PCA

Machine Learning

Hamid R Rabiee – Zahra Dehghanian

Spring 2025

Sharif University
of Technology

Dimensionality Reduction: Feature Selection vs. Feature Extraction

- Feature **selection**
  - Select a subset of a given feature set

- Feature **extraction**
  - A linear or non-linear transform on the original feature space

$$\begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix} \rightarrow \begin{bmatrix} x_{i_1} \\ \vdots \\ x_{i_{d'}} \end{bmatrix}$$

Feature Selection ($d' < d$)

$$\begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix} \rightarrow \begin{bmatrix} y_1 \\ \vdots \\ y_{d'} \end{bmatrix} = f\left(\begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}\right)$$

Feature Extraction

Feature Extraction

- Unsupervised feature extraction:

$$\boldsymbol{X} = \begin{bmatrix} x_1^{(1)} & \cdots & x_d^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(N)} & \cdots & x_d^{(N)} \end{bmatrix}$$

Feature Extraction

A mapping $f: \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$
Or
only the transformed data

$$\boldsymbol{X}' = \begin{bmatrix} x'^{(1)}_1 & \cdots & x'^{(1)}_{d'} \\ \vdots & \ddots & \vdots \\ x'^{(N)}_1 & \cdots & x'^{(N)}_{d'} \end{bmatrix}$$

- Supervised feature extraction:

$$\boldsymbol{X} = \begin{bmatrix} x_1^{(1)} & \cdots & x_d^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(N)} & \cdots & x_d^{(N)} \end{bmatrix}$$

$$Y = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(N)} \end{bmatrix}$$

Feature Extraction

A mapping $f: \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$
Or
only the transformed data

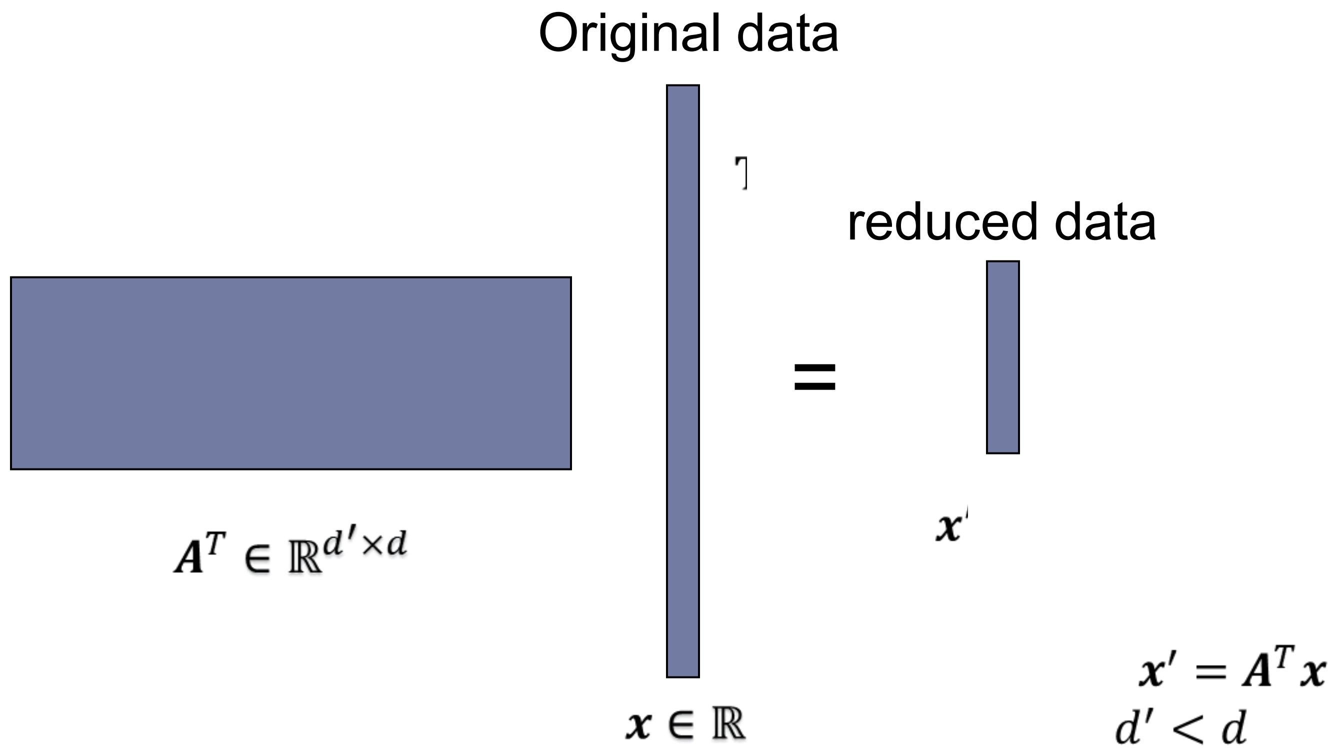
$$\boldsymbol{X}' = \begin{bmatrix} x'^{(1)}_1 & \cdots & x'^{(1)}_{d'} \\ \vdots & \ddots & \vdots \\ x'^{(N)}_1 & \cdots & x'^{(N)}_{d'} \end{bmatrix}$$

Unsupervised Feature Reduction

- Visualization and interpretation: projection of high-dimensional data onto 2D or 3D.
- Data compression: efficient storage, communication, or and retrieval.
- Pre-process: to improve accuracy by reducing features
  - As a preprocessing step to reduce dimensions for supervised learning tasks
  - Helps avoiding overfitting
- Noise removal
  - E.g, "noise" in the images introduced by minor lighting variations, slightly different imaging conditions,

Linear Transformation

- For linear transformation, we find an explicit mapping $f(\boldsymbol{x}) = \boldsymbol{A}^T\boldsymbol{x}$ that can transform also new data vectors.



