Machine learning in computer vision

Lesson 2

2 approaches

Feature selection:

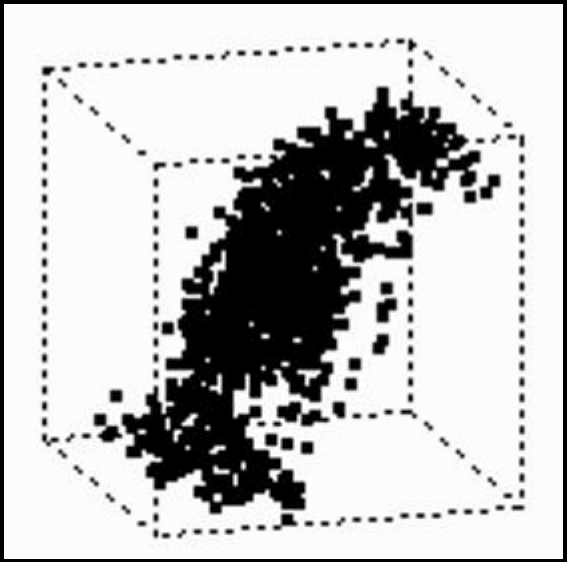
subset of original features

Feature transformation:

transformation of the original features to less-dimensional space

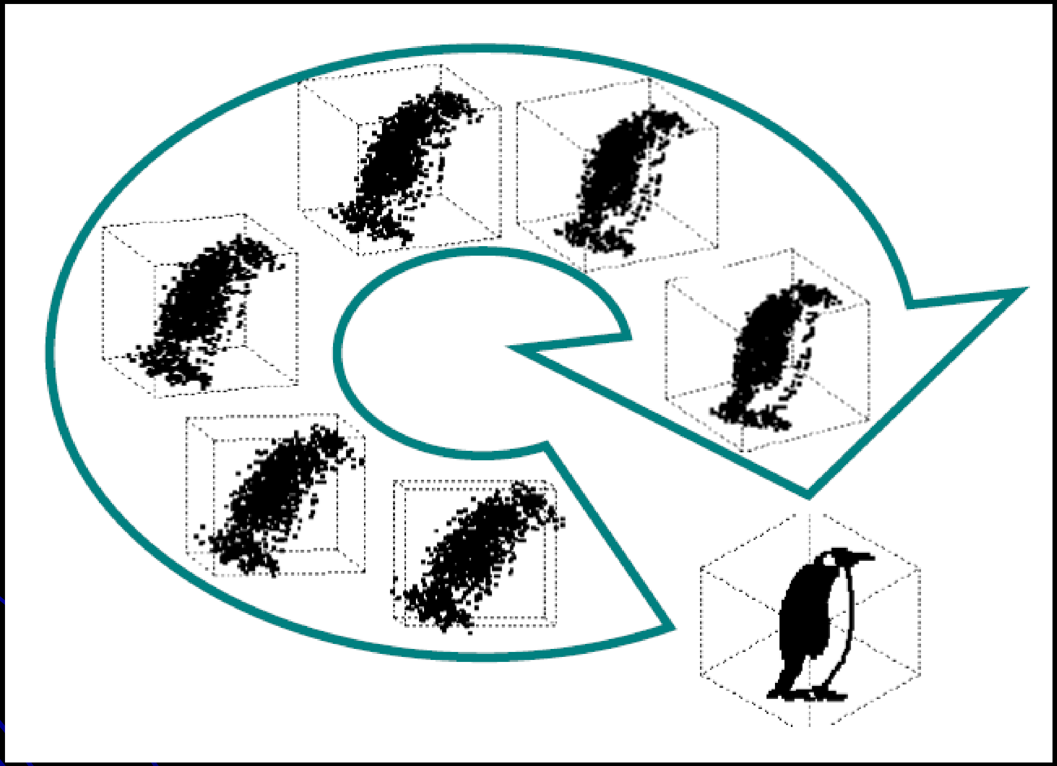
Transformation to less-dimensional space

Data in 3D



How to project to 2D?

Transformation
to less-dimensional space



Feature transformation

Unsupervised (information loss is minimized)

- Principal Component Analysis (PCA)
- Latent Semantic Indexing (LSI)
- Independent Component Analysis (ICA)
- …

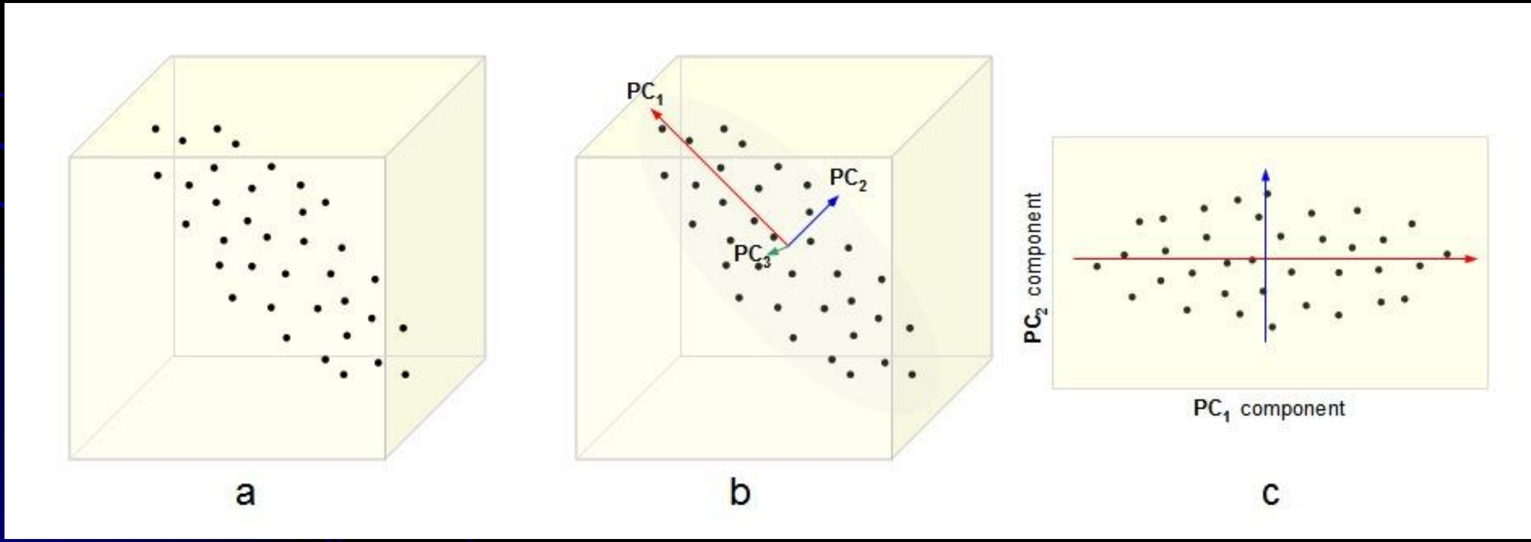
Supervised (interclass distance is maximized)

- Linear Discriminant Analysis (LDA)
- Canonical Correlation Analysis (CCA)
- Partial Least Squares (PLS)
- …

Principal Component Analysis (PCA)

Karhunen-Loeve, K-L method

PCA - looking for a subspace with highest variance

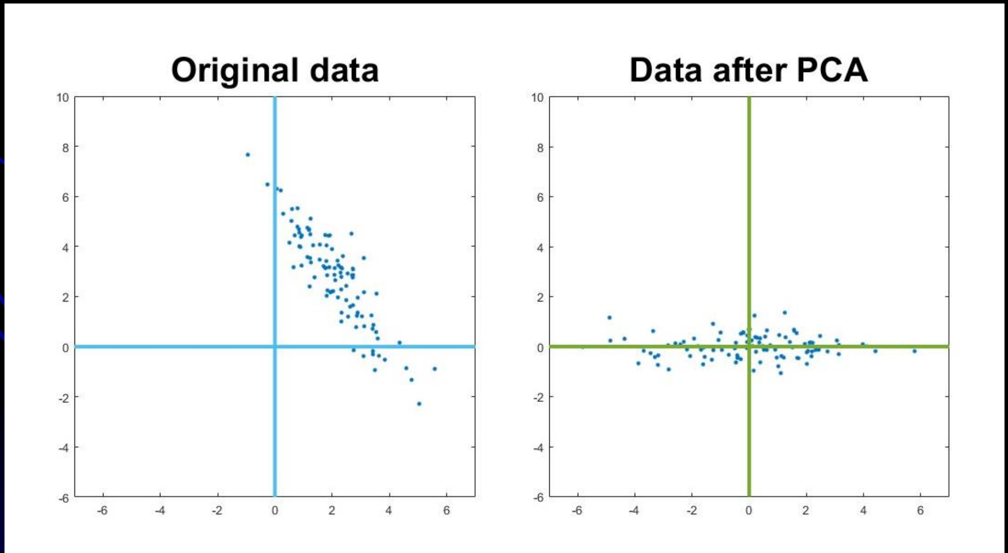


PCA

Rotates and translates the axes s.t. the first new axis is in the direction of maximum variance in the data

D-dimensional feature vectors: $\{\mathrm{x}_1, \ldots \mathrm{x}_N\}$

New orthonormal basis: $\{\mathrm{b}_1, \ldots \mathrm{b}_D\}$, $\mathrm{b}_i^{\mathrm{T}} \mathrm{b}_j = \delta_{ij}$



PC1 derivation

Vector $\mathbf{x}_i$ projection $x'_{i1} = \mathbf{b}_1^T \mathbf{x}_i$

Mean $\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$ projection $\bar{x}'_1 = \mathbf{b}_1^T \bar{\mathbf{x}}$

Variance $Var_1 = \frac{1}{N}\sum_{i=1}^{N}(x'_{i1} - \bar{x}'_1)^2$

$$= \frac{1}{N}\sum_{i=1}^{N}(\mathbf{b}_1^T \mathbf{x}_i - \mathbf{b}_1^T \bar{\mathbf{x}})^2$$

$$= \frac{1}{N}\sum_{i=1}^{N} \left(\mathbf{b}_1^T (\mathbf{x}_i - \bar{\mathbf{x}})\right)^2$$

$$= \mathbf{b}_1^T \boldsymbol{\Sigma} \mathbf{b}_1$$

First axis in the direction of the highest variance $\rightarrow$ constrained optimization

Lagrange multipliers

constrained optimization $$\max_{\mathbf{b}_1} \mathbf{b}_1^T \boldsymbol{\Sigma} \mathbf{b}_1 \,, \quad s.t.\, \mathbf{b}_1^T \mathbf{b}_1 = 1$$

Lagrange function optimization

$$L = \mathbf{b}_1^T \boldsymbol{\Sigma} \mathbf{b}_1 - \lambda(\mathbf{b}_1^T \mathbf{b}_1 - 1)$$

$$\frac{\partial L}{\partial \mathbf{b}_1} = 2\boldsymbol{\Sigma}\mathbf{b}_1 - 2\lambda \mathbf{b}_1 \equiv 0$$

$$\boldsymbol{\Sigma}\mathbf{b}_1 = \lambda \mathbf{b}_1$$ eigenvalue

Variance $Var_1 = \mathbf{b}_1^T \boldsymbol{\Sigma} \mathbf{b}_1 = \lambda \mathbf{b}_1^T \mathbf{b}_1 = \lambda$

