Dimensionality Reduction summary

G.Giannoulis, G. Voulgaris, Prof. Ioannis Pitas
Aristotle University of Thessaloniki
pitas@csd.auth.gr
www.aiia.csd.auth.gr
Version 2.8

Artificial Intelligence &
Information Analysis Lab

Dimensionality reduction

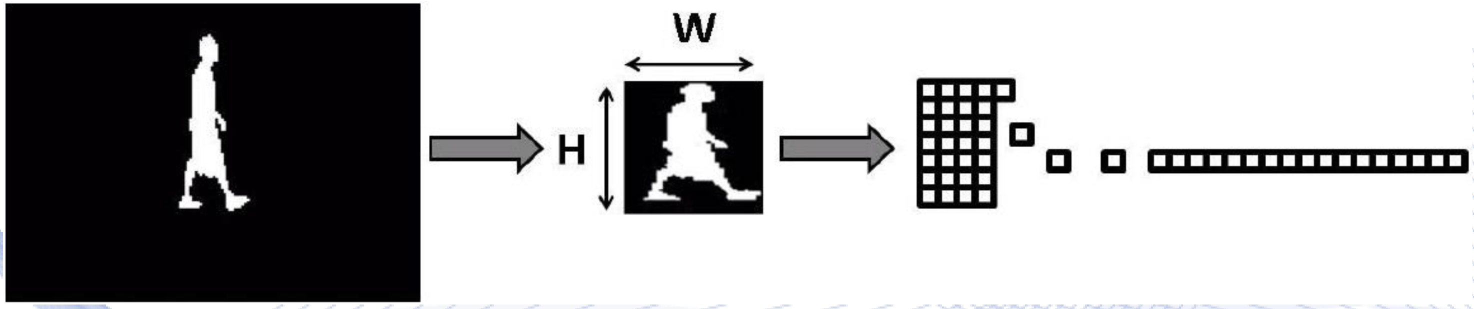
- Introduction
- Feature selection
- Principal Component Analysis
- Linear Discriminant Analysis
- Multidimensional Scaling
- Non-negative matrix factorization

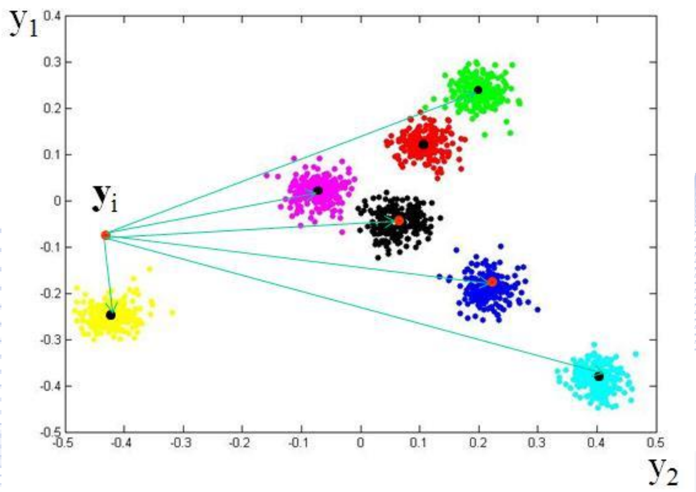
Dimensionality Reduction

- Given a data sample $\mathbf{x} \in \mathbb{R}^n$, compute a new sample representation of reduced dimensionality $\hat{\mathbf{x}} \in \mathbb{R}^d$.
- Typically, lower dimensionality satisfies $d \ll n$.
- The representation $\hat{\mathbf{x}}$ is meant:
    - to capture relevant high level information from the initial sample $\mathbf{x}$;
    - provide abstraction from detail;
    - increase robustness to noise;
    - if $d = 2$, dimensionality reduction to $\hat{\mathbf{x}} \in \mathbb{R}^2$, allows data mapping for visualization;
- Helps us solving the ***curse of dimensionality*** problem.

Dimensionality Reduction

- Example: ***Human posture visualization***.
- Dimensionality reduction from $\mathbf{p} \in \mathbb{R}^{HW}$ to $\mathbf{y} \in \mathbb{R}^2$





Binary human body image.

Posture image of fixed size.

Posture vector $\mathbf{p} \in \mathbb{R}^{HW}$.

