BLG 540E
TEXT RETRIEVAL SYSTEMS

Latent Semantic Indexing

Arzucan Özgür

Faculty of Computer and Informatics, İstanbul Techical University

April 15, 2011

Today's topic

- Latent Semantic Indexing
  - Term-document matrices are very large
  - But the number of topics that people talk about is small (in some sense)
    - Clothes, movies, politics, …
  - Can we represent the term-document space by a lower dimensional latent space?

Vector Space Model: Pros

- **Automatic** selection of index terms
- **Partial matching** of queries and documents *(dealing with the case where no document contains all search terms)*
- **Ranking** according to **similarity score** *(dealing with large result sets)*
- **Term weighting** schemes *(improves retrieval performance)*
- Various extensions
  - Document clustering
  - Relevance feedback (modifying query vector)
- Geometric foundation

Problems with Lexical Semantics

- Ambiguity and association in natural language
  - **Polysemy**: Words often have a **multitude of meanings** and different types of usage *(more severe in very heterogeneous collections)*.
    - bank, jaguar, hot
  - The vector space model is unable to discriminate between different meanings of the same word.

$$\mathrm{sim}_{\mathrm{true}}(d, q) < \cos(\angle(\vec{d}, \vec{q}))$$

Problems with Lexical Semantics

- **Synonymy**: Different terms may have an **identical or a similar meaning**
  - Large/big, Spicy/hot, Car/automobile
- No associations between words are made in the vector space representation.

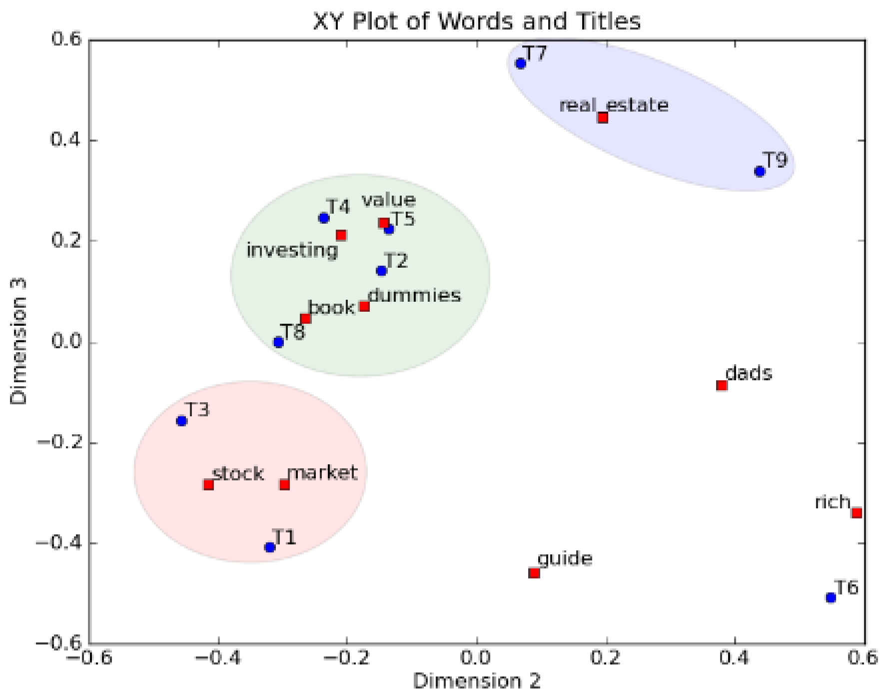
$$\mathrm{sim}_{\mathrm{true}}(d, q) > \cos(\angle(\vec{d}, \vec{q}))$$

Latent Semantic Indexing (LSI)

- Perform a **low-rank approximation** of **document-term matrix** (typical rank **100-300**)
- General idea
  - Map documents (*and* terms) to a **low-dimensional** representation.
  - Design a mapping such that the low-dimensional space reflects **semantic associations** (latent semantic space, identification of hidden (latent) concepts).
  - Compute document similarity based on the **inner product** in this **latent semantic space**

