Statistical Methods in Artificial Intelligence
CSE471 - Monsoon 2016 : Lecture 15

Avinash Sharma
CVIT, IIIT Hyderabad

Lecture Plan

- Revision from Previous Lecture
- Fisher's Linear Discriminant Analysis (LDA)
- Multiple Discriminant Analysis (MDA)
- Fisher Faces
  - Algorithm
  - Fisher-face Plots/Code
- Limitation of Fisher's LDA
- Support Vector Machine (Next Class)

Principal Component Analysis (PCA)

- Let data matrix $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_i, \dots, \mathbf{x}_n\}$ and $\mathbf{x}_i \in \mathbb{R}^d$.
- 1-dimensional representation: Let $\mathbf{x} = \mathbf{x}_0$

$$J_0(\mathbf{x}_0) = \sum_{i=1}^{n} \|\mathbf{x}_0 - \mathbf{x}_i\|^2, \qquad \mathbf{m} = \arg\min_{\mathbf{x}_0} J_0(\mathbf{x}_0) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$$

- 2-dimensional representation: Let $\mathbf{x} = \mathbf{m} + a\mathbf{e}$

$$J_1(a_1, \dots, a_n, \mathbf{e}) = \sum_{i=1}^{n} \|(\mathbf{m} + a_{\boldsymbol{i}}\mathbf{e}) - \mathbf{x}_i\|^2, \qquad a_i = \mathbf{e}^{\boldsymbol{T}}(\mathbf{x}_i - \mathbf{m})$$

$$J_1(\mathbf{e}) = -\mathbf{e}^{\boldsymbol{T}}\mathbf{S}\mathbf{e}^{\boldsymbol{T}} + \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{m}\|^2, \qquad \mathbf{v} = \arg\min_{\mathbf{e}} J_1(\mathbf{e})$$

$$\mathbf{S}\mathbf{v} = \lambda\mathbf{v}, \qquad \|\mathbf{v}\| = 1 \text{ where } \mathbf{S} = \sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^{\boldsymbol{T}}$$

Principal Component Analysis (PCA)

- k -dimensional representation:

Let $\mathbf{x} = \mathbf{m} + \sum_{i=1}^{k} a_i \mathbf{e}_i$

$$\mathbf{v}_1, \dots, \mathbf{v}_k = arg \max_{\mathbf{e}_{1,} \dots, \mathbf{e}_k} J_k = \sum_{i=1}^{n} \left\| \left(\mathbf{m} + \sum_{j=1}^{k} a_j \mathbf{e}_j \right) - \mathbf{x}_i \right\|^2 ,$$