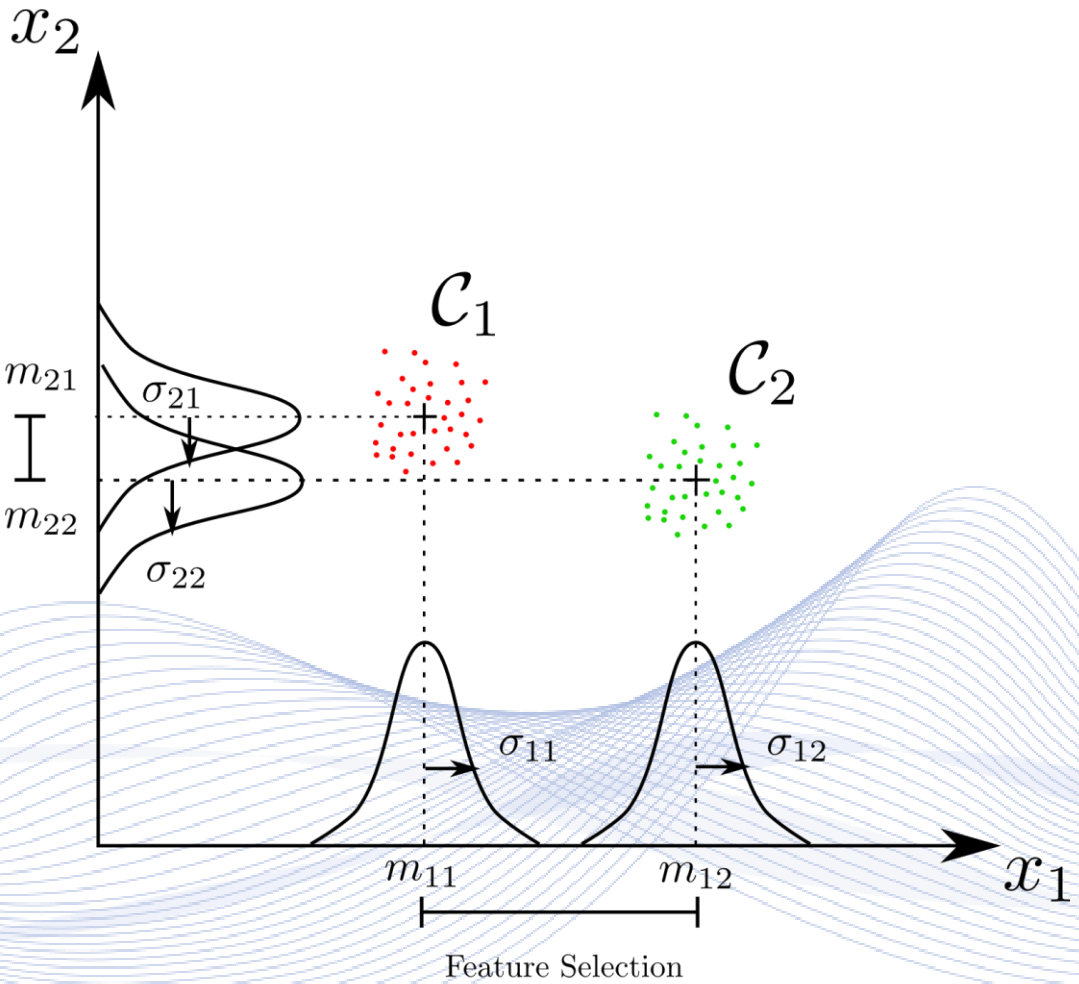
Posture visualization $\mathbf{y} \in \mathbb{R}^2$.

Feature selection

- This is the easiest way to do dimensionality reduction.
- Given N samples $\mathbf{x}_j = [x_{1j}, x_{2j}, \dots, x_{nj}]^T \in \mathbb{R}^n, j = 1, \dots, N$, only the d most informative features are retained, forming a new sample representation of reduced dimensionality $\hat{\mathbf{x}}_j \in \mathbb{R}^d$.

- For a two-class problem:
  - Feature $x_{ij}, j = 1, \dots, N$ pdf location estimates should be far apart.
  - Feature $x_{ij}, j = 1, \dots, N$ pdf dispersion estimates should be small.

Feature selection



Feature selection in the 2D space.

Principal Component Analysis

- Let $\mathbf{v}_1$ be a ***principal component*** or principal direction vector satisfying:

$$\mathbf{v}_1^T\mathbf{v}_1 = 1.$$

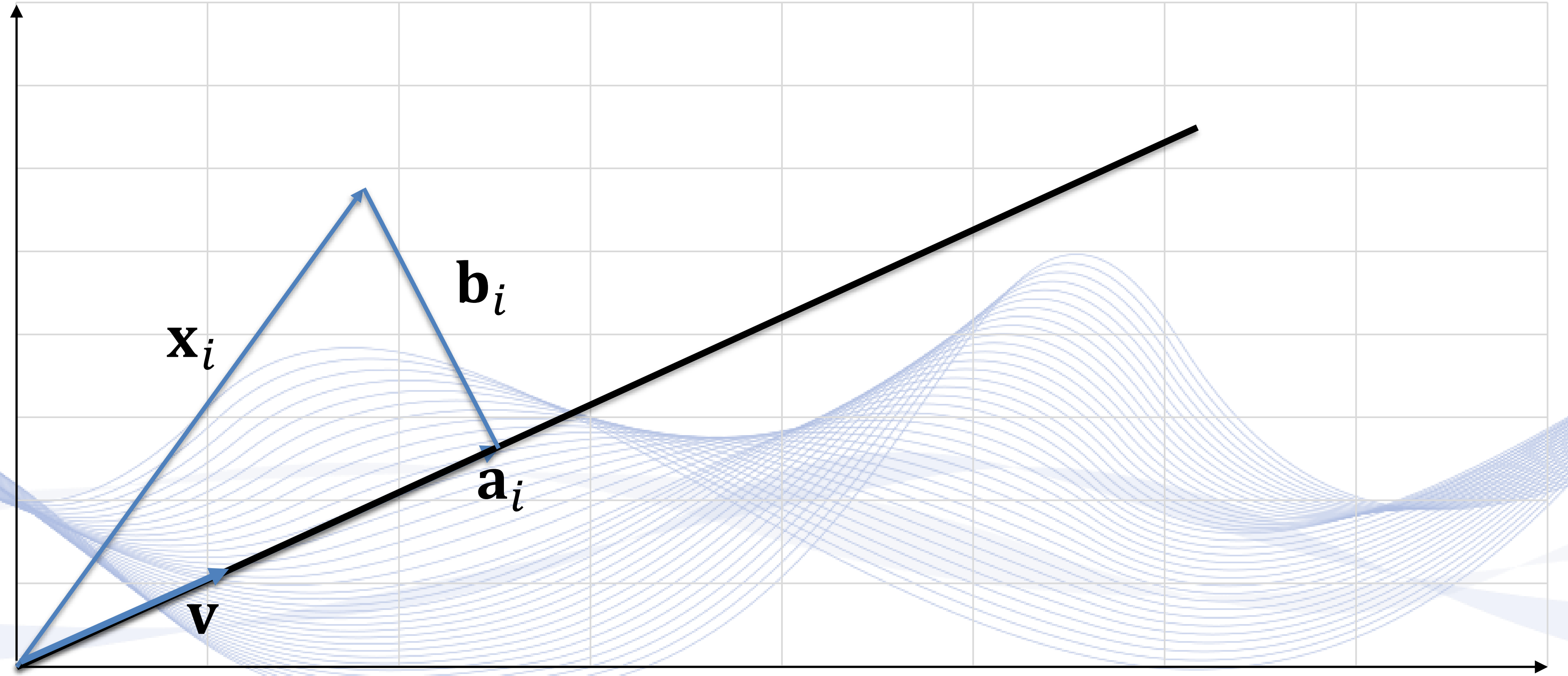
- A set of N points $\mathbf{x}_i \in \mathbb{R}^n, i = 1, \dots, N$, be approximated by their projection on a unit vector $\mathbf{v}_1$:

$$\mathbf{a}_i = (\mathbf{x}_i^T\mathbf{v}_1)\,\mathbf{v}_1 = (\mathbf{v}_1^T\mathbf{x}_i)\,\mathbf{v}_1.$$

- The approximation error vector becomes:

$$\mathbf{b}_i = \mathbf{x}_i - \mathbf{a}_i = \mathbf{x}_i - \left(\mathbf{x}_i^T\mathbf{v}_1\right)\mathbf{v}_1.$$

Principal Component Analysis



Principal Component Analysis

Principal Component Analysis (***PCA***):

If $\mathbf{v}_1$, $\mathbf{v}_2$,..., $\mathbf{v}_d$ are unit vectors $\mathbf{v}_i^T\mathbf{v}_i = 1$ that are perpendicular to each other: $\mathbf{v}_i^T\mathbf{v}_j = 0, (i \neq j)$ form a basis of the a d-dimensional space $\mathbb{R}^d$, and if $\hat{\mathbf{x}}$ is the representation of the n-dimensional vector $\mathbf{x}$:

$$\hat{\mathbf{x}} = \sum_{j=1}^{d} (\mathbf{v}_j^T \mathbf{x})\mathbf{v}_j,$$

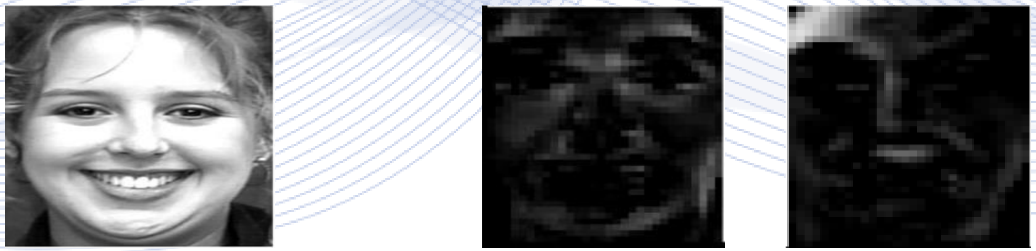
- $\mathbf{v}_j, j = 1, \dots, d$: basis vectors forming a new coordinate system in the d-dimensional space $\mathbb{R}^d$.

Principal Component Analysis

Eigenfaces:

- Reduce facial image (vector $\mathbf{x}$) dimensionality.
- $\mathbf{v}_i, i = 1, \dots, d$: basis image vectors (eigenfaces).
- A facial image is express as a weighted sum of eigenfaces:

$$\hat{\mathbf{x}} = \sum_{j=1}^{d} (\mathbf{v}_j^T \mathbf{x}) \mathbf{v}_j.$$



a) Facial image; b) Example eigenfaces.

