Latent Semantic Indexing (LSI) An Example

(taken from Grossman and Frieder's Information Retrieval, Algorithms and Heuristics)

A "collection" consists of the following "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.*

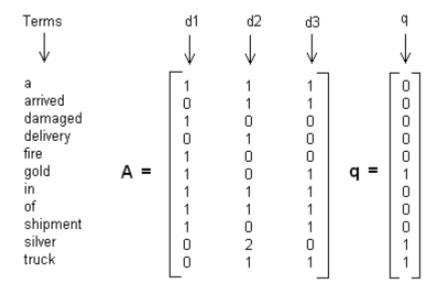
Suppose that we use the term frequency as term weights and query weights. The following document indexing rules are also used:

- stop words were not ignored
- text was tokenized and lowercased
- no stemming was used
- terms were sorted alphabetically

We wish to use this example to illustrate how LSI works.

Problem: Use Latent Semantic Indexing (LSI) to rank these documents for the query *gold silver truck*.

Step 1: Set term weights and construct the term-document matrix **A** and query matrix:



Step 2: Decompose matrix **A** matrix and find the **U**, **S** and **V** matrices, where $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathsf{T}}$

Step 3: Implement a Rank 2 Approximation by keeping the first two columns of **U** and **V** and the first two columns and rows of **S**.

Step 4: Find the new document vector coordinates in this reduced 2-dimensional space.

Rows of **V** holds eigenvector values. These are the coordinates of individual document vectors, hence

d1(-0.4945, 0.6492) d2(-0.6458, -0.7194) d3(-0.5817, 0.2469)

Step 5: Find the new query vector coordinates in the reduced 2-dimensional space.

$$q = q^T U_k S_k^{-1}$$

Note: These are the new coordinate of the query vector in two dimensions. Note how this matrix is now different from the original query matrix **q** given in **Step 1**.

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

Step 6: Rank documents in decreasing order of query-document cosine similarities.

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

$$sim(\mathbf{q}, \mathbf{d}_1) = \frac{(-0.2140)(-0.4945) + (-0.1821)(0.6492)}{\sqrt{(-0.2140)^2 + (-0.1821)^2} \sqrt{(-0.4945)^2 + (0.6492)^2}} = -0.0541$$

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

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

Ranking documents in descending order

$$d_2 > d_3 > d_1$$

We can see that document d2 scores higher than d3 and d1. Its vector is closer to the query vector than the other vectors.