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

1

Matric formulation

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

$$(w_1 \ \dots \ w_M) \bullet \begin{bmatrix} c_{11} & \dots & c_{1N} \\ c_{M1} & \dots & c_{MN} \end{bmatrix} = (s_1 \ \dots \ s_N)$$

query vector

document vector

scores

$$s_x = \Sigma_{i=1}^{M}(w_i * c_{ix})$$

2

Set-up

C the M×N (term×doc.) matrix of non-negative weights
- of rank r (r ≤ 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

3

Use Singular Value Decomposition (SVD)

Theorem:
M×N matrix C of rank r has a
singular value decomposition $C = U\Sigma V^T$
Where:
U M×r matrix
with columns = orthonormal eigenvectors of CC^T
V N×r matrix
with columns = orthonormal eigenvectors of C^TC
Σ r×r 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*

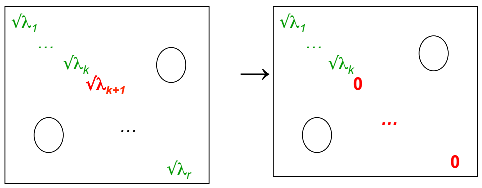
$$\begin{bmatrix} \sqrt{\lambda_1} & & & & & & \\ & \sqrt{\lambda_2} & & & & 0 & \\ & & \dots & & & & \\ & & & \sqrt{\lambda_r} & & & \\ & 0 & & & 0 & & \\ & & & & & \dots & \\ & & & & & & 0 \end{bmatrix}$$

4

Reduce Rank

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

  Σ_K is M×N diagonal matrix: $\Sigma(i,i) = \sqrt{\lambda_i}$ for $1 \le i \le$ **k**
  $\Sigma(i,j) = 0$ otherwise



Reduced Rank Approximation of C

- Approximation:

$$C_k = U\Sigma_k V^T$$

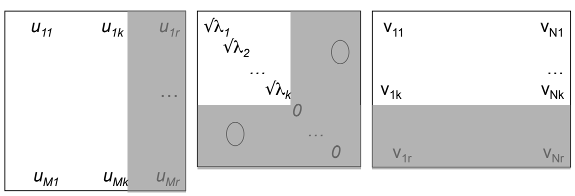
[M×N] [M×r] [r×r] [r×N]

- Theorem:

  C_k is the best rank-k approximation to C under the least square fit (Frobenius) norm

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

Reduced dimension matrices



C_k = M×N **U_k** M×k **Σ_k** k×k **V_k^T** k×N

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 – strength of each factor
- find some semantic relationship?
  - "concept space"?
  - correlating terms to find structure
    - synonomy
    - polysomy

    "people choose same main terms <20% time"

Using the Approximation

- V_k^T as a representation of documents in a k-dimensional space

  $C_k = (U_k \, \Sigma_k \, V_k^T)$

  $(\Sigma_k)^{-1} U_k^T \, C_k = (\Sigma_k)^{-1} U_k^T \, U_k \, \Sigma_k \, V_k^T = V_k^T$

- $C_k^T \, C_k = (U_k \, \Sigma_k \, V_k^T)^T (U_k \, \Sigma_k \, V_k^T)$

  $= (V_k \, \Sigma_k^T \, U_k^T)(U_k \, \Sigma_k \, V_k^T)$

  $= V_k \, (\Sigma_k)^2 \, (V_k)^T$ compares documents

  recalling $(V_k^T)(V_k) = (U_k^T)(U_k) = I$

Remember C_k is still M×N but is rank-k approximation

9

Adding a new document

add new document $\boldsymbol{d}^{new}$ to C_k => add column $\boldsymbol{d}_k^{new}$ to V_k^T

Transform $\mathbf{d}^{new}$ into the k-dimensional space version $\boldsymbol{d}_k^{new}$

$V_k^T = (\Sigma_k)^{-1} (U_k)^T \, C_k$ => $(\Sigma_k)^{-1} (U_k)^T \boldsymbol{d}^{new} = \boldsymbol{d}_k^{new}$

$$\begin{bmatrix} u_{11} & \cdots & u_{1k} \\ & \cdots & \\ u_{M1} & \cdots & u_{Mk} \end{bmatrix} \begin{bmatrix} \sqrt{\lambda_1} & & 0 \\ & \sqrt{\lambda_2} \quad \cdots & \\ 0 & & \sqrt{\lambda_k} \end{bmatrix} \left[\begin{array}{ccc|c} v_{11} & & v_{N1} & d_k^{new}(1) \\ & \cdots & & \cdots \\ v_{1k} & & v_{Nk} & d_k^{new}(k) \end{array}\right]$$

$C_k =$ M×(N+1) U_k M×k Σ'_k k×k V'^T_k k×(N+1)

10

Querying with the Approximation

- Transform query vector **q** into rank-k space
- Use same transformation as for new document:

$$(\Sigma_k)^{-1} (U_k)^T \boldsymbol{q} = \boldsymbol{q}_k$$

recalling $(V'^T_k)(V'_k) = (U'^T_k)(U'_k) = I$

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.:

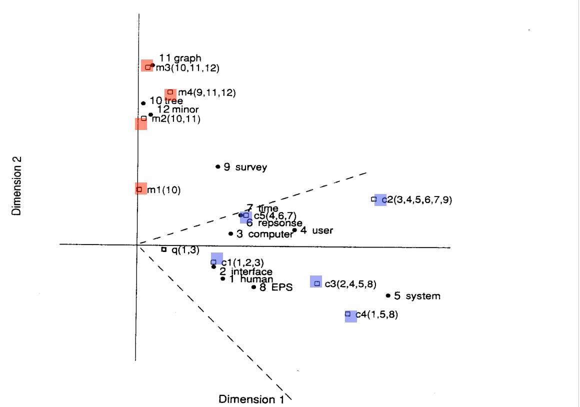
Matrix V_k^T for k=2

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

Deerwester, Dumais, et al Figure 1

2-D Plot of Terms and Docs from Example



Summary

- LSI is a specific application of SVD and rank reduction
  - terms, documents, concepts
- Gives reduced-rank and reduced-size approximation to C
- LSI can be viewed as a preprocessor for
  - query evaluation
  - clustering

16

Remarks

- sparse C gives dense C_k
  - approximations
- SVD computation can be costly
  - do once (or rarely)

17