Latent Semantic Analysis

- **Latent semantic space**: illustrating example
- Similar words and documents mapped to similar locations in the lower dimensional latent space.



http://www.puffinwarellc.com/index.php/news-and-articles/articles/33-latent-semantic-analysis-tutorial.html?start=1

Linear Algebra Background

Eigenvalues & Eigenvectors

- **Eigenvectors** (for a square $m \times m$ matrix $\mathbf{S}$)

$$\mathbf{S}\mathbf{v} = \lambda \mathbf{v}$$

Example

$$\begin{pmatrix} 6 & -2 \\ 4 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \end{pmatrix} = 2 \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

(right) eigenvector: $\mathbf{v} \in \mathbb{R}^m \neq \mathbf{0}$

eigenvalue: $\lambda \in \mathbb{R}$

- How many eigenvalues are there at most?

$$\mathbf{S}\mathbf{v} = \lambda \mathbf{v} \iff (\mathbf{S} - \lambda \mathbf{I})\,\mathbf{v} = \mathbf{0}$$

only has a non-zero solution if $|\mathbf{S} - \lambda \mathbf{I}| = 0$

This is a mth order equation in λ which can have **at most m distinct solutions** (roots of the characteristic polynomial) – can be complex even though S is real.

Eigenvectors and eigenvalues

- Example:

$$S = \begin{pmatrix} -1 & 3 \\ 2 & 0 \end{pmatrix} \qquad S - \lambda I = \begin{pmatrix} -1-\lambda & 3 \\ 2 & -\lambda \end{pmatrix}$$

- $|S-\lambda I| = (-1-\lambda)*(-\lambda)-3*2=0$
- Then: $\lambda+\lambda^2-6=0$; $\lambda_1=2$; $\lambda_2=-3$
- For $\lambda_1=2$:

$$\begin{pmatrix} -3 & 3 \\ 2 & -2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 0$$

- Solutions: $x_1=x_2$

Matrix-vector multiplication

$$S = \begin{bmatrix} 30 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

has eigenvalues 30, 20, 1 with corresponding eigenvectors

$$v_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \qquad v_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \qquad v_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Any vector (say $x = \begin{pmatrix} 2 \\ 4 \\ 6 \end{pmatrix}$) can be viewed as a combination of the eigenvectors: $x = 2v_1 + 4v_2 + 6v_3$

Matrix vector multiplication

- Thus a matrix-vector multiplication such as *Sx* (*S*, *x* as in the previous slide) can be rewritten in terms of the eigenvalues/vectors:

$$Sx = S(2\mathrm{v}_1 + 4\mathrm{v}_2 + 6\mathrm{v}_3)$$

$$Sx = 2\,Sv_1 + 4\,Sv_2 + 6\,Sv_3 = 2\lambda_1 v_1 + 4\lambda_2 v_2 + 6\lambda_3 v_3$$

$$Sx = 60\,v_1 + 80\,v_2 + 6\mathrm{v}_3$$

- Even though *x* is an arbitrary vector, the action of *S* on *x* is determined by the eigenvalues/vectors.

Matrix vector multiplication

- Suggestion: the effect of "small" eigenvalues is small.
- If we ignored the smallest eigenvalue (1), then instead of

$$\begin{pmatrix} 60 \\ 80 \\ 6 \end{pmatrix} \quad \text{we would get} \quad \begin{pmatrix} 60 \\ 80 \\ 0 \end{pmatrix}$$

- These vectors are similar (in cosine similarity, etc.)

Eigenvalues & Eigenvectors

For symmetric matrices, eigenvectors for distinct eigenvalues are **orthogonal**

$$Sv_{\{1,2\}} = \lambda_{\{1,2\}} v_{\{1,2\}}, \text{ and } \lambda_1 \neq \lambda_2 \Rightarrow v_1 \cdot v_2 = 0$$

All eigenvalues of a real symmetric matrix are **real**.

$$\text{if } |S - \lambda I| = 0 \text{ and } \mathrm{S} = S^T \Rightarrow \lambda \in \Re$$

All eigenvalues of a positive semidefinite matrix are **non-negative**

$$\overbrace{\forall\, w \in \Re^n, w^T S w \geq 0}, \text{ then if } \mathrm{Sv} = \lambda v \Rightarrow \lambda \geq 0$$

Example

- Let

$$S = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

Real, symmetric.