Linear Transformation

- Linear transformation are simple mappings

$$\mathbf{x}' = \mathbf{A}^T \mathbf{x} \quad (\mathbf{x}'_j = \mathbf{a}_j^T \mathbf{x}) \qquad j = 1, \ldots, d$$

$$\boldsymbol{A} = \begin{bmatrix} \boxed{\begin{matrix} a_{11} \\ \vdots \\ a_{d1} \end{matrix}} & \begin{matrix} \cdots \\ \ddots \\ \cdots \end{matrix} & \boxed{\begin{matrix} a_{1d} \\ \vdots \\ a_{dd'} \end{matrix}} \end{bmatrix}$$

$\mathbf{a}_1$ $\qquad$ $\mathbf{a}_{d'}$

Linear Dimensionality Reduction

- Unsupervised
  - **Principal Component Analysis (PCA)**
  - Singular Value Decomposition (SVD)
  - Independent Component Analysis (ICA)
  - Multi Dimensional Scaling (MDS)
  - Canonical Correlation Analysis (CCA)
  - …

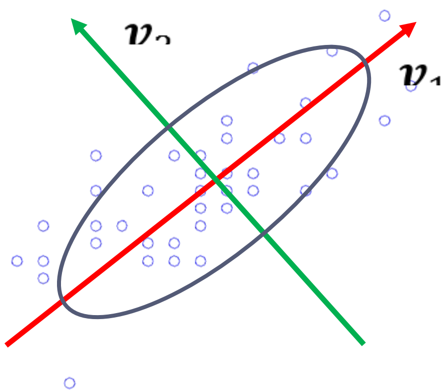
Principal Component Analysis (PCA)

- Also known as Karhonen-Loeve (KL) transform

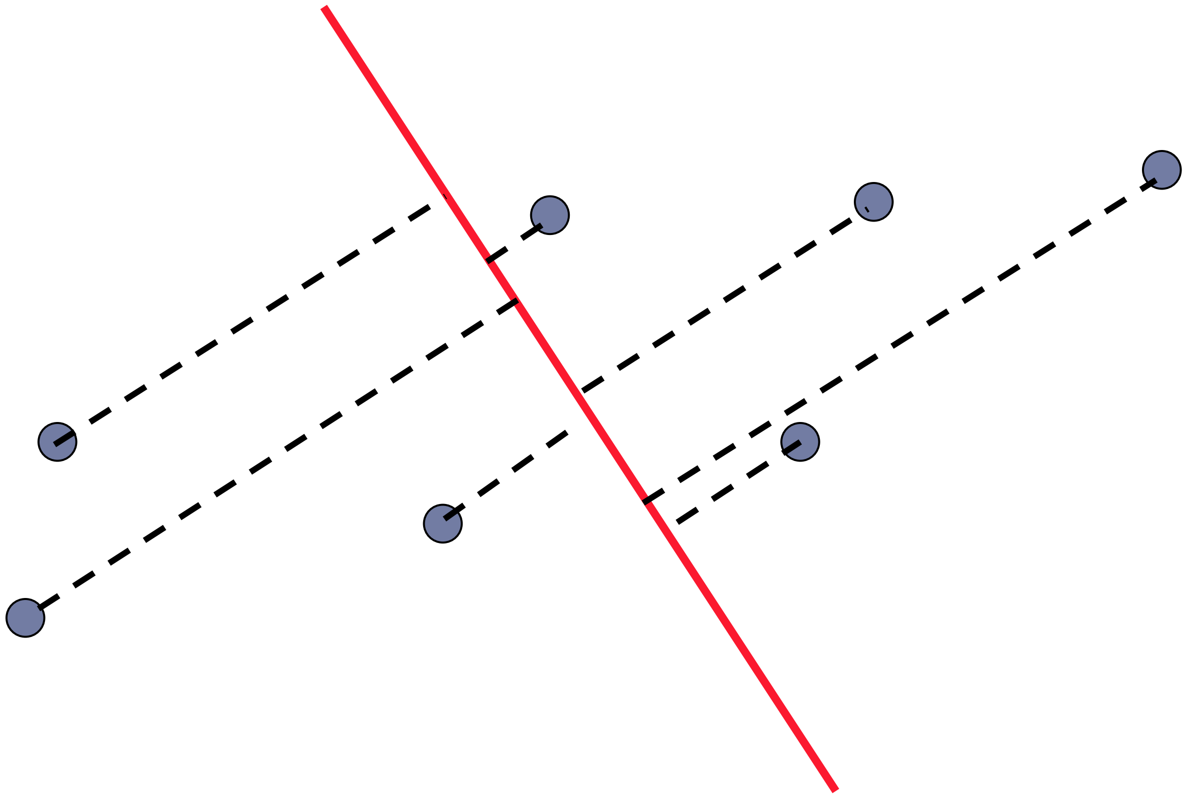
- Principal Components (PCs): **orthogonal** vectors that are **ordered** by the fraction of the total information (variation) in the corresponding directions
  - Find the directions at which data approximately lie

