

Document Retrieval

- ◆ Given:
 - Corpus of documents and query
- ◆ Task:
 - Find documents relevant to query

1

Email Searching

- ◆ Given:
 - Corpus of email messages
 - Query of search words
- ◆ Task:
 - Find messages with same meaning as search words

2

Representation

- ◆ Dictionary: n -tuple of words
 - ◆ Document: n -tuple of integers
- $$\vec{x} = \langle x_1, x_2, \dots, x_n \rangle$$
- x_i = number of times word i occurs in message

3

Example

- ◆ Human-computer interaction messages:
 - c_1 : “**human** machine interface computer applications”
 - c_2 : “survey of user opinion of computer **system** response time”
 - c_3 : “**system** and **human system** engineering testing of EPS”

...

4

Example

- ◆ Graph-theory messages:
 - m_1 : “generation of random, binary, unordered *trees*”
 - m_2 : “intersection *graph* of paths in *trees*”
 - m_3 : “*graph* minors: a survey”
 - ...

5

◆ Dictionary:

$\langle \text{human, system, trees, graph, ...} \rangle$

◆ Documents:

$$\vec{c}_1 = \langle 1, 0, 0, 0, \dots \rangle \quad \vec{m}_1 = \langle 0, 0, 1, 0, \dots \rangle$$

$$\vec{c}_2 = \langle 0, 1, 0, 0, \dots \rangle \quad \vec{m}_2 = \langle 0, 0, 1, 1, \dots \rangle$$

$$\vec{c}_3 = \langle 1, 2, 0, 0, \dots \rangle \quad \vec{m}_3 = \langle 0, 0, 0, 1, \dots \rangle$$

◆ Corpus:

$$X = \{ \vec{c}_1, \vec{c}_2, \vec{c}_3, \vec{m}_1, \vec{m}_2, \vec{m}_3, \dots \}$$

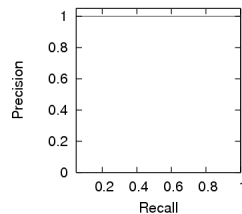
6

Measuring Success

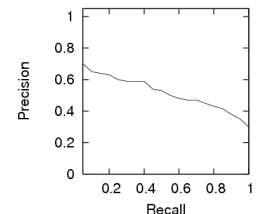
- ◆ Recall
 - Proportion of relevant documents that are returned
 - Breadth of results
- ◆ Precision
 - Proportion of returned documents that are relevant
 - Accuracy of results

7

Precision vs. Recall



Optimal behavior.



Real-world behavior.

8

Keyword Indexing

- ◆ Relevance: common word usage
- ◆ Ex: SMART by Gerard Salton
- ◆ Fundamental limitations
 - Synonymy
 - Polysemy

9

Synonymy

- ◆ Multiple words with the same meaning
dog = canine
- ◆ Lowers recall – relevant documents missed if lacking index word
- ◆ People use same terms less than 20% of the time

10

Polysemy

- ◆ One word has multiple meanings
 - Bank: financial institution
 - Bank: to rely on
- ◆ No way to disambiguate word meaning
- ◆ Lowers precision – irrelevant documents returned

11

Latent Semantic Indexing

- ◆ Performs conceptual matching
 - Identify synonyms using word co-occurrence
 - Retrieve documents with query words, or their synonyms!
 - Compensates for different word usage

12

Indexing

- ◆ Create association matrix
 - A_{ij} is frequency of i^{th} term in j^{th} document
 - Rows correspond to terms
 - Columns correspond to documents in X

13

Terms	Documents						
	c_1	c_2	c_3	m_1	m_2	m_3	...
<i>human</i>	1	0	1	0	0	0	
<i>system</i>	0	1	2	0	0	0	
<i>trees</i>	0	0	0	1	1	0	
<i>graph</i>	0	0	0	0	1	1	
...							

14

Indexing

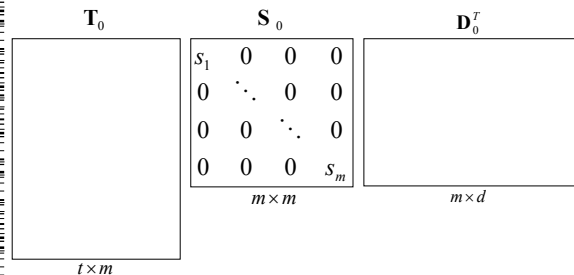
- ◆ Decompose matrix using Singular Value Decomposition (SVD)

