GENERAL ASSEMBLY

INTRO TO DATA SCIENCE

LECTURE 13: DIMENSIONALITY REDUCTION

I. DIMENSIONALITY REDUCTION
II. PRINCIPAL COMPONENTS ANALYSIS
III. SINGULAR VALUE DECOMPOSITION
IV. OTHER METHODS

EXERCISE:
IV. DIMENSIONALITY REDUCTION IN SCIKIT-LEARN

I. DIMENSIONALITY REDUCTION

Q: What is dimensionality reduction?

Q: What is dimensionality reduction?

A: A set of techniques for reducing the size (in terms of features, records, and/or bytes) of the dataset under examination.

Q: What is dimensionality reduction?

A: A set of techniques for reducing the size (in terms of features, records, and/or bytes) of the dataset under examination.

In general, the idea is to regard the dataset as a matrix and to decompose the matrix into simpler, meaningful pieces.

Q: What is dimensionality reduction?

A: A set of techniques for reducing the size (in terms of features, records, and/or bytes) of the dataset under examination.

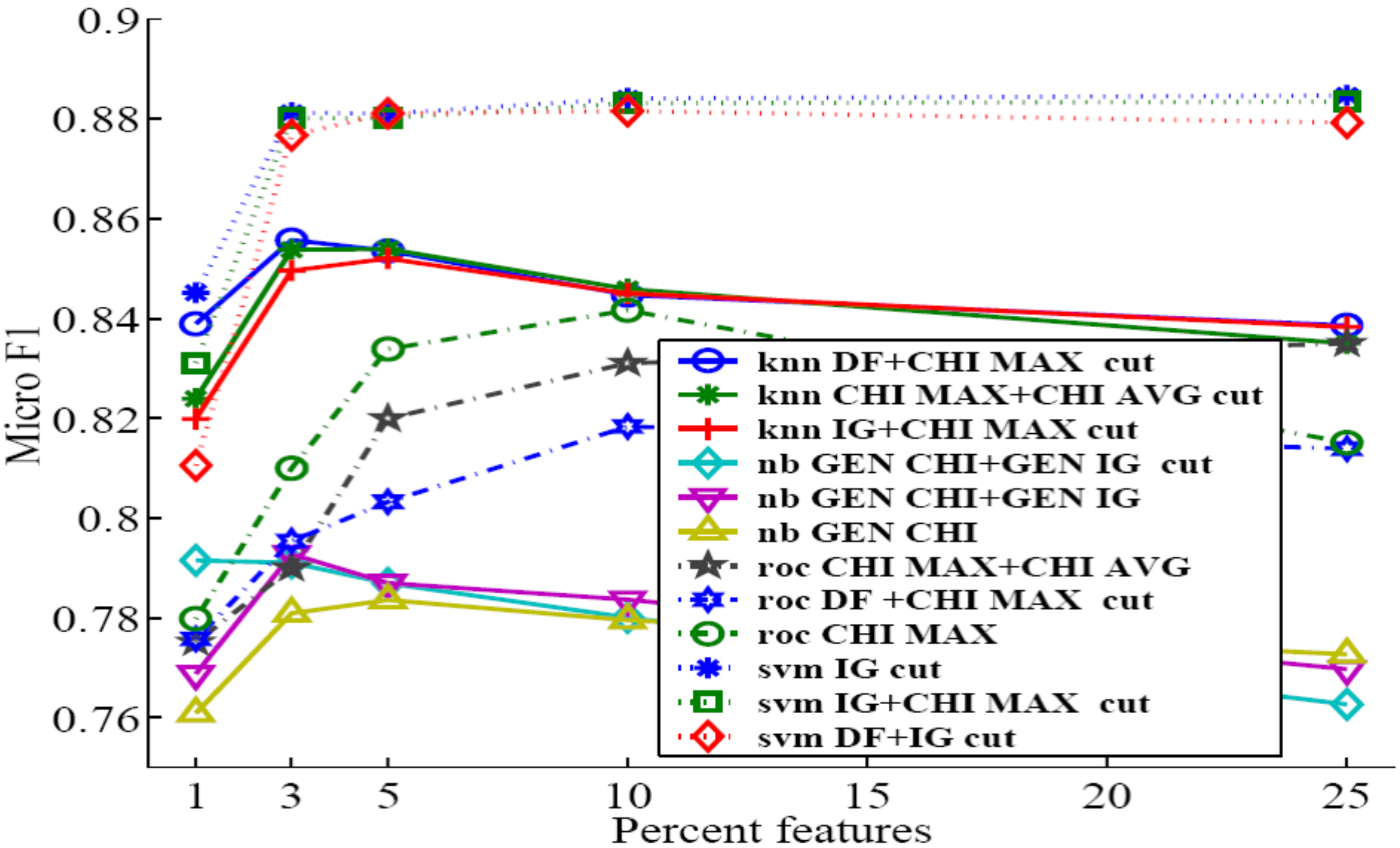
In general, the idea is to regard the dataset as a matrix and to decompose the matrix into simpler, meaningful pieces.

