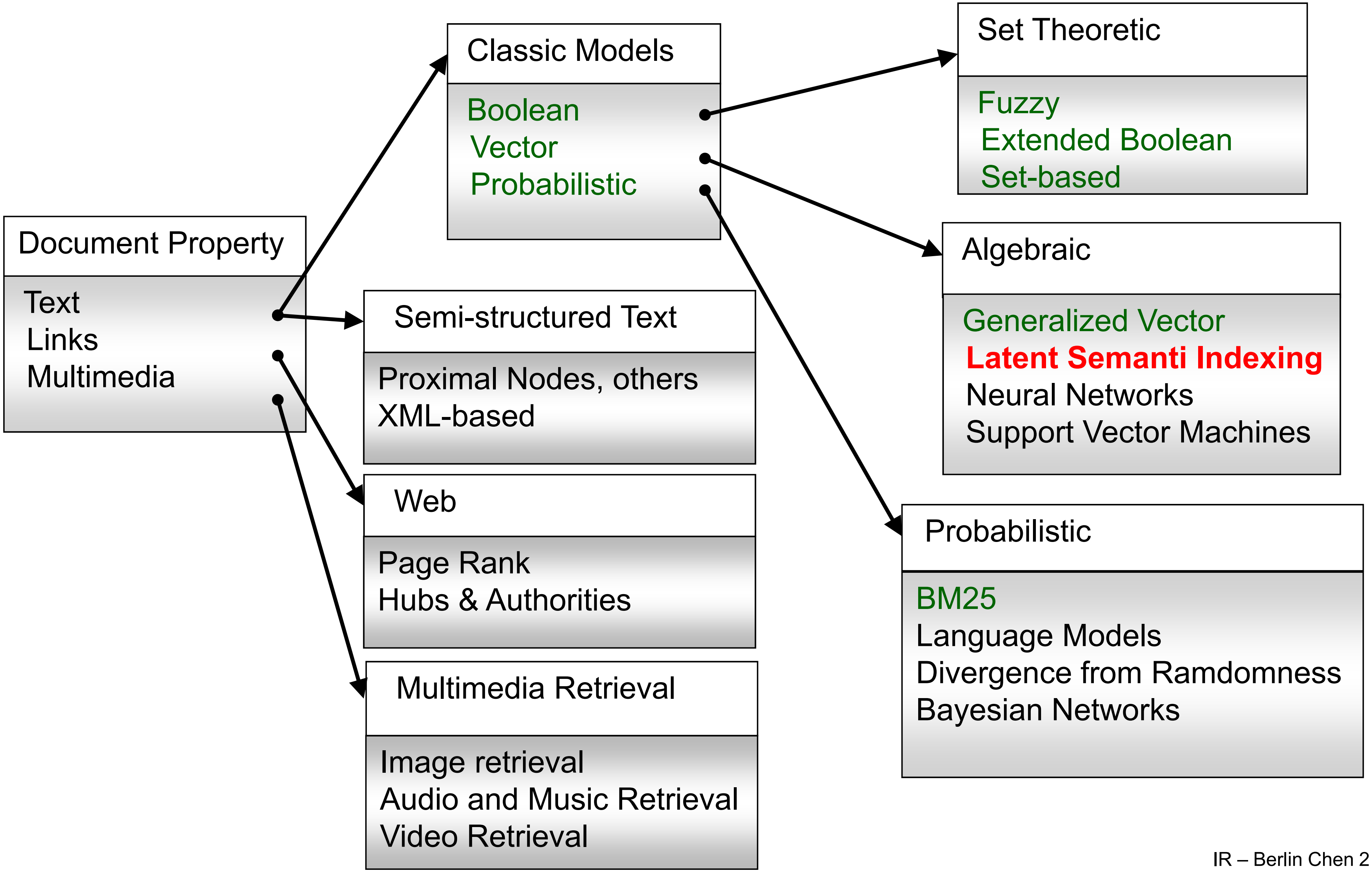
Latent Semantic Analysis

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

References:

1. 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*," SIGIR1988
2. J.R. Bellegarda, "*Latent semantic mapping*," IEEE Signal Processing Magazine, September 2005
3. T. K. Landauer, D. S. McNamara, S. Dennis, W. Kintsch (eds.) , *Handbook of Latent Semantic Analysis*, Lawrence Erlbaum, 2007
4. *Modern Information Retrieval,* Chapter 3

Taxonomy of Classic IR Models



Document Property

Text
Links
Multimedia

Classic Models

Boolean
Vector
Probabilistic

Semi-structured Text

Proximal Nodes, others
XML-based

Web

Page Rank
Hubs & Authorities

Multimedia Retrieval

Image retrieval
Audio and Music Retrieval
Video Retrieval

Set Theoretic

Fuzzy
Extended Boolean
Set-based

Algebraic

Generalized Vector
Latent Semanti Indexing
Neural Networks
Support Vector Machines

Probabilistic

BM25
Language Models
Divergence from Ramdomness
Bayesian Networks

Classification of IR Models Along Two Axes

- Matching Strategy
  - Literal term matching (matching word patterns between the query and documents)
    - E.g., Vector Space Model (VSM), Language Model (LM)
  - Concept matching (matching word meanings between the query and documents)
    - E.g., Latent Semantic Analysis (LSA), Probabilistic Latent Semantic Analysis (PLSA), Latent Dirichlet Allocation (LDA), Word Topic Model (WTM)
- Learning Capability
  - Term weighting, query expansion, document expansion, etc.
    - E.g., Vector Space Model, Latent Semantic Indexing
    - Most models are based on linear algebra operations
  - Solid theoretical foundations (optimization algorithms)
    - E.g., Language Model, Probabilistic Latent Semantic Analysis, Latent Dirichlet Allocation, Word Topic Model
    - Most models also belong to the language modeling (LM) approach

Two Perspectives for IR Models (cont.)

- Literal Term Matching vs. Concept Matching

中國解放軍蘇愷戰機

中共新一代空軍戰力

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, so literal terms in a user’s query may not match those of a relevant document

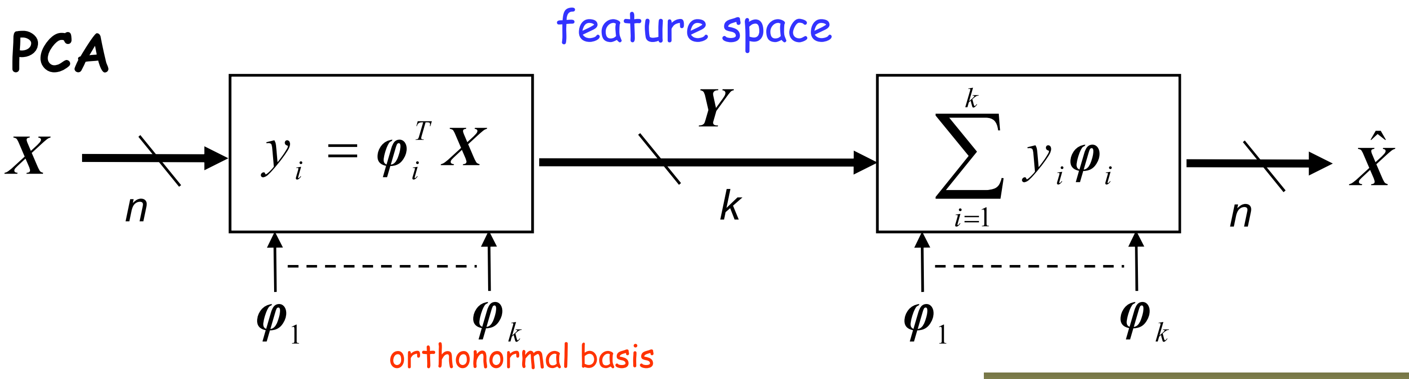
Latent Semantic Analysis (LSA)

