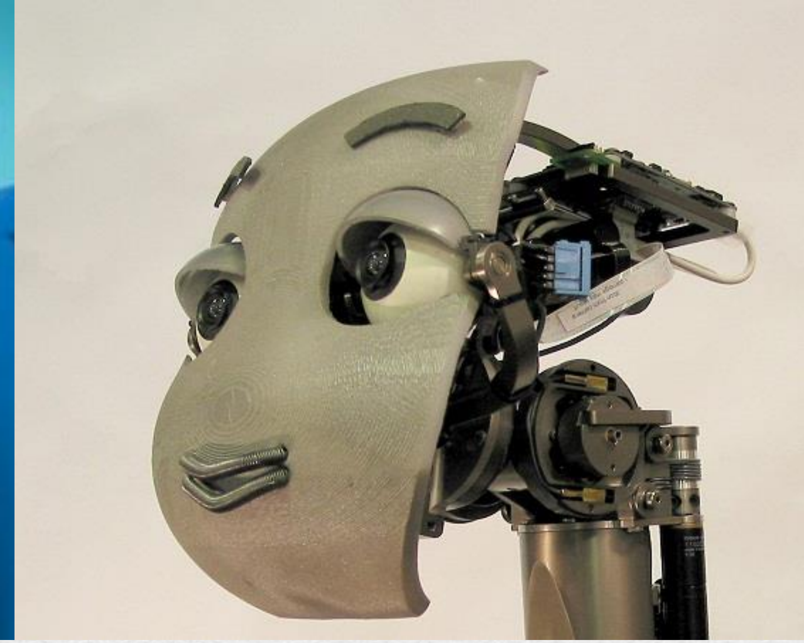



# THỊ GIÁC MÁY TÍNH …

## COMPUTER VISION

Ms. HTHT

Phone Number: 0989 567 488

# Chương 4. Phân tích thành phần chính (PCA) và phân tích khác biệt tuyến tính (LDA)

1. Phân tích thành phần chính PCA
2. Phân tích khác biệt tuyến tính LDA

# 4.1. Phân tích thành phần chính PCA (Principal Component Analysis )

# Một số khái niệm thống kê liên quan

1) Độ lệch chuẩn

Độ lệch chuẩn (Standard Deviation) là "Khoảng cách trung bình từ giá trị trung bình của tập dữ liệu tới một điểm" . Vậy độ lệch chuẩn = $\sqrt{\sigma}$

$$\sigma^2 = \frac{\sum_{i=1}^{N}(x_i - \mu)^2}{N} = V(X)$$

µ: giá trị trung bình của tập dữ liệu

$$\mu = \frac{\sum_{i=1}^{N} x_i}{N} = \frac{x_1 + x_2 + \ldots + x_N}{N} = E(X)$$

## 2) Phương sai

Phương sai của một bảng số liệu là số đặc trưng cho <u>độ phân tán</u> của các số liệu so với số trung bình của nó.

$$\sigma^2 = \frac{\sum_{i=1}^{N}\left(x_i - \mu\right)^2}{N} = V\left(X\right)$$

# VD: Tính phương sai và độ lệch chuẩn của 2 dãy số sau:

A= [6, 7, 8, 4, 5, 6]

B= [10, 2, 3, 9]

A: 6, 7, 8, 4, 5, 6

B: 10, 2, 3, 9

$$\overline{x_A} = 6$$

$$\overline{x_B} = 6$$

Độ lệch của mỗi số liệu so với số trung bình là:

Của A: (6-6) ; (7-6) ; (8-6) ; (4-6) ; (5-6) ; (6-6)

Của B: (10-6) ; (2-6) ; (3-6) ; (9-6)

Phương sai kí hiệu là $s^2$

$$s_A^2 = \frac{(6-6)^2 + (7-6)^2 + (8-6)^2 + (4-6)^2 + (5-6)^2 + (6-6)^2}{6} \approx 1{,}29$$

$$s_B^2 = \frac{(10-6)^2 + (2-6)^2 + (3-6)^2 + (9-6)^2}{4} \approx 3{,}53$$

3) Hiệp phương sai:

Được áp dụng cho nhiều chiều thay vì một chiều như độ lệch chuẩn và phương sai:

$$\mathrm{Var(X)} = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(X_i - \bar{X})}{(n-1)}$$

$$\mathrm{Cov(X,Y)} = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)}$$

4) Ma trận hiệp phương sai

Hiệp phương sai luôn được tính giữa hai chiều. Nếu chúng ta có một tập dữ liệu nhiều hơn hai chiều thì sẽ có nhiều hơn 1 hiệp phương sai được tính.

+ Với 3 chiều thì sẽ có 3 hiệp phương sai

+ Với n chiều thì sẽ có n(n-1)/2 hiệp phương sai

Để thuận tiện lấy tất cả các giá trị hiệp phương sai có thể giữa các chiều khác nhau, tính toán và đưa vào một ma trận. Nếu số chiều của tập dữ liệu là n, ta có ma trận vuông cấp n như sau:

