GENERAL ASSEMBLY

INTRO TO DATA SCIENCE

LECTURE 16: DIMENSION REDUCTION

DATA EXPLORATION

SUPERVISED LEARNING: REGRESSION

SUPERVISED LEARNING: CLASSIFICATION

UNSUPERVISED LEARNING

- CLUSTERING
- DIMENSION REDUCTION (TODAY)

VARIOUS TOPICS

0. PRESENTATIONS DATA EXPLORATION FOR FINAL PROJECT

I. DIMENSIONALITY REDUCTION

II. SINGULAR VALUE DECOMPOSITION (SVD)

III. PRINCIPAL COMPONENT ANALYSIS (PCA)

IV. NOTEBOOK EXAMPLES & EXERCISES

- EXPLAIN PITFALLS OF WORKING IN HIGH DIMENSIONS
- DESCRIBE EXAMPLES OF USEFUL APPLICATIONS OF DIM. RED.
- BE ABLE TO APPLY SVD AND PCA IN PYTHON, AND TO DRAW INFERENCES OF LOWER-DIMENSIONAL STRUCTURES

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) of the dataset under examination.

Q: What is dimensionality reduction?

A: A set of techniques for reducing the size (in terms of features) 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) 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 motivations for dimensionality reduction?

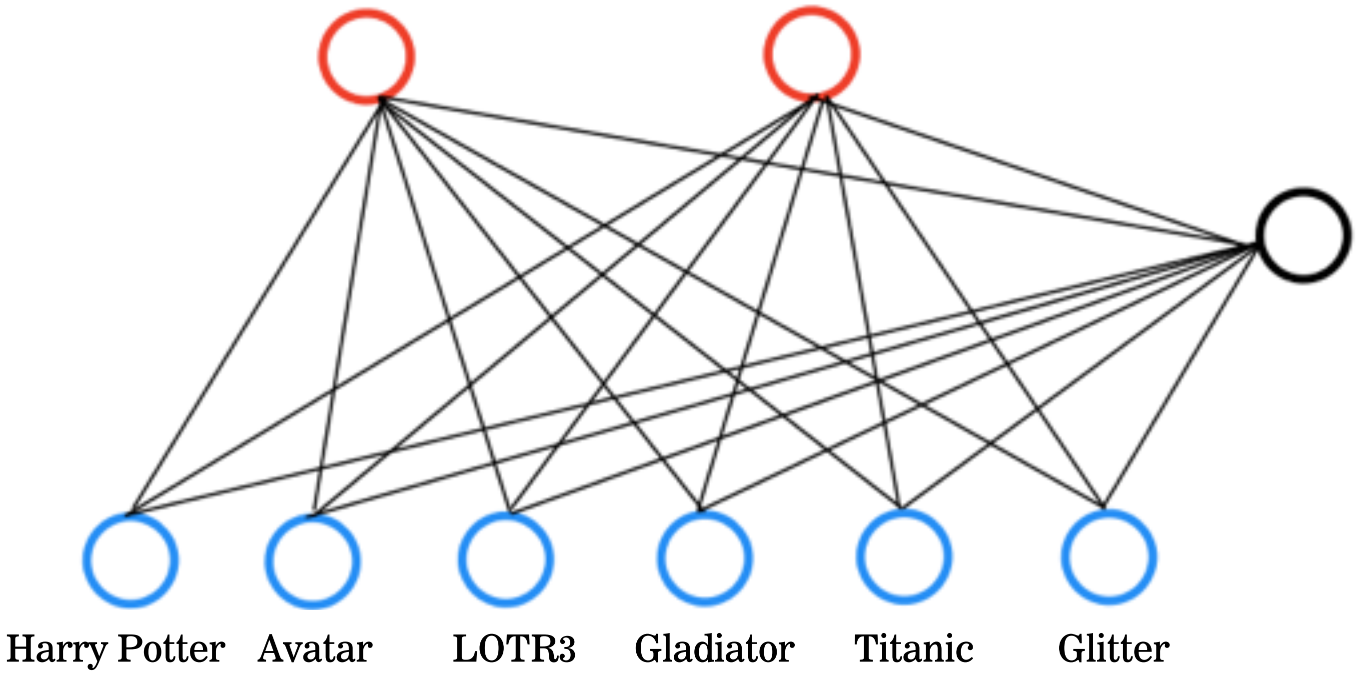
Q: What are the motivations for dimensionality reduction?

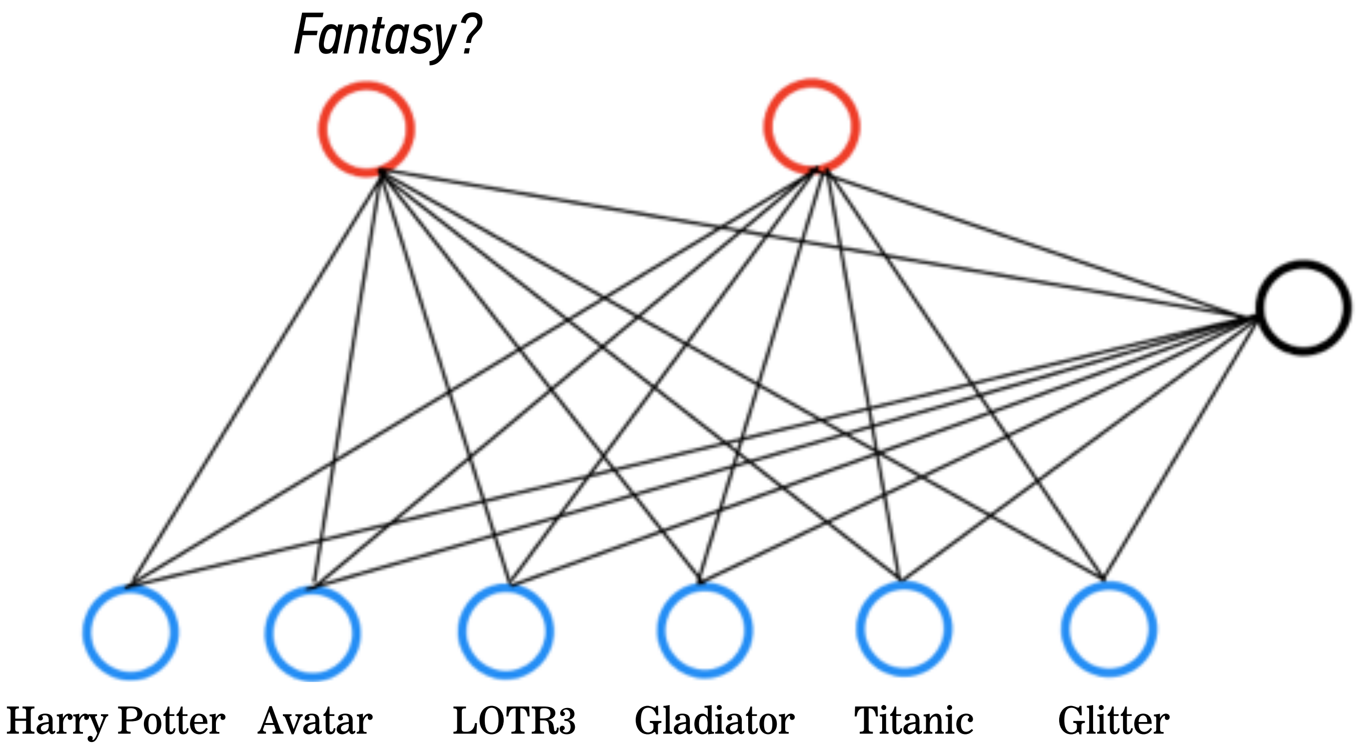
The number of features in our dataset can be difficult to manage, or even misleading (eg, if the relationships are actually simpler than they appear).