Dimensionality reduction is frequently performed as a pre-processing step before another learning algorithm is applied.

Q: What are the goals of dimensionality reduction?

Q: What are the goals of dimensionality reduction?

- *reduce computational expense*
- *reduce susceptibility to overfitting*
- *reduce noise in the dataset*
- *enhance our intuition*



0.9
0.88
0.86
0.84
0.82
0.8
0.78
0.76
Micro F1
1
3
5
10
15
20
25
Percent features
knn DF+CHI MAX cut
knn CHI MAX+CHI AVG cut
knn IG+CHI MAX cut
nb GEN CHI+GEN IG cut
nb GEN CHI+GEN IG
nb GEN CHI
roc CHI MAX+CHI AVG
roc DF +CHI MAX cut
roc CHI MAX
svm IG cut
svm IG+CHI MAX cut
svm DF+IG cut

Q: How is dimensionality reduction performed?

Q: How is dimensionality reduction performed?

A: There are two approaches: feature selection and feature extraction.

Q: How is dimensionality reduction performed?

A: There are two approaches: feature selection and feature extraction.

feature selection – *selecting a subset of features using an external criterion (filter) or the learning algo accuracy itself (wrapper)*

feature extraction – *mapping the features to a lower dimensional space*

II. PRINCIPAL COMPONENT ANALYSIS

Principal component analysis is a dimension reduction technique that can be used on a matrix of any dimensions.

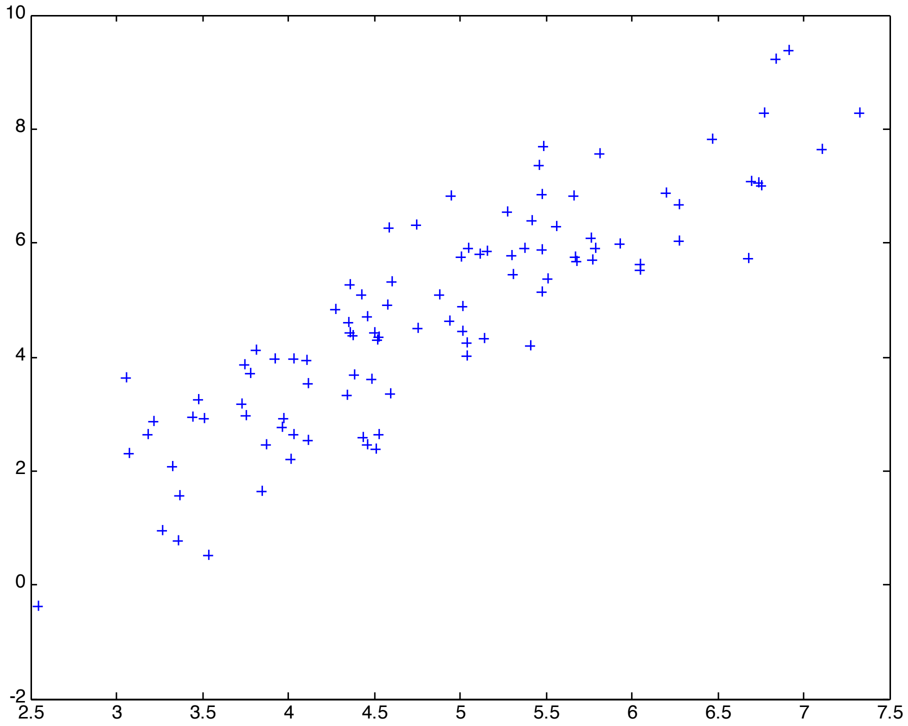
Principal component analysis is a dimension reduction technique that can be used on a matrix of any dimensions.

