SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COURSE NAME : 19CS732 INFORMATION RETRIEVAL TECHNIQUES

IVYEAR / VII SEMESTER

Unit 2- MODELING AND RETRIEVAL EVALUATION

Topic 5 : Latent Semantic Indexing Model

Problem

- Optimizing content for organic search visibility has evolved in line with Google’s advancements.
- Equally, search engines still have challenges when trying to understand the meaning of words in context.

Latent Semantic Indexing Model

- 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).
  - Compute document similarity based on the **inner product** in this **latent semantic space**

Latent Semantic Indexing Model-Cont..

- Application of Latent Semantic Analysis (LSA) to Information Retrieval
- Motivations
  - Unreliable evidence
  - *synonomy:* Many words refer to same object

    Affects recall
  - *polysemy:* Many words have multiple meanings

    Affects precision

The document ranking problem

Sample Term by Document matrix

	access	*document*	*retrieval*	*information*	*theory*	*database*	*indexing*	*computer*	REL	MATCH
Doc 1	x	x	x			x	x		R	
Doc 2				x*	x			x*		M
Doc 3			x	x*				x*	R	M

Query: "IDF in *computer*-based *information* look-up"

Source: Deerwater et al. 1990

LSA Solution

- Terms are overly noisy
  - analogous to overfitting in Term-by-Document matrix
- Terms and documents should be represented by vectors in a "latent" semantic space
- LSI essentially infers knowledge from co-occurrence of terms
- Assume "errors" (sparse data, non-co-occurrences) are normal and account for them

LSA Methods

- Start with a Term-by-Document matrix (A, like fig. 15.5)
- Optionally weight cells
- Apply Singular Value Decomposition:
  - t = # of terms
  - d = # of documents
  - n = min(t, d)

$$A_{t\times d} = T_{t\times n} \times S_{n\times n} \times (D_{d\times n})^T$$

- Approximate using k (semantic) dimensions:

$$\hat{A}_{t\times d} = T_{t\times k} \times S_{k\times k} \times (D_{d\times k})^T$$

LSA Methods –Cont..

- So that the Euclidean distance is minimized (hence, a least squares method)
- Each row of T is a measure of similarity for a term to a semantic dimension
- Likewise for D

LSA Application

- Querying for Information Retrieval: query is a psuedo-document:
  - weighted sum over all terms in query of rows of T
  - compare similarity to all documents in D using cosine similarity measure
- Document similarity: vector comparison of D
- Term similarity: vector comparison of T

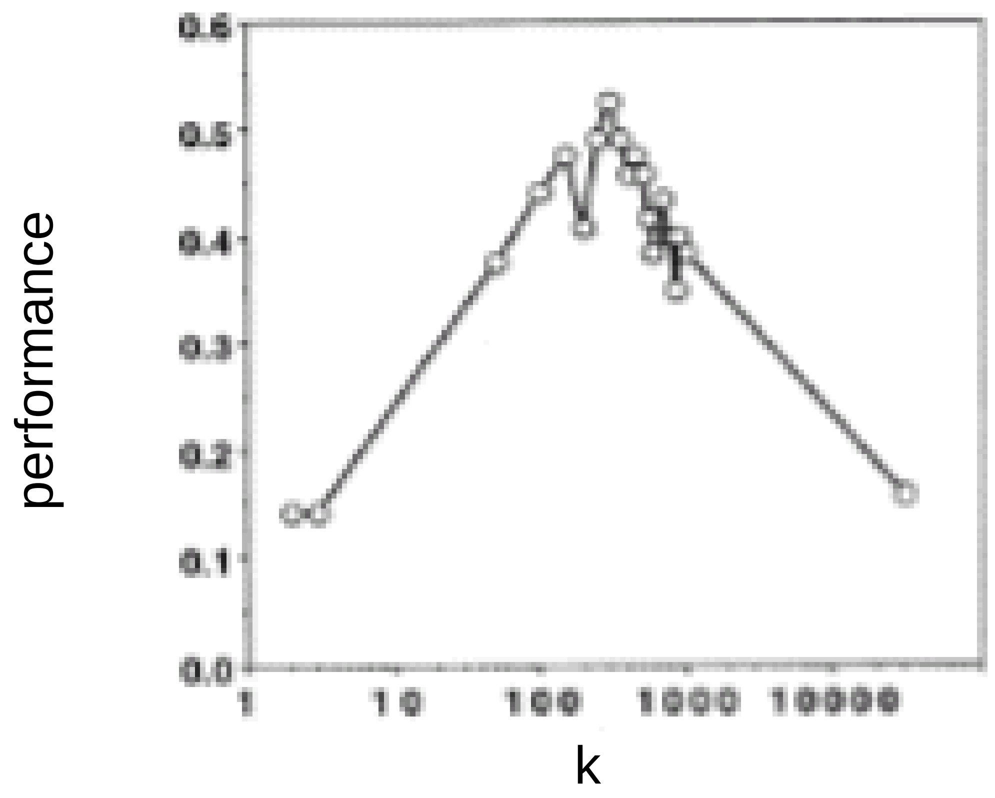
LSA Application -Cont..

