Face Recognition

Arnaldo J. Abrantes

Problem

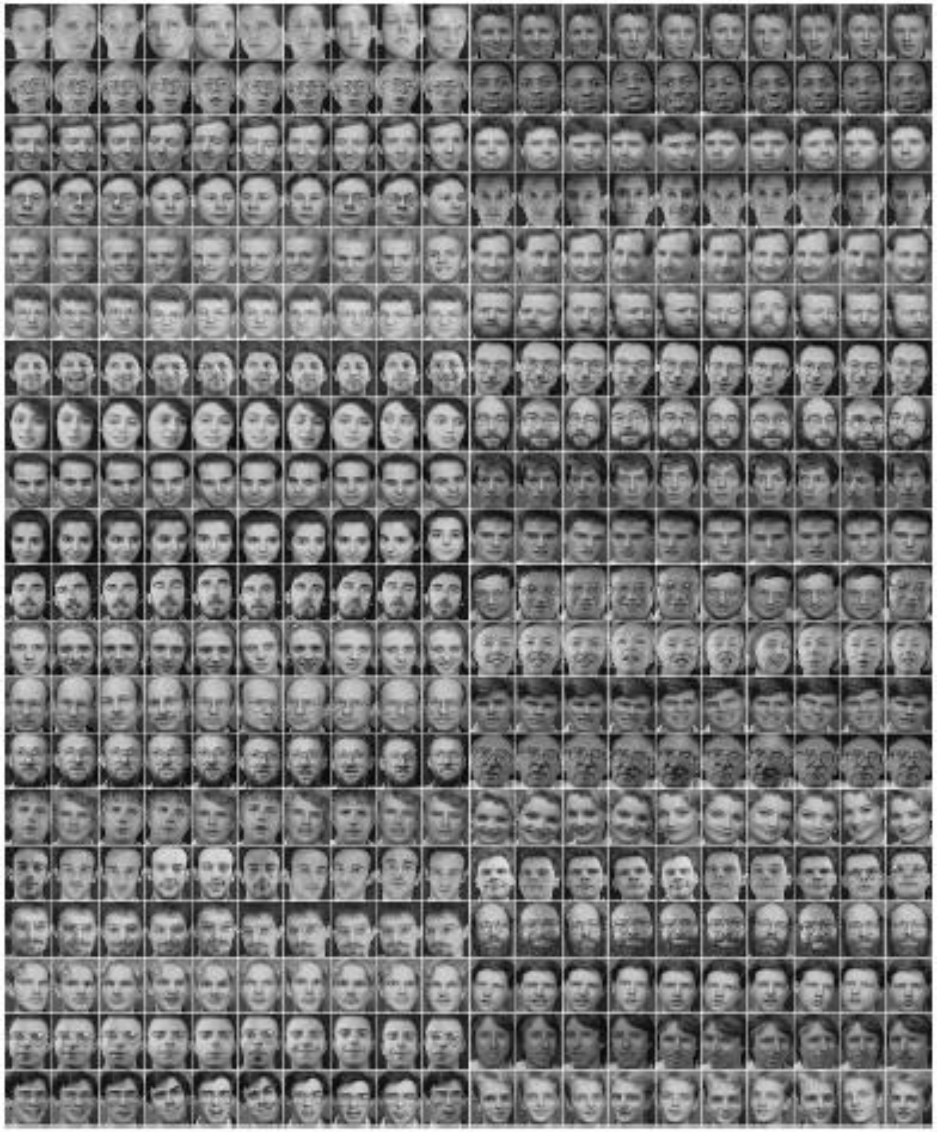
- **Goal**: Given a set of faces with known identity (training set) and a set of faces not identified, belonging to the same group of people (test set), we intend to identify each person in the test set

- **Difficulty**: The faces are immersed in a high dimensional space

- **Strategy**: Reduce the dimensionality of space through linear combination of features – Projection in sub-spaces

- **Two methods:**
  - *Eigenfaces* – PCA (*Principal Component Analysis*)
  - *Fisherfaces* – MDA (*Multiple Discriminant Analysis*)

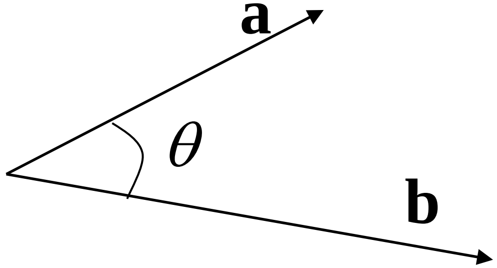
Training Set



Revision - Vector Spaces

- Let $\mathbf{a}, \mathbf{b} \in \Re^n$ be two vectors

$$\mathbf{a} = (a_1, \ldots, a_n)^T \qquad \mathbf{b} = (b_1, \ldots, b_n)^T$$



– Vector norm

$$\|\mathbf{a}\| = \left(\mathbf{a}^T\mathbf{a}\right)^{1/2}$$

– Inner product

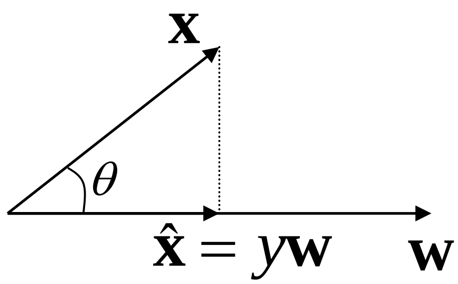
$$< \mathbf{a}, \mathbf{b} > = \|\mathbf{a}\|\|\mathbf{b}\|\cos\theta = \mathbf{a}^T\mathbf{b}$$

– Angle between two vectors

$$\theta = \cos^{-1}\frac{\mathbf{a}^T\mathbf{b}}{\left(\mathbf{a}^T\mathbf{a}\mathbf{b}^T\mathbf{b}\right)^{1/2}}$$

