http://poloclub.gatech.edu/cse6242

CSE6242 / CX4242: Data & Visual Analytics

Text Analytics (Text Mining)

Concepts, Algorithms, LSI/SVD

Duen Horng (Polo) Chau

Assistant Professor

Associate Director, MS Analytics

Georgia Tech

Partly based on materials by
Professors Guy Lebanon, Jeffrey Heer, John Stasko, Christos Faloutsos, Parishit Ram (GT PhD alum; SkyTree), Alex Gray

Text is everywhere

We use documents as primary information artifact in our lives

Our access to documents has grown tremendously thanks to the Internet

- **WWW**: webpages, Twitter, Facebook, Wikipedia, Blogs, ...
- **Digital libraries**: Google books, ACM, IEEE, ...
- Lyrics, closed caption... (youtube)
- Police case reports
- legislation (law)
- reviews (products, rotten tomatoes)
- medical reports (EHR - electronic health records)
- job descriptions

Big (Research) Questions

... in understanding and gathering information from text and document collections

- establish authorship, authenticity; plagiarism detection
- classification of genres for narratives (e.g., books, articles)
- tone classification; sentiment analysis (online reviews, twitter, social media)
- code: syntax analysis (e.g., find common bugs from students' answers)

Popular NLP libraries

- **Stanford NLP**
- **OpenNLP**
- **NLTK** (python)

tokenization, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, parsing

Outline

- **Preprocessing** (e.g., stemming, remove stop words)
- **Document representation** (most common: bag-of-words model)
- **Word importance** (e.g., word count, TF-IDF)
- **Latent Semantic Indexing** (find "concepts" among documents and words), which helps with **retrieval**

To learn more: Prof. Jacob Eisenstein's
CS 4650/7650 Natural Language Processing

Stemming

Reduce words to their **stems** (or base forms)

> **Words**: compute, computing, computer, ...
>
> **Stem**: comput

Several classes of algorithms to do this:

- Stripping suffixes, lookup-based, etc.

http://en.wikipedia.org/wiki/Stemming
Stop words: http://en.wikipedia.org/wiki/Stop_words

Bags-of-words model

Represent each **document** as a **bag of words**, ignoring words' ordering. Why? For simplicity.

- Unstructured text -> a vector of numbers
- e.g., docs: "I like visualization", "I like data".
  - "I": 1,
  - "like": 2,
  - "data": 3,
  - "visualization": 4
- "I like visualization" -> [1, 1, 0, 1]
- "I like data" -> [1, 1, 1, 0]

TF-IDF

(**a word's importance score** in a **document**, among **N documents**)

When to use it? Everywhere you use "word count", you may use TF-IDF.

TF: term frequency
= #appearance a document
(high, if terms appear many times in this document)

IDF: inverse document frequency
= log(**N** / #document containing that term)
(penalize "common" words appearing in almost any documents)

Final score = TF * IDF
(higher score -> more important)

Example: http://en.wikipedia.org/wiki/Tf–idf#Example_of_tf.E2.80.93idf

Vector Space Model

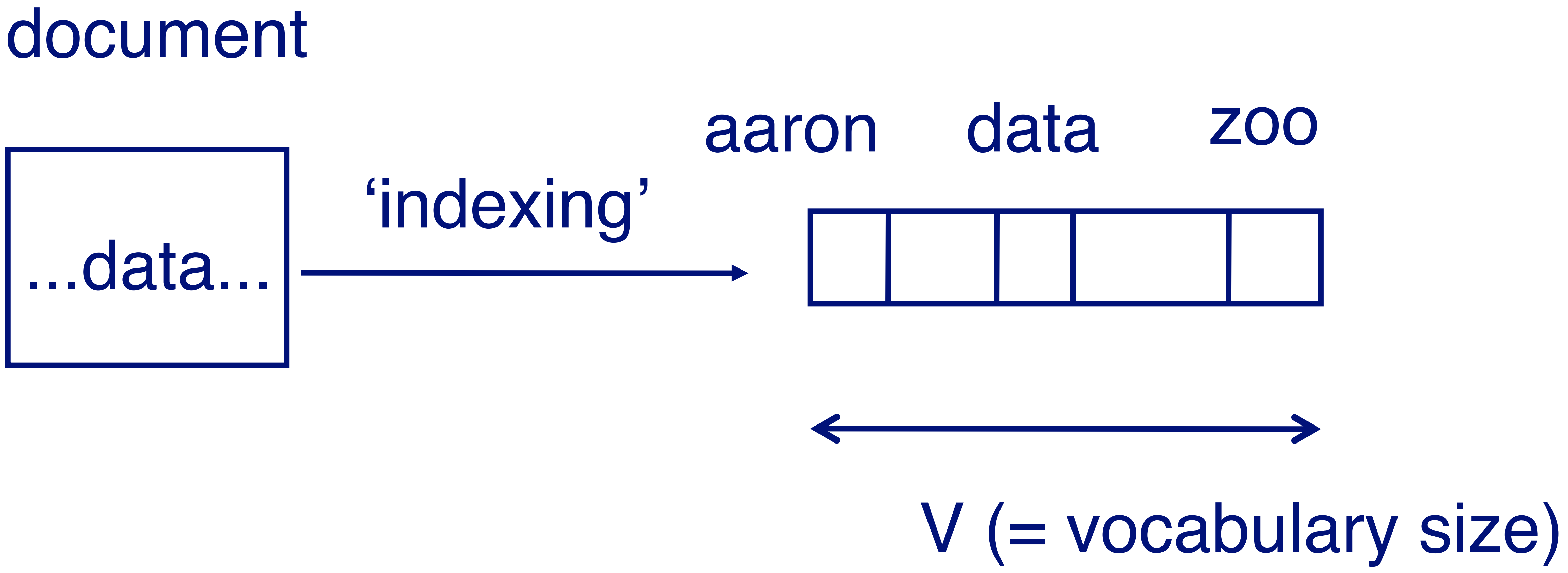
Each document -> vector

Each query -> vector

Search for documents -> find “similar” vectors

Vector Space Model and Clustering

- Main idea:

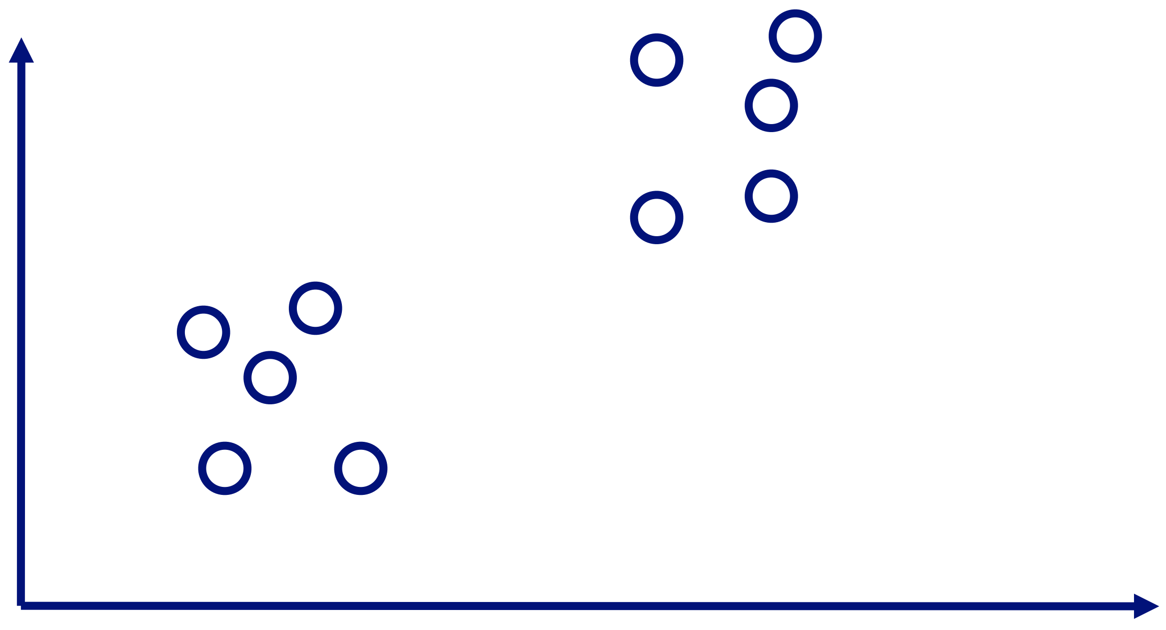


Outline - detailed

- main idea
- cluster search
- cluster generation
- evaluation

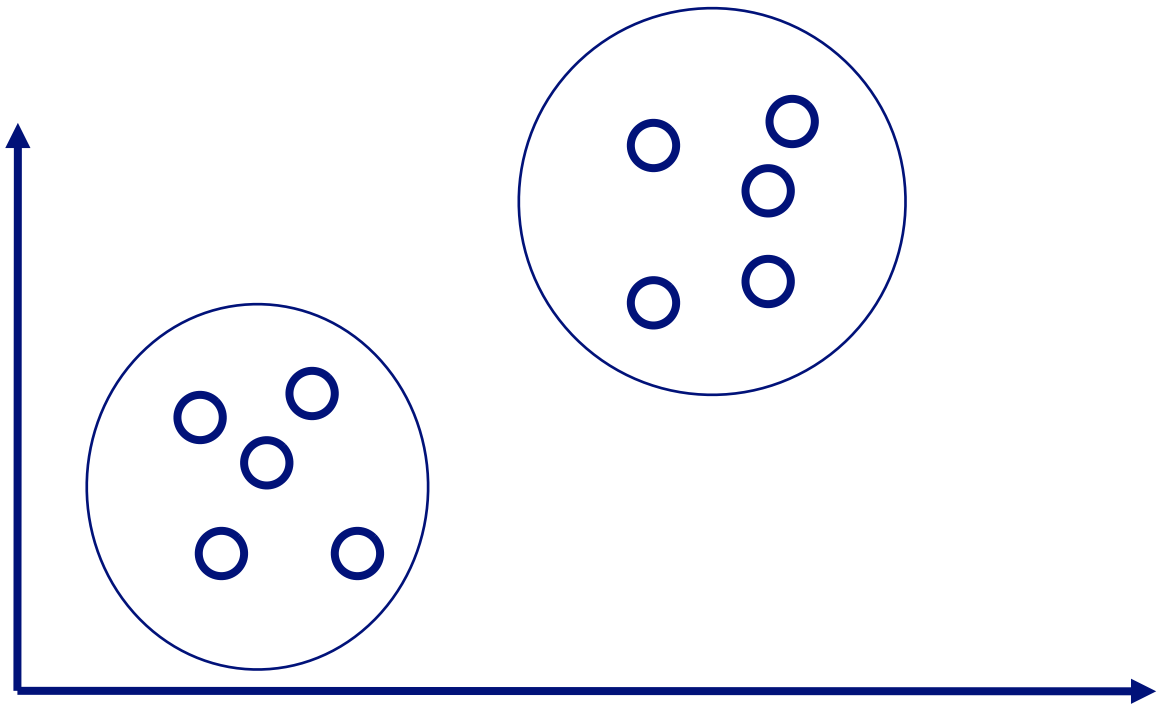
Cluster generation

- Problem:
  - given N points in V dimensions,
  - group them



Cluster generation

- Problem:
  - given N points in V dimensions,
  - group them



Cluster generation

We need

- Q1: document-to-document similarity
- Q2: document-to-cluster similarity

Cluster generation

Q1: document-to-document similarity

(recall: ‘bag of words’ representation)

- D1: {‘data’, ‘retrieval’, ‘system’}
- D2: {‘lung’, ‘pulmonary’, ‘system’}
- distance/similarity functions?