$$\mathbf{A} = \mathbf{T}_0 \mathbf{S}_0 \mathbf{D}_0^T$$

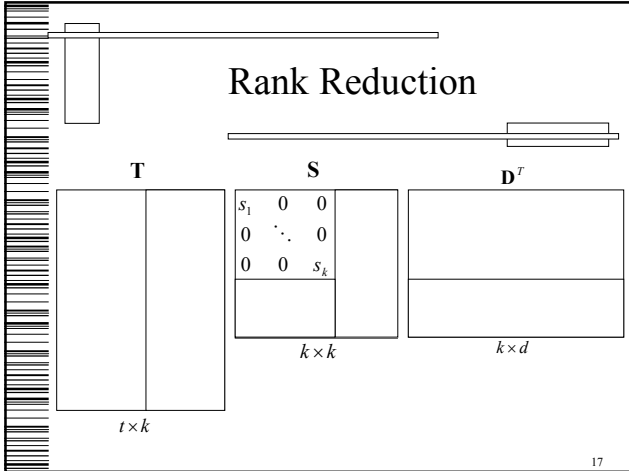
- \mathbf{T}_0 and \mathbf{D}_0 hold singular vectors for terms and documents
- \mathbf{S}_0 is diagonal matrix of singular values
- Non-zero singular values are factors

15

Decomposition

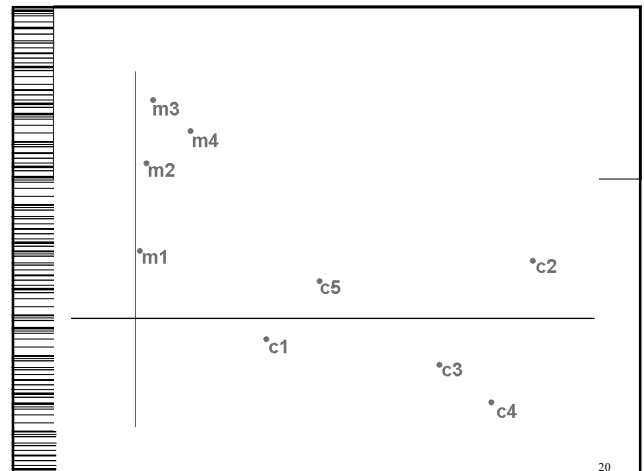


16



- ### Indexing
- ◆ Reduce the rank
 - Truncate all but k most *significant* factors
- $$\hat{A} = TSD^T$$
- \hat{A} is better model of corpus
 - Captures “signal” of co-occurrence
 - Disregards “noise” of particular usage
- 18

- ### Example
- ◆ What if we truncate example association matrix for $k=2$ factors?
 - ◆ We can interpret this geometrically...
- 19



Retrieval

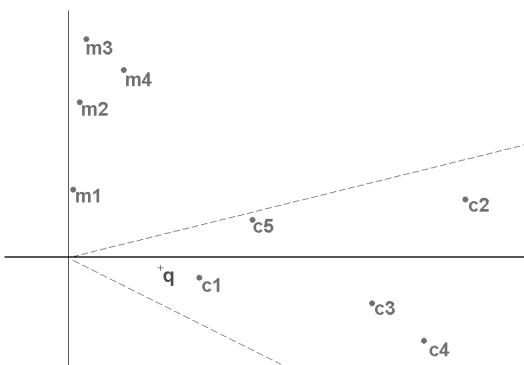
- ◆ Given query, create document vector \vec{q}
- ◆ Transform \vec{q} into reduced space
$$\vec{q}_k = \vec{q}^T \mathbf{TS}^{-1}$$
- ◆ Compare \vec{q}_k to document vectors in $\hat{\mathbf{A}}$
 - Return documents with small cosine

21

Example

- ◆ Query: “human computer interaction”
- ◆ Build document vector
$$\vec{q} = \langle 1, 0, 0, 0, \dots \rangle$$
- ◆ Transform query into \mathbf{R}^2
$$\vec{q}_2 = \vec{q}^T \mathbf{TS}^{-1}$$

22



23

Tuning the Model

- ◆ Model complexity related to number of factors
 - Too few: over-generalize and ignore data
 - Too many: over-fit and end up with keyword indexing

24

And it works...

- ◆ LSI outperforms keyword-matching
 - Ignores word order in documents!

- ◆ Some open problems
 - Does not fully address polysemy
 - Need way to find optimal rank of \hat{A}

25

References

- [1] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.
- [2] G. Furnas, T. Landauer, L. Gomez, and T. Dumais. Statistical semantics: Analysis of the potential performance of keyword information systems. *Bell System Technical Journal*, 62(6):1753–1806, 1983.
- [3] Michael W. Berry and Murray Browne. *Understanding Search Engines*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999.
- [4] Michael W. Berry, Susan T. Dumais, and Gavin W. O'Brien. Using linear algebra for intelligent information retrieval. Technical Report UT-CS-94-270, 1995.

26