Vector Spaces Projection

- Consider the orthogonal projection



$$\mathbf{w}^T\mathbf{x} = \|\mathbf{x}\|\|\mathbf{w}\|\cos\theta$$

$$\cos\theta = \frac{y\|\mathbf{w}\|}{\|\mathbf{x}\|}$$

$$\boxed{y = \frac{\mathbf{w}^T\mathbf{x}}{\|\mathbf{w}\|^2}}$$

- Hypothesis: orthonormal basis

$$(\mathbf{w}_1,\ldots,\mathbf{w}_m) \quad m \le n$$

$$\mathbf{w}_i^T\mathbf{w}_j = \begin{cases} 1 & se \quad i = j \\ 0 & se \quad i \ne j \end{cases}$$

$$\hat{\mathbf{x}} = \sum_{i=1}^{m} y_i\mathbf{w}_i \qquad \boxed{y_i = \mathbf{w}_i^T\mathbf{x}}$$

Two interpretations

- Interpretation 1: $\mathbf{x} \in \Re^n \xrightarrow{W} \mathbf{y} \in \Re^m$ $\quad W = \left(\mathbf{w}_1 | \mathbf{w}_2 | \ldots | \mathbf{w}_m\right)$

$$\mathbf{y} = (y_1, \ldots, y_m)^T$$

$$\mathbf{y} = W^T \mathbf{x}$$

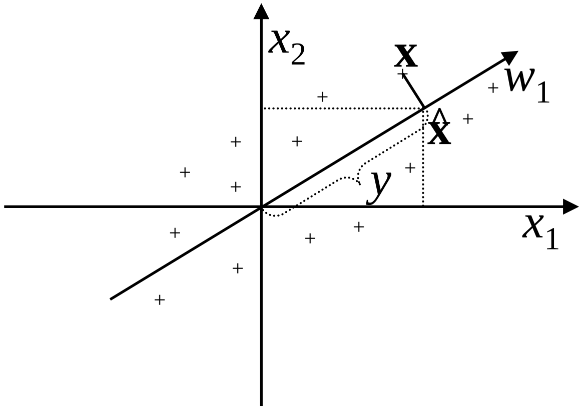
$\downarrow$

n x m

- Interpretations 2: $\mathbf{x} \in \Re^n \xrightarrow{P} \hat{\mathbf{x}} \in \Re^n$ $\quad P = WW^T$

$\downarrow$

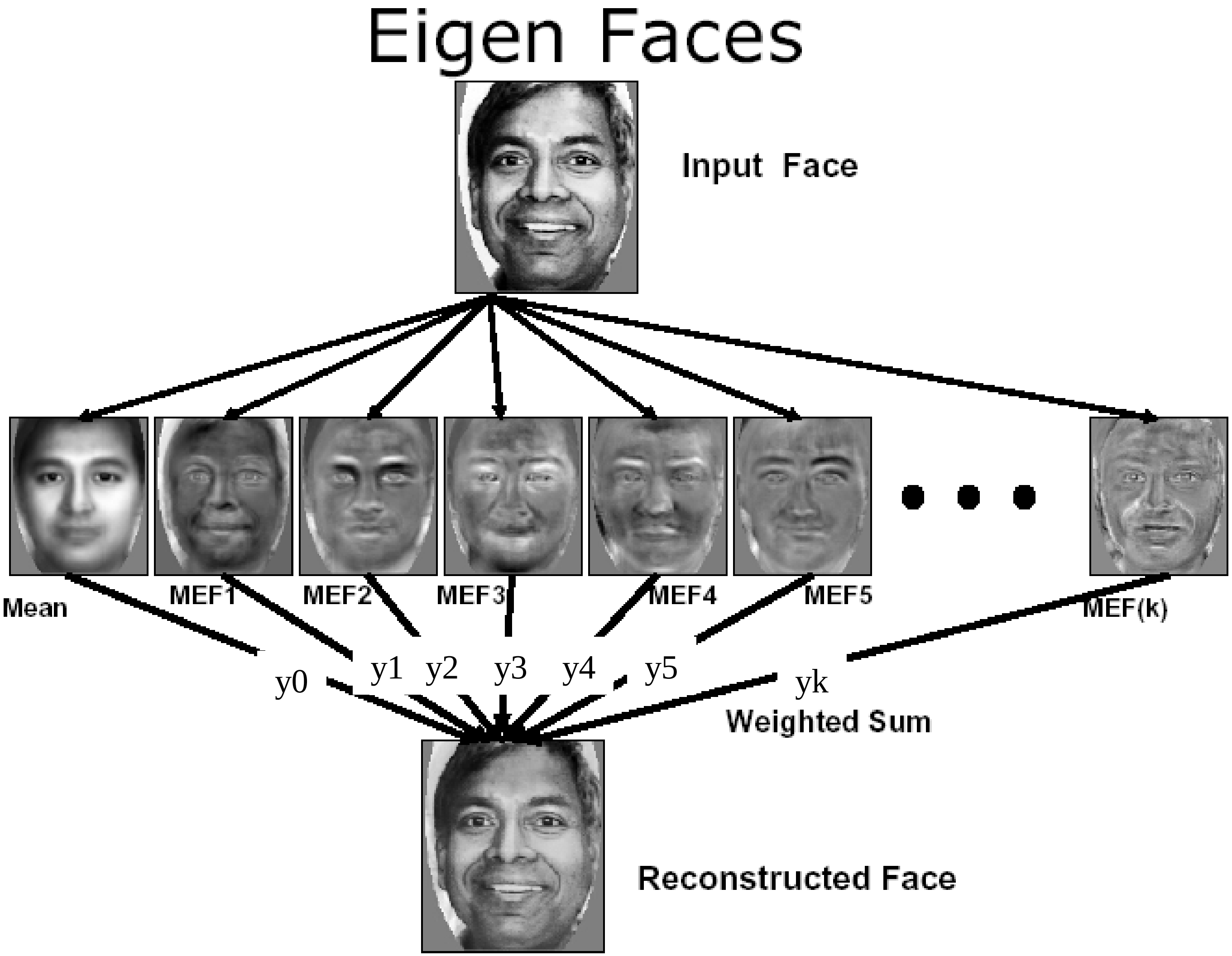
Projection operator:
• Idempotent – $P^2 = I$



$$\begin{aligned}\hat{\mathbf{x}} &= y_1\mathbf{w}_1 + \ldots + y_m\mathbf{w}_m \\ &= \mathbf{w}_1^T\mathbf{x}\mathbf{w}_1 + \ldots + \mathbf{w}_m^T\mathbf{x}\mathbf{w}_m \\ &= \left(\mathbf{w}_1\mathbf{w}_1^T\right)\mathbf{x} + \ldots + \left(\mathbf{w}_m\mathbf{w}_m^T\right)\mathbf{x} \\ &= P\mathbf{x}\end{aligned}$$