Cluster generation

A1: # of words in common

A2: .... normalized by the vocabulary sizes

A3: .... etc

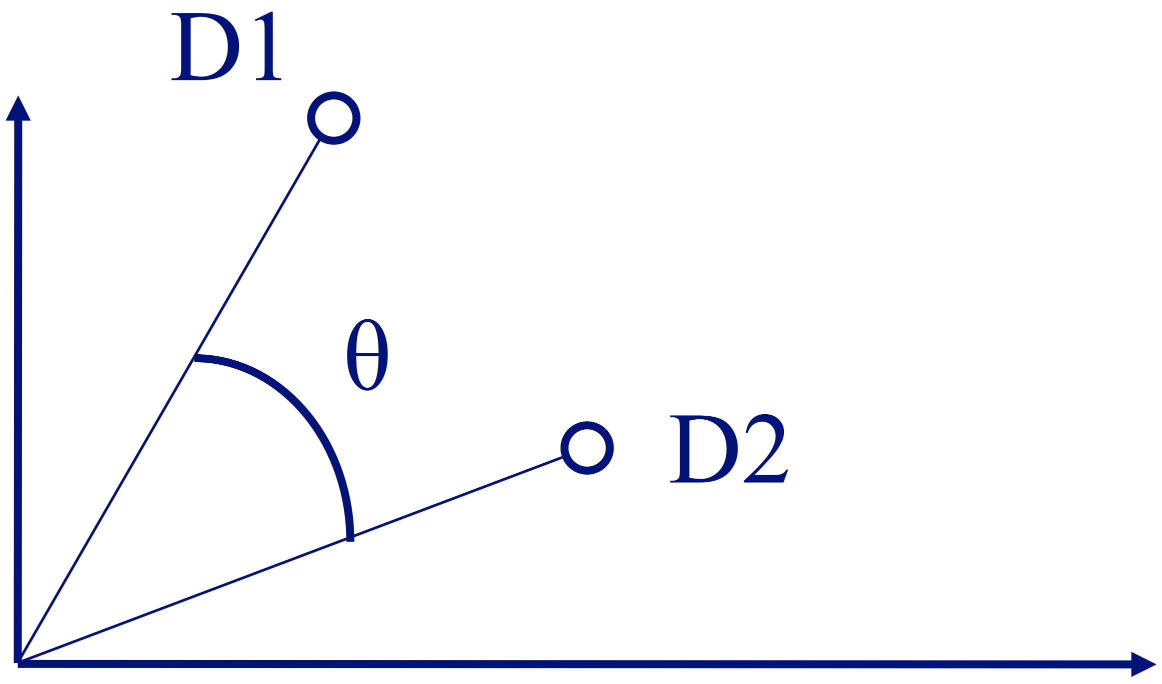
About the same performance - prevailing one:
cosine similarity

Cluster generation

cosine similarity:

$$\text{similarity}(D1, D2) = \cos(\theta) =$$

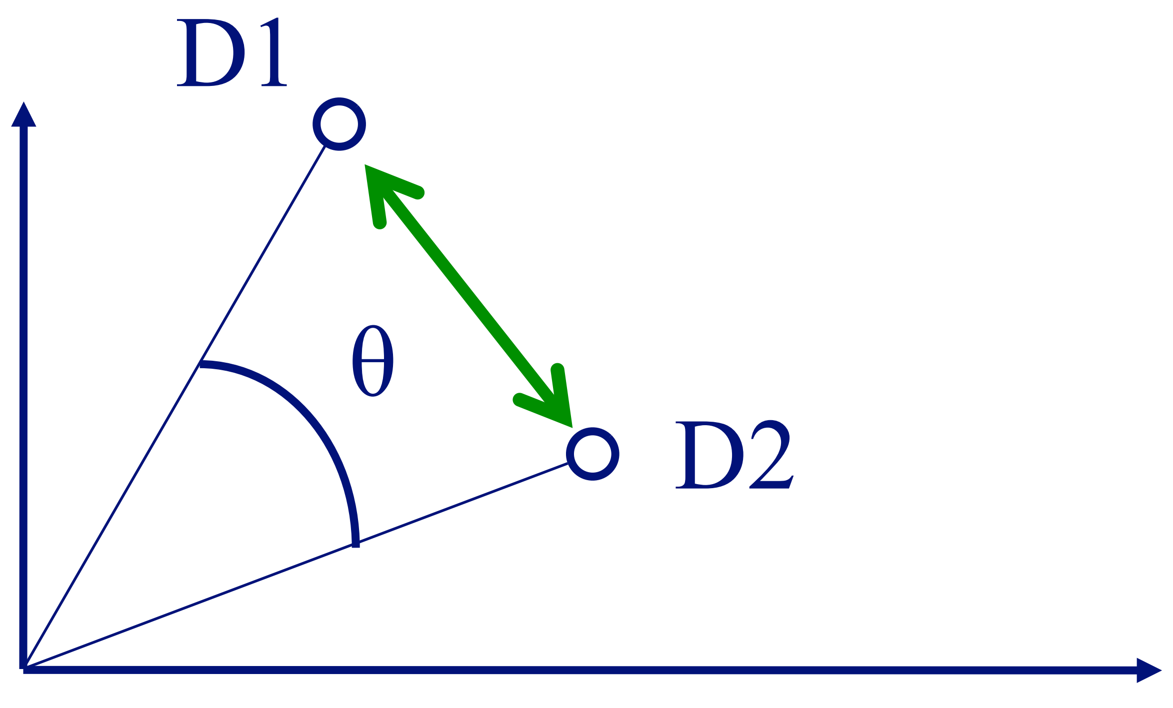
$$\text{sum}(v_{1,i} * v_{2,i}) / [\text{len}(v_1) * \text{len}(v_2)]$$



Cluster generation

cosine similarity - observations:

- related to the Euclidean distance
- weights $v_{i,j}$: according to tf/idf



Cluster generation

tf (‘term frequency’)

- high, if the term appears very often in this document.

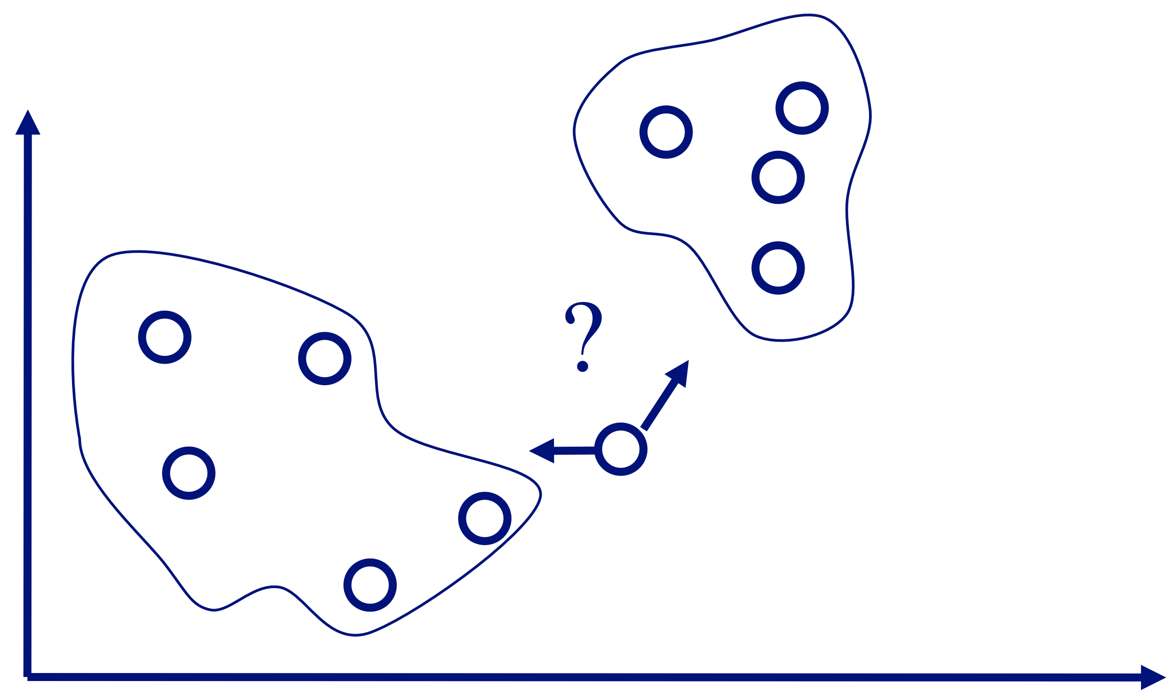
idf (‘inverse document frequency’)

- penalizes ‘common’ words, that appear in almost every document

Cluster generation

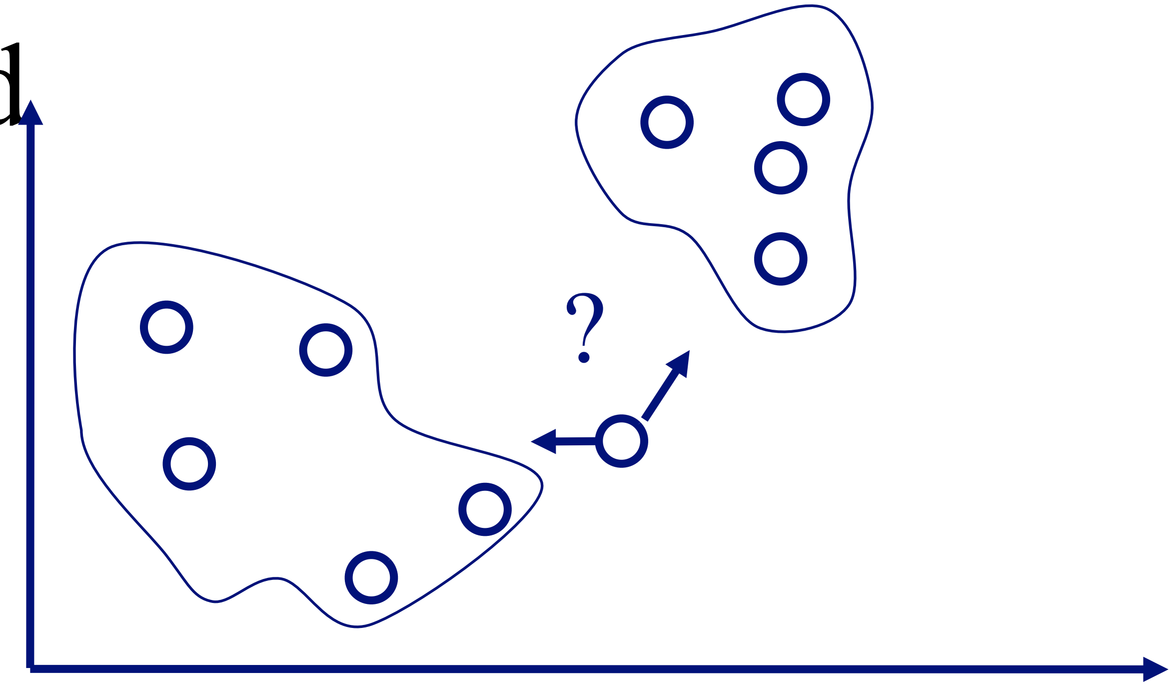
We need

- Q1: document-to-document similarity
- Q2: document-to-cluster similarity



Cluster generation

- A1: min distance (‘single-link’)
- A2: max distance (‘all-link’)
- A3: avg distance (gives same cluster ranking as A4, but different values)
- A4: distance to centroid



Cluster generation

- A1: min distance (‘single-link’)
  - leads to elongated clusters
- A2: max distance (‘all-link’)
  - many, small, tight clusters
- A3: avg distance
  - in between the above
- A4: distance to centroid
  - fast to compute

Cluster generation

We have

- document-to-document similarity
- document-to-cluster similarity

Q: How to group documents into ‘natural’ clusters

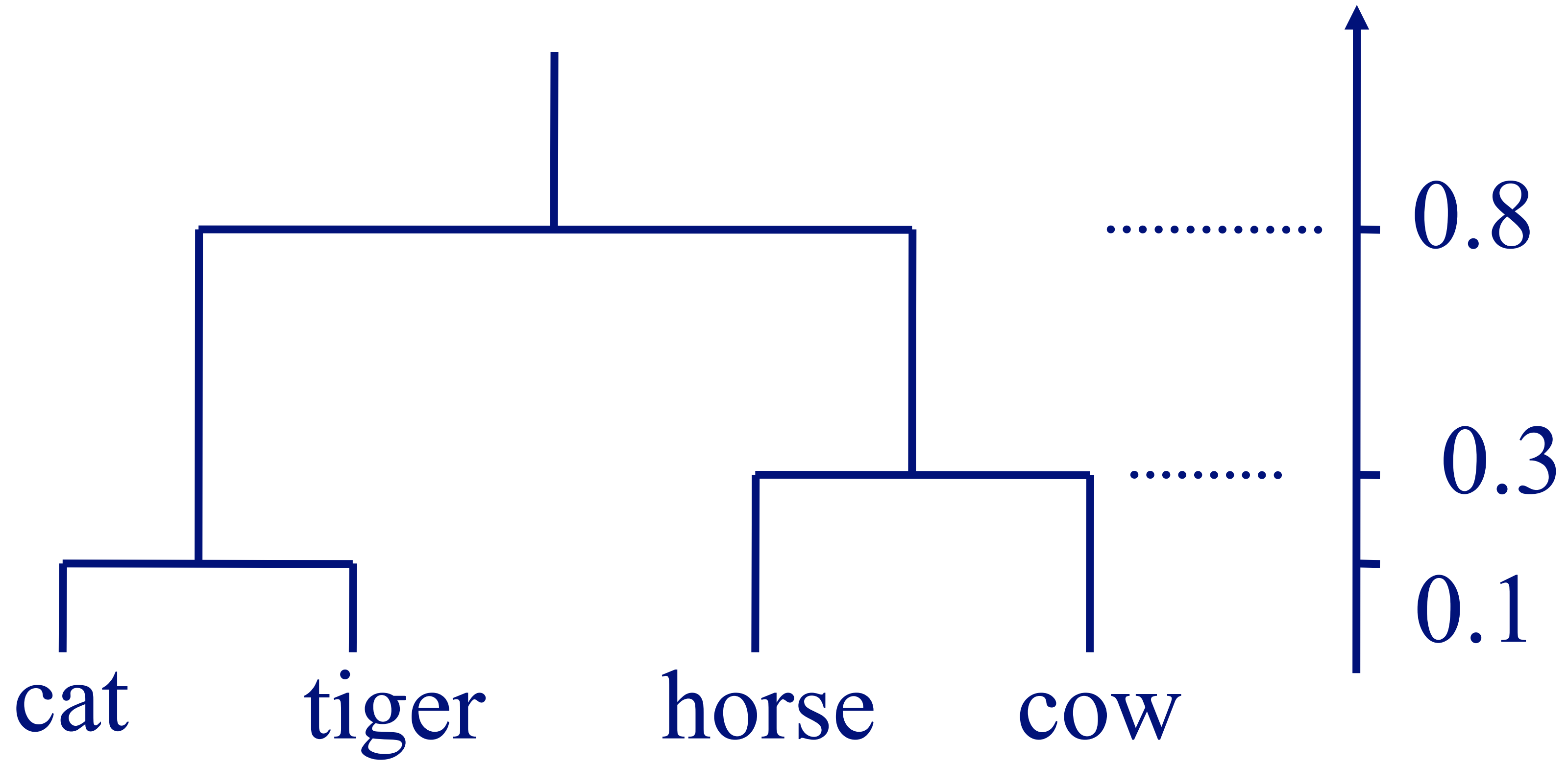
Cluster generation

A: *many-many* algorithms - in two groups [VanRijsbergen]:

- theoretically sound (O(N^2))
  - independent of the insertion order
- iterative (O(N), O(N log(N))

Cluster generation - ‘sound’ methods

- Approach#1: dendrograms - create a hierarchy (bottom up or top-down) - choose a cut-off (how?) and cut



Cluster generation - ‘sound’ methods

- Approach#2: min. some statistical criterion (eg., sum of squares from cluster centers)
  - like ‘k-means’
  - but how to decide ‘k’?

