ECE8813
Statistical Natural Language Processing

Lectures 16-17: Latent Semantic Analysis and Indexing

Chin-Hui Lee
School of ECE, Georgia Tech
Atlanta, GA 30332, USA
chl@ece.gatech.edu

Outline of Computational Semantics

- Semantics in understanding documents
- Approaches to semantic analysis
- One key technique: latent semantic analysis
  - Building latent semantic space
  - Projection of a text unit in latent semantic space
  - Semantic similarity measure
- Application areas

From Syntax to Semantics

- Syntax - structure of words, phrases and sentences
- Semantics - meaning of and relationships among words in a sentence
  - Extracting an important *meaning* from a given text document
  - Contextual meaning

Approaches to Semantic Analysis

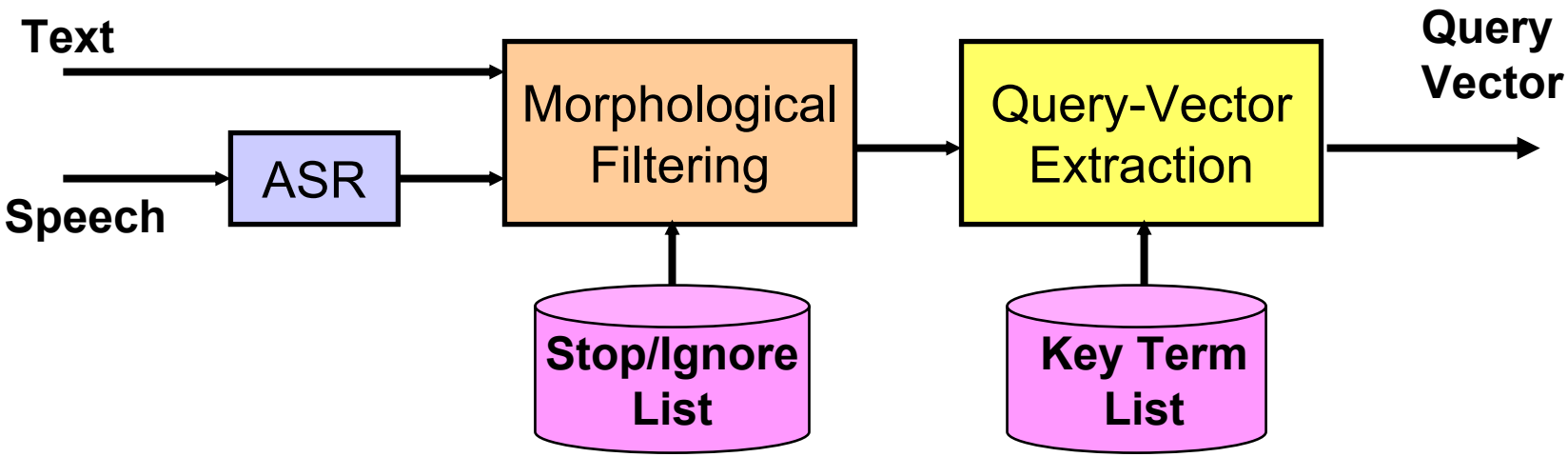
- Compositional semantics
  - parse tree to derive a hierarchical structure
  - informational and intentional meaning
  - rule based
- Classification
  - Bayesian approach
- Statistics-algebraic approach
  - Latent semantic analysis/indexing (LSA/LSI)
    - Fully automatic: extracting and inferring relations of expected contextual usage of words in documents
    - No manual effort: Using no hand-constructed dictionaries, knowledge bases, semantic networks and grammar

Information Retrieval Issues

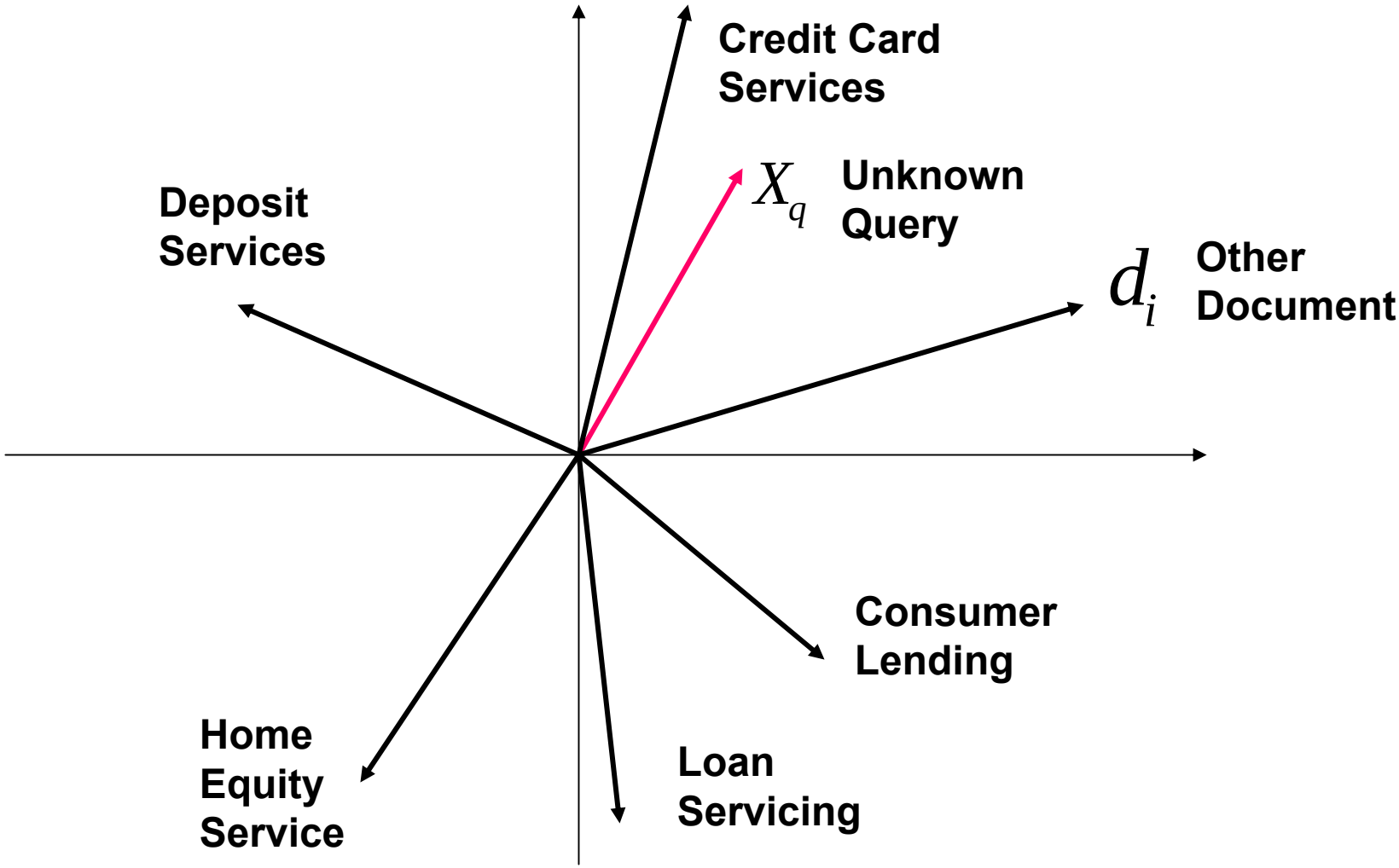
- Information Retrieval in the 1980s
- Given a collection of documents: retrieve documents that are relevant to a given query
- Match terms in documents to terms in query
- Vector space method (VSM)
  - Computational similarity measures
  - From language theory to language engineering

Query Vector Feature Extraction

- Text pre-processing (SMART, Salton, 1971)
  - Extract root form of a word, e.g. *check* for *checking*
  - Remove ignore words, e.g. *um, uh*
  - Remove stop words, e.g. *I would like to*
  - Apply threshold to remove "not-important" terms
  - Count occurrences of remaining key terms



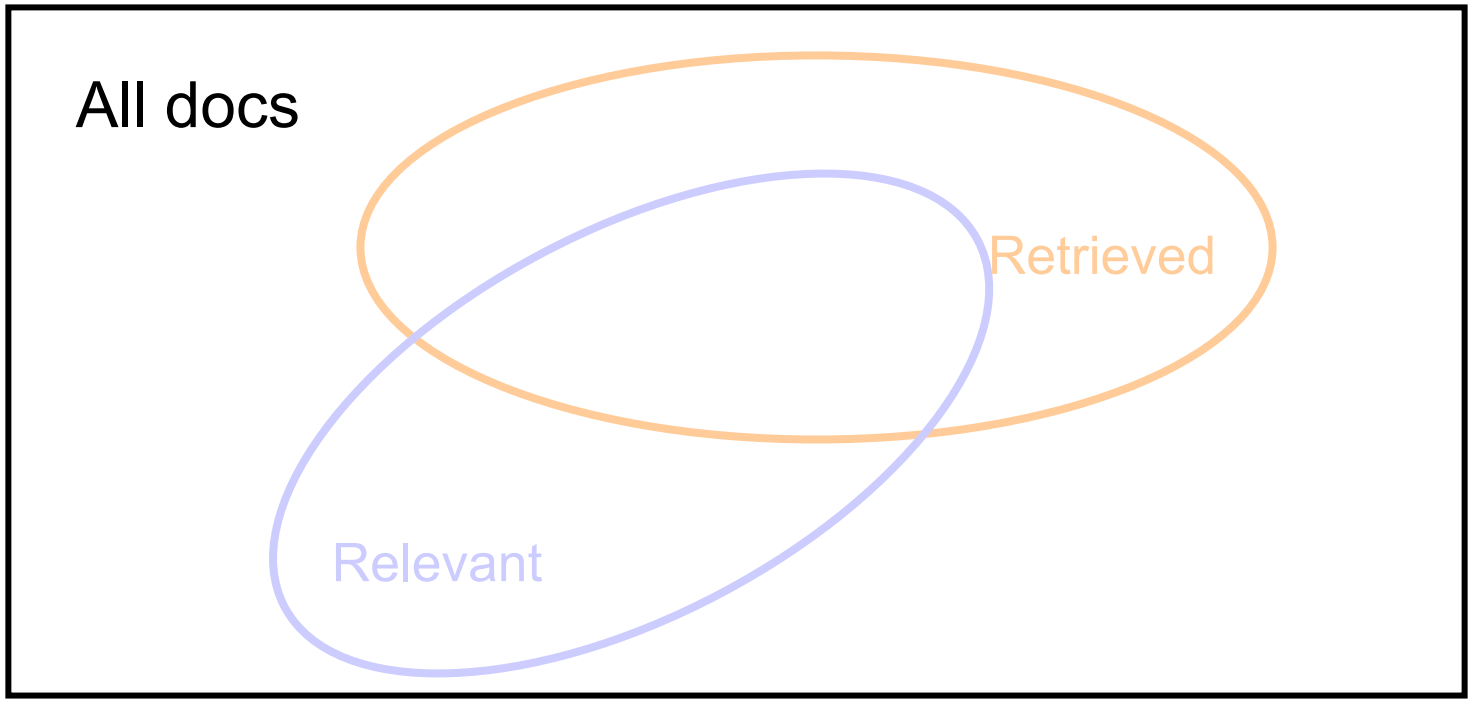
Vector Space Representation



The Vector Space Method

- Term (rows) by document (columns) matrix, based on co-occurrences
- Translation into vectors in a vector space, one vector for each document
- Direction cosine to measure similarity (distance) between vectors (documents)
  - — small angle = large cosine => similar
  - — large angle = small cosine => dissimilar

