# MATH4280

Lecture Notes 1: Singular value decomposition (SVD)

# Singular value decomposition (SVD)

- One of the most important matrix factorizations
- Foundation of many data analytical tools
- Provide a systematic way to determine a low-dimensional approximation to high-dimensional data
  - e.g. image, audio, video, fluid flow
- It is a data-driven approach, patterns are discovered purely from data

# Definition of SVD

- We will analyze a large data set

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_m \\ | & | & & | \end{bmatrix} \qquad \mathbf{X} \in \mathbb{C}^{n \times m}$$

- Each column $\mathbf{x}_k \in \mathbb{C}^n$ represents one data, called snapshot
  - e.g. different images, state of a physical system at different times
- The dimension n is usually very large
- m is the number of snapshots

- The SVD is a unique matrix factorization given by

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$$

- Here, $\mathbf{U} \in \mathbb{C}^{n\times n}$ and $\mathbf{V} \in \mathbb{C}^{m\times m}$ are unitary matrices
- $\mathbf{\Sigma} \in \mathbb{R}^{n\times m}$ is a real matrix, with real and nonnegative entries on diagonal, and zeros off the diagonal

- In most cases, $n \geq m$, so we can write $\mathbf{\Sigma} = \begin{bmatrix} \hat{\mathbf{\Sigma}} \\ \mathbf{0} \end{bmatrix}$

- The columns of U and V are called left singular vectors and right singular vectors respectively

- We can write

$$\mathbf{X} = \mathbf{U\Sigma V}^* = \begin{bmatrix} \hat{\mathbf{U}} & \hat{\mathbf{U}}^{\perp} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{\Sigma}} \\ \mathbf{0} \end{bmatrix} \mathbf{V}^* = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\mathbf{V}^*$$

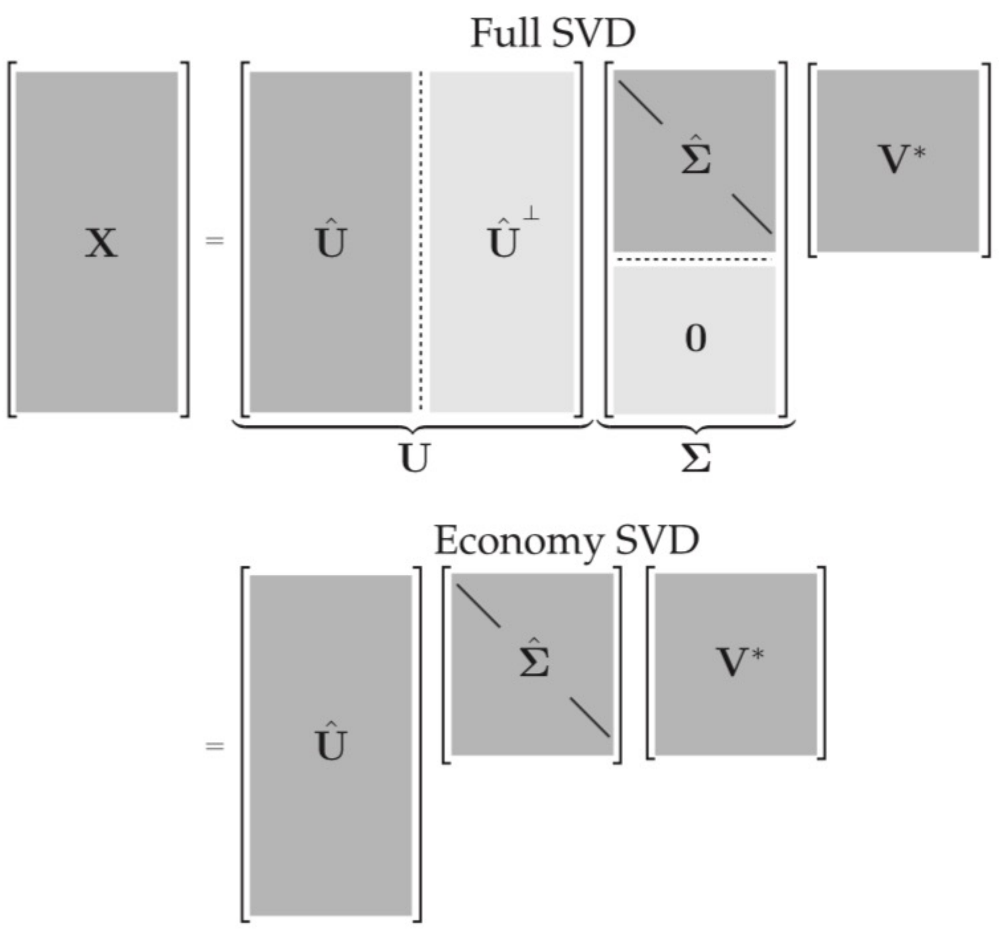
- The diagonal elements of

$$\hat{\mathbf{\Sigma}} \in \mathbb{C}^{m \times m}$$

are called the singular values, arranged from large to small

- The rank of X is equal to the number of nonzero singular values

Schematic of SVD



# Matrix approximation using SVD

- The optimal rank-r approximation to X is given by the rank-r SVD truncation

Frobenius norm

$$\boxed{\underset{\tilde{\mathbf{X}},\ s.t.\ \mathrm{rank}(\tilde{\mathbf{X}})=r}{\operatorname{argmin}}} \boxed{\|\mathbf{X} - \tilde{\mathbf{X}}\|_F} = \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^*$$