Cluster generation

one way to estimate # of clusters k: the ‘cover coefficient’ [Can+] ~ SVD

LSI - Detailed outline

- LSI
  - problem definition
  - main idea
  - experiments

Information Filtering + LSI

- [Foltz+,'92] Goal:
  - users specify interests (= keywords)
  - system alerts them, on suitable news-documents
- Major contribution:
  LSI = Latent Semantic Indexing
  - latent ('hidden') concepts

Information Filtering + LSI

Main idea

- map each document into some **'concepts'**
- map each term into some **'concepts'**

'Concept':~ a set of terms, with weights,
e.g. DBMS_concept:
"data" (0.8),
"system" (0.5),
"retrieval" (0.6)

Information Filtering + LSI

Pictorially: term-document matrix (BEFORE)

	'data'	'system'	'retrieval'	'lung'	'ear'
TR1	1	1	1		
TR2	1	1	1		
TR3				1	1
TR4				1	1

Information Filtering + LSI

Pictorially: **concept-document** matrix and...

	'DBMS-concept'	'medical-concept'
TR1	1	
TR2	1	
TR3		1
TR4		1

Information Filtering + LSI

... and **concept-term** matrix

	'DBMS-concept'	'medical-concept'
data	1	
system	1	
retrieval	1	
lung		1
ear		1

Information Filtering + LSI

Q: How to search, e.g., for ‘system’?

Information Filtering + LSI

A: find the corresponding concept(s); and the corresponding documents

	'DBMS-concept'	'medical-concept'
data	1	
system	1	
retrieval	1	
lung		1
ear		1

	'DBMS-concept'	'medical-concept'
TR1	1	
TR2	1	
TR3		1
TR4		1

Information Filtering + LSI

A: find the corresponding concept(s); and the corresponding documents

	'DBMS-concept'	'medical-concept'
data	1	
system	1	
retrieval	1	
lung		1
ear		1

	'DBMS-concept'	'medical-concept'
TR1	1	
TR2	1	
TR3		1
TR4		1

Information Filtering + LSI

Thus it 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 ‘statistical thesaurus’ automatically
  - to reduce dimensionality (down to few “concepts”)
- How exactly SVD works? (Details, next)

Singular Value Decomposition (SVD)
Motivation

Problem #1

Text - LSI uses SVD find “concepts”

Problem #2

Compression / dimensionality reduction

SVD - Motivation

Problem #1: text - LSI: find "concepts"

term / document	data	information	retrieval	brain	lung
CS-TR1	1	1	1	0	0
CS-TR2	2	2	2	0	0
CS-TR3	1	1	1	0	0
CS-TR4	5	5	5	0	0
MED-TR1	0	0	0	2	2
MED-TR2	0	0	0	3	3
MED-TR3	0	0	0	1	1

SVD - Motivation

Customer-product, for recommendation system:

$$\begin{array}{cc} & \begin{array}{ccccc} \text{bread} & \text{lettuce} & \text{tomatos} & \text{beef} & \text{chicken} \end{array} \\ \begin{array}{c} \text{vegetarians} \\ \\ \text{meat eaters} \end{array} & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

SVD - Motivation

Problem #2:
Compress / reduce dimensionality

Problem - Specification

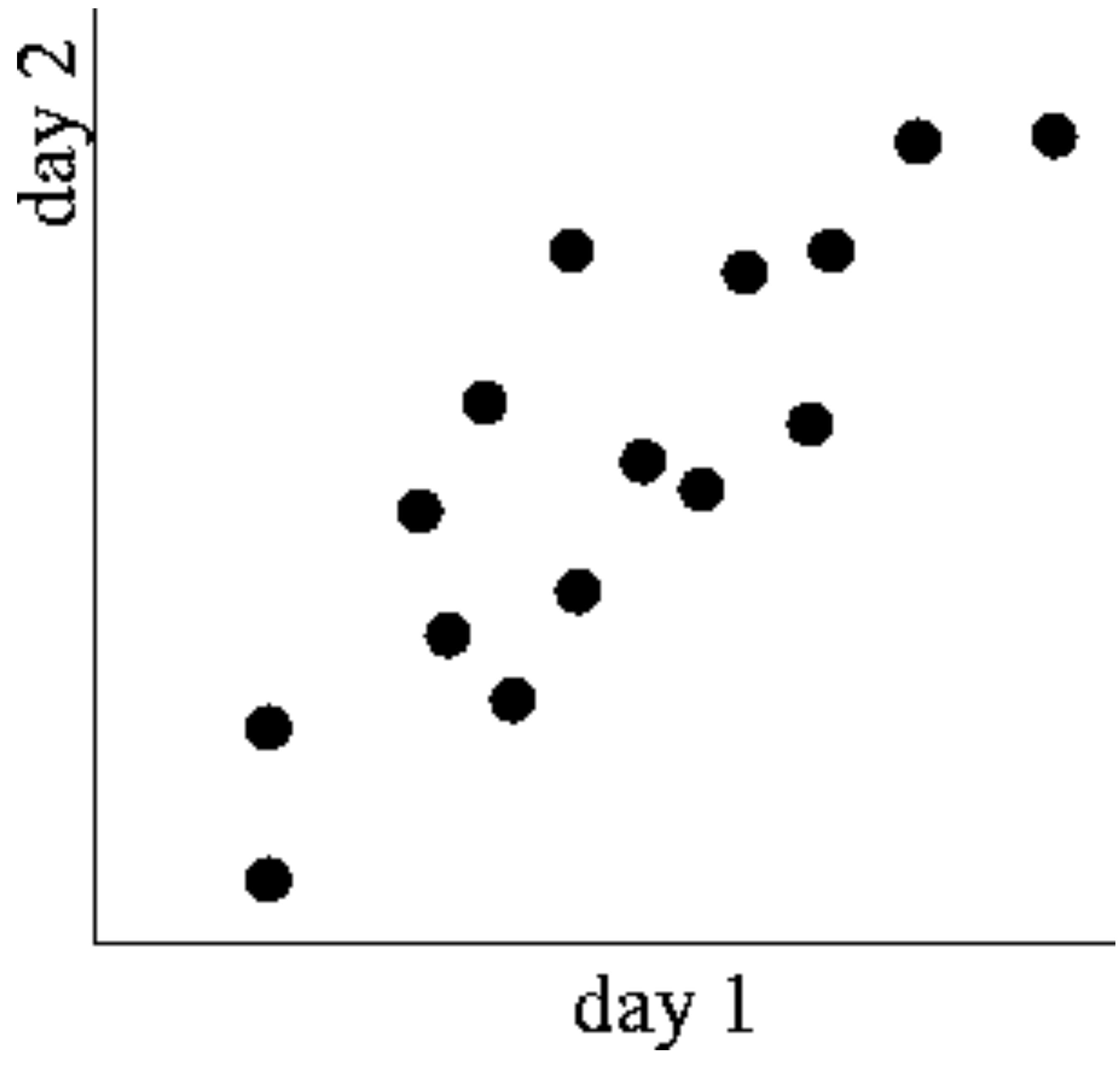
~10^6 rows; ~10^3 columns; no updates

Random access to any cell(s)

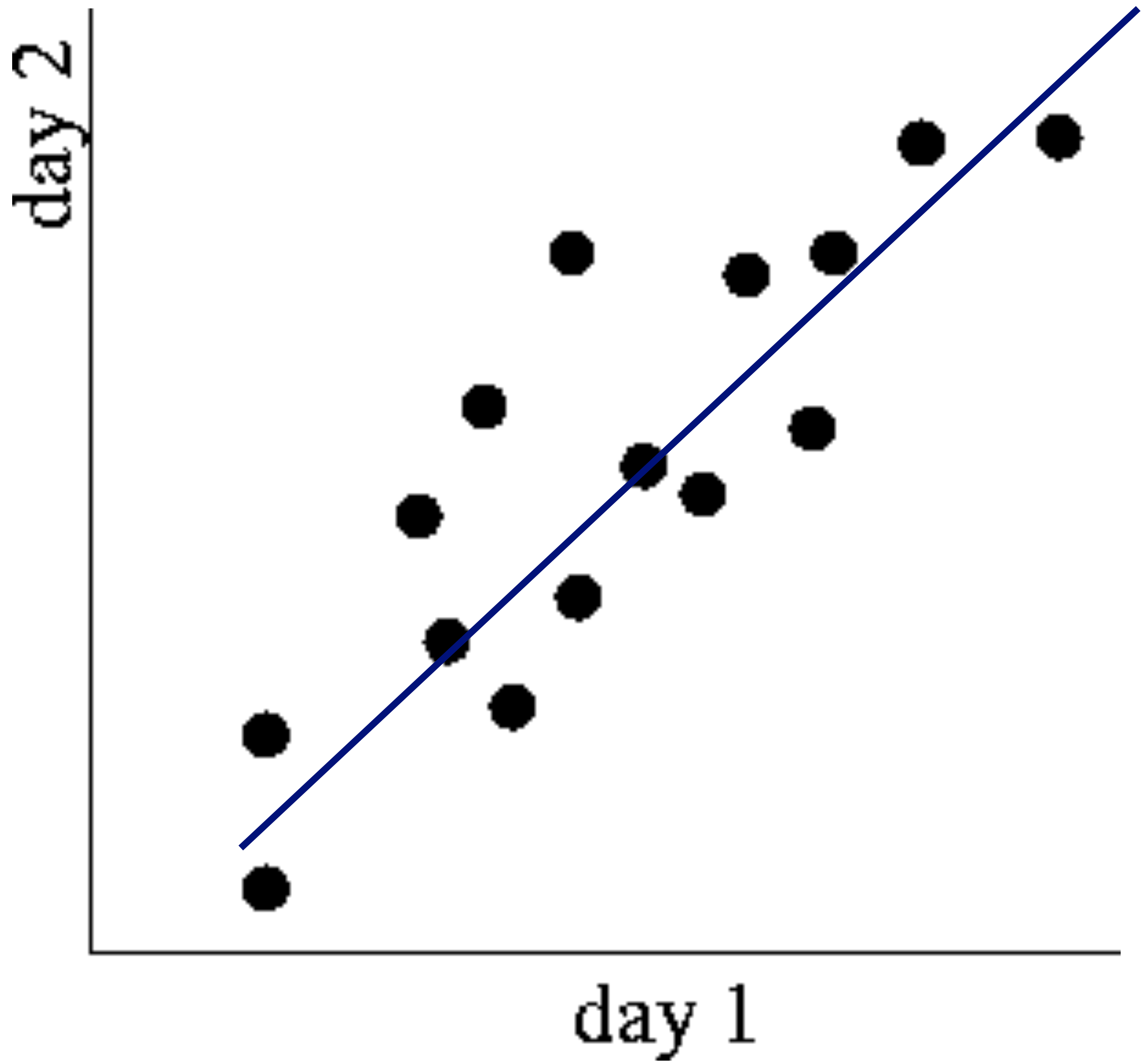
Small error: OK

day / customer	We 7/10/96	Th 7/11/96	Fr 7/12/96	Sa 7/13/96	Su 7/14/96
ABC Inc.	1	1	1	0	0
DEF Ltd.	2	2	2	0	0
GHI Inc.	1	1	1	0	0
KLM Co.	5	5	5	0	0
Smith	0	0	0	2	2
Johnson	0	0	0	3	3
Thompson	0	0	0	1	1

SVD - Motivation



SVD - Motivation



SVD - Definition

(reminder: matrix multiplication)

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \times \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \end{bmatrix}$$

3×2 $\qquad$ 2×1

SVD - Definition

(reminder: matrix multiplication)

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \times \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \end{bmatrix}$$

3×2 $\quad$ 2×1 $\quad$ 3×1

SVD - Definition

(reminder: matrix multiplication)