We'd like to represent a user's taste profile by a select number of dimensions, rather than their rating of each and every movie

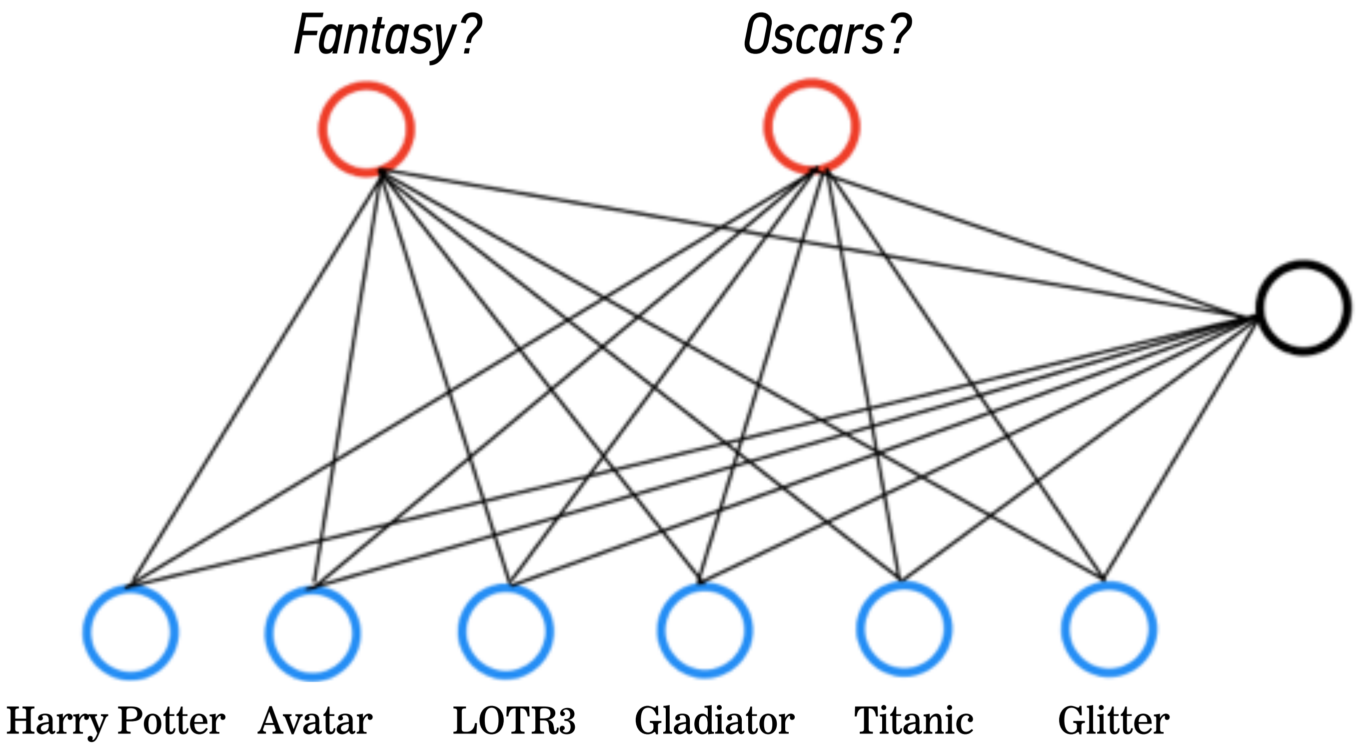
We'd like to represent a user's taste profile by a select number of dimensions, rather than their rating of each and every movie

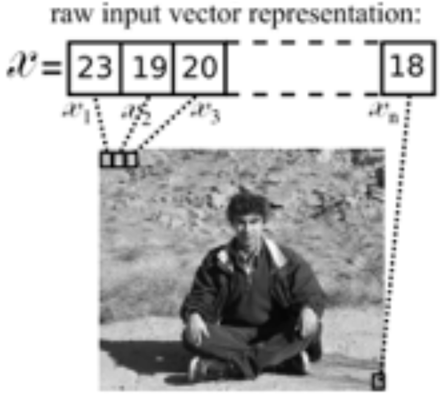
Harry Potter Avatar LOTR3 Gladiator Titanic Glitter


Harry Potter
Avatar
LOTR3
Gladiator
Titanic
Glitter


Fantasy?
Harry Potter
Avatar
LOTR3
Gladiator
Titanic
Glitter

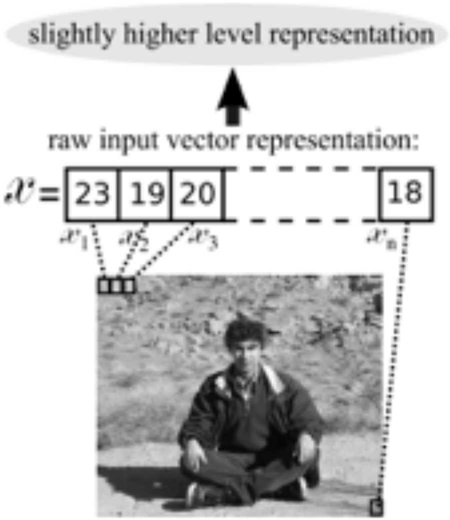
DIMENSIONALITY REDUCTION – EXAMPLE

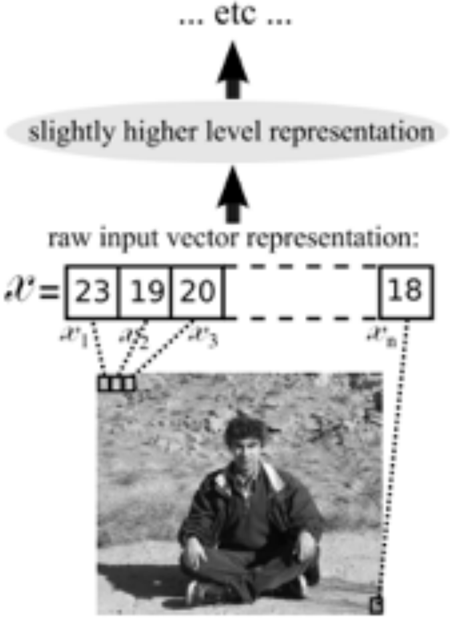




raw input vector representation:
x =
23
19
20
18
x1
x2
x3
xn

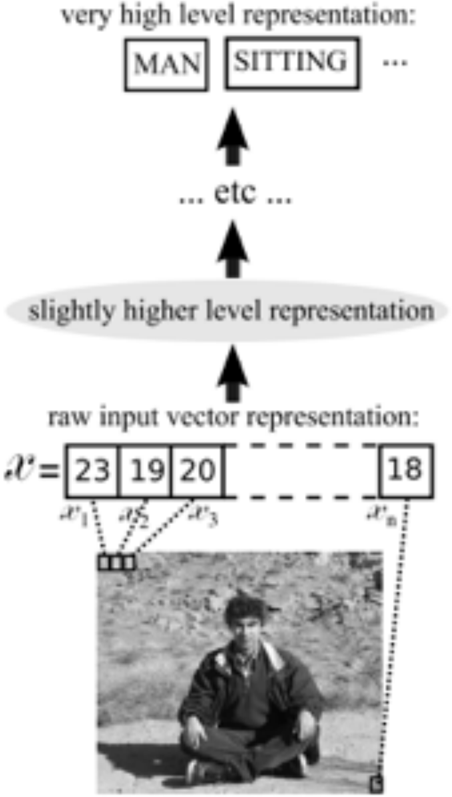
DIMENSIONALITY REDUCTION – EXAMPLE





... etc ...
slightly higher level representation
raw input vector representation:
x = 23 19 20 18
x1 x2 x3 xn

DIMENSIONALITY REDUCTION – EXAMPLE



Q: What is the goal of dimensionality reduction?

Q: What is the goal of dimensionality reduction?

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

The goal of feature extraction is to create a new set of coordinates that simplify the representation of the data.

Q: What are some applications of dimensionality reduction?

Q: What are some applications of dimensionality reduction?

- *document clustering*
- *image recognition/computer vision*
- *recommender systems*

II. SINGULAR VALUE DECOMPOSITION (SVD)

Consider a matrix A with N rows and n features.

Consider a matrix A with N rows and n features.

The **singular value decomposition** *of A is given by:*

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

Consider a matrix A with N rows and n features.

The **singular value decomposition** *of A is given by:*

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

(N x n) (N x N) (N x n) (n x n)

Consider a matrix A with N rows and n features.

The **singular value decomposition** *of A is given by:*

$$\underset{(N \times n)}{A} = \underset{(N \times N)}{U} \; \underset{(N \times n)}{\Sigma} \; \underset{(n \times n)}{V^T}$$

st. U, V are **orthogonal** *matrices and Σ is a* **diagonal** *matrix.*

Consider a matrix A with N rows and n features.