Standard Evaluation Measures in IR



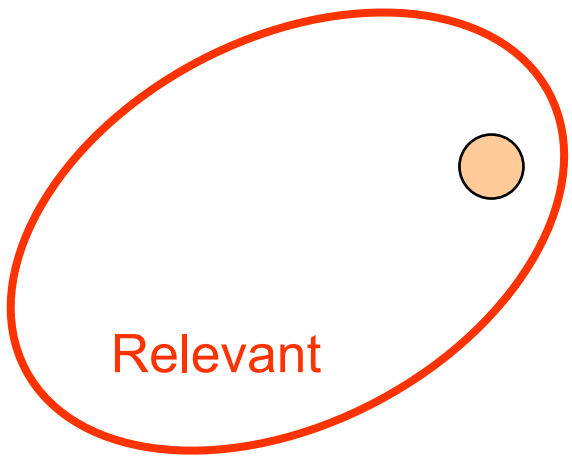
$$\text{Recall} = \frac{|\text{RelRetrieved}|}{|\text{Rel in Collection}|}$$

$$\text{Precision} = \frac{|\text{RelRetrieved}|}{|\text{Retrieved}|}$$

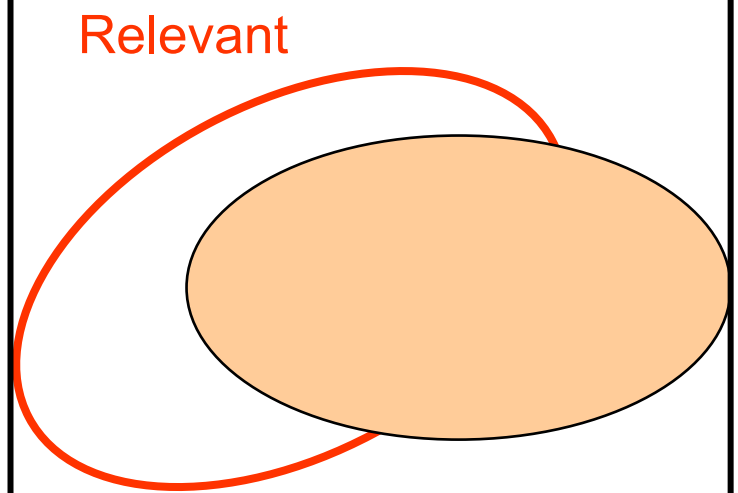
Significance: Precision vs. Recall

Get as much good stuff while at the same time getting as little junk as possible

Very high precision, very low recall

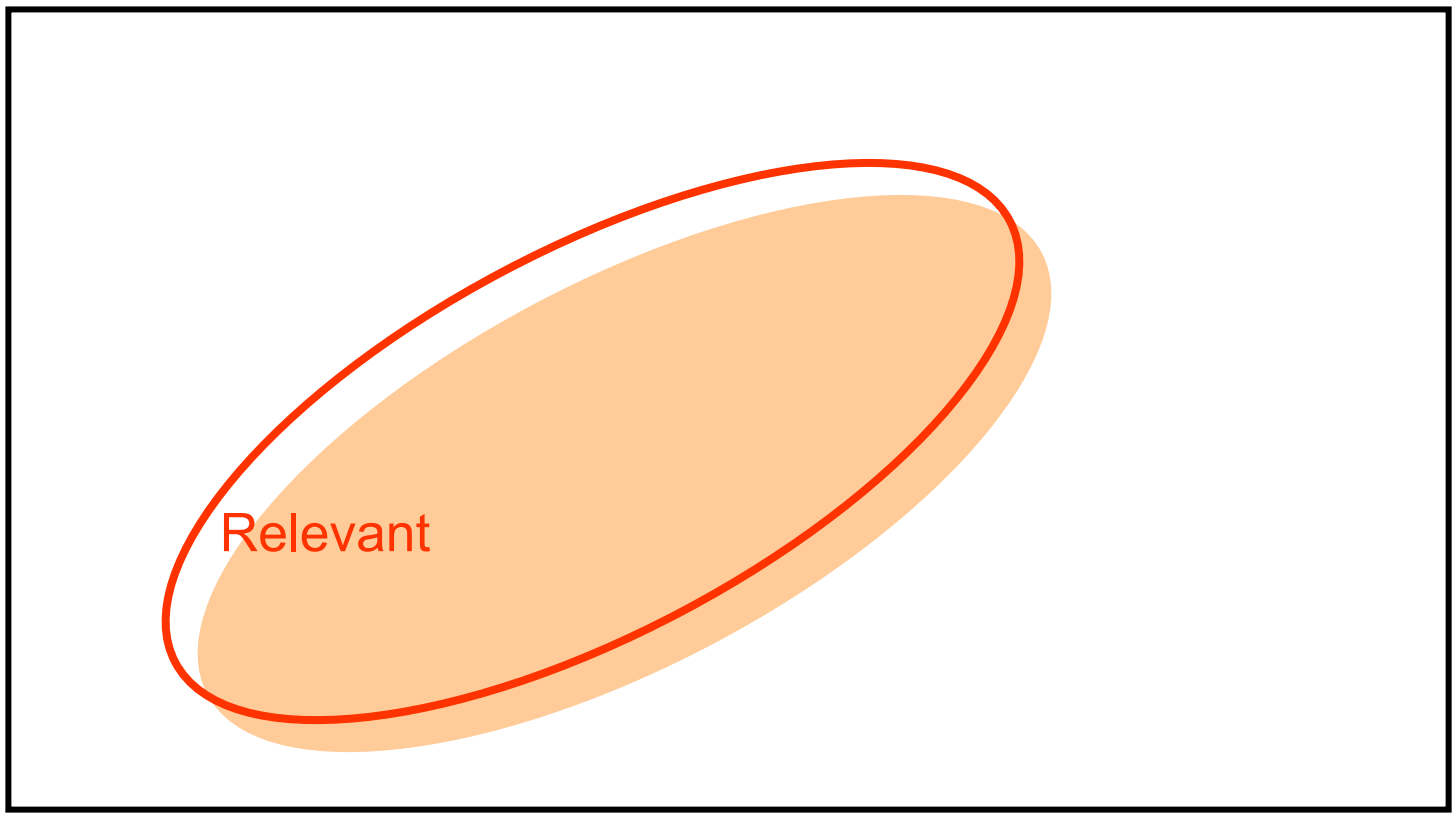


High recall, but low precision



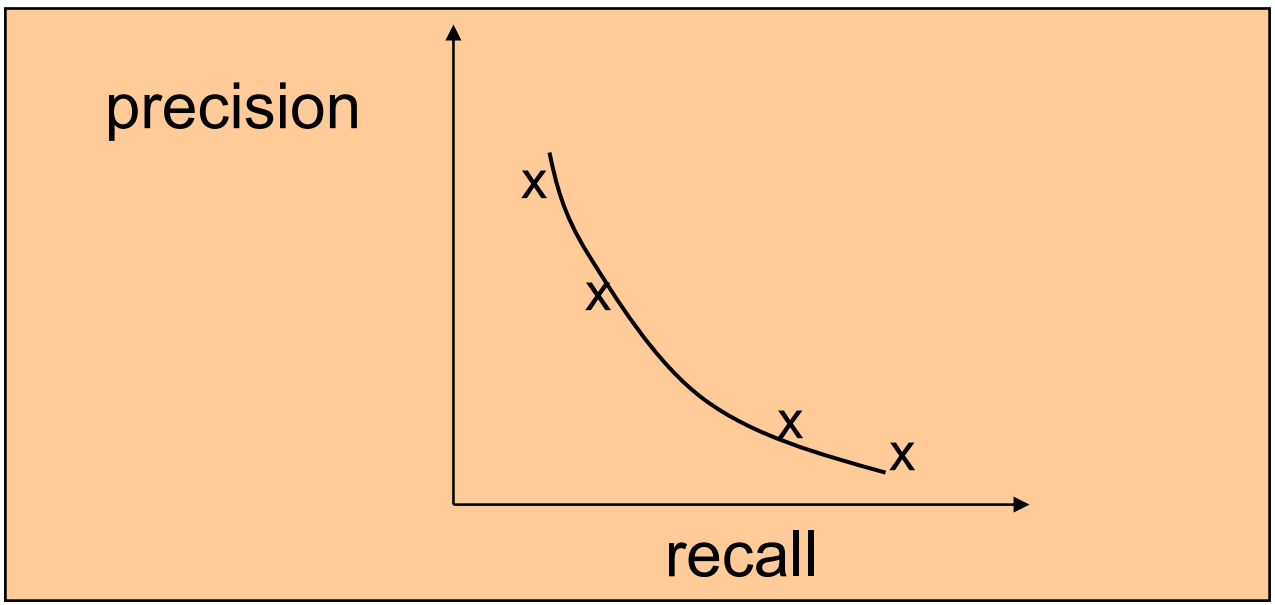
Retrieved vs. Relevant Documents

High precision, high recall (at last!)



Precision/Recall Curves

- There is a tradeoff between precision and recall, so measure precision at different levels of recall (ROC !!)
- Note: this is an AVERAGE over MANY queries
- Also interest in minimizing area under the ROC curve



Synonymy and Polysemy

- Two problems that arose using the vector space model: LSA was proposed to address these two problems
  - synonymy: many ways to refer to the same object, e.g. car and automobile
    - leading to poor recall
  - polysemy: most words have more than one distinct meaning, e.g.model, python, chip
    - leading to poor precision

Some History

- Latent Semantic Indexing was developed at Bellcore (now Telcordia) in 1988. It was patented in 1989
- LSI usually refers to indexing in IR, while LSA refers to everything else
- Papers at http://lsa.colorado.edu/
- More resources at http://lsi.argreenhouse.com/lsi/LSI.html
- Some first papers

1. Dumais, S. T., Furnas, G. W., Landauer, T. K. and Deerwester, S. (1988), "Using latent semantic analysis to improve information retrieval." In Proceedings of CHI'88: Conference on Human Factors in Computing, New York: ACM, 281-285.
2. Deerwester, S., Dumais, S. T., Landauer, T. K., Furnas, G. W. and Harshman, R.A. (1990) "Indexing by latent semantic analysis." Journal of the Society for Information Science, 41(6), 391-407.
3. Foltz, P. W. (1990) "Using Latent Semantic Indexing for Information Filtering". In R. B. Allen (Ed.) Proceedings of the Conference on Office Information Systems, Cambridge, MA, 40-47.