PC2 derivation

Vector $\mathbf{x}_i$ projection $x'_{i2} = \mathbf{b}_2^T \mathbf{x}_i$

Mean $\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$ projection $\bar{x}'_2 = \mathbf{b}_2^T \bar{\mathbf{x}}$

Variance $Var_2 = \frac{1}{N}\sum_{i=1}^{N}(x'_{i2} - \bar{x}'_2)^2$

$$= \frac{1}{N}\sum_{i=1}^{N}(\mathbf{b}_2^T \mathbf{x}_i - \mathbf{b}_2^T \bar{\mathbf{x}})^2$$

$$= \frac{1}{N}\sum_{i=1}^{N} (\mathbf{b}_2^T (\mathbf{x}_i - \bar{\mathbf{x}}))^2$$

$$= \mathbf{b}_2^T \mathbf{\Sigma} \mathbf{b}_2$$

Second axis in the direction of the highest variance $\rightarrow$ constrained optimization

PC2 derivation

constrained optimization

$$\max_{\mathbf{b_2}} \mathbf{b}_2^T \mathbf{\Sigma} \mathbf{b}_2 \,,$$

$$s.t. \begin{array}{l} \mathbf{b}_2^T \mathbf{b}_2 = 1 \\ \mathbf{b}_1^T \mathbf{b}_2 = 0 \end{array}$$

Lagrange function optimization

$$L = \mathbf{b}_2^T \mathbf{\Sigma} \mathbf{b_2} - \lambda(\mathbf{b}_2^T \mathbf{b_2} - 1) - \mu(\mathbf{b}_1^T \mathbf{b}_2)$$

$$\frac{\partial L}{\partial \mathbf{b_2}} = 2\mathbf{\Sigma}\mathbf{b_2} - 2\lambda\mathbf{b_2} \equiv 0$$

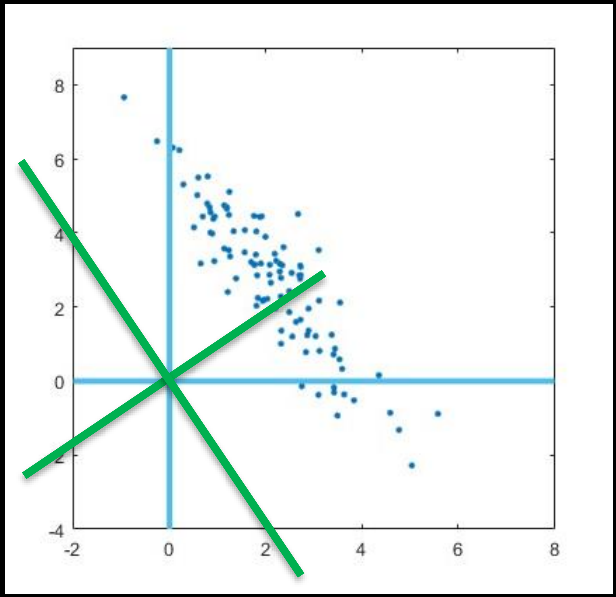
$$\mathbf{\Sigma}\mathbf{b_2} = \lambda\mathbf{b_2}$$ eigenvalue

Variance $Var_2 = \mathbf{b}_2^T \mathbf{\Sigma} \mathbf{b}_2 = \lambda \mathbf{b}_2^T \mathbf{b}_2 = \lambda$

…

Defining the new space

We have directions of new basis vectors $\{\mathbf{b}_1, \ldots \mathbf{b}_D\}$



Defining the new space

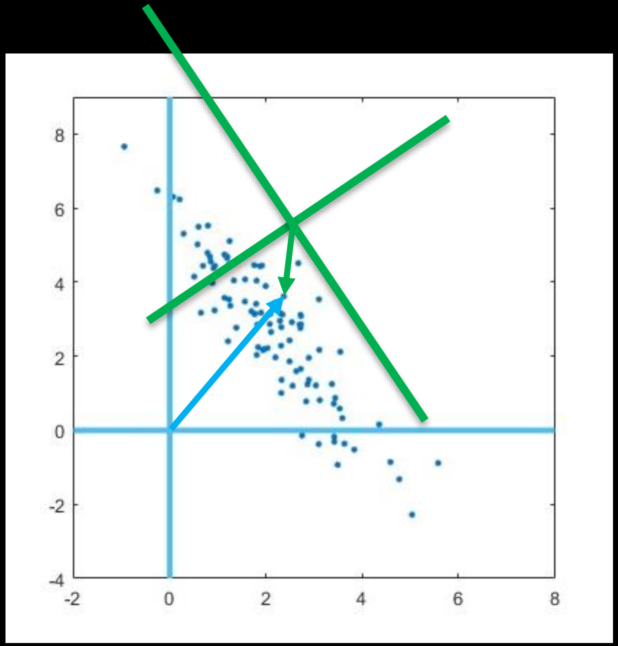
We need the new origin: **p**

Vector in original space $\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_D \end{bmatrix}$

Vector in transformed space $\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_D \end{bmatrix}$

We want to minimize the error between original and transformed vectors

$$\mathbf{y} = \mathbf{p} + \sum_{j=1}^{D} y_j \mathbf{b}_j$$



Defining the new space

The error between original and projected vectors (in the original space)

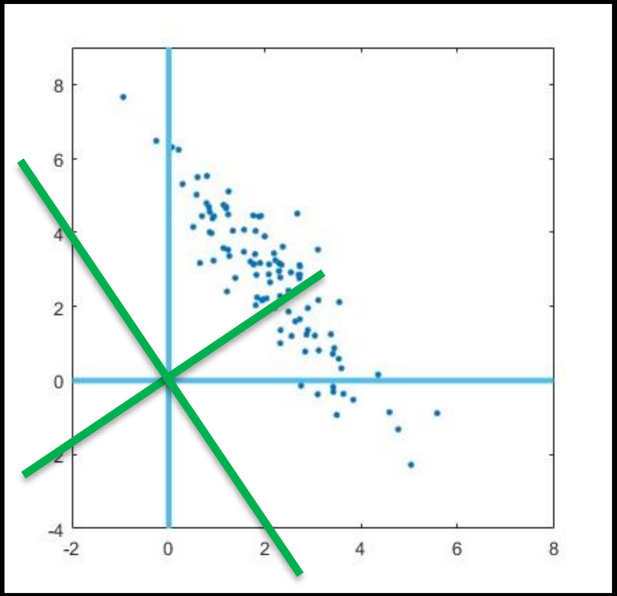
$$E = \sum_{i=1}^{N} \left\| \mathbf{x}_i - \left(\mathbf{p} + \sum_{j=1}^{D} y_{ij}\mathbf{b}_j\right) \right\|^2$$

$$= \sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{p}\|^2 - 2\sum_{i=1}^{N} y_{ij}\mathbf{b}_j^T(\mathbf{x}_i - \mathbf{p}) + \sum_{i=1}^{N}\sum_{j=1}^{D} y_{ij}^2$$

Solution

$$y_{ij} = \mathbf{b}_j^T(\mathbf{x}_i - \bar{\mathbf{x}})$$

$$\mathbf{p} = \bar{\mathbf{x}}$$



PCA steps

1. Compute covariance matrix

$$\mathbf{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \bar{\mathbf{x}})\,(\mathbf{x}_i - \bar{\mathbf{x}})^T = \frac{1}{N}\mathbf{X}\mathbf{X}^T,$$

$$[\mathbf{X}]_{\boldsymbol{D}\times\boldsymbol{N}} = [\mathbf{x}_1 - \bar{\mathbf{x}}, \ldots \mathbf{x}_D - \bar{\mathbf{x}}]$$

2. Compute eigenvectors of matrix

$$\mathbf{\Sigma}\mathbf{b}_j = \lambda_j \mathbf{b}_j$$

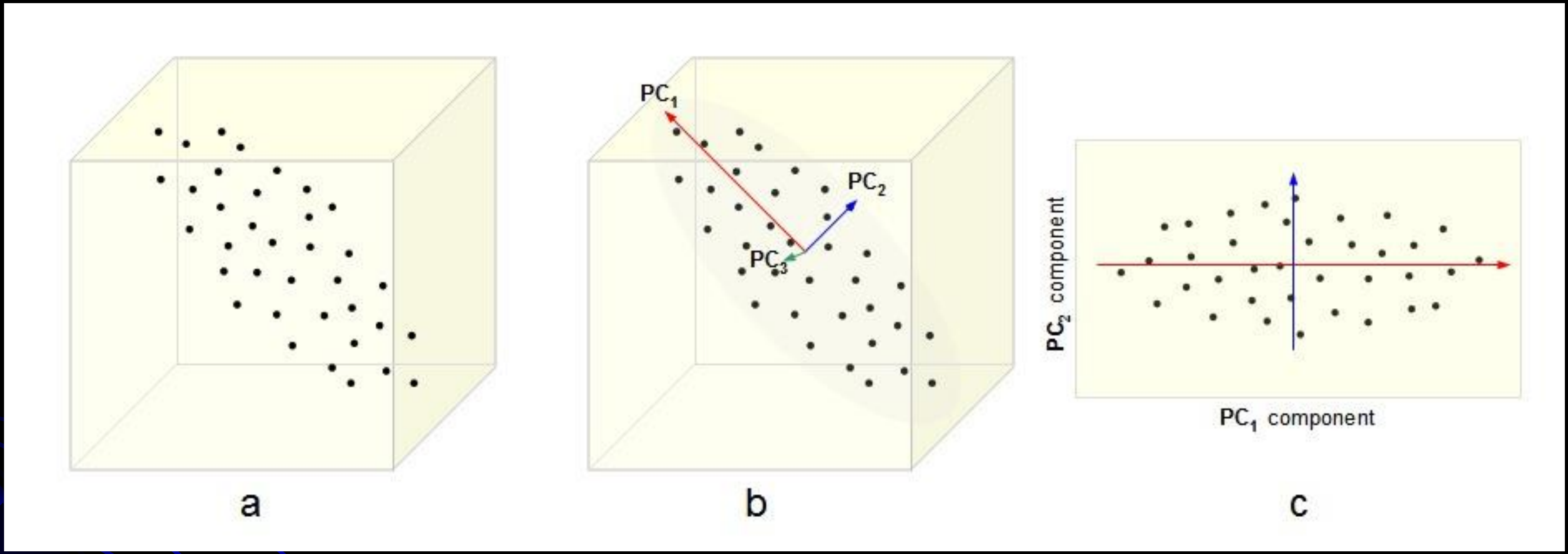
3. Compute coordinates of transformed vectors