$$C^{nxn} = \left(c_{i,j}, c_{i,j}\right) = cov(dim_i, dim_j)$$

$dim_{i,j}$ là chiều thứ i

# Phân tích thành phần chính (Principal Component Analysis )

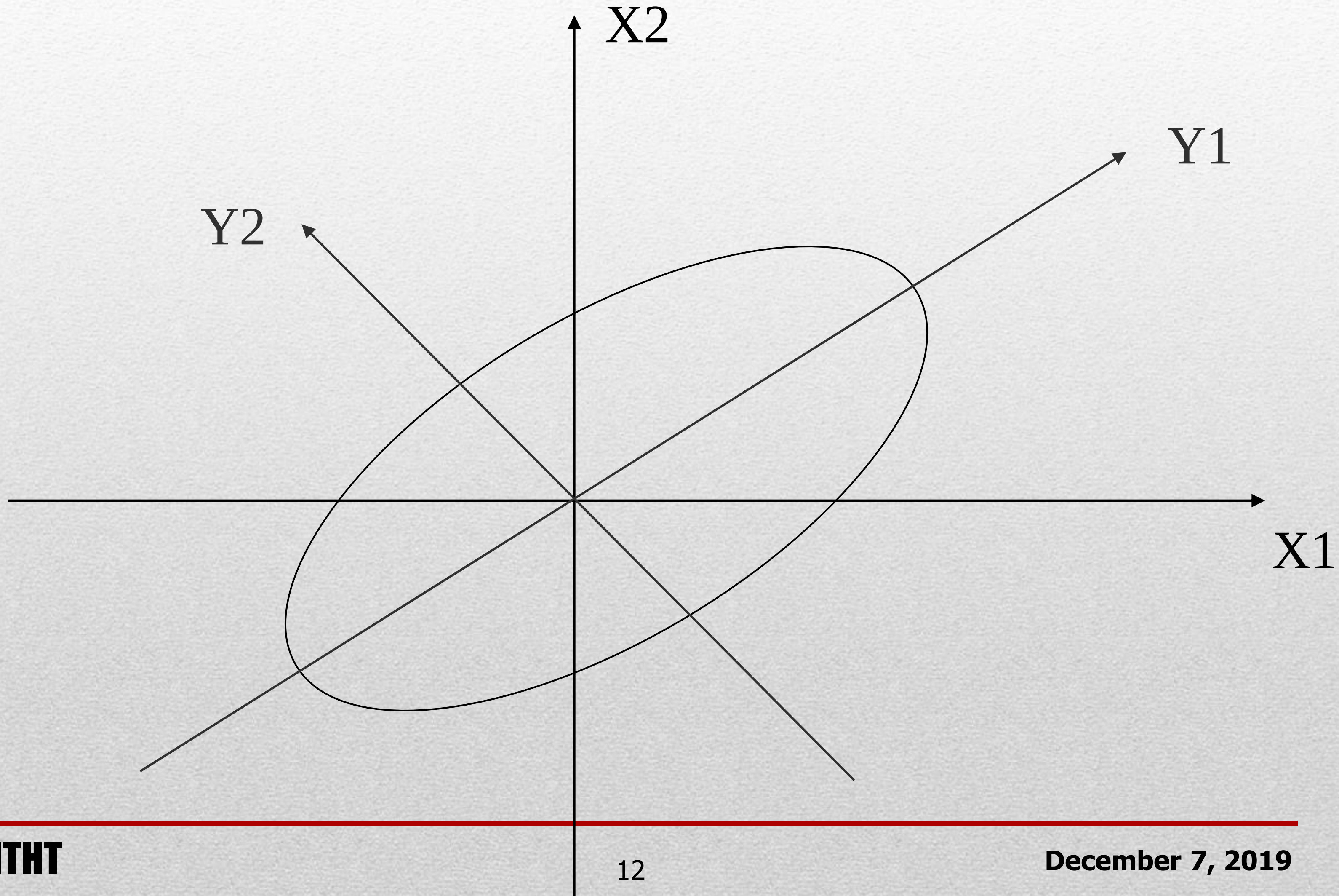
Cho *N* vector dữ liệu *k*-chiều, tìm *c* (<= *k)* vector trực giao tốt nhất để trình diễn dữ liệu.

- Tập dữ liệu gốc được rút gọn thành N vector dữ liệu c chiều: *c thành phần chính* (chiều được rút gọn).

Mỗi vector dữ liệu là tổ hợp tuyến tính của các vector thành phần chính.

- *Chỉ áp dụng cho dữ liệu số*.
- *Dùng khi số chiều vector lớn.*

# Phân tích thành phần chính (PCA)



# Rút gọn kích thước số

Phương pháp tham số

- *Giả sử dữ liệu phù hợp với mô hình nào đó*, ước lượng tham số mô hình, lưu chỉ các tham số, và không lưu dữ liệu (ngoại trừ các ngoại lai có thể có)
- Mô hình tuyến tính loga (Log-linear models): lấy giá trị tại một điểm trong không gian M-chiều như là tích của các không gian con thích hợp

Phương pháp không tham số

- Không giả thiết mô hình
- Tập hợp chính: biểu đồ (histograms), phân cụm (clustering), lấy mẫu (sampling)