LSA: Four Implementation Steps

- Term by document matrix: tend to be sparse
- Converting matrix entries to weights
- Rank-reduced Singular Value Decomposition (SVD) performed on matrix
  - all but the k highest singular values are set to 0
  - produces k-dimensional approximation of the original matrix (in least-squares sense)
  - this is the “semantic space”
- Compute similarities between entities in semantic space (usually with cosine)

A Small Example

- Technical Memo Titles

c1: *Human* machine *interface* for 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, ordered *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*

A Small Example – 2

	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

r (human.user) = -.38 r (human.minors) = -.29

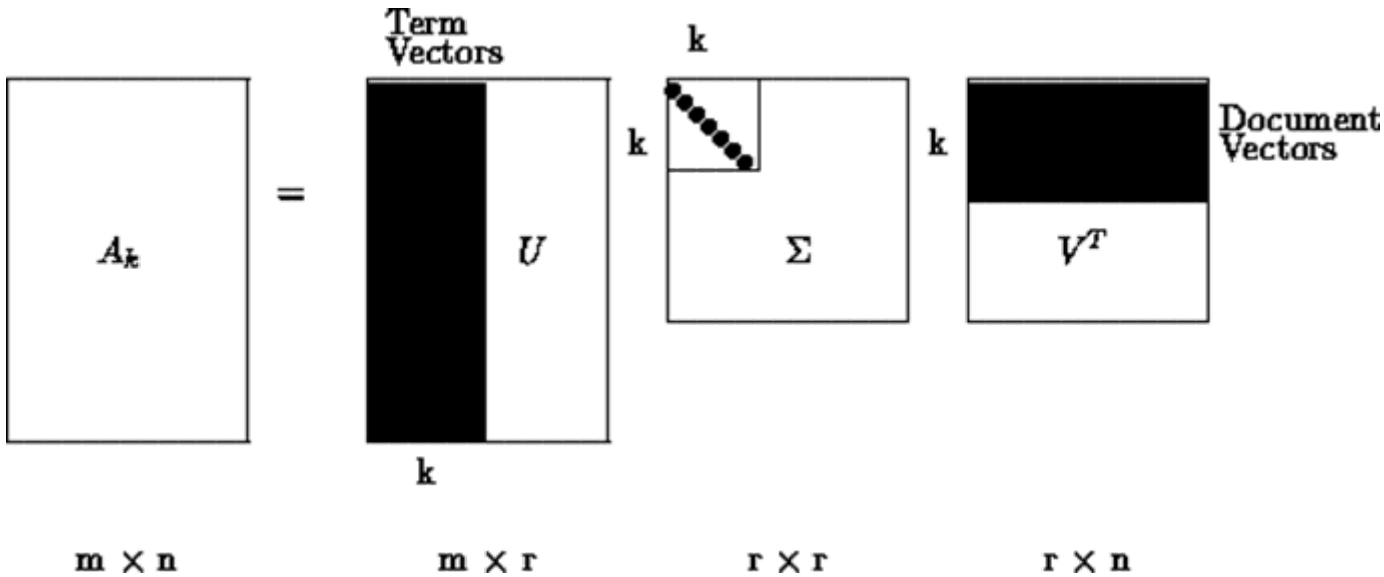
A Small Example - 3

- Singular Value Decomposition

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

- Dimension Reduction

$$\{\sim A\}\sim=\{\sim U\}\{\sim S\}\{\sim V\}^T$$



A Small Example – 4

{U} =

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

A Small Example – 5

$\{S\} =$

3.34

2.54

2.35

1.64

1.50

1.31

0.85

0.56

0.36

A Small Example – 6

{V} =

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

A Small Example – 7

	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

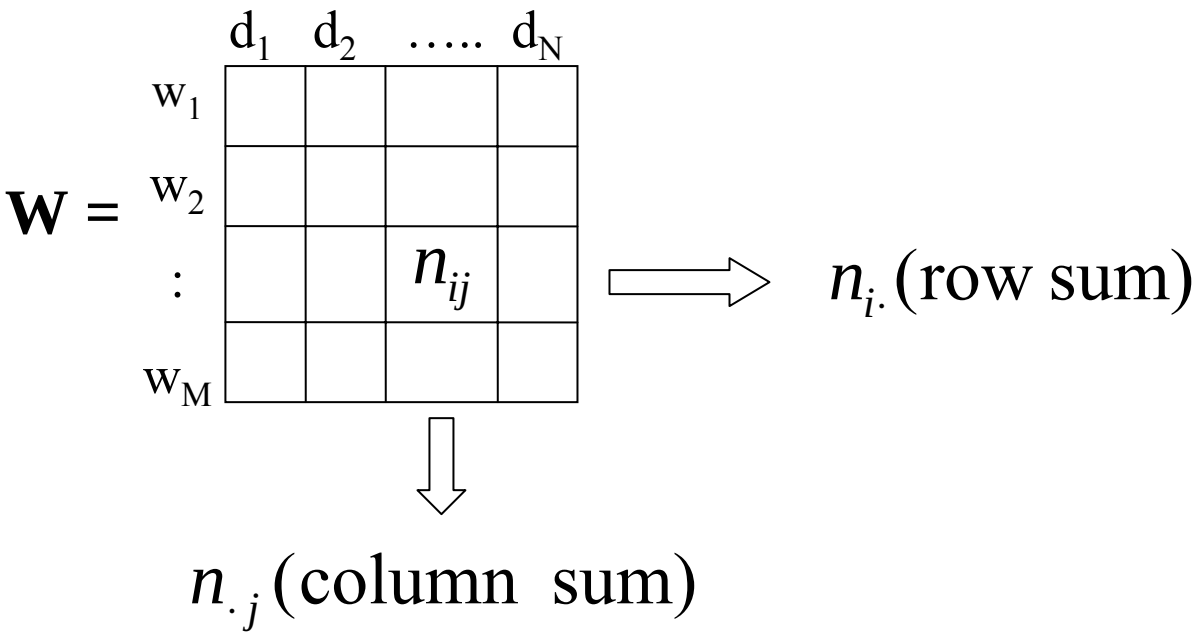
r (human.user) = .94 r (human.minors) = -.83

Caveats

- LSA is a “bag-of-words” technique
  - Deerwester, et al, “Indexing by latent semantic analysis”, Journal of American Information Science, 41-6, 391-407, 19990
- Blind to word-order, or syntax in text
  - Global semantics vs. local syntax information
- Future directions
  - Add syntactic information to LSA ?
  - Integrate local syntax, LSA semantics and global pragmatics

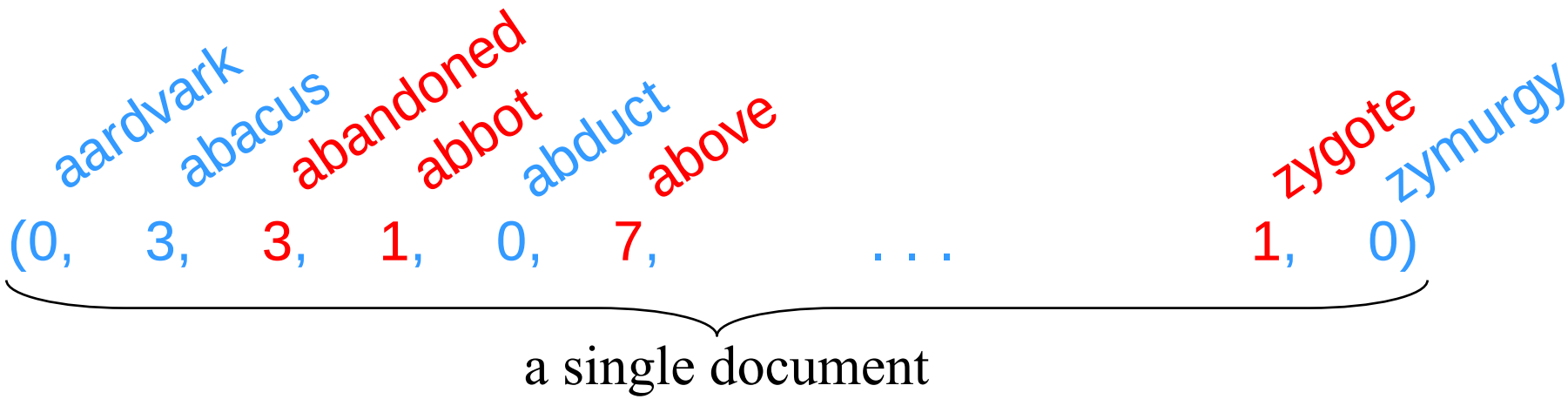
Word-Document Co-Occurrence

- Given - *N* documents, vocabulary size *M*
- Generate a word-documents co-occurrence matrix **W**



LSA Count in the Column Vector

- A trick from Information Retrieval
  - Each **document** (paragraph or sentence) in the training document corpus is a length-*M* vector



LSA Mathematical Framework

- LSA Matrix (also known as Routing Matrix) *C*

$$c_{ij} = (1-\varepsilon_i) n_{ij} / n_{\cdot j} \text{ (scaling and normalization)}$$