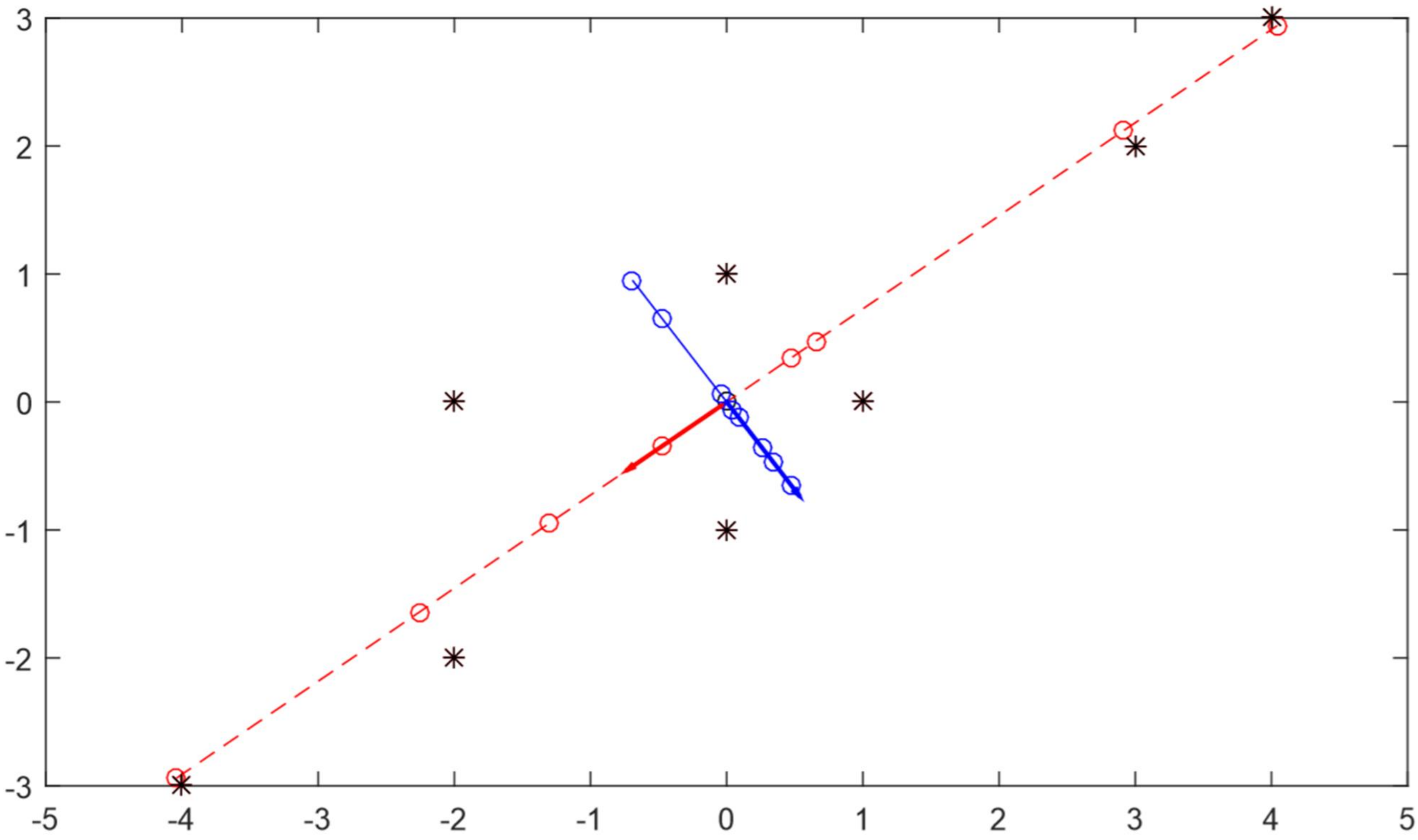
$$\text{for } k \ll d$$

where,

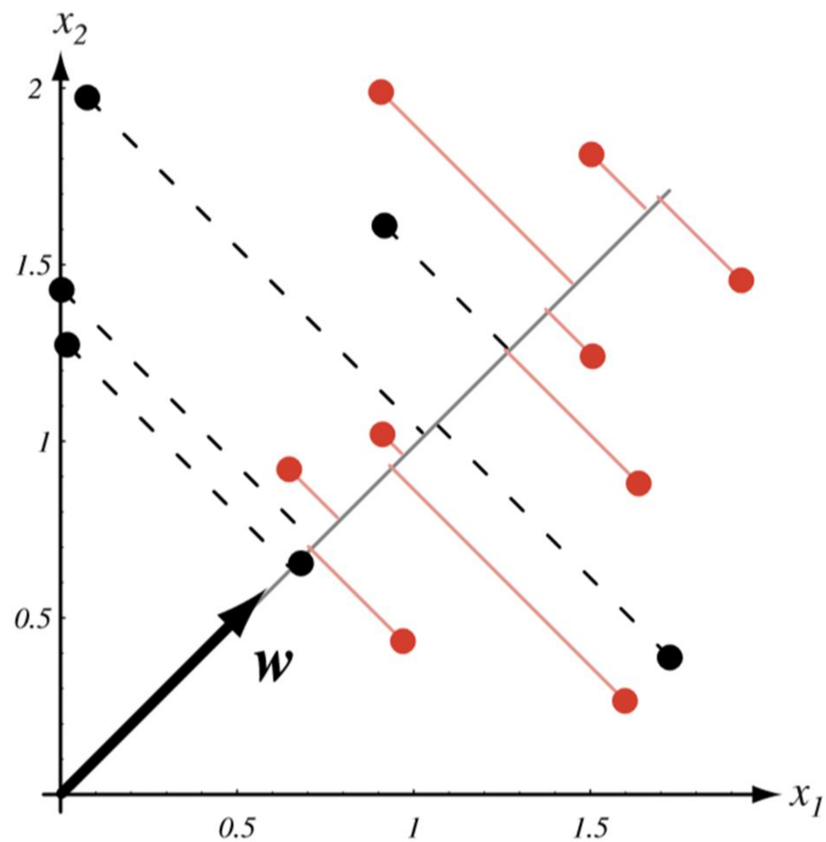
$$\mathbf{S}\mathbf{v}_i = \lambda_i \mathbf{v}_i ,$$

$$\mathbf{v}_i \perp \mathbf{v}_j , \|\mathbf{v}_i\| = 1 \ \forall \ i, j \ \in \{1, .., k\}$$