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \times \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} -1 \\ \\ \\ \end{bmatrix}$$

$$3 \times 2 \qquad 2 \times 1 \qquad 3 \times 1$$

SVD - Definition

(reminder: matrix multiplication)

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \text{ x } \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ \end{bmatrix}$$

3 x 2 $\quad$ 2 x 1 $\quad$ 3 x 1

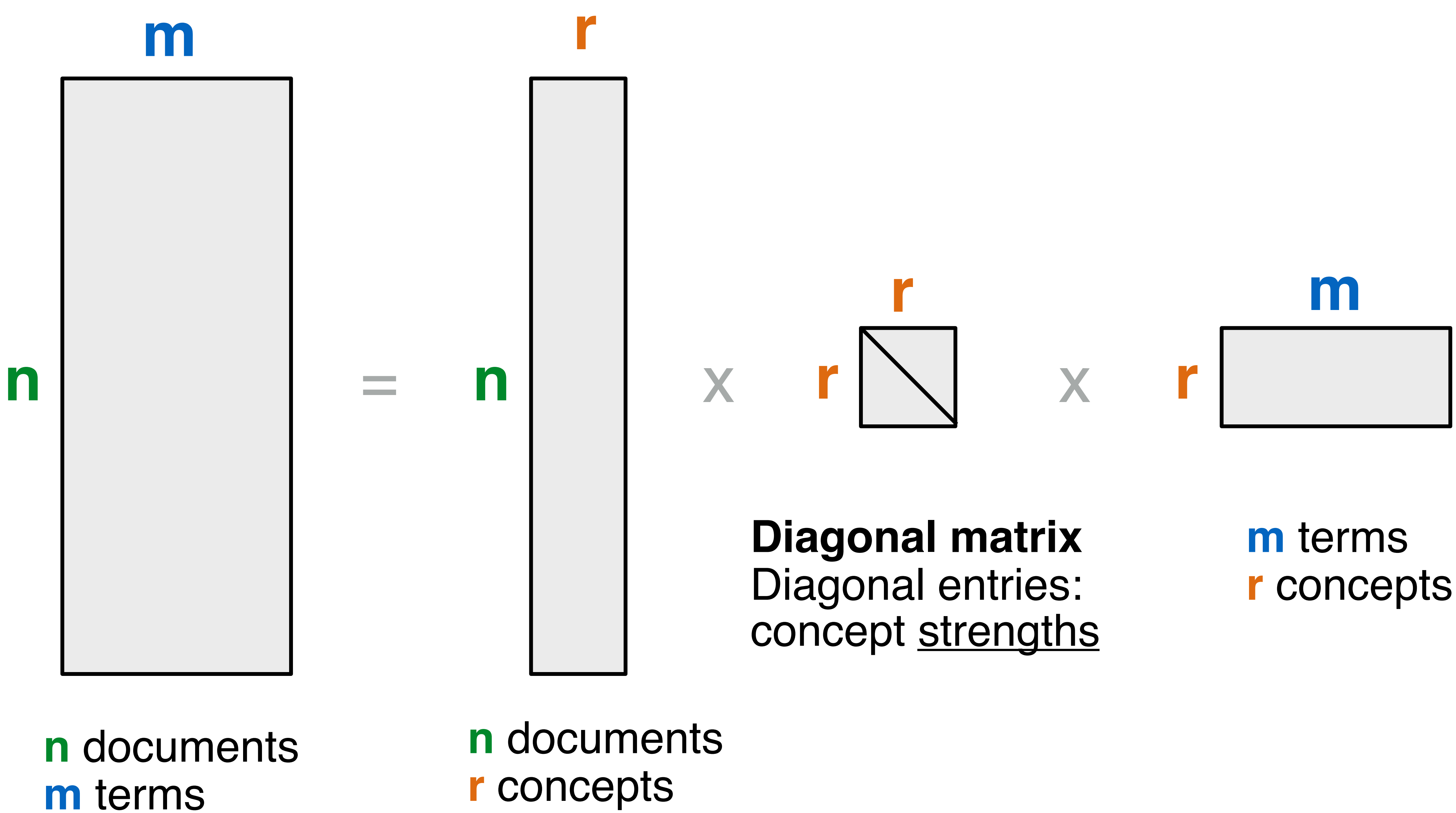
SVD - Definition

(reminder: matrix multiplication)

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \times \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}$$

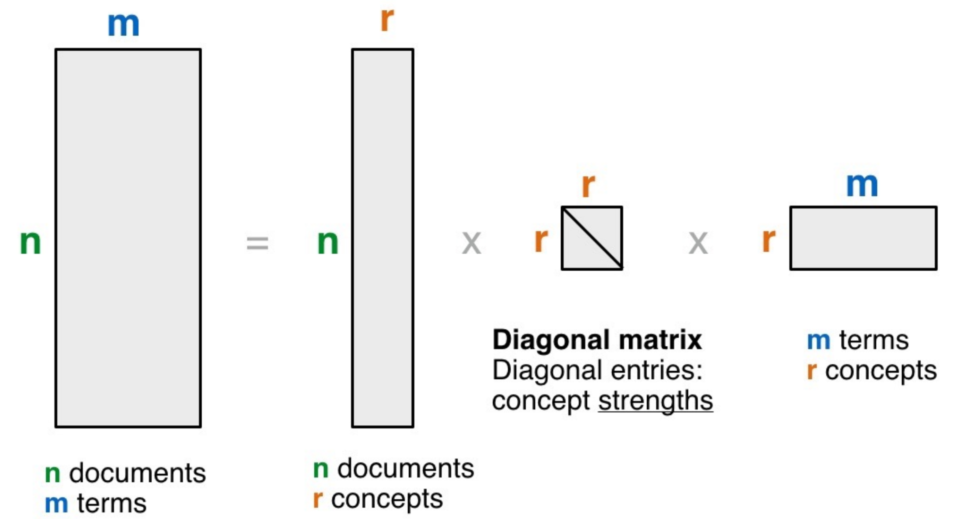
SVD Definition (in picture)

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



SVD Definition (in words)

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



A: n x m matrix

e.g., n documents, m terms

U: n x r matrix

e.g., n documents, r concepts

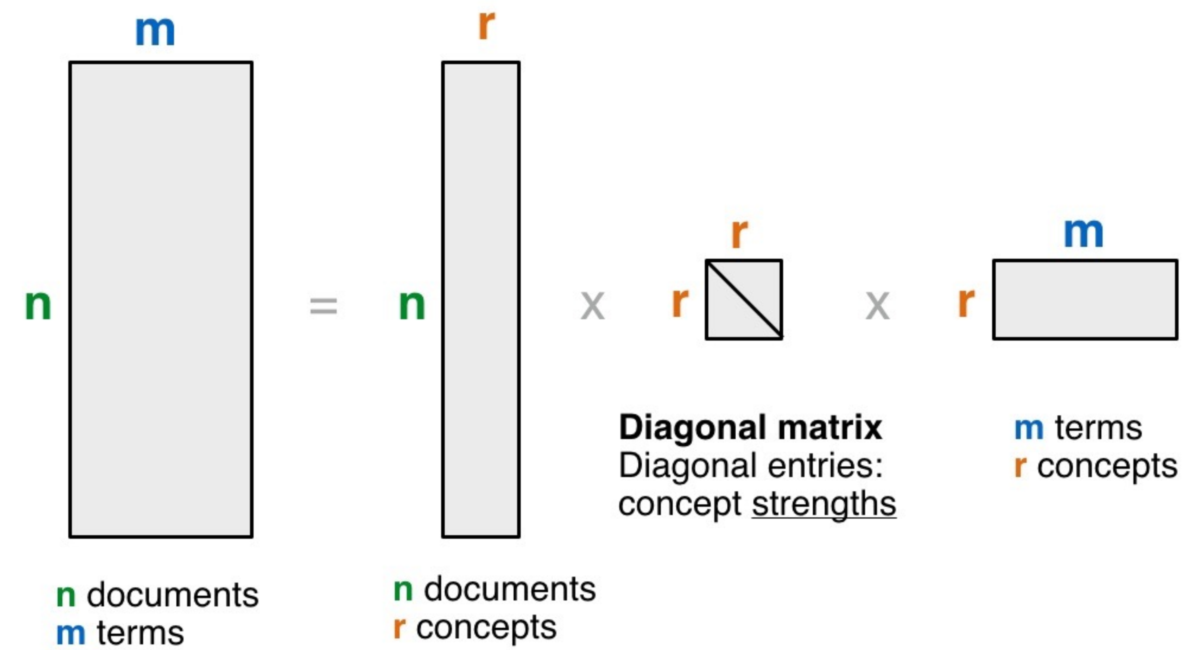
Λ: r x r diagonal matrix

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 $\mathbf{A}$ into

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

$\mathbf{U}, \Lambda, \mathbf{V}$: **unique**, most of the time

$\mathbf{U}, \mathbf{V}$: 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

$\mathbf{A} = \mathbf{U}\,\mathbf{\Lambda}\,\mathbf{V}^{\mathrm{T}}$ - example:

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

SVD - Example

- $\mathbf{A} = \mathbf{U}\, \Lambda\, \mathbf{V}^T$ - example:

CS-concept

MD-concept

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

SVD - Example

- $\mathbf{A} = \mathbf{U}\, \Lambda\, \mathbf{V}^T$ - example:

document-to-concept similarity matrix

CS-concept

MD-concept

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

SVD - Example

- $\mathbf{A} = \mathbf{U}\, \boldsymbol{\Lambda}\, \mathbf{V}^T$ - example:

‘strength’ of CS-concept

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

SVD - Example

- $\mathbf{A} = \mathbf{U}\,\Lambda\,\mathbf{V}^T$ - example:

term-to-concept similarity matrix

CS-concept

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

SVD - Example

- $\mathbf{A} = \mathbf{U}\,\mathbf{\Lambda}\,\mathbf{V}^T$ - example:

term-to-concept similarity matrix

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

CS-concept

$$\begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

SVD - Interpretation #1

‘documents’, ‘terms’ and ‘concepts’:

$\mathbf{U}$: document-to-concept similarity matrix

$\mathbf{V}$: term-to-concept similarity matrix

Λ: diagonal elements: concept “strengths”

SVD - Interpretation #1

‘documents’, ‘terms’ and ‘concepts’:

Q: if $\mathbf{A}$ is the document-to-term matrix, what is the similarity matrix $\mathbf{A}^T \mathbf{A}$?

A:

Q: $\mathbf{A}\,\mathbf{A}^T$?

A:

SVD - Interpretation #1

‘documents’, ‘terms’ and ‘concepts’:

Q: if $\mathbf{A}$ is the document-to-term matrix, what is the similarity matrix $\mathbf{A}^{\mathsf{T}}\mathbf{A}$?

A: term-to-term ([m x m]) similarity matrix

Q: $\mathbf{A}\mathbf{A}^{\mathsf{T}}$?

A: document-to-document ([n x n]) similarity matrix

SVD properties

- $\mathbf{V}$ are the eigenvectors of the *covariance matrix* $\mathbf{A}^{\mathsf{T}}\mathbf{A}$

$$\mathbf{X}^{\mathsf{T}}\mathbf{X} = \left(\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathsf{T}}\right)^{\mathsf{T}}\left(\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathsf{T}}\right) = \mathbf{V}\boldsymbol{\Sigma}^2\mathbf{V}^{\mathsf{T}}$$

- $\mathbf{U}$ are the eigenvectors of the *Gram (inner-product) matrix* $\mathbf{A}\mathbf{A}^{\mathsf{T}}$

$$\mathbf{X}\mathbf{X}^{\mathsf{T}} = \left(\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathsf{T}}\right)\left(\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathsf{T}}\right)^{\mathsf{T}} = \mathbf{U}\boldsymbol{\Sigma}^2\mathbf{U}^{\mathsf{T}}$$