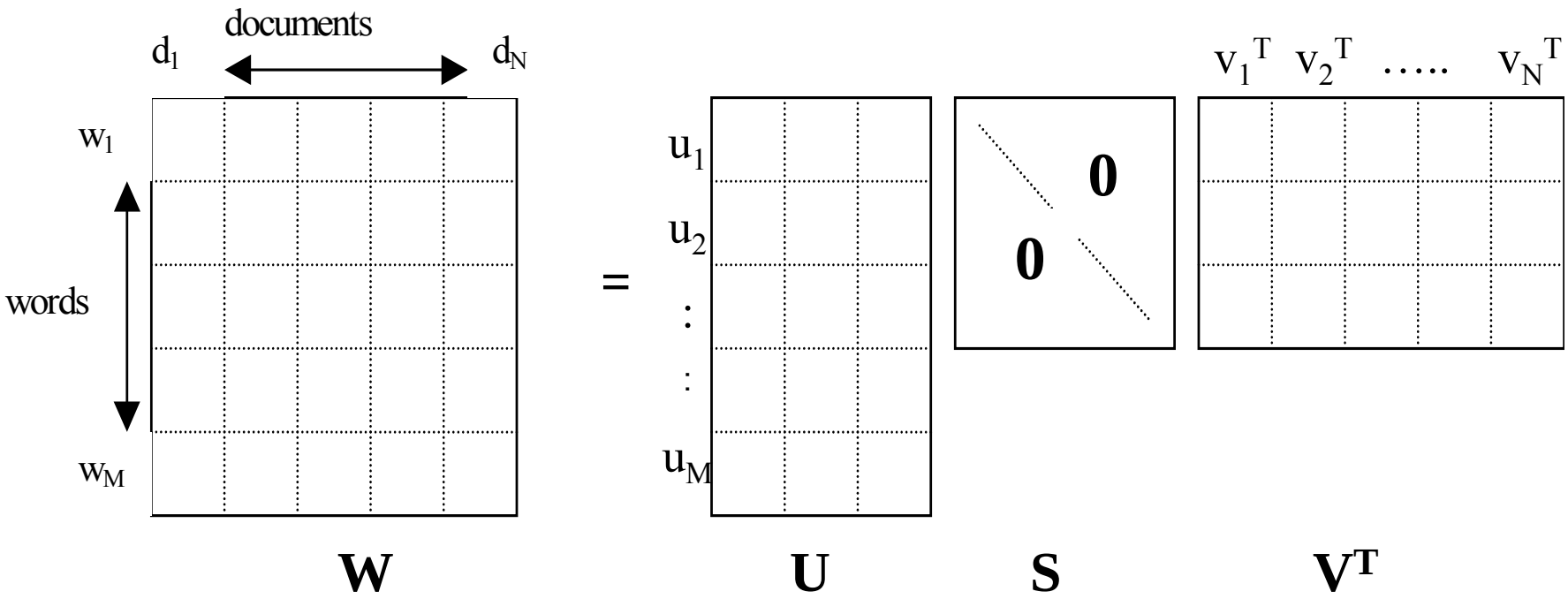
  - number of times word w_i occurs in A_j : n_{ij}
  - total number of words present in A_j : $n_{\cdot j}$ (column sum)
  - total number of w_i occurs in *A* : $n_{i\cdot}$ (row sum)
  - "indexing" power of w_i in corpus *A* : $\eta_i = 1-\varepsilon_i$
  - normalized entropy:

$$\varepsilon_i = -\frac{1}{\log N}\sum_{j=1}^{N}\frac{n_{ij}}{n_{i\cdot}}\log\frac{n_{ij}}{n_{i\cdot}} \qquad 0 \le \varepsilon_i \le 1$$

$$\begin{cases} \varepsilon_i = 0 & \text{if } n_{ij} = n_{i\cdot} \quad \text{maximum indexing power} \\ \varepsilon_i = 1 & \text{if } n_{ij} = \frac{n_{i\cdot}}{N} \quad \text{no power (equally probable)} \end{cases}$$

Singular Value Decomposition (SVD)

- Two-factor analysis of raw observations in engineering

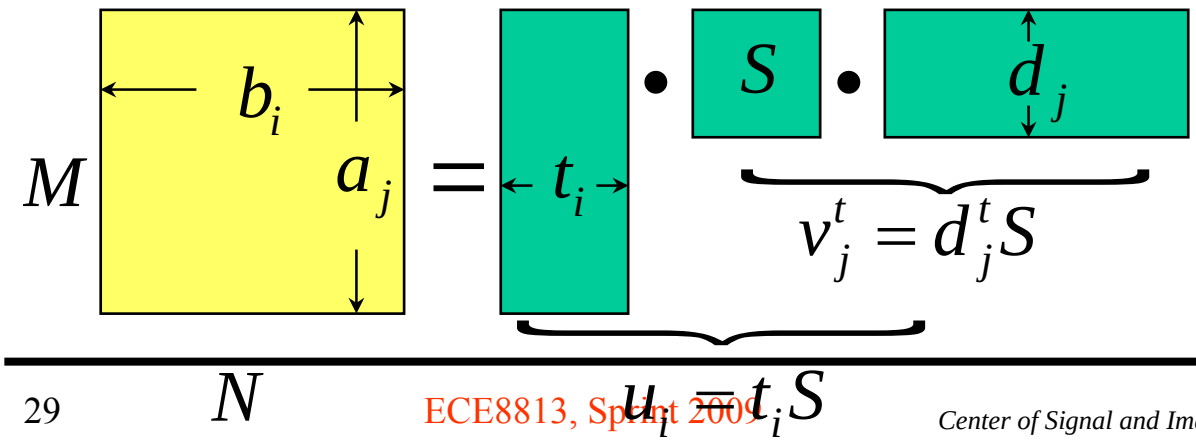


SVD Approximation

- Dimensionality reduction
  - Best rank-*R* approximation
  - Optimal energy preservation
  - Captures major structural associations between words and documents
  - Removes ‘noisy’ observations

LSA Feature Space

- **Mapping into latent semantic space *S***
  - each document vector a_j (*N* column vectors of matrix *W*) is mapped to an (1xR)-vector $v_j^t = d_j^t S$
  - each term vector b_i (*M* row vectors of matrix *W*) is mapped to an (1xR)-vector $u_i = t_i S$
  - each query vector (a new Mx1 vector) is mapped to an (1xR)-vector through the pseudo-document vector
  - closeness in the *S* space is much easier measured for both document-document and term-term comparisons



$W = USV^t$ (SVD)

$M \approx 10{,}000$

$N \approx 100{,}000$

$R \approx 150 - 200$

Semantic Similarity Measure

- To find similarity between two documents, project them in LS space
- Then calculate the cosine measure between their projection
- With this measure, various problems can be addressed e.g., natural language understanding, cognitive modeling etc.

Confidence Scoring

- Inner Product: $s(x,y) = x \bullet y^t$
- Cosine: $s(x,y) = x \bullet y^t \Big/ |x||y|$ or $\cos^{-1}[s(x,y)]$
- Confidence Scoring: Sigmoid function fitting

$$Conf\ (s;\alpha,\beta) = [1 + e^{-(\alpha s+\beta)}]^{-1}$$

- Other Scores
  - Euclidean, Manhattan, etc.
- Generalized Scores
  - between any two vectors: $s(x,y) = f(x,y;\Gamma)$

Similarity in LSA

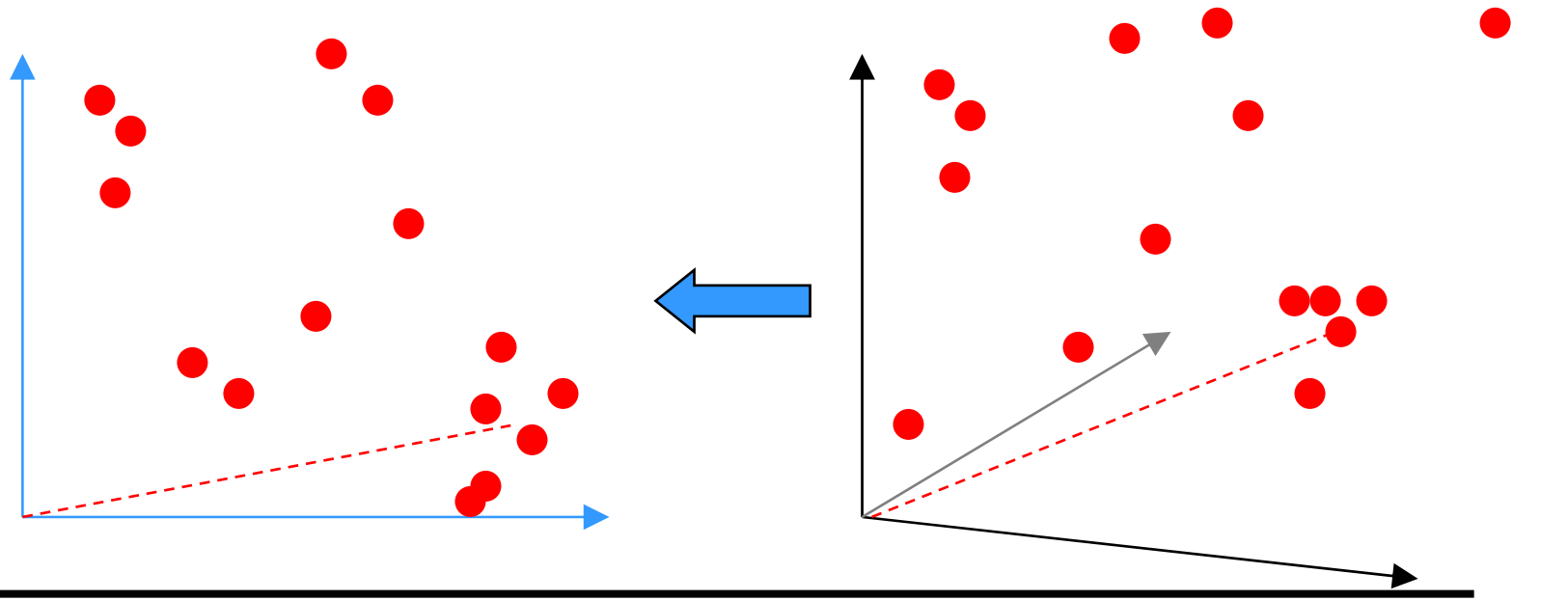
- The vector of a passage is the vector sum of the vectors standing for the words it contains
- Similarity of any two words or two passages is computed as the cosine between them in the semantic space:
  - Identical meaning: value of cosine = 1
  - Unrelated meaning: value of cosine = 0
  - Opposite meaning: value of cosine = -1
- Number of dimensions used is an important issue
  - Small dimensions (small singular values) represent local unique components
  - Large dimensions capture similarities and differences

A Geometric Illustration

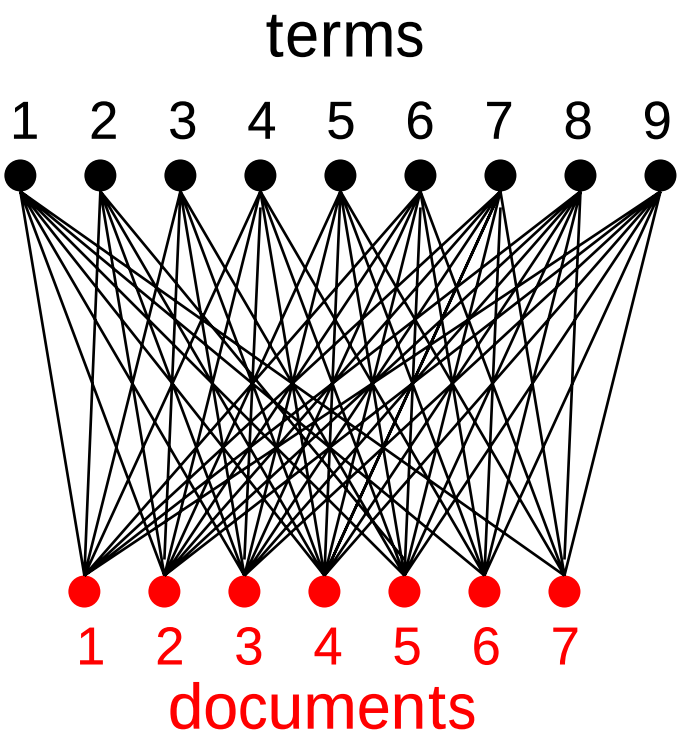
- Each **document** in corpus is a length-*M* vector, plot all documents in corpus
- Reduced-dimension plot is a perspective drawing of true plot, projecting it onto a few axes (solved by SVD for the best set of axes), ignoring noisy axes, approximating vectors through linear combinations and topology preservation