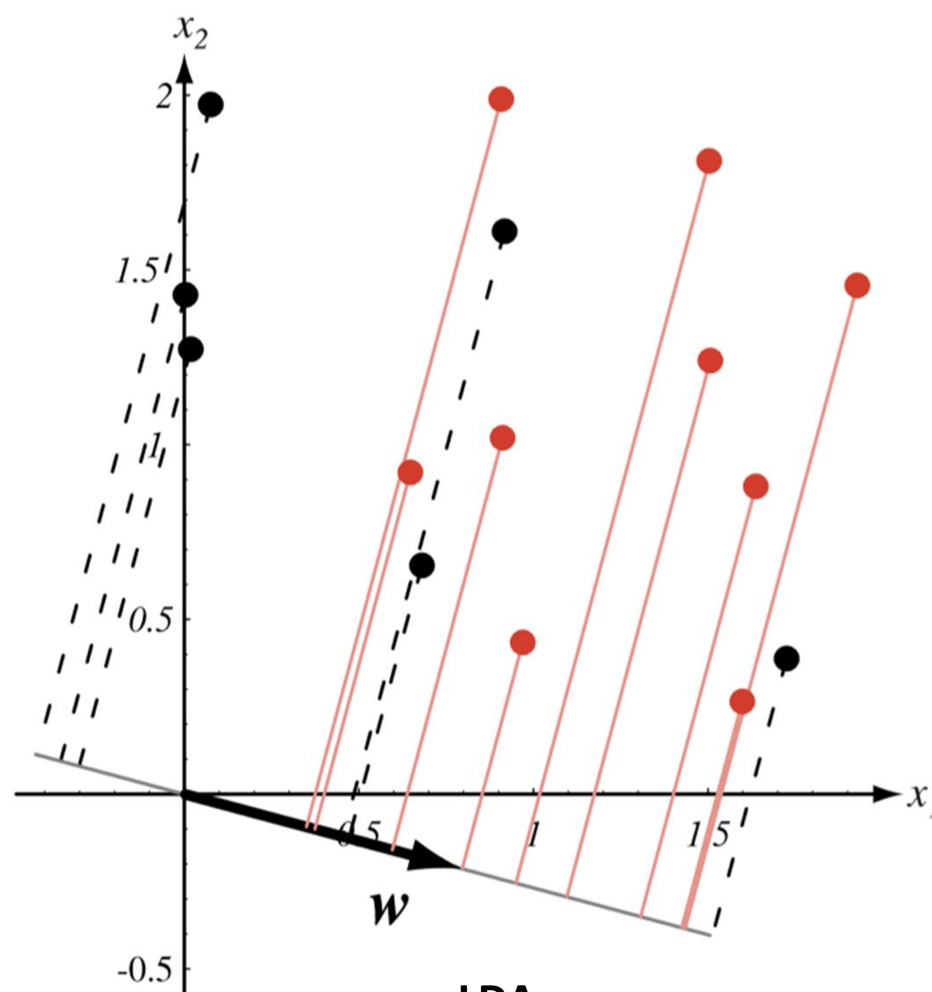
PCA: Practical Example



Fisher's Linear Discriminant Analysis (LDA)



PCA



LDA

LDA: Derivation

inter-class: $|\tilde{m}_1 - \tilde{m}_2| = |w^{\mathrm{T}}(m_1 - m_2)|$

intra-class: $\tilde{s}_i^2 = \sum_{y \in Y_i} (y - \tilde{m}_i)^2$

want to maximize: $J(w) = \dfrac{|\tilde{m}_1 - \tilde{m}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$

$y, \tilde{m}_1, \tilde{m}_2 : [\ ]_1^1$ $\quad (w^{\mathrm{T}}x - w^{\mathrm{T}}m_i) : [\ ]_1^1$

$x, w, m_1, m_2 : [\ ]_1^{\mathrm{D}}$ $\quad S_{\mathrm{B}}, S_{\mathrm{w}} : [\ ]_{\mathrm{D}}^{\mathrm{D}}$

$$\tilde{s}_i^2 = \sum_{x \in D_i} (w^{\mathrm{T}}x - w^{\mathrm{T}}m_i)(w^{\mathrm{T}}x - w^{\mathrm{T}}m_i)^{\mathrm{T}} = \sum_{x \in D_i} w^{\mathrm{T}}(x - m_i)(x - m_i)^{\mathrm{T}} w = w^{\mathrm{T}} S_i w$$

$$\tilde{s}_1^2 + \tilde{s}_2^2 = w^{\mathrm{T}} S_1 w + w^{\mathrm{T}} S_2 w = w^{\mathrm{T}} S_{\mathrm{w}} w$$

$$|\tilde{m}_1 - \tilde{m}_2|^2 = (w^{\mathrm{T}}m_1 - w^{\mathrm{T}}m_2)^2 = w^{\mathrm{T}}(m_1 - m_2)(m_1 - m_2)^{\mathrm{T}} w = w^{\mathrm{T}} S_{\mathrm{B}} w$$

want to maximize: $J(w) = \dfrac{w^{\mathrm{T}} S_{\mathrm{B}} w}{w^{\mathrm{T}} S_{\mathrm{w}} w}$

$S_{\mathrm{B}} w = \lambda S_{\mathrm{w}} w$

DF's for the Normal Density

- Case 2: Hyperellipsoidal Clusters $(\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma})$

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\boldsymbol{\Sigma}| + \ln P(\omega_i)$$

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) + \ln P(\omega_i)$$

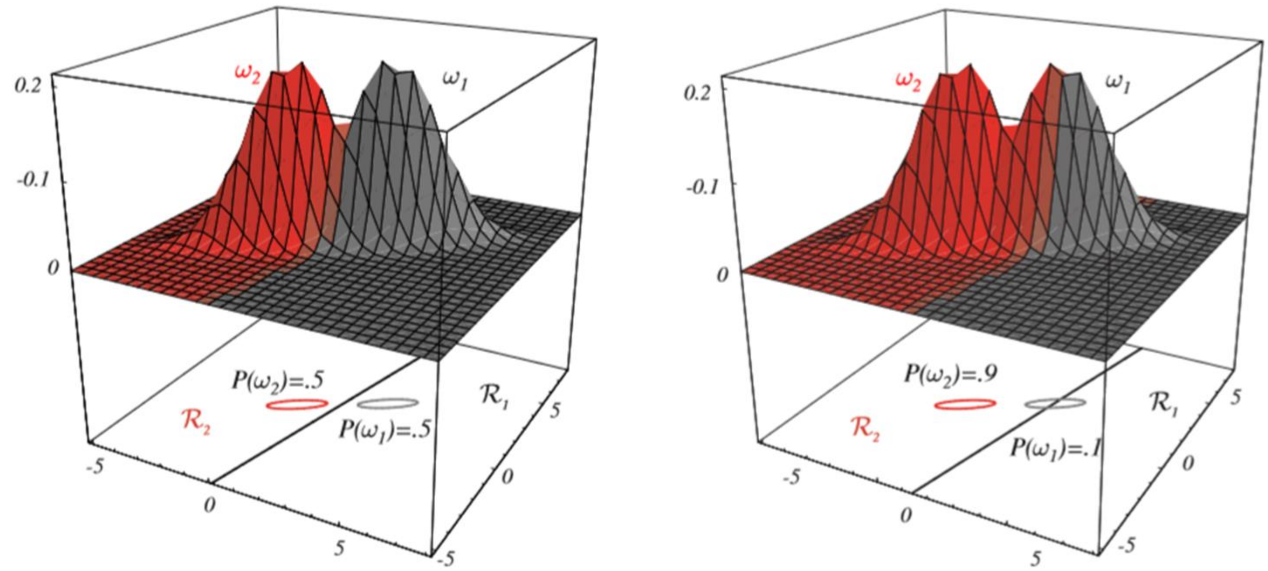
- Linear Discriminant Function for Two-category case:

$$\mathbf{w}^T(\mathbf{x}-\mathbf{x_0}) = 0$$

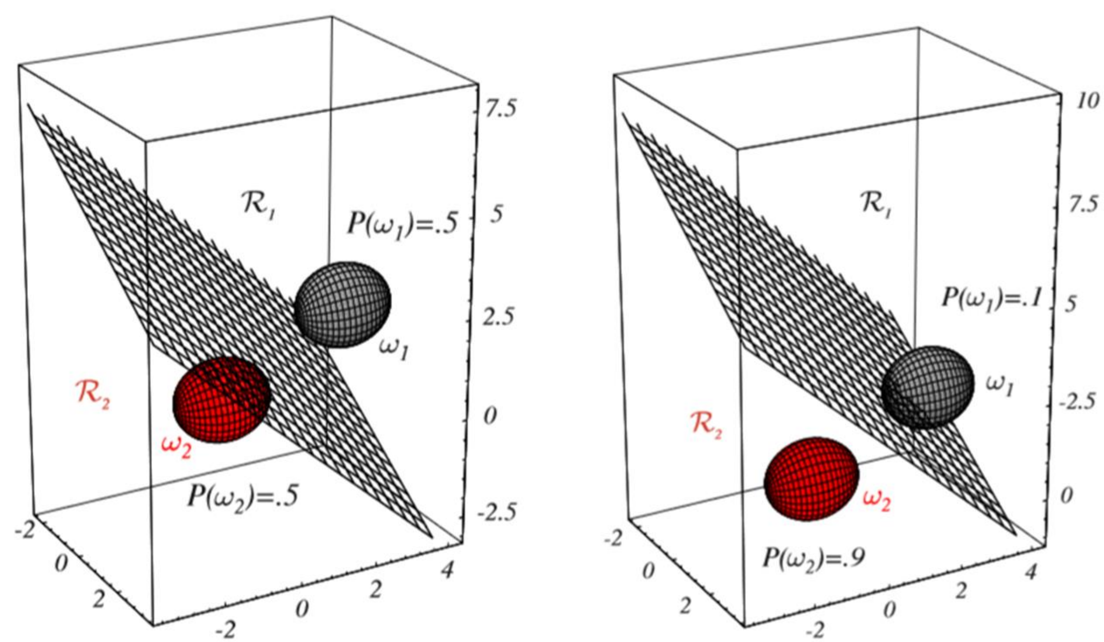
where, $\mathbf{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)$

and $\mathbf{x_0} = \frac{1}{2}(\boldsymbol{\mu}_i + \boldsymbol{\mu}_j) - \frac{\ln\left[P(\omega_i)/P(\omega_j)\right]}{(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j)}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)$