This procedure produces a new basis, each of whose components retain as much variance from the original data as possible.

Principal component analysis is a dimension reduction technique that can be used on a matrix of any dimensions.

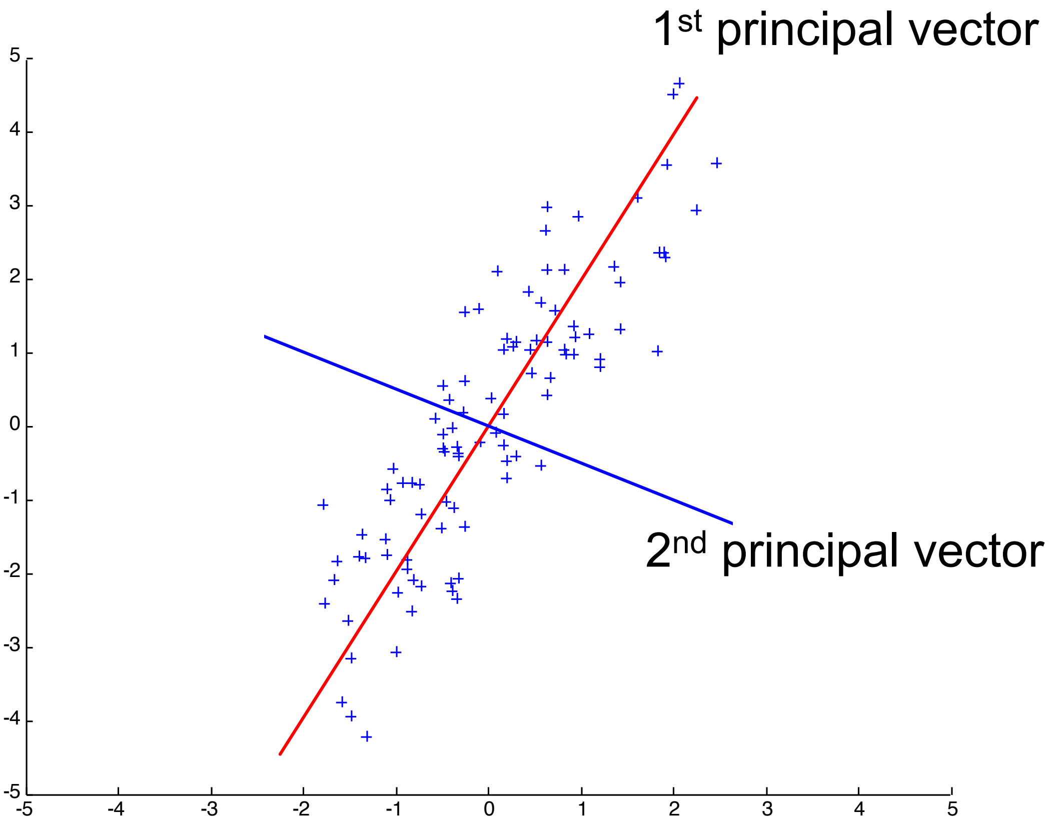
This procedure produces a new basis, each of whose components retain as much variance from the original data as possible.