$\hat{\mathbf{x}} = WW^T\mathbf{x}$ $\qquad$ $\hat{\mathbf{x}} = W\mathbf{y}$

Eigenfaces


Eigen Faces
Input Face
Mean
MEF1
MEF2
MEF3
MEF4
MEF5
MEF(k)
y0
y1
y2
y3
y4
y5
yk
Weighted Sum
Reconstructed Face

Problem Formulation

- How to choose the basis of representation?

$$W = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_m)$$

- Consider having a training set with N vectors (faces)

$$\mathbf{x}_1, \ldots, \mathbf{x}_N$$

- Minimizing least squares criterion (PCA)

$$J_m = \sum_{k=1}^{N} \left\| \mathbf{x}_k - \hat{\mathbf{x}}_k \right\|^2$$

$$\hat{\mathbf{x}}_k = \boldsymbol{\mu} + \sum_{i=1}^{m} y_{ki} \mathbf{w}_i$$

$$\boldsymbol{\mu} = \frac{1}{N} \sum_{k=1}^{N} \mathbf{x}_k$$

↑

Mean Face

- What are the “optimal” values for y_{ki} and $\mathbf{w}_i$?

Example – case *m* = 1

- Orthogonal projection

$$J_1 = \sum_{k=1}^{N} \left\| \mathbf{x}_k - (\boldsymbol{\mu} + y_{k1}\mathbf{w}_1) \right\|^2$$

$$= \sum_{k=1}^{N} \left\| y_{k1}\mathbf{w}_1 - (\mathbf{x}_k - \boldsymbol{\mu}) \right\|^2$$

$$= \sum_{k=1}^{N} y_{k1}^2 - 2\sum_{k=1}^{N} y_{k1}\mathbf{w}_1^T(\mathbf{x}_k - \boldsymbol{\mu}) + \sum_{k=1}^{N} \left\| \mathbf{x}_k - \boldsymbol{\mu} \right\|^2$$

$$\frac{dJ_1}{dy_{k1}} = 2y_{k1} - 2\mathbf{w}_1^T(\mathbf{x}_k - \boldsymbol{\mu}) = 0$$

$$y_{k1} = \mathbf{w}_1^T(\mathbf{x}_k - \boldsymbol{\mu})$$

- In what direction?

$$J_1 = \sum_{k=1}^{N} y_{k1}^2 - 2\sum_{k=1}^{N} y_{k1}^2 + \sum_{k=1}^{N} \left\| \mathbf{x}_k - \boldsymbol{\mu} \right\|^2$$

$$= -\sum_{k=1}^{N} \left[\mathbf{w}_1^T(\mathbf{x}_k - \boldsymbol{\mu})\right]^2 + \sum_{k=1}^{N} \left\| \mathbf{x}_k - \boldsymbol{\mu} \right\|^2$$

$$= -\sum_{k=1}^{N} \mathbf{w}_1^T(\mathbf{x}_k - \boldsymbol{\mu})(\mathbf{x}_k - \boldsymbol{\mu})^T\mathbf{w}_1 + \sum_{k=1}^{N} \left\| \mathbf{x}_k - \boldsymbol{\mu} \right\|^2$$

$$= -\mathbf{w}_1^T S \mathbf{w}_1 + \sum_{k=1}^{N} \left\| \mathbf{x}_k - \boldsymbol{\mu} \right\|^2 \longleftarrow$$

Minimize

Scatter matrix:

$$S = \sum_{k=1}^{N} (\mathbf{x}_k - \boldsymbol{\mu})(\mathbf{x}_k - \boldsymbol{\mu})^T$$

$$\uparrow$$

(*n* x *n*)