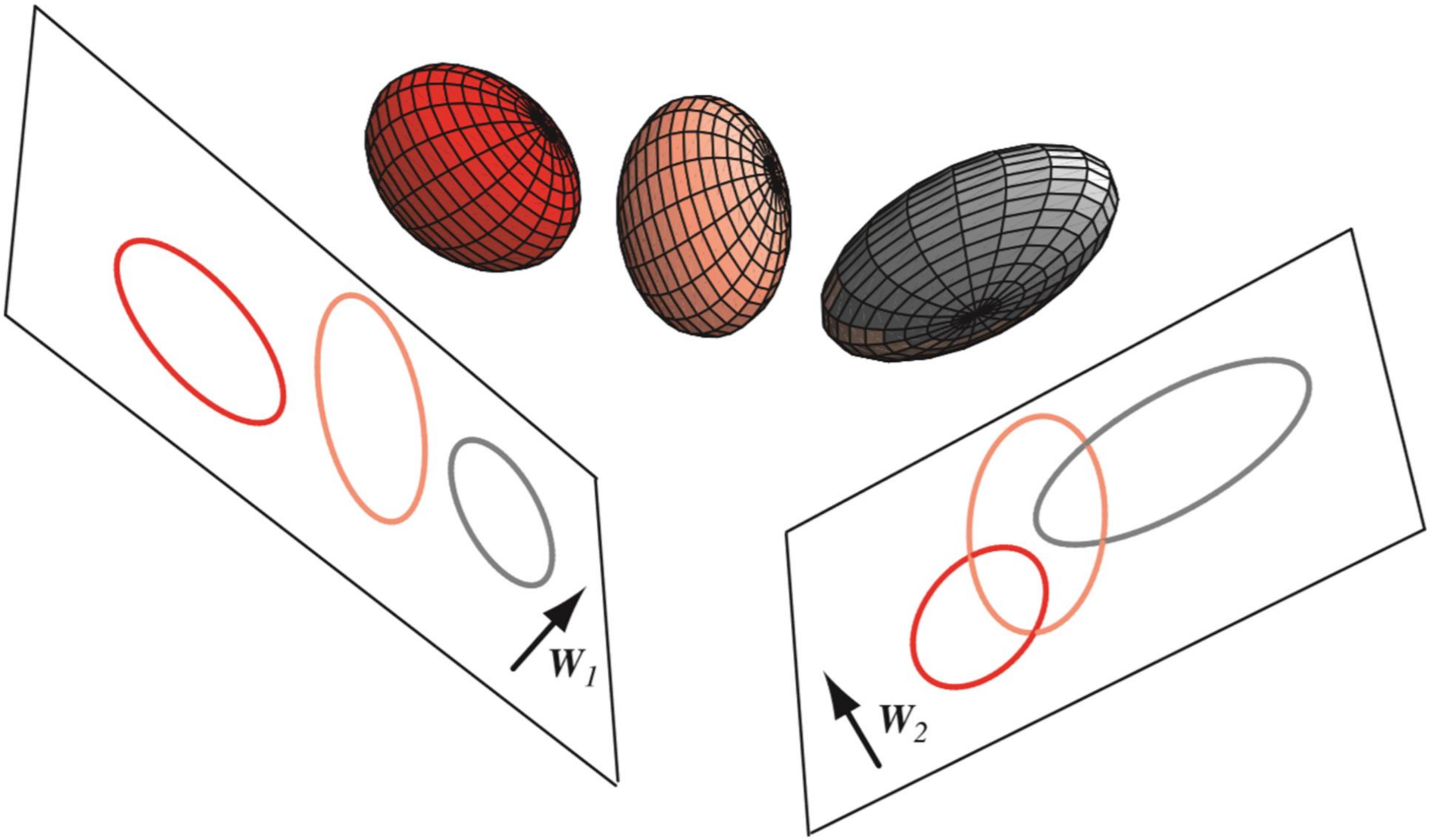
2D


ω_2
ω_1
0.2
-0.1
0
$P(\omega_2)$=.5
$P(\omega_1)$=.5
$\mathcal{R}_2$
$\mathcal{R}_1$
-5
0
5
$P(\omega_2)$=.9
$P(\omega_1)$=.1

3D


$\mathcal{R}_1$
$\mathcal{R}_2$
$P(\omega_1)$=.5
ω_1
ω_2
$P(\omega_2)$=.5
7.5
5
2.5
0
-2.5
-2
0
2
4
$P(\omega_1)$=.1
$P(\omega_2)$=.9
10

Multiple Discriminant Analysis (MDA)



Multiple Discriminant Analysis (MDA)

$$S_B = \sum_{i=1}^{c} N_i (m_i - m)(m_i - m)^T$$

c-1

$$S_w = \sum_{i=1}^{c} \sum_{x \in D_i} (x - m_i)(x - m_i)^T$$

N-c

want to maximize: $J(W) = \dfrac{|W^T S_B W|}{|W^T S_w W|}$

with $W = [w_1 \ w_2 \ldots\ldots w_m]$

$S_B w_i = \lambda_i S_w w_i$

$m \le c - 1$

Problem: S_w is always singular

$$S_B, S_w : \underset{D}{\left[\;\;\right]}^{D} \quad W : \underset{m}{\left[\;\;\right]}^{D} \quad W_{PCA} : \underset{N\text{-}c}{\left[\;\;\right]}^{D}$$

Fisherface solution:

$$W_{PCA} = \arg \max_{W} |W^T S_T W| \text{ where } S_T = \sum_{x} (x - m)(x - m)^T$$

$$W_{FLD} = \arg \max_{W} \frac{|W^T W_{PCA}{}^T S_B W_{PCA} W|}{|W^T W_{PCA}{}^T S_w W_{PCA} W|}$$