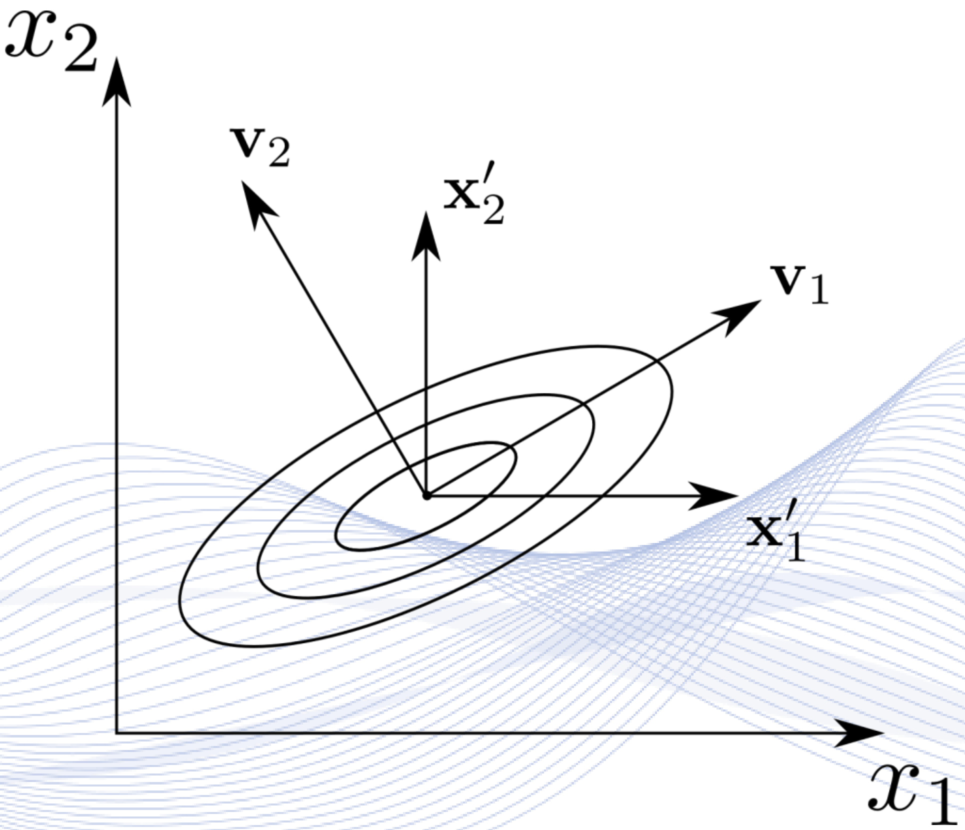
Principal Component Analysis

- PCA can be performed on the autocorrelation matrix $\mathbf{R}_\mathbf{X} = E\{\mathbf{X}\mathbf{X}^T\}$ of random vectors $\mathbf{X}$ belonging to data set $\mathcal{D}$, instead of working on data samples that form matrix $\mathbf{X}$ resulting in matrix $\mathbf{X}\mathbf{X}^T$.

- PCA can be applied after centering the data at their arithmetic mean vector:

$$\mathbf{x}_i' = \mathbf{x}_i - \left(\frac{\sum_{i=1}^{N} \mathbf{x}_i}{N}\right).$$

Principal Component Analysis



Geometrical axes translation/rotation.

Artificial Intelligence & Information Analysis Lab

Principal Component Analysis

- Similarly, PCA can be performed on ψοωαριανψε matrix $\mathbf{C_X}$ of random vectors $\mathbf{X}$ belonging to data set $\mathcal{D}$:

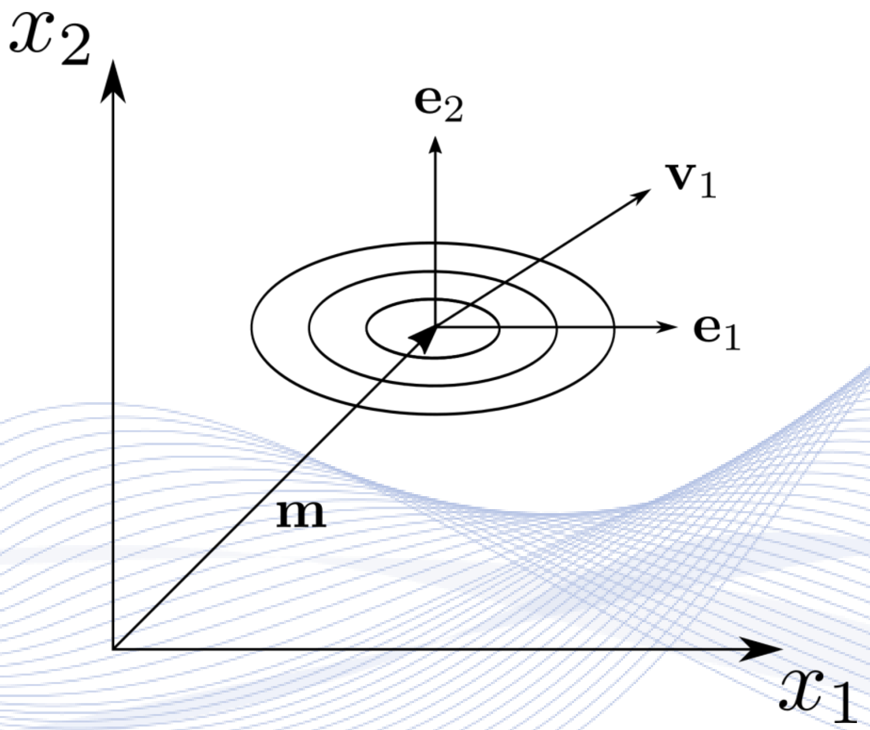
$$\mathbf{C_X} = E\{(\mathbf{X} - \mathbf{m_X})(\mathbf{X} - \mathbf{m_X})^T\}.$$

- As:

$$\mathbf{R_X} = \mathbf{C_X} + \mathbf{m_X}\,\mathbf{m_X}^T,$$

a large expected vector $\mathbf{m_X}$ of random vector $\mathbf{X}$ may dominate $\mathbf{R_X}$, hence greatly influencing its eigenanalysis.

