Dimensionality reduction

CE-477: Machine Learning - CS-828: Theory of Machine Learning
Sharif University of Technology
Fall 2024

Fatemeh Seyyedsalehi

References of the lecture are mentioned in the last slide

Unsupervised learning problems

- Density estimation
  - Fit a continuous distribution to discrete data.
  - Parametric and non-parametric methods
- Dimensionality reduction
  - Data often lies near a low-dimensional subspace (or manifold) in feature space.
- Clustering
  - Partition data into groups of similar/nearby points.

Dimensionality reduction

- 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
$(d' < d)$

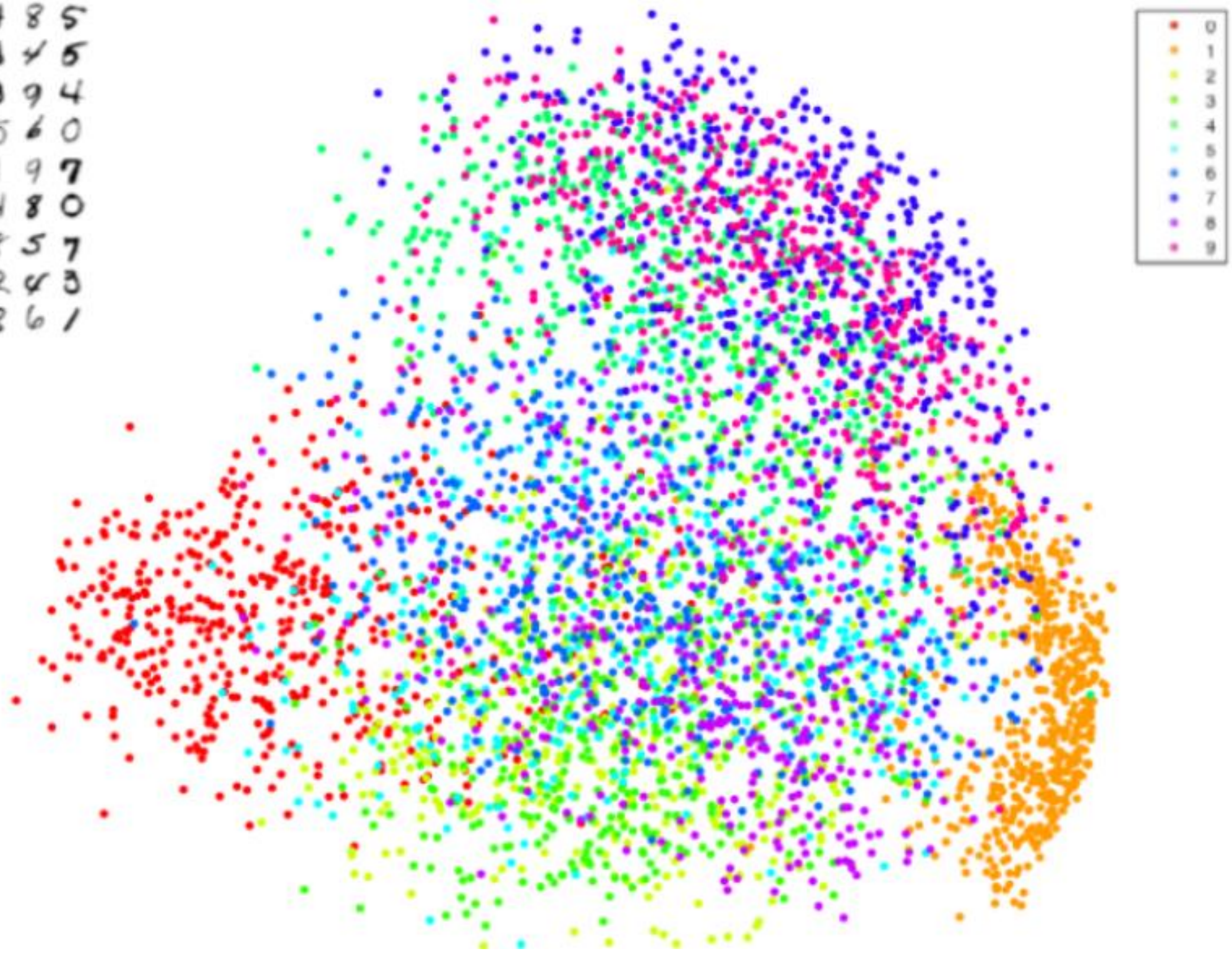
$$\begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix} \rightarrow \begin{bmatrix} y_1 \\ \vdots \\ y_{d'} \end{bmatrix} = f\left(\begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}\right)$$

Feature
Extraction

Dimensionality reduction benefits

- Visualization and interpretation: projection of high-dimensional data onto 2D or 3D.
- Data compression: efficient storage, communication, or 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.

