SVD - LSI

Mi Islita

Information Retrieval Intelligence

Your Source for Information Retrieval and Intelligence
"Where Marketing Meets Science"

http://www.miislita.com

Singular Value Decomposition

- In 1965 G. Golub and W. Kahan introduced Singular Value Decomposition (SVD) as a decomposition technique for calculating the singular values, pseudo-inverse and rank of a matrix.
- The conventional way of doing this was to convert a matrix to a row-echolon form.
- The rank of a matrix is then given by the number of nonzero rows or columns of the echolon form, whichever of these two numbers is smaller.

SVD

- Equation 1: $\mathbf{A = USV^T}$

- **U** is a matrix whose columns are the eigenvectors of the $\mathbf{AA^T}$ matrix. These are termed the *left eigenvectors.*

- **S** is a matrix whose diagonal elements are the singular values of **A**. This is a diagonal matrix, so its nondiagonal elements are zero by definition.

- **V** is a matrix whose columns are the eigenvectors of the $\mathbf{A^TA}$ matrix. These are termed the *right eigenvectors*.

- $\mathbf{V^T}$ is the transpose of **V**.

On Dimensionality Reduction

- When computing the SVD of a matrix is desirable to reduce its dimensions by keeping its first *k* singular values
- Since these are ordered in decreasing order along the diagonal of **S** and this ordering is preserved when constructing **U** and $\mathbf{V^T}$, keeping the first *k* singular values is equivalent to keeping the first *k* rows of **S** and $\mathbf{V^T}$ and the first *k* columns of **U**. Equation 1 reduces to
- Equation 2: $\mathbf{A^* = U^* S^* V^T}$
- This process is termed dimensionality reduction, and A* is referred to as the Rank *k* Approximation of A
- The top *k* singular values are selected as a mean for developing a **"latent semantics"** representation of **A** that is now free from noisy dimensions.

Latent semantics

- This "latent semantics" representation is a specific data structure in low-dimensional space in which documents, terms and queries are embedded and compared.
- Dimensionality reduction is a noise reduction process.
- Dimensionality reduction is somewhat an arbitrary process. How many *k* dimensions to keep can lead to the so-called "dimensionality reduction curse" in which performance is affected.
- In 1988 Deerwester et. al. used SVD to deal with the vocabulary problem in human-computer interaction and called their approach Latent Semantic Indexing (LSI)

Computing S

- Equation 1: $\mathbf{A = USV^T}$

- **S** is computed by the following procedure:

1. $\mathbf{A^T}$ and $\mathbf{A^TA}$ are computed.
2. the eigenvalues of $\mathbf{A^TA}$ are determined and sorted in descending order, in the absolute sense. The nonnegative square roots of these are the singular values of **A**.
3. **S** is constructed by placing singular values in descending order along its diagonal.