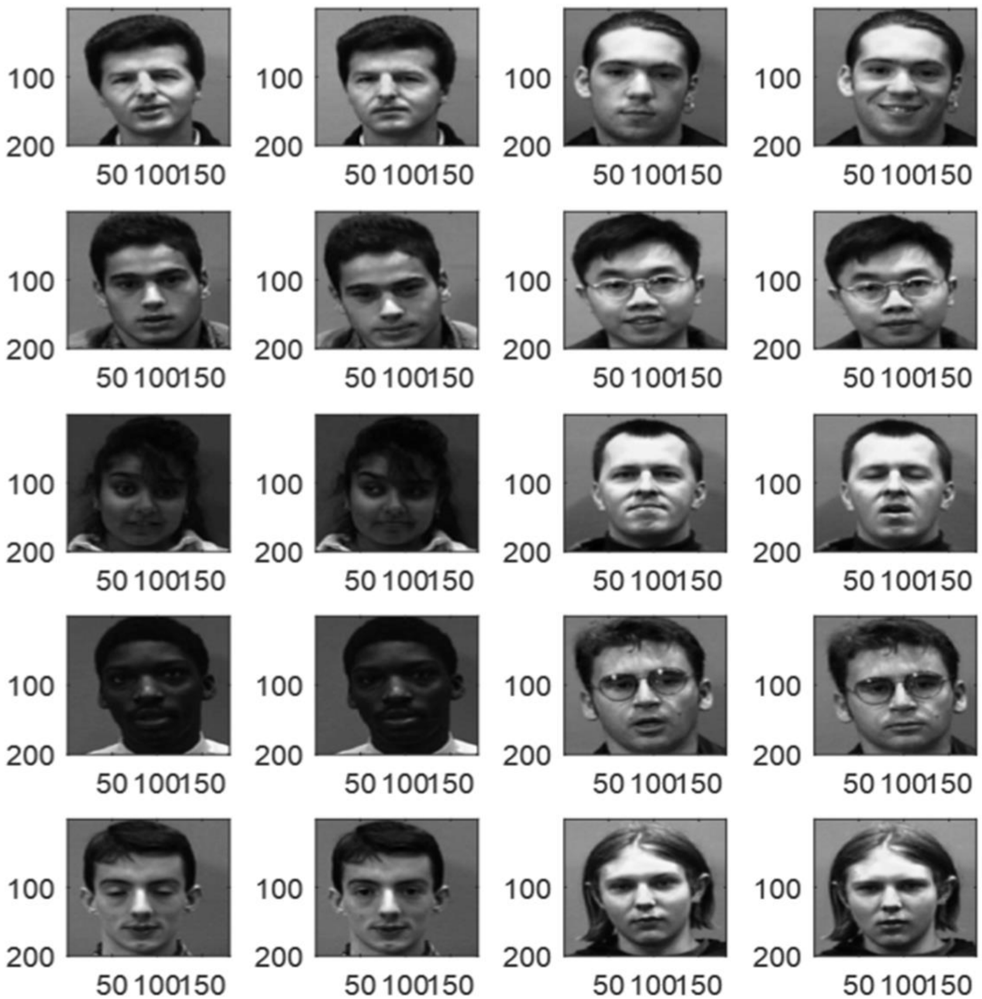
Fisher Faces

Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection
P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman
Journal of Cognitive Neuroscience, 3(1), pp. 71-86, 1991.

Algorithm to compute Fisherfaces:

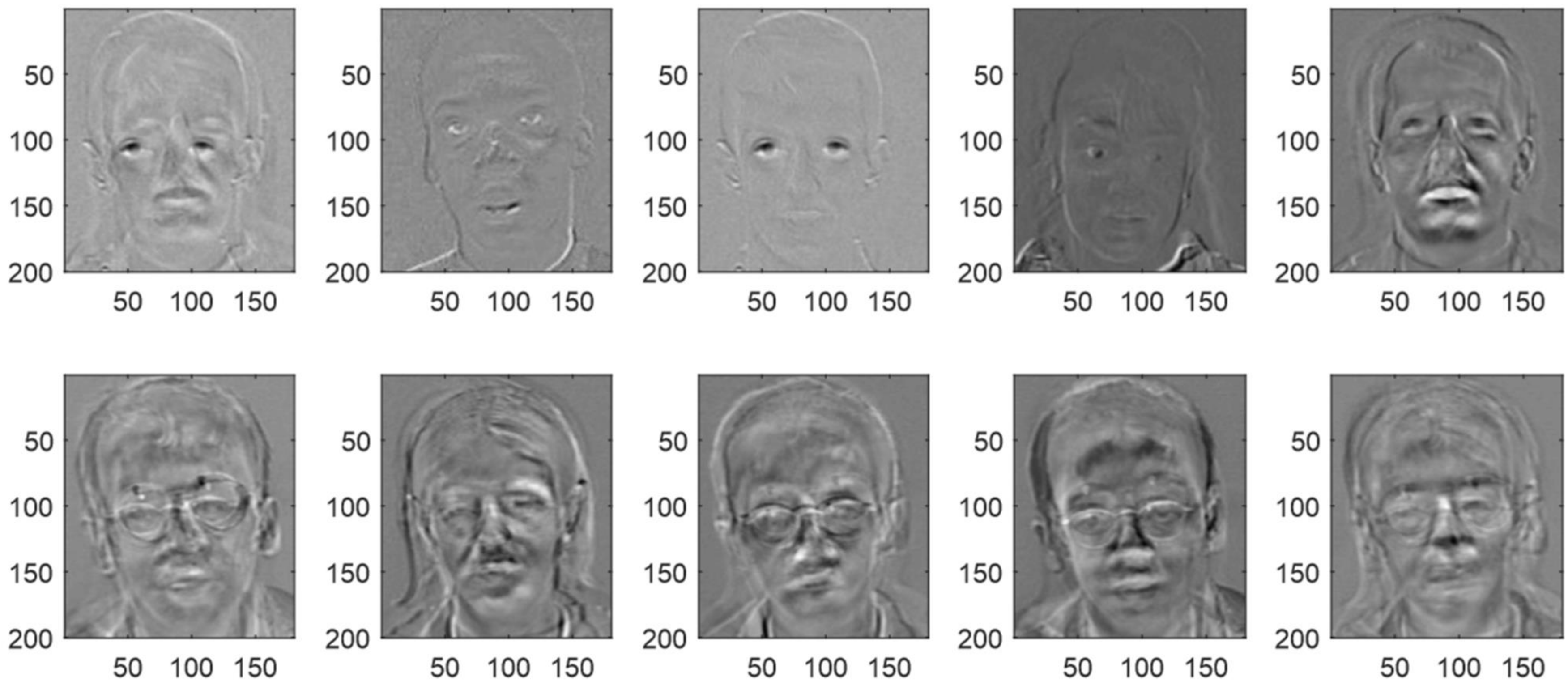
1. Vectorize grey scale training images to $d \times 1$ vector (from $w \times h$ matrix)
2. Compute Scatter Matrix $(d \times d)$ using the Mean vector and all data points
3. Project each training & test image to $(N - c)$ dimensional subspace using the $(N - c)$ eigenvectors associated to largest eigenvalues of the Scatter matrix
4. Computer $(N - c) \times (N - c)$ (within and between class) Scatter Matrices S_W and S_B
5. Computer $(c - 1)$ Fisher vectors as the largest eigenvectors of ${S_W}^{-1} S_B$
6. Project each of the $(N - c)$ dimensional train and test images (already projected in PCA space) to $(c - 1)$ dimensional Fisher space.