Reduced-dimensionality plot

True plot in *M* dimensions



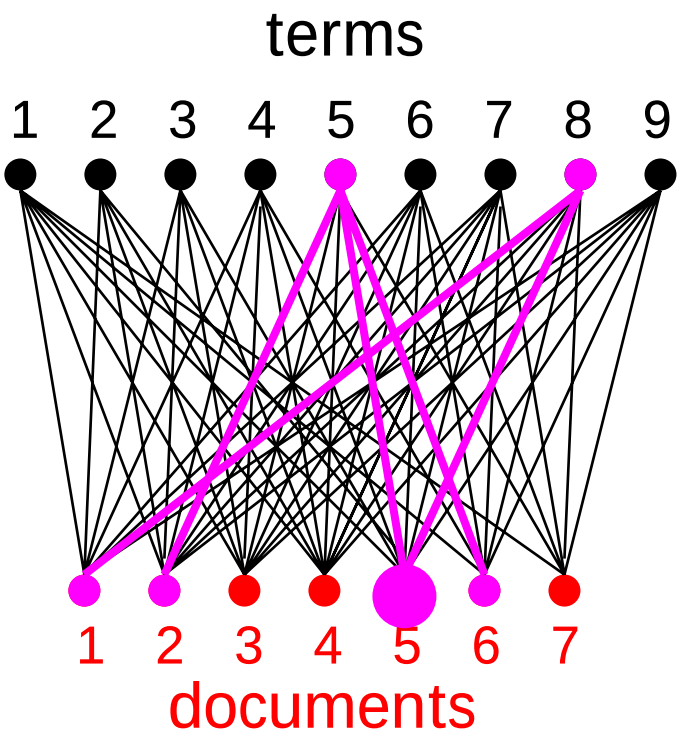
Another Perspective (Similar to ANN)



- **Matrix** of strengths (how strong is each term in each document?)
- Each connection has a weight given by the matrix.

ANN-Based Query

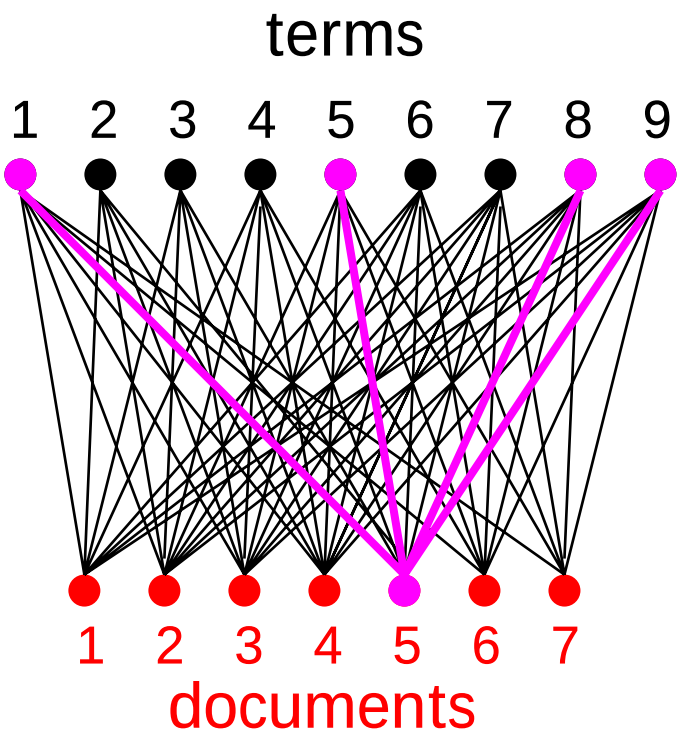
- Which documents are terms 5 and 8 strong in?



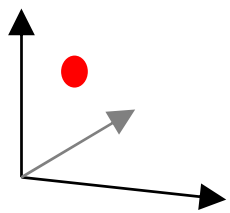
This answers a query consisting of terms 5 and 8!
- really just matrix multiplication: term vector (query) x strength matrix = doc vector.

ANN-Based Reverse Query

- Conversely, what terms are strong in document 5?



gives doc 5's coordinates!



Singular Value Decomposition

M × N, Term × Document matrix (M >> N)

$\mathbf{W} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_N]$ and $\mathbf{d}_i = [d_{i1}, d_{i2}, \ldots, d_{iM}]^T$

Consider linear combination of terms

$$u_1 d_{i1} + u_2 d_{i2} + \ldots + u_M d_{iM} = \mathbf{u}^T \mathbf{d}_i$$

which maximizes

$(\mathbf{u}^T\mathbf{W})(\mathbf{W}^T\mathbf{u}) = \mathbf{u}^T\mathbf{W}\mathbf{W}^T\mathbf{u}$

Subject to $\mathbf{u}^T\mathbf{u} = 1$

Singular Value Decomposition (Cont.)

Maximize $\mathbf{u}^T\mathbf{W}\mathbf{W}^T\mathbf{u}$ s.t. $\mathbf{u}^T\mathbf{u} = 1$

Construct Langrangian $\mathbf{u}^T\mathbf{W}\mathbf{W}^T\mathbf{u} - \lambda\mathbf{u}^T\mathbf{u}$

Vector of partial derivatives set to zero

$$\mathbf{W}\mathbf{W}^T\mathbf{u} - \lambda\mathbf{u} = (\mathbf{W}\mathbf{W}^T - \lambda\mathbf{I})\,\mathbf{u} = 0$$

As $\mathbf{u} \neq \mathbf{0}$ then $\mathbf{W}\mathbf{W}^T - \lambda\mathbf{I}$ must be singular i.e

$$|\mathbf{W}\mathbf{W}^T - \lambda\mathbf{I}| = 0$$

This is a polynomial in λ of degree M with *characteristic* roots – called the eigenvalues

(German eigen = *own*, *unique to*, *particular to*)

Singular Value Decomposition (Cont.)

- The first root is called the prinicipal eigenvalue which has an associated orthonormal ($\mathbf{u}^T\mathbf{u} = 1$) *eigenvector* $\mathbf{u}$
- Subsequent roots are ordered such that $\lambda_1 > \lambda_2 > \ldots > \lambda_M$ with rank(W) non-zero values.
- Eigenvectors form an orthonormal basis i.e. $\mathbf{u}_i^T\mathbf{u}_j = \delta_{ij}$
- The eigenvalue decomposition of $\mathbf{W}\mathbf{W}^T = \mathbf{U}\boldsymbol{\Sigma}\mathbf{U}^T$

where $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M]$ and $\boldsymbol{\Sigma} = \text{diag}[\lambda_1, \lambda_2, \ldots, \lambda_M]$

- Similarly the eigenvalue decomposition of $\mathbf{W}^T\mathbf{W} = \mathbf{V}\boldsymbol{\Sigma}\mathbf{V}^T$
- The SVD is closely related to the above $\mathbf{W}=\mathbf{U}\,\boldsymbol{\Sigma}^{1/2}\,\mathbf{V}^T$
- The left eigenvectors **U,** right eigenvectors **V,** singular values = square root of eigenvalues of matrix **Σ,** the singular matrix $\mathbf{S}=\boldsymbol{\Sigma}^{1/2}$

SVD Properties

$\mathbf{W}=\mathbf{U}\,\mathbf{S}\,\mathbf{V}^T = \sum_{i=1..N} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$

$\mathbf{S}_K = \sum_{i=1..K} \sigma_i \mathbf{u}_i \mathbf{v}_i^T = \mathbf{U}_K\,\mathbf{S}_K\,\mathbf{V}_K^T$ and K<N (K<M)

$\mathbf{U}_K^T\mathbf{U}_K = \mathbf{I}_K = \mathbf{V}_K^T\mathbf{V}_K$

- Then $\mathbf{S}_K$ is best rank-K approximation to $\mathbf{S}$
- K-dim orthonormal projections $\mathbf{S}^{-1}{}_K\,\mathbf{U}_K^T\,\mathbf{S}_K = \mathbf{V}_K^T$ preserve the maximum amount of variability
- If we assume that columns of $\mathbf{S}$ are multivariate Gaussian then $\mathbf{V}$ defines principal axes of ellipse of constant variance λ_i in the original space

SVD Properties (Cont.)

- There is an implicit assumption that the observed data distribution is multivariate Gaussian
- Can consider as a probabilistic generative model – latent variables are Gaussian – sub-optimal in likelihood terms for non-Gaussian distribution
- Employed in signal processing for noise filtering – dominant subspace contains majority of information bearing part of signal
- Similar rationale when applying SVD to LSI

Computing SVD

- A numerical approach
  - Random initialization of vector $\mathbf{u}^0$

    Set $\quad \mathbf{u}^{1u} = \mathbf{W}\mathbf{W}^T\mathbf{u}^0$ and $\mathbf{u}^1 = \mathbf{u}^{1u} / \sqrt{(\mathbf{u}^{1u})^T \mathbf{u}^{1u}}$

    then $\quad \mathbf{u}^{2u} = \mathbf{W}\mathbf{W}^T\mathbf{u}^1$ and $\mathbf{u}^2 = \mathbf{u}^{2u} / \sqrt{(\mathbf{u}^{2u})^T \mathbf{u}^{2u}}$

    then $\quad \mathbf{u}^{iu} = \mathbf{W}\mathbf{W}^T\mathbf{u}^{i-1}$ and $\mathbf{u}^i = \mathbf{u}^{iu} / \sqrt{(\mathbf{u}^{iu})^T \mathbf{u}^{iu}}$

    As $i \rightarrow \infty$, $\mathbf{u}^i \rightarrow \mathbf{u}_1$, $\sqrt{(\mathbf{u}^{iu})^T \mathbf{u}^{iu}} \rightarrow \lambda_1$
- Subsequent eigenvalues use deflation

$$\mathbf{u}^{1u} = (\mathbf{W}\mathbf{W}^T - \lambda_1 \mathbf{u}_1 \mathbf{u}_1^T)\mathbf{u}^0$$

- Note for term document matrix computation of $\mathbf{u}_1$
- Inexpensive – subsequent eigenvalues require matrix-vector operations on dense matrix

LSA Applications (I)