Principal components

- If data has a Gaussian distribution $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, the direction of the largest variance can be found by the eigenvector of $\boldsymbol{\Sigma}$ that corresponds to the largest eigenvalue of $\boldsymbol{\Sigma}$

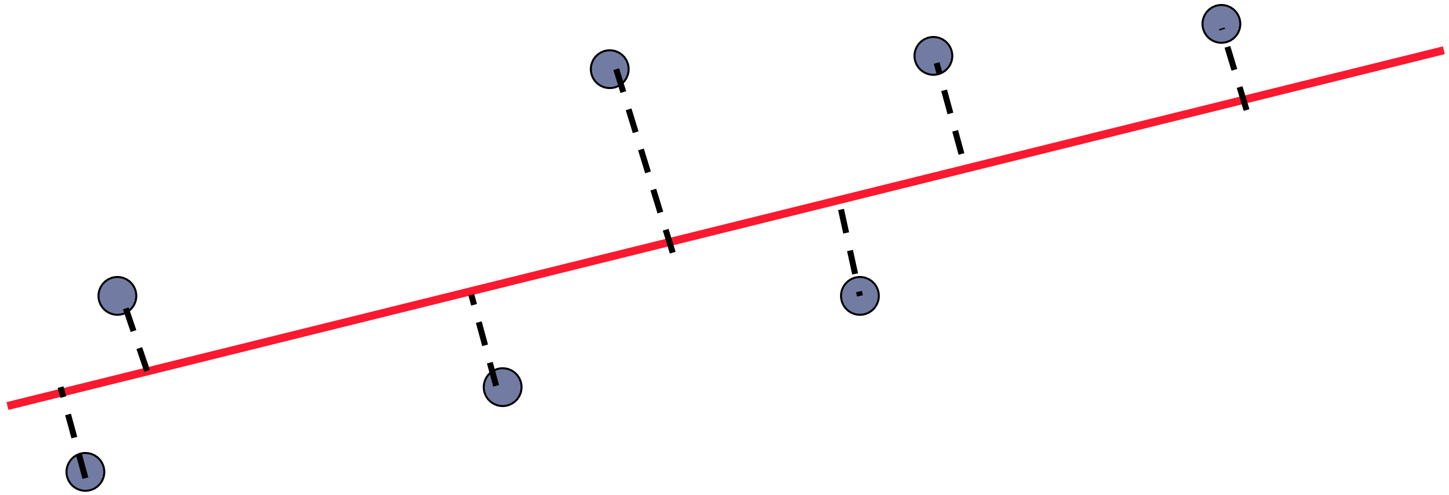


Example: random direction



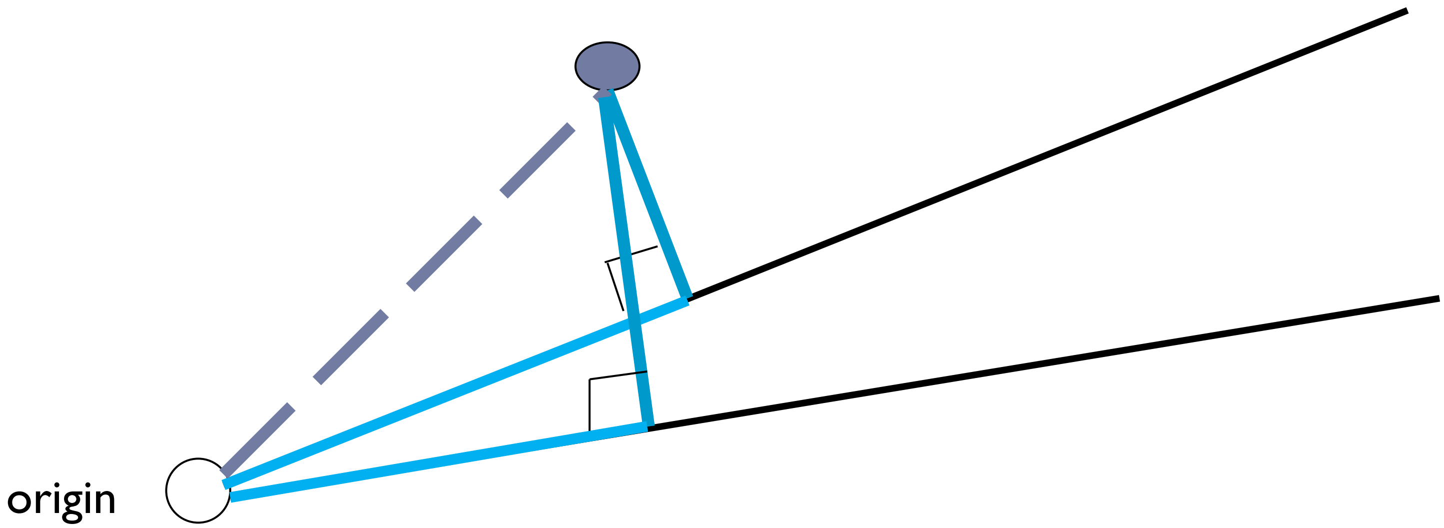
Example: principal component

- Find the direction that preserves important aspect of data



Least Squares Error and Maximum Variance Views Are Equivalent (1-dim Interpretation)

- When data are mean-removed:
  - Minimizing sum of square distances to the line is equivalent to maximizing the sum of squares of the projections on that line (Pythagoras).



