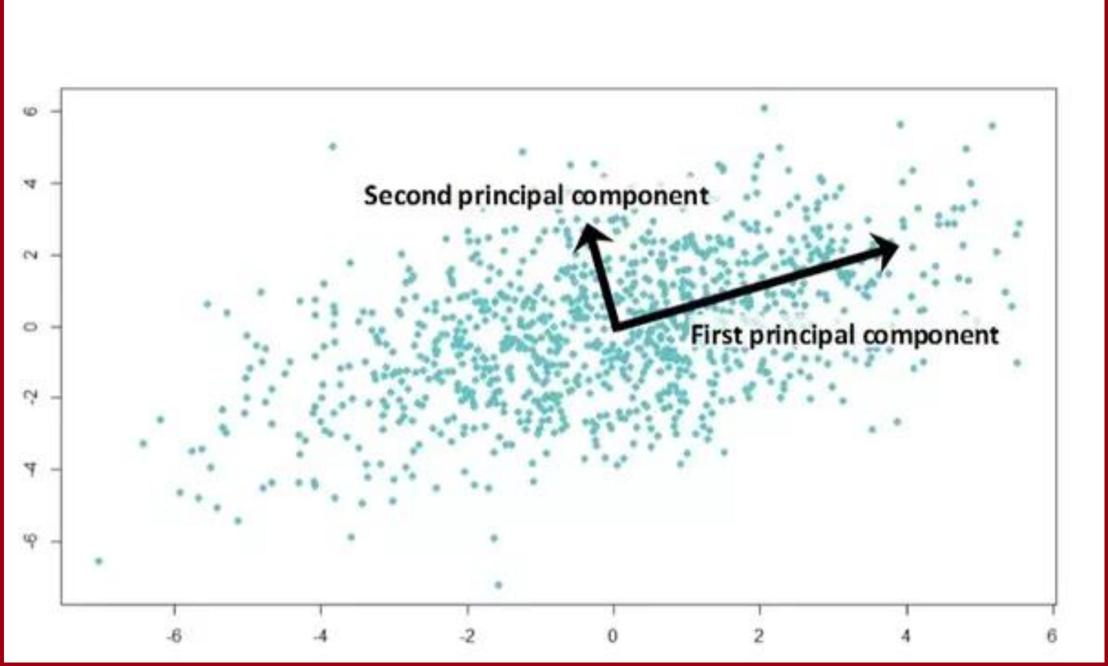
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

UNIVERSITÀ DEGLI STUDI DI PADOVA



Principal Component Analysis

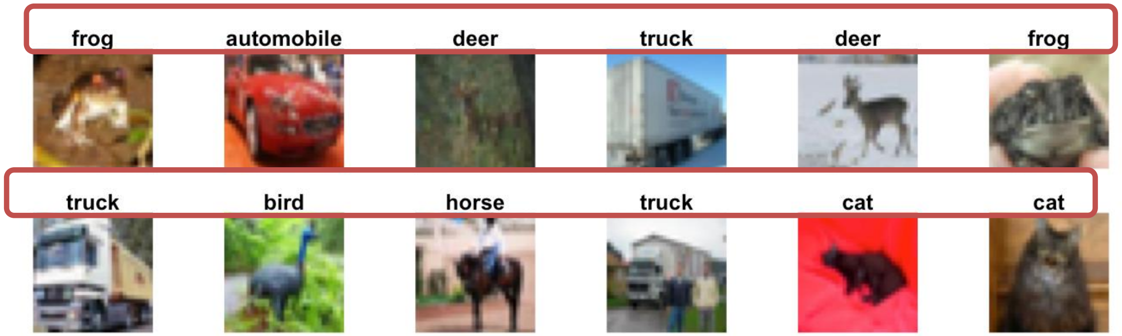
Machine Learning 2023-24

UML Book Chapter 23

Slides P. Zanuttigh (derived from F. Vandin slides)

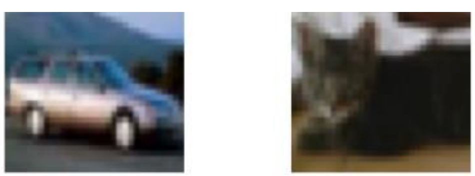
DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