Optimization problem with restrictions

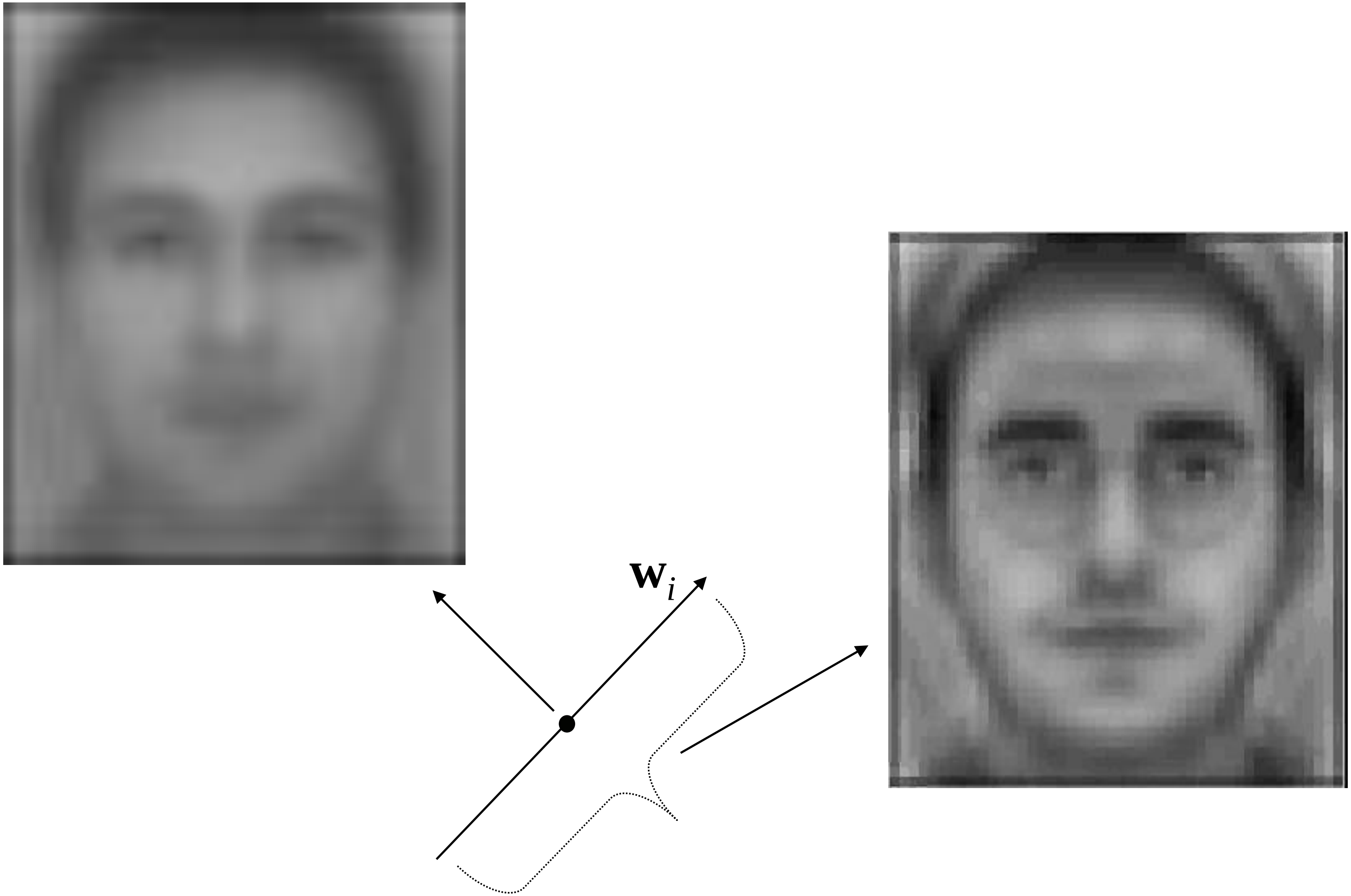
- Minimize J_1 is the same as:
  - Maximize $\mathbf{w}_1^T S \mathbf{w}_1$
  - Subject to the restriction $\left\|\mathbf{w}_1\right\| = 1$
- Method of Lagrange multipliers $u = \mathbf{w}_1^T S \mathbf{w}_1 - \lambda\left(\mathbf{w}_1^T \mathbf{w}_1 - 1\right)$

$$\frac{\partial u}{\partial w_1} = 2S\mathbf{w}_1 - 2\lambda\mathbf{w}_1 = 0 \longrightarrow S\mathbf{w}_1 = \lambda\mathbf{w}_1$$

$$\mathbf{w}_1^T S \mathbf{w}_1 = \lambda \longleftarrow$$ Maximizing *u* corresponds to choose the eigenvector whose eigenvalue is maximum

- How to generalize this reasoning to $m > 1$?

Example



Difficulty

- In general, *S* matrix has a dimension of *n*, very high. However, $\operatorname{rank}(S) \le N - 1$ ← (Usually much less than *n*)
- Solution
  - Define matrix *A* as:

$$A = \left(\mathbf{x}_1 - \boldsymbol{\mu} \middle| \mathbf{x}_2 - \boldsymbol{\mu} \middle| \cdots \middle| \mathbf{x}_N - \boldsymbol{\mu} \right)$$

(*n* x *N*)

$$S = AA^T$$

$$S\mathbf{w}_i = \lambda_i \mathbf{w}_i$$

  - Compute the eigenvector/eigenvalues of lowest dimension matrix *R*

$$R = A^T A$$

$$R\mathbf{v}_i = \varepsilon_i \mathbf{v}_i$$

  - Relate the two sets

$$A^T A\mathbf{v}_i = \varepsilon_i \mathbf{v}_i$$

$$AA^T A\mathbf{v}_i = \varepsilon_i A\mathbf{v}_i$$

$$S(A\mathbf{v}_i) = \varepsilon_i (A\mathbf{v}_i)$$

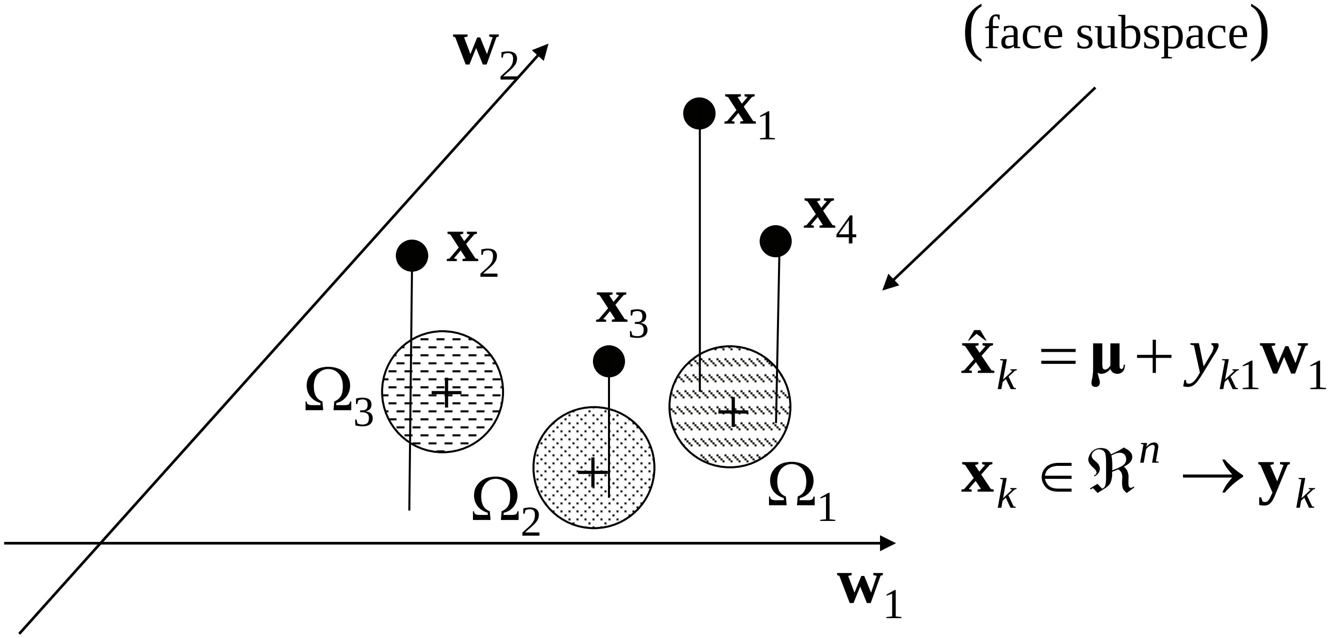
$$\left\{ \begin{array}{l} \mathbf{w}_i = A\mathbf{v}_i \\ \lambda_i = \varepsilon_i \end{array} \right.$$