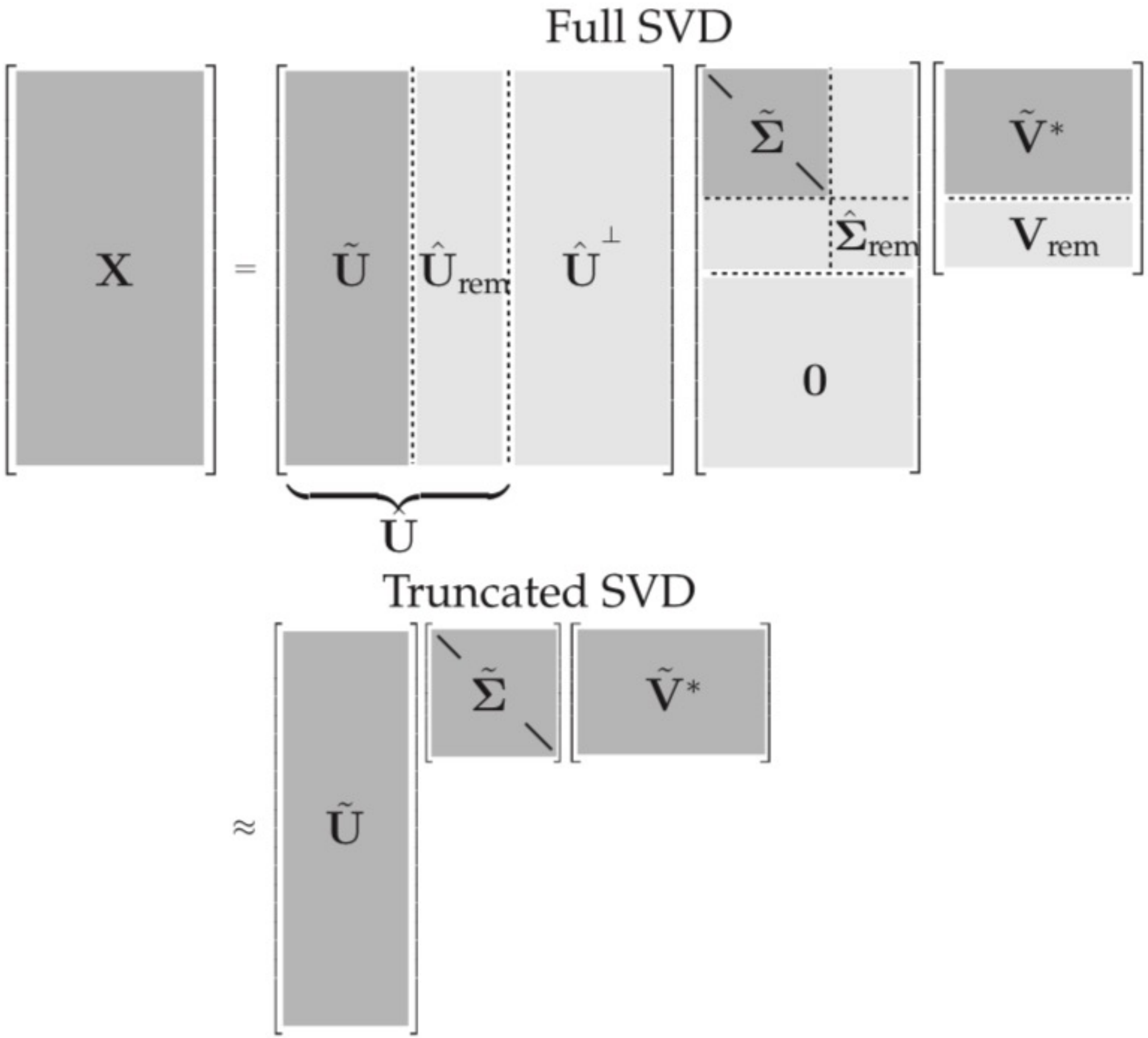
- Here, $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}$ are the first r leading columns of U and V
- $\tilde{\mathbf{\Sigma}}$ is the leading r x r sub-block of $\mathbf{\Sigma}$

---

- We can also use the following formula

$$\tilde{\mathbf{X}} = \sum_{k=1}^{r} \sigma_k \mathbf{u}_k \mathbf{v}_k^* = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^* + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^* + \cdots + \sigma_r \mathbf{u}_r \mathbf{v}_r^*$$

Q: but how is it possible to compute the first r terms, without calculating all terms?



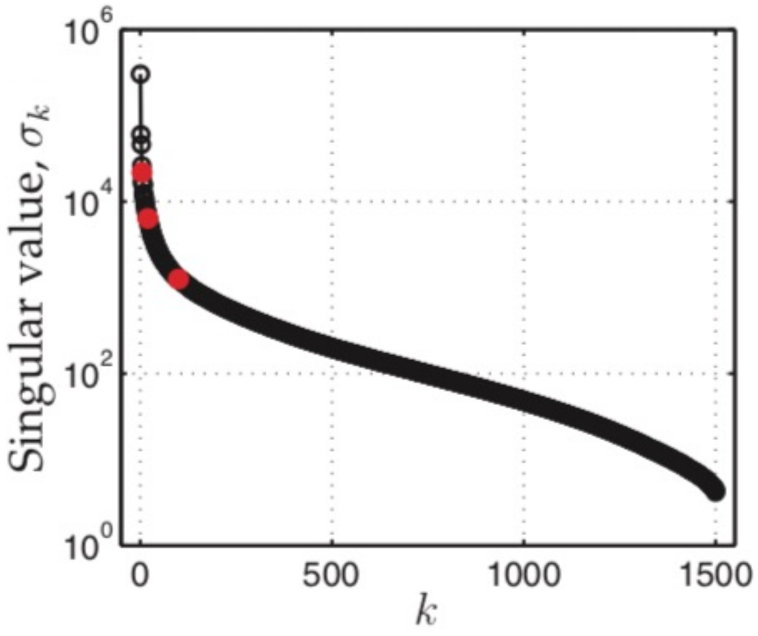

Schematic of truncated SVD

# Example: image compression

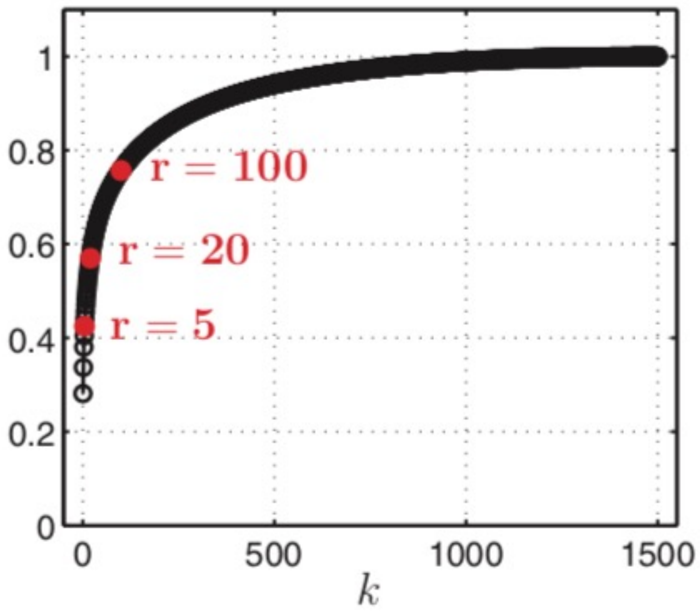
A digital image can be considered as a matrix