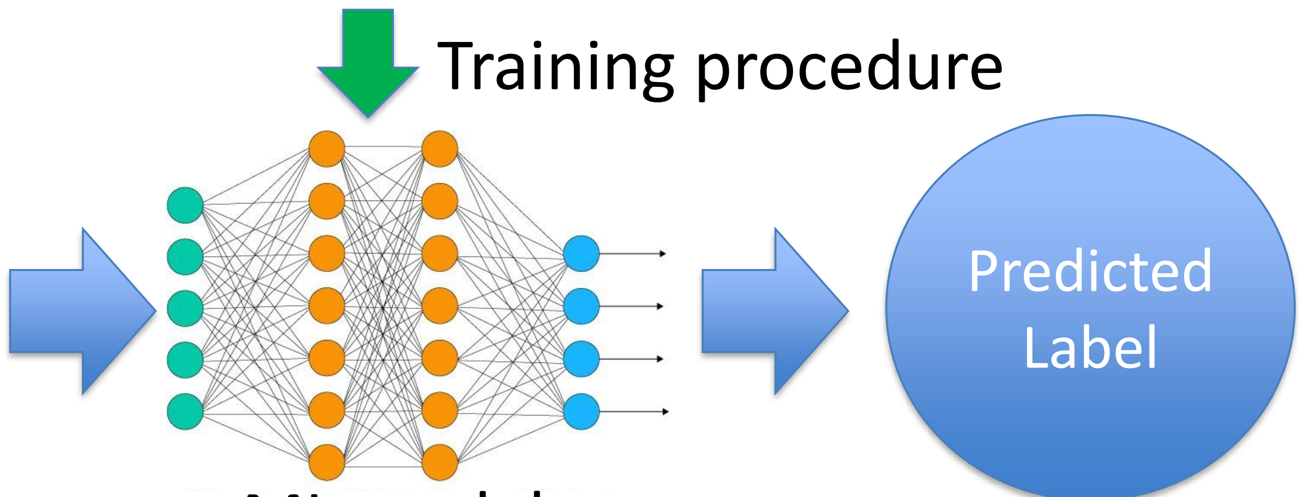
Recall: Supervised Learning



Training data
with labels

Training procedure



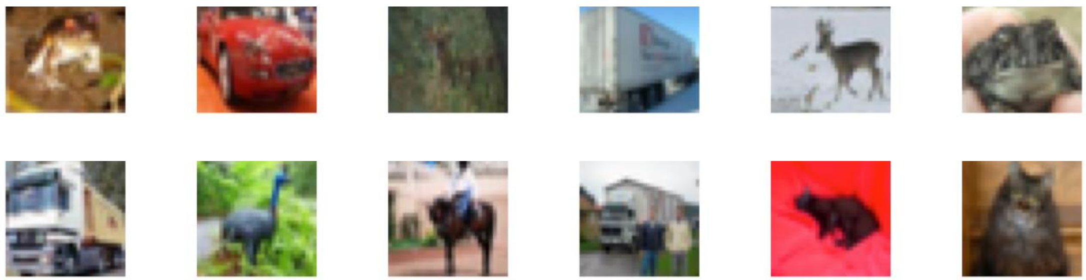


Data to be analyzed

ML model
(training: estimate parameters)

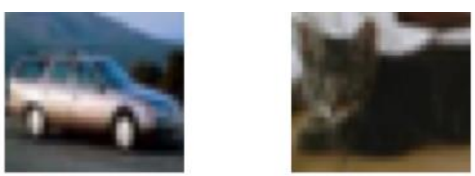
DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