- Also called Latent Semantic Indexing (LSI), Latent Semantic Mapping (LSM), or Two-Mode Factor Analysis
  - Three important claims made for LSA
    - The **semantic information** can derived from a word-document co-occurrence matrix
    - The **dimension reduction** is an essential part of its derivation
    - **Words and documents can be represented as points** in the Euclidean space
  - LSA exploits the meanings of words by removing “noise” that is present due to the variability in word choice
    - Namely, synonymy and polysemy that are found in documents

T. Landauer, D. S. McNamara, S. Dennis, & W. Kintsch (Eds.), *Handbook of Latent Semantic Analysis.* Hillsdale, NJ: Erlbaum.

LSA: Schematic Representation

- Dimension Reduction and Feature Extraction
  - **PCA**



feature space

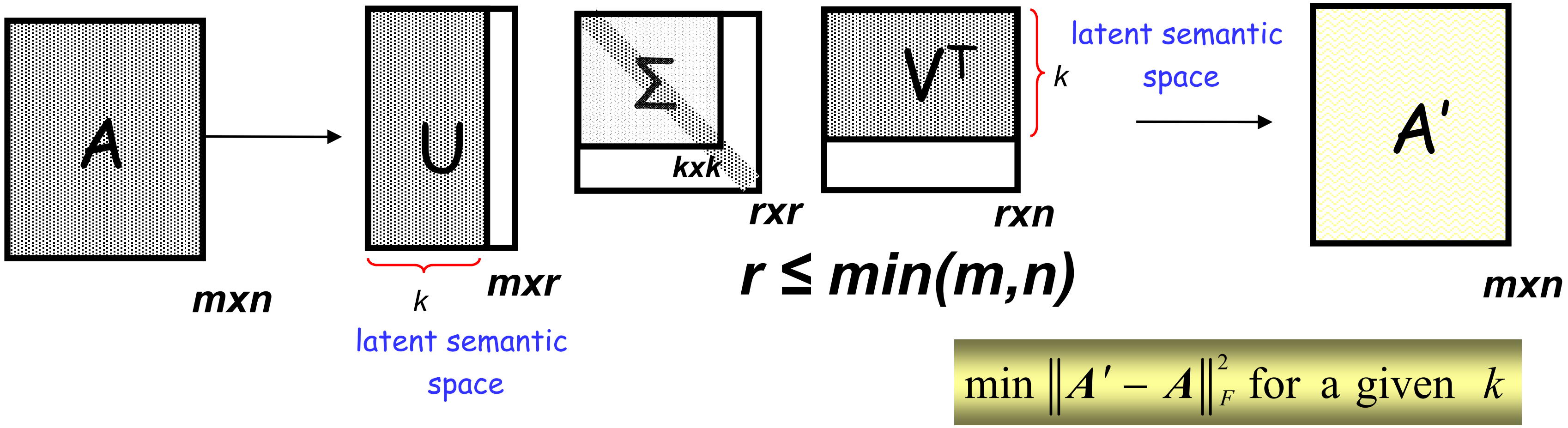
X n $y_i = \varphi_i^T X$ Y k $\sum_{i=1}^{k} y_i \varphi_i$ n $\hat{X}$

φ_1 φ_k φ_1 φ_k

orthonormal basis

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

  - **SVD (in LSA)**



A mxn U mxr k latent semantic space Σ kxk rxr V

V^T k latent semantic space rxn A' mxn

$r \le \min(m,n)$

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

LSA: Balancing Two Opposing Effects

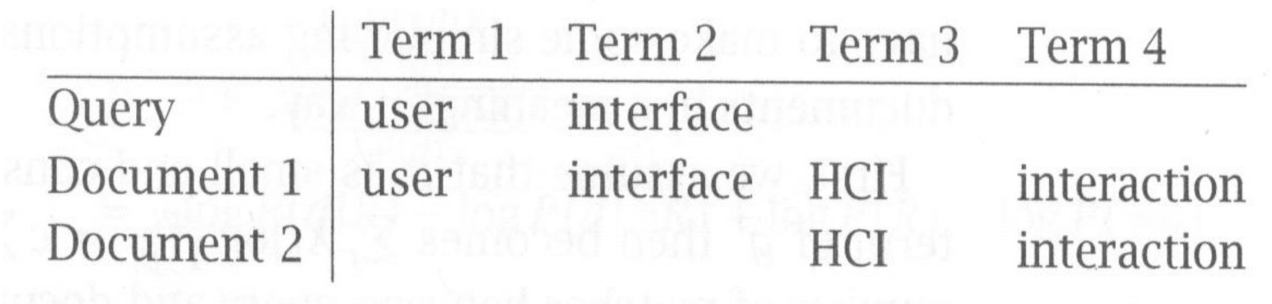
- First, *k* should be large enough to allowing fitting all the (semantic) structure in the real data

- Second, *k* should be small enough to allow filtering out the non-relevant representational details (which are present in the conventional index-term based representation)

Therefore, as will be seen shortly, LSA provides a mechanism for elimination of noise (presented in index-based representations) and removal of redundancy.

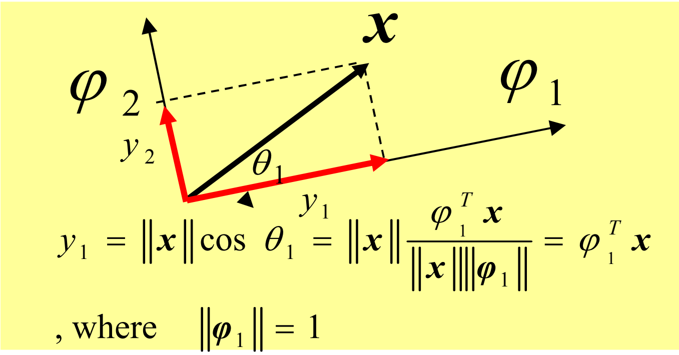
LSA: An Example

- 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 $\boldsymbol{x}$:

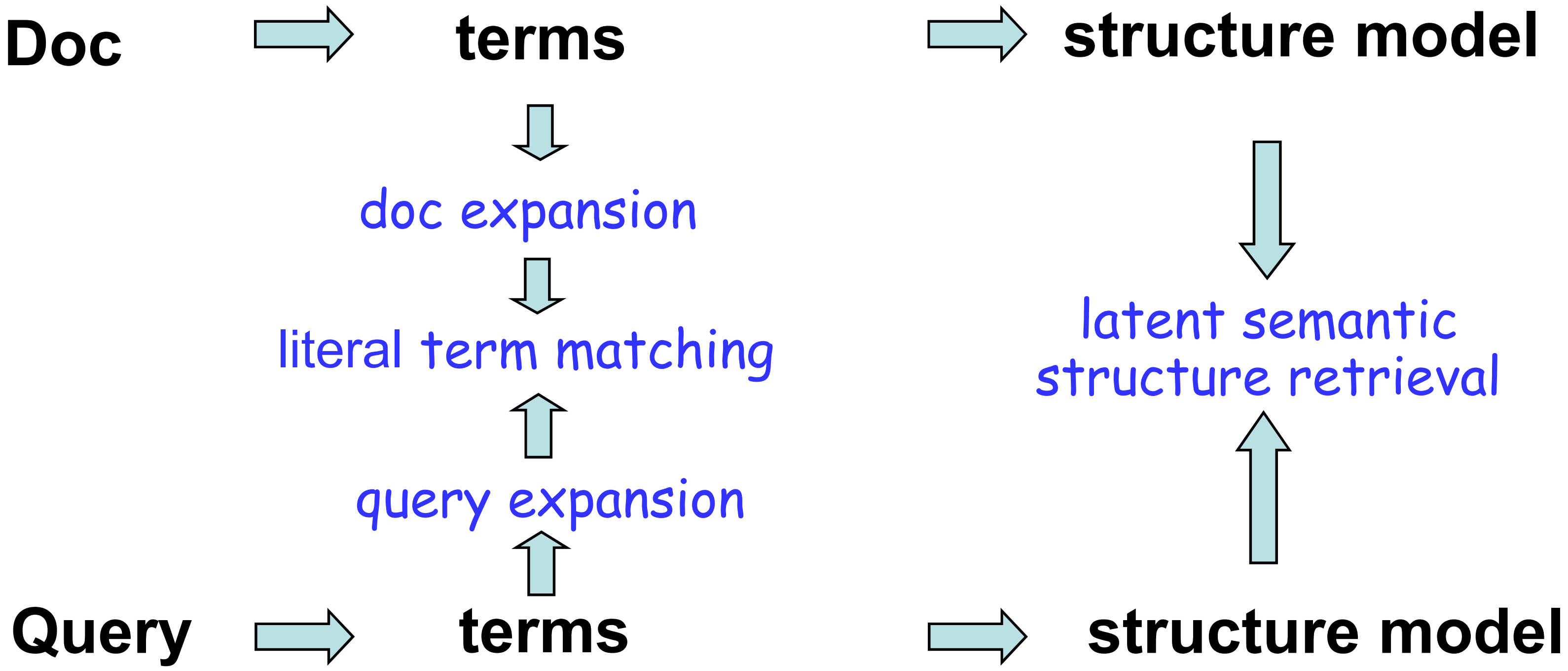


$$y_1 = \|\boldsymbol{x}\| \cos \theta_1 = \|\boldsymbol{x}\| \frac{\varphi_1^T \boldsymbol{x}}{\|\boldsymbol{x}\| \|\boldsymbol{\varphi}_1\|} = \varphi_1^T \boldsymbol{x}$$

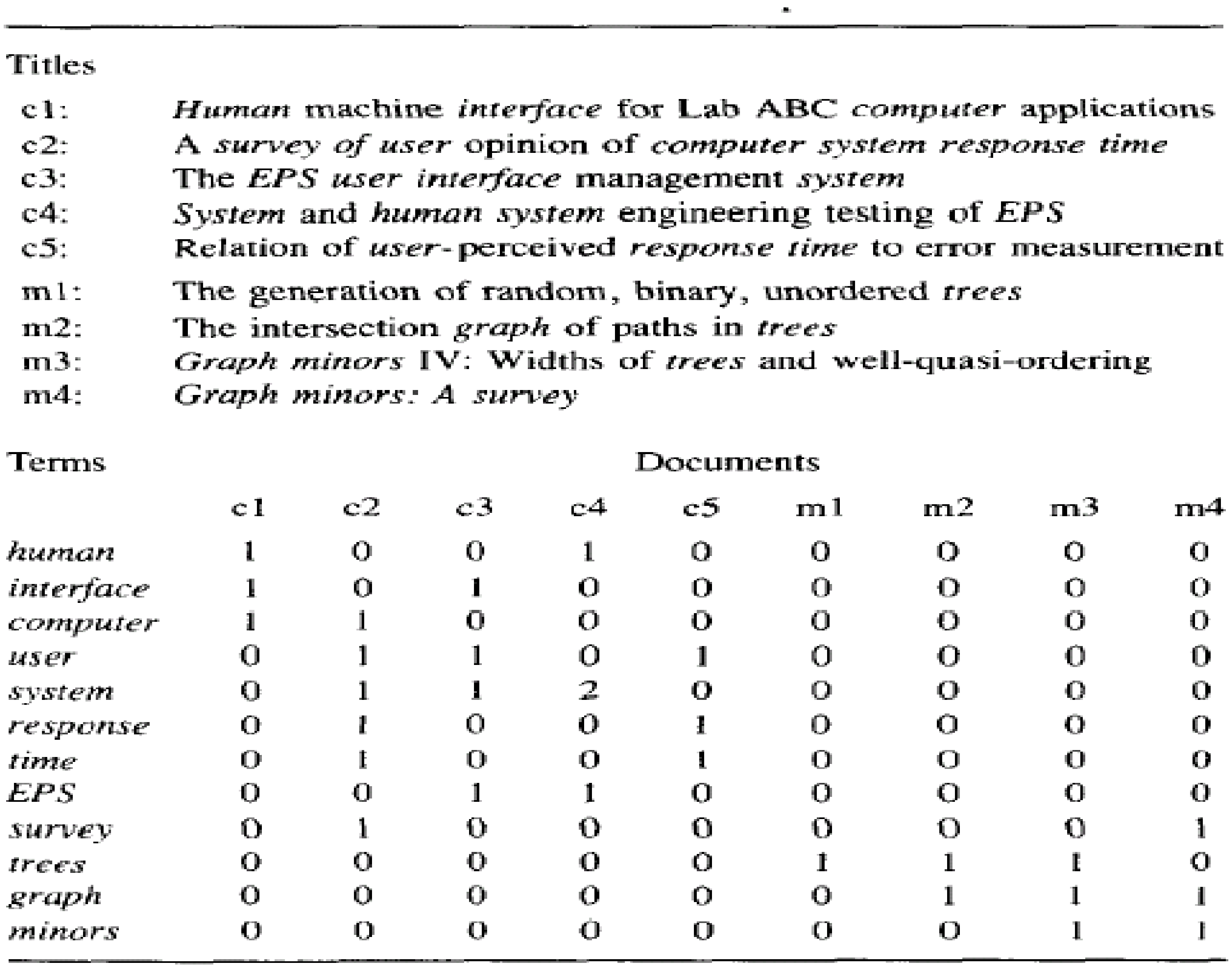
, where $\|\boldsymbol{\varphi}_1\| = 1$

LSA: Latent Structure Space

- Two alternative frameworks to circumvent vocabulary mismatch



LSA: Another Example (1/2)