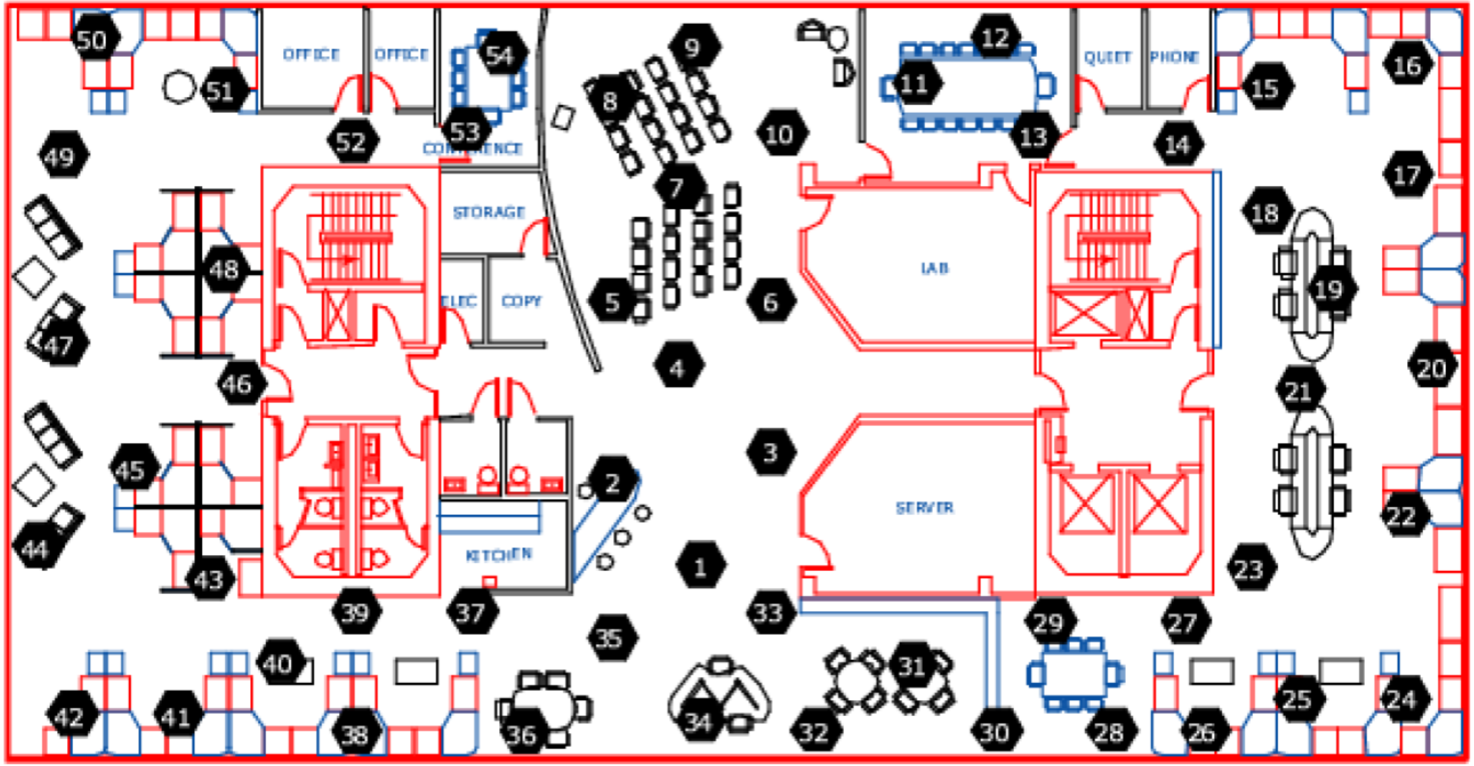
The PCA of a matrix A *boils down to the* **eigenvalue decomposition** *of the* **covariance matrix** *of* A.