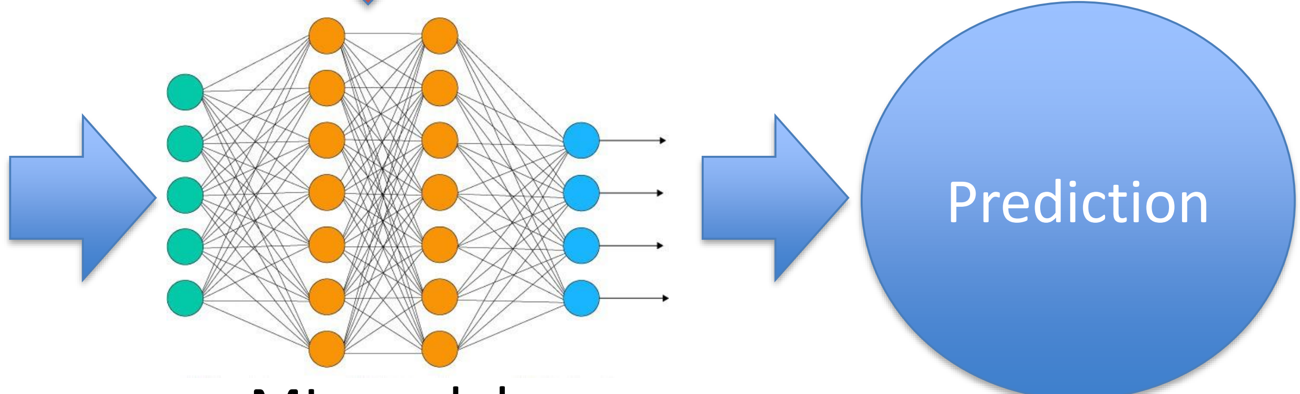
Unsupervised Learning: Training Data is Unlabeled



Training data (unlabeled)

Training procedure





Data to be analyzed

ML model
(training: estimate parameters)

Unsupervised Learning:
Training data is not labeled

Unsupervised Learning Techniques

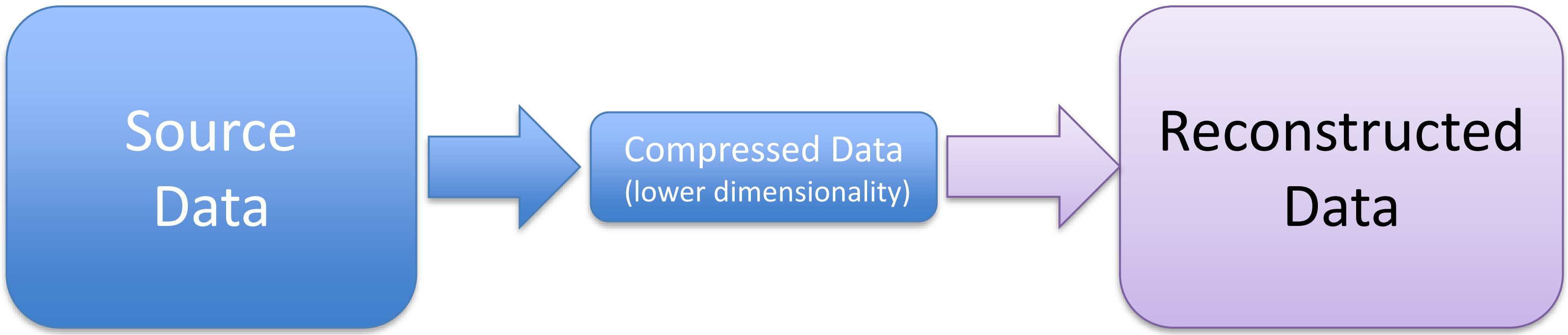
We are going to see only a couple of unsupervised learning tasks and a few very simple and commonly used methods

- *Clustering (already seen)*
  - *K-means*
  - *Linkage-based clustering*
- *Dimensionality reduction*
  - *Principal Component Analysis (PCA)*

There are many other techniques (*not part of this course*)