# Làm thế nào để tìm các thành phần chính?

**Bước 1**: lấy vài số liệu

**Bước 2**: lấy hiệu các giá trị trung bình các biến số

**Bước 3**: tìm các vectơ riêng và các giá trị riêng của ma trận đồng phương sai

**Bước 4**: tìm các thành phần chính bằng cách chiếu các quan sát lên các vectơ riêng

**Bước 5**: tính toán các tải (loading) chẳng hạn như sự tương quan giữa các biến số gốc và các thành phần chính

# Ví dụ 2D: bước 1 lấy số liệu

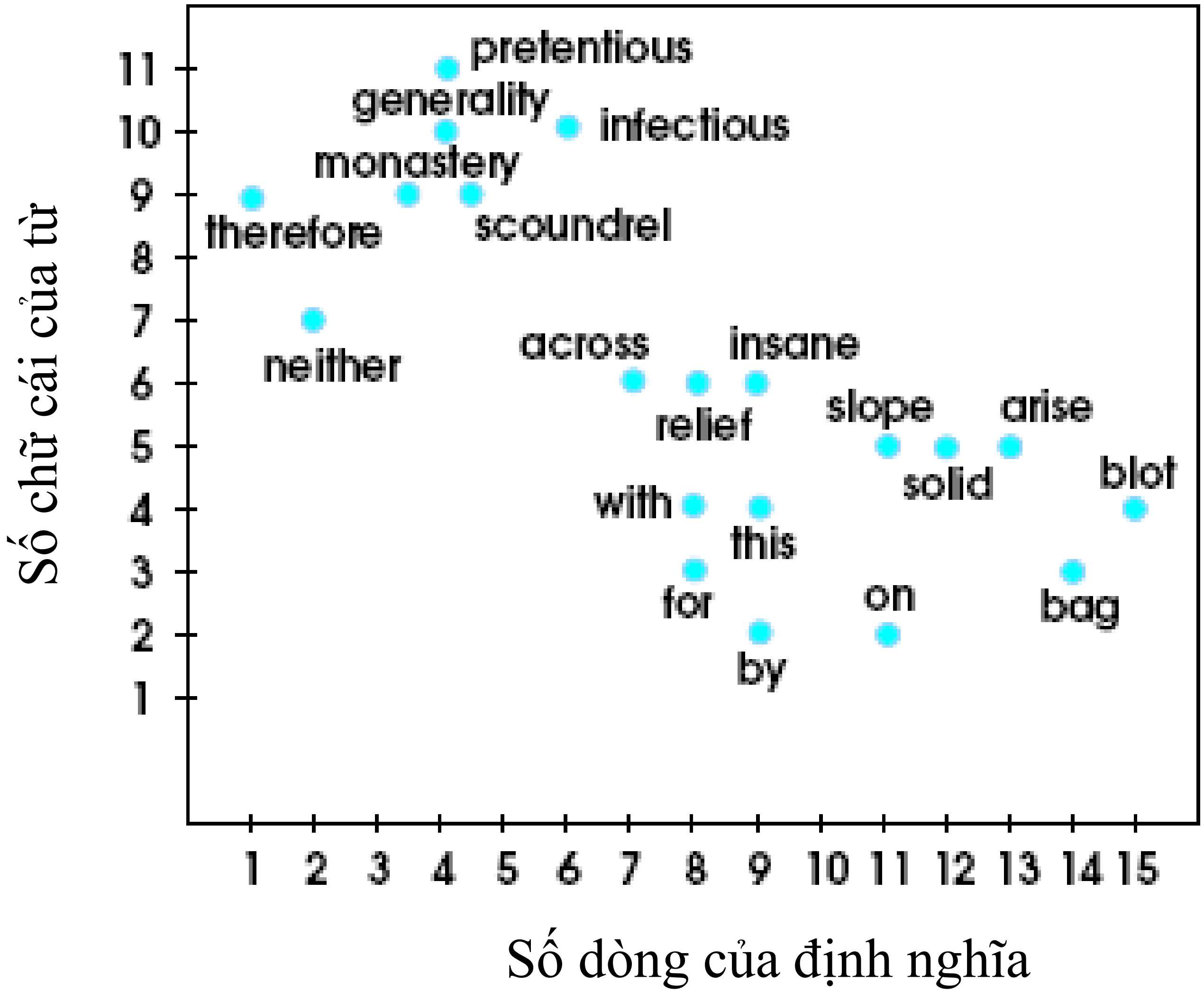
20 từ :

Biến 1 = số chữ cái

Biến 2 = số dòng dùng để định nghĩa từ trong từ điển

| Word | Length | Number of Lines |
|---|---|---|
| bag | 3 | 14 |
| across | 6 | 7 |
| on | 2 | 11 |
| insane | 6 | 9 |
| by | 2 | 9 |
| monastery | 9 | 4 |
| relief | 6 | 8 |
| slope | 5 | 11 |
| scoundrel | 9 | 5 |
| with | 4 | 8 |
| neither | 7 | 2 |
| pretentious | 11 | 4 |
| solid | 5 | 12 |
| this | 4 | 9 |
| for | 3 | 8 |
| therefore | 9 | 1 |
| generality | 10 | 4 |
| arise | 5 | 13 |
| blot | 4 | 15 |
| infectious | 10 | 6 |
| $\sum$ | 120 | 160 |
| M | 6 | 8 |

# Ví dụ 2D: bước 1 lấy số liệu



# Ví dụ 2D: bước 2 Lấy hiệu trung bình

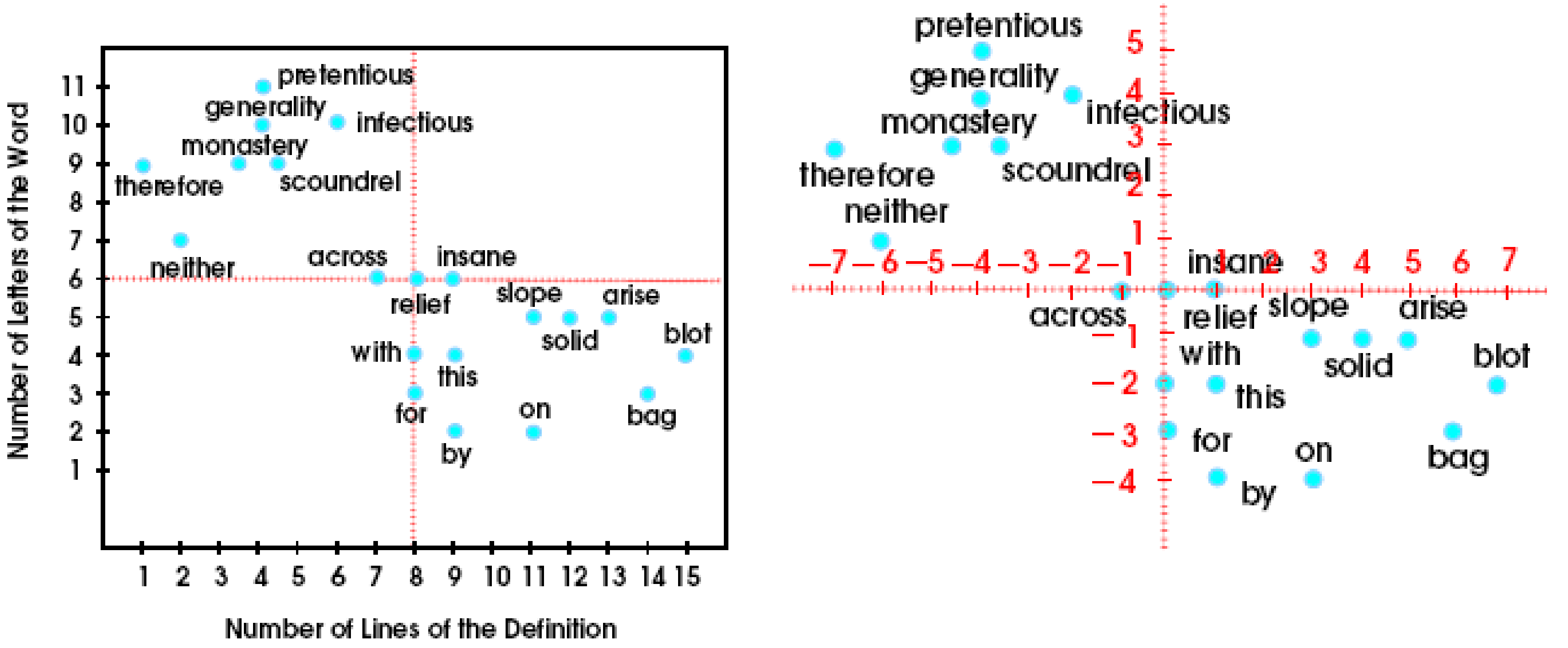
Y = “Chiều dài của từ ”
$M_Y = 6$
$y = (Y - M_Y)$