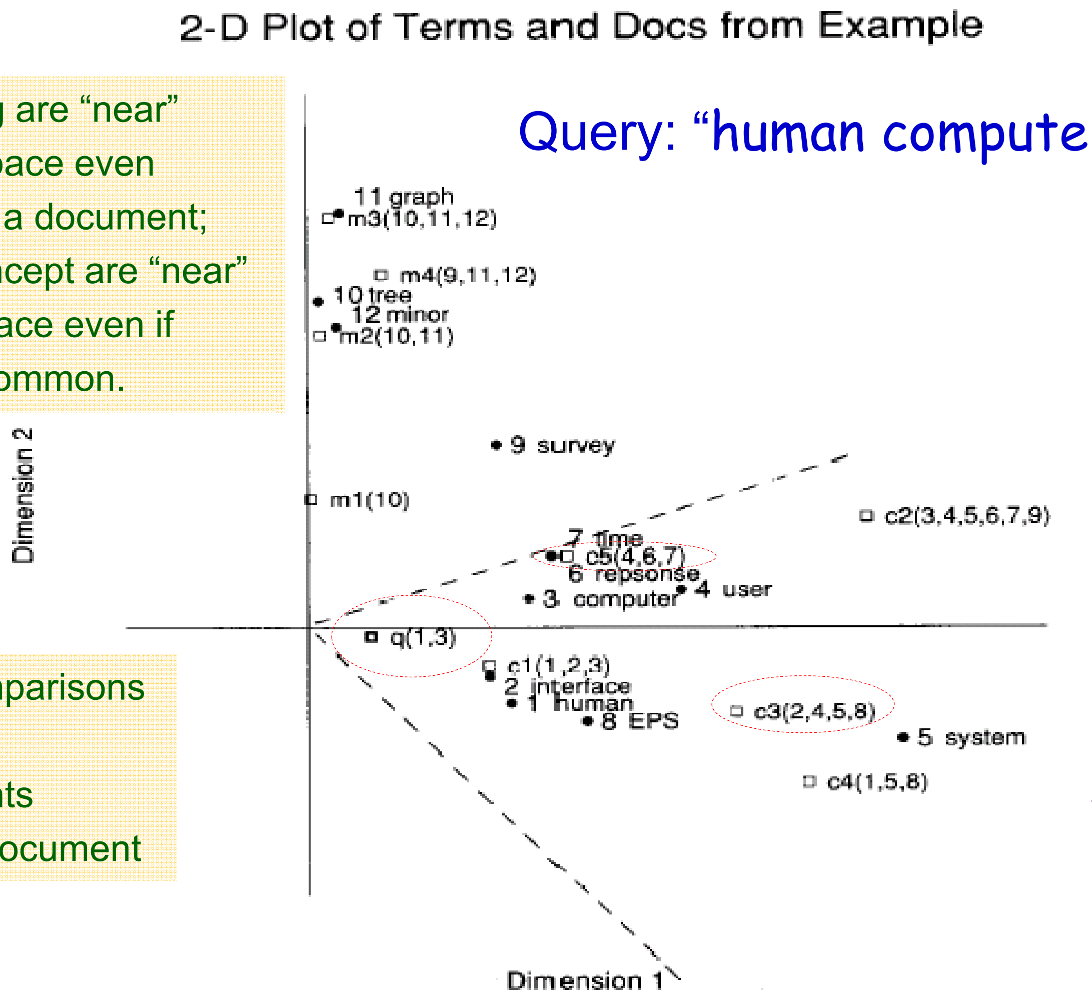
Titles

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)

Words similar in meaning are “near” each other in the LSA space even if they never co-occur in a document; Documents similar in concept are “near” each other in the LSA space even if they share no words in common.



Query: “human computer interaction”

An OOV word

Three sorts of basic comparisons

- Compare two words
- Compare two documents
- Compare a word to a document

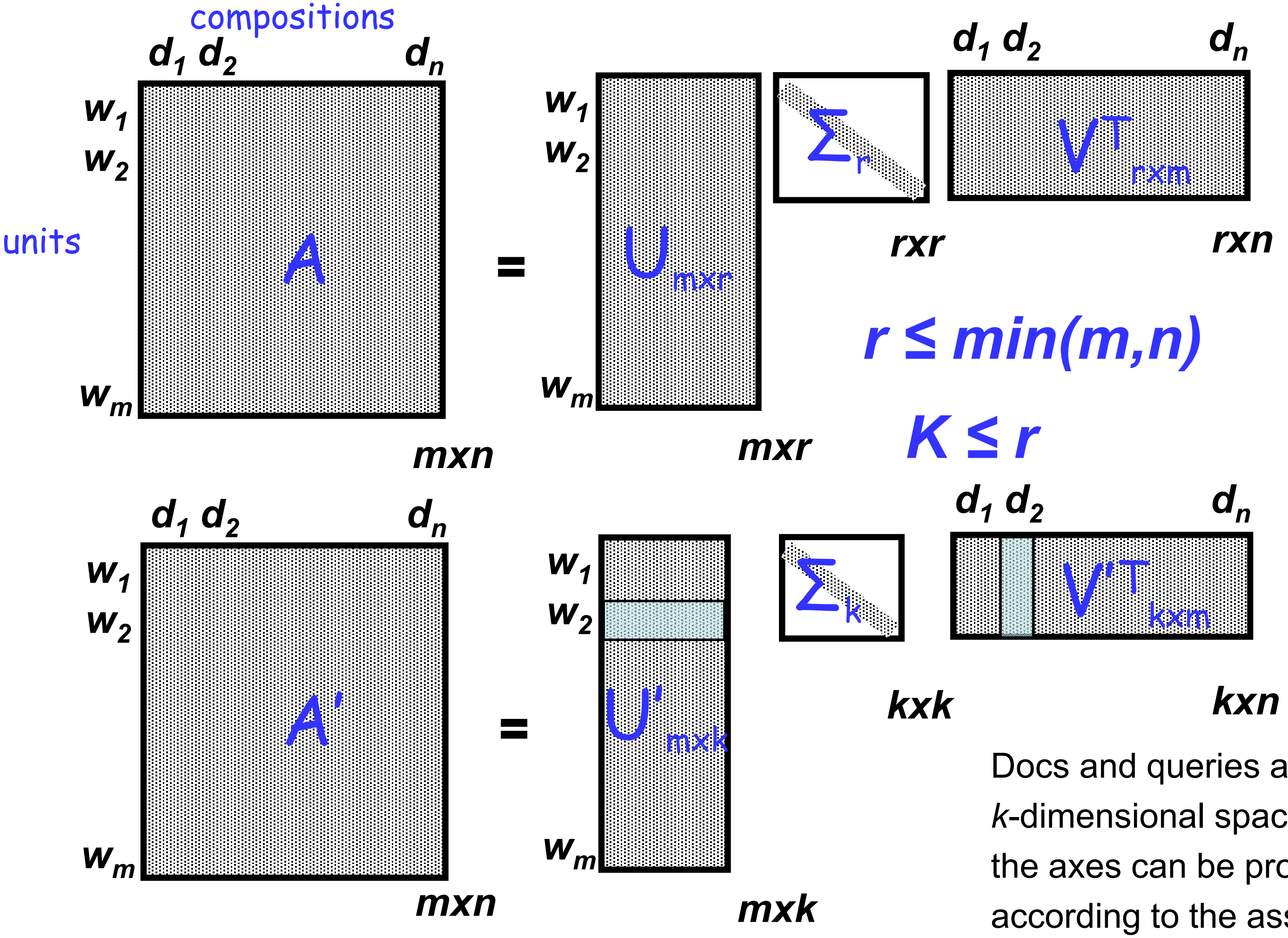
FIG. 1. A two-dimensional plot of 12 Terms and 9 Documents from the sampe TM set. Terms are represented by filled circles. Documents are shown as open squares, and component terms are indicated parenthetically. The query ("human computer interaction") is represented as a pseudo-document at point q. Axes 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 (1/10)

