

Where are we?

- Refinement/Personalization of results
- Study techniques of

Recommender systems

- Content filtering
 - Applying content filtering to search
- Collaborative filtering
 - Nearest neighbor methods
 - Applying nearest neighbor method to search



- Matrix factorization methods
- Detour to Latent Semantic indexing 1

Latent Semantic Indexing: Introduction

- Analysis of **term-document interaction** for corpus of text documents
- Standard vector model:
 - document vector of term weights
- Goals:
 - **reduce dimension** of document vectors
 - **uncover latent factors**:
 - document as vector of **factor weights**
- uses of theory of linear algebra

2

Matrix formulation

- M** - number of terms in lexicon
- N** - number of documents in collection
- C** the **MxN** (termxdoc.) **matrix of weights** ≥ 0 (our old w_{ij})

$$\begin{pmatrix} c_{11} & \dots & c_{M1} \\ \vdots & & \vdots \\ c_{1N} & \dots & c_{MN} \end{pmatrix} \cdot \begin{pmatrix} w_{1q} \\ \vdots \\ w_{Mq} \end{pmatrix} = \begin{pmatrix} s_{1q} \\ \vdots \\ s_{Nq} \end{pmatrix}$$

document vector query vector scores

$$s_{xq} = \sum_{i=1}^M (C_{ix} * w_{iq})$$

3

Set-up

- C** the **MxN** (termxdoc.) **matrix of non-negative weights**
 - of **rank r** ($r \leq \min(M,N)$)
 - documents are **columns** of **C**

consider **CC^T** and **C^TC**:

- symmetric,
- share the same **eigenvalues** $\lambda_1, \lambda_2, \dots$
 - $\lambda_1, \lambda_2, \dots$ are indexed in **decreasing order**
- **C^TC(i,j)** measures **similarity** documents **i** and **j**
- **CC^T(i,j)** measures strength **co-occurrence terms** **i** and **j**

4

Use Singular Value Decomposition (SVD)

Theorem:

MxN matrix **C** of rank **r** has a

singular value decomposition $C = U\Sigma V^T$

Where:

U MxM matrix

with columns = **orthogonal eigenvectors** of **CC^T**

V NxN matrix

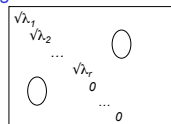
with columns = **orthogonal eigenvectors** of **C^TC**

Σ MxN **diagonal** matrix:

$\Sigma(i,i) = \sqrt{\lambda_i}$ for $1 \leq i \leq r$

$\Sigma(i,j) = 0$ otherwise

$\sqrt{\lambda_i}$ called **singular values**

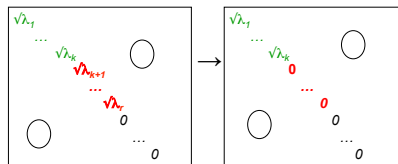


5

Reduce Rank

- **Reduce rank** of **Σ** from **r** to **k**
keep only **k largest singular values**

Σ_k is MxN diagonal matrix: $\Sigma(i,i) = \sqrt{\lambda_i}$ for $1 \leq i \leq k$
 $\Sigma(i,j) = 0$ otherwise



6

Reduced Rank Approximation of C

- Approximation: $C_k = U \Sigma_k V^T$
 $[M \times N] \quad [M \times M] \quad [M \times N] \quad [N \times N]$
- Theorem: C_k is the best rank-k approximation to C under the least square fit (Frobenius) norm

$$= \sqrt{\sum_{i=1}^M \sum_{j=1}^N (C(i,j) - C_k(i,j))^2}$$

7

Reduced dimension matrices

$C_k = U'_k \Sigma'_k V_k^T$
 $M \times N \quad M \times k \quad k \times k \quad k \times N$

8

Semantic Interpretation

- remaining k dimensions: k factors
- View V_k^T as a representation of documents in the k-dimensional space
- View U_k^T as a representation of terms in the k-dimensional space
- Σ_k scales between them
- find some semantic relationship?
 - “concept space”?
 - correlating terms to find structure
 - synonymy
 - polysomy

“people choose same main terms <20% time”

9

Using the Approximation

- V_k^T as a representation of documents in a k-dimensional space
- Transform query vector q into that space:

$$C_k^T C_k = (U'_k \Sigma'_k V_k^T)^T (U'_k \Sigma'_k V_k^T) = (V_k \Sigma_k^T U_k^T) (U'_k \Sigma'_k V_k^T)$$

$$= V_k (\Sigma'_k)^2 (V_k^T)^T \quad \text{compares documents}$$
- $C_k^T q$ should = $V_k (\Sigma'_k)^2 q_k$ compare doc. to query
- $q_k = (\Sigma'_k)^{-2} V_k^T C_k^T q = (\Sigma_k^{-1})^2 V_k^T V_k \Sigma_k^T U_k^T q$

$$= (\Sigma'_k)^{-1} (U'_k)^T q$$

recalling $(V_k^T)(V_k) = (U_k^T)(U_k) = I_{10}$

Adding a new document

add new document d^{new} to $C_k \Rightarrow$ add column d_k^{new} to V_k^T
 Transform d^{new} into the k-dimensional space version d_k^{new}

$$V_k^T = (\Sigma'_k)^{-1} (U'_k)^T C_k \quad \Rightarrow \quad (\Sigma'_k)^{-1} (U'_k)^T d^{new} = d_k^{new}$$

$C_k = U'_k \Sigma'_k V_k^T$
 $M \times (N+1) \quad M \times k \quad k \times k \quad k \times (N+1)$

11

Original LSI paper:

Deerwester, Dumais, et. al.
Indexing by Latent Semantic Analysis
 Journal of the Society for Information Science,
 41(6), 1990, 391-407.

Example from that paper follows

12

Deerwester, Dumais et. al. Table:

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

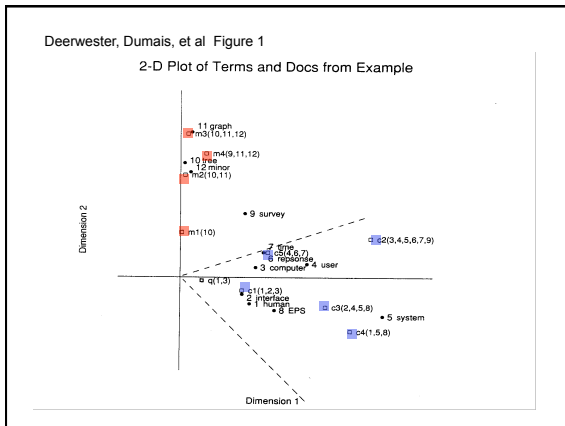
13

Deerwester, Dumais et. al. example, cont.:

Matrix $V'_k{}^T$ for $k=2$

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

14



Summary

- LSI uses SVD to get a **reduced-rank** and **reduced-size** approximation to C
- LSI can be viewed as a **preprocessor** for
 - query evaluation
 - clustering
- SVD **computation** can be **costly**
 - do once (or rarely)

16