$$\mathbf{x}_i' = \mathbf{B}^T(\mathbf{x}_i - \bar{\mathbf{x}})$$

$$\mathbf{X}' = \mathbf{B}^T\mathbf{X}$$

$$\mathbf{B}=[\mathbf{b}_1, \ldots \mathbf{b}_D]$$

Dimensionality reduction

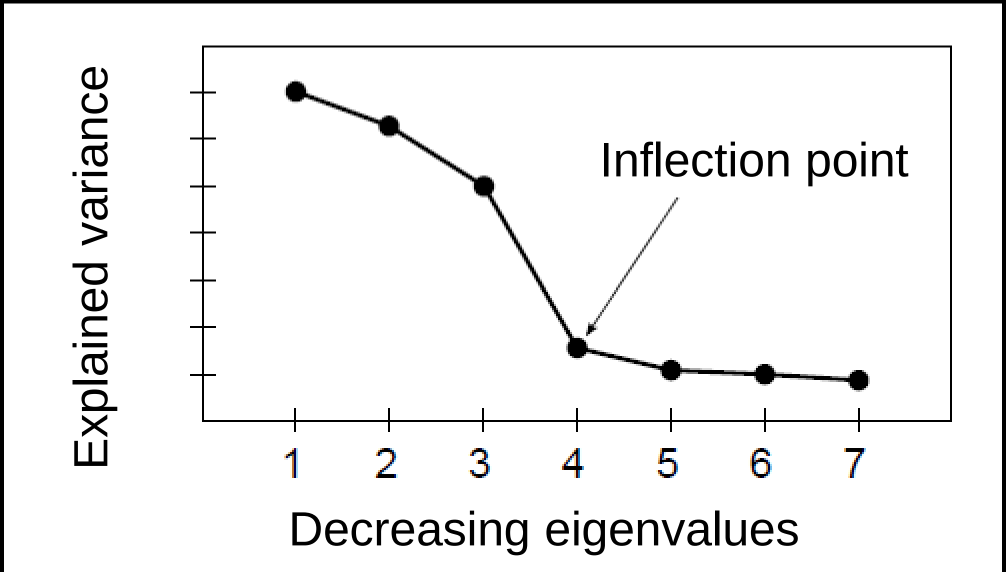


Number of principal components

Scree plot

Explained variance

$$\frac{\lambda_j}{\sum_{j=1}^{D} \lambda_j}$$



- Inflection point - where the “unimportant” eigenvalues start

here the optimal number is 3

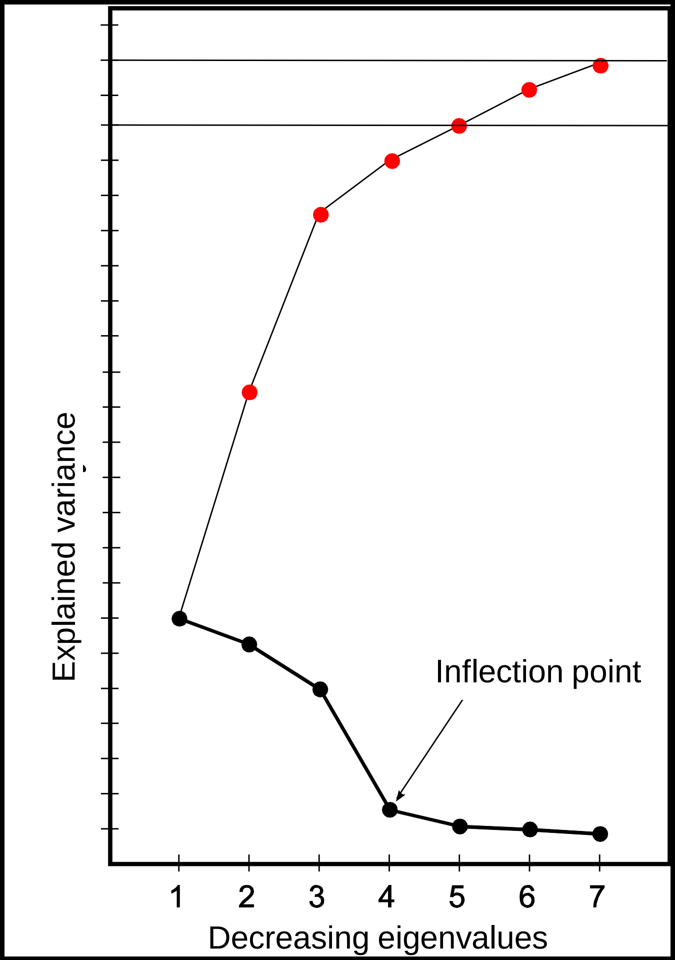
Number of principal components

Proportion of explained variance in j-th component

$$\frac{\lambda_j}{\sum_{j=1}^{D} \lambda_j}$$

Cumulative proportion of explained variance

$$\frac{\sum_{j=1}^{K} \lambda_j}{\sum_{j=1}^{D} \lambda_j} > 0.9 \text{ or } 0.95$$



PCA computation

$N < D$, $r = \mathrm{rank}(\mathbf{X}) = \mathrm{rank}(\boldsymbol{\Sigma}) \leq N$

At most r non-zero eigenvalues

$$\boldsymbol{\Sigma} = \frac{1}{N}\mathbf{X}\mathbf{X}^T$$

$D \times D$

$O(D^3)$

$$\begin{aligned}
\boldsymbol{\Sigma}\mathbf{b}_j &= \lambda_j \mathbf{b}_j \\
\frac{1}{N}\mathbf{X}\mathbf{X}^T\mathbf{b}_j &= \lambda_j \mathbf{b}_j \\
\frac{1}{N}\mathbf{X}^T\mathbf{X}\mathbf{X}^T\mathbf{b}_j &= \lambda_j \mathbf{X}^T\mathbf{b}_j \\
\frac{1}{N}\mathbf{X}^T\mathbf{X}\mathbf{p}_j &= \lambda_j \mathbf{p}_j,
\end{aligned}$$

$\mathbf{p}_j = \mathbf{X}^T\mathbf{b}_j$ - eigenvector of matrix $\frac{1}{N}\mathbf{X}^T\mathbf{X}$

PCA computation

$$\begin{aligned} \frac{1}{N}\mathbf{X}^T\mathbf{X}\mathbf{p}_j &= \lambda_j\mathbf{p}_j \\ \frac{1}{N}\mathbf{X}\mathbf{X}^T\mathbf{X}\mathbf{p}_j &= \lambda_j\mathbf{X}\mathbf{p}_j \\ \mathbf{\Sigma}\mathbf{X}\mathbf{p}_j &= \lambda_j\mathbf{X}\mathbf{p}_j \end{aligned} \Rightarrow \mathbf{b}_j \propto \mathbf{X}\mathbf{p}_j$$

Suppose $||\mathbf{p}_j|| = 1$

we look for $\mathbf{b}_j$, s.t. $||\mathbf{b}_j|| = 1$

$$\mathbf{b}_j = \frac{1}{\sqrt{N\lambda_j}}\,\mathbf{X}\mathbf{p}_j$$

SVD

$N < D$

singular value decomposition of A

$$\boxed{\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T}$$

$$[\mathbf{A}]_{N \times D} = [\mathbf{U}]_{N \times N} [\mathbf{S}]_{N \times D} [\mathbf{V}^T]_{D \times D}$$

$$\mathbf{U} = [\mathbf{u}_1, \ldots \mathbf{u}_N], \ \mathbf{V} = [\mathbf{v}_1, \ldots \mathbf{v}_D] \text{ a } \mathbf{S} = diag(\sigma_1, \ldots \sigma_{rank(\mathbf{A})})$$

v - eigenvector of $\mathbf{A}^T\mathbf{A}$ $\lambda = \sigma^2$

u - eigenvector of $\mathbf{A}\mathbf{A}^T$

$$\mathbf{\Sigma} = \frac{1}{N}\mathbf{X}\mathbf{X}^T$$

PCA - SVD connection

$$\mathbf{X} = [\mathbf{x}_1 - \bar{\mathbf{x}}, \ldots \mathbf{x}_N - \bar{\mathbf{x}}]$$

$$\mathbf{Y} = \frac{1}{\sqrt{N}} \mathbf{X}^T \longrightarrow \mathbf{Y}^T \mathbf{Y} = \frac{1}{N} \mathbf{X} \mathbf{X}^T = \mathbf{\Sigma}$$

$$\mathbf{Y} = \mathbf{U} \mathbf{S} \mathbf{V}^T$$

$\mathbf{V}$ - eigenvectors of matrix $\mathbf{Y}^T \mathbf{Y} = \mathbf{\Sigma}$

We can use SVD instead of PCA

SVD – numerically stable

Using PCA

$$\mathbf{x}' = \sum_{j=1}^{D} \mathbf{b}_j^T (\mathbf{x} - \overline{\mathbf{x}}) = \mathbf{B}^T \mathbf{x} = (x'_1, \ldots x'_D)^T$$

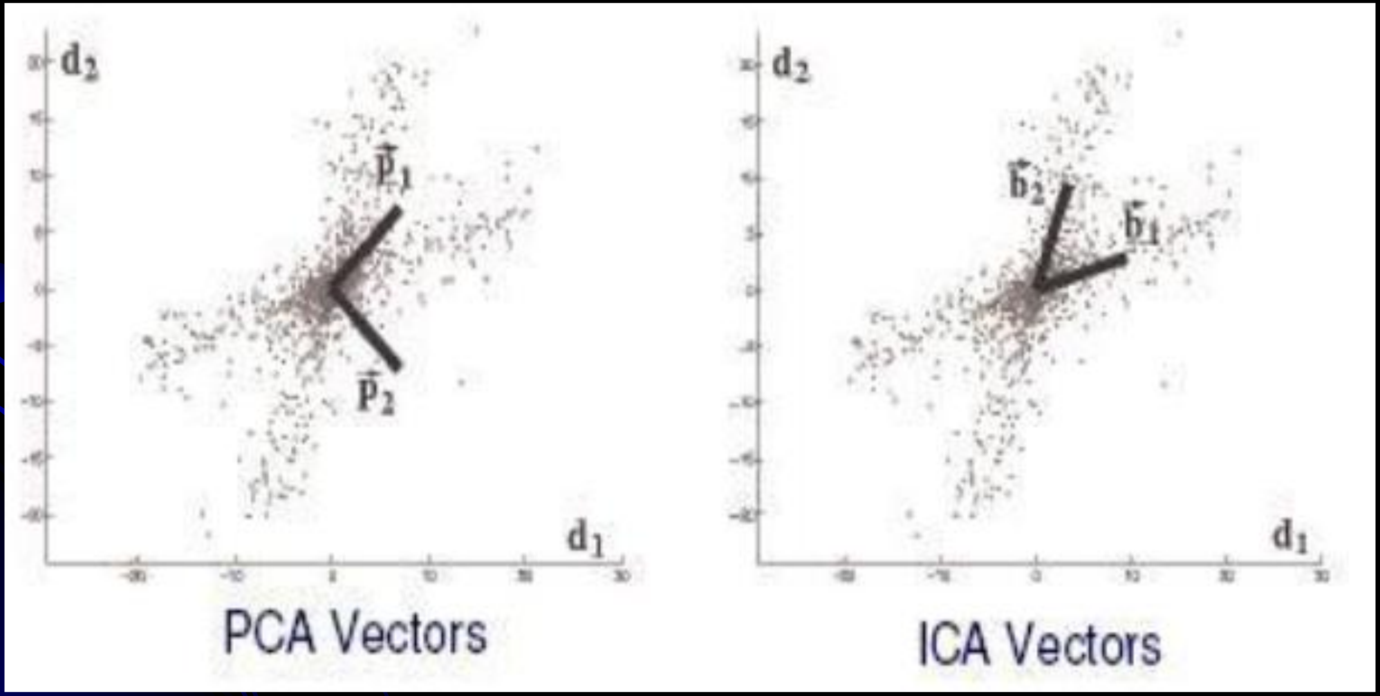
$$\mathbf{x} = \overline{\mathbf{x}} + \sum_{j=1}^{D} \mathbf{b}_j x'_j$$