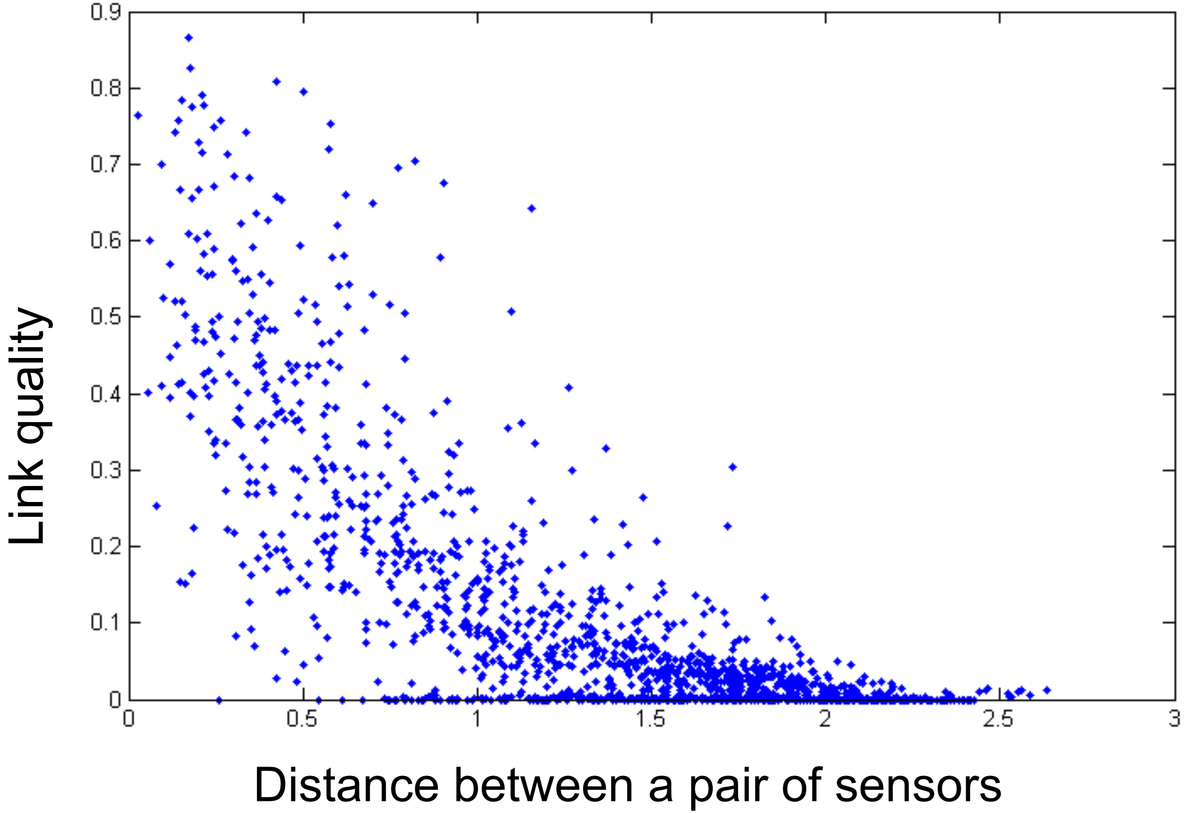
10
8
6
4
2
0
-2
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7
7.5

- Gives best axis to project
- Minimum RMS error
- Principal vectors are orthogonal