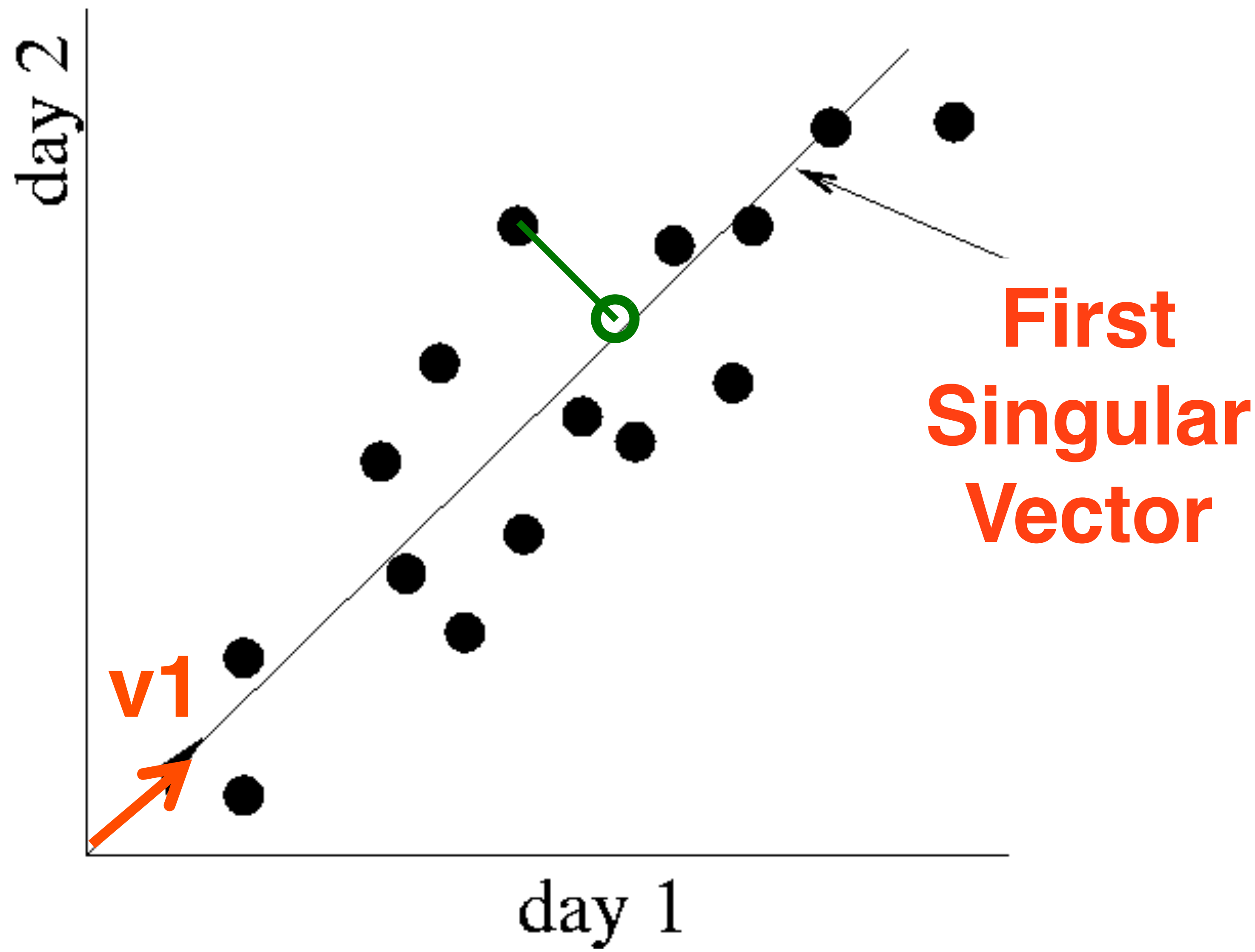
Thus, SVD is closely related to PCA, and can be numerically more stable.
For more info, see:

http://math.stackexchange.com/questions/3869/what-is-the-intuitive-relationship-between-svd-and-pca
Ian T. Jolliffe, *Principal Component Analysis* (2nd ed), Springer, 2002.
Gilbert Strang, *Linear Algebra and Its Applications* (4th ed), Brooks Cole, 2005.

SVD - Interpretation #2

Best axis to project on

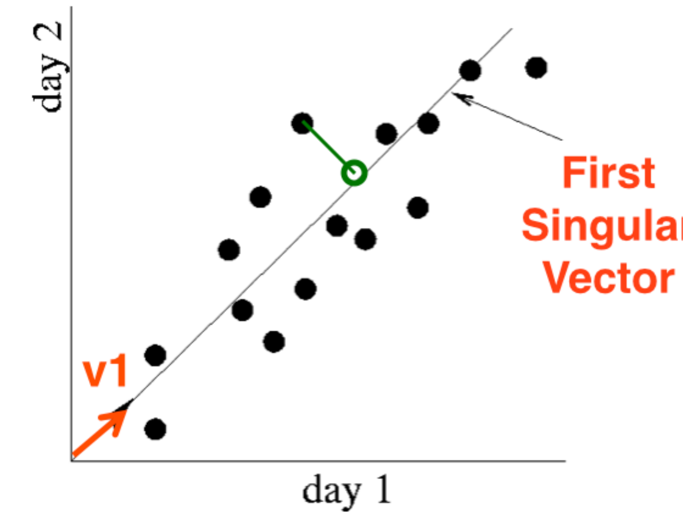
('best' = min sum of squares of projection errors)



min RMS error

Beautiful visualization explaining PCA:
http://setosa.io/ev/principal-component-analysis/

SVD - Interpretation #2



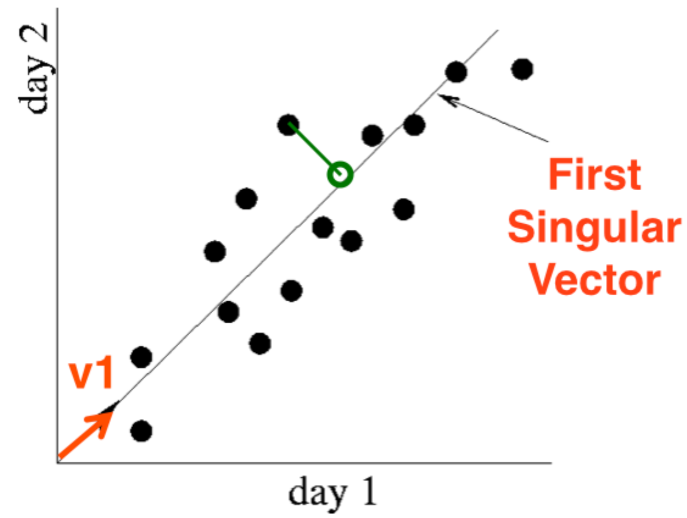
- $\mathbf{A} = \mathbf{U}\ \mathbf{\Lambda}\ \mathbf{V}^T$ - example:

variance ('spread') on the v1 axis

v1

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \times \begin{bmatrix} \boxed{0.58 \quad 0.58 \quad 0.58 \quad 0 \quad 0} \\ 0 \quad 0 \quad 0 \quad 0.71 \quad 0.71 \end{bmatrix}$$

SVD - Interpretation #2



- $\mathbf{A} = \mathbf{U}\ \Lambda\ \mathbf{V}^T$ - example:
  - $\mathbf{U}\ \Lambda$ gives the **coordinates** of the points in the projection axis

$$
\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \mathbf{x} \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \mathbf{x} \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}
$$

SVD - Interpretation #2

- More details
- Q: how exactly is dim. reduction done?

$$
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
2 & 2 & 2 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 0 & 0 & 2 & 2 \\
0 & 0 & 0 & 3 & 3 \\
0 & 0 & 0 & 1 & 1
\end{bmatrix}
=
\begin{bmatrix}
0.18 & 0 \\
0.36 & 0 \\
0.18 & 0 \\
0.90 & 0 \\
0 & 0.53 \\
0 & 0.80 \\
0 & 0.27
\end{bmatrix}
\times
\begin{bmatrix}
9.64 & 0 \\
0 & 5.29
\end{bmatrix}
\times
\begin{bmatrix}
0.58 & 0.58 & 0.58 & 0 & 0 \\
0 & 0 & 0 & 0.71 & 0.71
\end{bmatrix}
$$

SVD - Interpretation #2

- More details
- Q: how exactly is dim. reduction done?
- A: set the smallest singular values to zero:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & \xcancel{5.29} \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

SVD - Interpretation #2

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \sim \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

SVD - Interpretation #2

$$
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
2 & 2 & 2 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 0 & 0 & 2 & 2 \\
0 & 0 & 0 & 3 & 3 \\
0 & 0 & 0 & 1 & 1
\end{bmatrix}
\sim
\begin{bmatrix}
0.18 & \cancel{0} \\
0.36 & \cancel{0} \\
0.18 & \cancel{0} \\
0.90 & \cancel{0} \\
0 & \cancel{0.53} \\
0 & \cancel{0.80} \\
0 & \cancel{0.27}
\end{bmatrix}
\mathbf{x}
\begin{bmatrix}
9.64 & 0 \\
0 & \cancel{0}
\end{bmatrix}
\mathbf{x}
\begin{bmatrix}
0.58 & 0.58 & 0.58 & 0 & 0 \\
\cancel{0} & \cancel{0} & \cancel{0} & \cancel{0.71} & \cancel{0.71}
\end{bmatrix}
$$

SVD - Interpretation #2

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \sim \begin{bmatrix} 0.18 \\ 0.36 \\ 0.18 \\ 0.90 \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{x} \begin{bmatrix} 9.64 \end{bmatrix} \text{x} \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \end{bmatrix}$$

SVD - Interpretation #2

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \sim \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

SVD - Interpretation #2

Exactly equivalent:

"spectral decomposition" of the matrix:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \mathbf{x} \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \mathbf{x} \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

SVD - Interpretation #2

Exactly equivalent:

'spectral decomposition' of the matrix:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} | & | \\ u_1 & u_2 \\ | & | \end{bmatrix} \text{x} \begin{bmatrix} \lambda_1 & \emptyset \\ \emptyset & \lambda_2 \end{bmatrix} \text{x} \begin{bmatrix} \text{—} \; v_1 \; \text{—} \\ \text{—} \; v_2 \; \text{—} \end{bmatrix}$$

SVD - Interpretation #2

Exactly equivalent:

'spectral decomposition' of the matrix:

$$\begin{array}{cc} & \xleftarrow{\quad m \quad} \\ n\updownarrow & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array} = \lambda_1 \; u_1 \; v^T{}_1 + \lambda_2 \; u_2 \; v^T{}_2 + \ldots$$

SVD - Interpretation #2

Exactly equivalent:

'spectral decomposition' of the matrix:

$$
\begin{array}{cc}
 & \overleftrightarrow{\quad m \quad} \\
n \updownarrow &
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
2 & 2 & 2 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 0 & 0 & 2 & 2 \\
0 & 0 & 0 & 3 & 3 \\
0 & 0 & 0 & 1 & 1
\end{bmatrix}
\end{array}
=
\overbrace{\lambda_1 \; \underbrace{u_1}_{n \times 1} \; \underbrace{v^T_1}_{1 \times m} + \lambda_2 \; u_2 \; v^T_2 + \ldots}^{r \text{ terms}}
$$

SVD - Interpretation #2

approximation / dim. reduction:

by keeping the first few terms (Q: how many?)

$$
n \left\{ \overbrace{\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}}^{m} \right. = \lambda_1 \quad u_1 \quad v^T_1 + \lambda_2 \quad u_2 \quad v^T_2 + \ldots
$$

assume: $\lambda_1 >= \lambda_2 >= \ldots$

SVD - Interpretation #2

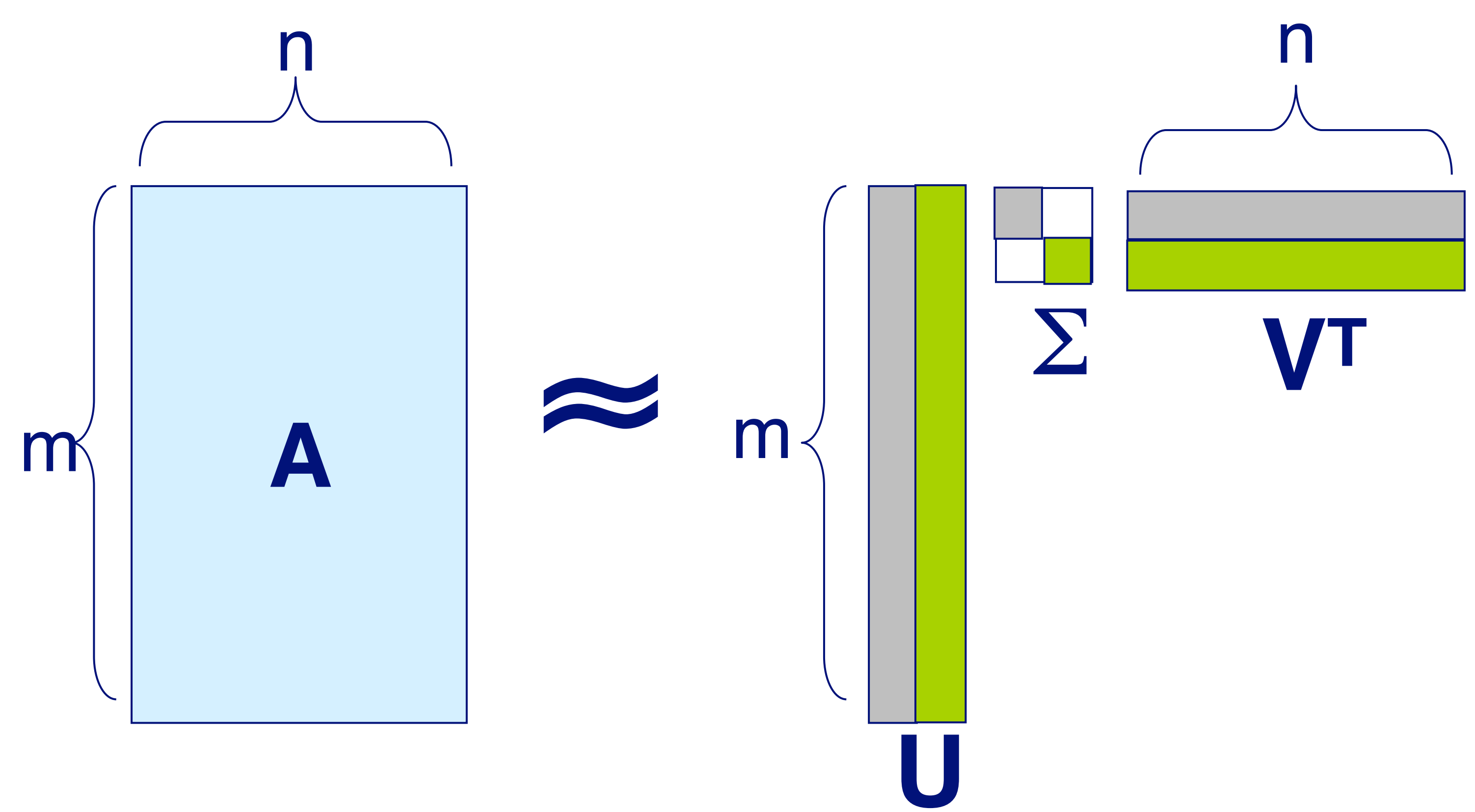
A (heuristic - [Fukunaga]): keep 80-90% of 'energy' (= sum of squares of λ_i 's)