Using K eigenvectors (eigenfaces)

$$\tilde{\mathbf{x}} \approx \overline{\mathbf{x}} + \sum_{j=1}^{K} \mathbf{b}_j x'_j$$

Independent Components Analysis (ICA)

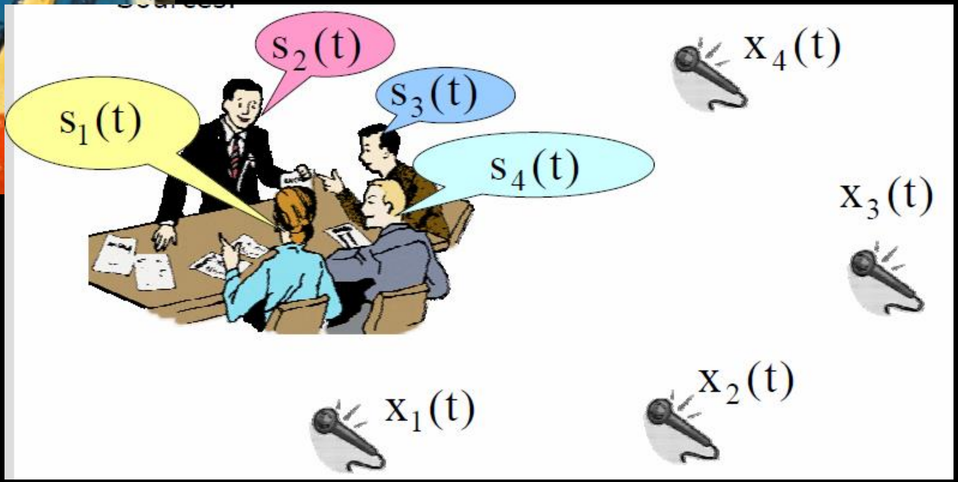
Components not orthogonal



ICA

vector represented as linear combination of non-Gaussian random variables ("independent components")

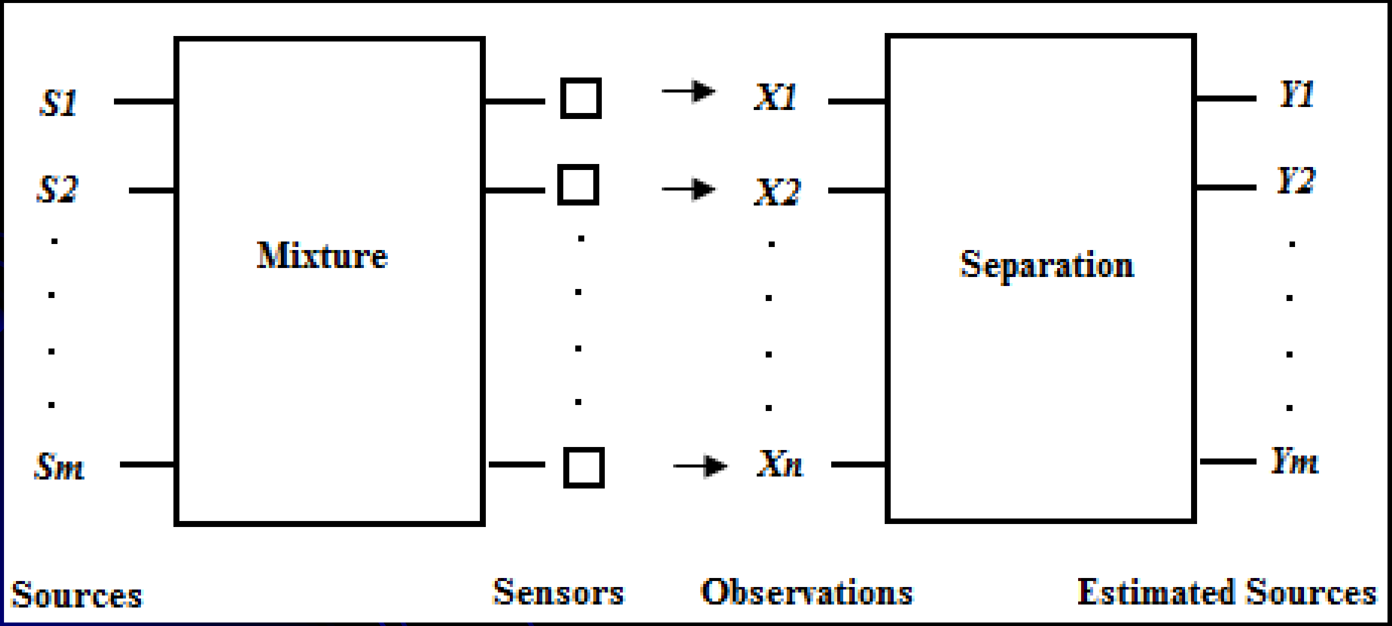
Cocktail party problem



$$x_i(t) = a_{i1}*s_1(t) + a_{i2}*s_2(t) + a_{i3}*s_3(t) + a_{i4}*s_4(t)$$

ICA

$$\mathbf{X} = \mathbf{A}\,\mathbf{S}$$
$$\mathbf{Y} = \mathbf{W}\,\tilde{\mathbf{X}}$$



ICA assumptions

$$p(s_1, s_2, \ldots, s_n) = p(s_1)p(s_2)\ldots p(s_n)$$

$E(s_i)=0$

$Var(s_i)=1$

non-Gaussianity

$E\{SS^T\}=I$

ICA procedure

Preprocessing:

Centering $\mathbf{X}' = \mathbf{X} - \bar{\mathbf{X}}$

Whitening $\tilde{\mathbf{X}} = \mathbf{B}\,\mathbf{X}'$, s.t. $\boldsymbol{\Sigma}_{\tilde{\mathbf{X}}} = \mathbf{I}$

Eigenvalues: $\mathbf{X}'\mathbf{X}'^{T} = \boldsymbol{\Sigma} = \mathbf{V}\boldsymbol{S}\mathbf{V}^{T}$

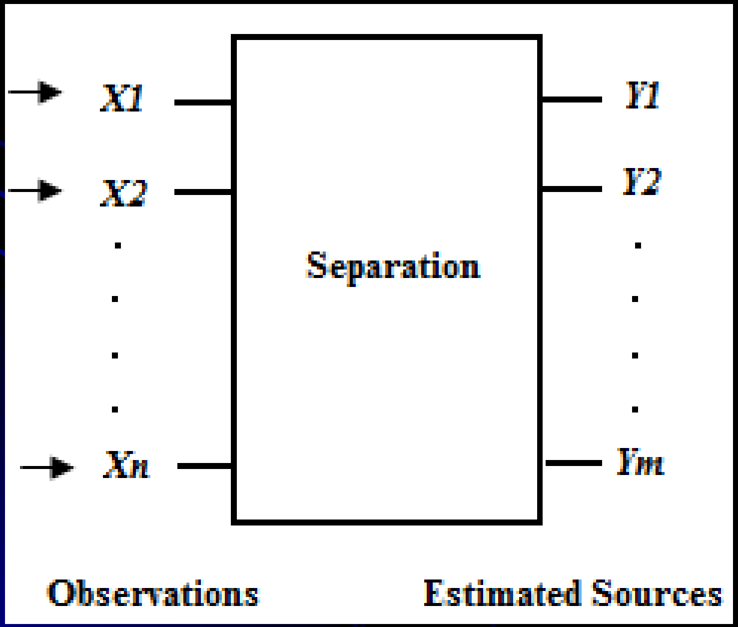
$$d_{ij} = s_{ij}^{-1/2}$$

$$\tilde{\mathbf{X}} = \mathbf{V}\mathbf{D}\mathbf{V}^{T}\,\mathbf{X}'$$

ICA procedure

Looking for directions w_i, to maximize non-Gaussianity

$$\mathbf{Y} = \mathbf{W}\,\tilde{\mathbf{X}}$$



Non-Gaussianity measures

skewness and kurtosis (3rd, 4th central moment)

Negentropy $J(y) = H(y_G) - H(y)$

aproximation $J(y) \propto \left[E\big(G(y)\big) - E(G(y_G))\right]^2$

$$G_1(u) = \frac{1}{a_1} \log\cosh a_1 u, \quad G_2(u) = -\exp(-u^2/2)$$

ICA algorithm

w that maximizes non-gaussianity

$$J(\mathrm{w}^T \mathrm{x}) \propto \left[E\left(G(\mathrm{w}^T \mathrm{x})\right) - E(G(y_G))\right]^2$$

constraint $\|w\|^2 = 1$

constant for w

Lagrange: L = E(G(w^Tx)) – λ(w^Tw – 1)

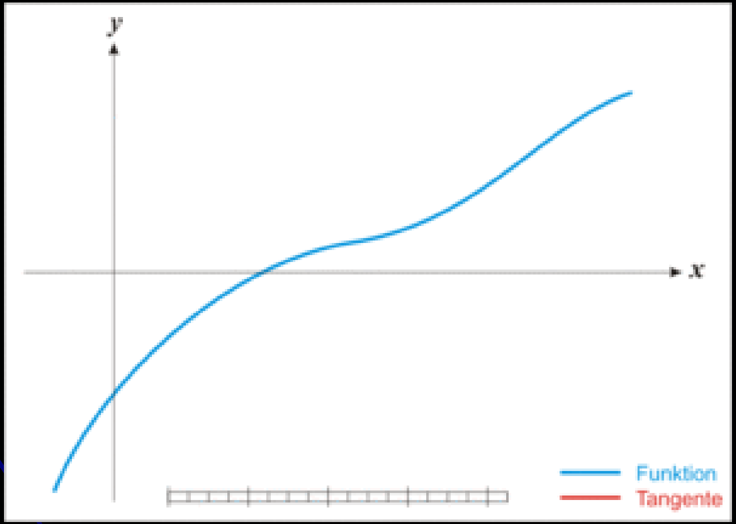
Derivation: L' = $\boxed{E(x.g(w^Tx)) - \lambda w}$ ≡ 0

solve using Newton method

f(w)

Newton method – root finding

$$\mathbf{w}_{i+1} = \mathbf{w}_i - \frac{f(\mathbf{w}_i)}{f'(\mathbf{w}_i)}$$



https://en.wikipedia.org/wiki/Newton%27s_method

FastICA algorithm, 1 direction

1. Random starting vector w
2. $\mathbf{w}^{+} = E\{\mathbf{x}g(\mathbf{w}^{T}\mathbf{x})\} - E\{g'(\mathbf{w}^{T}\mathbf{x})\}\mathbf{w}$
3. $\mathbf{w} = \mathbf{w}^{+}/\|\mathbf{w}^{+}\|$
4. Repeat 2.,3. until convergence

FastICA, more directions

FastICA for each direction, decorelation after each iteration

$$\mathbf{w}_{p+1} = \mathbf{w}_{p+1} - \sum_{j=1}^{p} \mathbf{w}_{p+1}^T \mathbf{w}_j \mathbf{w}_j$$
$$\mathbf{w}_{p+1} = \mathbf{w}_{p+1} / \sqrt{\mathbf{w}_{p+1}^T \mathbf{w}_{p+1}}$$

FastICA for all directions, symetric decorelation at the end

ICA ambiguites

Amplitudes of separated signals cannot be determined.

