Alternative feature spaces

Basics: Eigenvector, Eigenvalue

For a square matrix *A*:

$A\mathbf{x} = \lambda\mathbf{x}$

where **x** is a vector (eigenvector), and λ a scalar (eigenvalue)

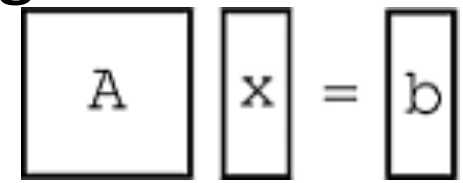
$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \lambda \begin{bmatrix} x \\ y \end{bmatrix},$$

$$\begin{pmatrix} 3 & -1 \\ -1 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 2 \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

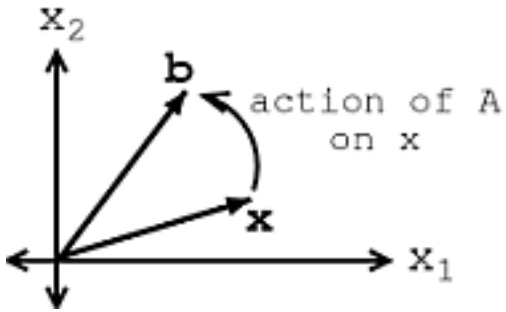
$$\begin{pmatrix} 3 & -1 \\ -1 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = 4 \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

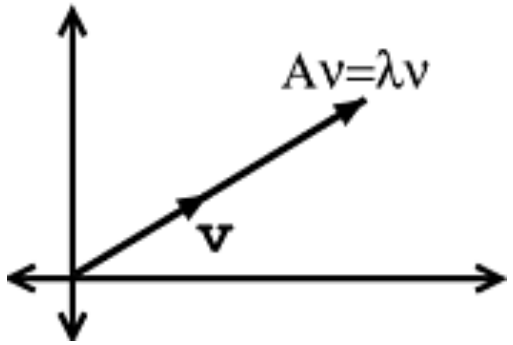
Why using eigenvector?

Linear algebra: $A\,\boldsymbol{x} = b$



Eigenvector: $A\,\boldsymbol{x} = \lambda\,\boldsymbol{x}$





Why using eigenvector

- Eigenvectors are orthogonal (seen as being independent)
- Eigenvector represents the basis of the original vector *A*
- Useful for
  - Solving linear equations
  - Determine the natural frequency of bridge
  - …

Latent Semantic Analysis

- **Lexical matching at term level inaccurate (claimed)**
- **Polysemy – words with number of ‘meanings’ – term matching returns irrelevant documents – impacts precision**
- **Synonomy – number of words with same ‘meaning’ – term matching misses relevant documents – impacts recall**
- **Fewer dimensions → dimension reduction**
- **Keep k strongest dimensions: remove noise**

LSA assumes that there exists a LATENT structure in word usage – obscured by variability in word choice

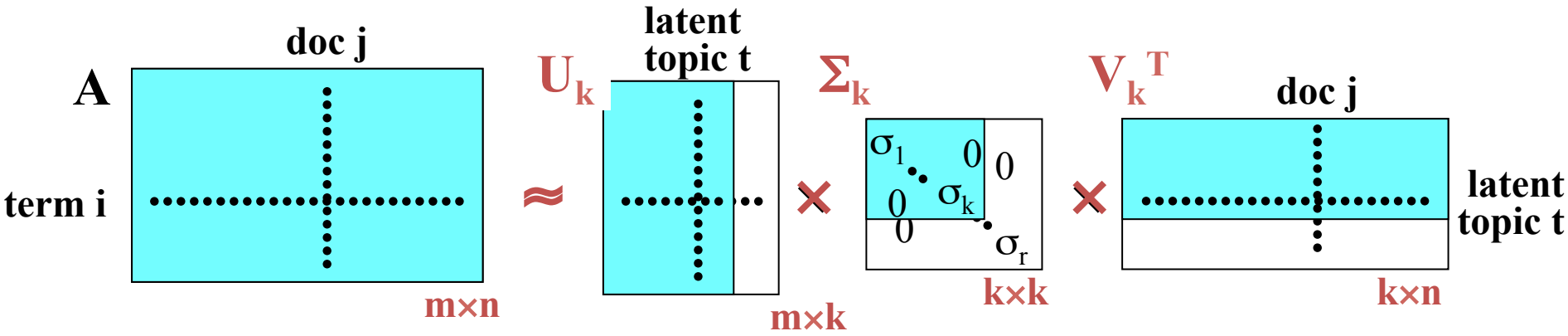
Word usage defined by term and document co-occurrence – matrix structure

Latent Semantic Indexing (LSI)

[Deerwester et al. 1990]

A is the m×n term-document matrix. Then:

- U and U_k are the m×r and m×k term-topic similarity matrices,
- V and V_k are the n×r and n×k document-topic similarity matrices,
- $A \times A^T$ and $A_k \times A_k^T$ are the term-term similarity matrices,
- $A^T \times A$ and $A_k^T \times A_k$ are the document-document similarity matrices



mapping of m×1 vectors into latent-topic space:

$$d_j : U_k^T \times d_j =: d_j'$$
$$q : U_k^T \times q =: q'$$

scalar-product similarity in latent-topic space: $d_j'^T \times q' = ((\Delta_k V_k^T)_{*j})^T \times q'$

LSI: Indexing and Query Processing

- The matrix $\Delta_k V_k^T$ corresponds to a **„topic index“** and is stored in a suitable data structure.
  Instead of $\Delta_k V_k^T$ the simpler **index V_k^T** could be used.