- Then

$$S - \lambda I = \begin{bmatrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{bmatrix} \Rightarrow$$

$$|S - \lambda I| = (2-\lambda)^2 - 1 = 0.$$

- The eigenvalues are 1 and 3 (nonnegative, real).
- The eigenvectors are orthogonal (and real):

$$\begin{pmatrix} 1 \\ -1 \end{pmatrix} \qquad \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Plug in these values and solve for eigenvectors.

Eigen/diagonal Decomposition

Let $\mathbf{S} \in \mathbb{R}^{m \times m}$ be a **square** matrix with ***m*** **linearly independent eigenvectors**

Theorem: There exists an **eigen decomposition**

(cf. matrix diagonalization theorem)

diagonal

$$\boxed{\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}}$$

- Columns of ***U*** are **eigenvectors** of ***S***
- Diagonal elements of $\mathbf{\Lambda}$ are **eigenvalues** of $\mathbf{S}$

$$\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_m), \quad \lambda_i \geq \lambda_{i+1}$$

Diagonal decomposition: why/how

Let ***U*** have the eigenvectors as columns:
$$U = \begin{bmatrix} v_1 & \ldots & v_n \end{bmatrix}$$

Then, ***SU*** can be written

$$SU = S\begin{bmatrix} v_1 & \ldots & v_n \end{bmatrix} = \begin{bmatrix} \lambda_1 v_1 & \ldots & \lambda_n v_n \end{bmatrix} = \begin{bmatrix} v_1 & \ldots & v_n \end{bmatrix}\begin{bmatrix} \lambda_1 & & \\ & \ldots & \\ & & \lambda_n \end{bmatrix}$$

Thus ***SU=UΛ***, or ***U^{-1}SU=Λ***

And ***S=UΛU^{-1}.***

Diagonal decomposition - example

Recall $$S=\begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix};\ \lambda_1=1,\lambda_2=3.$$

The eigenvectors $\begin{pmatrix}1\\ -1\end{pmatrix}$ and $\begin{pmatrix}1\\ 1\end{pmatrix}$ form $$U=\begin{bmatrix}1 & 1\\ -1 & 1\end{bmatrix}$$

Inverting, we have $$U^{-1}=\begin{bmatrix}1/2 & -1/2\\ 1/2 & 1/2\end{bmatrix}$$

Recall $UU^{-1}=I$.

Then, $\boldsymbol{S=U\Lambda U^{-1}}$ **=** $$\begin{bmatrix}1 & 1\\ -1 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 3\end{bmatrix}\begin{bmatrix}1/2 & -1/2\\ 1/2 & 1/2\end{bmatrix}$$

Example continued

Let's divide ***U*** (and multiply ***U***$^{-1}$) by $\sqrt{2}$

Then, ***S*** **=** $$\begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$

$\boldsymbol{Q}$ $\qquad$ $\boldsymbol{\Lambda}$ $\qquad$ $(\boldsymbol{Q^{-1} = Q^{T}})$

Symmetric Eigen Decomposition

- If $\mathbf{S} \in \mathbb{R}^{m \times m}$ is a **symmetric** matrix:
- **Theorem**: There exists a (unique) **eigen decomposition**

$$S = Q \Lambda Q^T$$

- where **Q** is **orthogonal:**
  - $Q^{-1} = Q^T$
  - Columns of **Q** are normalized eigenvectors
  - Columns are orthogonal.
  - (everything is real)

everything so far needs square matrices

- Recall $M \times N$ term-document matrices …

Singular Value Decomposition

For an $M \times N$ matrix $\mathbf{A}$ of rank r there exists a factorization (Singular Value Decomposition = **SVD**) as follows:

$$A = U \Sigma V^T$$

$\boxed{M \times M}$ $\boxed{M \times N}$ $\boxed{V \text{ is } N \times N}$

The columns of ***U*** are orthogonal eigenvectors of $\boldsymbol{AA^T}$.