There is a sign ambiguity associated with separated signals.

The order of separated signals cannot be determined.

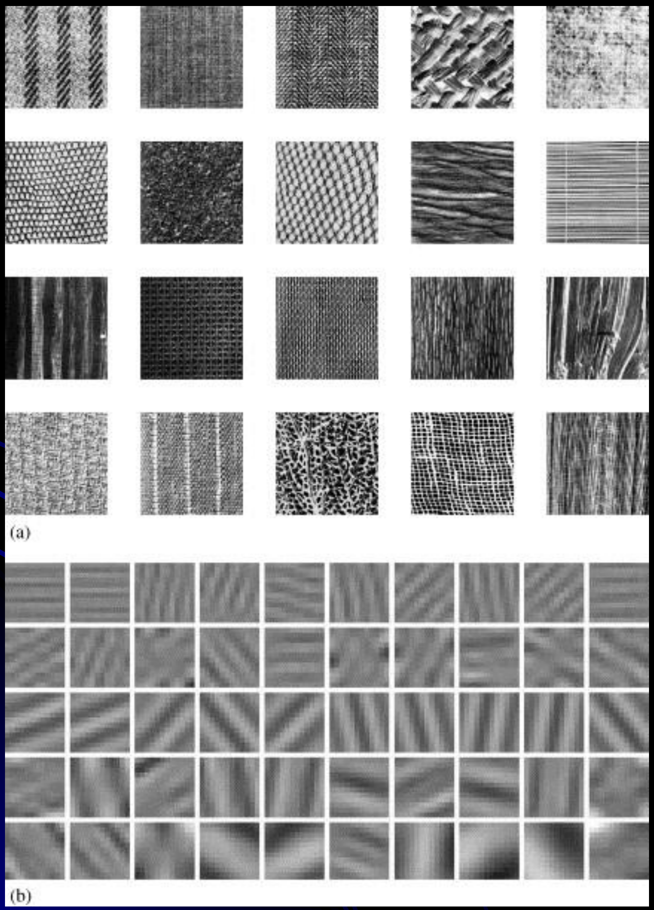
Reduction

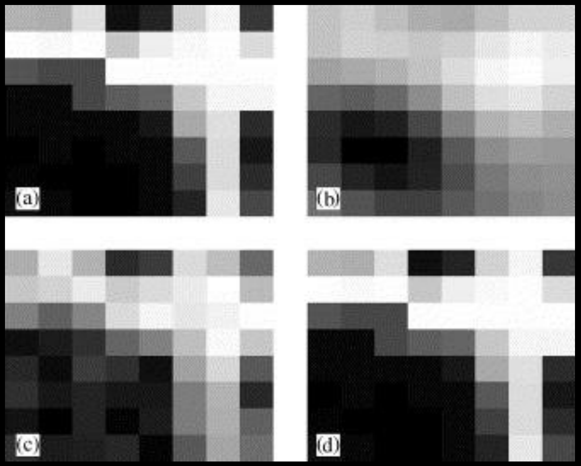
ICs not ranked

1. Compute ICA for $K<D$
2. During whitening we retain K PCs
3. Compute ICA for $K=D$, analyze the mixing matrix A

ICA applications

ICs





Reconstruction of original (8×8) image using ICA basis functions.

(a) Original image,

(b) using 10 basis functions, *NSE*≈0.4,

(c) using 30 basis functions, *NSE*≈0.1

(d) using 63 basis functions, *NSE*≈0.

R. Jenssen, T. Eltoft
Independent component analysis for texture segmentation
Pattern Recognit., 36 (10) (2003), pp. 2301-2315

Bibliography

Aapo Hyvärinen and Erkki Oja: Independent Component Analysis: Algorithms and Applications, *Neural Networks, 13(4-5):411-430, 2000*

http://research.ics.aalto.fi/ica/icademo/

Independent Component Analysis of Textures in Angiography Images

http://www.ia.pw.edu.pl/~wkasprza/PAP/ICCVG04c.pdf

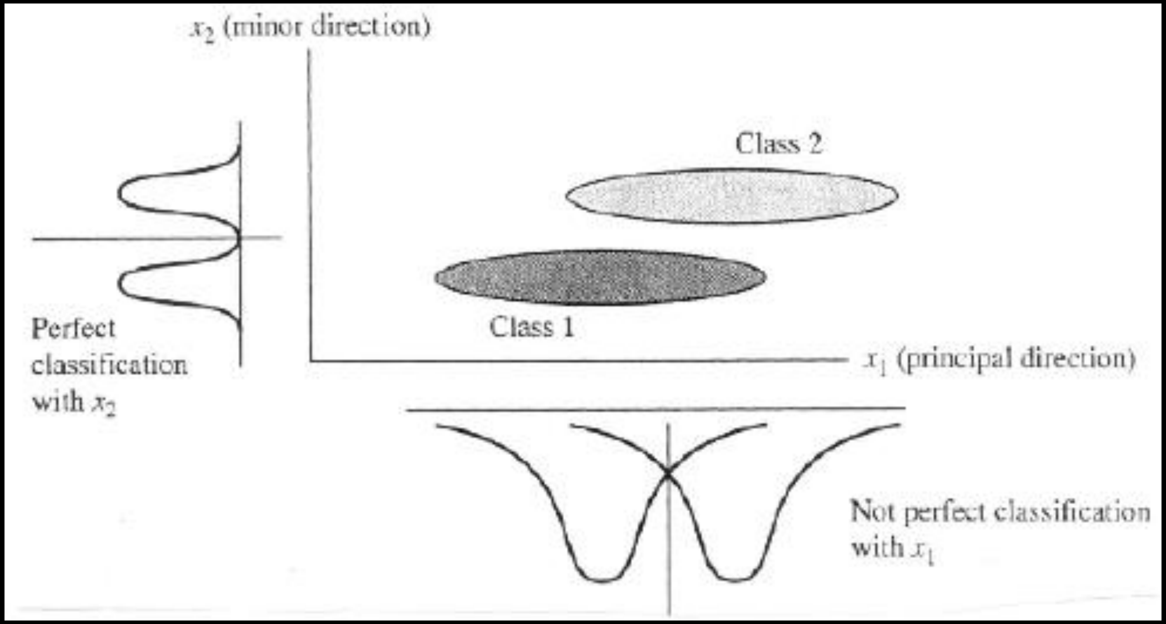
Independent Component Analysis of Textures

https://pdfs.semanticscholar.org/a734/1ead68514c5eb39cc2e907df62fd280e7f87.pdf

Cons of unsupervised methods

sometimes not optimal for classification task

do not take into account the class membership



Linear Discriminant Analysis (LDA)

- Supervised method
- Dimensionality reduction with class separability
- Investigates intraclass and interclass relations

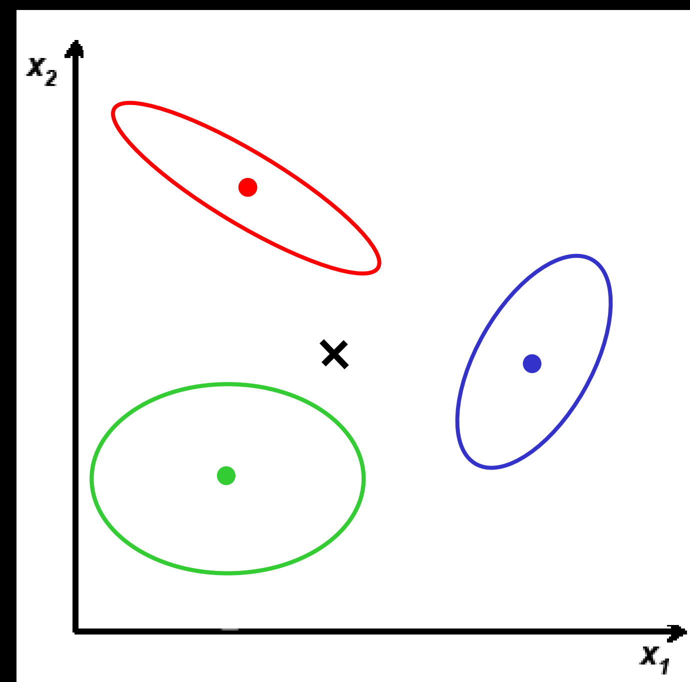
Fisher LDA

D-dimensional feature vectors: $\{\mathrm{x}_1, \ldots \mathrm{x}_N\}$

C classes: $|\omega_j| = N_j$

$$\bar{\mathbf{x}}_j = \frac{1}{|\omega_j|} \sum_{\mathbf{x} \in \omega_j} \mathbf{x} = \frac{1}{N_j} \sum_{\mathbf{x} \in \omega_j} \mathbf{x}$$

