$$u_2 = \frac{\vec{w_2}}{|\vec{w_2}|}$$

$$= \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{pmatrix}.$$

$$\therefore \quad U = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix}.$$

Now for V

$$B^T B \cancel{A^T A} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

$$= \begin{bmatrix} 5 & 4 \\ 4 & 5 \end{bmatrix}$$

$$\begin{bmatrix} 5 & 4 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$5x_1 + 4x_2 = \lambda x_1$$

$$4x_1 + 5x_2 = \lambda x_2.$$

$$(5-\lambda)x_1 + 4x_2 = 0$$

$$4x_1 + (5-\lambda)x_2 = 0$$

$$\begin{vmatrix} (5-\lambda) & 4 \\ 4 & (5-\lambda) \end{vmatrix} = 0$$

$$(5-\lambda)(5-\lambda) - 16 = 0$$

$$25 - 10\lambda + \lambda^2 - 16 = 0$$

$$\lambda^2 - 10\lambda + 9 = 0$$

$$\lambda^2 - 9\lambda - 1\lambda + 9 = 0$$

$$\lambda(\lambda - 9) - 1(\lambda - 9) = 0$$

$$(\lambda - 1)(\lambda - 9) = 0$$

$$\lambda = 1, \quad \lambda = 9$$

$\underline{\lambda = 1}$

$$(5-\lambda)x_1 + 4x_2 = 0$$

$$4x_1 + 4x_2 = 0$$

$$\therefore x_1 = -x_2$$

If $x_1 = 1$; $x_2 = -1$.

$\underline{\lambda = 9}$

$$-4x_1 + 4x_2 = 0$$

$$x_1 = x_2$$

If $x_1 = 1$ $x_2 = 1$.

$\therefore$ Eigenvectors $\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

$v_1 = [1, 1]$

$\vec{u_1} = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$.

$\vec{w_2} = \vec{v_2} - \vec{u_1} \cdot \vec{v_2} * \vec{u_1}$

$= (1, -1) - \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right) \cdot (1, -1) * \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$

$= (1, -1) - (0)\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$

$= (1, -1)$.

$\vec{u_2} = \left(\frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}\right)$.

$\therefore V = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{+1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{pmatrix}$.

$\therefore U = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix}$ $V^T = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix}$

$$S = \begin{bmatrix} \sqrt{9} & 0 \\ 0 & \sqrt{1} \end{bmatrix}$$

$$= \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix}.$$

$$\therefore \; USV^T = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix}$$

$$V = A^T A$$

$$U = A A^T$$

$$\rightarrow \begin{bmatrix} \; \end{bmatrix}$$

SVD (LSI).

Latent Semantic indexing has two advantages over the vector space model:-

1) Synonymy: Synonymy refers to the case where two different words have the same meaning. (say car and automobile)

2) Polysemy: Polysemy refers to the case where term has multiple meanings.

Synonymy and polysemy are handled in LSI. but not in vector space model.

Even for a collection of modest size, the term document matrix C is likely to have several tens of thousands of rows and columns, and a rank in tens of thousands as well.

In LSI, we use the SVD, to construct a low rank approximation C_K to the term-document matrix, for a value of K far smaller than the original rank of C.

Thus each row/column is mapped to K-dimensional space, this space is defined by the K principal eigen vectors (corresponding to the largest eigen values) of C^TC and CC^T

Note that matrix C_K is itself still an $m \times n$ matrix irrespective of K.

A query vector is mapped to its representation in the LSI space by the transformation

$$\vec{q_K} = \Sigma_K^{-1} U_K^T \vec{q}$$

Now, we may use cosine similarities to compute the similarity between a query and a document, between two documents or between two terms.

If a query is close to a document in the original space, it remains relatively close in the K-dimensional space.

Consider the term document matrix

	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
voyage	1	0	0	1	1	0
trip	0	0	0	1	0	1

Matrix A

its SVD is,

U as below

-0.44	-0.30	+0.57	0.58	0.25
-0.13	-0.33	-0.59	0.00	0.73
-0.48	-0.51	-0.37	0	-0.61
-0.70	0.35	0.15	-0.58	0.16
-0.26	0.65	-0.41	0.58	-0.09

Σ as below,

2.16	0	0	0	0
0	1.59	0	0	0
0	0	1.28	0	0
0	0	0	1.00	0
0	0	0	0	0.39

V^T

-0.75	-0.28	-0.20	-0.45	-0.33	-0.12
-0.29	-0.53	-0.19	0.63	0.22	0.41
0.28	-0.75	0.45	-0.20	0.12	-0.33
0.00	0.00	0.58	0.00	-0.58	0.58
-0.53	0.29	0.63	0.19	0.41	-0.22

FOR K=2,

Σ_2 is

2.16	0	0	0	0
0	1.59	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

From this we compute C_2.

-1.62	-0.60	-0.44	-0.97	-0.70	-0.26
-0.46	-0.84	-0.30	1.00	0.35	0.65
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Truncated $(V)^T$ is

-1.62	-0.60	-0.44	-0.97	-0.70	-0.26
-0.46	-0.84	-0.30	1.00	0.35	0.65

The retrieval quality may improve by the dimensionality reduction.

Curse of dimensionality

The curse of dimensionality refers to various phenomena that arise when analyzing and organizing data in high dimensional spaces (often with hundreds or thousands of dimensions) that do not occur in low dimensional settings. ~~such~~.

The common theme of these problems is that when the dimensionality increases, the volume of the space increases so fast that the available data becomes sparse.

This sparsity is problematic for any method that requires statistical significance. Organizing and searching data often relies on detecting areas where objects form groups with similar properties; in high dimensional data however objects appear to be sparse and dissimilar in many ways which prevents common data organization strategies from being efficient.

Hence we need dimensionality reduction.

Application of LSI.

The new low dimensional space can be typically used to :

1. Compare the documents in low dimensional space (data clustering, document classification).

2. Find similar documents across languages, after analyzing a base set of translated documents (cross language retrieval).

3. Find relations between terms (synonymy and polysemy).

4. Given a query of terms, translate it into low dimensional space and find matching documents (information retrieval).

5. Find the best similarity between small groups of terms in a semantic way (ie in a context of knowledge corpus). eg: in a MCQ multiple choice questions answering model.