Latent Semantic Indexing

1

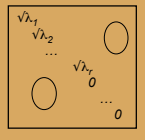
Summary: Singular Value Decomposition (SVD)

M - number of terms N - number of documents
C the term×document matrix of weights *(*our old w_{ij})
– of rank r (r ≤ min(M,N))
CC^T and C^TC: symmetric, share the same eigenvalues $\lambda_1, \lambda_2, \ldots$
– $\lambda_1, \lambda_2, \ldots$ are indexed in decreasing order

SVD: $C = U\Sigma V^T$

U is M×M; columns are orthogonal eigenvectors of CC^T
V is N×N; columns are orthogonal eigenvectors of C^TC
Σ is M×N diagonal matrix: $\Sigma(i,i) = \sqrt{\lambda_i}$ for *1* ≤ i ≤ r
$\Sigma(i,j) = 0$ otherwise

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

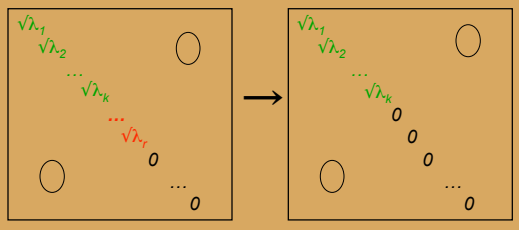


2

Summary: Reduced Rank Approximation of C

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

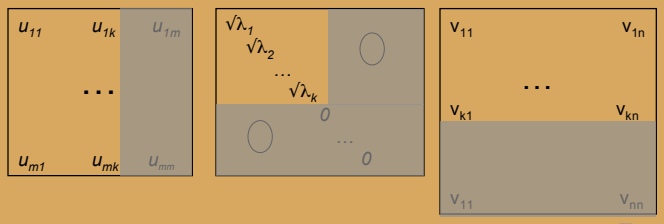
Σ_K is M×N diagonal matrix: $\Sigma(i,i) = \sqrt{\lambda_i}$ for $1 \le i \le k$
$\Sigma(i,j) = 0$ otherwise



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

3

Summary: Reduce dimension matrices
Query approximation



C_k =	U'_k	Σ'_k	$V'_k{}^T$
M×N	M×k	k×k	k×N

View $V'_k{}^T$ as a representation of documents in a k-dimensional space.

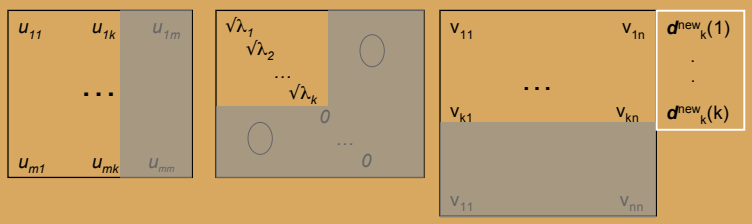
$V'_k{}^T = (\Sigma'_k)^{-1} (U'_k)^T C_k$ => $(\Sigma'_k)^{-1} (U'_k)^T \boldsymbol{q} = \boldsymbol{q}_k$

$C_k{}^T C_k = V'_k (\Sigma'_k)^2 (V'_k)^T$ => $V'_k (\Sigma'_k)^2 \boldsymbol{q}_k = C_k{}^T \boldsymbol{q}$

compares documents compares doc. to query

4

Adding a new document to approximation



C_k = U'_k Σ'_k V'^T_k

M×(N+1) M×k k×k k×(N+1)

View as a representation of documents in a k-dimensional space.

$V'^T_k = (\Sigma'_k)^{-1} (U'_k)^T C_k$ => $(\Sigma'_k)^{-1} (U'_k)^T \boldsymbol{d}^{new} = \boldsymbol{d}^{new}{}_k$

$\boldsymbol{d}_{new}$ is new document to add to C_k => add column $\boldsymbol{d}^{new}{}_k$ to V'^T_k

5

Alternative query approximation

Have: $C_k = U'_k \Sigma'_k V'^T_k$

Just used: $V'^T_k = (\Sigma'_k)^{-1} (U'_k)^T C_k$

as a representation of documents in a k-dimensional space

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

- Appropriate for adding vectors to V'^T_k and comparing queries to docs.

Consider again: $C_k^T C_k = V'_k (\Sigma'_k)^2 (V'_k)^T$

gives a second k-dimensional representation of documents:

$\Sigma'_k V'^T_k = (U'_k)^T C_k$

=> approximate $\boldsymbol{q}$ as $(U'_k)^T \boldsymbol{q} = \boldsymbol{q}'_k$

- Appropriate for comparing queries to docs

  $V'_k (\Sigma'_k) \boldsymbol{q}'_k = V'_k (\Sigma'_k) (U'_k)^T \boldsymbol{q} = C_k^T \boldsymbol{q}$

6

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

7

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

8

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

Matrix V^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

9

Deerwester, Dumais, et al Figure 1