Choosing k is difficult

- commonly k = 100, 150, 300 or so
- overfitting (superfluous dimensions) vs. underfitting (not enough dimensions)

What are the k semantic dimensions?

- undefined



Considerations of LSA

- Conceptually high recall: query and document terms may be disjoint
- Polysemes not handled well
- LSI: Unsupervised/completely automatic
- Language independent
- CL-LSI: Cross-Language LSI
  - weakly trained
- Computational complexity is high
  - optimization: random sampling methods
- Formal Linear Algebra foundations
- Models language acquisition in children

Activity

Latent Semantic Indexing (LSI)

Main idea

- map each **document** into some ‘**concepts**’
- map each **term** into some ‘**concepts**’

‘**Concept**’ : ~ a set of terms, with weights.

For example, DBMS_concept:
“data” (0.8),
“system” (0.5),
“retrieval” (0.6)

Latent Semantic Indexing (LSI)

~ pictorially (after) ~

term-concept matrix

	database concept	medical concept
data	1	
system	1	
retrieval	1	
lung		1
ear		1

… and

document-concept matrix

	database concept	medical concept
doc1	1	
doc2	1	
doc3		1
doc4		1

Latent Semantic Indexing (LSI)

Q: How to search, e.g., for "system"?
A: find the corresponding concept(s); and the corresponding documents

	database concept	medical concept
data	1	
system	1	
retrieval	1	
lung		1
ear		1

	database concept	medical concept
doc1	1	
doc2	1	
doc3		1
doc4		1

Latent Semantic Indexing (LSI)

Works like an **automatically constructed thesaurus**

We may retrieve documents that **DON'T** have the term "system", but they contain almost everything else ("data", "retrieval")

LSI - Discussion

Great idea,

- to derive 'concepts' from documents
- to build a 'thesaurus' automatically
- to reduce dimensionality (down to few "concepts")

How does LSI work?

Uses Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD)

Motivation

Problem #1

Find “concepts” in matrices

Problem #2

Compression / dimensionality reduction

	bread	lettuce	tomatos	beef	chicken
vegetarians	1	1	1		
	2	2	2		
	1	1	1		
	5	5	5		
meat eaters				2	2
				3	3
				1	1

SVD is a powerful, generalizable technique.