The **singular value decomposition** *of A is given by:*

$$\underset{(N \times n)}{A} = \underset{(N \times N)}{U} \; \underset{(N \times n)}{\Sigma} \; \underset{(n \times n)}{V^T}$$

st. U, V are **orthogonal** *matrices and Σ is a* **diagonal** *matrix.*

→ $UU^T = \mathbf{1},\ VV^T = \mathbf{1}$ → $\Sigma_{ij} = 0 \ \ (i \neq j)$

Consider a matrix A with N rows and n features.

The **singular value decomposition** *of A is given by:*

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

(N x n) (N x N) (N x n) (n x n)

st. U, V are **orthogonal** *matrices and Σ is a* **diagonal** *matrix.*

→ $UU^T = \mathbf{1},\ VV^T = \mathbf{1}$

→ $\Sigma_{ij} = 0 \quad (i \neq j)$

NOTE

Look in the notebook about SVD to dive into the mathematics behind this

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

(N x n) (N x N) (N x n) (n x n)

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

(N x n) (N x N) (N x n) (n x n)

U and V consist of the **eigenvectors** *of AA^T and A^TA, respectively.*

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

$$\underset{(N \times n)}{A} = \underset{(N \times N)}{U} \; \underset{(N \times n)}{\Sigma} \; \underset{(n \times n)}{V^T}$$

U and V consist of the **eigenvectors** *of AA^T and A^TA, respectively.*

The nonzero entries of Σ are the **singular values** *of A. These are real, nonnegative, and rank-ordered (decreasing from left to right).*

$$\underset{(N \times n)}{A} = \underset{(N \times N)}{U} \; \underset{(N \times n)}{\Sigma} \; \underset{(n \times n)}{V^T}$$

U and V consist of the **eigenvectors** *of AA^T and A^TA, respectively.*

The nonzero entries of Σ are the **singular values** *of A. These are real, nonnegative, and rank-ordered (decreasing from left to right).*

These are equal to the **eigenvalues** *of AA^T (or A^TA, equivalently)*

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

(N x n) (N x N) (N x n) (n x n)

U and V consist of the **eigenvectors** *of AA^T and A^TA, respectively.*

The nonzero entries of Σ are the **singular values** *of A. These are real, nonnegative, and rank-ordered (decreasing from left to right).*

These are equal to the **eigenvalues** *of AA^T (or A^TA, equivalently)*

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

(N x n) (N x N) (N x n) (n x n)

Because the singular values are ranked-ordered,
we could **truncate** *the diagonal matrix* Σ *to some dimension k,*
preserving most of the information in A.

$$A \approx U \; \Sigma \; V^T$$

(N x n) ~~(N x N) (N x n) (n x n)~~

(N x d) (d x d) (d x n)

Because the singular values are ranked-ordered,
we could **truncate** *the diagonal matrix* Σ *to some dimension k,*
preserving most of the information in A.

$$A \approx U \Sigma V^T$$

$$\underset{(N \times n)}{A} \approx \underset{\substack{\cancel{(N \times N)} \\ (N \times d)}}{U} \; \underset{\substack{\cancel{(N \times n)} \\ (d \times d)}}{\Sigma} \; \underset{\substack{\cancel{(n \times n)} \\ (d \times n)}}{V^T}$$

Because the singular values are ranked-ordered,
we could **truncate** *the diagonal matrix* Σ *to some dimension k,*
preserving most of the information in A.

With this step, we **reduce the dimensionality** *from n to d.*

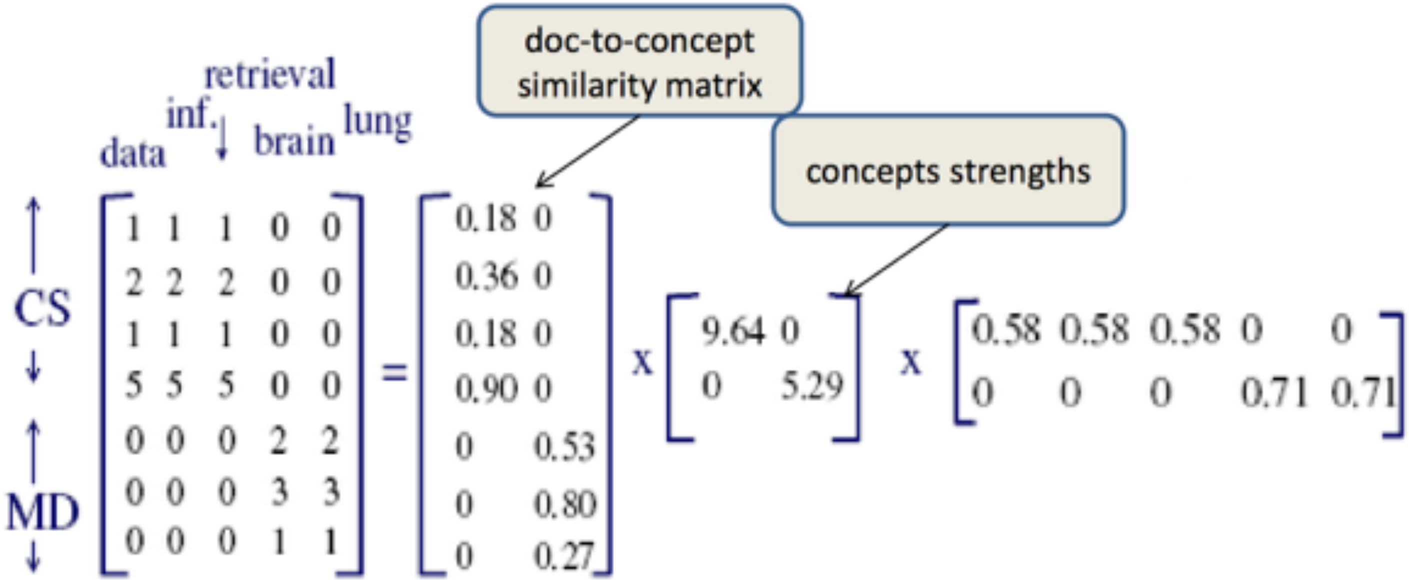
SINGULAR VALUE DECOMPOSITION

$$
\begin{array}{c}
\\
\text{CS} \\
\\
\\
\\
\text{MD} \\
\\
\end{array}
\begin{array}{c}
\begin{array}{ccccc} \text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung} \end{array} \\
\left[\begin{array}{ccccc}
1 & 1 & 1 & 0 & 0 \\
2 & 2 & 2 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 0 & 0 & 2 & 2 \\
0 & 0 & 0 & 3 & 3 \\
0 & 0 & 0 & 1 & 1
\end{array}\right]
\end{array}
$$

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

SINGULAR VALUE DECOMPOSITION

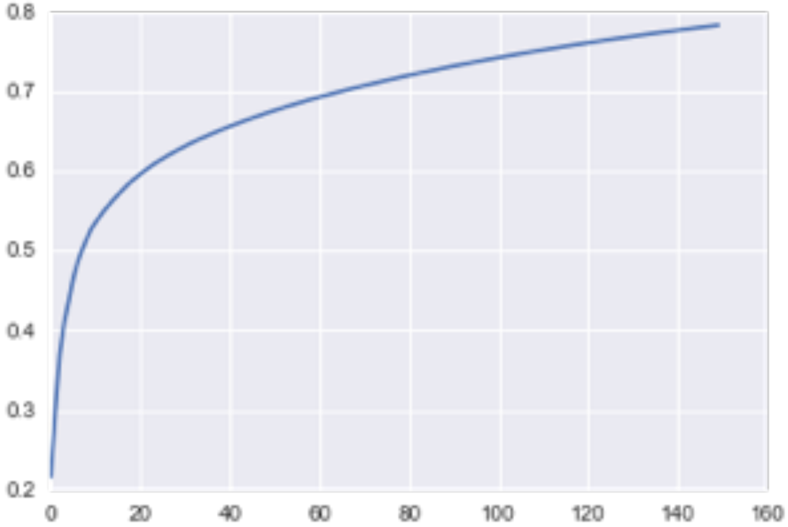
$$
\begin{array}{cc}
 & \begin{array}{ccccc} \text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung} \end{array} \\
\begin{array}{c} \\ \\ \text{CS} \\ \\ \\ \text{MD} \\ \\ \end{array} &
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
2 & 2 & 2 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 \\
5 & 5 & 5 & 0 &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