- Singular Value Decomposition (SVD)

Row A $\in R^n$

Col A $\in R^m$



$r \leq min(m,n)$

$K \leq r$

Both U and V has orthonormal column vectors

$U^TU=I_{rXr}$

$V^TV=I_{rXr}$

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

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

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 (2/10)

- "term-document" matrix *A* has to do with the co-occurrences between terms (or units) and documents (or compositions)
  - Contextual 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

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

negative normalized entropy

document length

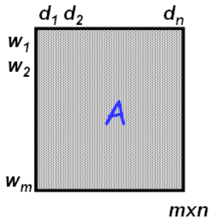
normalized entropy of term *i*

Total 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 (3/10)



- Singular Value Decomposition (SVD)
  - A^TA is symmetric nxn matrix
    - All eigenvalues λ_j are nonnegative real numbers

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

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

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

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

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

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

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

LSA: Theoretical Foundation (4/10)

- $\{Av_1, Av_2, \ldots., Av_r\}$ is an orthogonal basis of Col A

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

– Suppose that A (or A^TA) has rank $r \le n$

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

– 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)

$$\Rightarrow [u_1\ u_2 \ldots u_r]\Sigma_r = A\boxed{[v_1\ v_2\ \ \ v_r]}$$

V: an orthonormal matrix

Known in advance

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

$$\Rightarrow [u_1\ u_2 \ldots u_r \ldots u_m]\Sigma = A[\overbrace{v_1\ v_2 \ldots v_r \ldots v_n}]$$

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

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

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

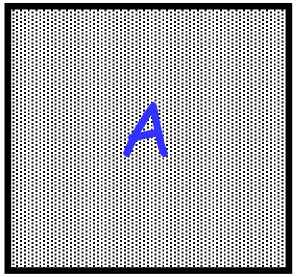
$$\|A\|_F^2 = \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2$$

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

LSA: Theoretical Foundation (5/10)

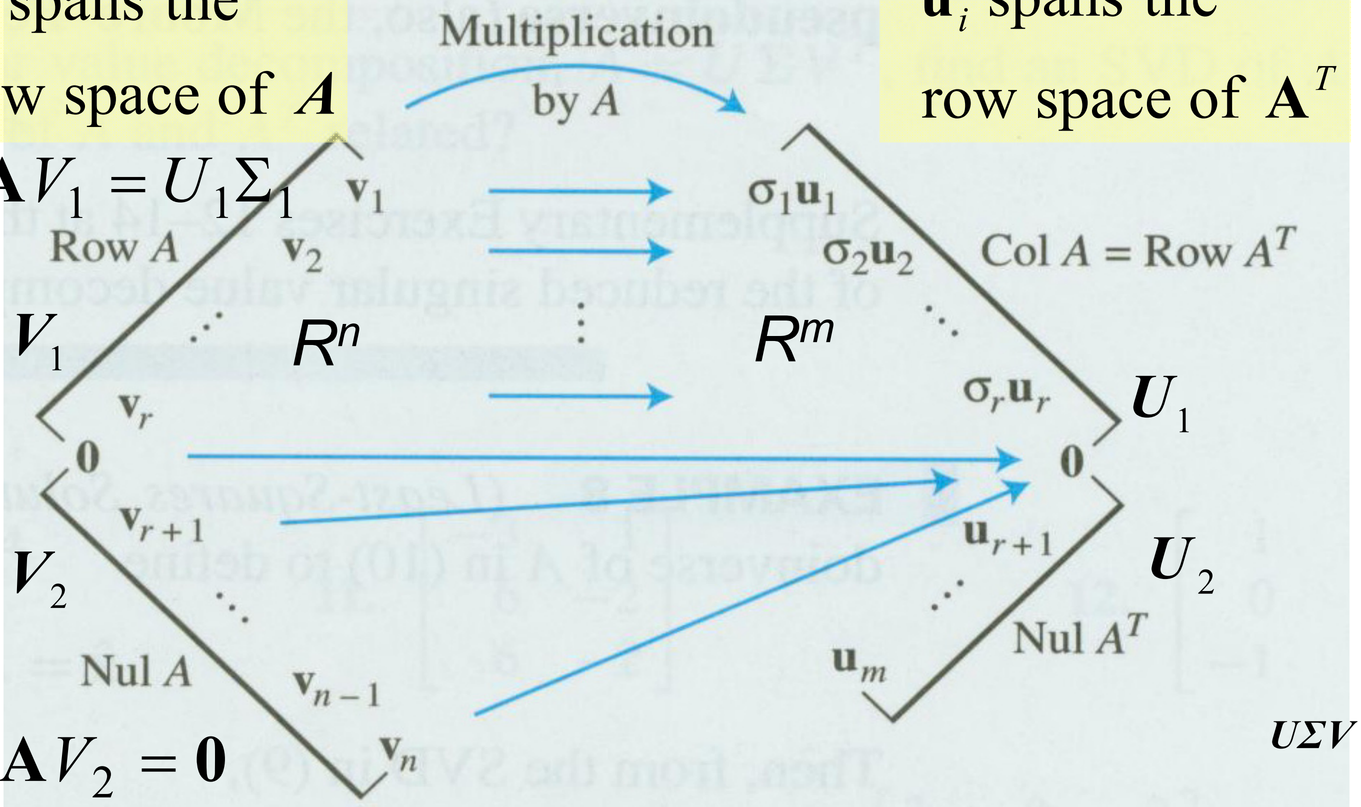
$\boldsymbol{v}_i$ spans the row space of $\mathbf{A}$

$\mathbf{u}_i$ spans the row space of $\mathbf{A}^T$



mxn

$$\mathbf{A}V_1 = U_1\Sigma_1$$



$$\mathbf{A}V_2 = \mathbf{0}$$

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

$$\boldsymbol{U\Sigma V}^T = \begin{pmatrix}\boldsymbol{U}_1 & \boldsymbol{U}_2\end{pmatrix}\begin{pmatrix}\boldsymbol{\Sigma}_1 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{pmatrix}\begin{pmatrix}\boldsymbol{V}_1^T\\ \boldsymbol{V}_2^T\end{pmatrix}$$
$$= \boldsymbol{U}_1\boldsymbol{\Sigma}_1\boldsymbol{V}_1^T$$
$$= \boldsymbol{A}\boldsymbol{V}_1\boldsymbol{V}_1^T$$
$$= \boldsymbol{A}$$