Dimensionality reduction benefits



Dimensionality reduction

- We introduce two methods:
  - Principal component analysis (PCA)
  - Independent component analysis (ICA)

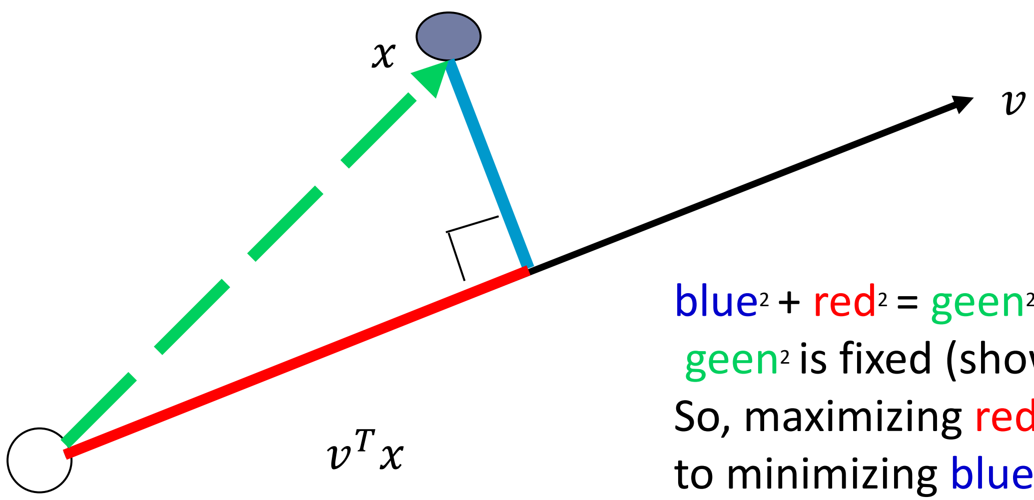
Principal component analysis (PCA)

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

- 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 component analysis (PCA)

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



blue^2 + red^2 = geen^2

geen^2 is fixed (shows data)

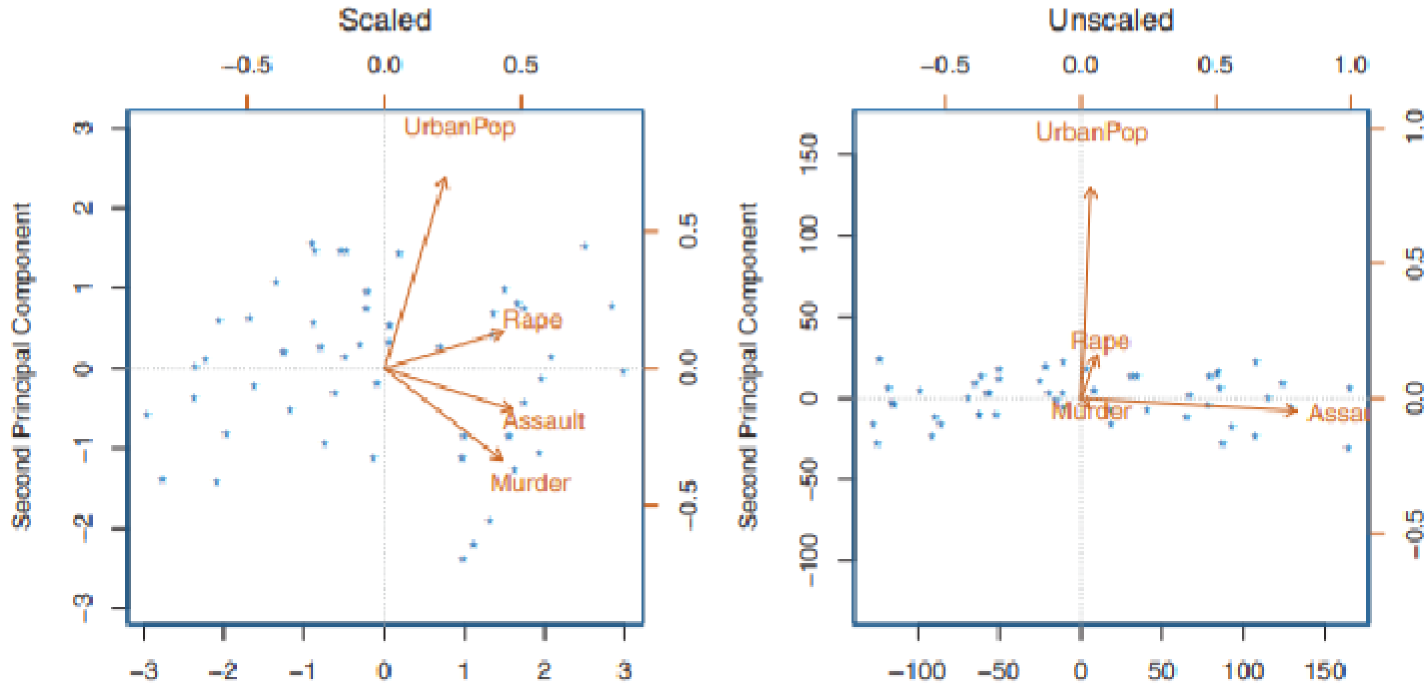
So, maximizing red^2 is equivalent to minimizing blue^2