Eigenfaces Algorithm

- Prepare the training set consisting of N faces, $\mathbf{x}_1 \ldots \mathbf{x}_N$, properly aligned
- Determine the mean face $\boldsymbol{\mu}$
- Define matrix A (size of $n \times N$) whose columns contain the "AC components" of the faces from the training set
- Compute the eigenvectors and eigenvalues of matrix $R = A^T A$ (size of $N \times N$)
- Select m (maximum of N-1) eigenvectors from R, associated to the highest eigenvalues. Define matrix V ($N \times m$), formed by the m eigenvectors of R
- Get matrix W ($n \times m$) by the relation $W = AV$ (not forgetting that the W columns form an orthonormal basis)
- Develop a face classifier (e.g., nearest neighbor) based on models trained with the observations (feature vectors) of the projections in the face subspace, $\mathbf{y} = W^T(\mathbf{x}-\boldsymbol{\mu})$

Face recognition – Illustration

- 2 *eigenfaces* – $\mathbf{w}_1$,$\mathbf{w}_2$
- 3 classes (known persons, Ω_1,..Ω_3)
- 4 faces to classify ($\mathbf{x}_1$,…,$\mathbf{x}_4$)

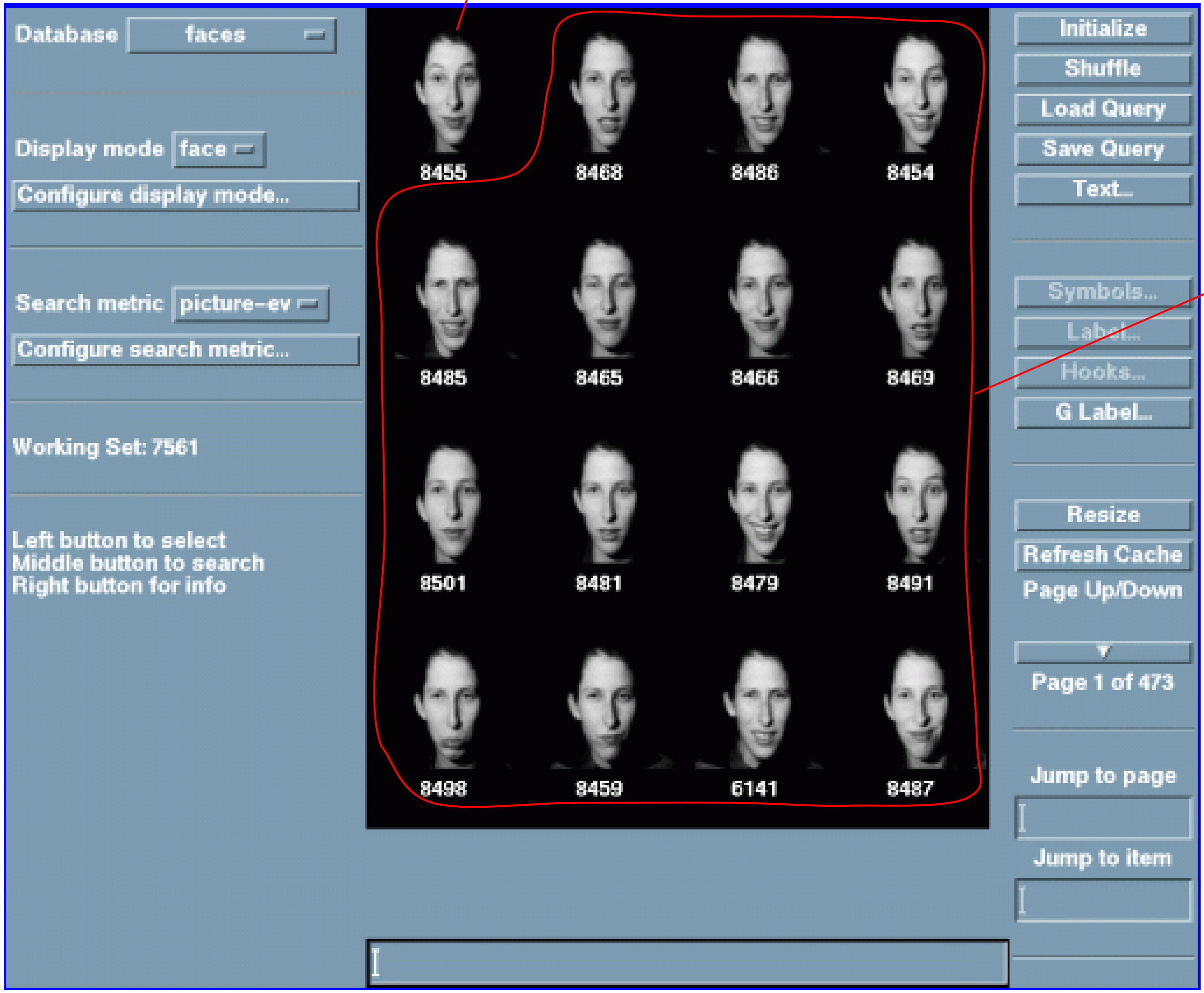


$$\hat{\mathbf{x}}_k = \boldsymbol{\mu} + y_{k1}\mathbf{w}_1 + y_{k2}\mathbf{w}_2$$

$$\mathbf{x}_k \in \Re^n \rightarrow \mathbf{y}_k \in \Re^m$$

Application example: Face retrieval from a database

Query face


Database faces

Display mode face

Search metric picture–ev

Working Set: 7561

Left button to select
Middle button to search
Right button for info

Page 1 of 473

The 15 most similar faces in a universe of 7562 faces

PCA disadvantages

- The PCA method is used to find a set of “optimal” directions in order to perform an efficient data **representation**;