Fisher Faces

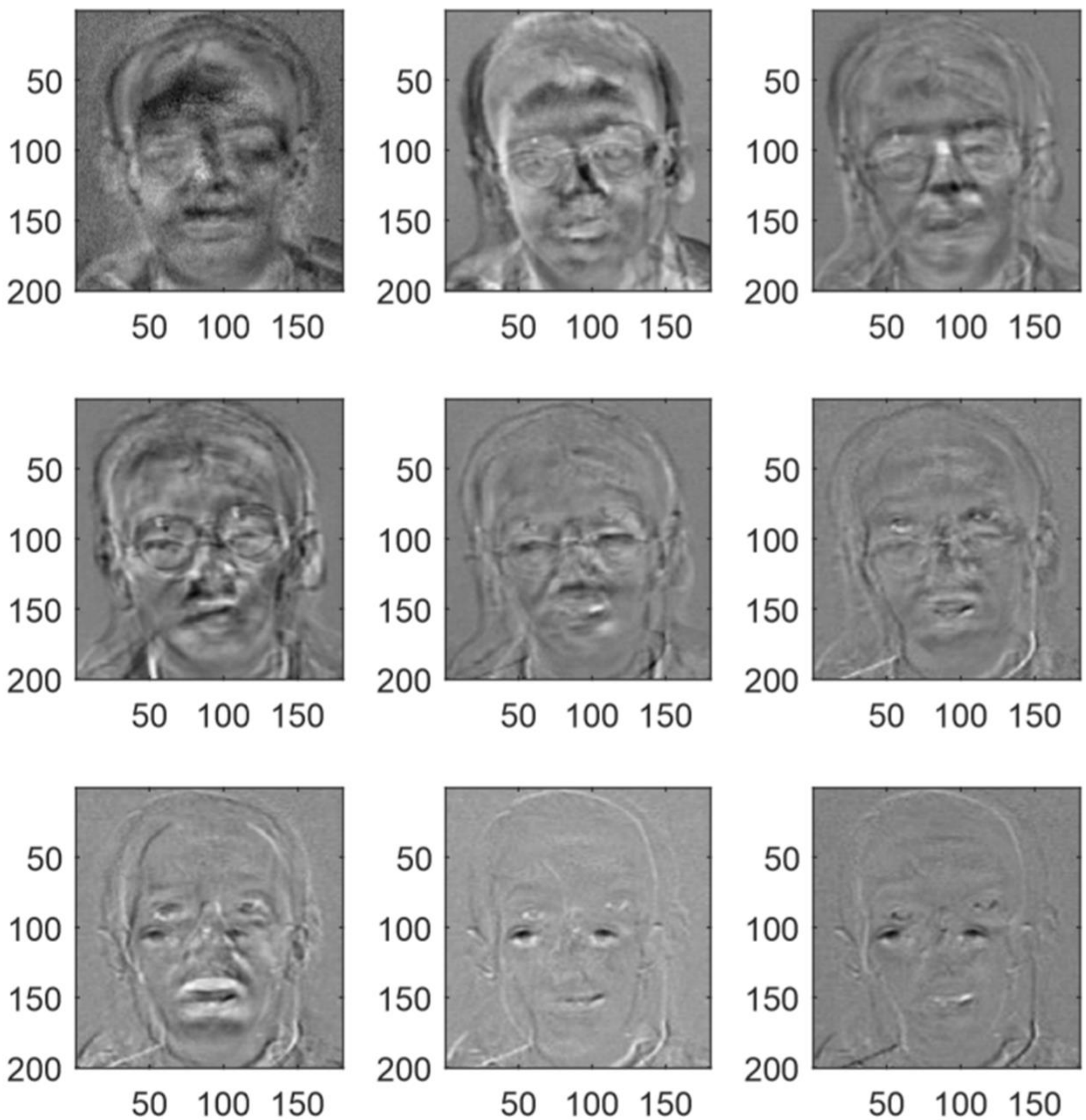
- 

Fisher Faces

- EigenFaces



Fisher Faces

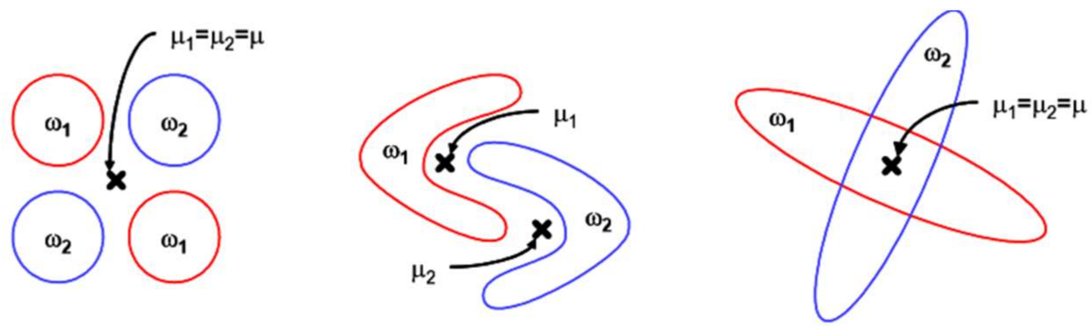


```
function [m_database V_PCA V_Fisher ProjectedImages_Fisher] = FisherfaceCore(T)
 % Original version by Amir Hossein Omidvarnia, October 2007
%                  Email: aomidvar@ece.ut.ac.ir

 Class_number = ( size(T,2) )/2; % Number of classes (or persons)
Class_population = 2; % Number of images in each class
P = Class_population * Class_number; % Total number of training images

%%%%%%%%%%%%%%%%%%%%%% calculating the mean image
m_database = mean(T,2);

%%%%%%%%%%% Calculating the deviation of each image from mean image
A = T - repmat(m_database,1,P);

%%%%%%%%%%%%%%%%Snapshot method of Eigenface algorithm
L = A'*A; % L is the surrogate of covariance matrix C=A*A'.
[V D] = eig(L); % Diagonal elements of D are the eigenvalues for both L=A'*A and C=A*A'.

%%%%%%%%%%%%Sorting and eliminating small eigenvalues
L_eig_vec = [];
for i = 1 : P-Class_number
   L_eig_vec = [L_eig_vec V(:,i)];
end
 %%%%%%%%%%% Calculating the eigenvectors of covariance matrix 'C'
V_PCA = A * L_eig_vec; % A: centered image vectors

%%%%%%%%% Projecting centered image vectors onto eigenspace
% Zi = V_PCA' * (Ti-m_database)
ProjectedImages_PCA = [];
for i = 1 : P
   temp = V_PCA'*A(:,i);
   ProjectedImages_PCA = [ProjectedImages_PCA temp];
end

%%%%%%%%%%%%%%%Calculating the mean of each class in eigenspace
m_PCA = mean(ProjectedImages_PCA,2); % Total mean in eigenspace
m = zeros(P-Class_number,Class_number);