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i$$



Scatter

Total scatter in the data

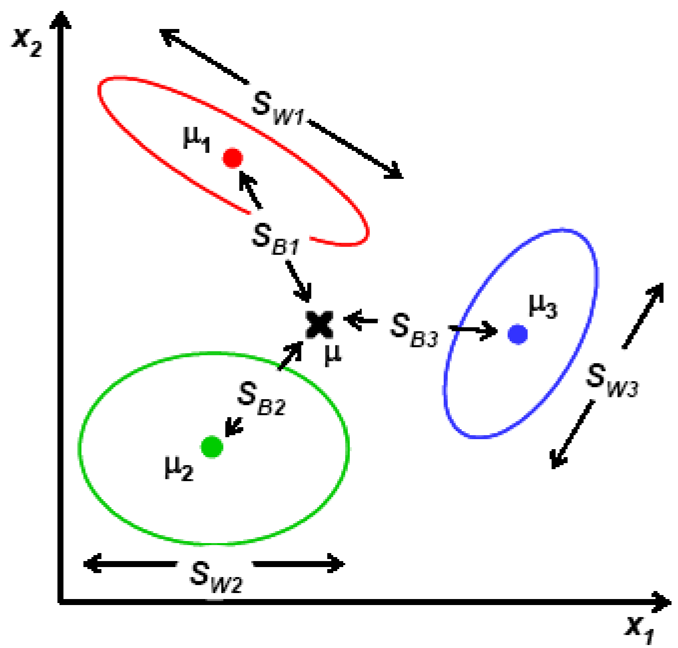
$$\mathbf{S} = \sum_{i=1}^{N} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T.$$

$$\mathbf{S} = \mathbf{S}_M + \mathbf{S}_V$$

Interclass scatter

$$\mathbf{S}_M = \sum_{j=1}^{C} N_j (\bar{\mathbf{x}}_\mathbf{j} - \bar{\mathbf{x}})(\bar{\mathbf{x}}_\mathbf{j} - \bar{\mathbf{x}})^T$$

$$\mathbf{S}_V = \sum_{j=1}^{C} \sum_{\mathbf{x} \in \omega_j} (\mathbf{x} - \bar{\mathbf{x}}_j)(\mathbf{x} - \bar{\mathbf{x}}_j)^T = \sum_{j=1}^{C} \mathbf{S}_j$$



Intraclass scatter

$$\mathbf{S}_j = \sum_{\mathbf{x} \in \omega_j} (\mathbf{x} - \bar{\mathbf{x}}_j)(\mathbf{x} - \bar{\mathbf{x}}_j)^T.$$

Projection to w

$$\mathbf{S} = \mathbf{S}_M + \mathbf{S}_V$$
$$\mathbf{S}_M = \sum_{j=1}^{C} N_j(\bar{\mathbf{x}}_\mathbf{j} - \bar{\mathbf{x}})(\bar{\mathbf{x}}_\mathbf{j} - \bar{\mathbf{x}})^T$$
$$\mathbf{S}_V = \sum_{j=1}^{C} \sum_{\mathbf{x}\in\omega_j} (\mathbf{x} - \bar{\mathbf{x}}_j)(\mathbf{x} - \bar{\mathbf{x}}_j)^T = \sum_{j=1}^{C} \mathbf{S}_j$$

$$x_i' = \mathbf{w}^T \mathbf{x}_i \qquad \bar{x}_j' = \mathbf{w}^T \bar{\mathbf{x}}_j$$

$$\mathbf{Q} = \mathbf{Q}_M + \mathbf{Q}_V$$
$$\mathbf{Q} = \sum_{i=1}^{N} (\mathbf{w}^T\mathbf{x}_i - \mathbf{w}^T\bar{\mathbf{x}})(\mathbf{w}^T\mathbf{x}_i - \mathbf{w}^T\bar{\mathbf{x}})^T = \mathbf{w}^T\mathbf{S}\mathbf{w}$$
$$\mathbf{Q}_M = \sum_{j=1}^{C} N_j(\mathbf{w}^T\bar{\mathbf{x}}_\mathbf{j} - \mathbf{w}^T\bar{\mathbf{x}})(\mathbf{w}^T\bar{\mathbf{x}}_\mathbf{j} - \mathbf{w}^T\bar{\mathbf{x}})^T = \mathbf{w}^T\mathbf{S}_M\mathbf{w}$$
$$\mathbf{Q}_V = \sum_{j=1}^{C} \sum_{\mathbf{x}\in\omega_j} (\mathbf{w}^T\mathbf{x} - \mathbf{w}^T\bar{\mathbf{x}}_j)(\mathbf{w}^T\mathbf{x} - \mathbf{w}^T\bar{\mathbf{x}}_j)^T = \mathbf{w}^T\mathbf{S}_V\mathbf{w}.$$

$$J(\mathbf{w}) = \frac{\mathbf{w}^T\mathbf{S}_M\mathbf{w}}{\mathbf{w}^T\mathbf{S}_V\mathbf{w}}$$

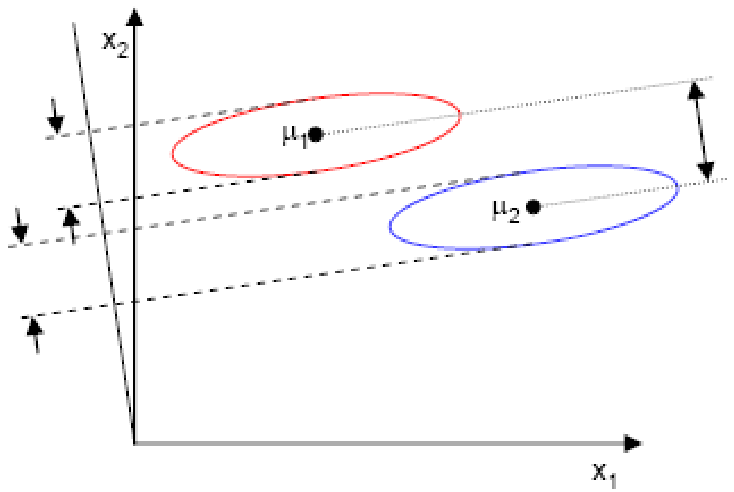
2 classes

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_M \mathbf{w}}{\mathbf{w}^T \mathbf{S}_V \mathbf{w}}$$

$$\frac{\mathrm{d}J}{\mathrm{d}\mathbf{w}} = (\mathbf{w}^T \mathbf{S}_M \mathbf{w}) 2\mathbf{S}_V \mathbf{w} - (\mathbf{w}^T \mathbf{S}_V \mathbf{w}) 2\mathbf{S}_M \mathbf{w} \equiv 0$$

$$\mathbf{S}_M \mathbf{w} = J \mathbf{S}_V \mathbf{w}$$

$$\mathbf{S}_V^{-1} \mathbf{S}_M \mathbf{w} = J \mathbf{w}$$



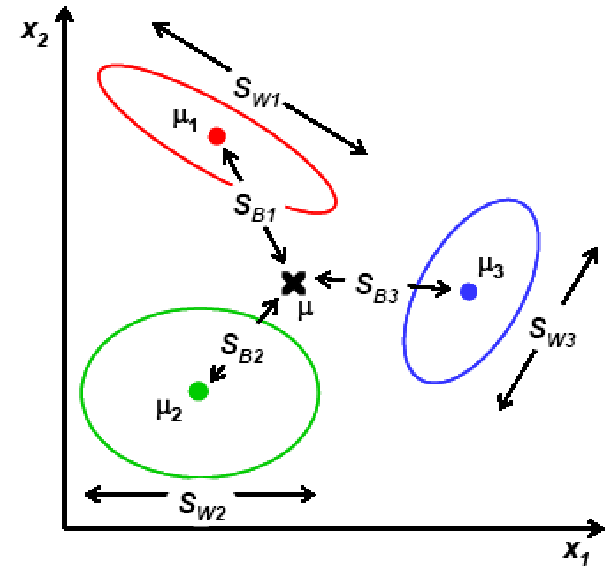
$$S_M = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^T$$
$$S_M \mathbf{v} = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^T \mathbf{v}$$
$$S_M \mathbf{v} = \alpha(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)$$

$$\mathbf{w} \propto S_V^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)$$

C classes

$$\mathbf{W} = [\mathbf{w}_1 | \mathbf{w}_2 | \ldots | \mathbf{w}_{C-1}]$$

$$J(\mathbf{w}) = \frac{|\mathbf{W}^T \mathbf{S}_M \mathbf{W}|}{|\mathbf{W}^T \mathbf{S}_V \mathbf{W}|}$$



Solve the generalized eigenvalue problem

$$(\mathbf{S}_M - \mathbf{S}_V \lambda)\mathbf{w} = 0.$$

Solve eigenvector problem $S_V^{-1} S_M \, \mathbf{w} - \lambda \mathbf{w} = \mathbf{0}$

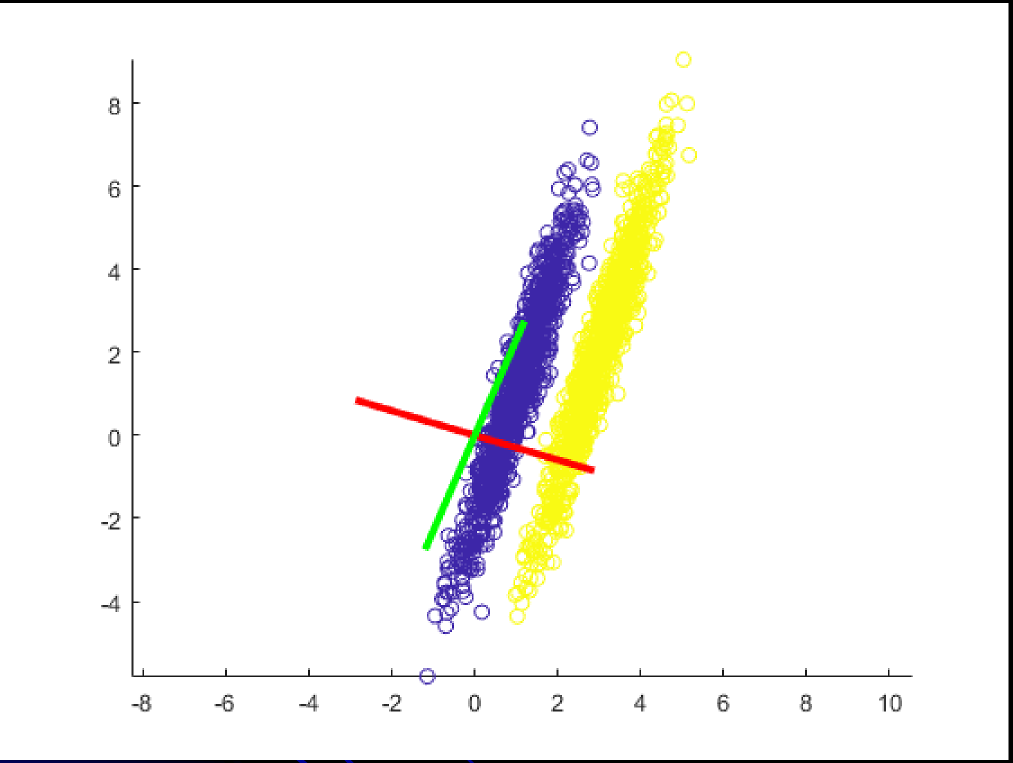
Algebraic solution

$$\mathrm{S}_M\,\mathbf{w} - \mathrm{S}_V\lambda\mathbf{w} = \mathbf{0}$$

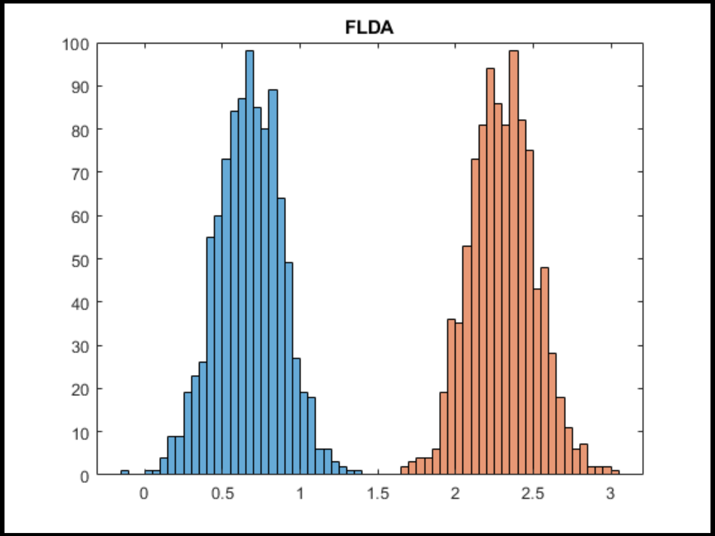
$$\mathrm{S}_M = \mathrm{V}\Psi\mathrm{V}^T$$
$$\mathrm{S}_V = \mathrm{U}\Phi\mathrm{U}^T$$
$$\mathrm{S}_V^{1/2} = \mathrm{U}\Phi^{1/2}\mathrm{U}^T$$
$$\mathrm{S}_M^{1/2} = \mathrm{V}\Psi^{1/2}\mathrm{V}^T$$