X = 

An image of resolution 2000 x 1500
The matrix X is 2000 x 1500




Singular values

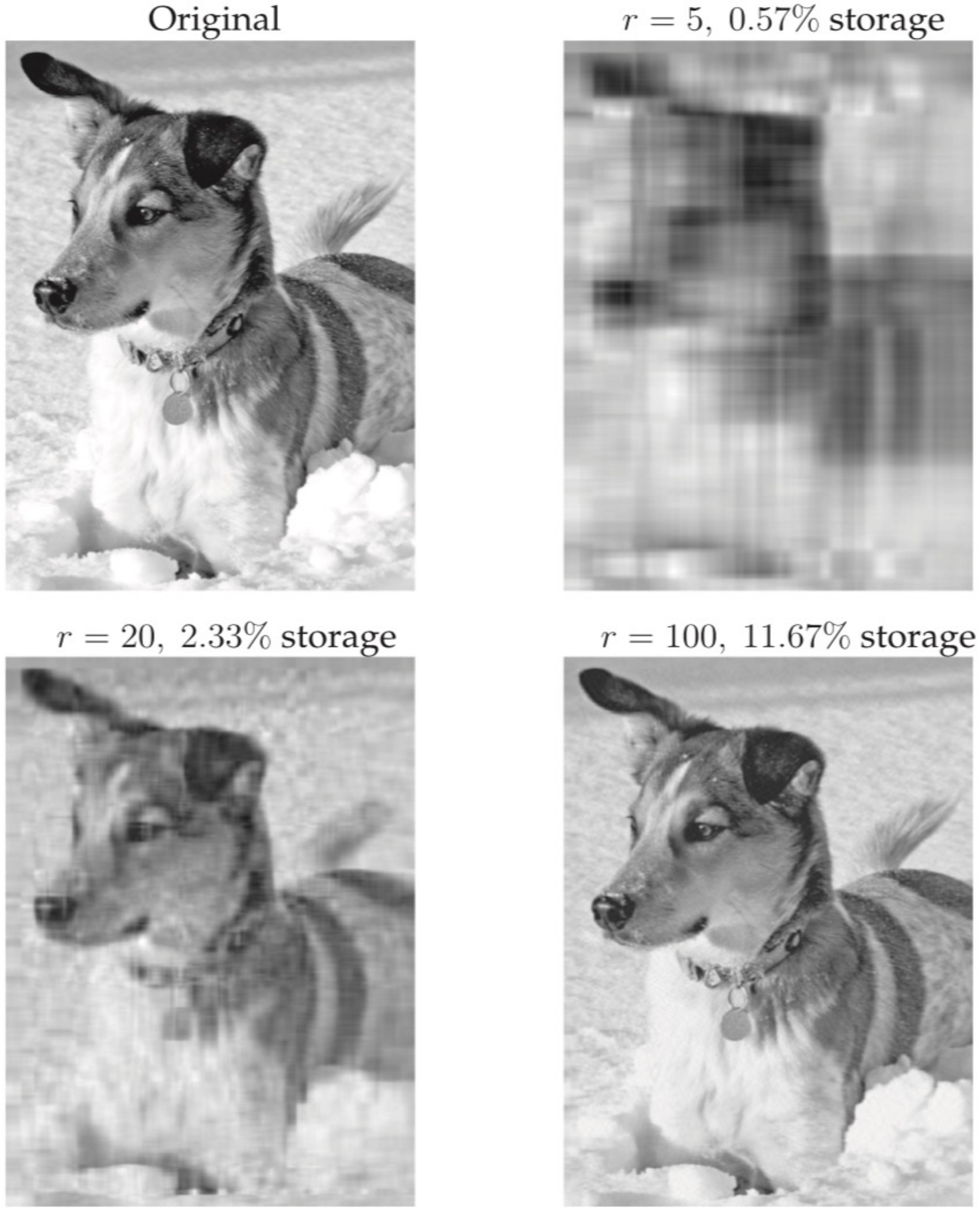



Cumulative energy

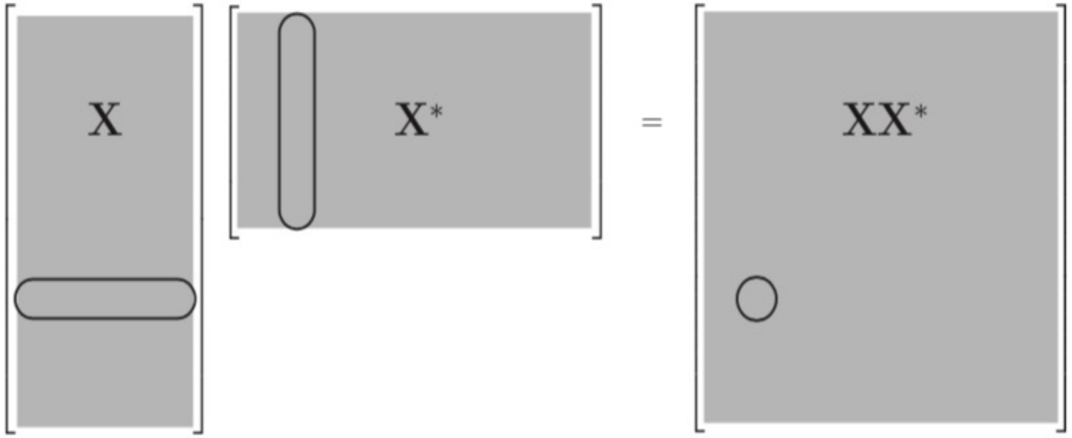
it equals to the sum of sigma_i ^2
divided by the sum of sigma_n^2

Results using various values of r

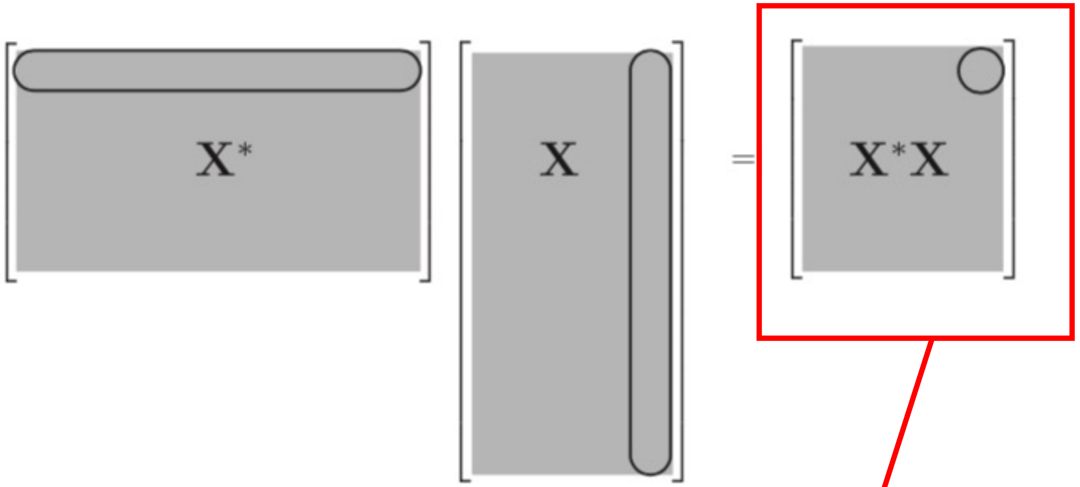
- Good result when r=100
- An example of compressibility