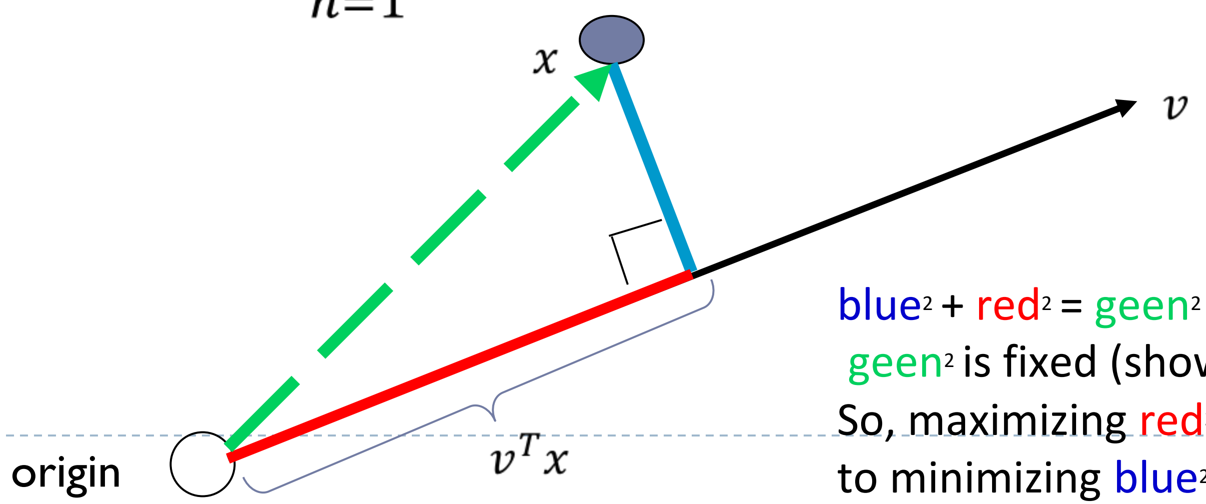
Two interpretations (for mean centered data)

- Maximum variance subspace

$$\operatorname*{argmax}_{v} \frac{1}{N}\sum_{n=1}^{N}\left(v^T x^{(n)}\right)^2 = v^T S v$$

- Minimum reconstruction error

$$\operatorname*{argmin}_{v} \sum_{n=1}^{N}\left\|x^{(n)} - \left(v^T x^{(n)}\right)v\right\|^2$$



blue² + red² = geen²
geen² is fixed (shows data)
So, maximizing red² is equivalent to minimizing blue²

Principal Component Analysis (PCA)

- Goal: **reducing the dimensionality** of the data while preserving important aspects of the data

- Two equal views: find directions for which
  - the **variation** presents in the dataset is as **much** as possible.
  - the **reconstruction error is minimized**.

- PCs can be found as the "**best" eigenvectors** of the covariance matrix of the data points.

PCA: Steps

- Input: $N \times d$ data matrix $\boldsymbol{X}$ (each row contain a d dimensional data point)
  - $\overline{\boldsymbol{x}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}^{(i)}$
  - $\widetilde{\boldsymbol{X}} \leftarrow$ Mean value of data points is subtracted from rows of $\boldsymbol{X}$
  - $\boldsymbol{S} = \frac{1}{N}\widetilde{\boldsymbol{X}}^T\widetilde{\boldsymbol{X}}$ (Covariance matrix)
  - Calculate eigenvalue and eigenvectors of $\boldsymbol{S}$
  - Pick d' eigenvectors corresponding to the largest eigenvalues and put them in the columns of $\boldsymbol{A} = [\underbrace{\boldsymbol{v}_1}_{\text{First PC}}, \dots, \underbrace{\boldsymbol{v}_{d'}}_{\text{d'-th PC}}]$
  - $\boldsymbol{X}' = \boldsymbol{X}\boldsymbol{A}$

Covariance Matrix

$$\boldsymbol{\mu}_x = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_d \end{bmatrix} = \begin{bmatrix} E(x_1) \\ \vdots \\ E(x_d) \end{bmatrix}$$

$$\boldsymbol{\Sigma} = E[(\boldsymbol{x} - \boldsymbol{\mu}_x)(\boldsymbol{x} - \boldsymbol{\mu}_x)^T]$$