doc-to-concept similarity matrix
concepts strengths
data
inf.
retrieval
brain
lung
CS
MD
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
=
0.18 0
0.36 0
0.18 0
0.90 0
0 0.53
0 0.80
0 0.27
x
9.64 0
0 5.29
x
0.58 0.58 0.58 0 0
0 0 0 0.71 0.71

$$
\begin{array}{cc}
 & \begin{matrix} \text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung} \end{matrix} \\
\begin{matrix} \\ \text{CS} \\ \\ \\ \\ \text{MD} \\ \\ \end{matrix} &
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
2 & 2 & 2 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 0 & 0 & 2 & 2 \\
0 & 0 & 0 & 3 & 3 \\
0 & 0 & 0 & 1 & 1
\end{bmatrix}
\end{array}
=
\underbrace{\begin{bmatrix}
0.18 & 0 \\
0.36 & 0 \\
0.18 & 0 \\
0.90 & 0 \\
0 & 0.53 \\
0 & 0.80 \\
0 & 0.27
\end{bmatrix}}_{\text{doc-to-concept similarity matrix}}
\times
\underbrace{\begin{bmatrix}
9.64 & 0 \\
0 & 5.29
\end{bmatrix}}_{\text{concepts strengths}}
\times
\underbrace{\begin{bmatrix}
0.58 & 0.58 & 0.58 & 0 & 0 \\
0 & 0 & 0 & 0.71 & 0.71
\end{bmatrix}}_{\text{term-to-concept similarity matrix}}
$$

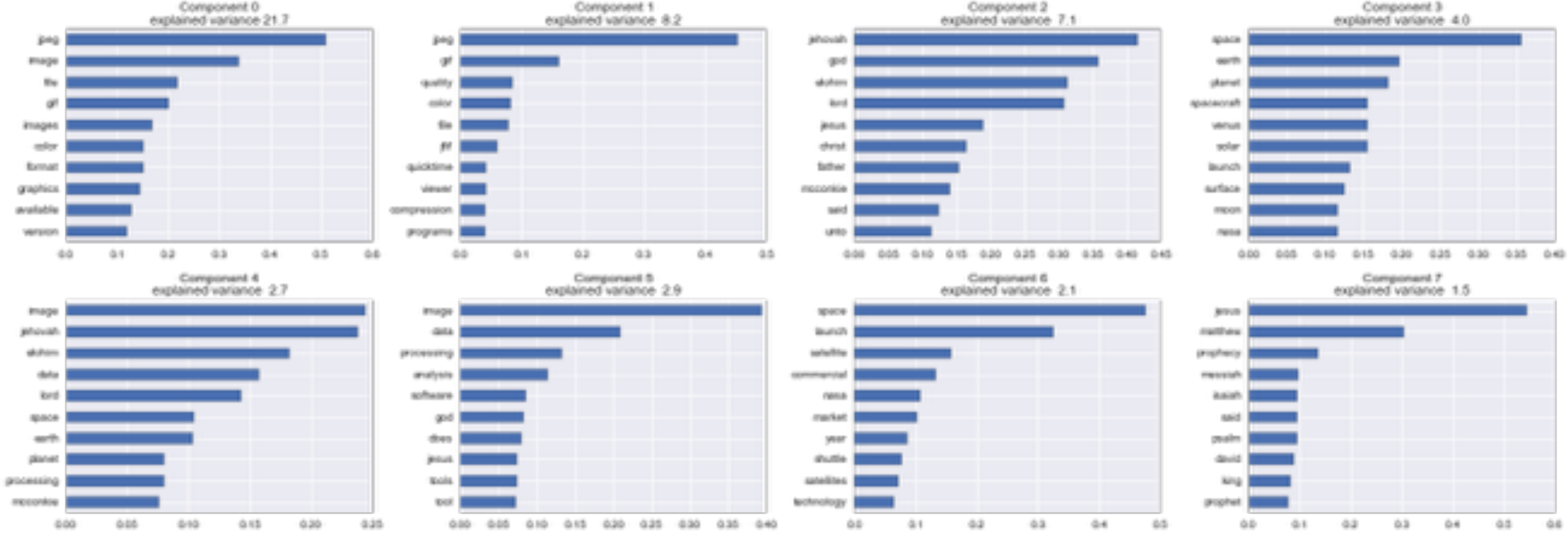
Singular Values by # of components

(cumulative as % of total)



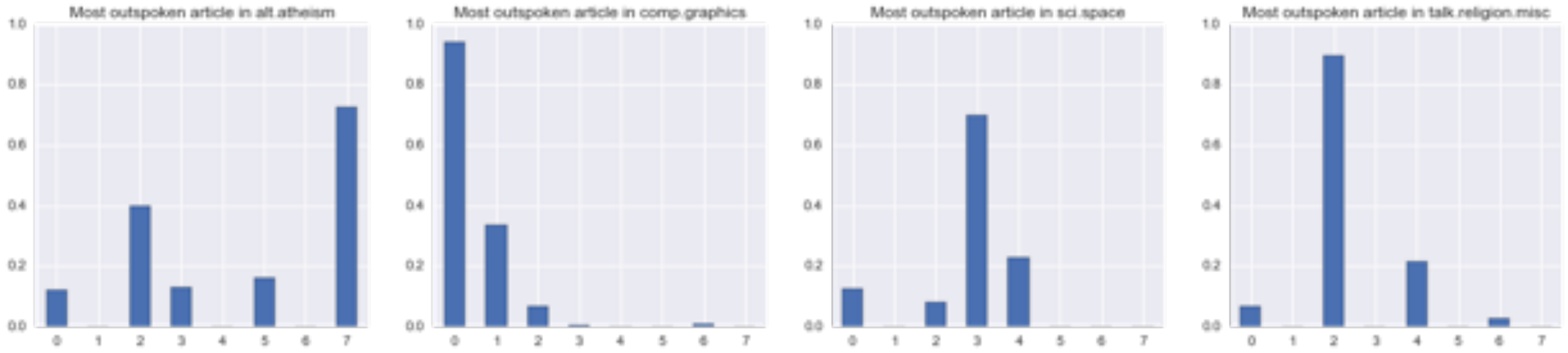
source: 20-newsgroups dataset, analysis in ipython notebook

SINGULAR VALUE DECOMPOSITION



source: 20-newsgroups dataset, analysis in ipython notebook

SINGULAR VALUE DECOMPOSITION



source: 20-newsgroups dataset, analysis in ipython notebook

III. PRINCIPAL COMPONENT ANALYSIS (PCA)

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.

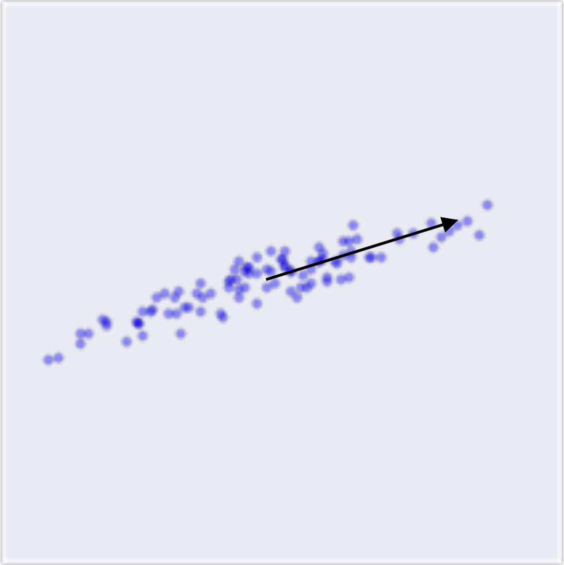
Principal component analysis is a dimension reduction technique that can be used on a matrix of any dimensions.

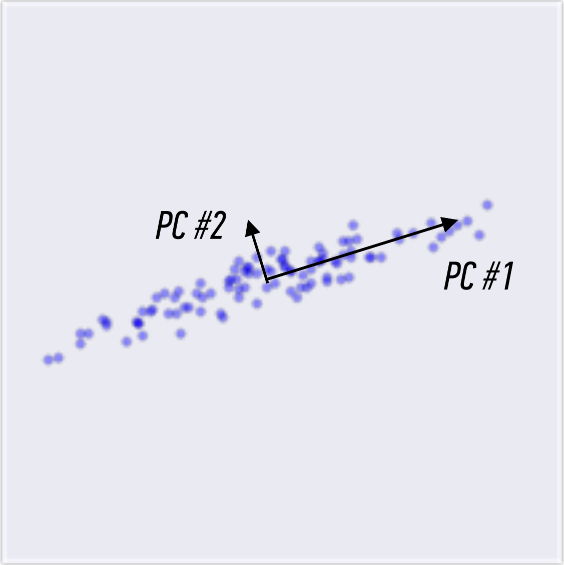
This procedure produces a new basis, each of whose components retain as much variance from the original data as possible.