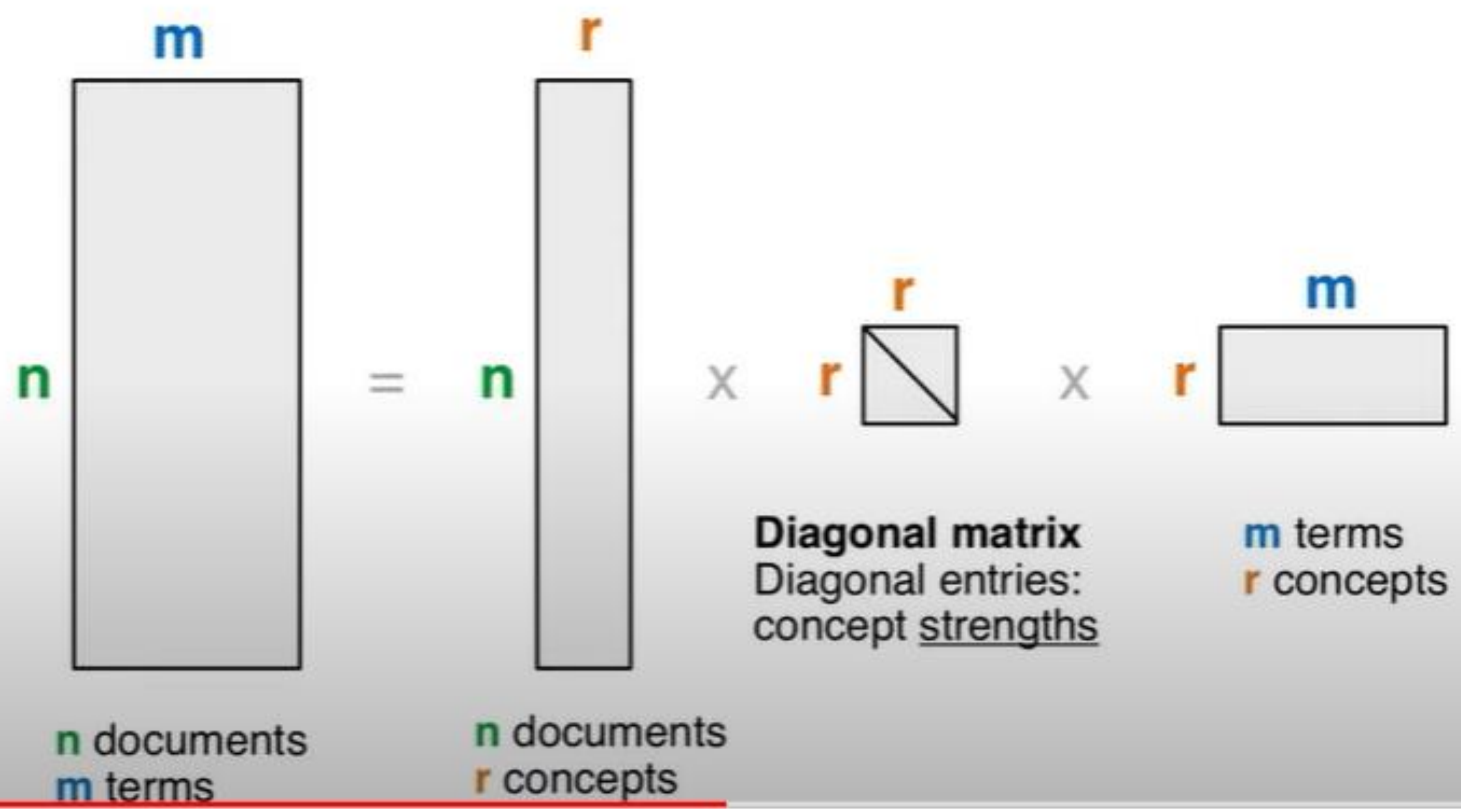
Songs / Movies / Products

Customers

1	1	1		
2	2	2		
1	1	1		
5	5	5		
			2	2
			3	3
			1	1

SVD Definition (pictorially)

$$\boxed{A_{[n \times m]} = U_{[n \times r]} \, \Lambda_{[r \times r]} \, (V_{[m \times r]})^T}$$



SVD Definition (in words)

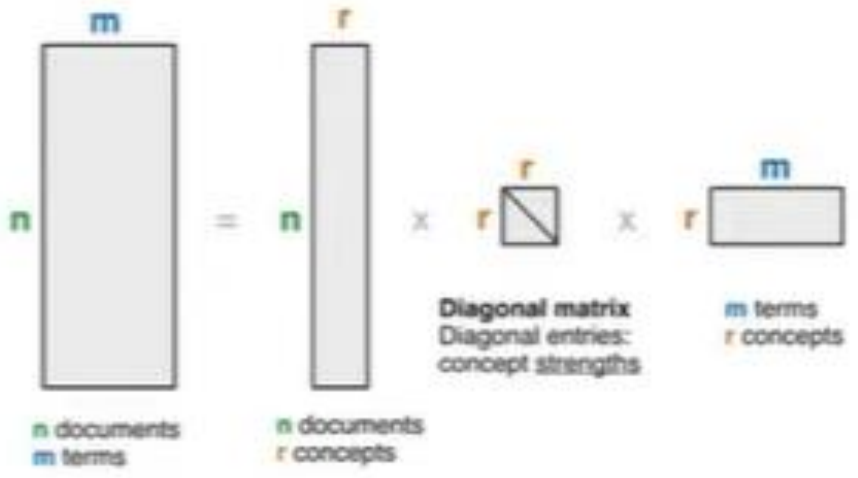
$$\boxed{A_{[n \times m]} = U_{[n \times r]}\, \Lambda_{[r \times r]}\, (V_{[m \times r]})^T}$$