- Mean shift clustering, spectral clustering….
- Compressive sensing

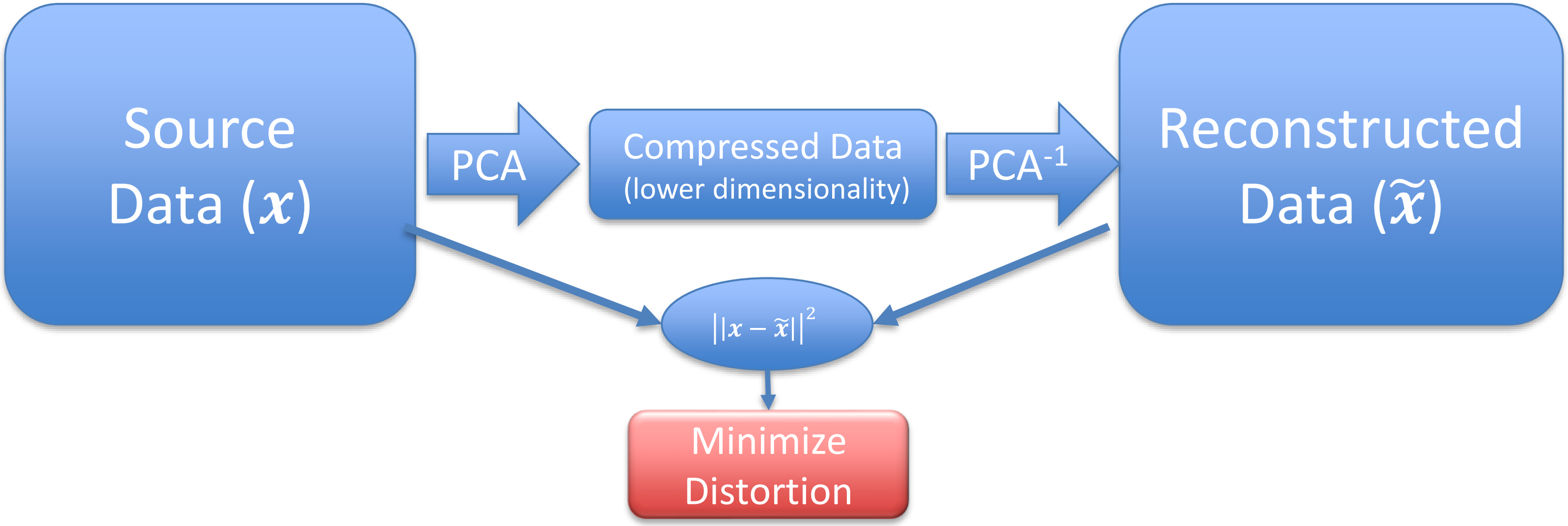
Dimensionality Reduction



- Take data from a highly dimensional space and project to a lower dimensional one
- Many applications:
  - Reduce number of features (learn with less samples, lower computation req.)
  - Capture most important aspects of the data for subsequent analysis
  - Data visualization
  - etc..
- By reducing dimensionality part of the information get lost
  - Lower dimensional data should be a *good approximation* of the higher dimensional representations
  - *Good approximation: lower dimensional data should allow for reconstructing the original data with a reasonable accuracy*

Dimensionality Reduction

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE



- Lower dimensional data should be a *good approximation* of the higher dimensional representations
- *Good approximation*: minimize error obtained by reprojecting the data back to the high dimensional space (→ similar to lossy data compression)

- Focus on linear mapping of the data (represented by a matrix multiplication)
- *Principal Component Analysis (PCA)*: find the linear mapping that minimizes the mean squared error in the reprojection

Principal Component Analysis (PCA)

- ❑ $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_m \in \mathbb{R}^d$: data points
- ❑ $W \in \mathbb{R}^{n,d}$ $(n < d)$: mapping $\boldsymbol{x} \rightarrow \boldsymbol{y} = W\boldsymbol{x}$
  - ➢ where $\boldsymbol{y} = W\boldsymbol{x} \in \mathbb{R}^n$ is a lower dimensional representation of $\boldsymbol{x} \in \mathbb{R}^d$
- ❑ $U \in \mathbb{R}^{d,n}$ $(n < d)$: inverse mapping $\boldsymbol{y} \rightarrow \tilde{\boldsymbol{x}} = U\boldsymbol{y}$
  - ➢ used to recover an approximation $\tilde{\boldsymbol{x}} = \boldsymbol{UW}\boldsymbol{x}$ of $\boldsymbol{x}$
- ❑ Target: find the lower dimensional representation that better approximates the data → that leads to minimum squared distance between $\tilde{\boldsymbol{x}}$ and $\boldsymbol{x}$
  - ➢ Corresponds to seek for the *n*-dimensional basis that best captures the variance in the d-dimensional data

$$\underset{W \in \mathbb{R}^{n,d}, U \in \mathbb{R}^{d,n}}{\operatorname{argmin}} \sum_{i=1}^{m} \| \boldsymbol{x}_i - \underbrace{UW\boldsymbol{x}_i}_{\tilde{\boldsymbol{x}}_i} \|_2^2$$