Principal component analysis (PCA)

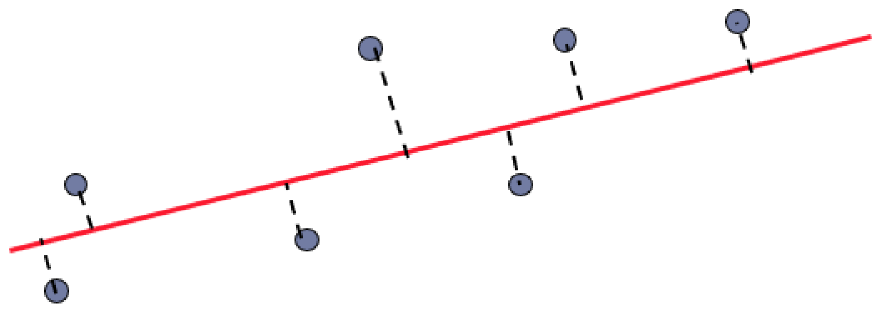
- We usually perform a preprocessing step.
  - Center the data
    - Zeroing out the mean of each feature
  - Scaling to normalize each feature to have variance 1
    - An arbitrary step.
    - May affect the final result !!
    - It helps when unit of measurements of features are different and some features may be ignored without normalization

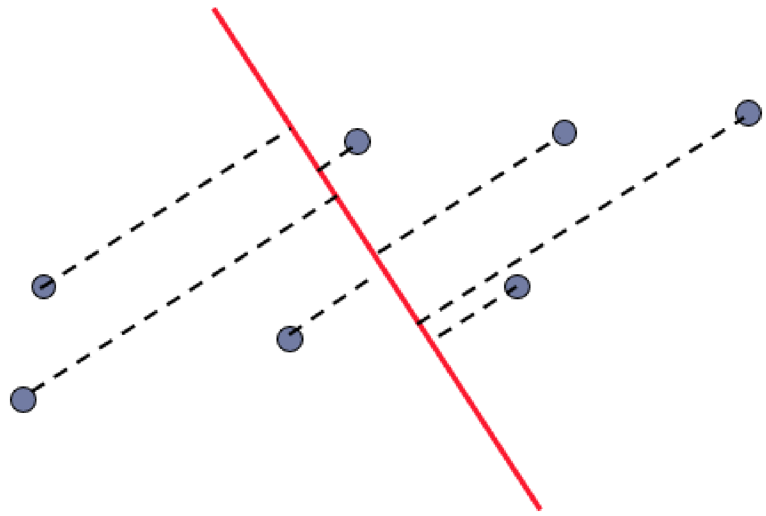
Principal component analysis (PCA)

- Scaling to normalize each feature may affect the final result !!