$$\mathrm{S}_M\left(\mathrm{S}_V^{-\frac{1}{2}}\mathrm{S}_V^{\frac{1}{2}}\right)\mathbf{w} - \mathrm{S}_V\left(\mathrm{S}_V^{-\frac{1}{2}}\mathrm{S}_V^{\frac{1}{2}}\right)\lambda\mathbf{w} = \mathbf{0} \quad \Big| \quad \mathrm{S}_V^{-1/2}.$$
$$\mathrm{S}_V^{-1/2}\mathrm{S}_M\,\mathrm{S}_V^{-1/2}\mathrm{S}_V^{1/2}\mathbf{w} - \mathrm{S}_V^{-1/2}\mathrm{S}_V\mathrm{S}_V^{-1/2}\mathrm{S}_V^{1/2}\lambda\mathbf{w} = \mathbf{0}$$
$$\mathrm{S}_V^{-1/2}\mathrm{S}_M\mathrm{S}_V^{-1/2}\mathbf{c} - \mathrm{S}_V^{-1/2}\mathrm{S}_V\mathrm{S}_V^{-1/2}\lambda\mathbf{c} = \mathbf{0}$$
$$\mathrm{U}\Phi^{-1/2}\mathrm{U}^T\mathrm{V}\Psi^{1/2}\,\Psi^{1/2}\mathrm{V}^T\mathrm{U}\Phi^{-1/2}\mathrm{U}^T\mathbf{c} - \lambda\mathbf{c} = \mathbf{0}$$
$$AA^T\mathbf{c} - \lambda\mathbf{c} = \mathbf{0}$$