$$
\begin{array}{cc}
 & \longleftarrow m \longrightarrow \\
n & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}
\end{array}
= \lambda_1 \quad u_1 \quad v^T_1 + \quad \lambda_2 \quad u_2 \quad v^T_2 + \ldots
$$

assume: $\lambda_1 >= \lambda_2 >= \ldots$

Pictorially: matrix form of SVD

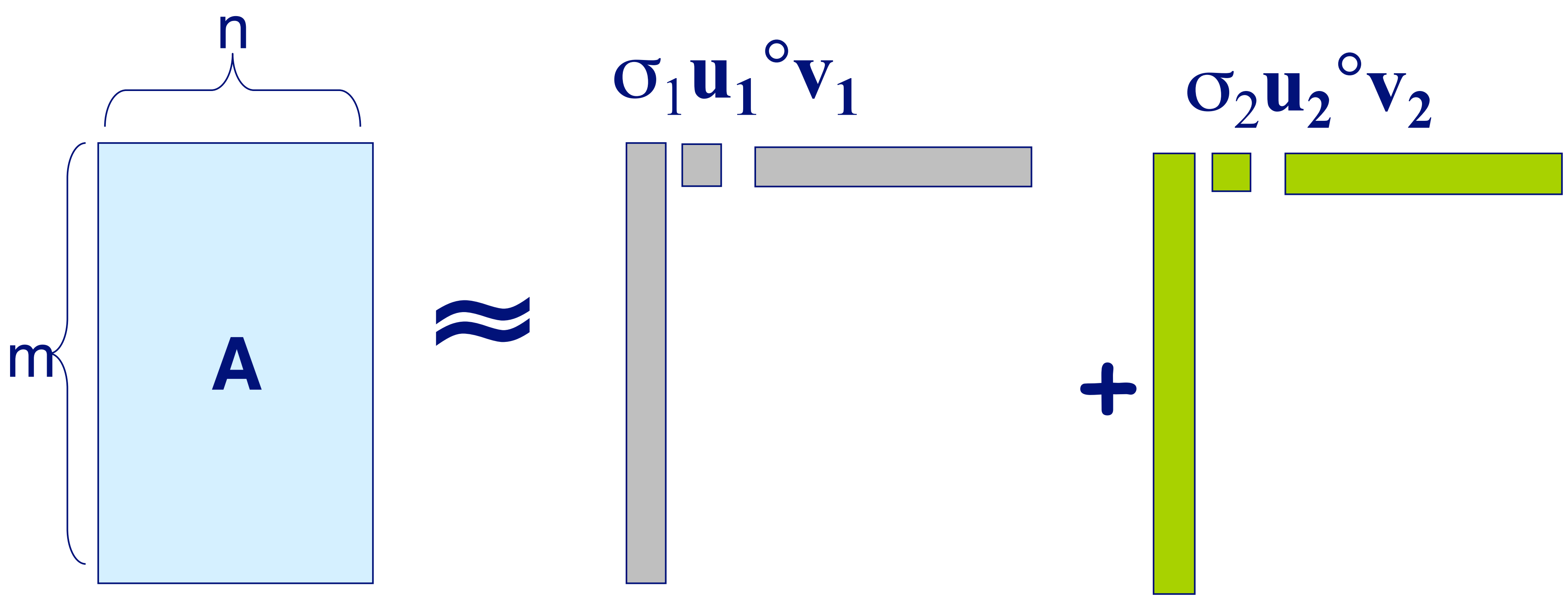
$$\mathbf{A} \approx \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_i \sigma_i \mathbf{u}_i \circ \mathbf{v}_i$$



– Best rank-k approximation in L2

Pictorially: Spectral form of SVD

$$\mathbf{A} \approx \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T = \sum_i \sigma_i \mathbf{u}_i \circ \mathbf{v}_i$$



– Best rank-k approximation in L2

SVD - Interpretation #3

- finds non-zero ‘blobs’ in a data matrix

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

SVD - Interpretation #3

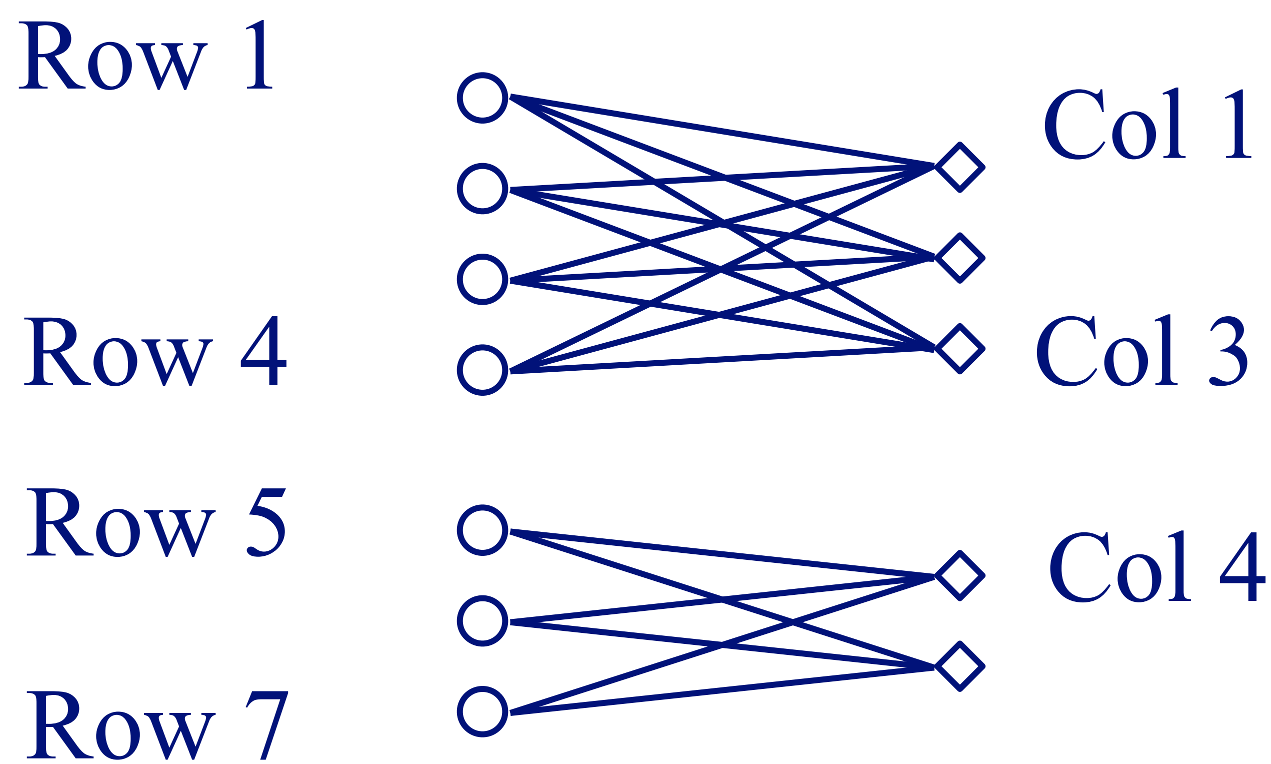
- finds non-zero 'blobs' in a data matrix

$$\left[\begin{array}{ccc|cc} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ \hline 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{array}\right] = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

SVD - Interpretation #3

- finds non-zero ‘blobs’ in a data matrix =
- ‘communities’ (bi-partite cores, here)

$$\left[\begin{array}{ccc|cc}
1 & 1 & 1 & 0 & 0 \\
2 & 2 & 2 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
\hline
0 & 0 & 0 & 2 & 2 \\
0 & 0 & 0 & 3 & 3 \\
0 & 0 & 0 & 1 & 1
\end{array}\right]$$



SVD algorithm

- Numerical Recipes in C (free)

SVD - Interpretation #3

- Drill: find the SVD, 'by inspection'!
- Q: rank = ??

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} \\ \\ ?? \\ \\ \\ \end{bmatrix} \times \begin{bmatrix} \\ ?? \\ \\ \end{bmatrix} \times \begin{bmatrix} & & ?? & & \end{bmatrix}$$

SVD - Interpretation #3

- A: rank = 2 (2 linearly independent rows/ cols)

$$
\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} | & | \\ ?? & ?? \\ | & | \end{bmatrix} \times \begin{bmatrix} ?? & 0 \\ 0 & ?? \end{bmatrix} \times \begin{bmatrix} \text{---} & ?? & \text{---} \\ \text{---} & ?? & \text{---} \end{bmatrix}
$$

SVD - Interpretation #3

- A: rank = 2 (2 linearly independent rows/ cols)

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} ?? & 0 \\ 0 & ?? \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

orthogonal??

SVD - Interpretation #3

- column vectors: are orthogonal - but not unit vectors:

$$
\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1/\mathrm{sqrt}(3) & 0 \\ 1/\mathrm{sqrt}(3) & 0 \\ 1/\mathrm{sqrt}(3) & 0 \\ 0 & 1/\mathrm{sqrt}(2) \\ 0 & 1/\mathrm{sqrt}(2) \end{bmatrix} \times \begin{bmatrix} ?? & 0 \\ 0 & ?? \end{bmatrix} \times \begin{bmatrix} 1/\mathrm{sqrt}(3) & 1/\mathrm{sqrt}(3) & 1/\mathrm{sqrt}(3) & 0 & 0 \\ 0 & 0 & 0 & 1/\mathrm{sqrt}(2) & 1/\mathrm{sqrt}(2) \end{bmatrix}
$$

SVD - Interpretation #3

- and the singular values are:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1/\mathrm{sqrt}(3) & 0 \\ 1/\mathrm{sqrt}(3) & 0 \\ 1/\mathrm{sqrt}(3) & 0 \\ 0 & 1/\mathrm{sqrt}(2) \\ 0 & 1/\mathrm{sqrt}(2) \end{bmatrix} \mathrm{x} \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix} \mathrm{x}$$

$$\begin{bmatrix} 1/\mathrm{sqrt}(3) & 1/\mathrm{sqrt}(3) & 1/\mathrm{sqrt}(3) & 0 & 0 \\ 0 & 0 & 0 & 1/\mathrm{sqrt}(2) & 1/\mathrm{sqrt}(2) \end{bmatrix}$$

SVD - Interpretation #3

- Q: How to check we are correct?

$$
\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1/\mathrm{sqrt}(3) & 0 \\ 1/\mathrm{sqrt}(3) & 0 \\ 1/\mathrm{sqrt}(3) & 0 \\ 0 & 1/\mathrm{sqrt}(2) \\ 0 & 1/\mathrm{sqrt}(2) \end{bmatrix} \times \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix} \times \begin{bmatrix} 1/\mathrm{sqrt}(3) & 1/\mathrm{sqrt}(3) & 1/\mathrm{sqrt}(3) & 0 & 0 \\ 0 & 0 & 0 & 1/\mathrm{sqrt}(2) & 1/\mathrm{sqrt}(2) \end{bmatrix}
$$

SVD - Interpretation #3

- A: SVD properties:
  - matrix product should give back matrix $\mathbf{A}$
  - matrix $\mathbf{U}$ should be column-orthonormal, i.e., columns should be unit vectors, orthogonal to each other
  - ditto for matrix $\mathbf{V}$
  - matrix Λ should be diagonal, with non-negative values

SVD - Complexity

O(n*m*m) or O(n*n*m) (whichever is less)

Faster version, if just want singular values

or if we want first k singular vectors

or if the matrix is sparse [Berry]

No need to write your own!

Available in most linear algebra packages (LINPACK, matlab, Splus/R, mathematica ...)

References

- Berry, Michael: http://www.cs.utk.edu/~lsi/
- Fukunaga, K. (1990). Introduction to Statistical Pattern Recognition, Academic Press.
- Press, W. H., S. A. Teukolsky, et al. (1992). Numerical Recipes in C, Cambridge University Press.

Case study - LSI

Q1: How to do queries with LSI?

Q2: multi-lingual IR (english query, on spanish text?)

Case study - LSI

Q1: How to do queries with LSI?