# SVD and $\boxed{\text{correlation matrix}}$




Taking inner products of rows




Taking inner products of columns

Compute this, because it is often cheaper (will discuss it in p.12)

Correlation matrix: This XX^* and X^*X provides an intuitive interpretation of the SVD, where the columns of U are eigenvectors of the coorelation matrix XX^*, and columns of V are eigenvectors of X^*X.

Thus the columns of U are hierarchielly ordered by how much correlation they capture in the columns of X.

- Using the definition of SVD

Correlation matrix

$$\mathbf{X}\mathbf{X}^* = \mathbf{U}\begin{bmatrix}\hat{\boldsymbol{\Sigma}} \\ \mathbf{0}\end{bmatrix}\mathbf{V}^*\mathbf{V}\begin{bmatrix}\hat{\boldsymbol{\Sigma}} & \mathbf{0}\end{bmatrix}\mathbf{U}^* = \mathbf{U}\begin{bmatrix}\hat{\boldsymbol{\Sigma}}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0}\end{bmatrix}\mathbf{U}^*$$

$$\boxed{\mathbf{X}^*\mathbf{X} = \mathbf{V}\begin{bmatrix}\hat{\boldsymbol{\Sigma}} & \mathbf{0}\end{bmatrix}\mathbf{U}^*\mathbf{U}\begin{bmatrix}\hat{\boldsymbol{\Sigma}} \\ \mathbf{0}\end{bmatrix}\mathbf{V}^* = \mathbf{V}\hat{\boldsymbol{\Sigma}}^2\mathbf{V}^*.}$$

Stick with this option, Sigma is simpler

- Since U and V are unitary matrices

$$\mathbf{X}\mathbf{X}^*\mathbf{U} = \mathbf{U}\begin{bmatrix}\hat{\boldsymbol{\Sigma}}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0}\end{bmatrix},$$

why we need to put it into this way?

$$\mathbf{X}^*\mathbf{X}\mathbf{V} = \mathbf{V}\hat{\boldsymbol{\Sigma}}^2.$$

So we use columns of V to produce our principal axis

- Columns of U are eigenvectors of XX*, and columns of V are eigenvectors of X*X Try to demonstrate why this statement is true
- Note that we order the singular values in descending order
- Columns of U are hierarchically ordered by the amount of correlations captured in the columns of X

Step 1: Use X^*XV = V \hat \Lambda^2  (solve Ax \ \lambda x) to solve V and \Lambda

# Method of snapshots

This is especially true when n>>m

- Computing U is expensive as the size of XX* is large
- Computing V is relatively cheap as the $\boxed{\text{size of X*X is small}}$
- We can compute the columns of U corresponding to nonzero singular values as follows

$$\tilde{\mathbf{U}} = \mathbf{X}\tilde{\mathbf{V}}\tilde{\boldsymbol{\Sigma}}^{-1}$$

(m x m)

(Pesudo-inverse implicitly compute SVD once on the matrix A)

# Pseudo-inverse

- Many physical systems may be represented as a linear system $\mathbf{Ax} = \mathbf{b}$
- In the overdetermined case with no solution, we will find the least-squares solution x that minimizes
- In the underdetermined case with infinitely many solutions, we will find the minimum norm solution x that minimizes
- We approximate the inverse of A by the inverse of the truncated SVD

Pseudo-inverse matrix of A

$$\boxed{\mathbf{A}^{\dagger}} \triangleq \tilde{\mathbf{V}}\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\mathbf{U}}^{*} \implies \mathbf{A}^{\dagger}\mathbf{A} = \mathbf{I}_{m \times m}$$

- The above is called the left pseudo-inverse of A
- Applying to Ax=b,

$$\mathbf{A}^{\dagger}\mathbf{A}\tilde{\mathbf{x}} = \mathbf{A}^{\dagger}\mathbf{b} \implies \tilde{\mathbf{x}} = \boxed{\tilde{\mathbf{V}}\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\mathbf{U}}^{*}}\mathbf{b}$$

Our line

# Example: simple data fitting

- Given a set of data points $(a_i\ , b_i)$, fit a straight line centered at the origin with slope $x$
- This results in the following problem

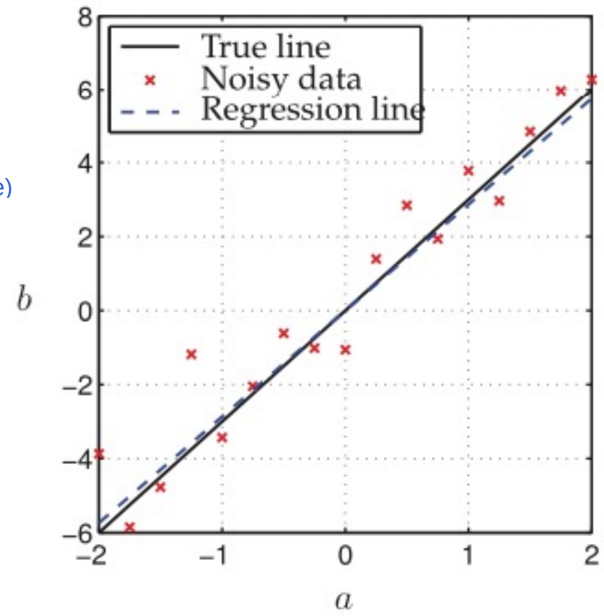
$$\begin{bmatrix} | \\ \mathbf{b} \\ | \end{bmatrix} = \begin{bmatrix} | \\ \mathbf{a} \\ | \end{bmatrix} x = \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^* x.$$

$$\Longrightarrow \quad x = \tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{U}}^*\mathbf{b}.$$