$U\Sigma = AV$

LSA: Theoretical Foundation (6/10)

- 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

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

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

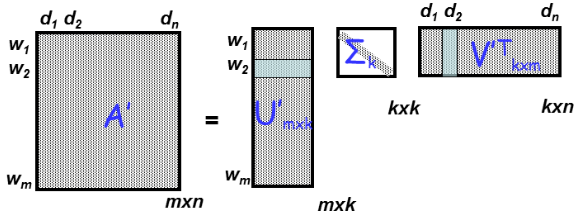
    - the *i*-th entry of a row of U is related to the projection of a corresponding row of A onto the *i*-th column of V

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

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

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

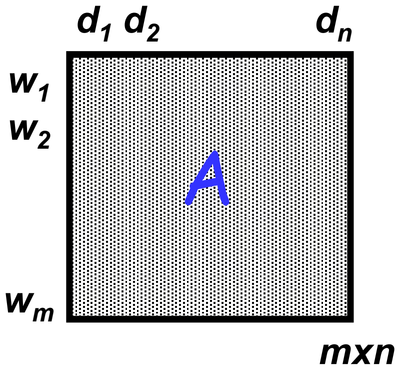
$$\Rightarrow V \Sigma = A^T U$$



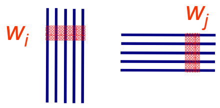
    - the *i*-th entry of a row of V is related to the projection of a corresponding row of A^T onto the *i*-th column of U

LSA: Theoretical Foundation (7/10)

- 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*')

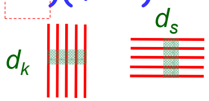
$U'=U_{mxk}$
$\Sigma'=\Sigma_k$
$V'=V_{nxk}$

- Compare two terms $A'A'^T=(U'\Sigma'V'^T)\ (U'\Sigma'V'^T)^T=U'\Sigma'\boxed{V'^TV'}\Sigma'^TU'^T=(U'\boxed{\Sigma'})(U'\Sigma')^T$
  - → dot product of two rows of *U'Σ'*

I_{rxr}

For stretching or shrinking

- Compare two docs $A'^TA'=(U'\Sigma'V'^T)^T\,'(U'\Sigma'V'^T)=V'\Sigma'^T\boxed{U^T\,U'}\Sigma'V'^T=(V'\boxed{\Sigma'})(V'\Sigma')^T$
  - → dot product of two rows of *V'Σ'*



- Compare a query word and a doc → each individual entry of A' (scaled by the square root of singular values)

LSA: Theoretical Foundation (8/10)

- **Fold-in**: find the representation for a pseudo-document *q*
  - For objects (new queries or docs) that did not appear in the original analysis
    - Fold-in a new *m*x1 query (or doc) vector

See Figure A in next page

Just like a row of V

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

The separate dimensions are differentially weighted.

Query is represented by the weighted sum of it constituent term vectors scaled by the inverse of singular values.

remember that

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

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

$$= V\Sigma\Sigma^{T}U = V\Sigma$$

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

$$\Rightarrow V = A^{T}U\Sigma^{-1}$$

  - 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 (docs are sorted in descending order of their cosine values)

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

row vectors

LSA: Theoretical Foundation (9/10)

- 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

remember that

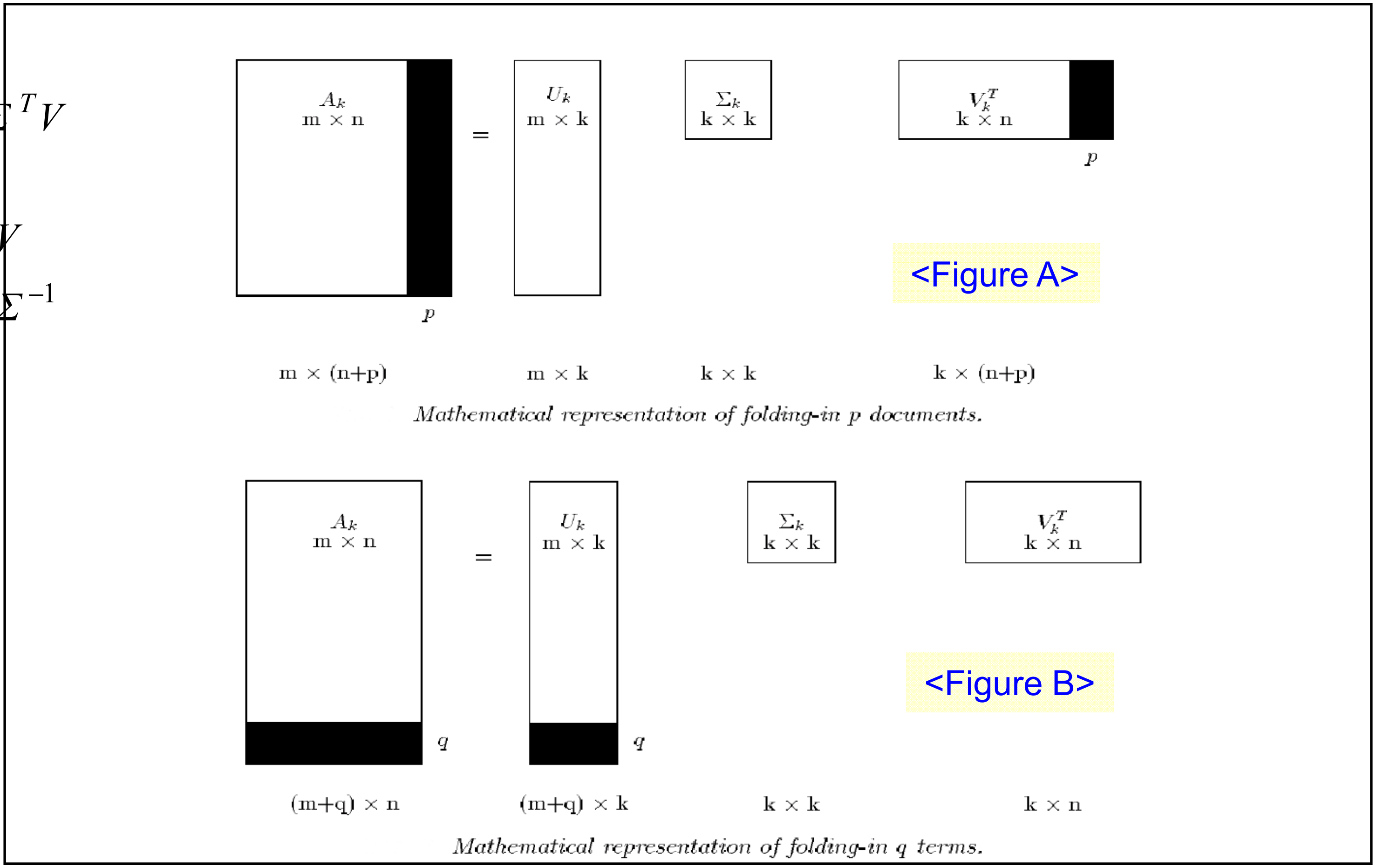
$$A = U\Sigma\Sigma^{T}$$

$$\Rightarrow AV = U\Sigma\Sigma^{T}V$$

$$= U\Sigma$$

$$\Rightarrow \quad U\Sigma = AV$$

$$\Rightarrow \quad U = AV\Sigma^{-1}$$



<Figure A>

Mathematical representation of folding-in p documents.