- Information retrieval and text categorization
  - Later lectures
  - TC-based multimedia applications with tokenization, e.g. language identification, image annotation, audio fingerprinting
- Natural language understanding and cognitive modeling
  - Automatic evaluation of students' answers (AutoTutor)
  - Prediction of how much an individual student will learn from a particular instructional text
  - Based on the similarity of an essay on a topic to a given text, Optimal text can be chosen
- Long-spam language modeling
  - Bellagarda, *Proc. IEEE*, August 2000.
  - Semantic classification
  - Semantically large span (N-gram + LSA) models

LSA Applications (II)

- Essay grading
  - LSA is trained on a large sample of text from the same domain as the topic of the essay
  - Each essay is compared to a large set of essays scored by experts and a subset of the most similar identified by LSA
  - The target essay is assigned a score consisting of a weighted
- Cross-language retrieval
  - Retrieval when queries and documents are in different languages
  - Overlapping set of documents (does not have to be large)
  - Rotation of the two semantic spaces, so there is correspondence on the overlapping set
  - Second language learning

Information Retrieval

- **IR:** "concept matching" vs "lexical matching" : relevant documents are associated with similar "concepts", but may not include exactly the same words
  - example approach: treating the query as a new document (by "folding-in"), and evaluating its "similarity" with all possible documents
- **Fold-in**
  - consider a new document outside of the training corpus T, but with similar language patterns or "concepts"
  - construct a new column d_p ,p>N, with respect to the M words
  - assuming U and S remain unchanged

    $d_p = U S v_p^T$ (just as a column in $W = USV^T$)

    $\underline{v}_p = v_p S = d_p^T U$ as an R-dim representation of the new document

    (i.e. obtaining the projection of d_p on the basis e_i of U by inner product)

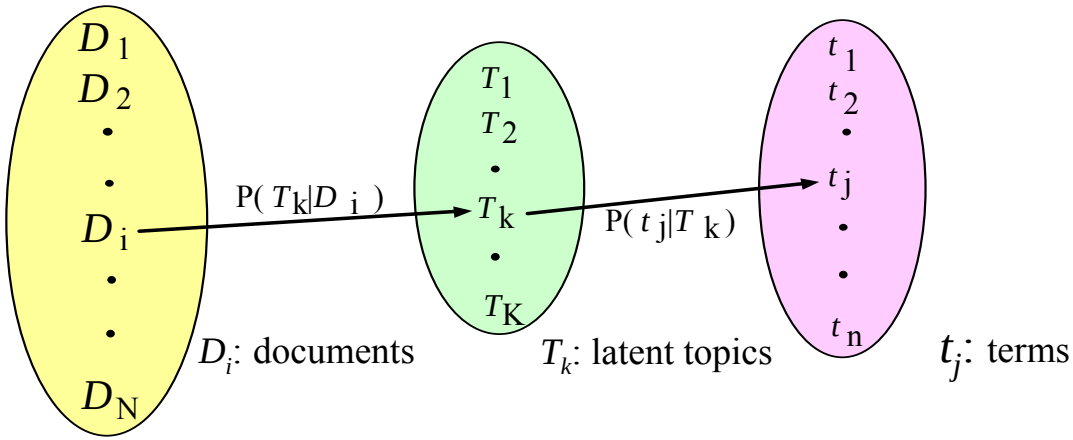
More on Latent Semantic Analysis

- Word usage defined by term and document co-occurrence – matrix structure
- Latent structure / semantics in word usage
- Clustering documents or words – no shared space
- Two mode factor analysis (SVD) – dyadic decomposition into 'latent semantic' factor space
- Cubic Computational Scaling – reasonable

Probabilistic Views on LSA

- PLSA (Probabilistic LSA)
- Factor Analytic Model
- Generative Model Representation
- Alternate Basis to the Principal Directions

Probabilistic LSA (PLSA)



- Exactly the same as LSA, using a set of latent topics $\{T_1, T_2, \cdots T_K\}$ to construct a new relationship between the documents and terms, but with a probabilistic framework

$$P(t_j \mid d_i) = \sum_{k=1}^{K} P(t_j \mid T_k) P(T_k \mid d_i)$$

- Trained with EM by maximizing the total log likelihood

$$L_T = \sum_{i=1}^{N} \sum_{j=1}^{M} c(t_j, d_i) \log P(t_j \mid d_i)$$

- $c(t_j, d_i)$: frequency count of term t_j in the document d_i

Factor Analytic Model

$\mathbf{d} = \mathbf{A}\mathbf{f} + \mathbf{n}$

$p(\mathbf{d}) = \sum_f p(\mathbf{d}|f)p(f)$

– This probabilistic representation underlies LSA where prior, p(f), with f denoting semantic factor, and likelihood, p(d|f), are both multivariate Gaussian

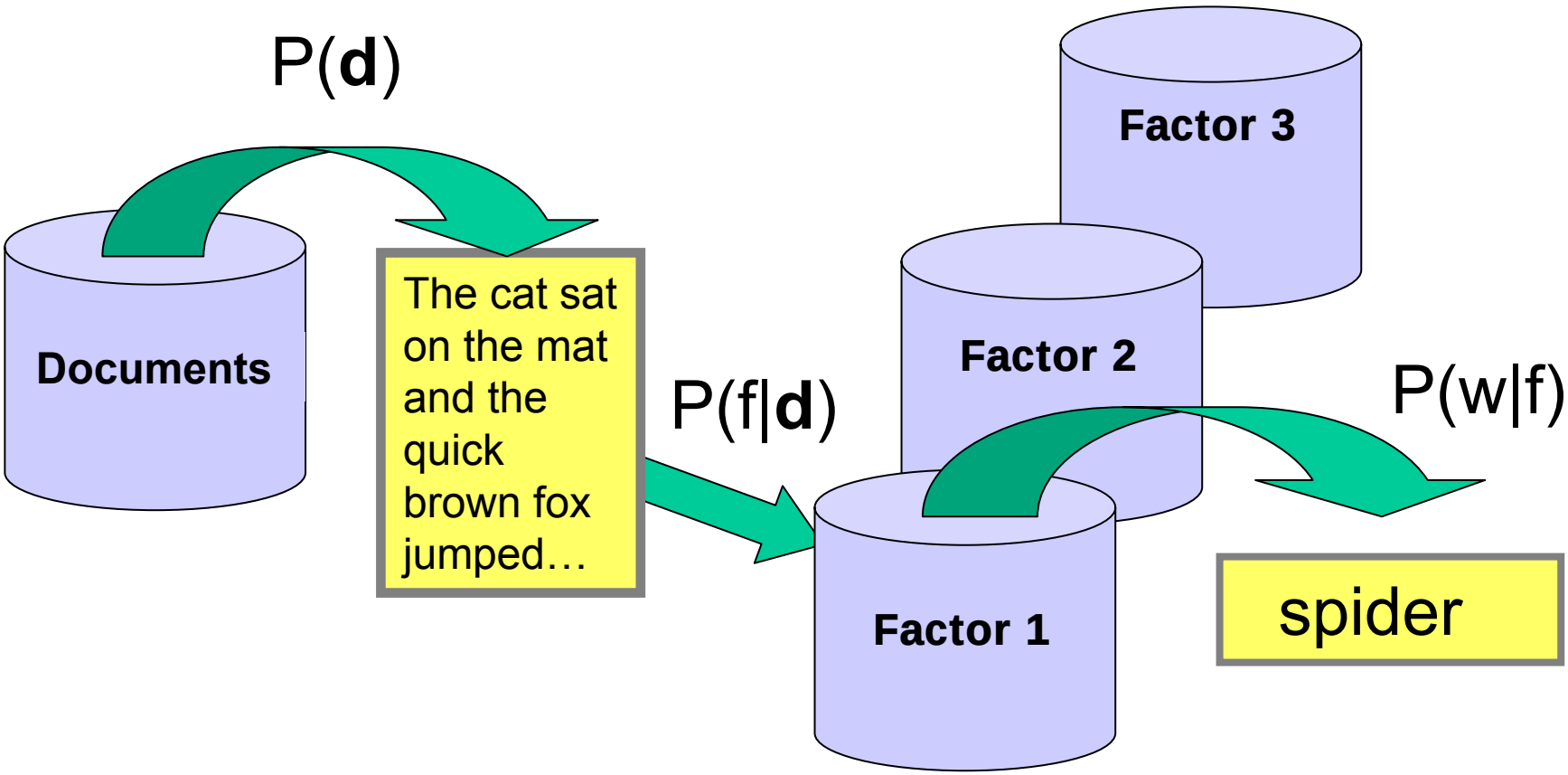
Generative Model Representation

– Generate a document **d** with probability p(**d**)

– Having observed **d** generate a semantic factor with probability p(f|**d**)

– Having observed a semantic factor generate a word with probability p(w|f)

Generative Model: Conceptual View



Generative Model (Cont.)

– Model representation as joint probability

$$p(\mathbf{d},w) = p(\mathbf{d})p(w|\mathbf{d})$$
$$= p(\mathbf{d})\sum_f p(w|f)p(f|\mathbf{d})$$