SVD

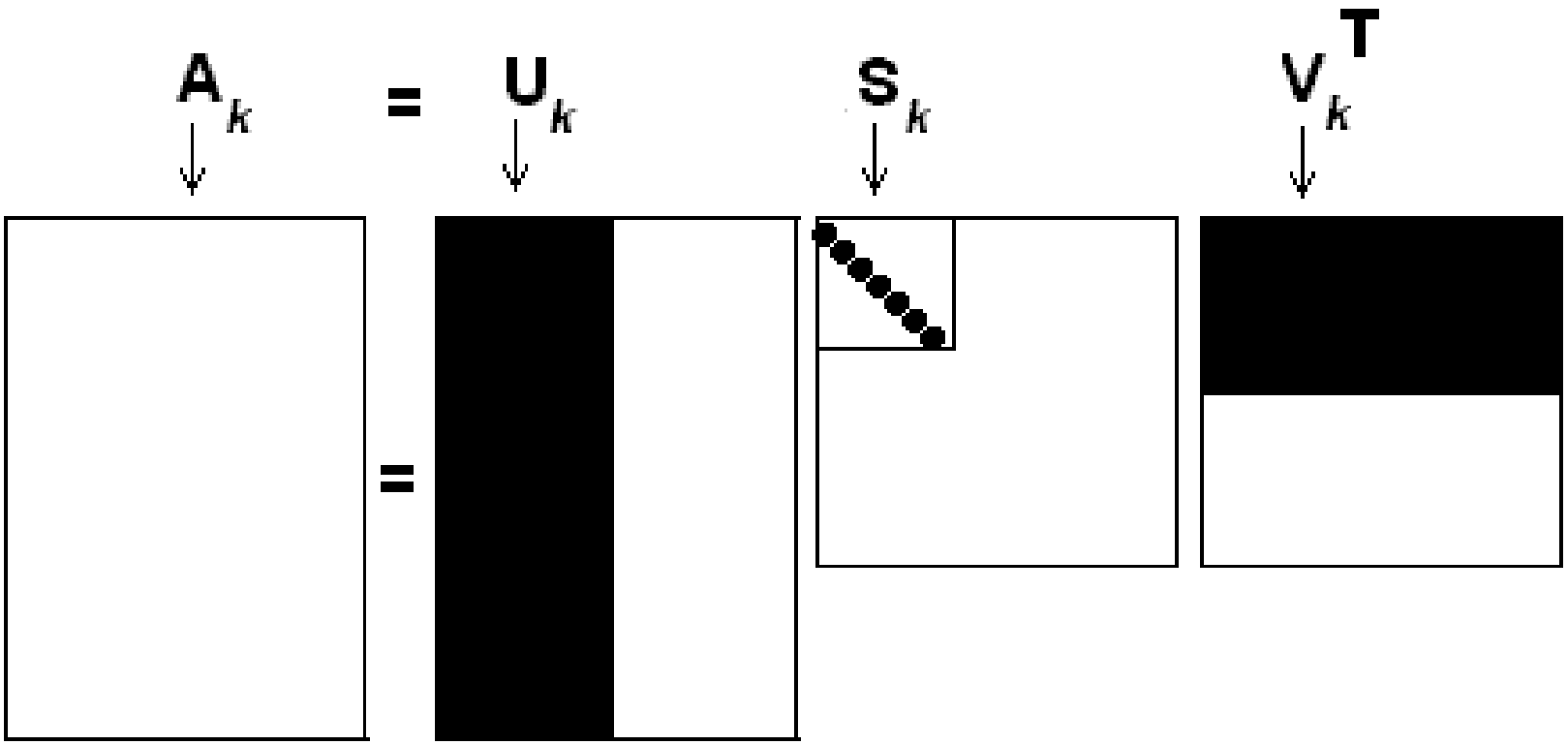
$$A = USV^T = \begin{bmatrix} 0.4472 & 0.8944 \\ 0.8944 & -0.4472 \end{bmatrix} \begin{bmatrix} 6.3245 & 0 \\ 0 & 3.1622 \end{bmatrix} \begin{bmatrix} 0.7071 & -0.7071 \\ 0.7071 & 0.7071 \end{bmatrix}$$

$$A = USV^T = \begin{bmatrix} 0.4472 & 0.8944 \\ 0.8944 & -0.4472 \end{bmatrix} \begin{bmatrix} 4.4721 & -4.4721 \\ 2.2360 & 2.2360 \end{bmatrix}$$

$$A = USV^T = \begin{bmatrix} 3.9998 & 0 \\ 2.9999 & -4.9997 \end{bmatrix} \approx \begin{bmatrix} 4 & 0 \\ 3 & -5 \end{bmatrix}$$

http://www.miislita.com/information-retrieval-tutorial/svd-lsi-tutorial-2-understanding.html

Reduced SVD



The shaded areas indicate the part of the matrices retained. The approximated matrix $\mathbf{A}_k$ is the Rank *k* Approximation of the original matrix and is defined as

Equation 8: $\mathbf{A}_k = \mathbf{U}_k\mathbf{S}_k\mathbf{V}^T_k$

LSI

- So, how does SVD is used in Latent Semantic Indexing (LSI)?
- In LSI, it is not the intent to reconstruct **A**. The goal is to find the best rank *k* approximation of **A** that would improve retrieval.
- The selection of *k* and the number of singular values in **S** to use is still an open area of research.
- During her ternure at Bellcore (now Telcordia), Microsoft's Susan Dumais mentioned in the 1995 presentation “Transcription of the Application” that her research group experimented with *k* values largely "**by seat of the pants**".