- Additionally the **term-topic mapping U_k** must be stored.

- A **query q** (an m×1 column vector) in the term vector space is transformed into query **$q' = U_k^T \times q$** (a k×1 column vector) and evaluated in the topic vector space (i.e. V_k)
  (e.g. by scalar-product similarity $V_k^T \times q'$ or cosine similarity)

- A **new document d** (an m×1 column vector) is transformed into **$d' = U_k^T \times d$** (a k ×1 column vector) and appended to the „index“ V_k^T as an additional column **(„folding-in“)**

LSI: Example

m=6 terms
- t1: bak(e,ing)
- t2: recipe(s)
- t3: bread
- t4: cake
- t5: pastr(y,ies)
- t6: pie

n=5 documents
- d1: How to bake bread without bread recipes
- d2: The classic art of Viennese Pastry
- d3: Numerical recipes: the art of scientific computing
- d4: Breads, pastries, pies and cakes: quantity baking recipes
- d5: Pastry: a book of best French recipes

$$A = \begin{pmatrix} 0.4446 & 0.0000 & 0.0000 & 0.3422 & 0.0000 \\ 0.1083 & 0.0000 & 1.0000 & 0.0833 & 0.4002 \\ 0.8892 & 0.0000 & 0.0000 & 0.3422 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.6010 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 & 0.1908 & 0.9164 \\ 0.0000 & 0.0000 & 0.0000 & 0.6010 & 0.0000 \end{pmatrix}$$

LSI: Example

$$A = \begin{pmatrix} -0.1337 & 0.4385 & -0.0916 & -0.0858 \\ -0.4039 & 0.0798 & 0.9089 & 0.06680 \\ -0.1909 & 0.7045 & -0.1092 & -0.5105 \\ -0.1336 & 0.3000 & -0.1298 & 0.5997 \\ -0.8642 & -0.3535 & -0.3464 & -0.0906 \\ -0.1336 & 0.3000 & -0.1298 & 0.5997 \end{pmatrix} \quad \mathrm{U}$$

$$\times \begin{pmatrix} 1.4543 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.1764 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.9980 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.7115 \end{pmatrix} \quad \Delta$$

$$\times \begin{pmatrix} -0.1878 & -0.5942 & -0.2777 & -0.3233 & -0.65571 \\ 0.7061 & -0.3005 & 0.0678 & 0.5873 & -0.2482 \\ -0.0396 & -0.3471 & 0.9107 & -0.2155 & 0.0464 \\ -0.6817 & -0.1274 & 0.0940 & 0.7100 & -0.0792 \end{pmatrix} \quad \mathrm{V^T}$$

LSI: Example

$$A_2 = \begin{pmatrix} 0.4008 & -0.0395 & 0.0890 & 0.3659 & -0.0006 \\ 0.1766 & 0.3209 & 0.1695 & 0.2451 & 0.3619 \\ 0.6373 & -0.0841 & 0.1333 & 0.5766 & -0.0237 \\ 0.2857 & 0.0094 & 0.0779 & 0.2701 & 0.0398 \\ -0.0576 & 0.8718 & 0.3209 & 0.1621 & 0.9273 \\ 0.2857 & 0.0094 & 0.0779 & 0.2701 & 0.0398 \end{pmatrix} = U_2 \times \Delta_2 \times V_2^T$$

LSI: Example

query q: baking bread
$q = (1\ 0\ 1\ 0\ 0\ 0)^T$

transformation into topic space with k=2
$q' = U_k^T \times q = (-0.3246\ 1.1430)^T$

scalar product similarity in topic space with k=2:
$sim(q, d1) = Vk_{*1}^T \times q' \approx 0.87$ $sim(q, d2) = Vk_{*2}^T \times q \approx -0.15$
$sim(q, d3) = Vk_{*3}^T \times q' \approx 0.17$ etc.

Folding-in of a new document d6:
algorithmic recipes for the computation of pie
$d6 = (0\ \ 0.7071\ \ 0\ \ 0\ \ 0\ \ 0.7071)^T$

transformation into topic space with k=2
$d6' = U_k^T \times d6 \approx (-0.3801\ 0.2686)$

d6' is appended to Vk^T as a new column