A: n x m matrix
  e.g., n documents, m terms

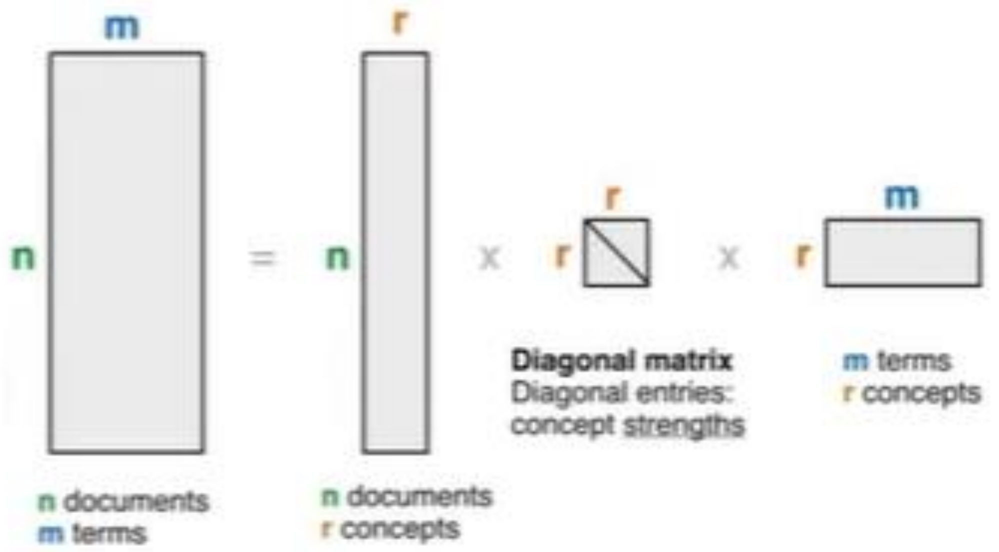
U: n x r matrix
  e.g., n documents, r concepts

Λ: r x r diagonal matrix
  r : rank of the matrix; strength of each 'concept'

V: m x r matrix
  e.g., m terms, r concepts



SVD - Properties



THEOREM [Press+92]:

always possible to decompose matrix **A** into

$\mathbf{A} = \mathbf{U} \Lambda \mathbf{V}^T$

U, Λ, **V**: **unique**, most of the time

U, **V**: column **orthonormal**

i.e., columns are unit vectors, and orthogonal to each other

$\mathbf{U}^T \mathbf{U} = \mathbf{I}$

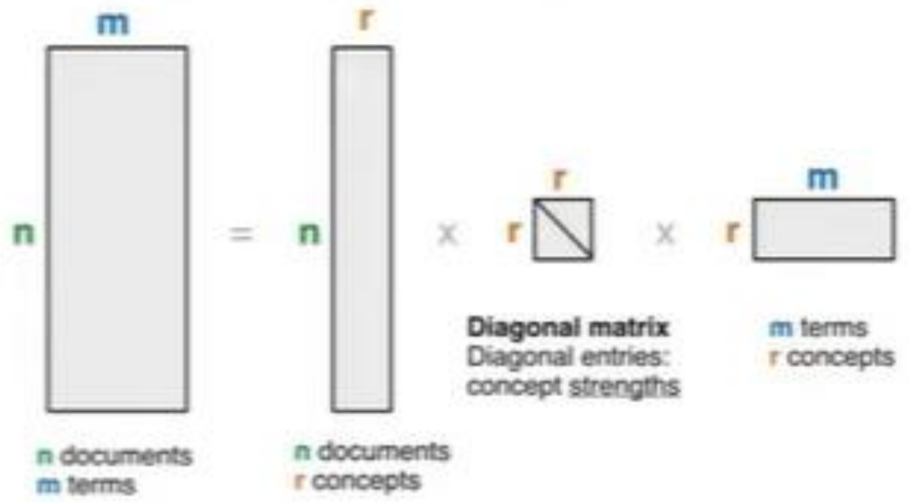
$\mathbf{V}^T \mathbf{V} = \mathbf{I}$

(**I**: identity matrix)