Problem: Eg., find documents with ‘data’

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

Case study - LSI

Q1: How to do queries with LSI?

A: map query vectors into 'concept space' – how?

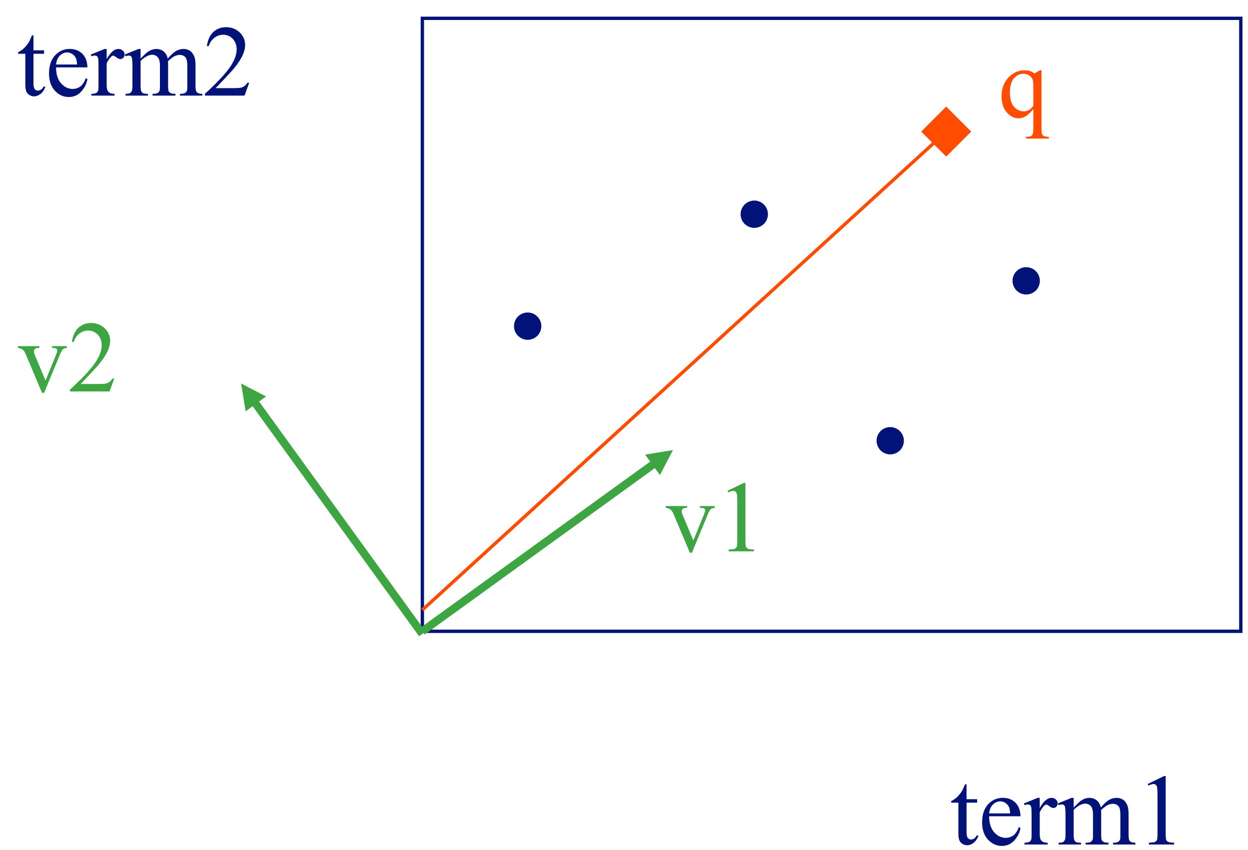
$$
\begin{array}{cc}
 & \begin{array}{ccccc} \text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung} \end{array} \\
\begin{array}{c} \text{CS} \\ \text{CS} \\ \text{CS} \\ \text{CS} \\ \text{MD} \\ \text{MD} \\ \text{MD} \end{array} &
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
2 & 2 & 2 &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

Case study - LSI

Q1: How to do queries with LSI?

A: map query vectors into 'concept space' – how?

$$q = \begin{array}{ccccc} \text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung} \\ \hline 1 & 0 & 0 & 0 & 0 \end{array}$$

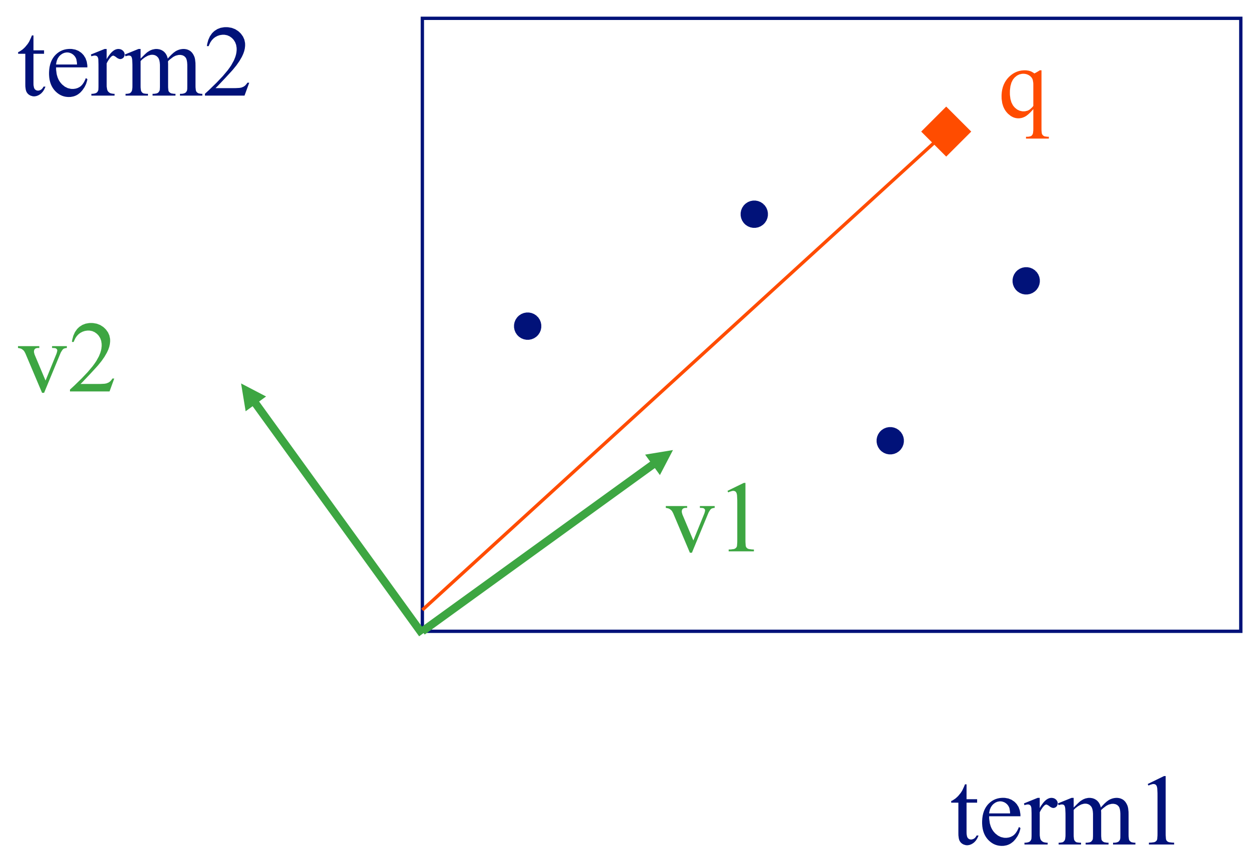


Case study - LSI

Q1: How to do queries with LSI?

A: map query vectors into 'concept space' – how?

$$
q = \begin{array}{ccccc} \text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung} \\ \end{array}
$$
$$
q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix}
$$



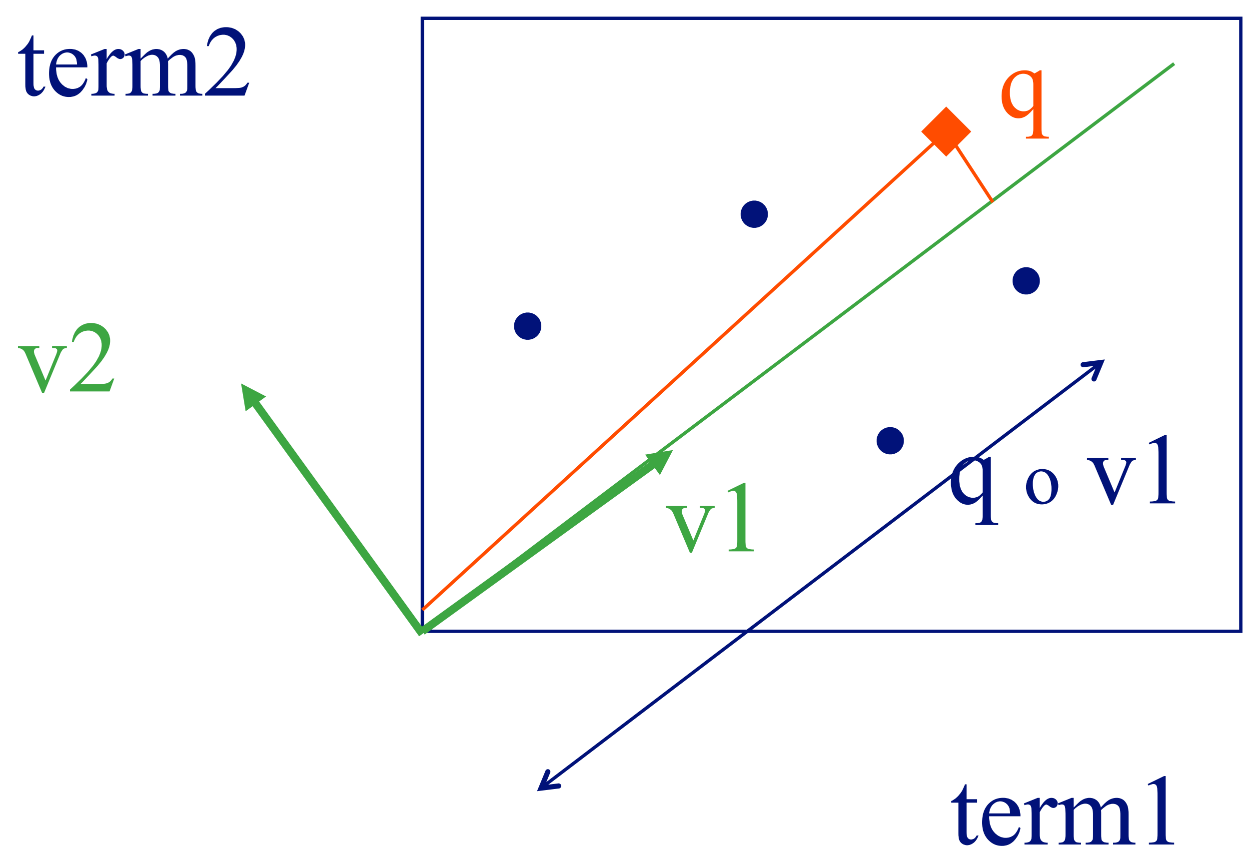
A: inner product
(cosine similarity)
with each 'concept' vector v_i

Case study - LSI

Q1: How to do queries with LSI?

A: map query vectors into 'concept space' – how?

$$
q= \begin{array}{ccccc} \text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung} \\ \end{array}
$$
$$
q= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix}
$$



A: inner product
(cosine similarity)
with each 'concept' vector v_i

Case study - LSI

compactly, we have:

$$q\,\mathbf{V} = q_{concept}$$

Eg:

$$q = \begin{array}{c} \begin{array}{ccccc} \text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung} \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \underbrace{\begin{bmatrix} 0.58 & 0 \\ 0.58 & 0 \\ 0.58 & 0 \\ 0 & 0.71 \\ 0 & 0.71 \end{bmatrix}}_{\text{term-to-concept similarities}} = \begin{array}{c} \text{CS-concept} \\ \begin{bmatrix} 0.58 & 0 \end{bmatrix} \end{array}$$

Case study - LSI

Drill: how would the document (‘information’, ‘retrieval’) be handled by LSI?

Case study - LSI

Drill: how would the document ('information', 'retrieval') be handled by LSI? A: SAME:

$d_{concept} = d \, \mathbf{V}$

Eg:

$$d = \begin{array}{ccccc} \text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung} \\ \end{array}$$

$$d = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.58 & 0 \\ 0.58 & 0 \\ 0.58 & 0 \\ 0 & 0.71 \\ 0 & 0.71 \end{bmatrix} = \begin{bmatrix} 1.16 & 0 \end{bmatrix}$$

CS-concept

term-to-concept similarities

Case study - LSI

Observation: document ('information', 'retrieval') will be retrieved by query ('data'), although it does not contain 'data'!!

$$d = \begin{array}{ccccc} \text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung} \\ \hline 0 & 1 & 1 & 0 & 0 \end{array} \dashrightarrow \begin{array}{cc} \text{CS-concept} & \\ \hline 1.16 & 0 \end{array}$$

$$q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix} \dashrightarrow \begin{bmatrix} 0.58 & 0 \end{bmatrix}$$

Case study - LSI

- Q1: How to do queries with LSI?
- Q2: multi-lingual IR (english query, on spanish text?)