Random direction vs. principal component





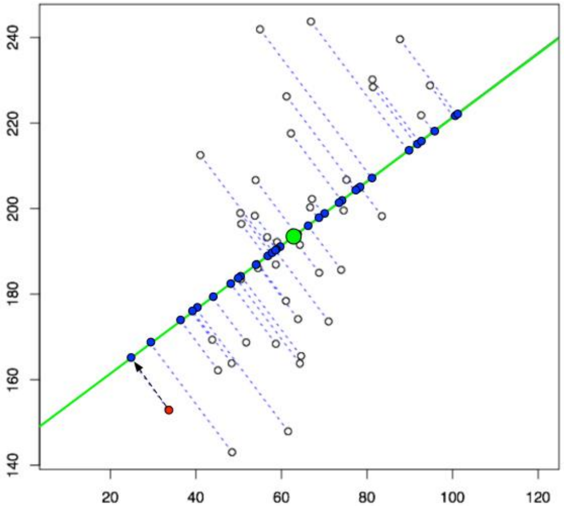
Principal component analysis (PCA)

- First view
  - Find directions with the maximum variations

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

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

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



Principal component analysis (PCA)

- We should optimize s.t. $v_1^T v_1 = 1$, to avoid the obvious solution $v \rightarrow \infty$

$$\max_{v} \frac{1}{N} \sum_{n=1}^{N} \left(v_1^T x^{(n)}\right)^2 = v_1^T S v_1$$
$$\text{s.t. } v_1^T v_1 = 1$$

- Eigenvector with maximum eigenvalue maximizes the objective
  - Using Lagrangian multiplier technique

$$L(v_1, \lambda_1) = v_1^T S v_1 + \lambda_1 (1 - v_1^T v_1)$$
$$\frac{\partial L}{\partial v_1} = 0 \Rightarrow 2 S v_1 - 2 \lambda_1 v_1 = 0$$
$$\Rightarrow S v_1 = \lambda_1 v_1$$

Principal component analysis (PCA)

- As we have $S\boldsymbol{v}_j = \lambda_j \boldsymbol{v}_j$,

$$\Rightarrow var\left(v_j^T x\right) = \boldsymbol{v}_j^T S \boldsymbol{v}_j = \lambda_j \boldsymbol{v}_j^T \boldsymbol{v}_j = \lambda_j$$

- The variance along an eigenvector $\boldsymbol{v}_j$ equals the eigenvalue λ_j

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 ...

- To reduce the dimension of the data to k, we select eigenvectors with the top k eigenvalues

PCA: Steps

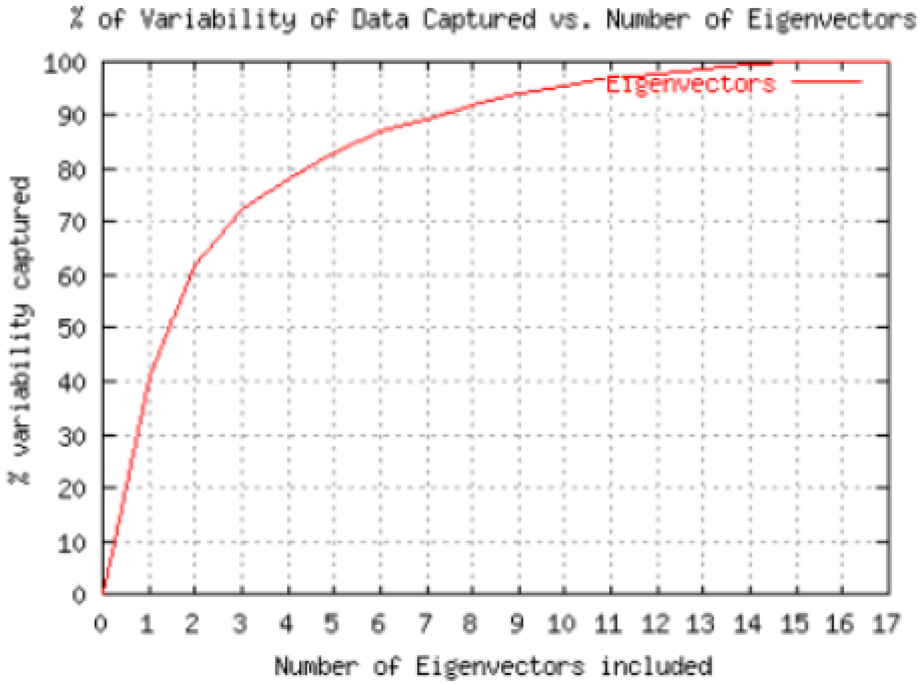
- Input: $N \times d$ data matrix $\boldsymbol{X}$ (each row contain a d dimensional data point)
  - $\bar{\boldsymbol{x}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}^{(i)}$
  - $\widetilde{\boldsymbol{X}} \leftarrow$ Mean value of data points is subtracted from rows of $\boldsymbol{X}$
  - $\boldsymbol{S}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