(that is just applying the formulas in the previous page)

- We have $\boxed{\tilde{\mathbf{\Sigma}} = \|\mathbf{a}\|_2,\ \tilde{\mathbf{V}} = 1, \text{ and } \tilde{\mathbf{U}} = \mathbf{a}/\|\mathbf{a}\|_2}$, so we obtain

$$x = \frac{\mathbf{a}^*\mathbf{b}}{\|\mathbf{a}\|_2^2}$$



# Principal Component Analysis (PCA)

- Provides a data-driven, hierarchical coordinate system to represent high-dimensional correlated data
- A number of measurements are collected, each measurement is a row of the large matrix X (where X is n x m)
- We compute the row-wise mean given by

$$\bar{\mathbf{x}}_j = \frac{1}{n}\sum_{i=1}^{n} \mathbf{X}_{ij}$$

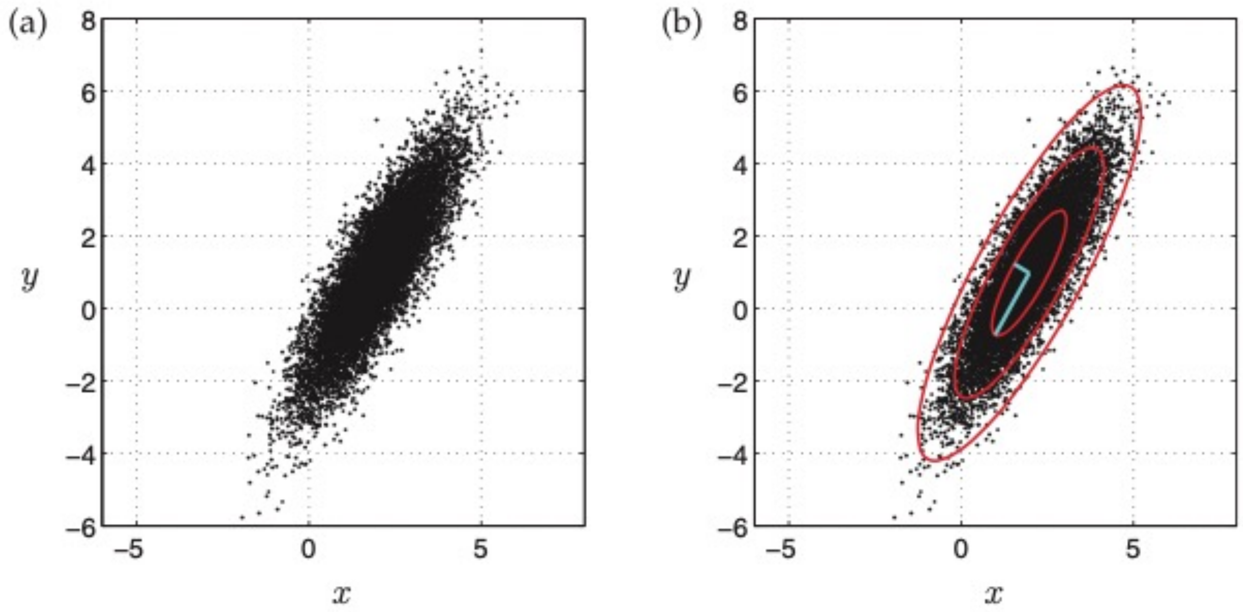
- And construct the mean matrix given by $\bar{\mathbf{X}} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \bar{\mathbf{x}}$

- The principal components are the singular vectors V of $\mathbf{B} = \mathbf{X} - \bar{\mathbf{X}}$ (or eigenvector of B*B)

# An illustration

- A set of $n$ data points in the $m = 2$ dimensional space
- The mean is $(2,1)$
- The $\boxed{\text{PCA modes}}$ are obtained by the eigenvectors of the $2{\times}2$ matrix $B^*B$

PCA modes = the principal components / eigenvectors of the covariance matrix of the data.



# Example: eigenfaces

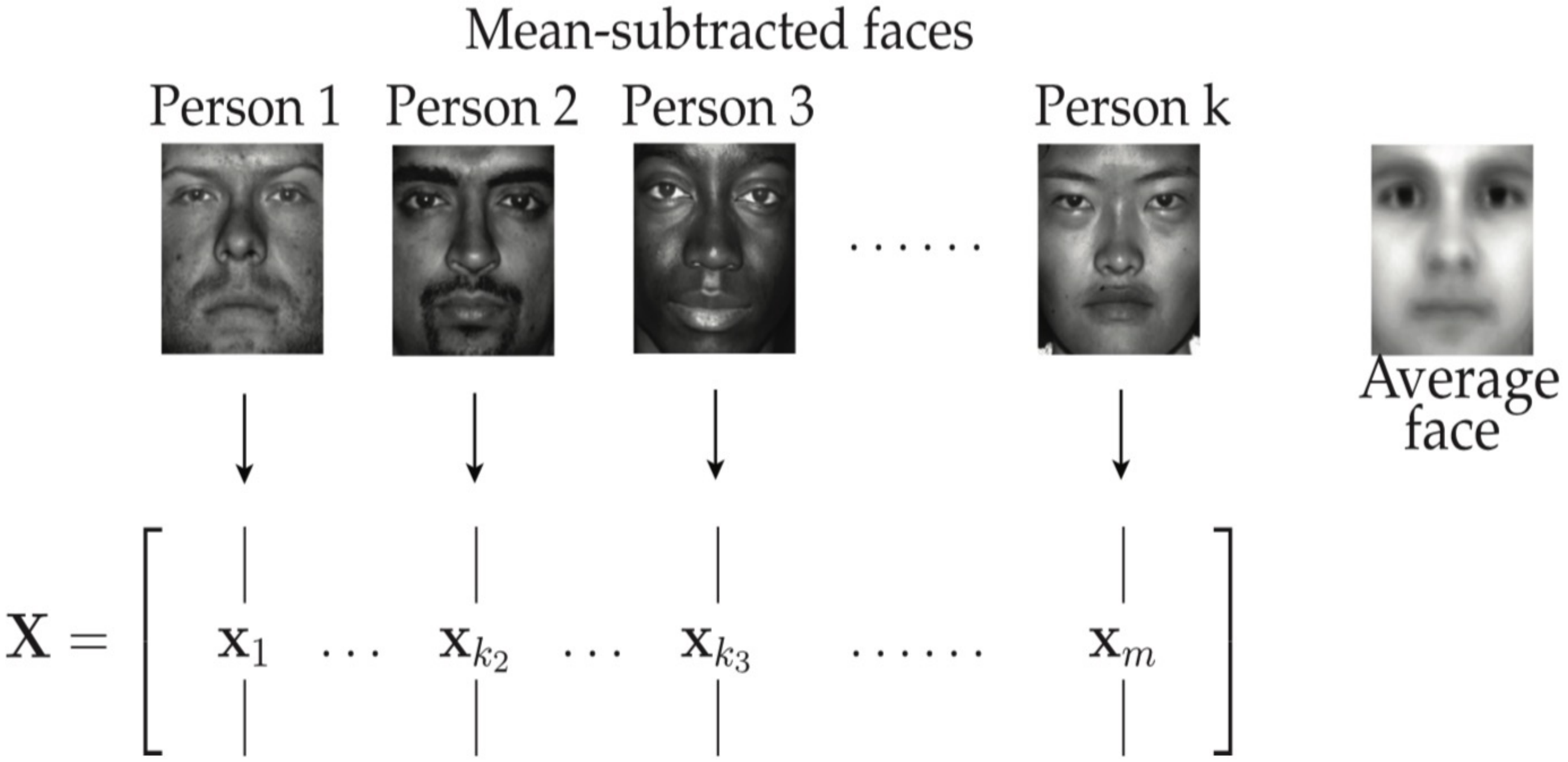
- Aim: use a large library of facial images to extract the most dominant correlation between images
- The result is a set of eigenfaces, which is a new coordinate system to represent the images
- The library contains images of 38 individuals, each of them has 64 images with various poses and lighting conditions
- The images of these 36 individuals will be used to construct the dominant correlations (PCA modes)
- The images of the other 2 individuals will be used to test the PCA modes