W = “Số dòng của định nghĩa”
$M_W = 8$
$w = (W - M_W)$

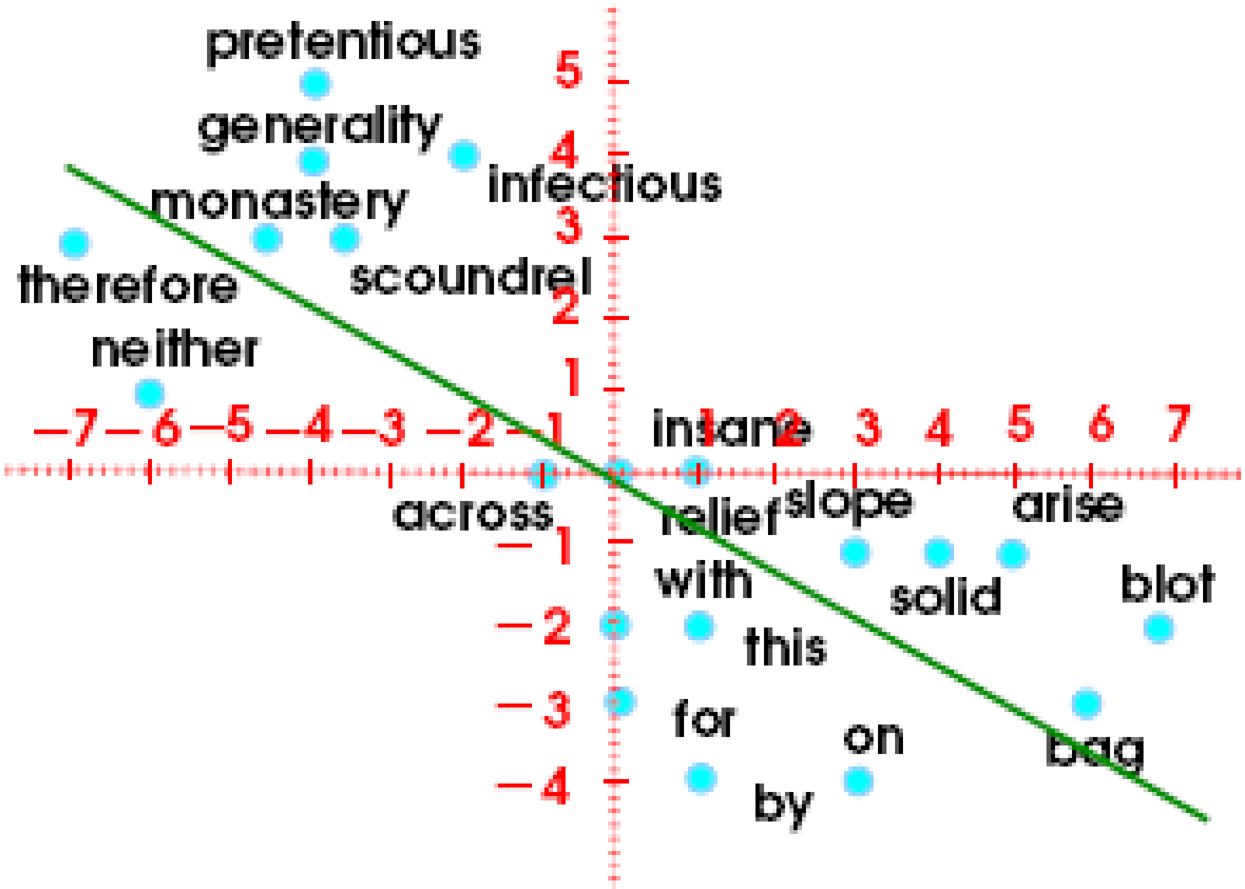
| Word | Length | Number of Lines |
|---|---|---|
| bag | 3 | 14 |
| across | 6 | 7 |
| on | 2 | 11 |
| insane | 6 | 9 |
| by | 2 | 9 |
| monastery | 9 | 4 |
| relief | 6 | 8 |
| slope | 5 | 11 |
| scoundrel | 9 | 5 |
| with | 4 | 8 |
| neither | 7 | 2 |
| pretentious | 11 | 4 |
| solid | 5 | 12 |
| this | 4 | 9 |
| for | 3 | 8 |
| therefore | 9 | 1 |
| generality | 10 | 4 |
| arise | 5 | 13 |
| blot | 4 | 15 |
| infectious | 10 | 6 |
| $\sum$ | 120 | 160 |
| M | 6 | 8 |

| Words | W | Y | w | y |
|---|---|---|---|---|
| bag | 14 | 3 | 6 | −3 |
| across | 7 | 6 | −1 | 0 |
| on | 11 | 2 | 3 | −4 |
| insane | 9 | 6 | 1 | 0 |
| by | 9 | 2 | 1 | −4 |
| monastery | 4 | 9 | −4 | 3 |
| relief | 8 | 6 | 0 | 0 |
| slope | 11 | 5 | 3 | −1 |
| scoundrel | 5 | 9 | −3 | 3 |
| with | 8 | 4 | 0 | −2 |
| neither | 2 | 7 | −6 | 1 |
| pretentious | 4 | 11 | −4 | 5 |
| solid | 12 | 5 | 4 | −1 |
| this | 9 | 4 | 1 | −2 |
| for | 8 | 3 | 0 | −3 |
| therefore | 1 | 9 | −7 | 3 |
| generality | 4 | 10 | −4 | 4 |
| arise | 13 | 5 | 5 | −1 |
| blot | 15 | 4 | 7 | −2 |
| infectious | 6 | 10 | −2 | 4 |
| $\sum$ | 160 | 120 | 0 | 0 |