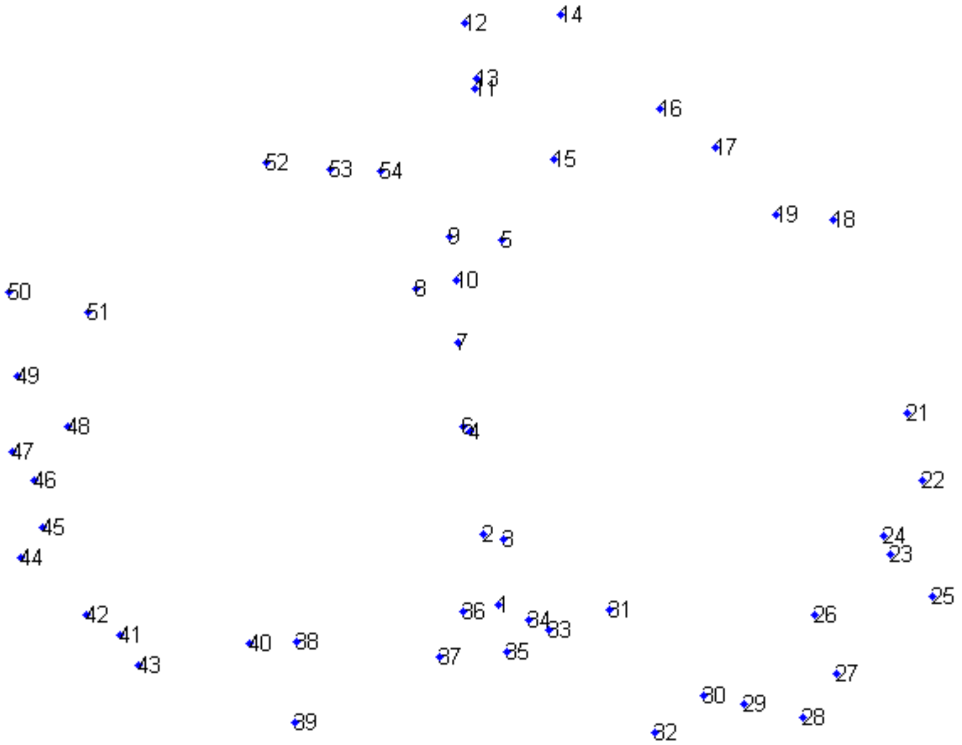


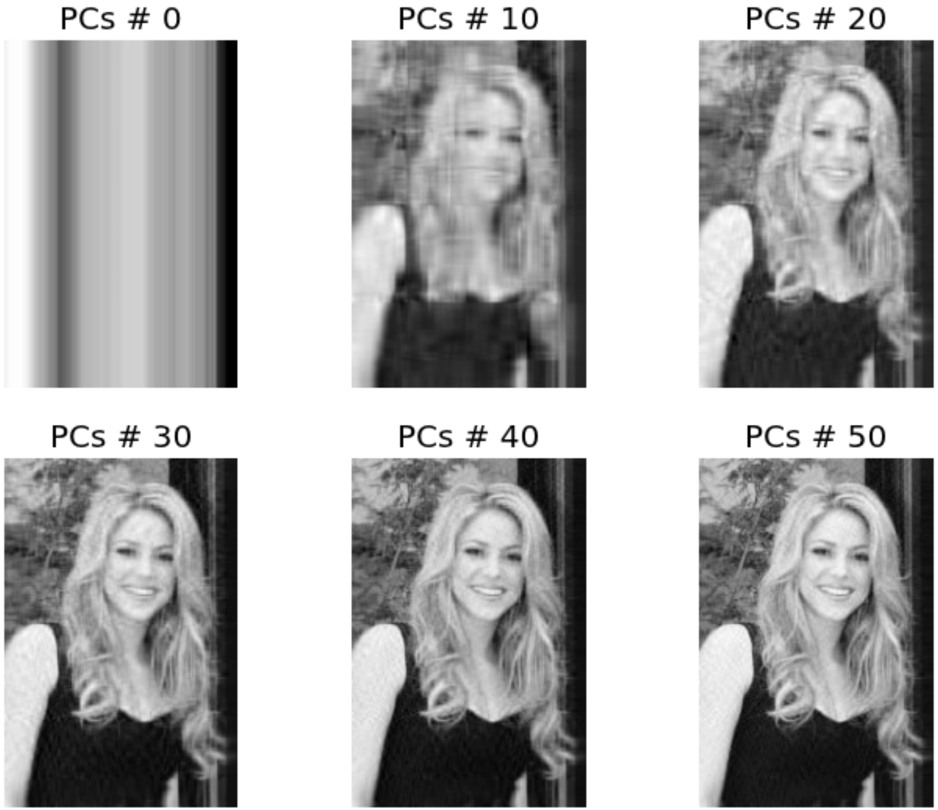


Sensors in Intel Berkeley Lab


0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.5
1
1.5
2
2.5
3
Link quality
Distance between a pair of sensors

- Given a 54x54 matrix of pairwise link qualities
- Do PCA
- Project down to 2 principal dimensions
- PCA discovered the map of the lab





ource: http://glowingpython.blogspot.it/2011/07/pca-and-image-compression-with-numpy.html

- PCA algorithm:
  - 1. X ← Create N x d data matrix, with one row vector x_n per data point
  - 2. X subtract mean x from each row vector x_n in X
  - 3. Σ ← covariance matrix of X
  - Find eigenvectors and eigenvalues of Σ
  - PC's ← the M eigenvectors with largest eigenvalues

- What if very large dimensional data?
  - e.g., Images ($d \geq 10^4$)
- Problem:
  - Covariance matrix Σ is size (d^2)
  - $d=10_4 \rightarrow |\Sigma| = 10^8$
- Singular Value Decomposition (SVD)!
  - efficient algorithms available
  - some implementations find just top N eigenvectors

III. SINGULAR VALUE DECOMPOSITION

Singular Value Decomposition

- Problem:
  - #1: Find concepts in text
  - #2: Reduce dimensionality

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 - Definition

$$\mathbf{A}_{[n \times m]} = \mathbf{U}_{[n \times r]} \, \Lambda_{[\, r \times r]} \, (\mathbf{V}_{[m \times r]})^{T}$$

- **A**: *n x m* matrix (e.g., n documents, m terms)
- **U**: *n x r* matrix (n documents, r concepts)
- Λ: *r x r* diagonal matrix (strength of each 'concept') (r: rank of the matrix)
- **V**: m x r matrix (m terms, r concepts)

SVD - Properties

THEOREM [Press+92]: always possible to decompose matrix **A** into **A** = **U** Λ **V**^{T} , where

- **U,** Λ, **V**: unique (*)
- **U**, **V**: column orthonormal (ie., columns are unit vectors, orthogonal to each other)
  - **U**^{T}**U** = **I**; **V**^{T}**V** = **I** (**I:** identity matrix)
- Λ**:** singular value are positive, and sorted in decreasing order