<Figure B>

Mathematical representation of folding-in q terms.

LSA: Theoretical Foundation (10/10)

- Note that the first *k* columns of *U* and *V* are orthogonal, but the rows of *U* and *V* (i.e., the word and document vectors), consisting *k* elements, are not orthogonal

- Alternatively, *A* can be written as the sum of *k* rank-1 matrices

$$A \approx A_k = \sum_{i=1}^{k} u_i \sigma_i v_i^T$$

  - u_i and v_i are respectively the eigenvectors of *U* and *V*

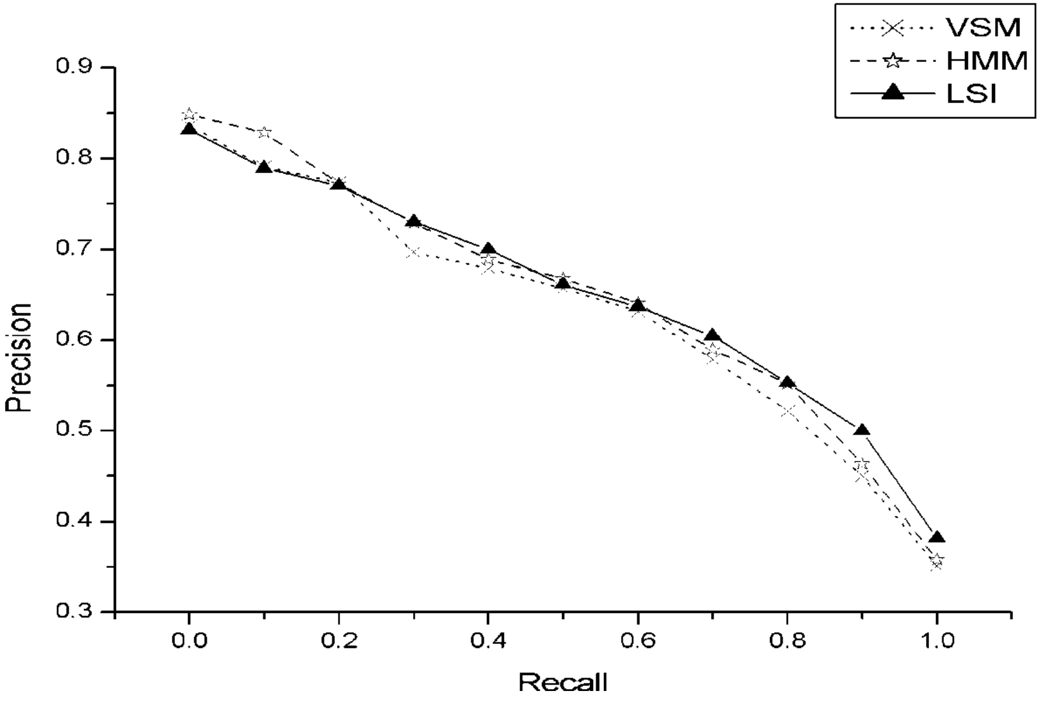
- LSA with relevance feedback (query expansion)

$$\hat{q}_{1\times k} = \left(q^T\right)_{1\times m} U_{m\times k} \Sigma_{k\times k}^{-1} + \left(d^T\right)_{1\times n} V_{n\times k}$$

  - d is a binary vector whose elements specify which documents to add to the query

LSA: A Simple 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)

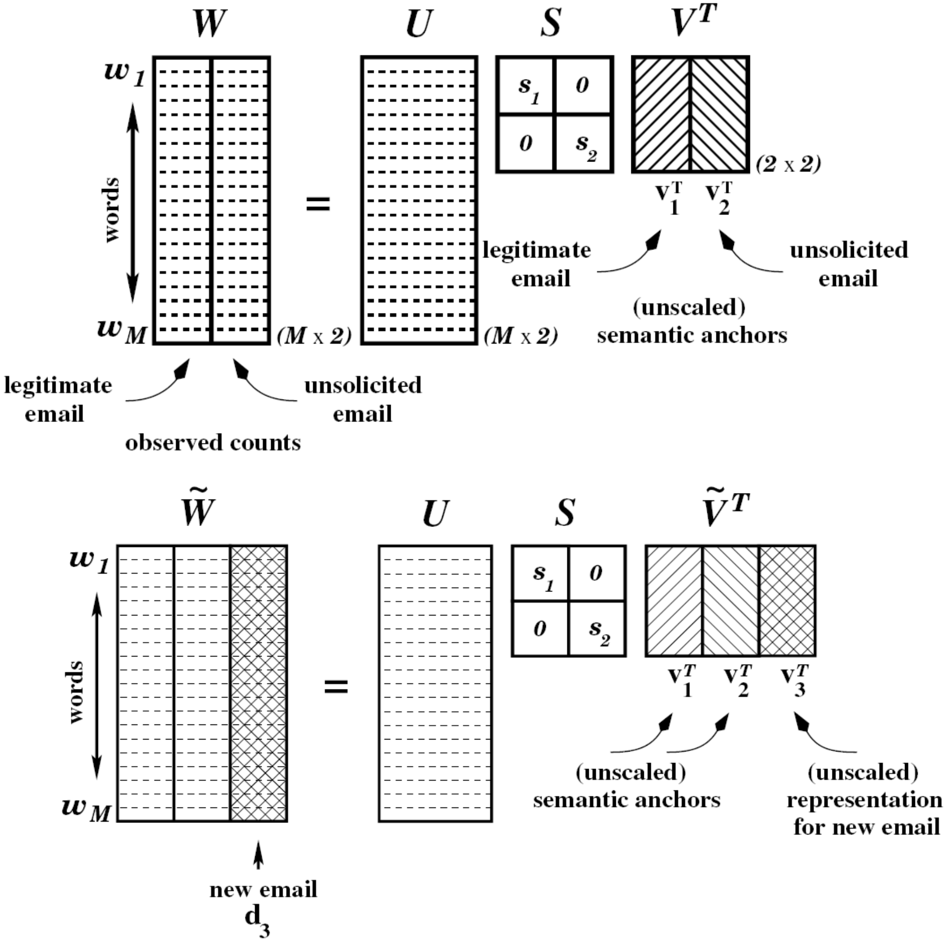
- Pro (Advantages)
  - A clean formal framework and a clearly defined optimization criterion (least-squares)
    - Conceptual simplicity and clarity
  - Handle synonymy problems ("heterogeneous vocabulary")
    - Replace individual terms as the descriptors of documents by independent "**artificial concepts**" 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
  - Contextual or positional information for words in documents is discarded (the so-called ***bag-of-words*** assumption)
  - 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)
  - To date, aside from folding-in, there is no optimal way to add information (new words or documents) to an existing word-document space
    - Re-compute SVD (or the reduced space) with the added information is a more direct and accurate solution