# Ví dụ 2D: bước 2 Lấy hiệu trung bình

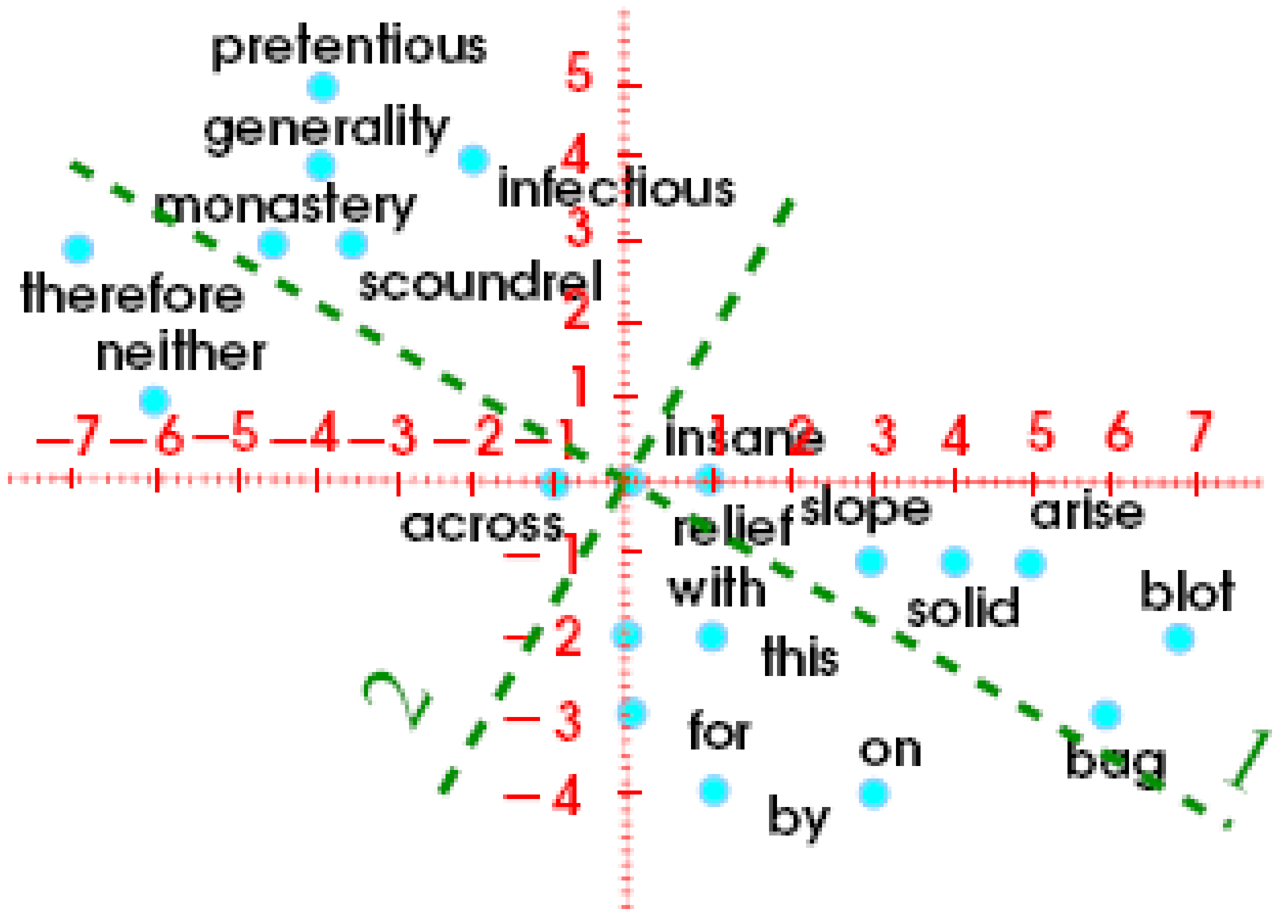


# Ví dụ 2D: bước 3 tìm các vectơ riêng

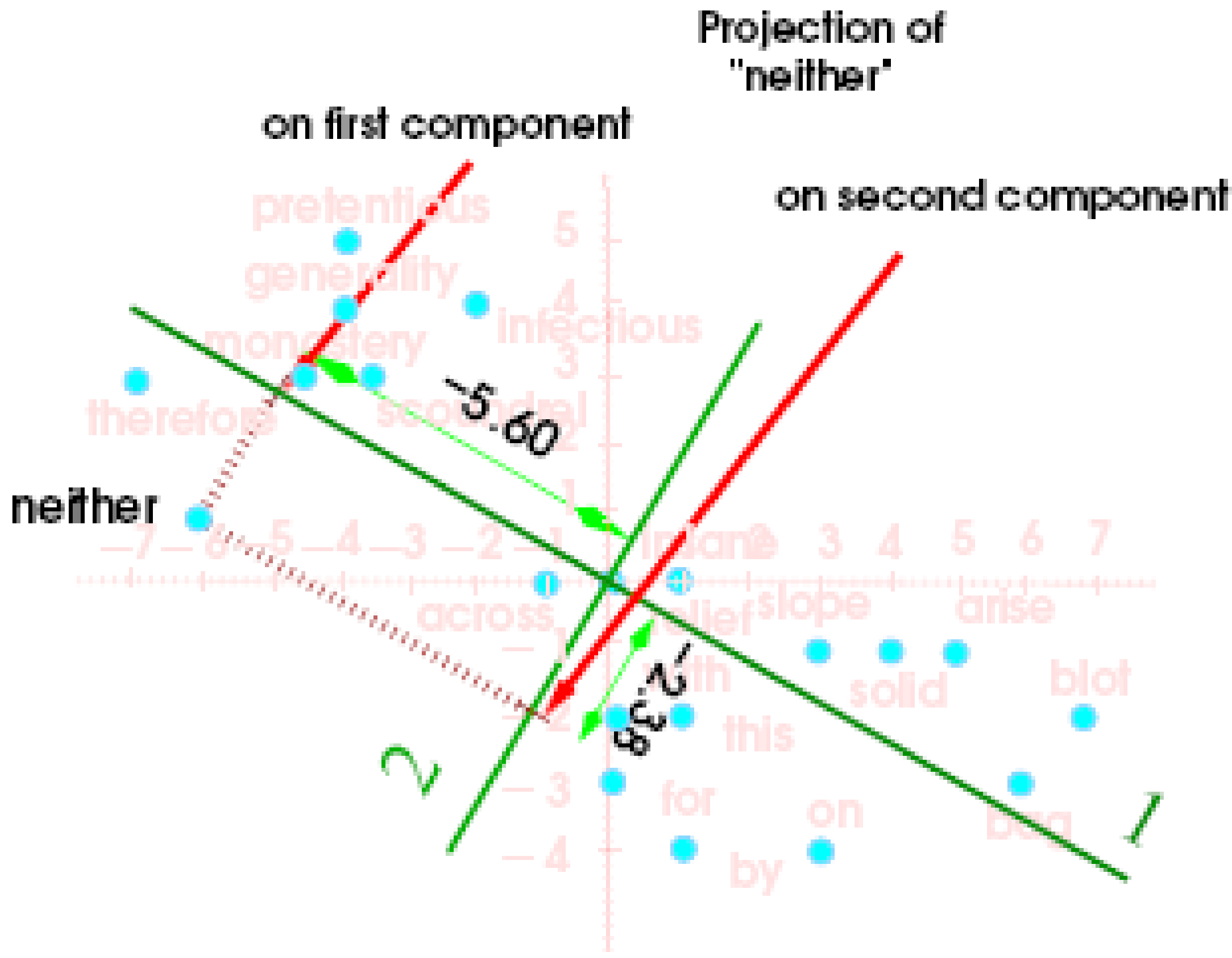



-

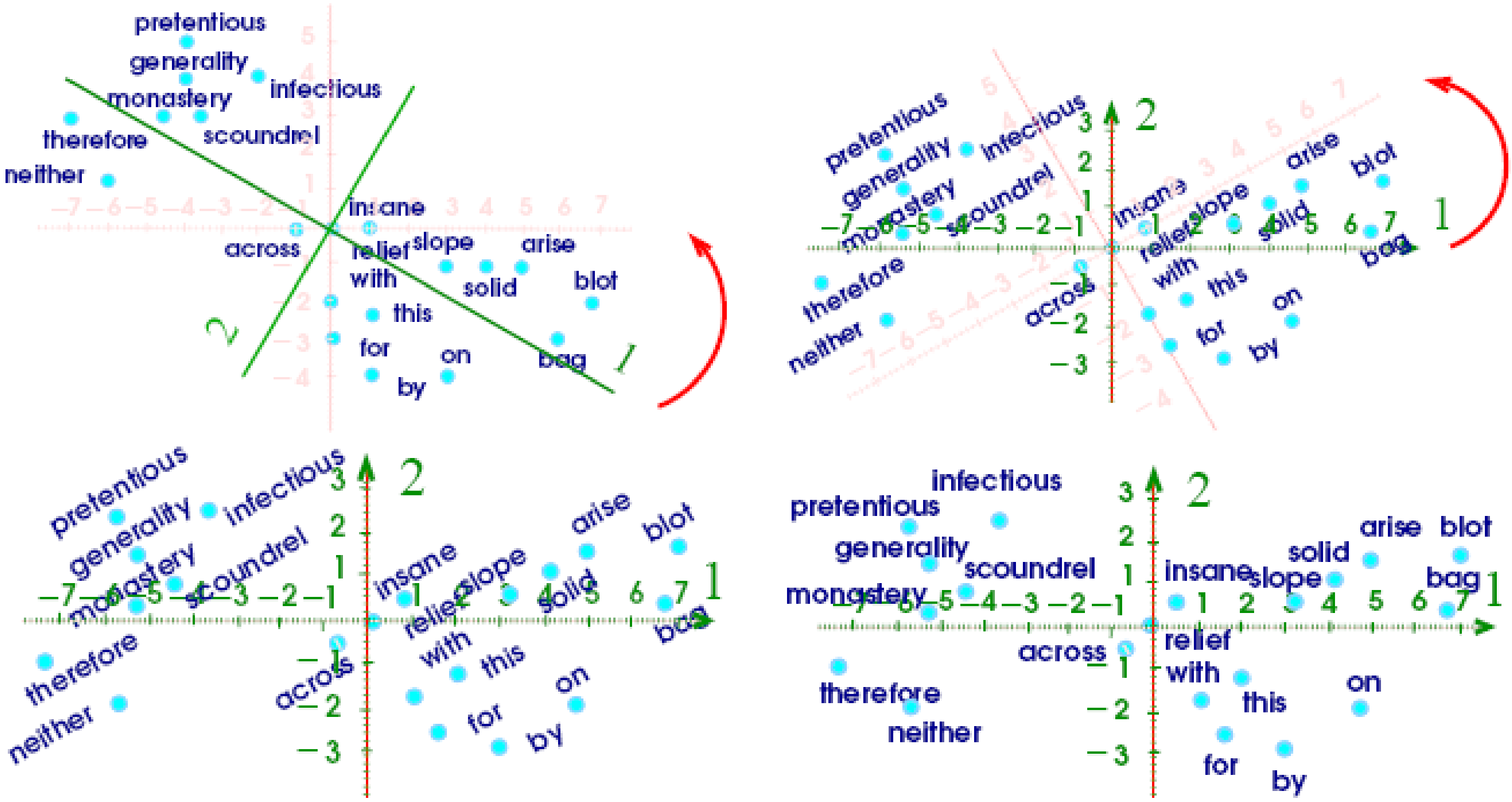
# Ví dụ 2D: bước 3 tìm các vectơ riêng



# Ví dụ 2D: chiếu các quan sát



# Ví dụ 2D: chiếu các quan sát



# Ví dụ 2D: tính toán các tải (loadings)

| Words | $W$ | $Y$ | $w$ | $y$ | $F_1$ | $F_2$ |
|---|---|---|---|---|---|---|
| bag | 14 | 3 | 6 | −3 | 6.67 | 0.69 |
| across | 7 | 6 | −1 | 0 | −0.84 | −0.54 |
| on | 11 | 2 | 3 | −4 | 4.68 | −1.76 |
| insane | 9 | 6 | 1 | 0 | 0.84 | 0.54 |
| by | 9 | 2 | 1 | −4 | 2.99 | −2.84 |
| monastery | 4 | 9 | −4 | 3 | −4.99 | 0.38 |
| relief | 8 | 6 | 0 | 0 | 0.00 | 0.00 |
| slope | 11 | 5 | 3 | −1 | 3.07 | 0.77 |
| scoundrel | 5 | 9 | −3 | 3 | −4.14 | 0.92 |
| with | 8 | 4 | 0 | −2 | 1.07 | −1.69 |
| neither | 2 | 7 | −6 | 1 | −5.60 | −2.38 |
| pretentious | 4 | 11 | −4 | 5 | −6.06 | 2.07 |
| solid | 12 | 5 | 4 | −1 | 3.91 | 1.30 |
| this | 9 | 4 | 1 | −2 | 1.92 | −1.15 |
| for | 8 | 3 | 0 | −3 | 1.61 | −2.53 |
| therefore | 1 | 9 | −7 | 3 | −7.52 | −1.23 |
| generality | 4 | 10 | −4 | 4 | −5.52 | 1.23 |