w and **d** conditionally independent given f

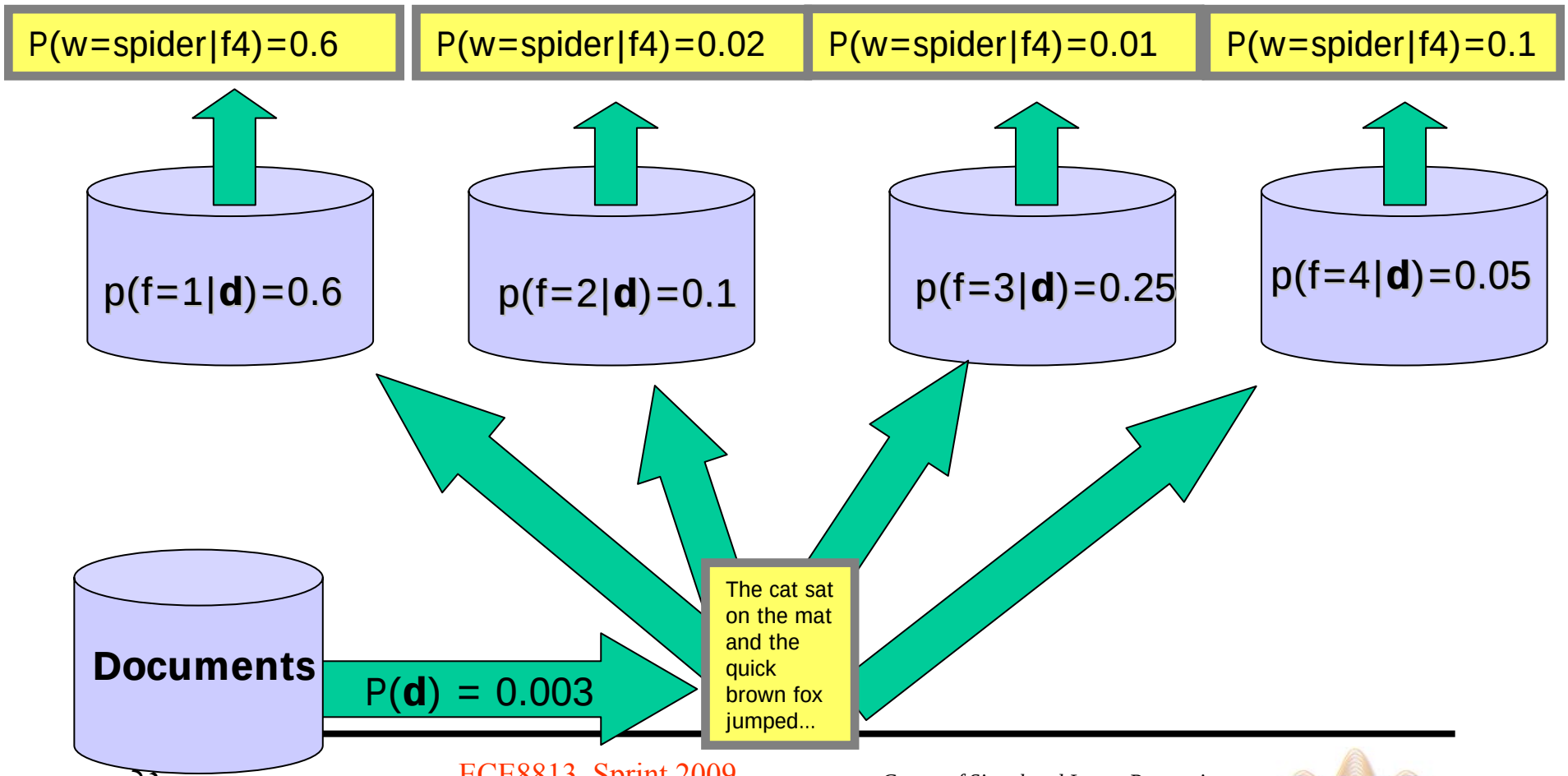
– $p(\mathbf{d},w) = \sum_f p(w|f)p(f)p(\mathbf{d}|f)$

– Note the similarity to PLSA

– Note similarity with $\mathbf{S}_K = \sum_{i=1..K} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$

Generative Model (Cont.)

$$p(\mathbf{d},w) = p(d)\sum_f p(w|f)p(f|\mathbf{d}) = 0.001$$



Generative Model (Cont.)

- Distributions of p(f|**d**) and p(w|f) are multinomial – counts in successive trials
- More appropriate than Gaussian
- Note that Term × Document matrix is a sample from the true distribution $p_t(\mathbf{d}, w)$
- $\sum_{ij} D(i,j) \log p(\mathbf{d}_j, w_i)$: cross-entropy between model and realization – maximize likelihood that the model $p(\mathbf{d}_j, w_i)$ generated the realization **D** – subject to conditions on p(f|**d**) and p(w|f)

Generative Model (Cont.)

– Estimation of p(f|**d**) and p(w|f) requires use of a standard EM algorithm.

– Expectation Maximization

- General iterative method for ML parameter estimation
- Ideal for 'missing variable' problems

– Estimate p(f|**d**,w) using current estimates of p(w|f) and p(f|**d**)

– Estimate new values of p(w|f) and p(f|**d**) using current estimate of p(f|**d**,w)

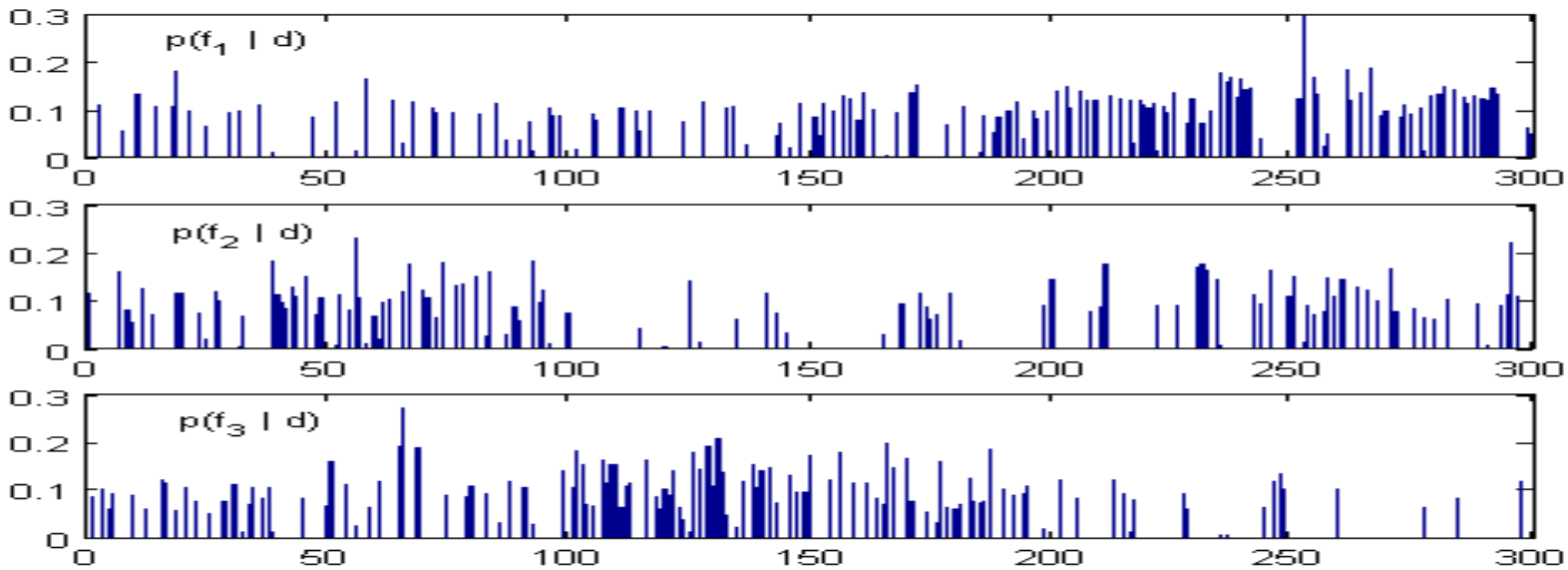
Generative Model (Cont.)

- Once parameters estimated
  - p(f|**d**) gives posterior probability that Semantic factor ‘f’ is associated with **d**
  - p(w|f) gives the probability of word ‘w’ being generated from semantic factor ‘f’
- Nice clear interpretation unlike **U** and **V** terms in SVD
- ‘Sparse’ representation: unlike SVD

Generative Model (Cont.)

– Take the toy collection generated – estimate p(f|**d**) and p(w|f)

– Graphical representation of p(f|**d**)



Alternate Basis to the Principal Directions

– Similarity between query and documents can be assessed in 'factor' space through LSA

– Similarity score = $\sum_f p(f|\mathbf{q})\, p(f|\mathbf{D})$ averaged product of query and doc posterior probabilities over all 'factors' – latent space

– Alternately note that **D** and **q** are sample instances from an unknown distribution

– All probabilities of word counts – estimated from **D** 'noisy'

–Employ $p(\mathbf{d}_j, w_i)$ as 'smoothed' version of tf and use 'cosine' measure $\sum_i p(\mathbf{D}, w_i) \times \mathbf{q}_i$ 'query expansion'

Alternate Basis to the Principal Directions

– Both forms of matching shown to improve on LSA

– Elegant statistically principled approach – can employ (in theory) Bayesian model assessment techniques

– Likelihood nonlinear function of parameters p(f|**d**) and p(w|f): huge parameter space, small number of relative samples, high bias and variance expected

– Assessment of correlation with likelihood and precision and recall yet to be studied in depth

Summary on Alternative LSA

– SVD defined basis provide P/R improvements over term matching
  - Interpretation difficult
  - Optimal dimension – open question
  - Variable performance on LARGE coll's
  - Supercomputing muscle required

– Probabilistic approaches provide improvements over SVD
  - Clear interpretation of decomposition
  - Optimal dimension – open question
  - High variability of results due to nonlinear optimization over HUGE parameter space

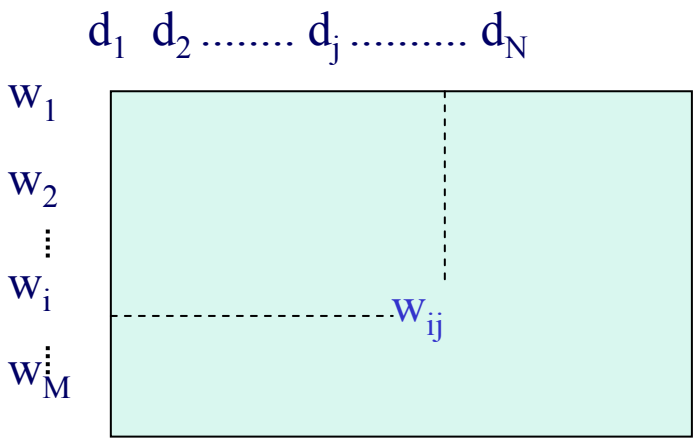
– Improvements marginal in relation to cost

LSA Summary

- SVD Algorithm complexity O(n^2k^3)
  - n = number of terms
  - k = dimension in semantic space (typically small ~50 to 350)
  - for stable document collection, only have to run once
  - dynamic document collections: might need to rerun SVD, but can also "fold in" new documents
- Finding optimal dimension for semantic space
  - precision-recall improve as dimension is increased until hits optimal, then slowly decreases until it hits standard vector model
  - run SVD once with big dimension, say k = 1000
  - in many tasks 150-350 works well, still room for research
- SVD assumes normally distributed data
  - term occurrence is not normally distributed
  - matrix entries are weights, not counts, which may be normally distributed even when counts are not

Word-Document Matrix

- **Vocabulary V of size M and Corpus T of size N**
  - $V=\{w_1,w_2,...w_i,..w_M\}$, w_i: the i-th word ,e.g. $M=2\times10^4$
  - $T=\{d_1,d_2,...d_j,..d_N\}$, d_j: the j-th document ,e.g. $N=10^5$
  - c_{ij}: number of times w_i occurs in d_j
  - n_j: total number of words present in d_j
  - $t_i = \Sigma_j c_{ij}$: total number of times w_i occurs in T