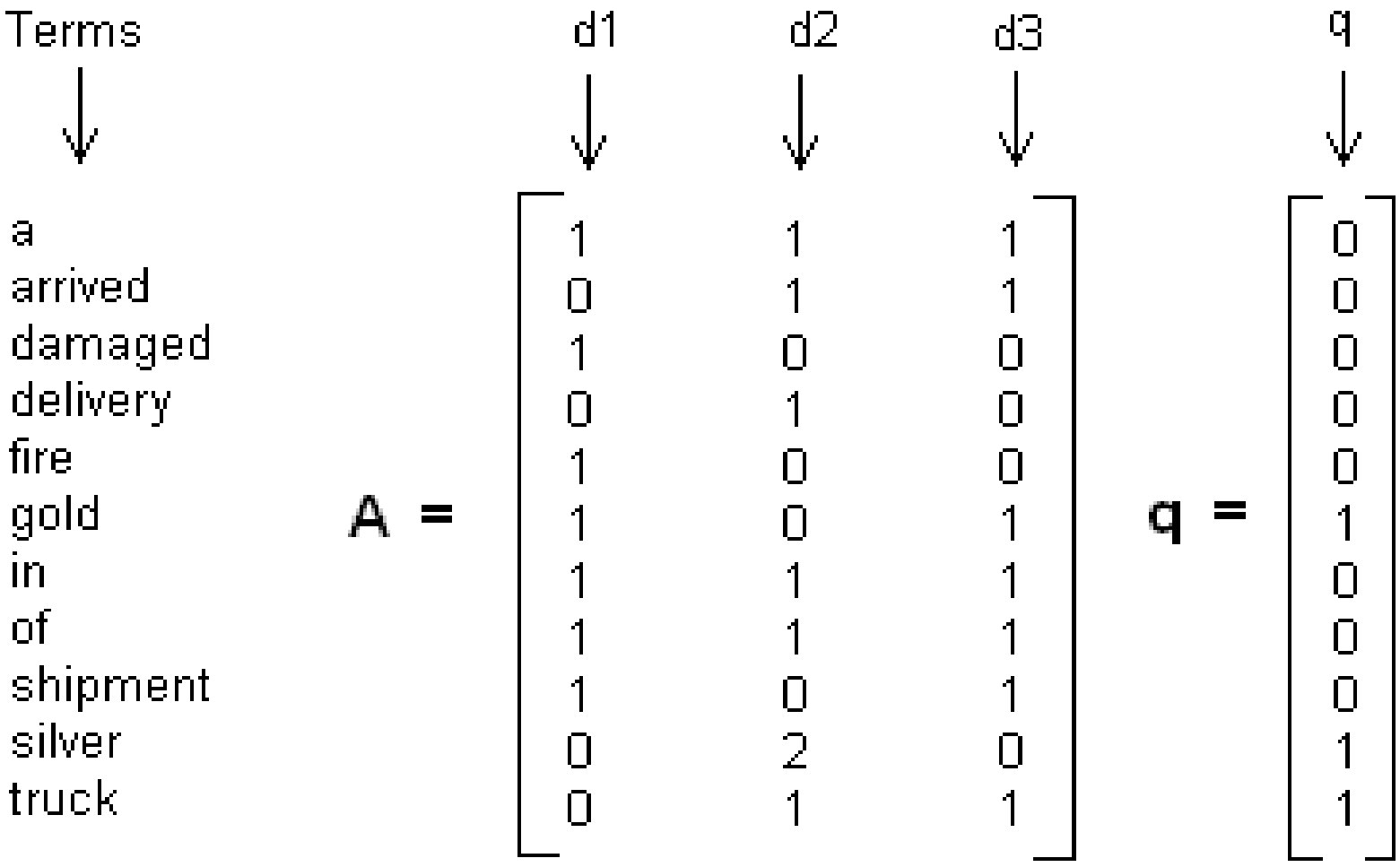
Example

- The query is *gold silver truck* and the "collection" consists of just three "documents":