Covariance Matrix

$$\boldsymbol{\mu}_x = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_d \end{bmatrix} = \begin{bmatrix} E(x_1) \\ \vdots \\ E(x_d) \end{bmatrix}$$

$$\boldsymbol{\Sigma} = E[(\boldsymbol{x} - \boldsymbol{\mu}_x)(\boldsymbol{x} - \boldsymbol{\mu}_x)^T]$$

- ML estimate of covariance matrix from data points $\{\boldsymbol{x}^{(i)}\}_{i=1}^{N}$:

$$\boldsymbol{S} = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x}^{(i)} - \overline{\boldsymbol{x}})(\boldsymbol{x}^{(i)} - \overline{\boldsymbol{x}})^T = \frac{1}{N}(\widetilde{\boldsymbol{X}}^T\widetilde{\boldsymbol{X}})$$

$$\widetilde{\boldsymbol{X}} = \begin{bmatrix} \widetilde{\boldsymbol{x}}^{(1)} \\ \vdots \\ \widetilde{\boldsymbol{x}}^{(N)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}^{(1)} - \overline{\boldsymbol{x}} \\ \vdots \\ \boldsymbol{x}^{(N)} - \overline{\boldsymbol{x}} \end{bmatrix} \qquad \overline{\boldsymbol{x}} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}^{(i)}$$

Mean-centered data

Find 1st principal component

- Find vector $\boldsymbol{v}_1$ that maximizes sample variance of the projected data:

$$\max_{v_1} \frac{1}{N}\sum_{n=1}^{N}\left(v_1^T x^{(n)} - v_1^T \bar{x}\right)^2$$

$$= \frac{1}{N}\sum_{n=1}^{N} v_1^T\left(x^{(n)} - \bar{x}\right)\left(x^{(n)} - \bar{x}\right)^T v_1$$

$$= v_1^T\left(\frac{1}{N}\sum_{n=1}^{N}\left(x^{(n)} - \bar{x}\right)\left(x^{(n)} - \bar{x}\right)^T\right)v_1 = v_1^T S v_1$$

$$\text{s.t. } v_1^T v_1 = 1$$

Find 1st principal component

- Find vector $\boldsymbol{v}$ that maximizes sample variance of the projected data:

$$\max_{v} \frac{1}{N}\sum_{n=1}^{N}\left(v_1^T x^{(n)} - v_1^T \bar{x}\right)^2 = v_1^T S v_1$$

$$\text{s.t. } v_1^T v_1 = 1$$

$$L(v_1, \lambda_1) = v_1^T S v_1 + \lambda_1 (1 - v_1^T v_1)$$

$$\frac{\partial L}{\partial v_1} = 0 \Rightarrow 2 S v_1 - 2\lambda_1 v_1 = 0$$

$$\Rightarrow S v_1 = \lambda_1 v_1$$

Eigenvector with maximum eigenvalue maximizes the objective

PCA Derivation: Relation between Eigenvalues and Variances

$$S\boldsymbol{v}_j = \lambda_j \boldsymbol{v}_j$$

$$\Rightarrow \underbrace{var(v_j^T x)}_{\text{Variance along j-th eigenvector}} = \boldsymbol{v}_j^T S \boldsymbol{v}_j = \lambda_j \boldsymbol{v}_j^T \boldsymbol{v}_j = \lambda_j$$

- Therefore, eigenvector with maximum eigenvalue maximizes the objective

Finding second principal component

$$\max_{v_2} v_2^T S v_2$$
$$\text{s.t. } v_2^T v_2 = 1$$
$$v_2^T v_1 = 0$$

$$L(v_2, \lambda_2, \alpha) = v_2^T S v_2 + \lambda_2 (1 - v_2^T v_2) - \alpha v_2^T v_1$$

Finding second principal component

$$\max_{v_2} v_2^T S v_2$$

$$\text{s.t. } v_2^T v_2 = 1$$

$$v_2^T v_1 = 0$$

$$L(v_2, \lambda_2, \alpha) = v_2^T S v_2 + \lambda_2(1 - v_2^T v_2) - \alpha v_2^T v_1$$

Finding α: $\frac{\partial L}{\partial v_2} = 0 \Rightarrow 2Sv_2 - 2\lambda_2 v_2 - \alpha v_1 = 0$

$$\Rightarrow 2v_1^T S v_2 - 2\lambda_2 v_1^T v_2 - \alpha v_1^T v_1 = 0$$

$$\Rightarrow 2\lambda_1 v_1^T v_2 - 2\lambda_2 \times 0 - \alpha = 0$$

$$\Rightarrow \alpha = 0$$

Finding second principal component

$$\max_{v_2} v_2^T S v_2$$

$$\text{s.t. } v_2^T v_2 = 1$$

$$v_2^T v_1 = 0$$

$$L(v_2, \lambda_2, \alpha) = v_2^T S v_2 + \lambda_2(1 - v_2^T v_2) - \cancelto{0}{\alpha v_2^T v_1}$$

Finding λ_2: $$\frac{\partial L}{\partial v_2} = 0 \Rightarrow 2Sv_2 - 2\lambda_2 v_2 = 0$$

$$\Rightarrow S v_2 = \lambda_2 v_2$$

v_2 is the eigenvector corresponding to the second largest eigenvalue

Find principal components

- For symmetric matrices, there exist eigen-vectors that are orthogonal.