The columns of ***V*** are orthogonal eigenvectors of $\boldsymbol{A^TA}$.

Eigenvalues $\lambda_1 \ldots \lambda_r$ of $\boldsymbol{AA^T}$ are the eigenvalues of $\boldsymbol{A^TA}$.

$$\sigma_i = \sqrt{\lambda_i}$$

$$\Sigma = diag\left(\sigma_1 \ldots \sigma_r\right)$$

Singular values.

In Matlab, use *[U,S,V] = svd (A)*

SVD example

Let $A = \begin{bmatrix} 1 & -1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$

Thus M=3, N=2. Its SVD is

$$\begin{bmatrix} 0 & 2/\sqrt{6} & 1/\sqrt{3} \\ 1/\sqrt{2} & -1/\sqrt{6} & 1/\sqrt{3} \\ 1/\sqrt{2} & 1/\sqrt{6} & -1/\sqrt{3} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{3} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$$

- Typically, the singular values arranged in decreasing order.

Low-rank Approximation

- SVD can be used to compute optimal **low-rank approximations**.
- Approximation problem: Find ***A**$_k$* of rank ***k*** such that

$$A_k = \min_{X:\,rank(X)=k} \|A - X\|_F$$

Frobenius norm

$$\|\mathrm{A}\|_F \equiv \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2}\,.$$

A_k and X are both $m \times n$ matrices.

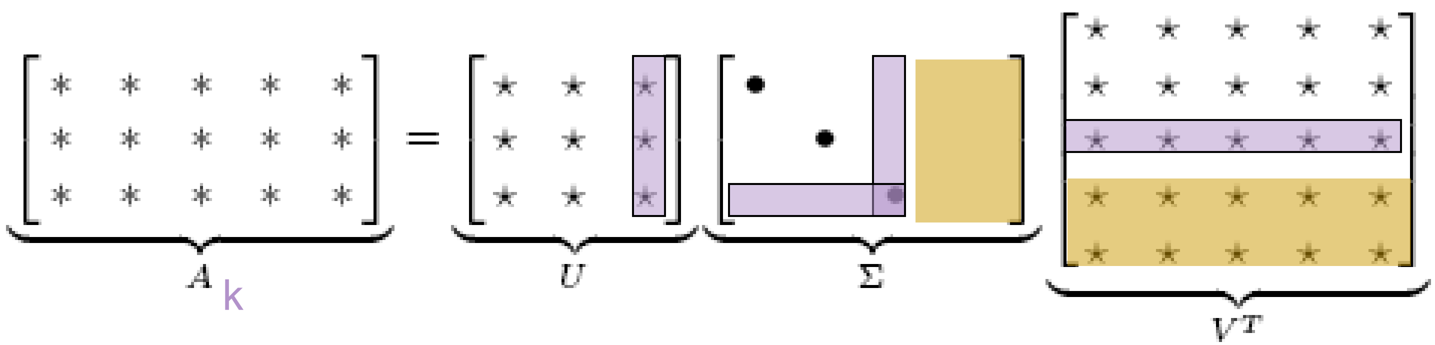
Typically, want $k \ll r$.