- d1: *Shipment of gold damaged in a fire.*
- d2: *Delivery of silver arrived in a silver truck.*
- d3: *Shipment of gold arrived in a truck.*

  - (a) stopwords **are not** ignored,
  - (b) text is tokenized and lowercased,
  - (c) no stemming is used and
  - (d) unique terms are sorted alphabetically.

Term count model



http://www.miislita.com/information-retrieval-tutorial/svd-lsi-tutorial-3-understanding.html

SVD results

$$U = \begin{bmatrix} -0.4201 & 0.0748 & -0.0460 \\ -0.2995 & -0.2001 & 0.4078 \\ -0.1206 & 0.2749 & -0.4538 \\ -0.1576 & -0.3046 & -0.2006 \\ -0.1206 & 0.2749 & -0.4538 \\ -0.2626 & 0.3794 & 0.1547 \\ -0.4201 & 0.0748 & -0.0460 \\ -0.4201 & 0.0748 & -0.0460 \\ -0.2626 & 0.3794 & 0.1547 \\ -0.3151 & -0.6093 & -0.4013 \\ -0.2995 & -0.2001 & 0.4078 \end{bmatrix} \quad S = \begin{bmatrix} 4.0989 & 0.0000 & 0.0000 \\ 0.0000 & 2.3616 & 0.0000 \\ 0.0000 & 0.0000 & 1.2737 \end{bmatrix}$$

$$V = \begin{bmatrix} -0.4945 & 0.6492 & -0.5780 \\ -0.6458 & -0.7194 & -0.2556 \\ -0.5817 & 0.2469 & 0.7750 \end{bmatrix} \quad V^T = \begin{bmatrix} -0.4945 & -0.6458 & -0.5817 \\ 0.6492 & -0.7194 & 0.2469 \\ -0.5780 & -0.2556 & 0.7750 \end{bmatrix}$$

http://www.miislita.com/information-retrieval-tutorial/svd-lsi-tutorial-3-understanding.html

Dimensionality reduction (figure 4)

$$k = 2$$

$$U \approx U_k = \begin{bmatrix} -0.4201 & 0.0748 \\ -0.2995 & -0.2001 \\ -0.1206 & 0.2749 \\ -0.1576 & -0.3046 \\ -0.1206 & 0.2749 \\ -0.2626 & 0.3794 \\ -0.4201 & 0.0748 \\ -0.4201 & 0.0748 \\ -0.2626 & 0.3794 \\ -0.3151 & -0.6093 \\ -0.2995 & -0.2001 \end{bmatrix} \qquad S \approx S_k = \begin{bmatrix} 4.0989 & 0.0000 \\ 0.0000 & 2.3616 \end{bmatrix}$$

$$V \approx V_k = \begin{bmatrix} -0.4945 & 0.6492 \\ -0.6458 & -0.7194 \\ -0.5817 & 0.2469 \end{bmatrix} \qquad V^T \approx V_k^T = \begin{bmatrix} -0.4945 & -0.6458 & -0.5817 \\ 0.6492 & -0.7194 & 0.2469 \end{bmatrix}$$

Incorporating the Query and Ranking the Documents

- Now to incorporate the query we use the procedure described by Berry, Dumais and OBrien in “Using Linear Algebra for Intelligent Information Retrieval” Since **S** is symmetric along its diagonal, $\mathbf{S^T = S}$ and from Equation 3 we can see that
- Equation 4: $\mathbf{A^T = (USV^T)^T = VSU^T}$
- Equation 5: $\mathbf{A^TUS^{-1} = VSU^TUS^{-1}}$
- Equation 6: $\mathbf{V = A^TUS^{-1}}$