- Let $v_1, \dots v_d$ denote the eigen-vectors of S such that:

$$v_i^T v_j = 0, \quad \forall i \neq j$$
$$v_i^T v_i = 1, \quad \forall i$$

PCA

- Eigenvalues: $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \cdots$
  - The first PC $\boldsymbol{v}_1$ is the the eigenvector of the sample covariance matrix S associated with the largest eigenvalue.
  - The 2nd PC $\boldsymbol{v}_2$ is the the eigenvector of the sample covariance matrix S associated with the second largest eigenvalue
  - And so on …

- Find eigenvectors with the top k eigenvalues

PCA: Steps

- Input: $N \times d$ data matrix $\boldsymbol{X}$ (each row contain a d dimensional data point)
  - $\bar{\boldsymbol{x}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}^{(i)}$
  - $\widetilde{\boldsymbol{X}} \leftarrow$ Mean value of data points is subtracted from rows of $\boldsymbol{X}$
  - $\boldsymbol{S} = \frac{1}{N}\widetilde{\boldsymbol{X}}^T\widetilde{\boldsymbol{X}}$ (Covariance matrix)
  - Calculate eigenvalue and eigenvectors of $\boldsymbol{S}$
  - Pick d' eigenvectors corresponding to the largest eigenvalues and put them in the columns of $\boldsymbol{A} = [\boldsymbol{v}_1, \dots, \boldsymbol{v}_{d'}]$
  - $\boldsymbol{X}' = \boldsymbol{X}\boldsymbol{A}$

First PC d’-th PC

Reconstruction

$$\boldsymbol{x}' = \begin{bmatrix} \boldsymbol{v}_1^T \boldsymbol{x} \\ \vdots \\ \boldsymbol{v}_{d'}^T \boldsymbol{x} \end{bmatrix}$$

$$\boldsymbol{A} = [\boldsymbol{v}_1, \dots, \boldsymbol{v}_{d'}]$$

$$\boldsymbol{x}' = \boldsymbol{A}^T(\boldsymbol{x} - \overline{\boldsymbol{x}})$$

$$\Rightarrow \widehat{\boldsymbol{x}} = \overline{\boldsymbol{x}} + \boldsymbol{A}\boldsymbol{x}' = \overline{\boldsymbol{x}} + \boldsymbol{A}\boldsymbol{A}^T(\boldsymbol{x} - \overline{\boldsymbol{x}})$$

- Incorporating all eigenvectors in $\boldsymbol{A} = [\boldsymbol{v}_1, \dots, \boldsymbol{v}_d]$:

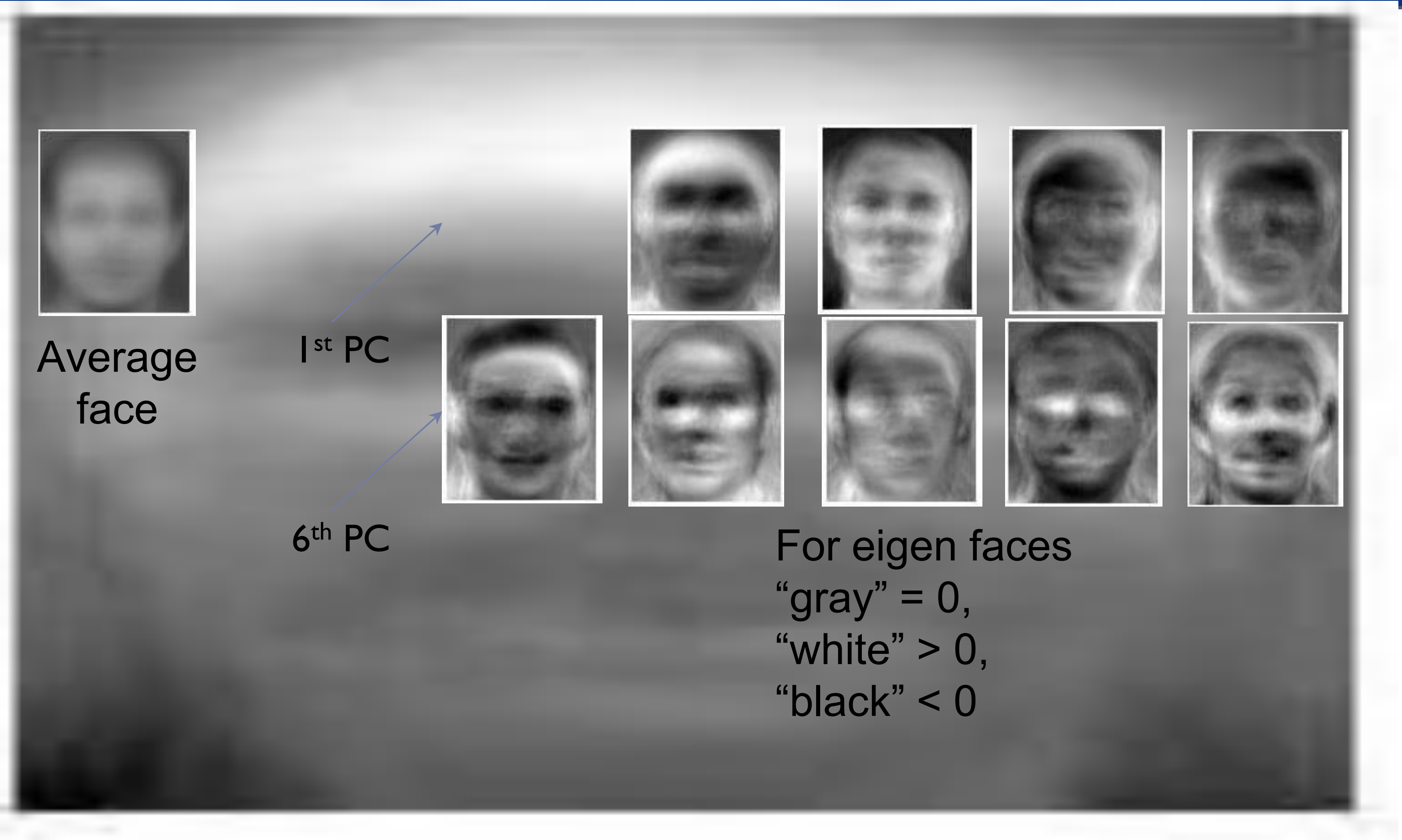
  $\Longrightarrow$ If $d' = d$ then $\boldsymbol{x}$ can be reconstructed exactly from $\boldsymbol{x}'$