Matrix Representation

- **Word-Document Matrix** $W = [w_{ij}]$
  - each row: a *N*-dim "feature vector" for w_i wrt all documents
  - each column: a *M*-dim "feature vector" for d_j wrt all words

$$\Rightarrow \varepsilon_i = -\frac{1}{\log N}\sum_{j=1}^{N}\left(\frac{c_{ij}}{t_i}\right)\log\left(\frac{c_{ij}}{t_i}\right),$$

normalized entropy (indexing power) of w_i in T

$$0 \le \varepsilon_i \le 1 \quad , \quad \varepsilon_i = 0 \quad \text{if } c_{ij} = t_i \text{ for some j and } c_{ij} = 0 \text{ for other j}$$

$$\varepsilon_i = 1 \quad \text{if } c_{ij} = t_i/\mathrm{N} \text{ for all j}$$

$$w_{ij} = (1-\varepsilon_i)\frac{c_{ij}}{n_j} \ , \text{ word frequencies in doucments,}$$

but normalized with document length and word entropy

Row Dimensionality Reduction

$$WW^T = \bar{U}\bar{S}_1^2\bar{U}^T$$

$$\bar{U} = [e_1, e_2, \ldots e_M], \bar{S}_1^2 = [s_i^2]_{M\times M},\ s_i^2 : \text{eigenvalues of } WW^T, s_i^2 \geq s_{i+1}^2$$

(i,j) element of WW^T : inner product of i-th and j-th rows of W

"similarity" between w_i and w_j

$$WW^T = \sum_i s_i^2 e_i e_i^T,\ e_i : \text{orthonormal eigenvectors}, \bar{U}^T\bar{U} = I_M$$

s_i^2 : weights (significance of the "component matrices" $e_i e_i^T$)

$$W_{M\times N} W_{N\times M}^T \approx U_{M\times R} S_{R\times R}^2 U_{R\times M}^T, U_{M\times R} = [e_1, e_2, \ldots . e_R]$$

R "concepts" or "latent semantic concepts"

- dimensionality reduction: selection of R largest eigenvalues (R=800 for example)

Column Dimensionality Reduction

$$W^T W = \overline{V}\,\overline{S_2^2}\,\overline{V}^T \qquad \overline{V} = [e'_1, e'_2, \ldots e'_N],\ s_i^2 : \text{eigenvalues of } W'W,$$

$$W^T W = \sum_i s_i^2 e'_i e'^T_i, \qquad e'_i : \text{orthonormal eigenvectors},\ \overline{V}^T \overline{V} = I_N$$

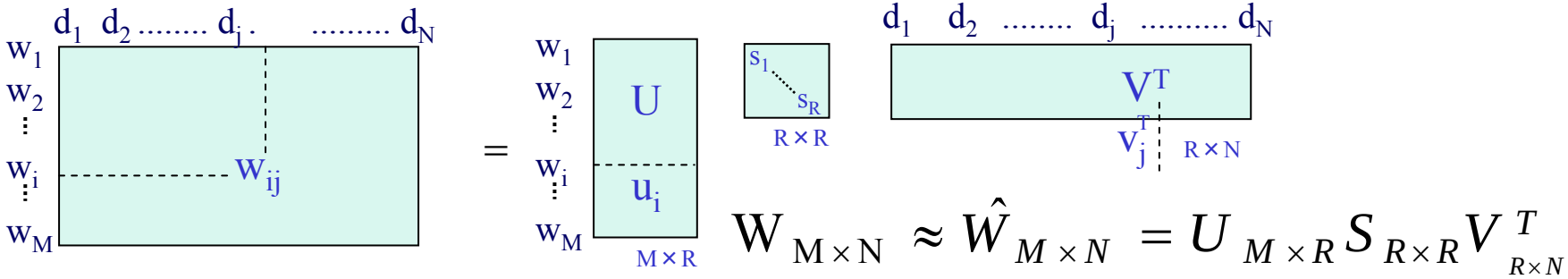
s_i^2 : weights (significance of the "component matrices" $e'_i\, e'^T_i$)

$$W^T_{N\times M} W_{M\times N} \approx V_{N\times R} S^2_{R\times R} V^T_{R\times N}, \qquad V_{N\times R} = [e'_1, e'_2 .. e'_R]$$

R "concepts" or "latent semantic concepts"

dimensionality reduction: selection of R largest eigenvalues

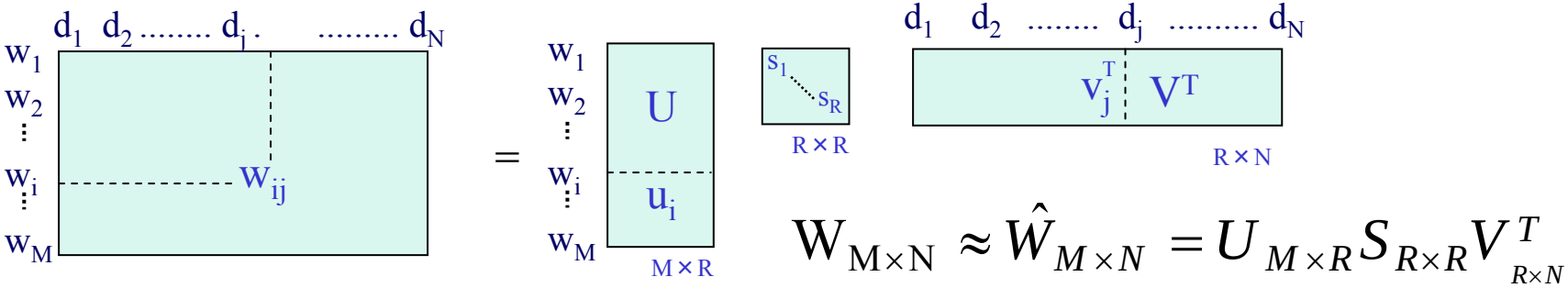
Singular Value Decomposition (SVD)



s_i: singular values, $s_1 \geq s_2 .... \geq s_R$, U: left singular matrix, V: right singular matrix

- **Vectors for word w_i: $u_i S = \underline{u}_i$ (a row)**
  - a vector with dimentionality N reduced to a vector $u_i S = \underline{u}_i$ with dimentionality R
  - "discrete" dimentionality defined by N documents reduced to "continuous" dimentionality defined by R "concepts"
  - the R row vectors of V^T, or column vectors of V, or eigenvectors $\{e'_1,..e'_R\}$, are the R orthouormal basis for the "latent semantic space" with dimentionality R, with which $u_i S = \underline{u}_i$ is represented
- **The Association Structure between words w_i and documents d_j is preserved with noisy information deleted, while the dimensionality is reduced to a common set of R "concepts"**

Singular Value Decomposition (SVD)



- **Vectors for document d_j: $v_j S = \underline{v}_j$ (a row, or $\underline{v}_j = S\, v_j^T$ for a column)**
  - a vector with dimentionality M reduced to a vector $v_j S = \underline{v}_j$ with dimention R
  - "discrete" dimentionality defined by M words reduced to "continuous" dimentionality defined by R "concepts"
  - the R columns of U, or eigenvectors$\{e_1,...e_R\}$, are the R orthonormal basis for the "latent semantic space" with dimensionality R, with which $v_j S = \underline{v}_j$ is represented
- **The Association Structure between words w_i and documents d_j is preserved with noisy information deleted, while the dimensionality is reduced to a common set of R "concepts"**

LSI Ranking

- The user query can be modelled as a pseudo-document in the original (W) matrix
- Assume the query is modelled as the document numbered 0 in the (W) matrix
- The matrix

$$(W)^t(W)_s$$

quantifies the relantionship 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 (represented as the document numbered 0)

Words and Documents

- Columns of U : orthonormal documents
- Columns of V : orthonormal words
- Word vector : $u_i S$
- Document vector : $v_j S$
- Words close in LS space appear in similar documents
- Documents close in LS space convey similar meaning

LSA as Knowledge Representation

- Projecting a new document, $\boldsymbol{x}$, in LS space
- Calculate the frequency count [x_i] of words in the document

$$\boldsymbol{d} = \boldsymbol{U}\,\boldsymbol{S}\,\boldsymbol{x}^{\boldsymbol{T}} \Rightarrow \boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{d} = \boldsymbol{S}\boldsymbol{x}^{\boldsymbol{T}}$$

- Thus,

$$\hat{\mathbf{d}}_{LSA} = \mathbf{S}y^{\mathbf{T}} = \mathbf{U}^{\mathbf{T}}\mathbf{d} = \sum_i (1-\varepsilon_i)x_i\mathbf{u}_i$$

Word Clustering

– Example applications: class-based language modeling, information retrieval, etc.
– Words with similar "semantic concepts" have "closer" location in the "latent semantic space"
  • they tend to appear in similar "types" of documents, although not necessarily in exactly the same documents
– Each component in the reduced word vector $u_j S = \underline{u}_j$ is the "association" of the word with the corresponding "concept"
– Example similarity measure between two words:

$$sim(w_i, w_j) = \frac{\underline{u}_i \cdot \underline{u}_j}{|\underline{u}_i| \cdot |\underline{u}_j|} = \frac{u_i S^2 u_j^T}{|u_i S| \cdot |u_j S|}$$

Document Clustering

– Example applications: clustered language modeling, language model adaptation, information retrieval, etc.
– Documents with similar "semantic concepts" have "closer" location in the "latent semantic space"
  • they tend to include similar "types" of words, although not necessarily exactly the same words
– Each component on the reduced document vector $v_j S = \underline{v}_j$ is the "association" of the document with the corresponding "concept"
– Example "similarity" measure between two documents:

$$sim(d_i, d_j) = \frac{\underline{v}_i \cdot \underline{v}_j}{\left|\underline{v}_i\right| \cdot \left|\underline{v}_j\right|} = \frac{v_i S^2 v_j}{\left|v_i S\right| \cdot \left|v_j S\right|}$$

Document Clustering

$W^T W = V S^2 V^T$

- Semantic similarity between two commands
  - More in the next class

$$K(d_i, d_j) = \cos(v_i \mathbf{S}, v_j \mathbf{S}) = \frac{v_i \mathbf{S}^2 \mathbf{v}^{\mathbf{T}}{}_j}{\|v_i \mathbf{S}\| \, \|v_j \mathbf{S}\|}$$

- From $W^T W$, find document clusters whose members have similarity measure exceeding a threshold (say 0.95)

More LSA References

1. J. Bellagarda, “Exploiting Latent Semantic Information in Statistical Language Modeling”, Proceedings of the IEEE, Aug 2000
2. “Special Issue on Language Modeling and Dialogue Systems”, IEEE Trans. on Speech & Audio Processing, Jan 2000
3. “Latent Semantic Mapping”, IEEE Signal Processing Magazine, Sept. 2005, Special Issue on Speech Technology in Human-Machine Communication
4. “Golub & Van Loan, “ Matrix Computations ”, 1989
5. “Probabilistic Latent Semantic Indexing”, ACM Special Interest Group on Information Retrieval (ACM SIGIR), 1999
6. “Spoken Document Understanding and Organization”, IEEE Signal Processing Magazine, Sept. 2005, Special Issue on Speech Technology in Human-Machine Communication

Summary

- Today's Class
  - Vector-based document representation and LSA
- Next Classes
  - Project plan finalize on 3/5 (presentation on 4/16)
  - Clustering, text categorization and information retrieval
- Reading Assignments
  - Manning and Schutze, Chapters 9 & 10