Principal Component Analysis

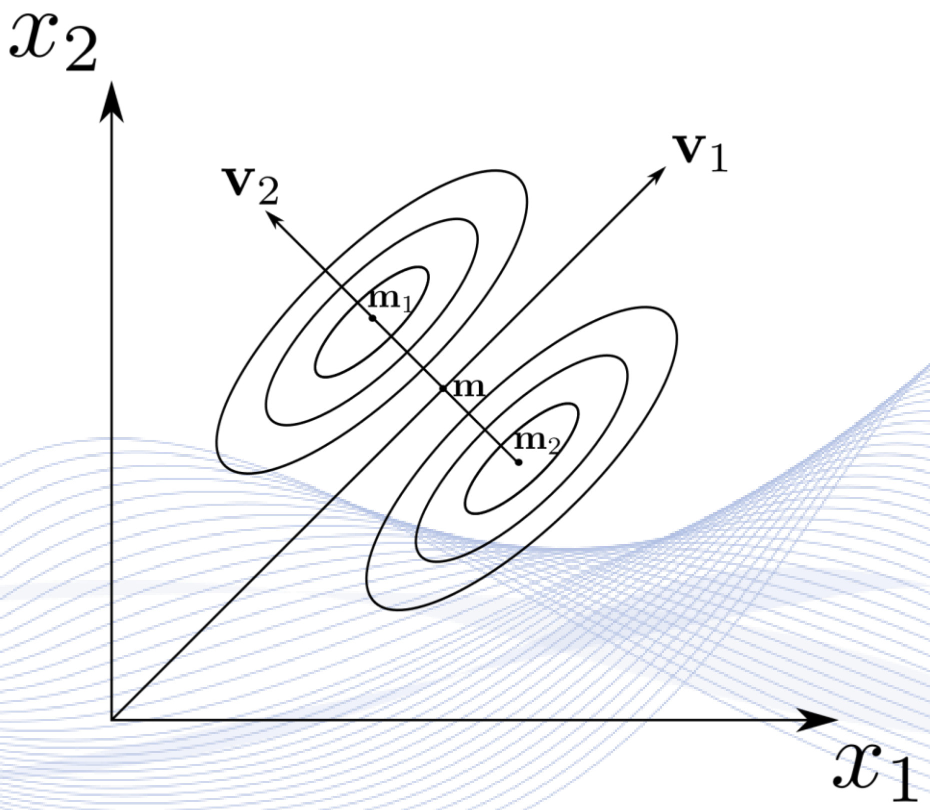


Influence of expected (mean) vectors on PCA.

Principal Component Analysis

- PCA does not employ class information.
- Efficient representation does not mean efficient classification between two classes!
- Eigenanalysis does not necessarily result in discriminant data representation.

Principal Component Analysis



Discriminant power of principal components.

Linear Discriminant Analysis

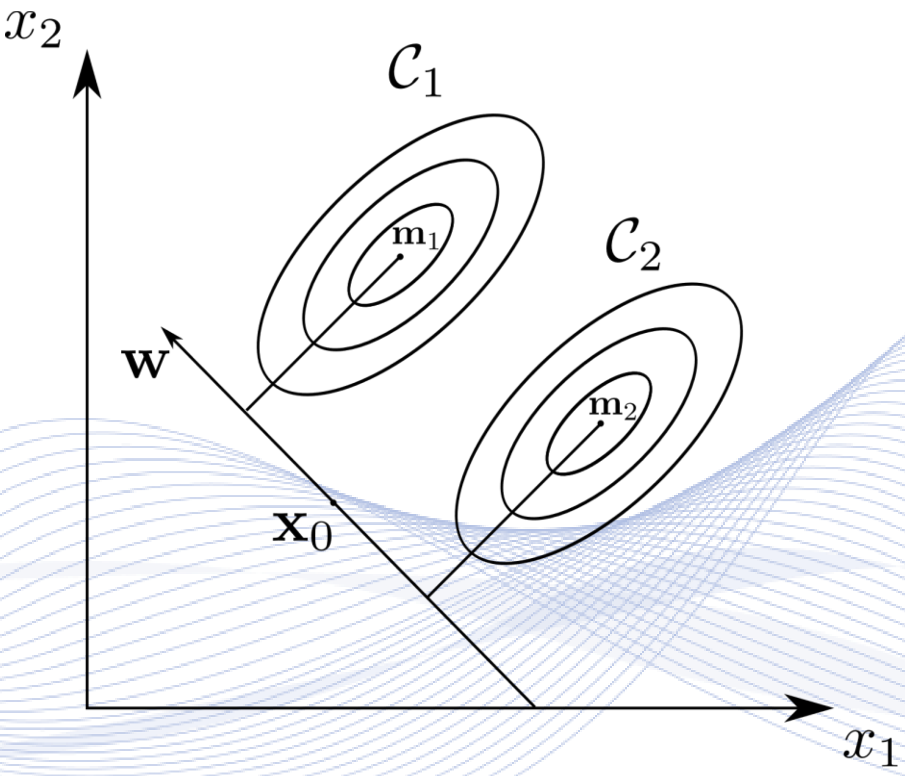
Linear Discriminant Analysis (***LDA***):

- Let data points $\mathbf{x} \in \mathbb{R}^n$ belong to two classes $\mathcal{C}_1$ and $\mathcal{C}_2$.
- LDA tries to find an optimal projection axis $\mathbf{w} \in \mathbb{R}^n$ that best separates the two classes.