A single image for each individual

All images of a specific individual

Each image is 192 pixels tall and 168 pixels wide

Each image is arranged as a column vector (32256 x 1), and is subtracted by the column mean




The matrix X has totally 2304 columns

Perform the SVD

$$\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^* \approx \tilde{\mathbf{U}}\tilde{\boldsymbol{\Sigma}}\tilde{\mathbf{V}}^*$$

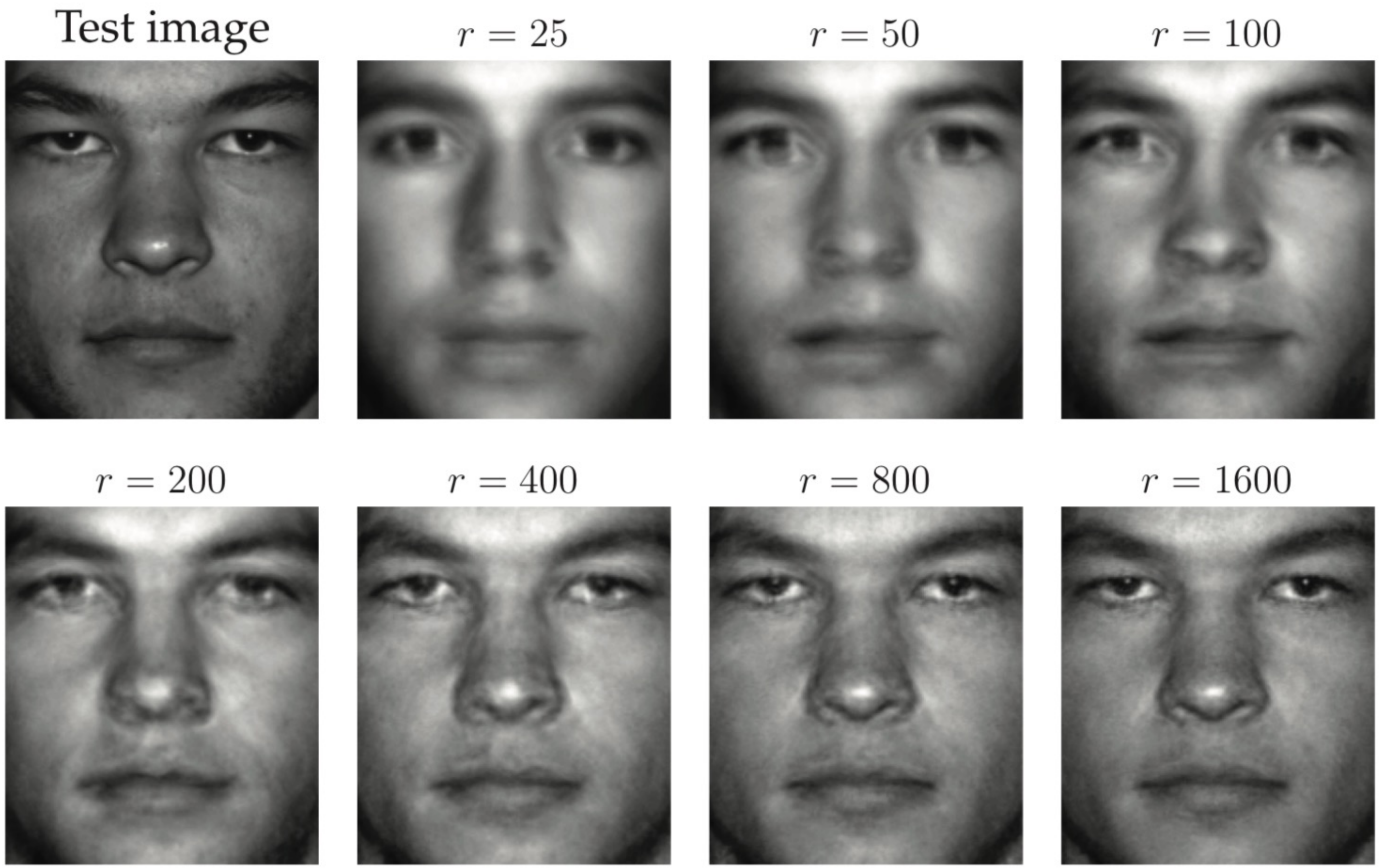
$$\tilde{\mathbf{U}} = \begin{bmatrix} | & | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 & \cdots & \mathbf{u}_r \\ | & | & | & & | \end{bmatrix}$$

Eigenfaces

The first r columns of U are the first r PCA modes

Take a test image and represent it using the first r PCA modes:

$$\tilde{\mathbf{x}}_{\text{test}} = \tilde{\mathbf{U}}\tilde{\mathbf{U}}^{*}\mathbf{x}_{\text{test}}$$