| arise | 13 | 5 | 5 | −1 | 4.76 | 1.84 |
| blot | 15 | 4 | 7 | −2 | 6.98 | 2.07 |
| infectious | 6 | 10 | −2 | 4 | −3.83 | 2.30 |
| $\sum$ | 120 | 160 | 0 | 0 | 0 | 0 |

Hệ số tương quan Pearson

$r\ (W, F_1) = 0.97$

# Ví dụ 2D: tính toán các tải (loadings)

| Words | W | Y | w | y | $F_1$ | $F_2$ |
|---|---|---|---|---|---|---|
| bag | 14 | 3 | 6 | −3 | 6.67 | 0.69 |
| across | 7 | 6 | −1 | 0 | −0.84 | −0.54 |
| on | 11 | 2 | 3 | −4 | 4.68 | −1.76 |
| insane | 9 | 6 | 1 | 0 | 0.84 | 0.54 |
| by | 9 | 2 | 1 | −4 | 2.99 | −2.84 |
| monastery | 4 | 9 | −4 | 3 | −4.99 | 0.38 |
| relief | 8 | 6 | 0 | 0 | 0.00 | 0.00 |
| slope | 11 | 5 | 3 | −1 | 3.07 | 0.77 |
| scoundrel | 5 | 9 | −3 | 3 | −4.14 | 0.92 |
| with | 8 | 4 | 0 | −2 | 1.07 | −1.69 |
| neither | 2 | 7 | −6 | 1 | −5.60 | −2.38 |
| pretentious | 4 | 11 | −4 | 5 | −6.06 | 2.07 |
| solid | 12 | 5 | 4 | −1 | 3.91 | 1.30 |
| this | 9 | 4 | 1 | −2 | 1.92 | −1.15 |
| for | 8 | 3 | 0 | −3 | 1.61 | −2.53 |
| therefore | 1 | 9 | −7 | 3 | −7.52 | −1.23 |
| generality | 4 | 10 | −4 | 4 | −5.52 | 1.23 |
| arise | 13 | 5 | 5 | −1 | 4.76 | 1.84 |
| blot | 15 | 4 | 7 | −2 | 6.98 | 2.07 |
