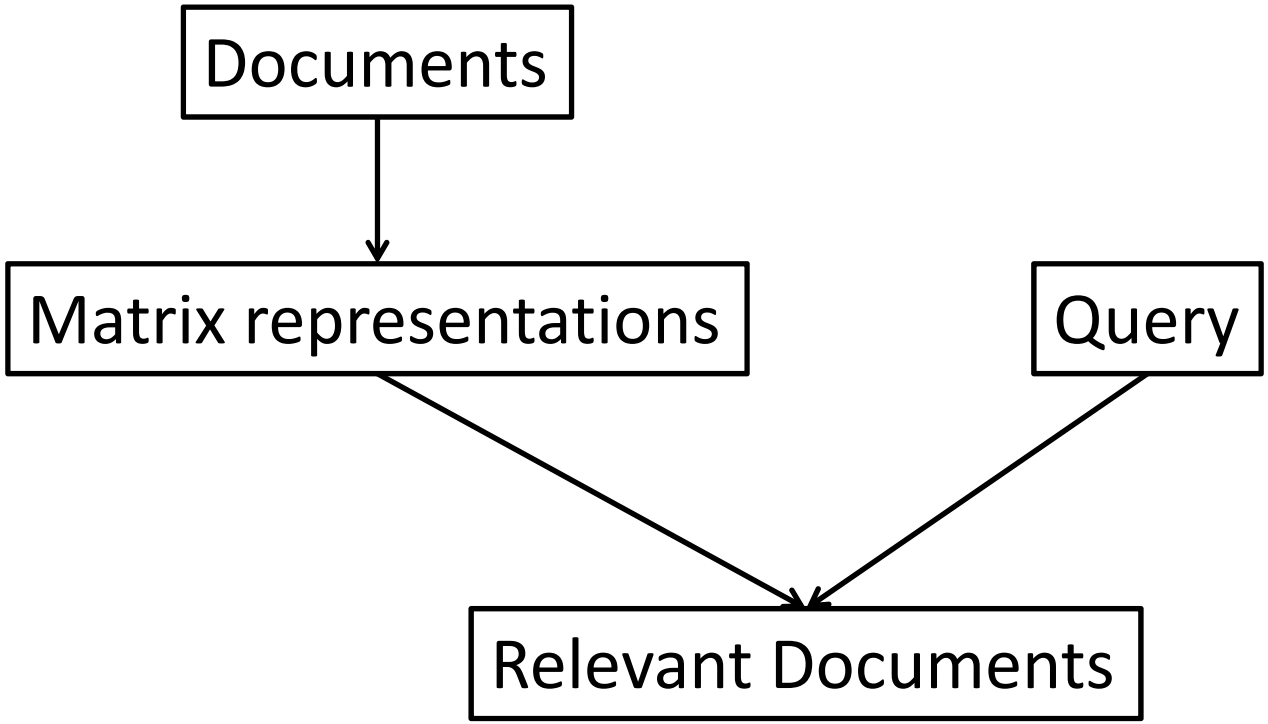
CS475 / CS675
Lecture 21: July 12, 2016

Application: Information Retrieval

Information Retrieval



Document Representation

- Vector space model
- A document collection composed of n documents that are indexed by m terms can be represented as an $m \times n$ term-by-document matrix A.
  - $a_{ij} =$ frequency at which term i occurs in document j
  - Picture:

Example

- *Understanding Search Engines: Mathematical Modeling and Text Retrieval* (2005) Berry & Browne

Terms	**Documents**
T1:	D1:
T2:	D2:
T3:	D3:
T4:	D4:
T5:	D5:
T6:	D6:
T7:	D7:
T8:	
T9:	

Example

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Example

- Collection:
  - Documents = book titles (7)
  - Terms = words used in the titles (9)
  - Term-by-document matrix is 9x7

- Queries are represented as 9x1 vectors
  - E.g., to retrieve books on “child proofing”

$$q = [0\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 0]^T$$

Simple Query Matching

- How do we find a match?
  - Query matching is about finding the documents most similar to the query
  - A common similarity measure is the cosine of the angle between the query vector and the document vector
  - Let a_j be the j^{th} column of A (i.e., j^{th} document). The cosine of the angle between a_j & q is

$$\cos\theta_j = \frac{a_j^T q}{\left||a_j|\right| \left||q|\right|} \qquad j = 1,2,\dots,n$$

Simple Query Matching

- Note: the dot-product essentially counts the frequency of the terms in q that appear in a_j
- The j^{th} document such that $\cos(\theta_j) \geq$ threshold will be judged as relevant
- In the example:

$$\cos\theta_1 = \cos\theta_4 = \cos\theta_7 = 0$$
$$\cos\theta_2 = \cos\theta_3 = 0.4082$$
$$\cos\theta_5 = \cos\theta_6 = 0.5$$

- Suppose threshold $= 0.5$. Then D_5 and D_6 are selected.