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

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

- 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

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

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

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

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

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

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

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

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

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 $\tilde{d}_{q-1}$:

$$\Pr(w_q \mid \tilde{d}_{q-1})$$

$$\approx K(w_q \mid \tilde{d}_{q-1})$$

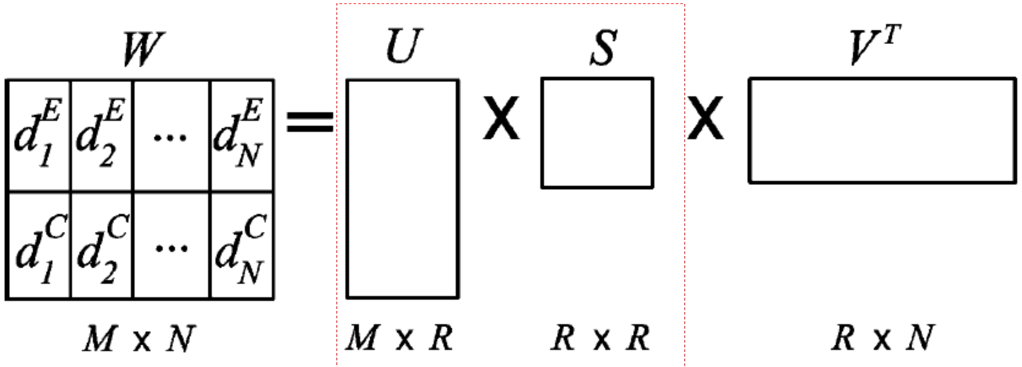
$$= \cos(u_q S^{1/2}, \tilde{v}_{q-1} S^{1/2}) = \frac{u_q S \tilde{v}_{q-1}^T}{\left\| u_q S^{1/2} \right\| \left\| \tilde{v}_{q-1} S^{1/2} \right\|}$$

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*").

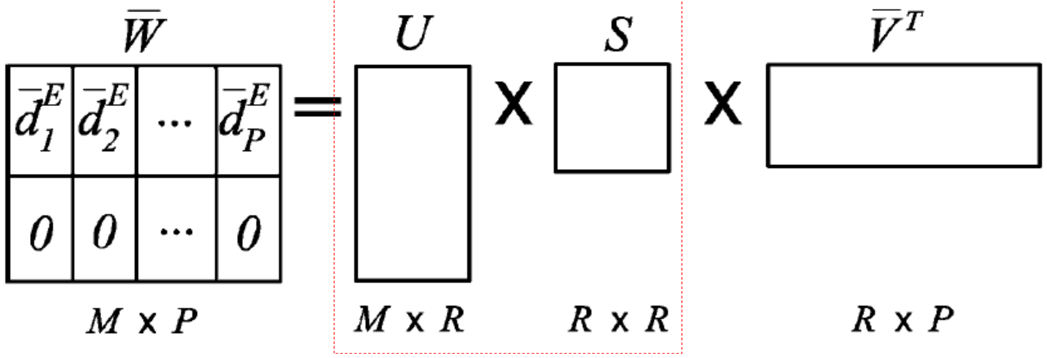
J. Bellegarda, Latent Semantic Mapping: Principles & Applications (Synthesis Lectures on Speech and Audio Processing), 2008

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

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

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

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

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

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

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

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

- 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

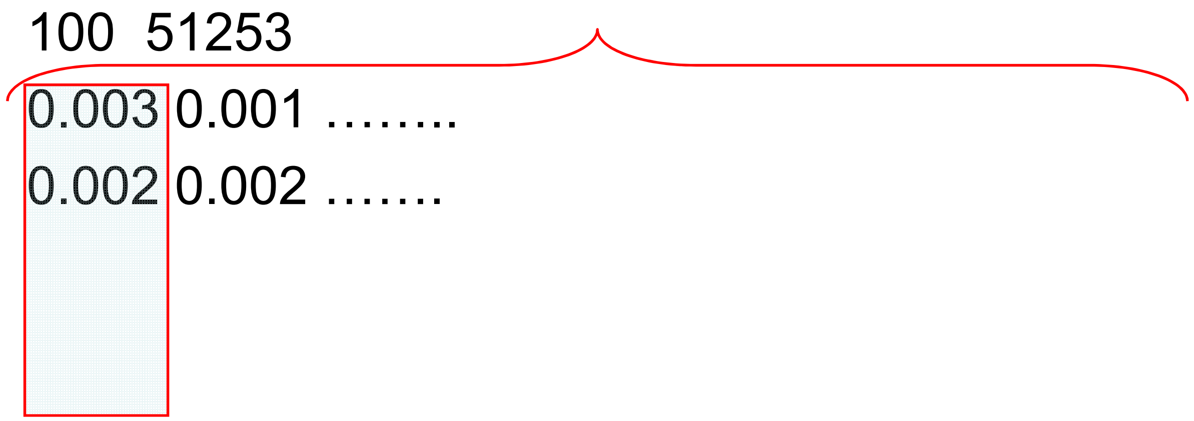
LSA100-Ut

LSA100-S

LSA100-Vt

LSA: Exercise (3/4)

- **LSA100-Ut**



51253 words

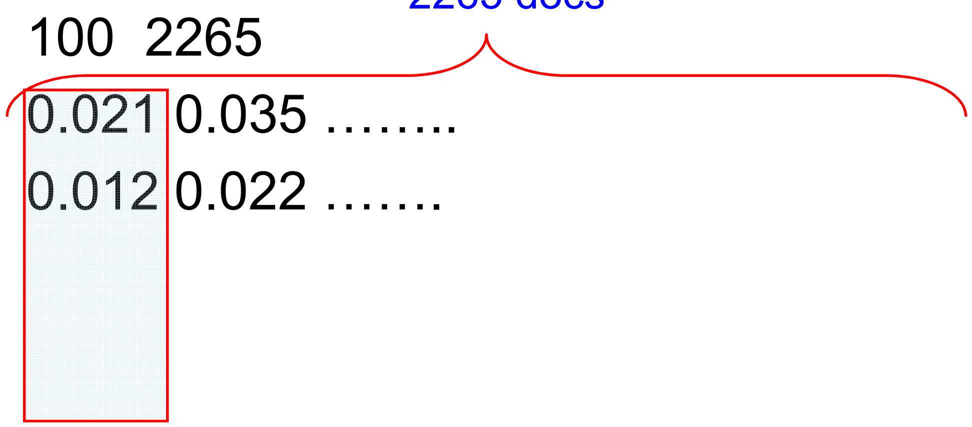
100 51253

0.003 0.001 ……..

0.002 0.002 …….

word vector (u^T): 1x100

- **LSA100-Vt**



2265 docs

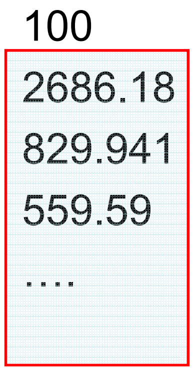
100 2265

0.021 0.035 ……..

0.012 0.022 …….

doc vector (v^T): 1x100

- **LSA100-S**



100

2686.18

829.941

559.59

….

100 eigenvalues

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

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