- However, the directions that are useful to **represent** may not be the best for **discriminating**

O Q O
Q O Q
Q O *O*

- The methods referred as Multi-Discriminant Analysis (MDA) try to find directions useful to perform an efficient discrimination

Problem

- Consider again the training set consisting of N vectors (faces)

$$\mathbf{x}_1, \ldots, \mathbf{x}_N \qquad \mathbf{x}_i \in \Re^n$$

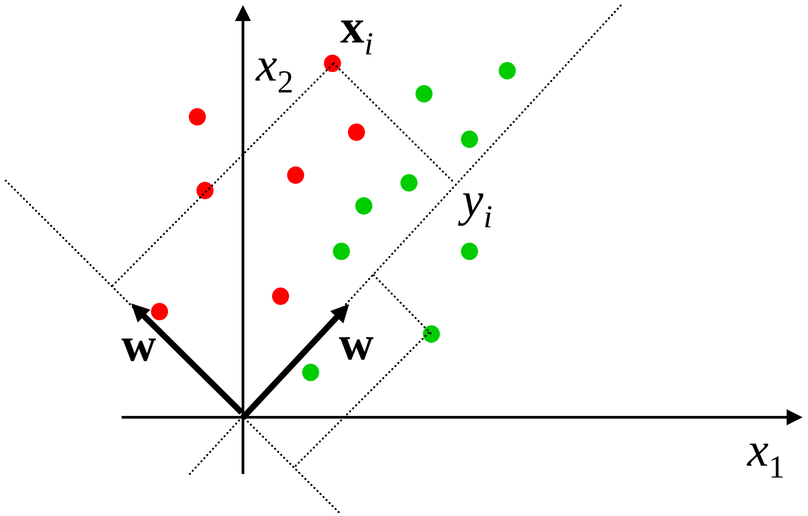
- However, consider now that the set is classified in 2 classes,
  - n_1 samples belong to class Ω_1
  - n_2 samples belong to class Ω_2

where $(n_1 + n_2 = N)$

- It is intended to project each of the N vectors into a given straight line (direction), obtaining the corresponding samples1d, $(y_1, \ldots, y_N)$

$$y_i = \mathbf{w}^T \mathbf{x}_i$$

$$\Re^n \to \Re$$



- **Question**: What is the "best" direction that separate (discriminate) samples from the two classes?

How to separate the clusters?

- **First idea**: Maximize the difference between the sample means after the projection
  - Sample means of the two classes $\boldsymbol{\mu}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in \Omega_i} \mathbf{x} \qquad i = 1,2$
  - Sample means after the projection $\tilde{\mu}_i = \frac{1}{n_i} \sum_{y \in \Psi_i} y$

$$= \frac{1}{n_i} \sum_{x \in \Omega_i} \mathbf{w}^T \mathbf{x} = \mathbf{w}^T \boldsymbol{\mu}_i$$

- Criteria to optimize

$$J(\mathbf{w}) = \left| \tilde{\mu}_1 - \tilde{\mu}_2 \right|$$

$$= \left| \mathbf{w}^T \left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2 \right) \right|$$

  - Does not work!

Fisher Criterion

- **Idea**: It is not enough just to separate the means of the two clusters; it is also necessary that the scatter measure round the mean should be reduced
  - *Scatter measure* (variance) $$\tilde{s}_i^2 = \sum_{y \in \Psi_i} \left(y - \tilde{\mu}_i \right)^2 \qquad i = 1,2$$

- Fisher Criterion $$J(\mathbf{w}) = \frac{\left| \tilde{\mu}_1 - \tilde{\mu}_2 \right|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$
  - Expressed explicitly as a function of **w** $$J(\mathbf{w}) = \frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_w \mathbf{w}}$$
  - Scatter Matrices

$$S_B = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \longleftarrow \text{Inter-classes}$$

$$S_i = \sum_{\mathbf{x} \in \Omega_i} (\mathbf{x} - \boldsymbol{\mu}_i)(\mathbf{x} - \boldsymbol{\mu}_i)^T$$

$$S_w = S_1 + S_2 \longleftarrow \text{Intra-classes}$$

Optimização

- Cost function to maximize

$$J(\mathbf{w}) = \frac{\mathbf{w}^T S_b \mathbf{w}}{\mathbf{w}^T S_w \mathbf{w}}$$