Principal component analysis (PCA)

- Eigen-vectors of symmetric matrices are orthogonal.
- Covariance matrix is symmetric.
  - Principal component are orthonormal
- We have,

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

Principal component analysis (PCA)



$$\frac{\sum_{i=d-k+1}^{d} \lambda_i}{\sum_{i=1}^{d} \lambda_i}$$

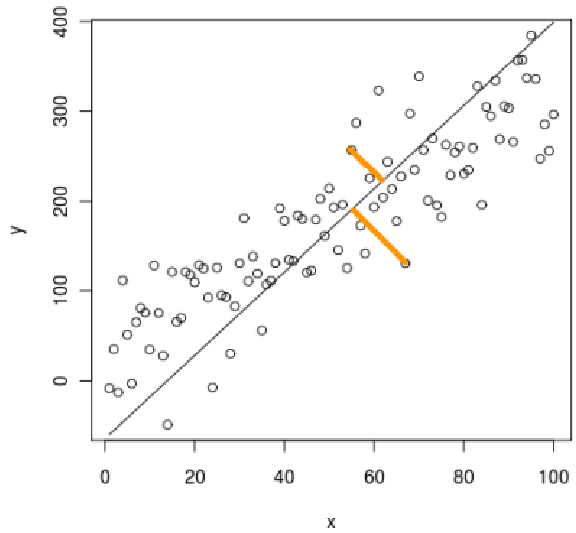
Principal component analysis (PCA)

- Second view
  - Find directions with the minimum reconstruction error

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

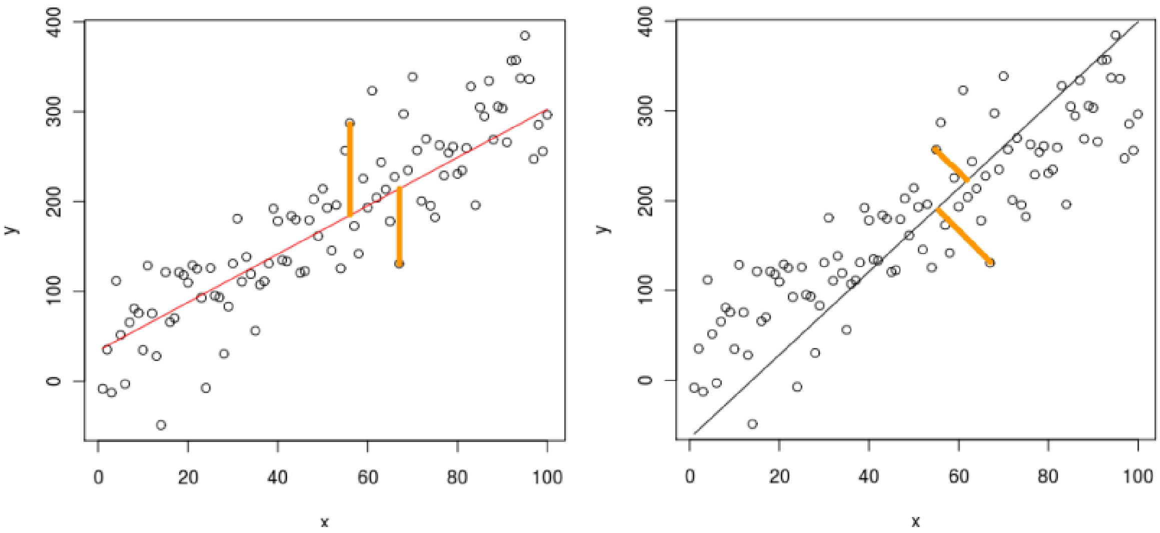
$$\text{s.t. } v_1^T v_1 = 1$$

- Show this it has an equal solution with the first view optimization problem



Principal component analysis (PCA)

- In linear regression, the projection direction is always vertical; whereas in PCA, the projection direction is orthogonal to the projection hyperplane