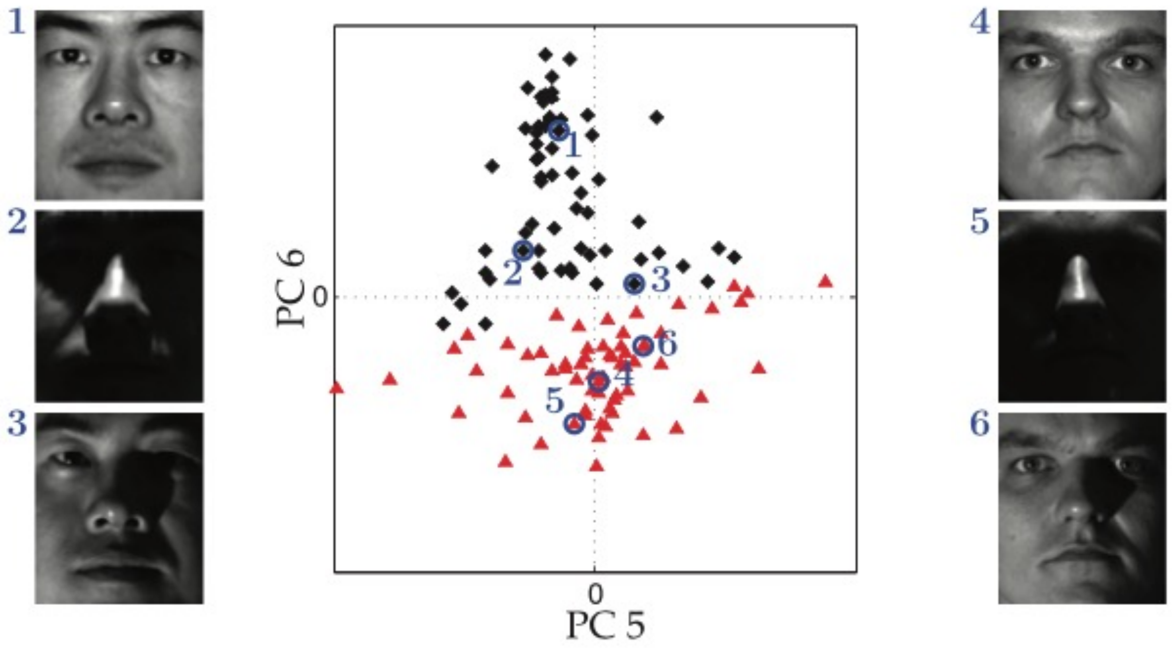



Note that this test image does not belong to the set of images for the PCA construction

We see that the PCA modes can be used to represent images efficiently

# Classifying images

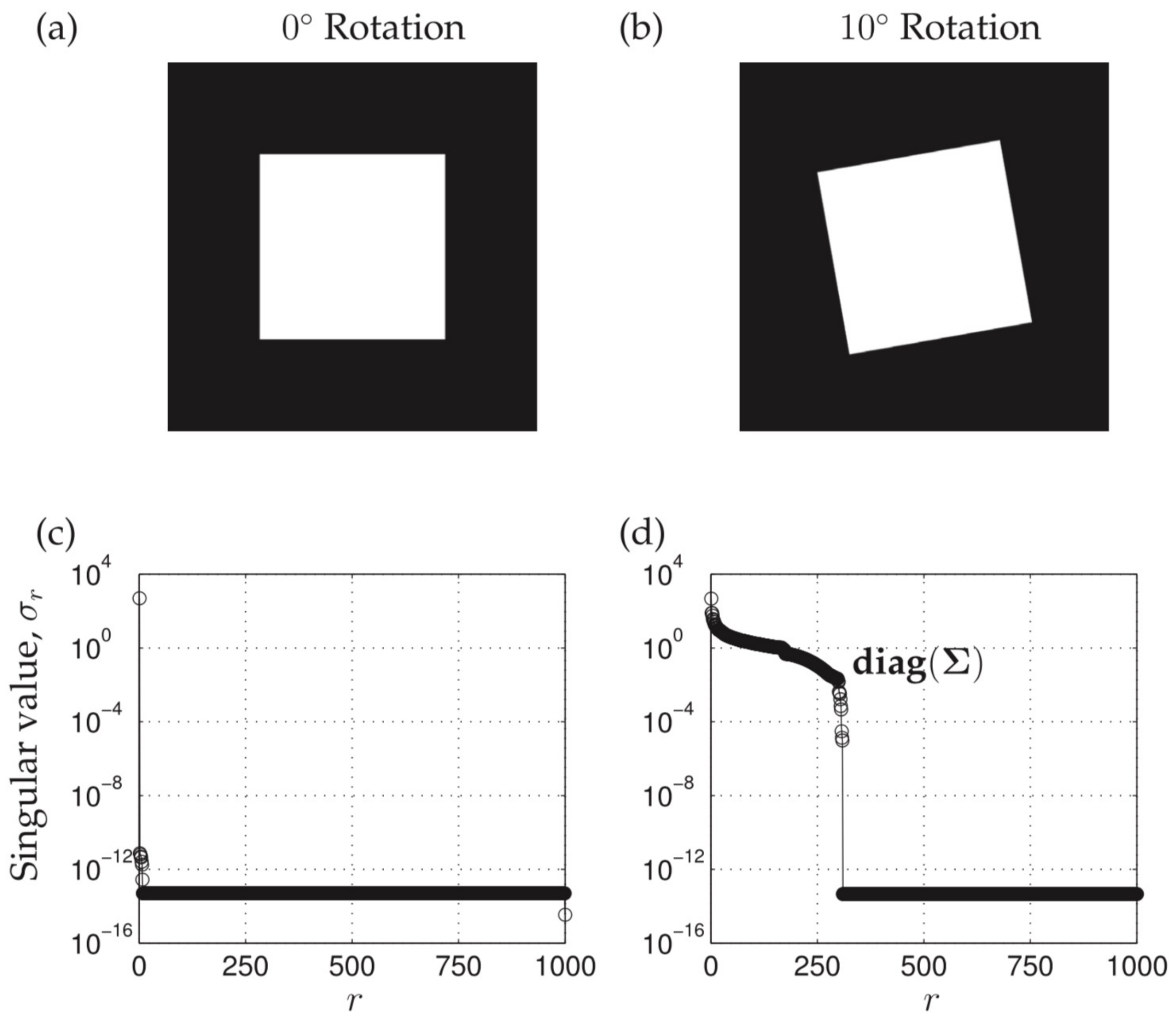
- Some PCA modes may capture the most common features
- Other PCA modes may be useful for distinguishing between images
- The following shows the 64 images of 2 individuals, using the 5th and the 6th PCA modes