| infectious | 6 | 10 | −2 | 4 | −3.83 | 2.30 |
| $\sum$ | 120 | 160 | 0 | 0 | 0 | 0 |

Hệ số tương quan Pearson

$$r\,(W, F_2) = 0.23$$

# Ví dụ 2D: tính toán các tải (loadings)

| Words | W | Y | w | y | $F_1$ | $F_2$ |
|---|---|---|---|---|---|---|
| bag | 14 | 3 | 6 | −3 | 6.67 | 0.69 |
| across | 7 | 6 | −1 | 0 | −0.84 | −0.54 |
| on | 11 | 2 | 3 | −4 | 4.68 | −1.76 |
| insane | 9 | 6 | 1 | 0 | 0.84 | 0.54 |
| by | 9 | 2 | 1 | −4 | 2.99 | −2.84 |
| monastery | 4 | 9 | −4 | 3 | −4.99 | 0.38 |
| relief | 8 | 6 | 0 | 0 | 0.00 | 0.00 |
| slope | 11 | 5 | 3 | −1 | 3.07 | 0.77 |
| scoundrel | 5 | 9 | −3 | 3 | −4.14 | 0.92 |
| with | 8 | 4 | 0 | −2 | 1.07 | −1.69 |
| neither | 2 | 7 | −6 | 1 | −5.60 | −2.38 |
| pretentious | 4 | 11 | −4 | 5 | −6.06 | 2.07 |
| solid | 12 | 5 | 4 | −1 | 3.91 | 1.30 |
| this | 9 | 4 | 1 | −2 | 1.92 | −1.15 |
| for | 8 | 3 | 0 | −3 | 1.61 | −2.53 |
| therefore | 1 | 9 | −7 | 3 | −7.52 | −1.23 |
| generality | 4 | 10 | −4 | 4 | −5.52 | 1.23 |
| arise | 13 | 5 | 5 | −1 | 4.76 | 1.84 |
| blot | 15 | 4 | 7 | −2 | 6.98 | 2.07 |
| infectious | 6 | 10 | −2 | 4 | −3.83 | 2.30 |
| $\sum$ | 120 | 160 | 0 | 0 | 0 | 0 |