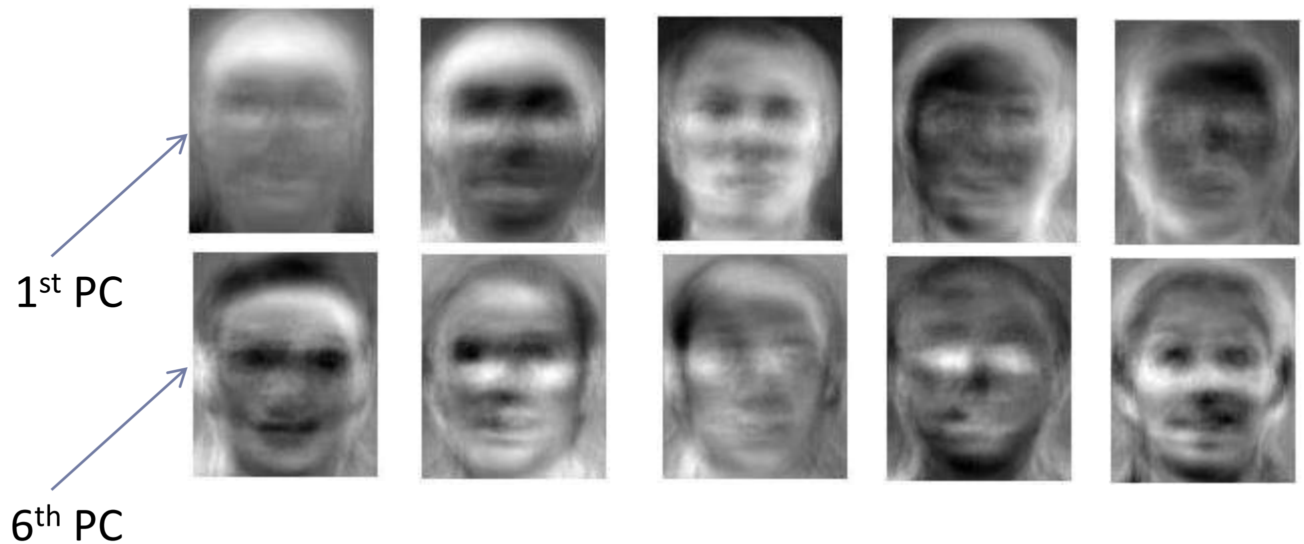


PCA on Faces: “Eigenfaces”

- ORL Database
  - Some images

PCA on Faces: “Eigenfaces”

Average face



PCA on Faces: “Eigenfaces”

Reconstructing a sample

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

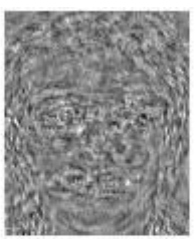
$$\widehat{\boldsymbol{x}} = \overline{\boldsymbol{x}} + \sum_{i=1}^{d'} x_i' \times \boldsymbol{v}_i$$