Similarity

- Equation 7: $\mathbf{d = d^T U S^{-1}}$
- Since in LSI a query is treated just as another document then the query vector is given by
- Equation 8: $\mathbf{q = q^T U S^{-1}}$
- Thus, in the reduced *k*-dimensional space we can write
- Equation 9: $\mathbf{d = d^T U_k S_k^{-1}}$
- Equation 10: $\mathbf{q = q^T U_k S_k^{-1}}$
- Equation 9 and Equation 10 contain the new coordinates of the vectors in this reduced space. Query-document cosine similarity measures are then possible using
- Equation 11: $\mathbf{sim(q, d) = sim(q^T U_k S_k^{-1}, d^T U_k S_k^{-1})}$

Query projection

- We can reuse Equation 9, 10 and 11 anytime we want to compare cosine similarities between documents and queries. However, with n number of documents this is a formidable task. So, let's simplify a bit further.
- From **V** we can see that for n number of documents, this matrix must contain n number of rows holding eigenvector values. Each of these rows then holds the coordinates of individual document vectors. From Figure 4 these coordinates are:
- d1(-0.4945, 0.6492)
  d2(-0.6458, -0.7194)
  d3(-0.5817, 0.2469)
- It is now clear that in LSI the "right" eigenvectors from the SVD algorithm are document vector coordinates. So, we just need to compute the new query vector coordinates in the reduced space (see Figure 5). An analogous analysis demonstrates that the rows of **U** holds term vector coordinates.

Reduced query

$$\mathbf{q} = \mathbf{q}^T \mathbf{U}_k \mathbf{S}_k^{-1} \qquad k = 2$$

$$\mathbf{q} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} -0.4201 & 0.0748 \\ -0.2995 & -0.2001 \\ -0.1206 & 0.2749 \\ -0.1576 & -0.3046 \\ -0.1206 & 0.2749 \\ -0.2626 & 0.3794 \\ -0.4201 & 0.0748 \\ -0.4201 & 0.0748 \\ -0.2626 & 0.3794 \\ -0.3151 & -0.6093 \\ -0.2995 & -0.2001 \end{bmatrix} \begin{bmatrix} \frac{1}{4.0989} & 0.0000 \\ 0.0000 & \frac{1}{2.3616} \end{bmatrix}$$

$$\mathbf{q} = \begin{bmatrix} -0.2140 & -0.1821 \end{bmatrix}$$

Cosine similarities in reduced space

$$\mathbf{sim(q,\ d)} = \frac{\mathbf{q} \bullet \mathbf{d}}{|\ \mathbf{q}\ |\ |\ \mathbf{d}\ |}$$

$$\mathbf{sim(q,\ d_1)} = \frac{(-0.2140)\ (-0.4945)\ +\ (-0.1821)\ (0.6492)}{\sqrt{(-0.2140)^2 + (-0.1821)^2}\ \sqrt{(-0.4945)^2 + (0.6492)^2}} = -0.0541$$