Case study - LSI

- Problem:
  - given many documents, translated to both languages (eg., English and Spanish)
  - answer queries across languages

Case study - LSI

- Solution: ~ LSI

$$
\begin{array}{c}
 \\
\text{CS} \\
 \\
 \\
 \\
\text{MD} \\
 \\
\end{array}
\begin{array}{ccccc}
\text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung} \\
\end{array}
\quad
\begin{array}{cc}
\text{datos} & \text{informacion} \\
\end{array}
$$

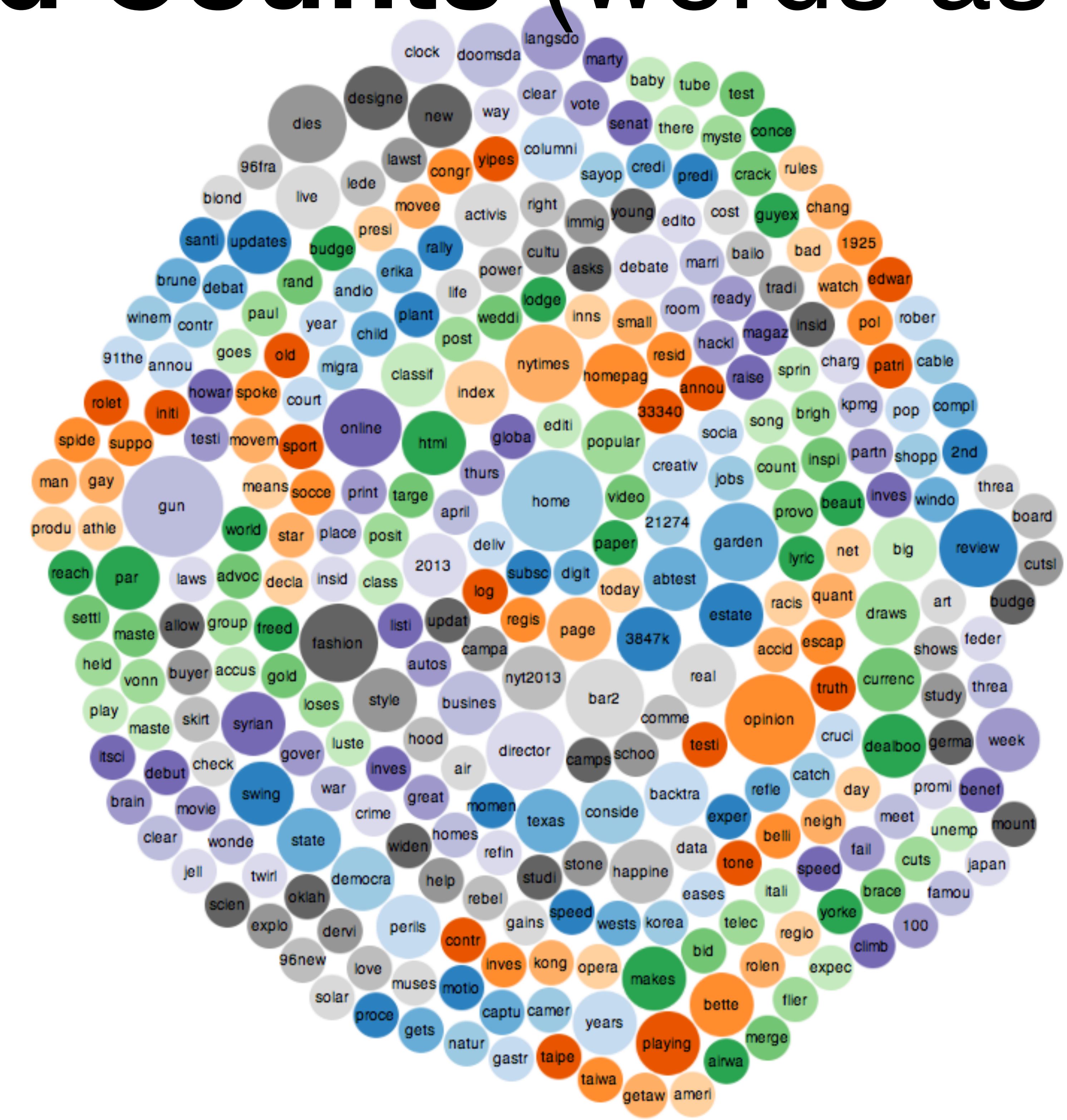
$$
\left[\begin{array}{ccccc:ccccc}
1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
2 & 2 & 2 & 0 & 0 & 1 & 2 & 2 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 & 5 & 5 & 4 & 0 & 0 \\
0 & 0 & 0 & 2 & 2 & 0 & 0 & 0 & 2 & 2 \\
0 & 0 & 0 & 3 & 3 & 0 & 0 & 0 & 2 & 3 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1
\end{array}\right]
$$

Switch Gear to
Text Visualization

Word/Tag Cloud (still popular?)

http://www.wordle.net

Word Counts (words as bubbles)



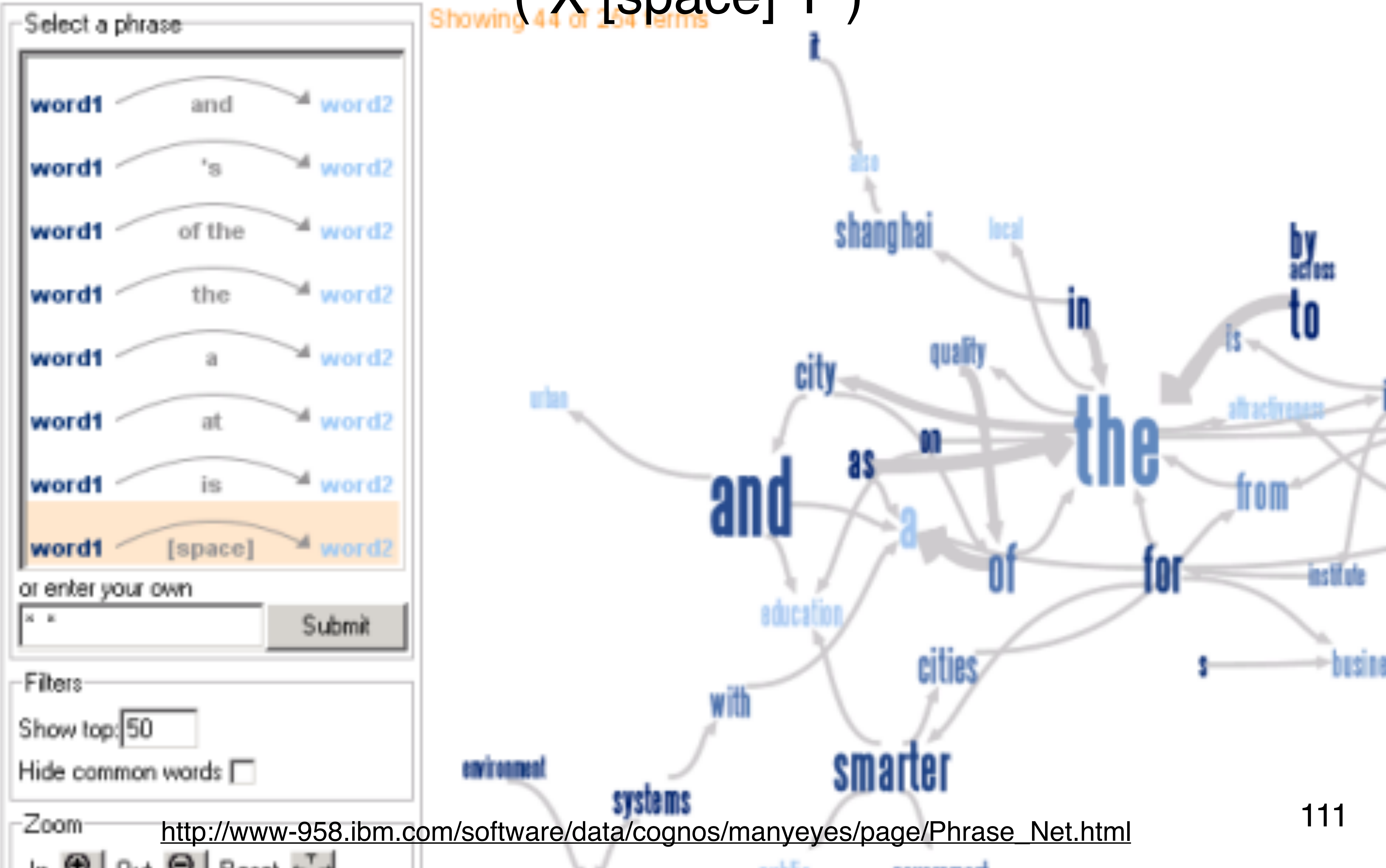
http://www.infocaptor.com/bubble-my-page

Word Tree

http://www.jasondavies.com/wordtree/

Phrase Net

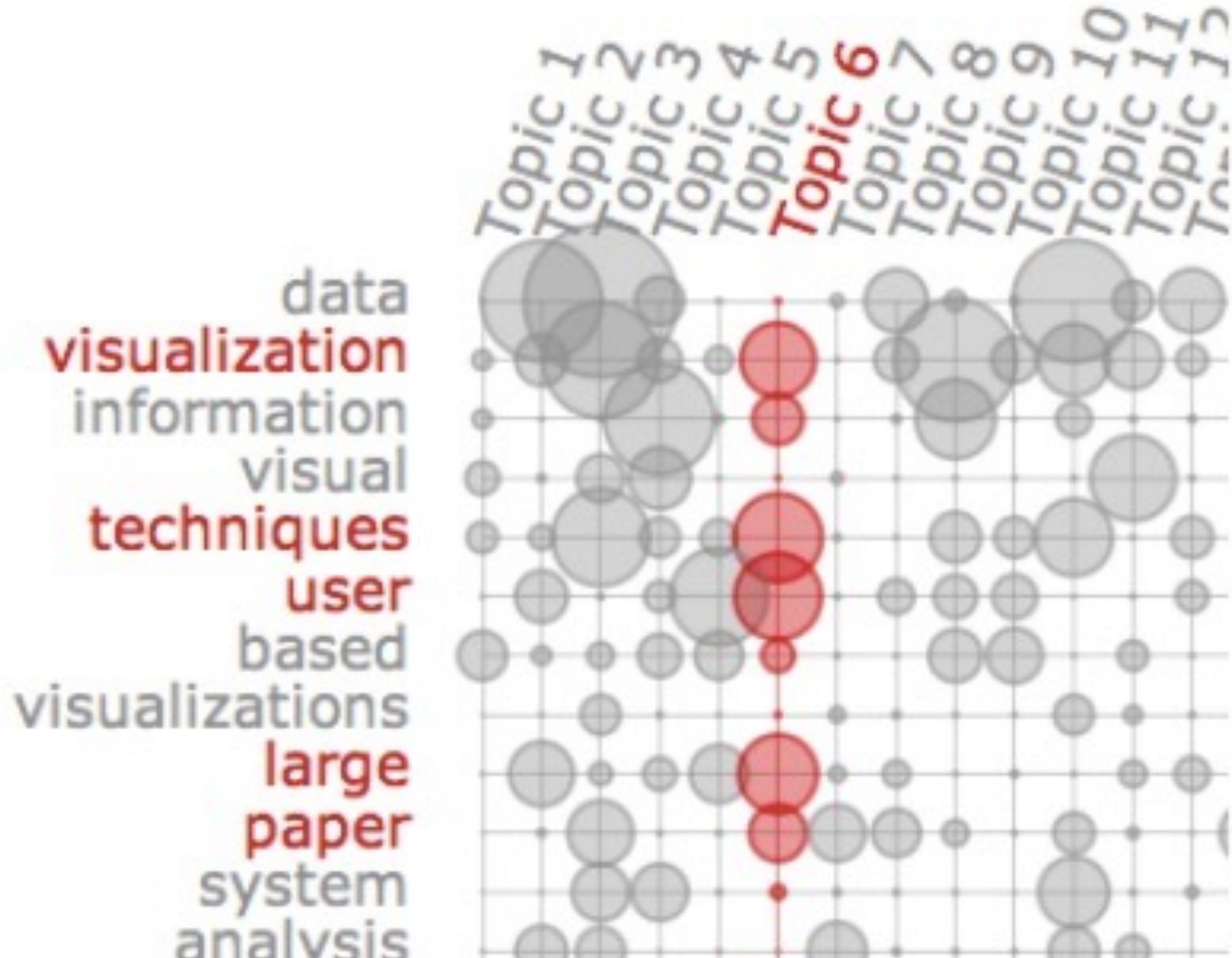
Visualize pairs of words satisfying a pattern
("X [space] Y")



http://www-958.ibm.com/software/data/cognos/manyeyes/page/Phrase_Net.html

Termite: Topic Model Visualization Analy

http://vis.stanford.edu/papers/termite



Termite: Topic Model Visualization

Analy

http://vis.stanford.edu/papers/termite