SVD - Interpretation

'documents', 'terms' and 'concepts':

- **U**: document-to-concept similarity matrix
- **V**: term-to-concept similarity matrix
- Λ: its diagonal elements: 'strength' of each concept

Projection:

- best axis to project on: ('best' = min sum of squares of projection errors)

SVD - Example

- **A** = **U** Λ **V**^{T} - example:

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

SVD - Example

- **A = U Λ V^T** - example:

doc-to-concept similarity matrix

CS-concept

MD-concept

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

SVD - Example

- **A** = **U** Λ **V**^{T} - example:

'strength' of CS-concept

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

SVD - Example

- **A** = **U** Λ **V**^{T} - example:

term-to-concept similarity matrix

CS-concept

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

SVD – Dimensionality reduction

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

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

SVD - Dimensionality reduction

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

SVD - Dimensionality reduction

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

LSI (latent semantic indexing)

Q1: How to do queries with LSI?

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

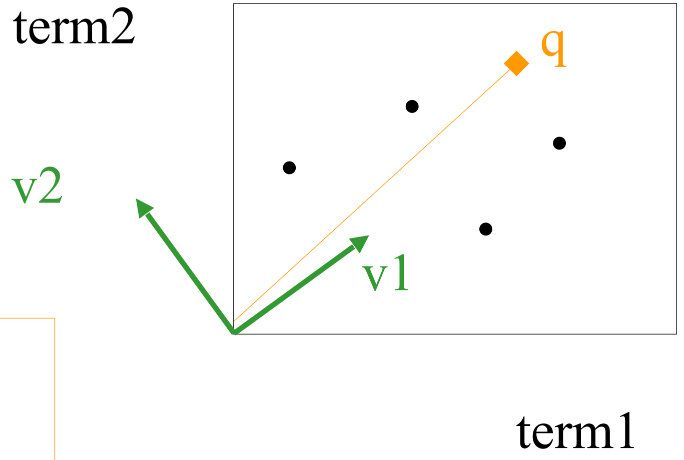
$$
\begin{array}{c}
\begin{array}{ccccc} \text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung} \end{array} \\
\begin{array}{c} \text{CS} \\ \\ \\ \text{MD} \end{array}
\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2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

LSI (latent semantic indexing)

Q: How to do queries with LSI?

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

$$
\begin{array}{cccccc}
 & \text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung} \\
q= & \bigg[\,1 & 0 & 0 & 0 & 0\,\bigg]
\end{array}
$$



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

LSI (latent semantic indexing)

compactly, we have:

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

e.g.:

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

$$
\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.58 & 0 \\ 0.58 & 0 \\ 0.58 & 0 \\ 0 & 0.71 \\ 0 & 0.71 \end{bmatrix} = \begin{array}{c} \text{CS-concept} \\ \begin{bmatrix} 0.58 & 0 \end{bmatrix} \end{array}
$$

term-to-concept similarities

Multi-lingual IR (English query, on Spanish text?)

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

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

Little example

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

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

Eg:

CS-concept

$$d = \begin{array}{ccccc} \text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung} \\ \end{array} \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.58 & 0 \\ 0.58 & 0 \\ 0.58 & 0 \\ 0 & 0.71 \\ 0 & 0.71 \end{bmatrix} = \begin{bmatrix} 1.16 & 0 \end{bmatrix}$$

term-to-concept similarities

Little example

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

$$\begin{array}{lcl}
 & \begin{array}{ccccc}\text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung}\end{array} & \begin{array}{cc}\text{CS-concept} & \end{array} \\
d= & \begin{bmatrix}0 & 1 & 1 & 0 & 0\end{bmatrix} & \rightarrow \begin{bmatrix}1.16 & 0\end{bmatrix} \\
q= & \begin{bmatrix}1 & 0 & 0 & 0 & 0\end{bmatrix} & \rightarrow \begin{bmatrix}0.58 & 0\end{bmatrix}
\end{array}$$

Multi-lingual IR

- Solution: ~ LSI

	data	inf.	retrieval	brain	lung	datos	informacion			
CS	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
MD	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

- Concatenate documents
- Do SVD on them
- Now when a new document comes project it into concept space
- Measure similarity in concept space