The 6th PCA modes can distinguish these two individuals

# Importance of data alignment

A 1000 x 1000 square matrix
X with
white = 1
black = 0




A small modification of the matrix on the left

Very different behavior on the singular values

- A pitfall of the SVD/PCA is data misalignment
- It depends on the coordinate system in which the data is represented

- On the contrary, SVD is invariant under unitary transformations (inner product preserving)

- One should use SVD/PCA carefully given the above points

# Randomized SVD (rSVD)

- A more efficient algorithm for matrix decomposition focusing on extracting dominant low-rank structure in the matrix

Step 1: Construct a random projection matrix $\mathbf{P} \in \mathbb{R}^{m \times r}$ to sample the column space of $\mathbf{X} \in \mathbb{R}^{n \times m}$

- r is the target rank
- Likely that P will project out important components of X
- Z approximates the column space of X with high probability

We assume X is low rank.

There is a structure in the dataset

Many of columns are dependent, so there are certain structure in the dataset.

$$\mathbf{Z} = \mathbf{X}\mathbf{P}$$

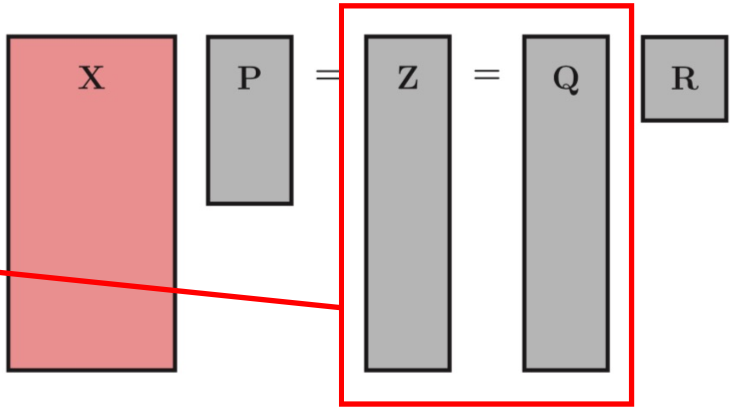
lead to a big reduction in column space

span the same column space of X

Then we perform QR decomposition of Z to obtain an orthonormal basis for X

why it is useful to obtain the orthonormal basis for X?

$$\mathbf{Z} = \mathbf{Q}\mathbf{R}$$



Step 2: Project X into a smaller space, and obtain a matrix Y

$$\mathbf{Y} = \boxed{\mathbf{Q}^*}\mathbf{X}$$

what the heck is this?

(Q: why it is a projection?)

Then we have

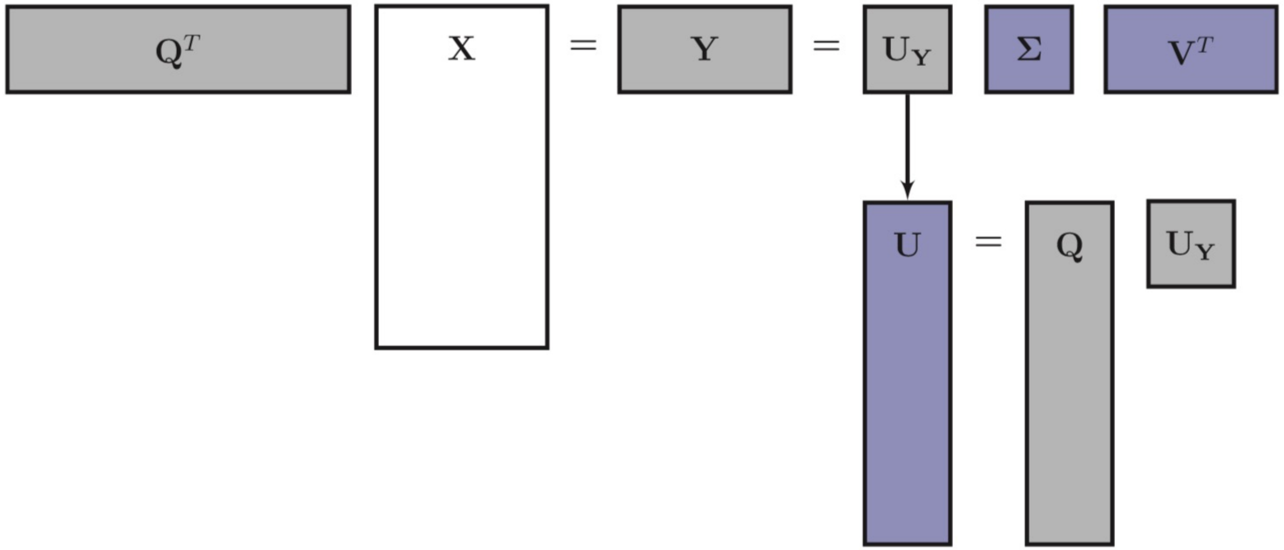
$$\mathbf{X} \approx \mathbf{QY}$$

(better agreement when the singular values decay fast)

Perform SVD on the smaller matrix Y

$$\boxed{\mathbf{Y} = \mathbf{U_Y \Sigma V^*}}$$

Note that $\mathbf{\Sigma}$ and $\mathbf{V}$ are the same for Y and X




Step 3: Reconstruct the left singular vectors by

$$\mathbf{U} = \mathbf{QU_Y}$$

- Oversampling
  - the matrix may not be of exactly rank r
  - increase the number of columns in the random matrix P from r to r + p
  - p = 5 or 10 works well
- Power iterations
  - the matrix may have slowly decay singular values
  - preprocess X by the power iterations

$$\boxed{\mathbf{X}^{(q)} = (\mathbf{X}\mathbf{X}^*)^q \mathbf{X}}$$

  - the singular values decays more rapidly

$$\boxed{\mathbf{X}^{(q)} = \mathbf{U}\mathbf{\Sigma}^{2q-1}\mathbf{V}^*}$$

- Error bound

L2-norm

$$\mathbb{E}\left(\|\mathbf{X} - \mathbf{Q}\mathbf{Y}\|_2\right) \leq \left(1 + \sqrt{\frac{r}{p-1}} + \frac{e\sqrt{r+p}}{p}\sqrt{m-r}\right)^{\frac{1}{2q+1}} \sigma_{k+1}(\mathbf{X})$$

Error would be small, if k is a good prediction of r