Low-rank Approximation

- Solution via SVD

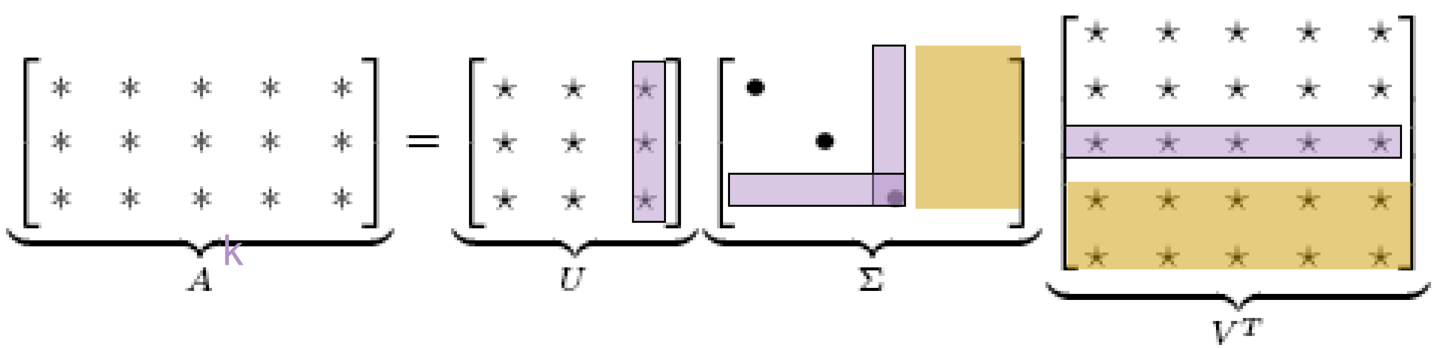
$$A_k = U \operatorname{diag}(\sigma_1, \ldots, \sigma_k, 0, \underbrace{\ldots, 0}_{})V^T$$

set smallest r-k singular values to zero



Reduced SVD

- If we retain only k singular values, and set the rest to 0, then we don't need the matrix parts in brown
- Then Σ is $k \times k$, U is $M \times k$, V^T is $k \times N$, and A_k is $M \times N$
- This is referred to as the reduced SVD
- It is the convenient (space-saving) and usual form for computational applications



Approximation error

- How good (bad) is this approximation?
- It's the best possible, measured by the Frobenius norm of the error:

$$\min_{X:rank(X)=k} \|A-X\|_F = \|A-A_k\|_F = \sigma_{k+1}$$

where the σ_i are ordered such that $\sigma_i \geq \sigma_{i+1}$.

Suggests why Frobenius error drops as *k* increased.

SVD Low-rank approximation

- Whereas the term-doc matrix A may have M=50000, N=10 million (and rank close to 50000)
- We can construct an approximation A_{100} with rank 100.
  - Of all rank 100 matrices, it would have the lowest Frobenius error.
- Great … but why would we??
- Answer: *Latent Semantic Indexing*

C. Eckart, G. Young, *The approximation of a matrix by another of lower rank*. Psychometrika, 1, 211-218, 1936.

Latent Semantic Indexing via the SVD

What it is

- From term-doc matrix A, we compute the approximation A_k.
- There is a row for each term and a column for each doc in A_k
- Thus docs live in a space of $k \ll r$ dimensions
  - These dimensions are not the original axes

Example

- Query: *gold silver truck*
- 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.*

$$
\begin{array}{l}
\text{Terms} \\
\downarrow \\
\text{a} \\ \text{arrived} \\ \text{damaged} \\ \text{delivery} \\ \text{fire} \\ \text{gold} \\ \text{in} \\ \text{of} \\ \text{shipment} \\ \text{silver} \\ \text{truck}
\end{array}
\qquad
\mathbf{A} = \begin{array}{c} \begin{array}{ccc} d1 & d2 & d3 \\ \downarrow & \downarrow & \downarrow \end{array} \\
\begin{bmatrix}
1 & 1 & 1 \\
0 & 1 & 1 \\
1 & 0 & 0 \\
0 & 1 & 0 \\
1 & 0 & 0 \\
1 & 0 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 0 & 1 \\
0 & 2 & 0 \\
0 & 1 & 1
\end{bmatrix}\end{array}
\qquad
\mathbf{q} = \begin{array}{c} q \\ \downarrow \\
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}\end{array}
$$

Compute the SVD of A

$A = USV^T$

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

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

Rank 2 Approximation

$$k = 2$$

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

$$S \approx S_k = \begin{bmatrix} 4.0989 & 0.0000 \\ 0.0000 & 2.3616 \end{bmatrix}$$