Simple Query Matching

- Notes
  1. D_7 is correctly excluded
  2. D_6 is incorrectly included
  3. $D_1 - D_4$ are relevant, but not included

- Conclusion: representation of documents based solely on term frequency is not adequate

Document embedding

- Idea: Embed each document into k dimensions where

$$k \leq \min(m, n)$$

  - E.g., $k = 2$ (2 dimensions)
  - Thus each doc_j is represented by (x_j, y_j)
  - The values x_j, y_j indicate the coordinates of the document in each dimension

Embedding

- Two-dimensional representation

Embedding

- The query "Child Home Safety" is clearly related to the document cluster $\{D_1, D_2, D_3, D_4\}$.
- Coordinates do not explicitly reflect term frequency within documents
- They model "global" usage patterns of terms so that related documents that may not share common terms are still represented by nearby vectors in k-dimensional subspace.
- Grouping of terms/docs according to concepts

Low-Rank Approximation by QR

Normalized $A = \begin{pmatrix} 0 & 0.5774 & 0 & 0.4472 & 0.7071 & 0 & 0.7071 \\ 0 & 0.5774 & 0.5774 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.7071 & 0.7071 \\ 0 & 0 & 0 & 0.4472 & 0 & 0 & 0 \\ 0 & 0.5774 & 0.5774 & 0 & 0 & 0 & 0 \\ 0.7071 & 0 & 0 & 0.4472 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.7071 & 0.7071 & 0 \\ 0 & 0 & 0.5774 & 0.4472 & 0 & 0 & 0 \\ 0.7071 & 0 & 0 & 0.4472 & 0 & 0 & 0 \end{pmatrix}$

Rank-4 approx by QR: $A_4 = \begin{pmatrix} 0 & 0.5774 & 0 & 0.4472 & 0.5983 & 0 & 0.5983 \\ 0 & 0.5774 & 0.5774 & 0 & 0.0544 & 0 & 0.0544 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.4472 & 0.2176 & 0 & 0.2176 \\ 0 & 0.5774 & 0.5774 & 0 & 0.0544 & 0 & 0.0544 \\ 0.7071 & 0 & 0 & 0.4472 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5774 & 0.4472 & -0.1088 & 0 & -0.1088 \\ 0.7071 & 0 & 0 & 0.4472 & 0 & 0 & 0 \end{pmatrix}$

Low-Rank Approximation by SVD

Normalized

$$A = \begin{pmatrix} 0 & 0.5774 & 0 & 0.4472 & 0.7071 & 0 & 0.7071 \\ 0 & 0.5774 & 0.5774 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.7071 & 0.7071 \\ 0 & 0 & 0 & 0.4472 & 0 & 0 & 0 \\ 0 & 0.5774 & 0.5774 & 0 & 0 & 0 & 0 \\ 0.7071 & 0 & 0 & 0.4472 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.7071 & 0.7071 & 0 \\ 0 & 0 & 0.5774 & 0.4472 & 0 & 0 & 0 \\ 0.7071 & 0 & 0 & 0.4472 & 0 & 0 & 0 \end{pmatrix}$$

Rank-4 SVD:

$$A_4 = \begin{pmatrix} -0.0018 & 0.5958 & -0.0148 & 0.4523 & 0.6974 & 0.0102 & 0.6974 \\ -0.0723 & 0.4938 & 0.6254 & 0.0743 & 0.0121 & -0.0133 & 0.0121 \\ 0.0002 & -0.0067 & 0.0052 & -0.0013 & 0.3569 & 0.7036 & 0.3569 \\ \mathbf{0.1968} & 0.0512 & 0.0064 & 0.2179 & 0.0532 & -0.0540 & 0.0532 \\ -0.0723 & 0.4938 & 0.6254 & 0.0743 & 0.0121 & -0.0133 & 0.0121 \\ 0.6315 & -0.0598 & 0.0288 & 0.5291 & -0.0008 & 0.0002 & -0.0008 \\ 0.0002 & -0.0067 & 0.0052 & -0.0013 & 0.3569 & 0.7036 & 0.3569 \\ \mathbf{0.2151} & 0.2483 & 0.4347 & 0.2262 & -0.0359 & 0.0394 & -0.0359 \\ 0.6315 & -0.0598 & 0.0288 & 0.5291 & -0.0008 & 0.0002 & -0.0008 \end{pmatrix}$$

Low-Rank Approximation

- The entries of A are nonnegative, but A_k may contain negative entries

- In our example, column 1 of A does not have "Health" and "Safety". But column 1 of A_k does have these 2 terms included

Query Matching

- Compare query vector q with columns in A_k.
- Let $e_j = j^{th}$ column of I. Then $A_k e_j = j^{th}$ column of A_k.
- As before, compute the cosines:

$$\cos\theta_j = \frac{\left(A_k e_j\right)^T q}{\left|\left|A_k e_j\right|\right|_2 \left|\left|q\right|\right|_2}$$

- Write $A_k e_j = U_k \Sigma_k V_k^T e_j = U_k S_j$ → Coords of $A_k e_j$ in concept space

Concept space

Query Matching

- Then $\left(A_k e_j\right)^T q = \left(U_k S_j\right)^T q = S_j^T (U_k^T q)$

Coords of query in concept space

- Hence $\cos\theta_j = \frac{S_j^T (U_k^T q)}{\left|\left|S_j\right|\right|_2 ||q||_2}$
- Notes:
  1. No need to actually form A_k
  2. $\left|\left|S_j\right|\right|$ can be computed once and stored

Interpretation

- The SVD of A (or A_k) converts documents in a "concept" space, whose basis is given by the columns of U_k
- Queries are converted to the same "concept" space and compared to the documents for similarity
- The concept space can be made significantly smaller than the document space i.e., $k \ll n$
- Documents that are related to the query but which may not contain the search words will also be returned.