scalar values

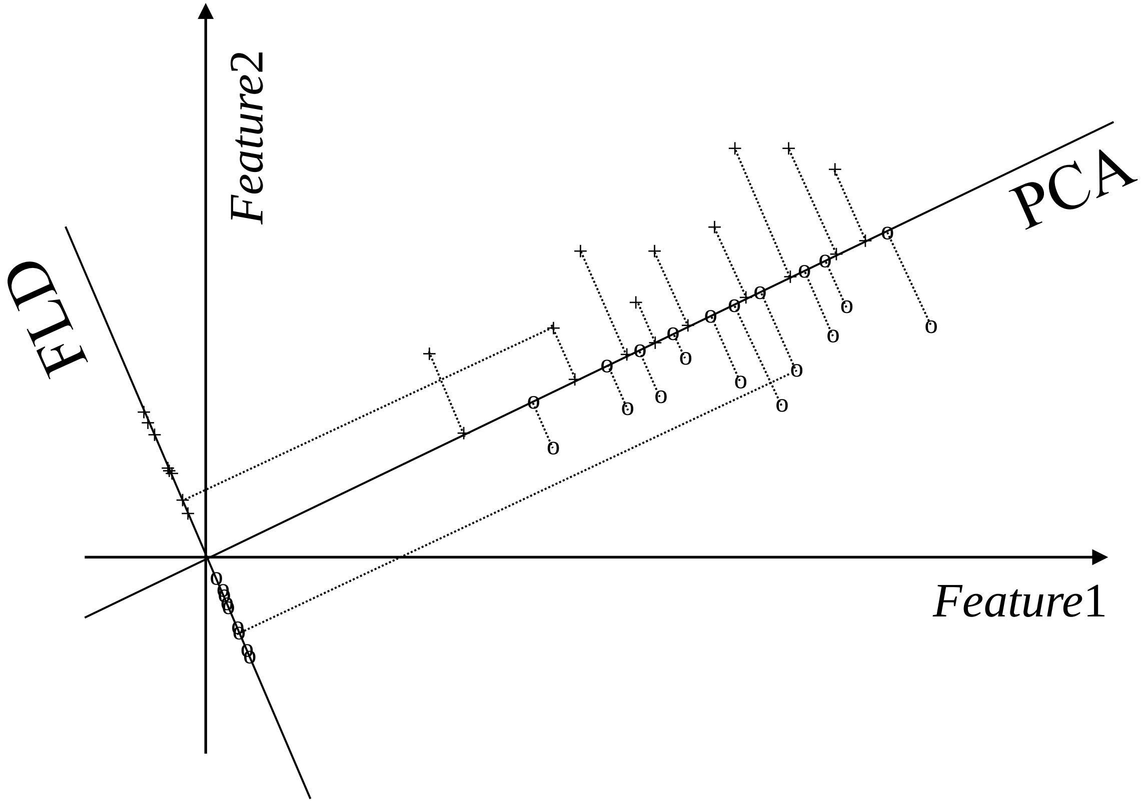
- Necessary condition

$$\frac{dJ}{d\mathbf{w}} = \frac{2S_b\mathbf{w}(\mathbf{w}^T S_w \mathbf{w}) - 2S_w\mathbf{w}(\mathbf{w}^T S_b \mathbf{w})}{(\mathbf{w}^T S_w \mathbf{w})^2} = 0$$

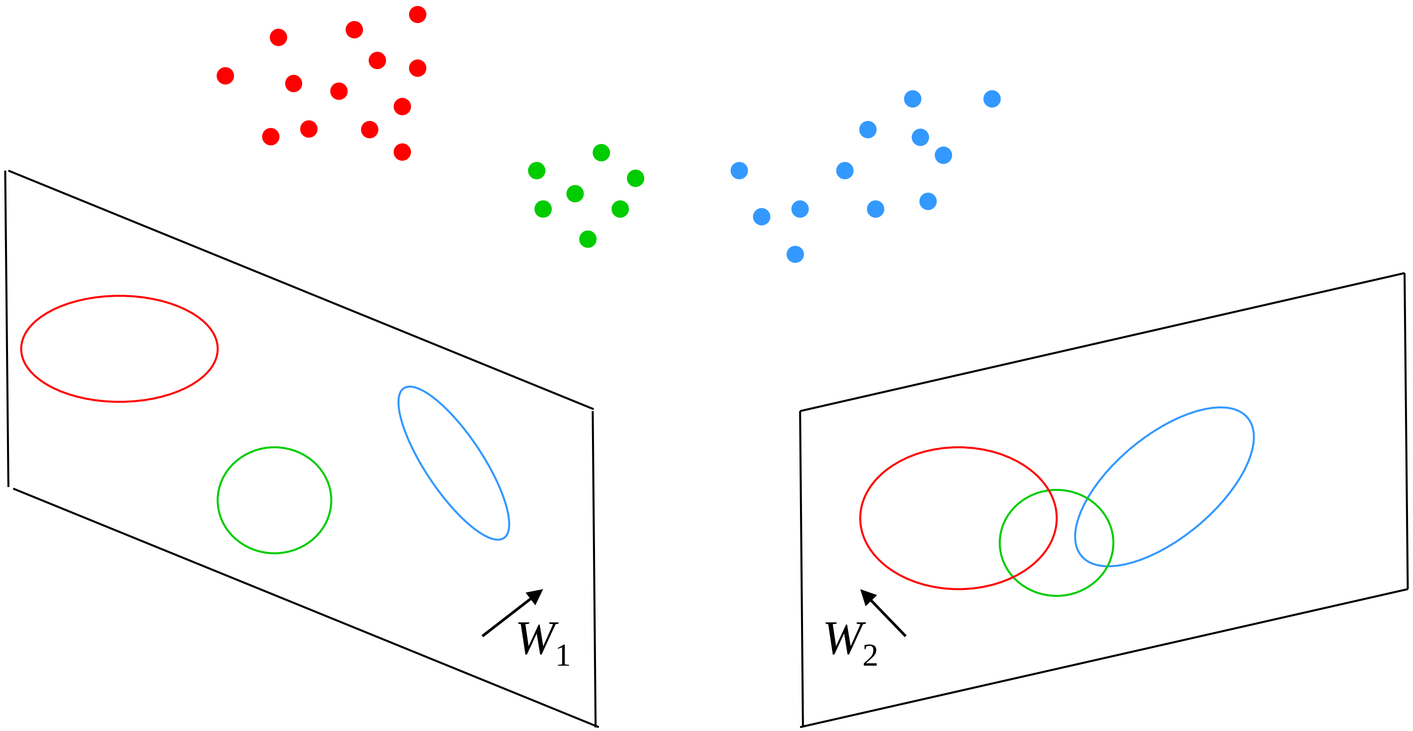
  - That is, $S_b\mathbf{w} = \lambda S_w \mathbf{w}$ Generalized eigenvalues/eigenvectors
  - Note that $S_b\mathbf{w} = \alpha(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$
  - **Solution** $\longrightarrow$ $\mathbf{w} = S_w^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$ S_w can not be singular!