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

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

Computing the query vector

$$A^T = (USV^T)^T = VSU^T$$

$$A^T US^{-1} = VSU^T US^{-1}$$

$$V = A^T US^{-1}$$

$$d = d^T US^{-1}$$

$$q = q^T US^{-1}$$

Thus, in the reduced *k*-dimensional space we can write

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

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

Computing the query vector

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

$$k = 2$$

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

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

Computing the similarity

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

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

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

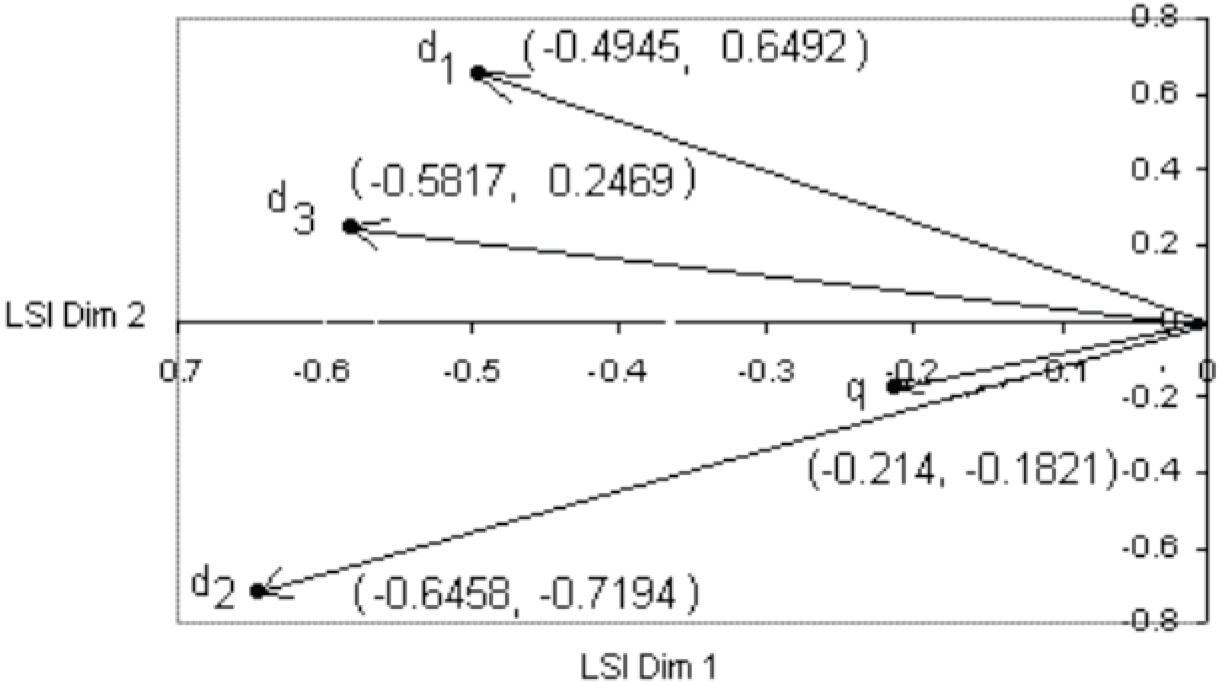
$$\text{sim}(q, d_2) = \frac{(-0.2140)(-0.6458) + (-0.1821)(-0.7194)}{\sqrt{(-0.2140)^2 + (-0.1821)^2}\sqrt{(-0.6458)^2 + (-0.7194)^2}} = 0.9910$$

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

Ranking documents in descending order

$$d_2 > d_3 > d_1$$

LSI Query Document Vectors



Resources

- *Introduction to Information Retrieval*, chapter 18.
- Some slides were adapted from
  - Prof. Dragomir Radev's lectures at the University of Michigan:
    - http://clair.si.umich.edu/~radev/teaching.html
  - the book's companion website:
    - http://nlp.stanford.edu/IR-book/information-retrieval-book.html
  - SVD and LSI Tutorial:
    - http://www.miislita.com/information-retrieval-tutorial/svd-lsi-tutorial-4-lsi-how-to-calculations.html