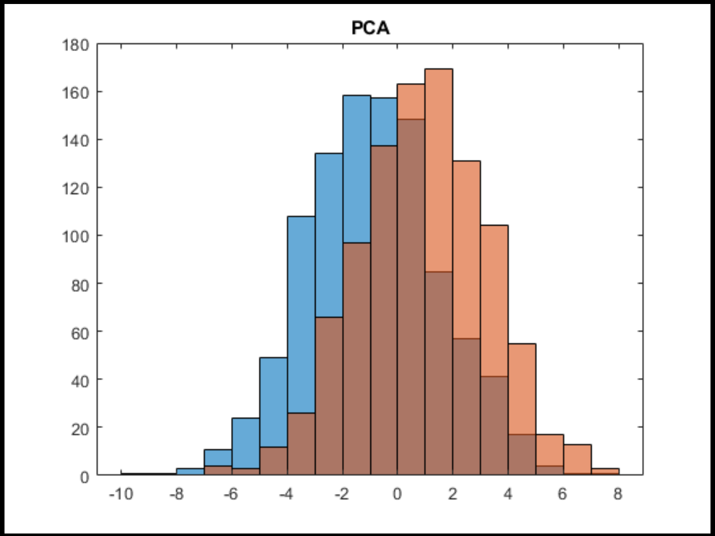
PCA vs LDA



8
6
4
2
0
-2
-4
-8
-6
-4
-2
0
2
4
6
8
10

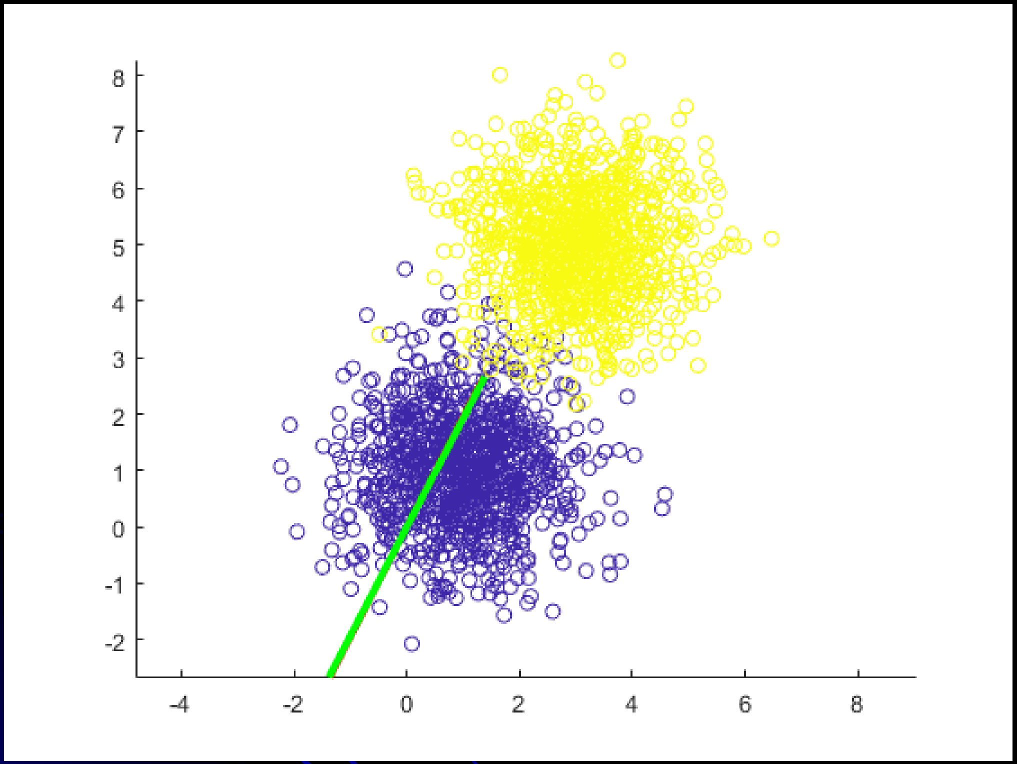


FLDA
100
90
80
70
60
50
40
30
20
10
0
0
0.5
1
1.5
2
2.5
3

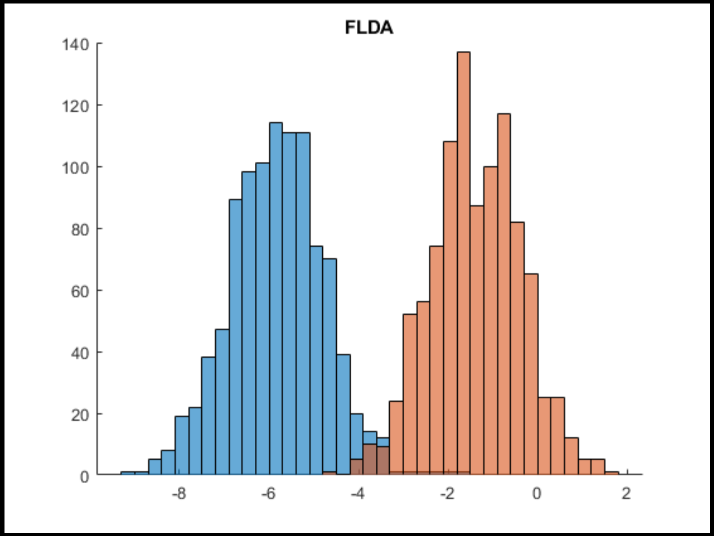


PCA
180
160
140
120
100
80
60
40
20
0
-10
-8
-6
-4
-2
0
2
4
6
8

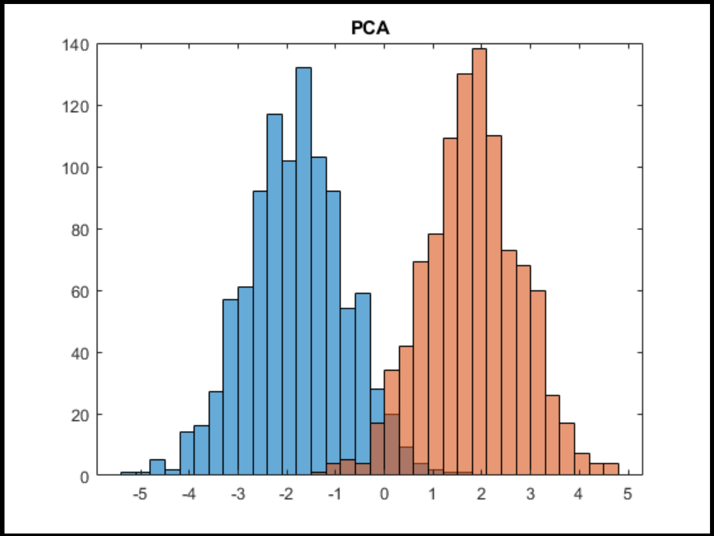
PCA vs LDA



8
7
6
5
4
3
2
1
0
-1
-2
-4
-2
0
2
4
6
8

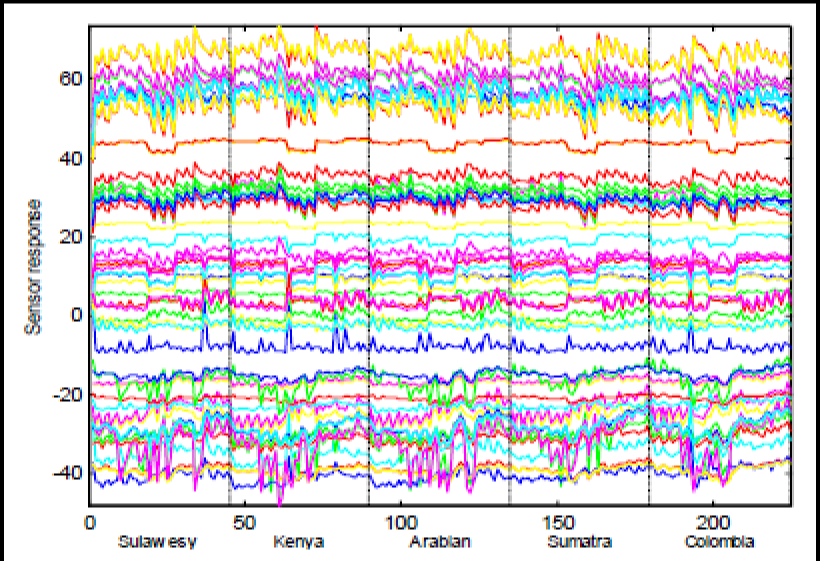


FLDA
140
120
100
80
60
40
20
0
-8
-6
-4
-2
0
2



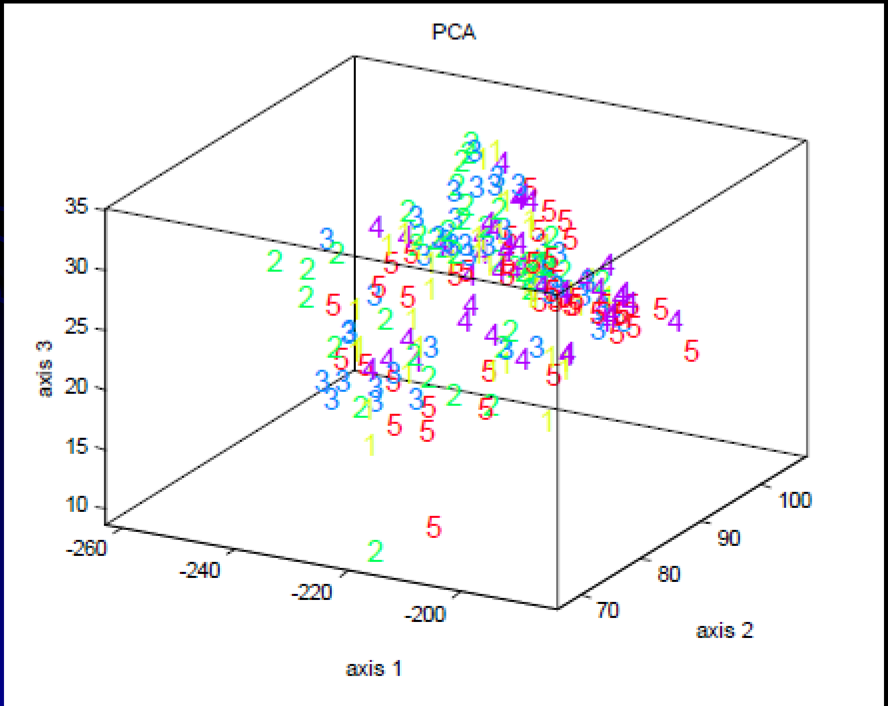
PCA
140
120
100
80
60
40
20
0
-5
-4
-3
-2
-1
0
1
2
3
4
5

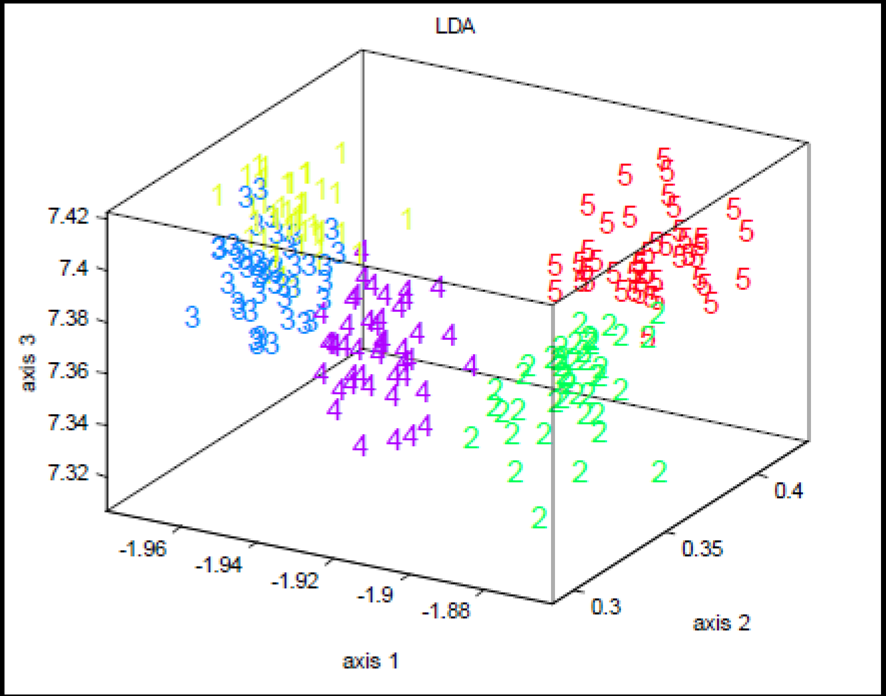
PCA vs LDA



Five types of coffee beans were presented to an array of gas sensors

For each coffee type, 45 “sniffs” were performed and the response of the gas sensor array was processed in order to obtain a 60-dimensional feature vector





http://courses.cs.tamu.edu/rgutier/csce666_f13/l10.pdf

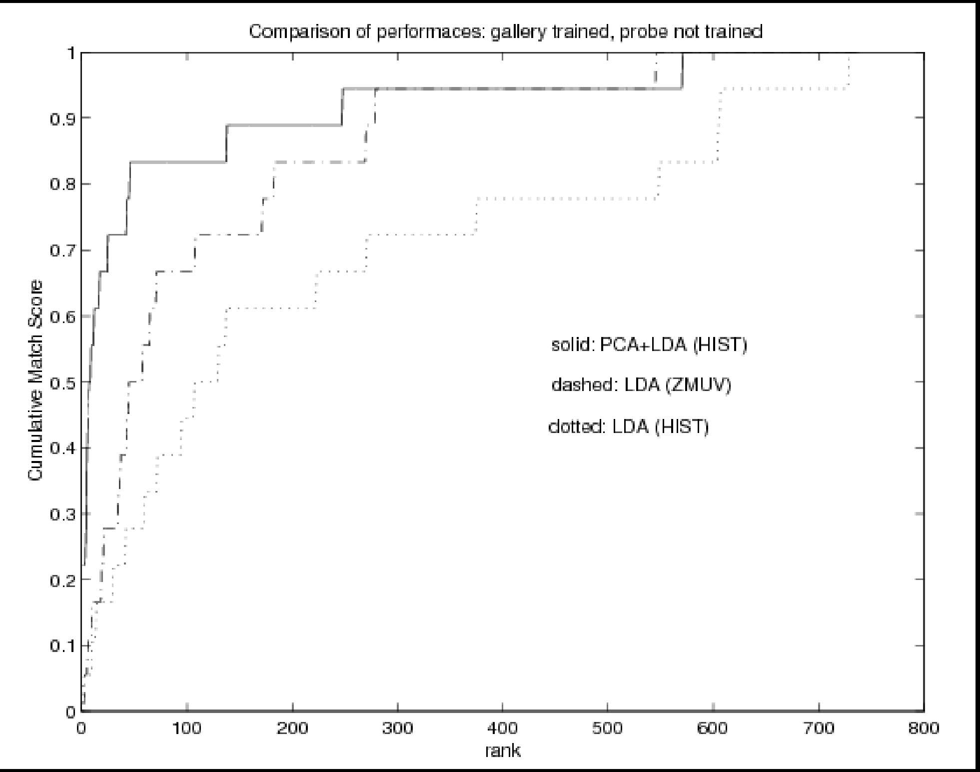
PCA – LDA combination

Use PCA lower the dimension

$$\begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_N \end{bmatrix} --> PCA --> \begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_K \end{bmatrix}$$

Find discriminative directions

$$\begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_K \end{bmatrix} --> LDA --> \begin{bmatrix} z_1 \\ z_2 \\ \cdots \\ z_{C-1} \end{bmatrix}$$



Zhao, W., Chellappa, R. and Krishnaswamy, A.
Discriminant Analysis of Principal Components for Face Recognition

PCA vs LDA

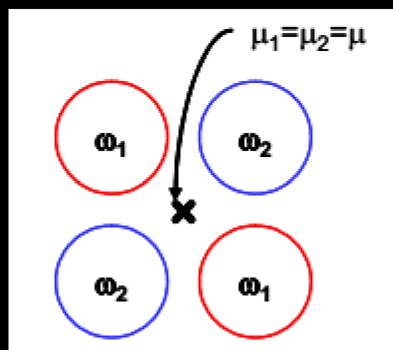
for small number of training data, PCA gives better results than LDA

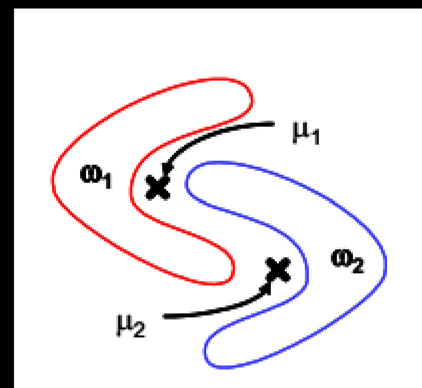
if we have enough training data for each class, LDA is better

A. Martinez, A. Kak, "PCA versus LDA", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 228-233, 2001.

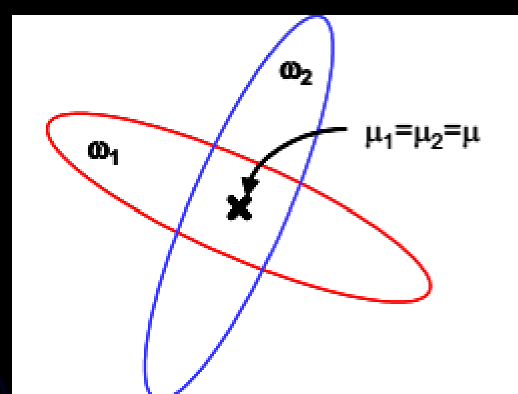
PCA vs LDA

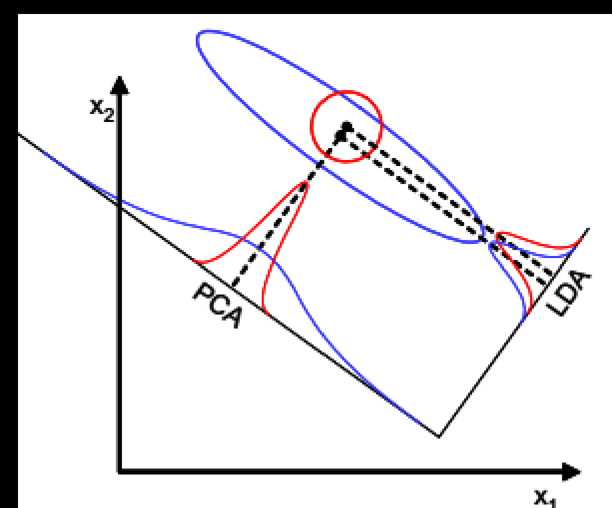
assume Gaussian distribution





If the class difference lies in variance but not mean, LDA fails





Nonlinear methods