Lemma

There exist an optimal solution (U^*, W^*) of $\underset{W \in \mathbb{R}^{n,d}, U \in \mathbb{R}^{d,n}}{\operatorname{argmin}} \sum_{i=1}^{m} \| \boldsymbol{x}_i - UW\boldsymbol{x}_i \|_2^2$ where:

- the columns of U^* are orthonormal (i.e., $(U^*)^T U^* = I$)
- $W^* = (U^*)^T$

Recall: orthonormal

- $\mathbf{u}_i^T \mathbf{u}_j = 0, \forall i \neq j$
- $\|\mathbf{u}_i\| = 1 = \mathbf{u}_i^T \mathbf{u}_i, \forall i$

Demonstration:

1. Fix U,W and consider the mapping $\mathbf{x} \to UW\boldsymbol{x}$
   - The range of the mapping is $R = \{UW\boldsymbol{x} \ ; \boldsymbol{x} \in \mathbb{R}^d\}$
2. $V \in \mathbb{R}^{d,n}$: matrix whose column form an orthonormal basis of R
   - Recall that $V^T V = I$ and $\forall \boldsymbol{x} \in R: V\boldsymbol{y} \ with \ y \in \mathbb{R}^n$
3. $\forall \boldsymbol{x} \in \mathbb{R}^d, \forall \boldsymbol{y} \in \mathbb{R}^n$:
   - $\|\boldsymbol{x} - V\boldsymbol{y}\|_2^2 = \|\boldsymbol{x}\|^2 + \boldsymbol{y}^T V^T V \boldsymbol{y} - 2\boldsymbol{y}^T V^T \boldsymbol{x} = \|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2 - 2\boldsymbol{y}^T (V^T \boldsymbol{x})$
4. Minimize $\|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2 - 2\boldsymbol{y}^T (V^T \boldsymbol{x})$ w.r.t y: set $\nabla = 0 \to 2\boldsymbol{y} - 2(V^T \boldsymbol{x}) = 0 \to \boldsymbol{y_{opt}} = V^T \boldsymbol{x}$
5. $\forall \boldsymbol{x}$: $\underset{\tilde{\boldsymbol{x}} \in R}{\operatorname{argmin}} \|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|_2^2 = V\boldsymbol{y_{opt}} = V(V^T \boldsymbol{x})$: it is the best approximation in subspace R
6. $\forall \boldsymbol{x}$: includes also $\boldsymbol{x_1}, .., \boldsymbol{x_m}$ (data vectors): $\sum_{i=1}^{m} \|\boldsymbol{x}_i - UW\boldsymbol{x}_i\|^2 \geq \sum_{i=1}^{m} \|\boldsymbol{x}_i - VV^T \boldsymbol{x}_i\|^2$, so we can replace U,W with VV^T without increasing the objective
7. Holds for $\forall U, W$: there exist a solution that minimize $\sum_{i=1}^{m} \|\boldsymbol{x_i} - UW\boldsymbol{x}_i\|^2$ with V orthonormal columns and $W = U^T$

Optimization Problem

There exist an optimal solution (U^*, W^*) of $\underset{W\in\mathbb{R}^{n,d}, U\in\mathbb{R}^{d,n}}{\operatorname{argmin}} \sum_{i=1}^{m} \|\boldsymbol{x}_i - UW\boldsymbol{x}_i\|_2^2$ where:

- the columns of U^* are orthonormal (i.e., $(U^*)^T U^* = I$)
- $W^* = (U^*)^T$

- Trace: Σ elements on diagonal
- It is a scalar
- $trace(A^T B) = trace(AB^T) =$ $= trace(B^T A) = trace(BA^T)$

The optimization problem can be rewritten as:

$$\underset{U\in\mathbb{R}^{d,n}: U^T U = I}{\operatorname{argmin}} \sum_{i=1}^{m} \|\boldsymbol{x}_i - UU^T\boldsymbol{x}_i\|_2^2$$