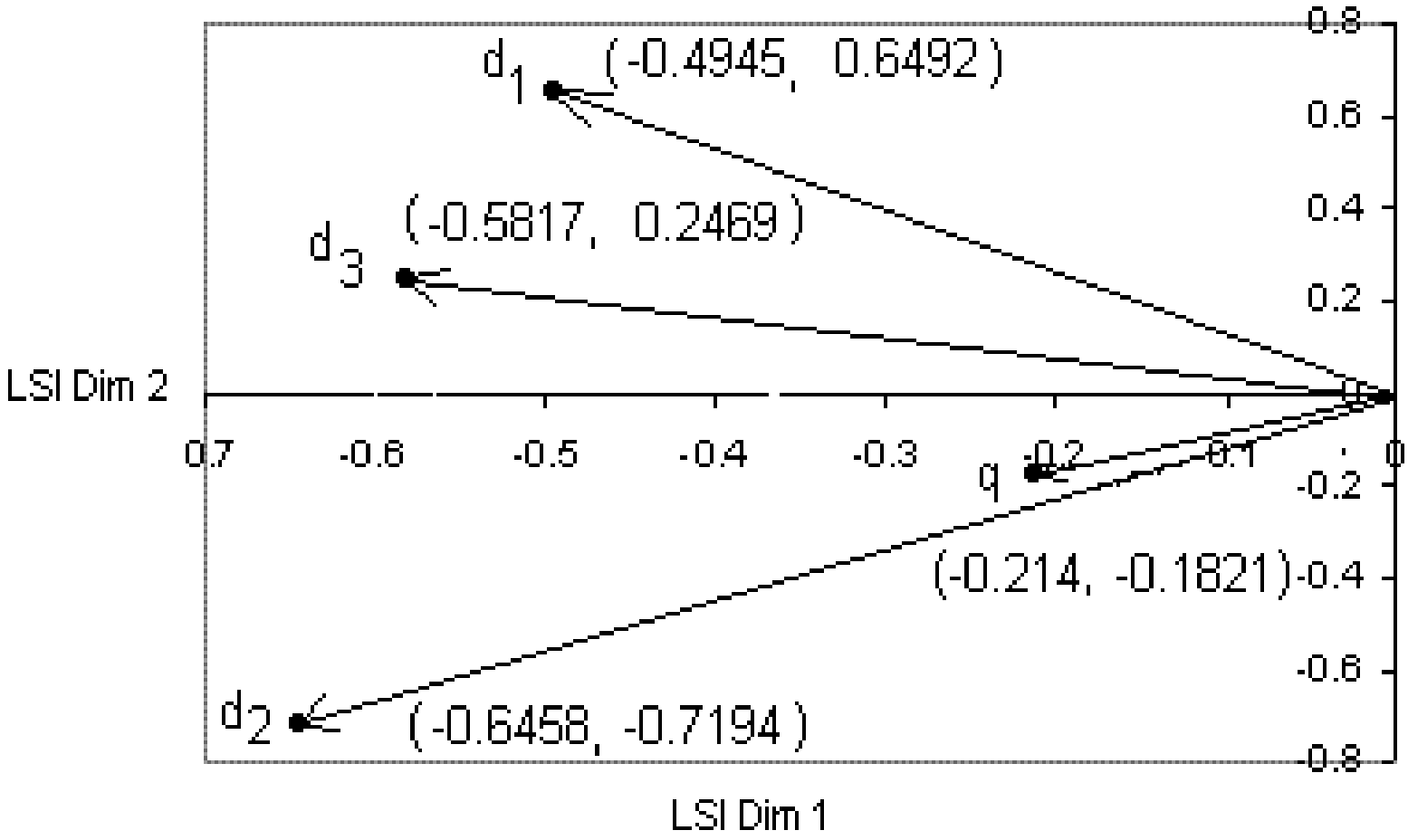
$$\mathbf{sim(q,\ d_2)} = \frac{(-0.2140)\ (-0.6458)\ +\ (-0.1821)\ (-0.7194)}{\sqrt{(-0.2140)^2 + (-0.1821)^2}\ \sqrt{(-0.6458)^2 + (-0.7194)^2}} = 0.9910$$

$$\mathbf{sim(q,\ d_3)} = \frac{(-0.2140)\ (-0.5817)\ +\ (-0.1821)\ (0.2469)}{\sqrt{(-0.2140)^2 + (-0.1821)^2}\ \sqrt{(-0.5817)^2 + (0.2469)^2}} = 0.4478$$

Ranking documents in descending order

$$\mathbf{d_2} > \mathbf{d_3} > \mathbf{d_1}$$

Vectors in reduced space



http://www.miislita.com/information-retrieval-tutorial/svd-lsi-tutorial-4-understanding.html

Summary

To rank documents with LSI, we just need to

1. compute weights using a specific term weight scoring system.
2. construct the term-document matrix **A**.
3. decompose **A**, compute and truncate all required matrices.
4. find new coordinates of query and document vectors in the reduced *k*-dimensional space.
5. sort documents in decreasing order of cosine similarity values.