$=$

$+x_1' \times$

$+x_2' \times$

$+ \cdots \quad +x_{256}' \times$



Average Face

PCA on Faces: Reconstructed Face

d'=1

d'=2

d'=4

d'=8

d'=16

d'=32

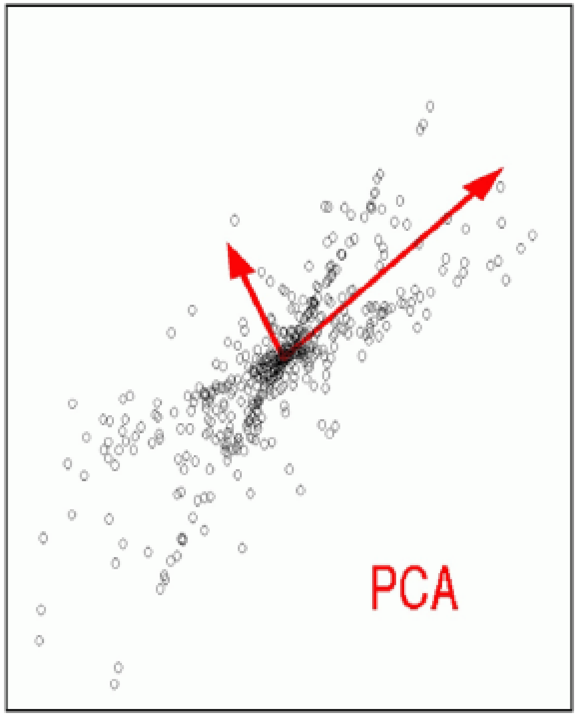
d'=64

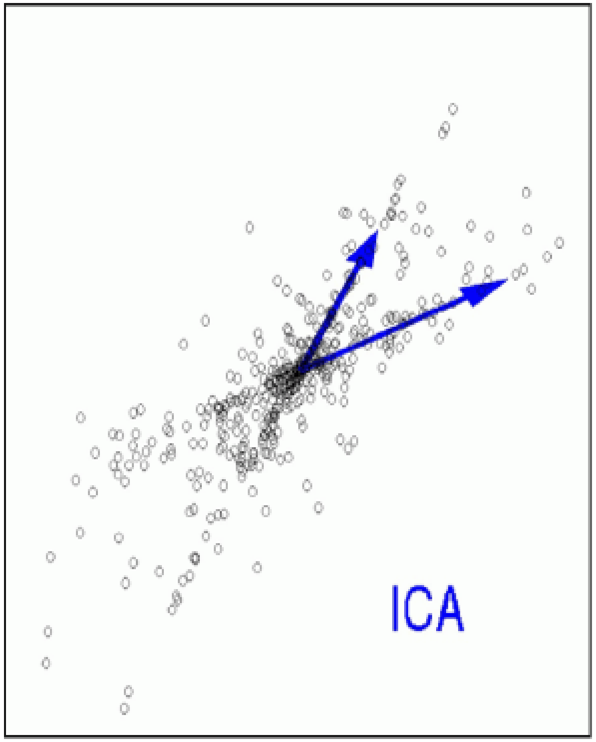
d'=128

d'=256

Original Image

PCA vs. ICA





PCA vs. ICA

ICA	PCA
Optimizes higher-order statistics.	PCA optimizes the covariance matrix of the data which represents second-order statistics.
ICA finds independent components.	PCA finds uncorrelated components.
It does not emphasize the components' reciprocal orthogonality.	It focuses on the major components' mutual orthogonality.
It decomposes the mixed signal into the signals of its separate sources.	It decreases the dimensions to avoid the overfitting issue.

Independent component analysis (ICA)

- cocktail party problem
  - d speakers are speaking simultaneously at a party, and any microphone placed in the room records only an overlapping combination of the d speakers' voices.
  - Each microphone is in a different distance from each of the speakers and records a different combination of the speakers' voices.

$$\boldsymbol{x} = A\boldsymbol{s}$$

  - $\boldsymbol{x}$: voice recorded in microphones in a specific time snapshot.
  - $\boldsymbol{s}$: sources
  - A: mixing matrix

Independent component analysis (ICA)

- $\boldsymbol{x} \in \mathbb{R}^m$
  - From each microphone we observe a random variable at time t.
  - This vector shows observed random variables from all m microphones.
- $\boldsymbol{s} \in \mathbb{R}^d$
  - Each source generates a random variable at time t.
  - This vector shows random variables generated by all d sources at time t.

- A: mixing matrix

$$\boldsymbol{x} = A\boldsymbol{s}$$

- Our goal is to find the unmixing matrix W:

$$\boldsymbol{s} = W\boldsymbol{x}$$

Independent component analysis (ICA)

- The joint distribution of independent sources:

$$p_s(\boldsymbol{s}) = \prod_{j=1}^{d} p_s(s_j)$$

- We have,

$$\boldsymbol{s} = W\boldsymbol{x}$$

- As $\boldsymbol{x}$ is a linear transform of $\boldsymbol{s}$, we can write the density of p_x as a function of p_s as follows,

$$p_x(\boldsymbol{x}) = \prod_{j=1}^{d} p_s(w_j^T \boldsymbol{x}) \;.\, |W|$$

Independent component analysis (ICA)

- Assuming a sigmoid CDF for p_s, σ, we can construct a likelihood function to estimate parameters W
  - $\boldsymbol{x}^i$ is the ith observed vector.
    - For example, the voice recorded in all microphones at time i.
  - Therefore, the unmixing matrix can be obtained.

$$\ln \prod_{i=1}^{n} p(\boldsymbol{x}^i; W) = \sum_{i=1}^{n} (\sum_{j=1}^{d} \log \sigma'(w_j^T \boldsymbol{x}^i) + \log |W|)$$