With some manipulations: $\|\boldsymbol{x} - UU^T\boldsymbol{x}\|_2^2 = \|\boldsymbol{x}\|^2 - 2x^T UU^T x + x^T UU^T UU^T x =$

$$= \|\boldsymbol{x}\|^2 - \boldsymbol{x}^T UU^T \boldsymbol{x} = \|\boldsymbol{x}\|^2 - trace\left(\boldsymbol{x}^T UU^T \boldsymbol{x}\right) = \|\boldsymbol{x}\|^2 - trace\left(U^T \boldsymbol{x}\boldsymbol{x}^T U\right)$$

$$\underset{U\in\mathbb{R}^{d,n}: U^T U = I}{\operatorname{argmax}} \; trace\left(U^T \sum_{i=1}^{m} \boldsymbol{x}_i \boldsymbol{x}_i^T \, U\right) = \underset{U\in\mathbb{R}^{d,n}: U^T U = I}{\operatorname{argmax}} \; trace(U^T A U)$$

Notice: $A = \sum_{i=1}^{m} \boldsymbol{x}_i \boldsymbol{x}_i^T$ is symmetric and positive semidefinite. It can be rewritten as $A = VDV^T$ where D is diagonal (with eigenvalues $D_{d,d} \geq 0$) and $V^T V = VV^T = I$ (the columns of V are the eigenvectors of A)

Theorem (PCA)

Let $\boldsymbol{x}_1, \boldsymbol{x}_2, .., \boldsymbol{x}_m$ be arbitrary vectors in $\mathbb{R}^d$

let $A = \sum_{i=1}^{m} \boldsymbol{x}_i \boldsymbol{x}_i^T$

let $\boldsymbol{u}_1, .., \boldsymbol{u}_n$ be *n* eigenvectors of A corresponding to the largest *n* eigenvalues of A

Then a solution of the PCA optimization $\underset{W \in \mathbb{R}^{n,d}, U \in \mathbb{R}^{d,n}}{\text{argmin}} \sum_{i=1}^{m} \|\boldsymbol{x}_i - UW\boldsymbol{x}_i\|_2^2 = \underset{U \in \mathbb{R}^{d,n}: U^T U = I}{\text{argmax}} \; trace(U^T A U)$

is to set U to be the matrix whose columns are $\boldsymbol{u}_1, .., \boldsymbol{u}_n$ and to set $W = U^T$

Notes:

- *Recall:* Decompose A as VDV^T (SVD decomposition, D diag. and $V^T V = V V^T = I$)
- It is a common practice to *"center"* the examples before applying PCA (i.e., subtract the mean)
- Computation time is $O(d^3) + O(md^2)$ (the first term for calculating eigenvalues and the second for constructing A)
- Trick for faster solution in case *d>>m* *(not part of the course)*

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Pseudocode

Input

$X \in \mathbb{R}^{m,d}$: matrix that contains m samples, one for each row

n: number of components

Algorithm

Compute $A = X^T X$

Perform eigenvalue decomposition of A

Let $\boldsymbol{u_1}, \ldots, \boldsymbol{u_n}$ be the eigenvectors of A corresponding to largest eigenvalues

