}\Sigma,\hat{d}\Sigma) = \frac{\hat{q}\Sigma^2\hat{d}^T}{\left|\hat{q}\Sigma\right|\left|\hat{d}\Sigma\right|}$$