%%%%%%%%%%%%%%%Calculating Within and Between class scatter matrices
Sw = zeros(P-Class_number,P-Class_number); % Initialization os Within Scatter Matrix
Sb = zeros(P-Class_number,P-Class_number); % Initialization of Between Scatter Matrix
for i = 1 : Class_number
   m(:,i) = mean( ( ProjectedImages_PCA(:,((i-1)*Class_population+1):i*Class_population) ), 2 )';

   S  = zeros(P-Class_number,P-Class_number);
   for j = ( (i-1)*Class_population+1 ) : ( i*Class_population )
      S = S + (ProjectedImages_PCA(:,j)-m(:,i))*(ProjectedImages_PCA(:,j)-m(:,i))';
   end
   Sw = Sw + S; % Within Scatter Matrix
   Sb = Sb + (m(:,i)-m_PCA) * (m(:,i)-m_PCA)'; % Between Scatter Matrix
End

 %%%%%%%%%%%%%%%%%%%%Calculating Fisher discriminant basis's
% We want to maximise the Between Scatter Matrix, while minimising the
% Within Scatter Matrix.
[J_eig_vec, J_eig_val] = eig(Sb,Sw); % Cost function J = inv(Sw) * Sb
J_eig_vec = fliplr(J_eig_vec);

%%%%%%%%%%%%%%Eliminating zero eigens and sorting in descend order
for i = 1 : Class_number-1
   V_Fisher(:,i) = J_eig_vec(:,i); % Largest (C-1) eigen vectors of matrix J
End

 %%%%%%%%%%%%%%%%%%%%Projecting images onto Fisher linear space
% Yi = V_Fisher' * V_PCA' * (Ti - m_database)
for i = 1 : Class_number*Class_population
   ProjectedImages_Fisher(:,i) = V_Fisher' * ProjectedImages_PCA(:,i);
end
```

Limitation of Fisher's LDA

- LDA produces at most $c-1$ feature projections

- LDA is a parametric method since it assumes unimodal Gaussian likelihoods



- LDA will fail when the discriminatory information is not in the mean but rather in the variance of the data