PCA on Faces: “Eigenfaces”

- ORL Database

Some Images

PCA on Faces: "Eigenfaces"



PCA on Faces:

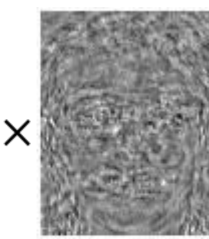
$\boldsymbol{x}$ is a $112 \times 92 = 10304$ dimensional vector containing intensity of the pixels of this image and $\widetilde{\boldsymbol{x}} = \boldsymbol{x} - \overline{\boldsymbol{x}}$

Feature vector=$[x'_1, x'_2, \dots, x'_{d'}]$

x'_i → The projection of $\boldsymbol{x}$ on the i-th PC

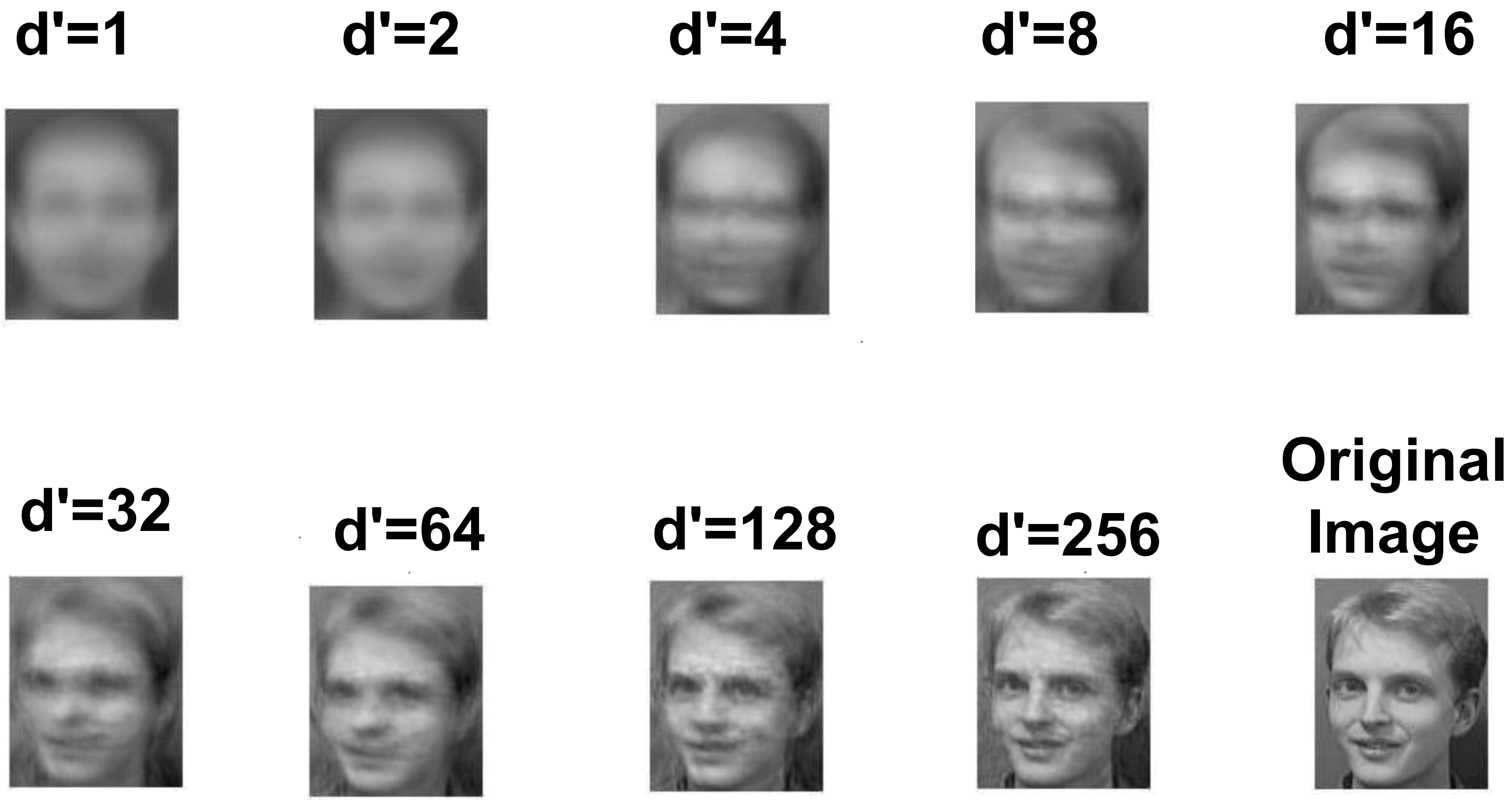
$\widehat{\boldsymbol{x}}$

$$\widehat{\boldsymbol{x}} = \overline{\boldsymbol{x}} + \sum_{i=1}^{d'} (\boldsymbol{v}_i^T \widetilde{\boldsymbol{x}}) \times \boldsymbol{v}_i$$

 $=$  $+x'_1 \times$  $+x'_2 \times$  $+ \cdots$ $+x'_{256} \times$ 

Average Face

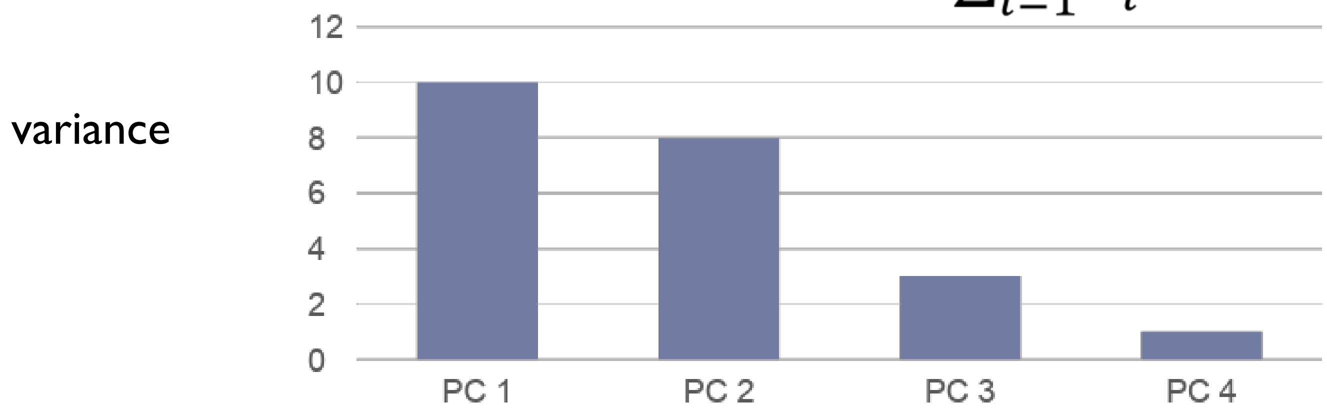
PCA on Faces: Reconstructed Face



Dimensionality reduction by PCA

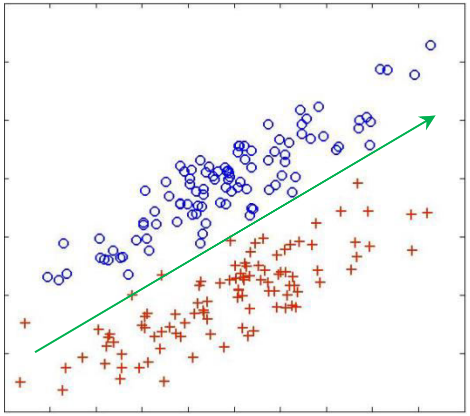
- Data may lie near a linear subspace of high-dimensional input space
- Only keep data projections onto principal components with **large** eigenvalues
- Plot of the eigenvalues (or variances of principal components) against their indices.

$$\frac{\sum_{i=1}^{d'} \lambda_i}{\sum_{i=1}^{d} \lambda_i} \times 100$$



Unsupervised feature extraction drawback

- PCA drawback: An excellent information packing transform does not necessarily lead to a good class separability.
  - The directions of the maximum variance may be useless for classification purpose



PCA vs. LDA

- Although LDA often provide more suitable features for classification tasks, PCA might outperform LDA in some situations:
  - When there are many unlabeled data while no or small amount of labeled data
    - when the number of samples per class is small (overfitting problem of LDA)
  - when the number of the desired features is more than $C - 1$
  - when the training data non-uniformly sample the underlying distribution

- Semi-supervised feature extraction
  - E.g., PCA+LDA, Regularized LDA, Locally FDA (LFDA)

PCA: Summary

- Global optimum is found by eigenvector method
- No parameter tuning
- However, it is limited to:
  - using second order statistics
  - limited to linear projections

Resources

- C. Bishop, “Pattern Recognition and Machine Learning”, Chapter 12.