- A data vector $\mathbf{x} \in \mathbb{R}^n$ is projected on projection axis $\mathbf{w}$ as follows:

$$\hat{x} = \mathbf{w}^T \mathbf{x}.$$

Linear Discriminant Analysis



LDA projection axis.

Linear Discriminant Analysis

- Fisher criterion becomes equivalent to maximizing ***Rayleigh quotient***:

$$r = \frac{\mathbf{w}^T \mathbf{S}_b \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w \mathbf{w}}.$$

- The optimal direction $\mathbf{w}$ given by generalized eigenanalysis:

$$\mathbf{S}_b \mathbf{w} = \lambda\, \mathbf{S}_w \mathbf{w},$$

- λ: the largest eigenvalue of matrix $\mathbf{S}_w^{-1} \mathbf{S}_b$.

Non-negative matrix factorization

- Data matrix $\mathbf{X}$ is an $n \times N$ matrix containing N data vectors $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$.
- It can be decomposed in a product of $n \times p$ and $p \times N$ matrices $\mathbf{F}$ and $\mathbf{H}$, respectively:

$$\mathbf{X} = \mathbf{FH}.$$

- p is smaller than N and n.

- All elements of matrices $\mathbf{F}, \mathbf{H}$ should be positive: $f_{ij} \geq 0$, $h_{kl} \geq 0$.

Non-negative matrix factorization

- Columns $\mathbf{f}_l, l = 1, \ldots, d$ of $\mathbf{F}$ are ***basis data vectors***.
- If $d \ll \min(n, N)$, we have dimensionality reduction.

- Original data vectors $\mathbf{x}_i, i = 1, \ldots, N$ can be reconstructed using only additive combinations of the resulting basis images:

$$\mathbf{x}_i = \sum_{l=1}^{d} h_{li} \mathbf{f}_l.$$

- Combination weights: coefficients in $\mathbf{H}$.
- Consistent with the psychological intuition regarding the objects representation in the human brain (i.e. combining parts to form the whole).

Non-negative matrix factorization

$\mathbf{x}_i =$  $\approx h_{1i}$  $+h_{2i}$  $+\cdots h_{li} +$  $+h_{pi}$ 

Data decomposition in NMF.

Multidimensional Scaling

Multidimensional scaling (***MDS***) is dimensionality reduction method, while preserving data dissimilarities (distances).

- Input: a data $\mathbf{x} \in \mathbb{R}^n$ dissimilarity matrix.
- Output: typically, it is a two-dimensional scatterplot.
- MDS applications:
  - Dimensionality reduction
  - Data visualization
  - Pattern recognition
  - Feature Extraction.

Multidimensional Scaling

- The dissimilarity type determines the MDS type:
  - Classical MDS.
  - Metric MDS.
  - Non-metric MDS.

Classical MDS

Classical MDS (cMDS):

- Consider a dissimilarity (distance) $N \times N$ matrix:

$$\mathbf{D} = [d_{ij}], \quad d_{ij} = \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2, \qquad \mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{R}^n.$$

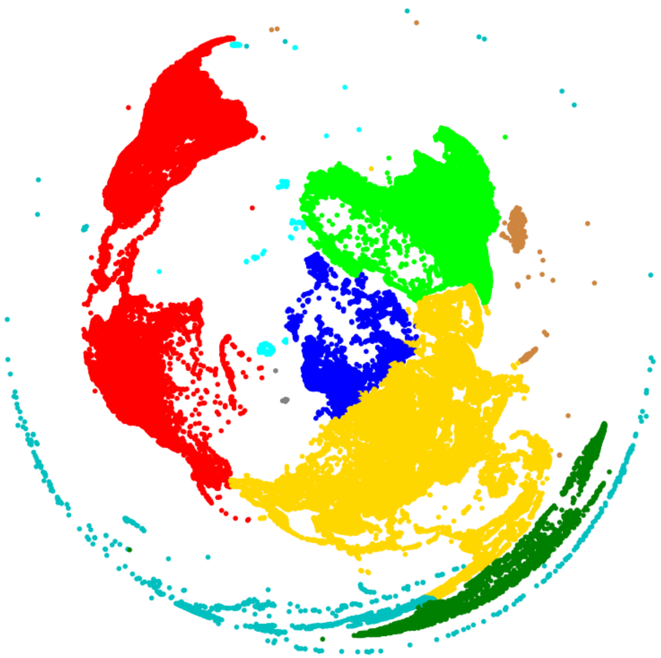
- cMDS seeks to find a mapping $\mathbf{x} \in \mathbb{R}^n \to \hat{\mathbf{x}} \in \mathbb{R}^d$ $(d \ll n)$, so that:

$$\hat{d}_{ij} = \left\|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j\right\|_2 \approx d_{ij}.$$

- Optimization problem to minimize function:

$$\min_{\hat{\mathbf{x}}_i, i=1,\dots N} \sum_{i<j} (d_{ij} - s\hat{d}_{ij})^2.$$

MDS application in cartography



Cartography using MDS and Harvesine distance.