Λ: **diagonal** matrix with non-negative diagonal entires, sorted in decreasing order

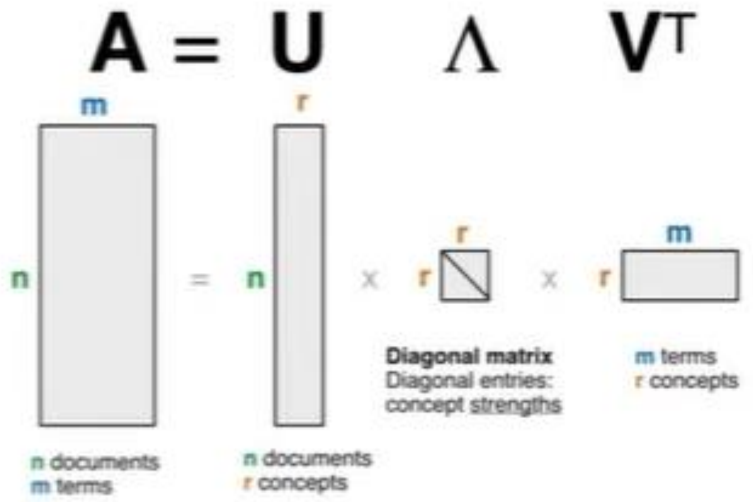
SVD - Example

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

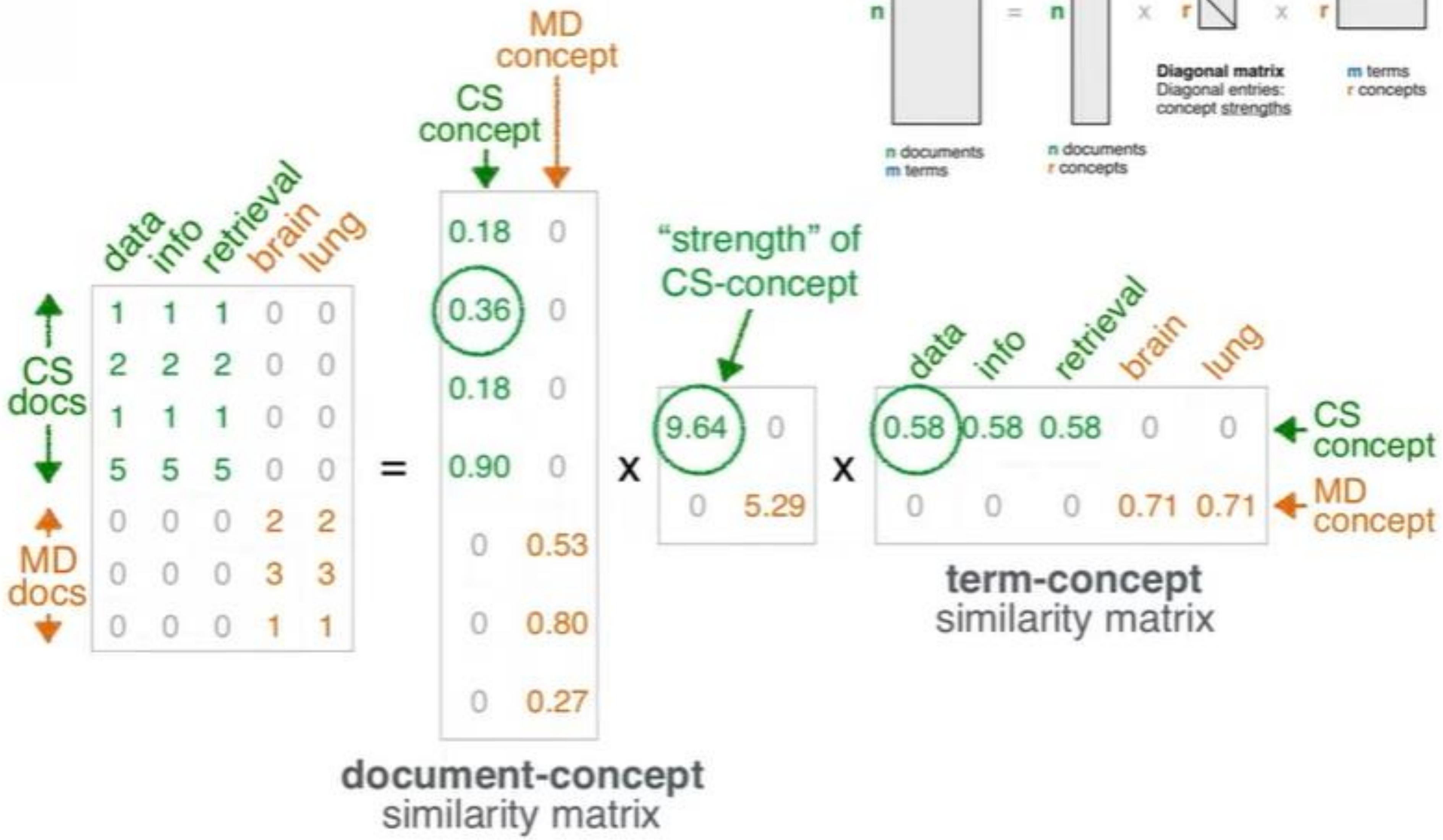


$$\begin{array}{c|ccccc|}
 & \text{data} & \text{info} & \text{retrieval} & \text{brain} & \text{lung} \\
\text{CS docs} & 1 & 1 & 1 & 0 & 0 \\
 & 2 & 2 & 2 & 0 & 0 \\
 & 1 & 1 & 1 & 0 & 0 \\
 & 5 & 5 & 5 & 0 & 0 \\
\text{MD docs} & 0 & 0 & 0 & 2 & 2 \\
 & 0 & 0 & 0 & 3 & 3 \\
 & 0 & 0 & 0 & 1 & 1
\end{array}
=
\begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix}
\times
\begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix}
\times
\begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

SVD - Example



A = U Λ V^T
m
r
n
n
r
r
m
r
n documents
m terms
n documents
r concepts
Diagonal matrix
Diagonal entries:
concept strengths
m terms
r concepts



data info retrieval brain lung
CS docs
MD docs
1 1 1 0 0
2 2 2 0 0
1 1 1 0 0
5 5 5 0 0
0 0 0 2 2
0 0 0 3 3
0 0 0 1 1
=
CS concept
MD concept
0.18 0
0.36 0
0.18 0
0.90 0
0 0.53
0 0.80
0 0.27
x
"strength" of CS-concept
9.64 0
0 5.29
x
data info retrieval brain lung
0.58 0.58 0.58 0 0
0 0 0 0.71 0.71
CS concept
MD concept
term-concept similarity matrix
document-concept similarity matrix

Disadvantages

- Since it is a distributional model, so not an efficient representation, when compared against state-of-the-art methods (say deep neural networks).
- Representation is dense, so hard to index based on indvidual dimensions.
- It is a linear model, so not the best solution to handle non linear dependencies