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

Principal Component Analysis (PCA) seeks the dimensions in which the most variance occurs

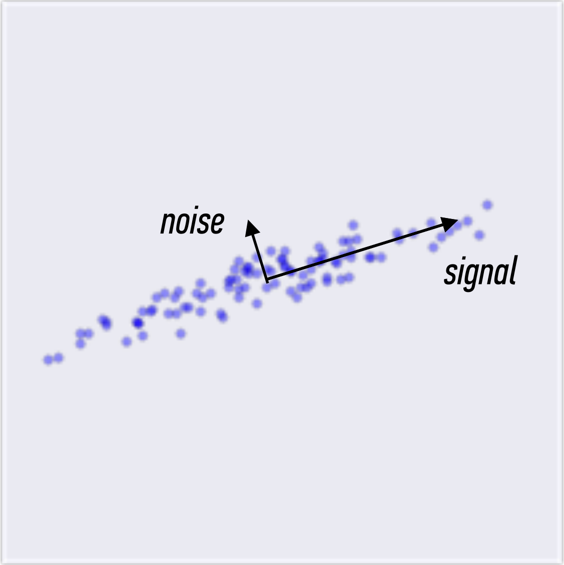
Principal Component Analysis (PCA) seeks the dimensions in which the most variance occurs





Principal Component Analysis (PCA) seeks the dimensions in which the most variance occurs

It can be seen as a transformation to a new orthogonal basis, ordered by variance



Principal Component Analysis (PCA) seeks the dimensions in which the most variance occurs

It can be seen as a transformation to a new orthogonal basis, ordered by variance

The idea is that the first principal components contain the most information, while the latter ones contain noise

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

COVARIANCE MATRICES

The **variance** *measures how far a set of numbers is spread out*

The **variance** *measures how far a set of numbers is spread out*

The variance of ***X*** *is given by* $\mathrm{Var}(X) = E[(X - \mu)^2]$

The **variance** *measures how far a set of numbers is spread out*

The variance of ***X*** *is given by* $\mathrm{Var}(X) = E\,[\,(X - \mu)^2\,]$

The **covariance** *measures how much two variables change together*

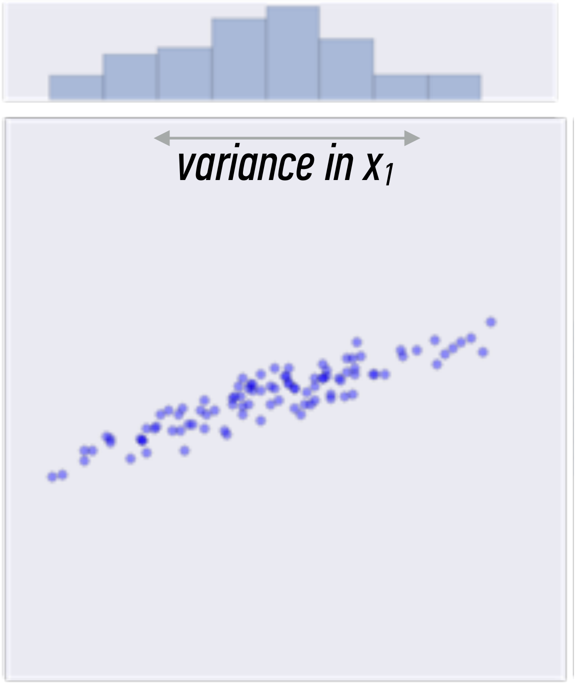
The **variance** *measures how far a set of numbers is spread out*

The variance of ***X*** *is given by* $\mathrm{Var}(X) = E\,[\,(X - \mu)^2\,]$

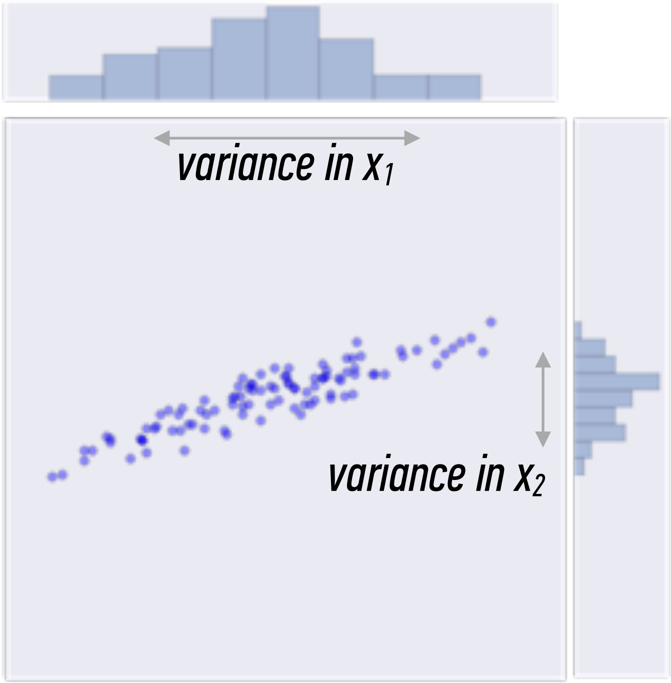
The **covariance** *measures how much two variables change together*

The covariance of ***X*** *and* ***Y*** *is given by* $\mathrm{Cov}(X, Y) = E\,[\,(X - \mu_X)(Y - \mu_Y)\,]$

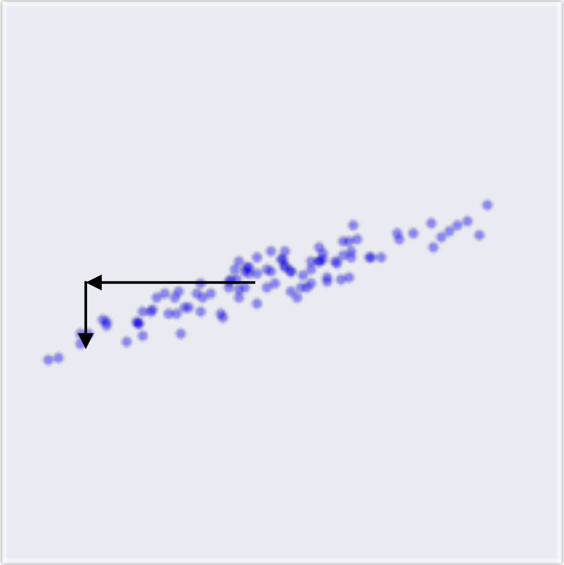
Let’s show that in an example



The **variance** *measures how far a set of numbers is spread out*

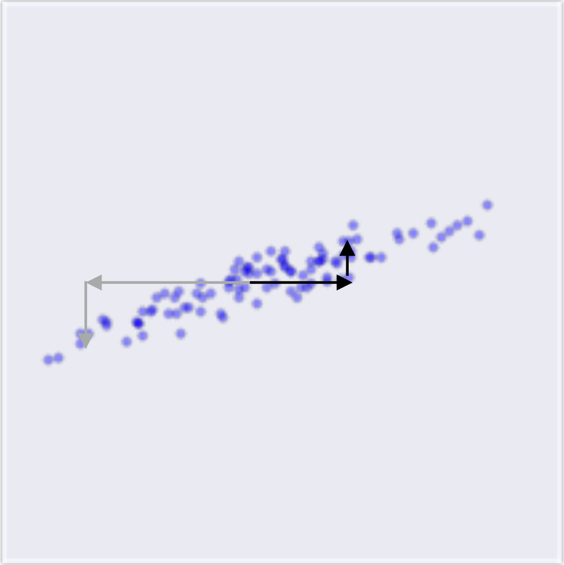


*The **variance** measures how far a set of numbers is spread out*



The **variance** *measures how far a set of numbers is spread out*

The **covariance** *measures how much two variables change together*



*The **variance** measures how far a set of numbers is spread out*

*The **covariance** measures how much two variables change together*

The **covariance matrix** *of a feature matrix* ***X*** *measures how much each pair of features change together*

$$
\mathrm{C} = \begin{bmatrix}
\mathrm{E}[(X_1-\mu_1)(X_1-\mu_1)] & \mathrm{E}[(X_1-\mu_1)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_1-\mu_1)(X_n-\mu_n)] \\
\mathrm{E}[(X_2-\mu_2)(X_1-\mu_1)] & \mathrm{E}[(X_2-\mu_2)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_2-\mu_2)(X_n-\mu_n)] \\
\vdots & \vdots & \ddots & \vdots \\
\mathrm{E}[(X_n-\mu_n)(X_1-\mu_1)] & \mathrm{E}[(X_n-\mu_n)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_n-\mu_n)(X_n-\mu_n)]
\end{bmatrix}
$$