MDS Summary

- If Euclidean data distances are used, classical MDS is convenient.
- For other dissimilarity types, iterative algorithms are more flexible as they allow optimal data re-scaling.
- They begin by a starting configuration and then modify it iteratively by reducing a stress function.

Dimensionality reduction

- Principal Component Analysis
- **Data Compression**
- Linear Discriminant Analysis
- Multidimensional Scaling

Data compression

Eigenanalysis for data compression.

- Data matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$, $\mathbf{x}_i \in \mathbb{R}^n$ has dimensions $n \times N$.
- Each data matrix column is a data vector.
- Matrix $\mathbf{X}\mathbf{X}^T$ is square and has dimensions $n \times n$.
- Matrix $\mathbf{X}^T\mathbf{X}$ is square and has dimensions $N \times N$.
  - $\mathbf{X}^T\mathbf{X}$ can be used for data compression!

SVD Data Compression

Singular Value Decomposition (***SVD***) for data compression.

- Data matrix $\mathbf{X}= [\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N]$, $\mathbf{x}_i \in \mathbb{R}^n$ has dimensions $n \times N$.
- Each data matrix column is a data vector.
- Matrix $\mathbf{X}$ has rank r ($r \leq \min\{n, N\}$).
- As typically, $n \ll N$, rank of matrix $\mathbf{X}$ satisfies $r \leq n$.

- Matrix $\mathbf{X}^T\mathbf{X}$ is square and has dimensions $N \times N$.

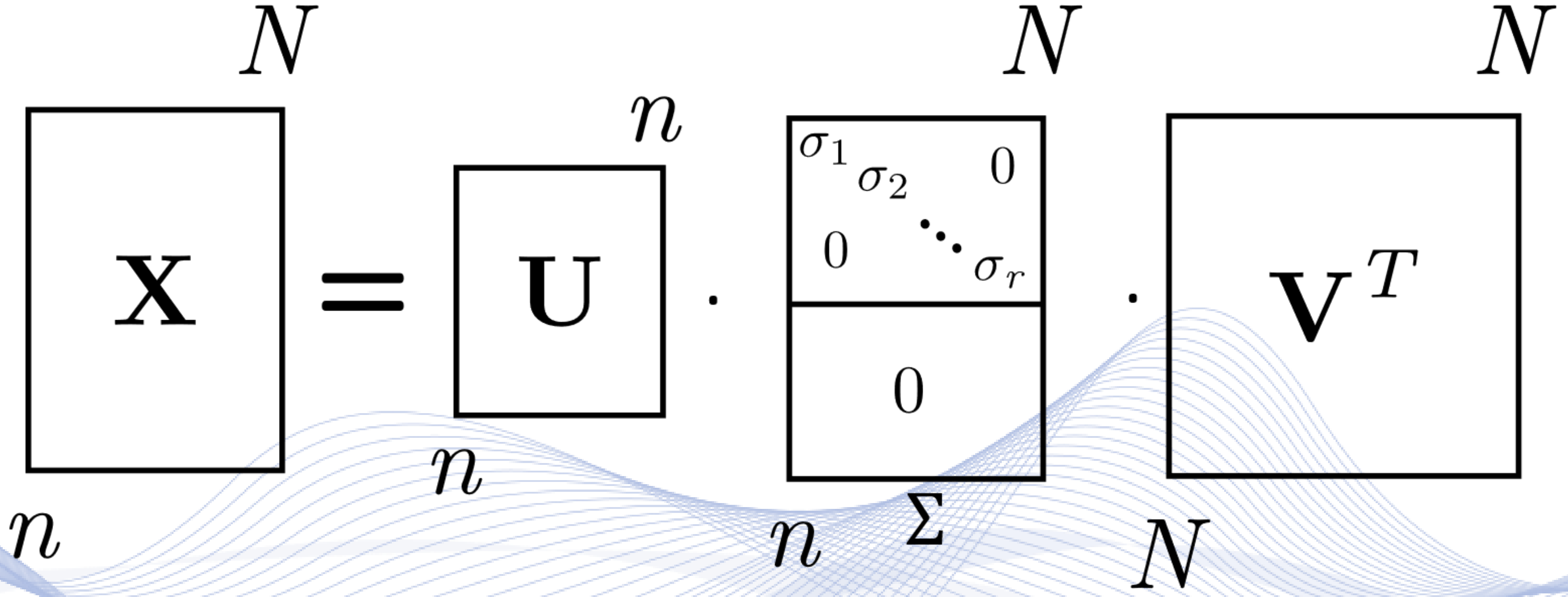
SVD Data Compression

SVD of data matrix $\mathbf{X}$:

$$\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_r] \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_r \end{bmatrix} \begin{bmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \\ \vdots \\ \mathbf{v}_r^T \end{bmatrix},$$

- $\boldsymbol{\Sigma}$ is a $n \times N$ matrix, whose r diagonal elements are the singular values $\sigma_1 \geq \sigma_2 \geq, \dots, \sigma_r \geq 0$ of $\mathbf{X}$.
- Vectors $\mathbf{u}_i, i = 1, \dots, n$, $\mathbf{v}_j$, $j = 1, \dots, N$ have dimensionality n, N respectively.