Comparison PCA vs. MDA

- What is the best way to combine features?
- Example: Combine two features in one.
  - Criteria of minimizing the square error (PCA)
  - Criteria of the scatter ratio (inter and intra classes – FLD).



N-dimensional Example



Multi-Discriminant Analysis (MDA)

- Number of classes, *c*, is greater than 2
  - The aim is to obtain *c*-1 discriminant functions

$$\Re^n \rightarrow \Re^{c-1}$$

- The total scatter is given by $S_T = \sum_{\mathbf{x}} (\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^T$

- S_T can be decomposed into intra (S_w) and inter (S_b) scatter matrices

$$S_T = \sum_{i=1}^{c} \sum_{x \in \Omega_i} (\mathbf{x}-\boldsymbol{\mu}_i+\boldsymbol{\mu}_i-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu}_i+\boldsymbol{\mu}_i-\boldsymbol{\mu})^T$$

$$= \sum_{i=1}^{c} \sum_{x \in \Omega_i} (\mathbf{x}-\boldsymbol{\mu}_i)(\mathbf{x}-\boldsymbol{\mu}_i)^T + \sum_{i=1}^{c} \sum_{x \in \Omega_i} (\boldsymbol{\mu}_i-\boldsymbol{\mu})(\boldsymbol{\mu}_i-\boldsymbol{\mu})^T$$

$$= S_w + \underbrace{\sum_{i=1}^{c} n_i (\boldsymbol{\mu}_i-\boldsymbol{\mu})(\boldsymbol{\mu}_i-\boldsymbol{\mu})^T}_{S_b}$$

$$S_w = \sum_{i=1}^{c} S_i$$

$$S_i = \sum_{\mathbf{x} \in \Omega_i} (\mathbf{x}-\boldsymbol{\mu}_i)(\mathbf{x}-\boldsymbol{\mu}_i)^T$$

Relationships - before and after the projection

- **Goal**: Determine the scatter matrix after the projection

$$\left(\Re^{n} \rightarrow \Re^{c-1}\right) \qquad W = \left(\mathbf{w}_1 \middle| \mathbf{w}_2 \middle| \ldots \middle| \mathbf{w}_{c-1}\right)$$

$$\mathbf{y} = W^T \mathbf{x}$$

n x c-1

- Define

$$\tilde{S}_w = \sum_{i=1}^{c} \sum_{\mathbf{y} \in \Psi_i} (\mathbf{y} - \tilde{\boldsymbol{\mu}}_i)(\mathbf{y} - \tilde{\boldsymbol{\mu}}_i)^T \qquad \tilde{\boldsymbol{\mu}}_i = \frac{1}{n_i} \sum_{\mathbf{y} \in \Psi_i} \mathbf{y}$$

$$\tilde{S}_b = \sum_{i=1}^{c} n_i (\tilde{\boldsymbol{\mu}}_i - \tilde{\boldsymbol{\mu}})(\tilde{\boldsymbol{\mu}}_i - \tilde{\boldsymbol{\mu}})^T \qquad \tilde{\boldsymbol{\mu}} = \frac{1}{N} \sum_{i=1}^{c} n_i \tilde{\boldsymbol{\mu}}_i$$

- The following relations are obtained,

$$\tilde{S}_w = W^T S_w W \qquad \tilde{S}_b = W^T S_b W$$

Optimization

- Generalizing the above procedure (case c = 2), for the case of multiple classes, the following optimization criterion is obtained

$$J(W) = \frac{\left|W^T S_b W\right|}{\left|W^T S_w W\right|} \longleftarrow$$

Determinants ratio

- Necessary condition, which must comply with the solution

$$S_b \mathbf{w}_i = \lambda S_w \mathbf{w}_i$$

- **Difficulty**
  - In the problem of the faces, the scatter matrix, S_w is singular
  - At best, there are N-c non-zero eigenvalues

$$\mathrm{rank}(S_w) \le N - c << n$$

Fisherfaces Algorithm

- The solution called *fisherfaces* involves 2 steps:

1. Using the PCA method to obtain an intermediate subspace

$$\Re^n \rightarrow \Re^{N-c}$$

$$\mathbf{y}_{pca} = W_{pca}^T \mathbf{x} \qquad W_{pca} = \arg\max_W \left| W^T S_T W \right|$$

2. Then, uses the MDA method to obtain the *c*-1 discriminant directions

$$\Re^{N-c} \rightarrow \Re^{c-1}$$

$$\begin{aligned} \mathbf{y}_{fld} &= W_{fld}^T \mathbf{y}_{pca} \\ &= W_{fld}^T W_{pca}^T \mathbf{x} \end{aligned} \qquad W_{fld} = \arg\max_W \frac{\left| W^T W_{pca}^T S_b W_{pca} W \right|}{\left| W^T W_{pca}^T S_w W_{pca} W \right|}$$

Fisherfaces Algorithm

- Given the faces $\mathbf{x}_1 \ldots \mathbf{x}_N$, properly aligned and classified into class c (i = 1,..,c), each with n_i elements
- Determine the mean face $\boldsymbol{\mu}$, and the mean face of each class $\boldsymbol{\mu}_i$
- Determine the scatter matrix S_T (n x n)
  - Determine m (maximum of N-c) non-zero eigenvectors, $\mathbf{W}_{pca}$ (see algorithm *eigenfaces*)
- Determine the S_b (n x n) and S_w (n x n) matrices
- Compute

$$\breve{S}_b = W_{pca}^T S_b W_{pca}$$
$$\breve{S}_w = W_{pca}^T S_w W_{pca}$$

- Determine the c-1 “larger eigenvectors” from the matrix

$$\breve{S}_w^{-1} \breve{S}_b \longleftarrow \quad m \text{ x } m$$