The **covariance matrix** *is always square*

- *diagonal elements c_{ii} give the variance of X_i*
- *off-diagonal elements c_{ij} give the covariance between X_i and X_j $(i \neq j)$*

$$C = \begin{bmatrix} E[(X_1-\mu_1)(X_1-\mu_1)] & E[(X_1-\mu_1)(X_2-\mu_2)] & \cdots & E[(X_1-\mu_1)(X_n-\mu_n)] \\ E[(X_2-\mu_2)(X_1-\mu_1)] & E[(X_2-\mu_2)(X_2-\mu_2)] & \cdots & E[(X_2-\mu_2)(X_n-\mu_n)] \\ \vdots & \vdots & \ddots & \vdots \\ E[(X_n-\mu_n)(X_1-\mu_1)] & E[(X_n-\mu_n)(X_2-\mu_2)] & \cdots & E[(X_n-\mu_n)(X_n-\mu_n)] \end{bmatrix}$$

Note that, if all features are scaled, i.e.,

when the mean of each feature $\boldsymbol{\mu} = E[\boldsymbol{X}]$ is equal to 0,

that we can write the covariance matrix as

$$\boldsymbol{C} = \boldsymbol{X}\,\boldsymbol{X}^T$$

Now write the **eigenvalue decomposition** *of the covariance matrix*

$$C = Q \Lambda Q^{-1}$$

Now write the **eigenvalue decomposition** *of the covariance matrix*

$$C = Q\Lambda Q^{-1}$$

Recall, for an eigenvector v *of* C *and its eigenvalue* λ*, we have:*

$$Cv = \lambda v$$

Now write the **eigenvalue decomposition** *of the covariance matrix*

$$C = Q \Lambda Q^{-1}$$

Recall, for an eigenvector v of C and its eigenvalue λ, we have:

$$Cv = \lambda v$$

We can write $C = Q \Lambda Q^{-1}$ where

- *The columns of Q are its eigenvalues, and*
- *The matrix Λ is a diagonal matrix containing its eigenvalues*

Now write the **eigenvalue decomposition** *of the covariance matrix*

$$C = Q\Lambda Q^{-1}$$

Recall, for an eigenvector v of C and its eigenvalue λ, we have:

$$Cv = \lambda v$$

We can write $C = Q\Lambda Q^{-1}$ where

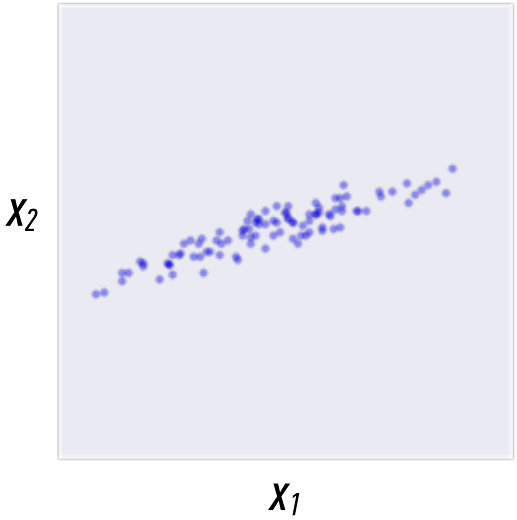
- *The columns of Q are its eigenvalues, and*
- *The matrix Λ is a diagonal matrix containing its eigenvalues*

NOTE

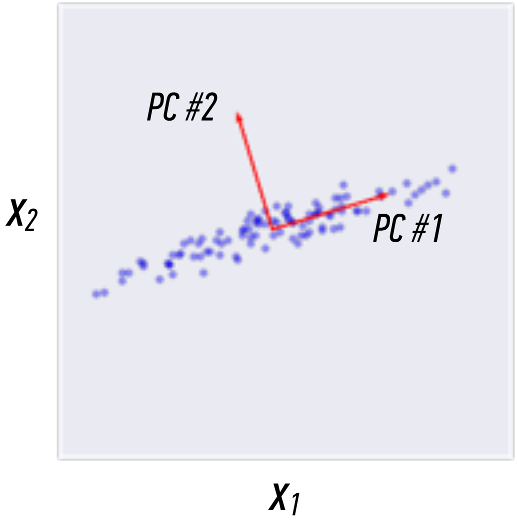
You can think of this as a change of **coordinate systems**. With these new coordinates, the matrix C simply scales vectors along the axes (i.e. no rotations)

BACK TO PCA

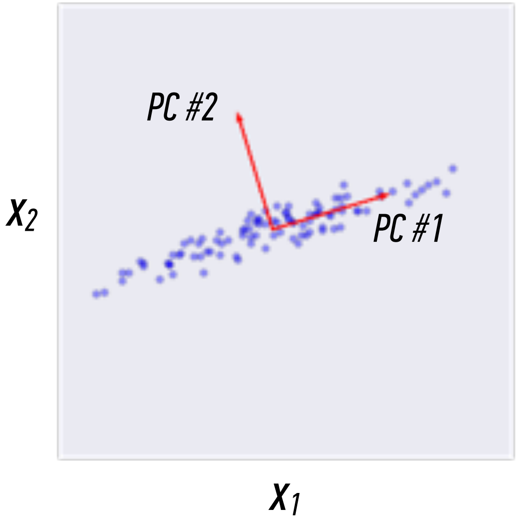
- *Principal Component Analysis (PCA) seeks the dimensions in which the most variance occurs*

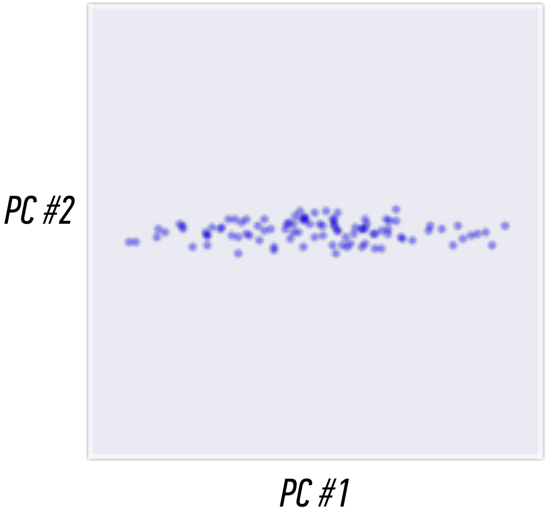


- *Principal Component Analysis (PCA) seeks the dimensions in which the most variance occurs*

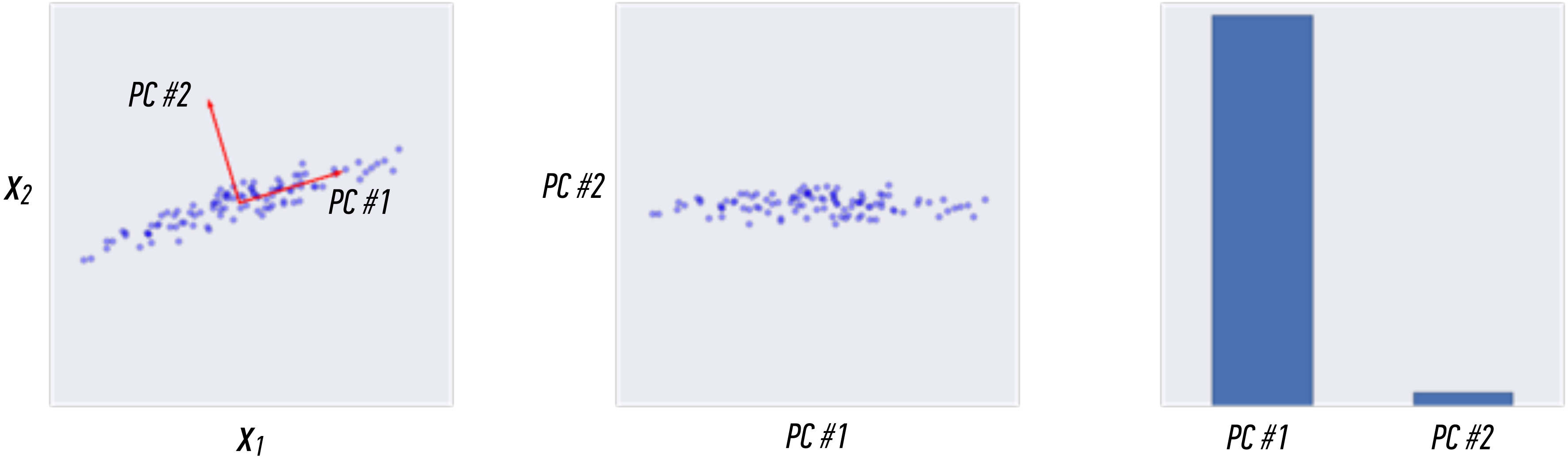


- *Principal Component Analysis (PCA) seeks the dimensions in which the most variance occurs*
- *It can be seen as a transformation to a new orthogonal basis*