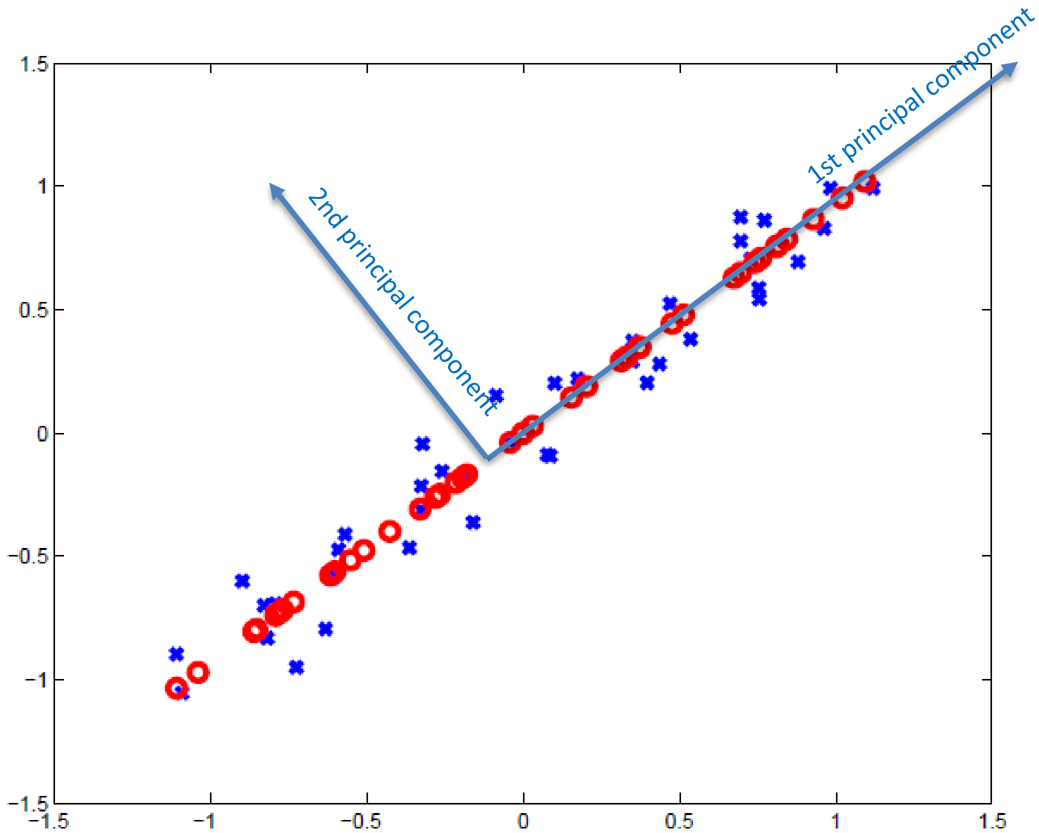
Output: $\boldsymbol{u_1}, \ldots, \boldsymbol{u_n}$

On the book: trick for *d>>m*
(not part of the course)

Eigenvectors of A:

1. First principal component (*p.c.*) = direction with largest projected variance
2. Second *p.c.* = orthogonal direction with largest projected variance
   - i.e., largest remaining variance after removing the first p.c.
3. ... iterate for all the other components (3…n) ….