- The latent topic dimension can not be chosen to arbitrary numbers. It depends on the rank of the matrix, so can't go beyond that.

Advantages

- Easy to implement, understand and use. There are many practical and scalable implementations available
- Performance: LSA is capable of assuring decent results , much better than plain vector space model. It works well on dataset with diverse topics.
- Synonymy: LSA can handle Synonymy problems to some extent (depends on dataset though)
- Runtime : Since it only involves decomposing your term document matrix, it is faster, compared to other dimensionality reduction models

Assessment 1

1. List out the Advantages of Latent Semantic Indexing Model
   a)____________________________________
   b)____________________________________
   c)____________________________________
   d)____________________________________

2. Identify the disadvantages of Latent Semantic Index
   a)____________________________________
   b)____________________________________
   c)____________________________________
   d)____________________________________

TEXT BOOKS:

1. Ricardo Baeza-Yates and Berthier Ribeiro-Neto, —Modern Information Retrieval: The Concepts and Technology behind Search, Second Edition, ACM Press Books, 2011.

2. Ricci, F, Rokach, L. Shapira, B.Kantor, —Recommender Systems Handbook||, First Edition, 2011.

REFERENCES:

1. C. Manning, P. Raghavan, and H. Schütze, —Introduction to Information Retrieval, Cambridge University Press, 2008.

2. Stefan Buettcher, Charles L. A. Clarke and Gordon V. Cormack, —Information Retrieval: Implementing and Evaluating Search Engines, The MIT Press, 2010.

THANK YOU