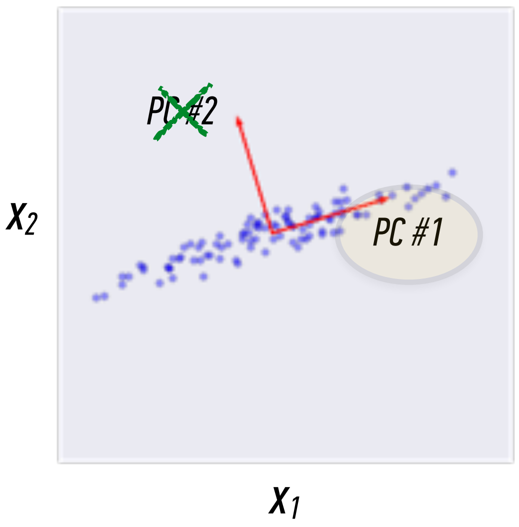


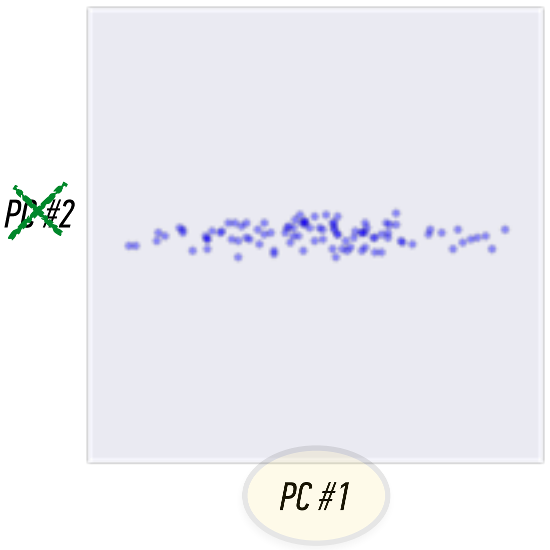


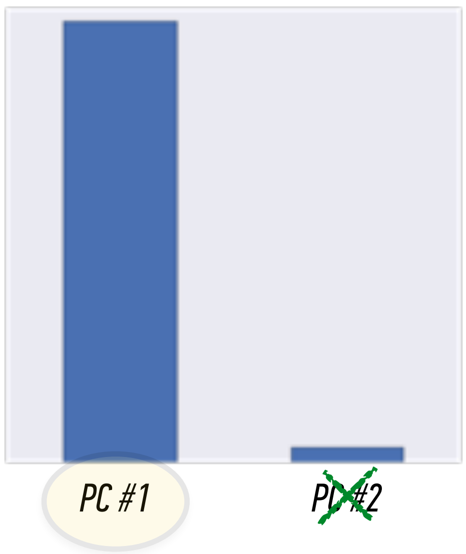
- *Principal Component Analysis (PCA) seeks the dimensions in which the most variance occurs*
- *It can be seen as a transformation to a new orthogonal basis*
- *The principal components are ordered by the size of their variance*



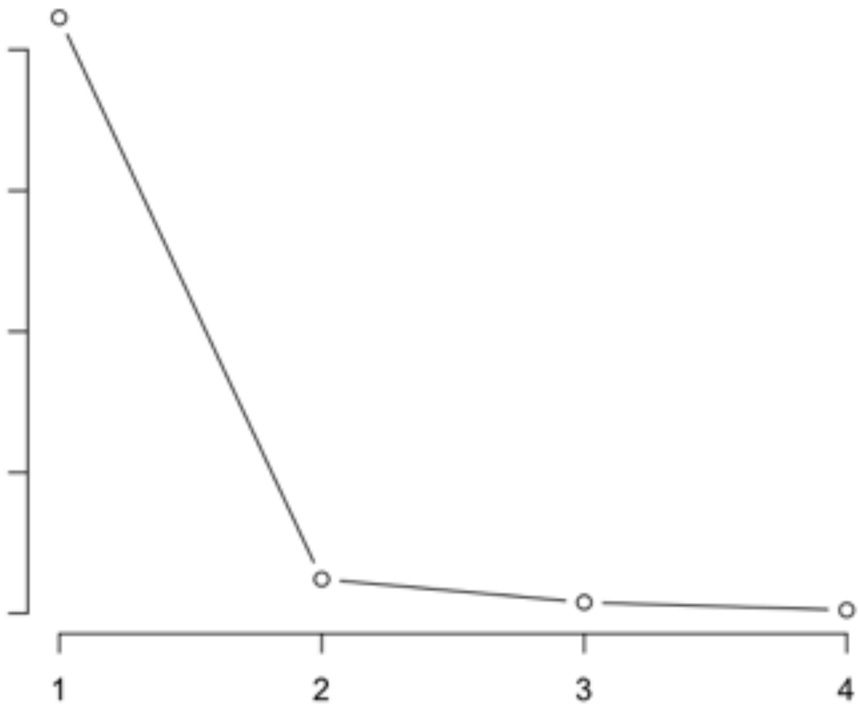
We can now **reduce the dimension** *by only looking at the first few principal components that explain the most variance*



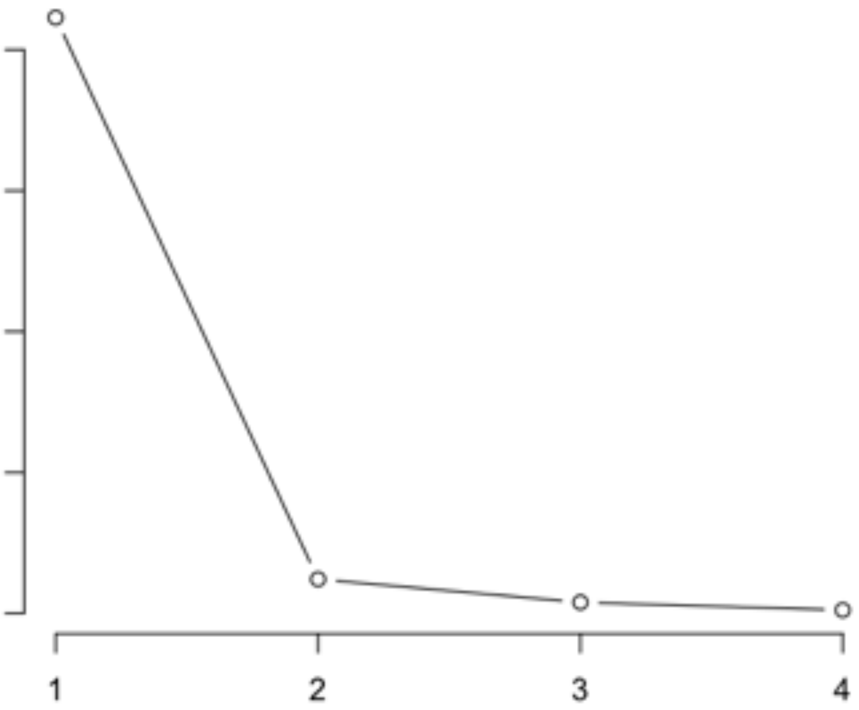




Principal components of Iris dataset



Principal components of Iris dataset



NOTE

Looking at this plot also gives you an idea of how many principal components to keep.

Apply the ***elbow*** *test*: keep only those pc's that appear to the left of the elbow in the graph.