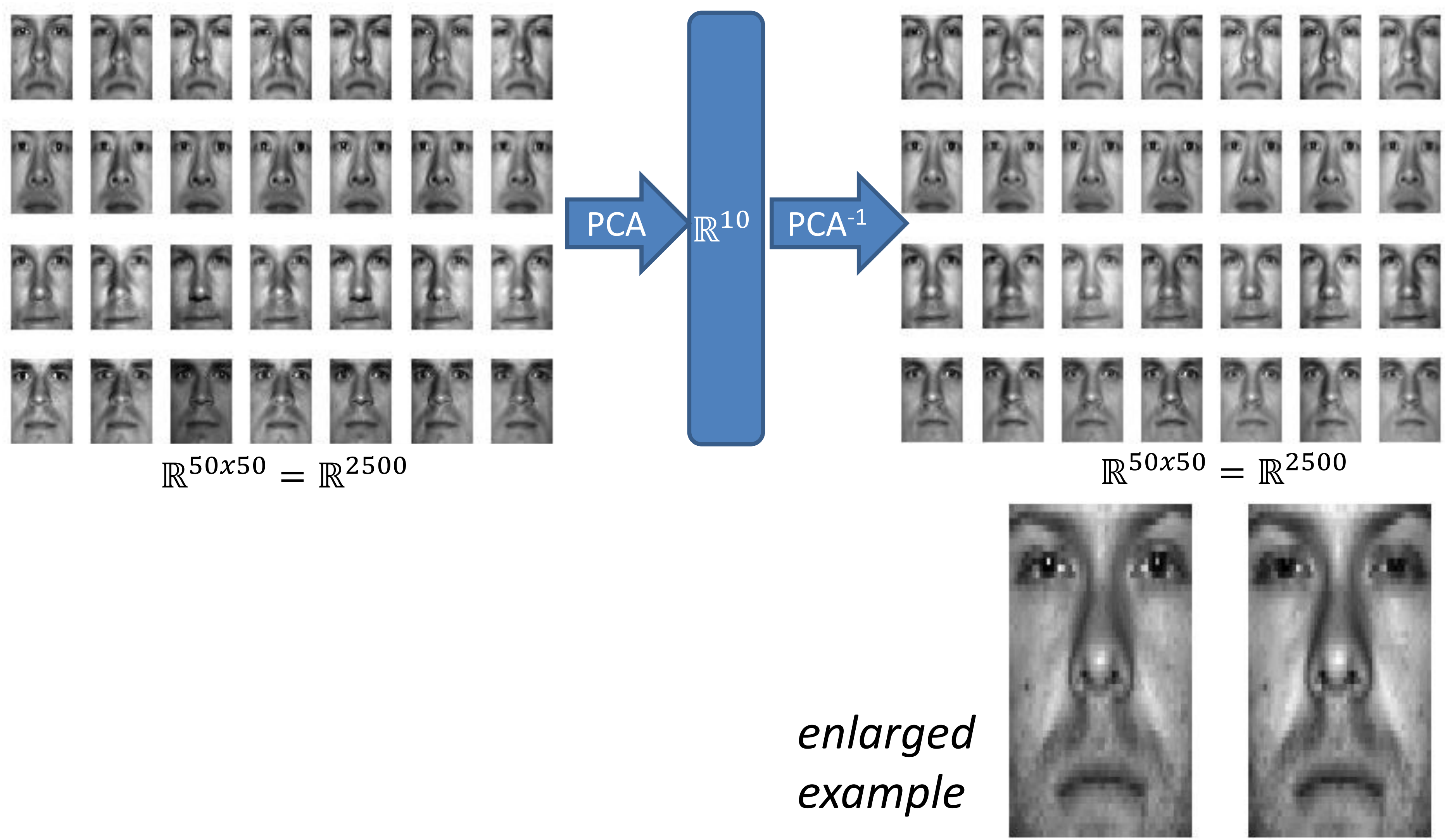
Example: From 2D to 1D



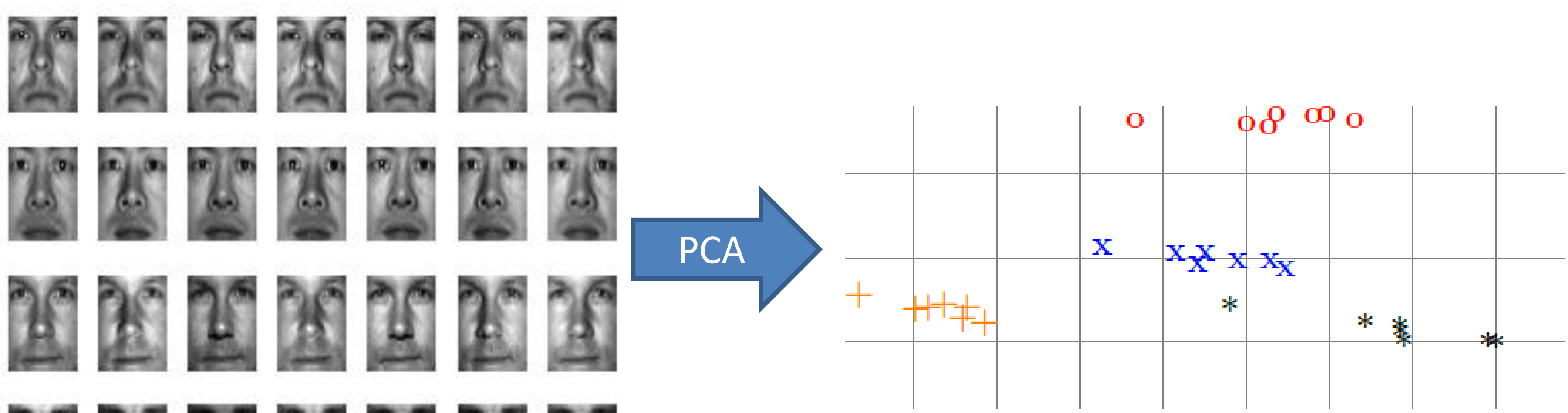
Set of 2D vectors (blue) and their reconstruction (red) after dimensionality reduction to 1D with PCA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Example: Face Compression



Example: Face Recognition



$\mathbb{R}^{50x50} = \mathbb{R}^{2500}$

$\mathbb{R}^2$

- Faces with the same type of mark (+, x, *, o) belong to the same individual
- PCA can be used for face recognition ! (*eigenfaces* algorithm)