SVD Data Compression



SVD of a data matrix.

Vector Quantization

- A data set $\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, $\mathbf{x}_i \in \mathbb{R}^n$ is to be clustered (partitioned).
- Desired cluster number $m \ll N$.
- Distance measure $d(\mathbf{x}, \mathbf{y})$ between two vectors $\mathbf{x}, \mathbf{y}$.
- Calculation of cluster centers.
- Sorting algorithm to decide vector proximity.

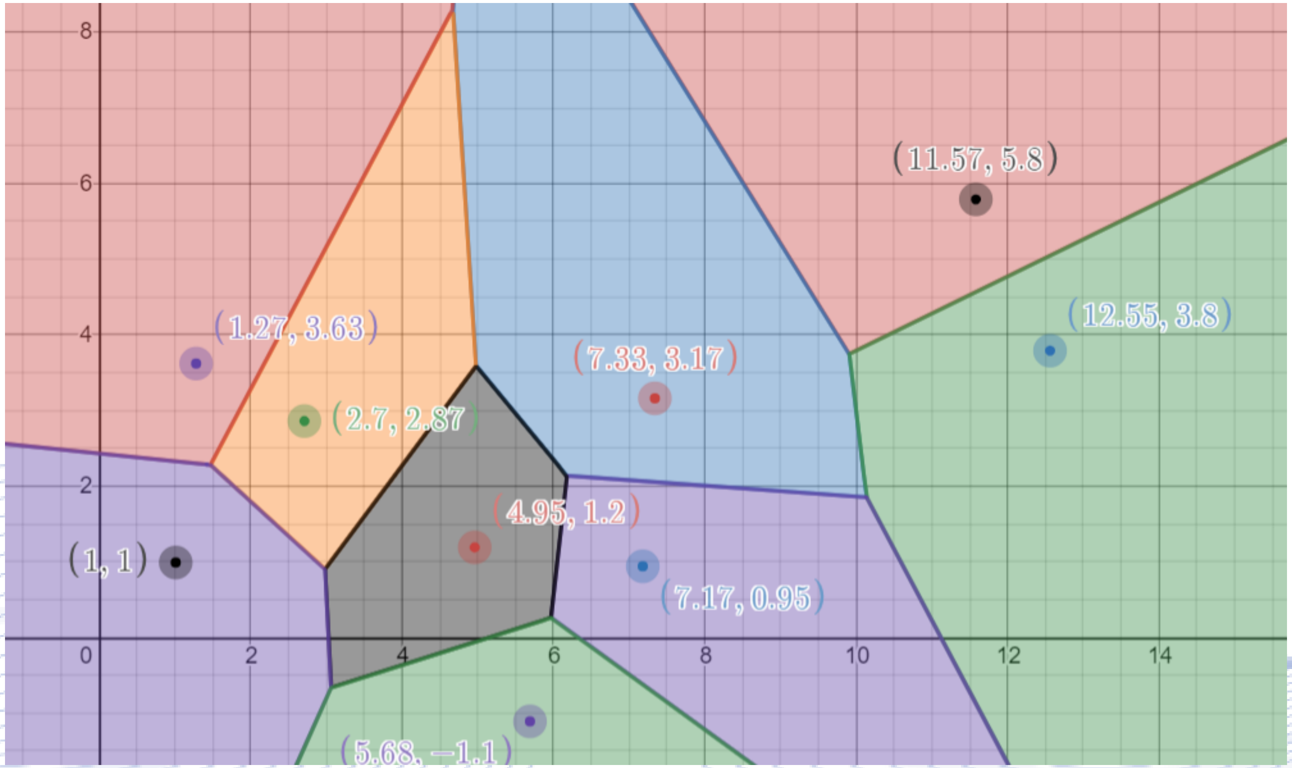
Vector Quantization

- Data vectors are partitioned in m clusters $\{\mathcal{C}_i, i = 1, \dots, m\}$.
- Mapping: $\mathbf{m} = \boldsymbol{Q}(\mathbf{x})$.
- $\mathbb{R}^{\boldsymbol{n}}$ is partitioned in m Voronoi regions (one per cluster).
- Each Voronoi region (cell) $\mathcal{R}_i$ is represented by $\mathbf{m}_i \in \mathbb{R}^n$, $i = 1, \dots, m$:

$$|\mathbf{x} - \mathbf{m}_i| < |\mathbf{x} - \mathbf{m}_j|, \quad i \neq j.$$

- Cluster $\mathcal{C}_i, i = 1, \dots, m$ vectors reside in $\mathcal{R}_i$.
- Voronoi cells may have regular structure.

Vector Quantization



Voronoi regions and clusters in $\mathbb{R}^2$.

Q & A

Thank you very much for your attention!

Contact: Prof. I. Pitas
pitas@csd.auth.gr