RELATION TO SVD

What is the relationship between PCA and SVD?

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

singular value decomposition of A

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

singular value decomposition of A

$$AA^T$$

covariance matrix of A
(assuming features are scaled)

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

singular value decomposition of A

$$AA^T = (U\Sigma V^T)(V\Sigma U^T)$$

covariance matrix of A
(assuming features are scaled)

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

singular value decomposition of A

$$AA^T = (U\Sigma V^T)(V\Sigma U^T)$$

covariance matrix of A
(assuming features are scaled)

$$= U\Sigma^2 U^T$$

Using $AA^T = \mathbf{1}$

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

singular value decomposition of A

$$AA^T = (U\Sigma V^T)(V\Sigma U^T)$$

covariance matrix of A
(assuming features are scaled)

$$= U\Sigma^2 U^T$$

eigenvalue decomposition of AA^T

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

singular value decomposition of A

$$AA^T = (U\Sigma V^T)(V\Sigma U^T)$$

covariance matrix of A
(assuming features are scaled)

$$= U\Sigma^2 U^T$$

eigenvalue decomposition of AA^T

eigenvectors of AA^T
(base transformation)

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

singular value decomposition of A

$$AA^T = (U\Sigma V^T)(V\Sigma U^T)$$

covariance matrix of A
(assuming features are scaled)

$$= U\Sigma^2 U^T$$

eigenvalue decomposition of AA^T

eigenvectors of AA^T
(base transformation)

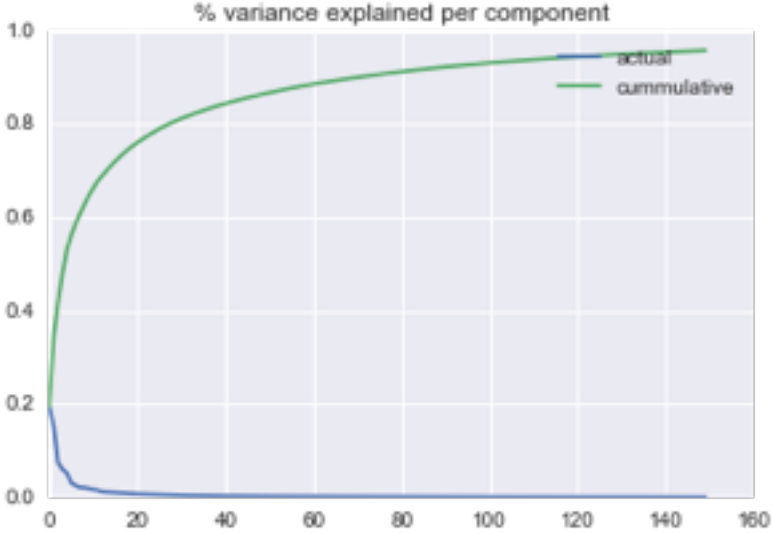
eigenvalues of AA^T
(variance of dimension)

EIGENFACES

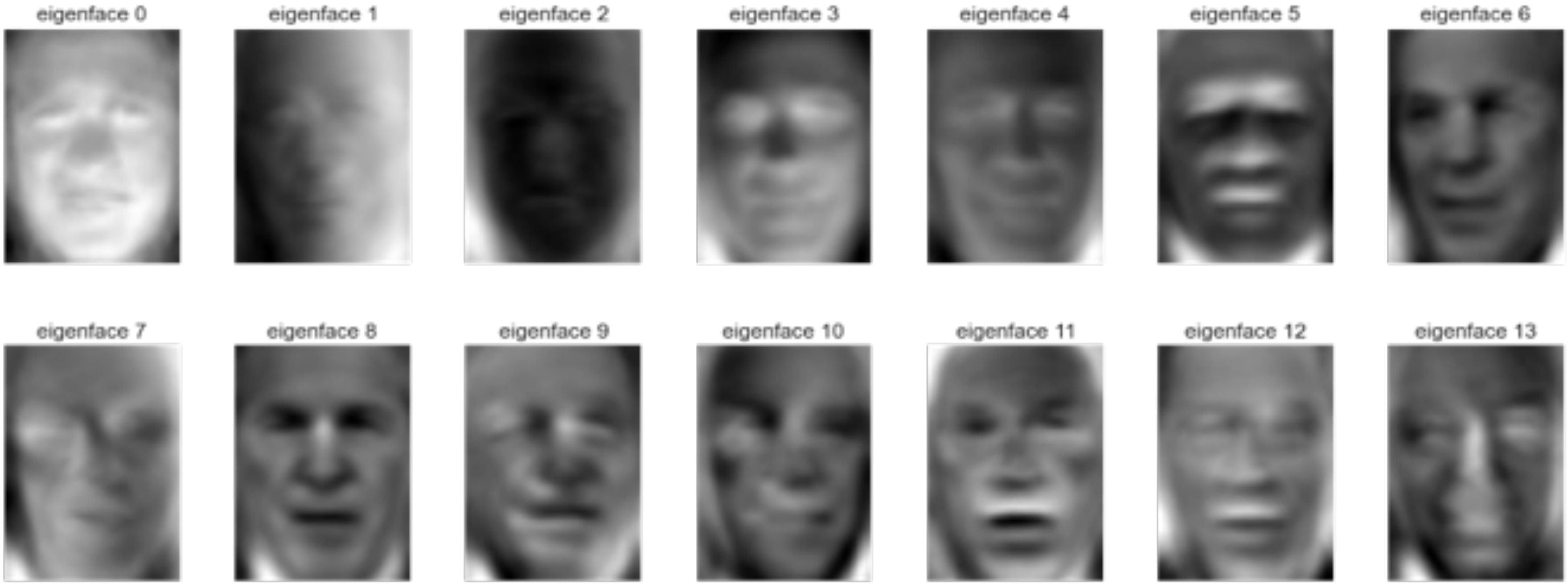

Ariel Sharon
77 images (5%)
Colin Powell
236 images (18%)
Donald Rumsfeld
121 images (9%)
George W Bush
530 images (41%)
Gerhard Schroeder
109 images (8%)
Hugo Chavez
71 images (5%)
Tony Blair
144 images (11%)

Average face

Average face



EIGENFACES


eigenface 0
eigenface 1
eigenface 2
eigenface 3
eigenface 4
eigenface 5
eigenface 6
eigenface 7
eigenface 8
eigenface 9
eigenface 10
eigenface 11
eigenface 12
eigenface 13

Average face

Average face

EIGENFACES

Average face

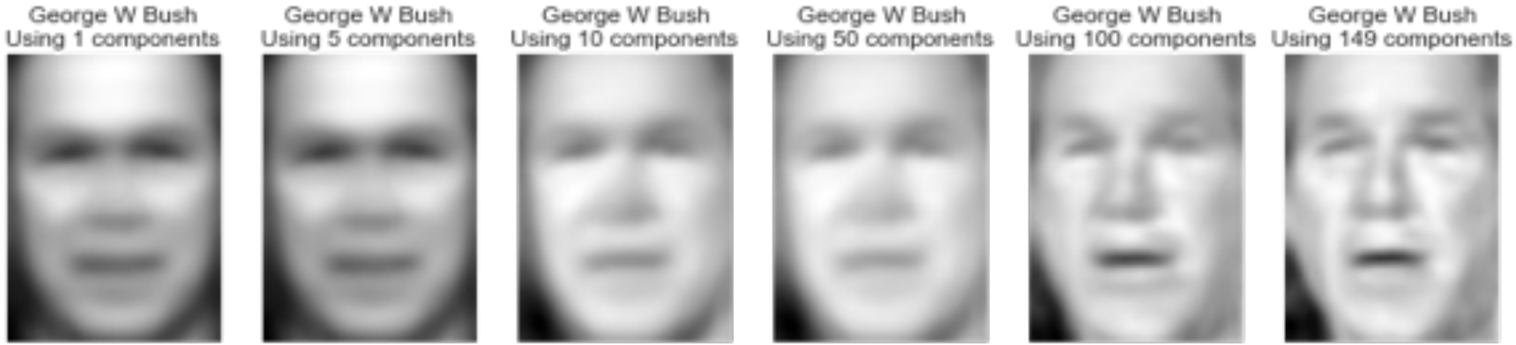
George W Bush
Using 1 components

George W Bush
Using 5 components

George W Bush
Using 10 components

EIGENFACES

Average face



DISCUSSION