Hệ số tương quan Pearson

$r(Y, F_1) = -0.87$

# Ví dụ 2D: tính toán các tải (loadings)

| Words | W | Y | w | y | $F_1$ | $F_2$ |
|---|---|---|---|---|---|---|
| bag | 14 | 3 | 6 | −3 | 6.67 | 0.69 |
| across | 7 | 6 | −1 | 0 | −0.84 | −0.54 |
| on | 11 | 2 | 3 | −4 | 4.68 | −1.76 |
| insane | 9 | 6 | 1 | 0 | 0.84 | 0.54 |
| by | 9 | 2 | 1 | −4 | 2.99 | −2.84 |
| monastery | 4 | 9 | −4 | 3 | −4.99 | 0.38 |
| relief | 8 | 6 | 0 | 0 | 0.00 | 0.00 |
| slope | 11 | 5 | 3 | −1 | 3.07 | 0.77 |
| scoundrel | 5 | 9 | −3 | 3 | −4.14 | 0.92 |
| with | 8 | 4 | 0 | −2 | 1.07 | −1.69 |
| neither | 2 | 7 | −6 | 1 | −5.60 | −2.38 |
| pretentious | 4 | 11 | −4 | 5 | −6.06 | 2.07 |
| solid | 12 | 5 | 4 | −1 | 3.91 | 1.30 |
| this | 9 | 4 | 1 | −2 | 1.92 | −1.15 |
| for | 8 | 3 | 0 | −3 | 1.61 | −2.53 |
| therefore | 1 | 9 | −7 | 3 | −7.52 | −1.23 |
| generality | 4 | 10 | −4 | 4 | −5.52 | 1.23 |
| arise | 13 | 5 | 5 | −1 | 4.76 | 1.84 |
| blot | 15 | 4 | 7 | −2 | 6.98 | 2.07 |
| infectious | 6 | 10 | −2 | 4 | −3.83 | 2.30 |
| $\sum$ | 120 | 160 | 0 | 0 | 0 | 0 |

Hệ số tương quan Pearson

$r\ (Y, F_2) = 0.50$

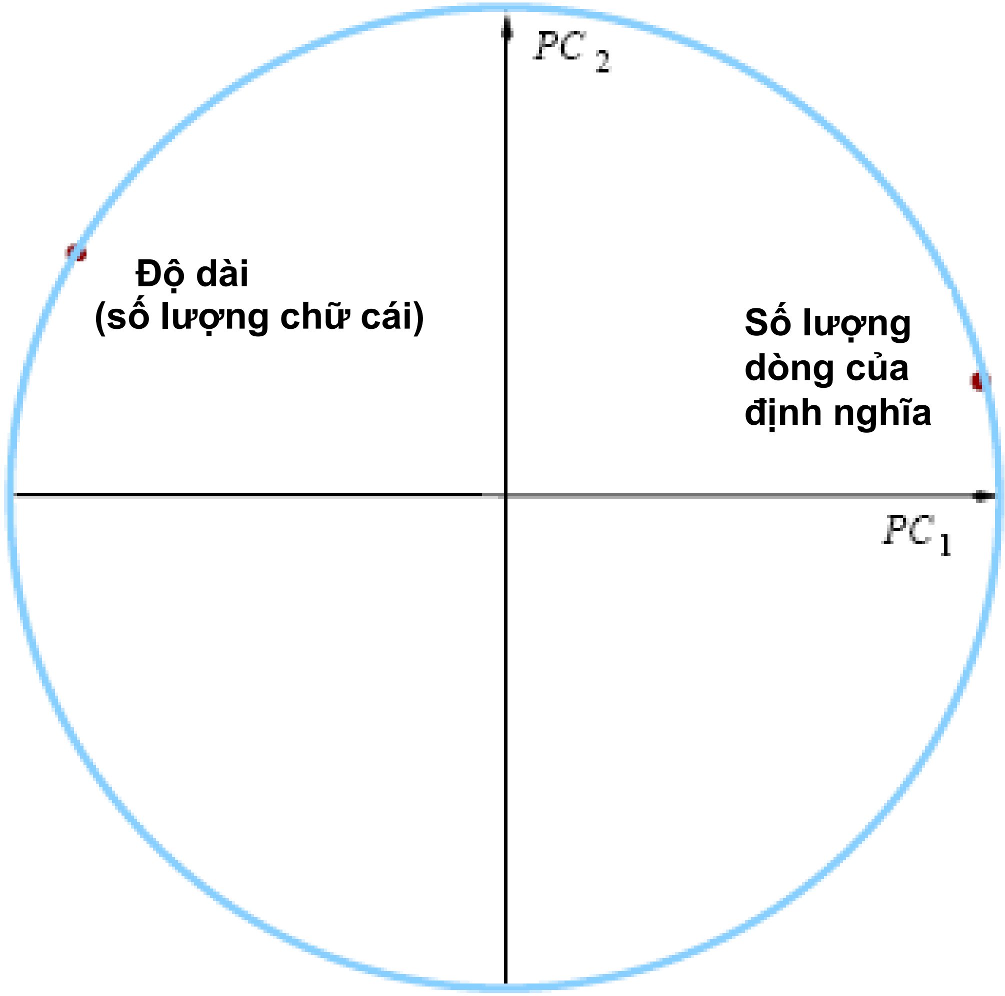
# Ví dụ 2D : vẽ vòng tròn tương quan

$r\ (W, F_1) = 0.97$

$r\ (W, F_2) = 0.23$

$r\ (Y, F_1) = -0.87$

$r\ (Y, F_2) = 0.50$



# Làm thế nào để tính phương sai explained variance ?

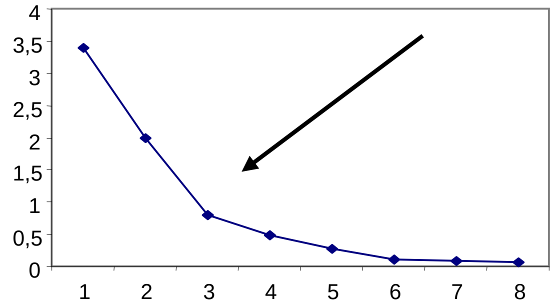
| Giá trị riêng | % phương sai | % phương sai tích luỹ |
|---|---|---|
| 392<br>52<br>444 | 88<br>12 | 88<br>100 |

$$\frac{392}{444} \text{ X } 100 = 88\%$$

# Giữ lại bao nhiêu thành phần

**Chuẩn Kaiser.** chỉ giữ lại thành phần có giá trị riêng lớn hơn 1

**The scree test.**




**Thông thường**. giữ lại những chiều mà có thể phân tích giải thích được

Khảo sát một vài giải pháp và chọn giải pháp “hợp lý nhất”

# Ta có nên chuẩn hoá số liệu ?

Có, nếu số liệu không được đo trên cùng một thang

$$Z_{Y_n} = \frac{Y_n - M_Y}{\sigma_Y}$$

Nếu không thì nó phụ thuộc vào:

Chuẩn hoá: cùng trọng lượng cho toàn bộ biến số

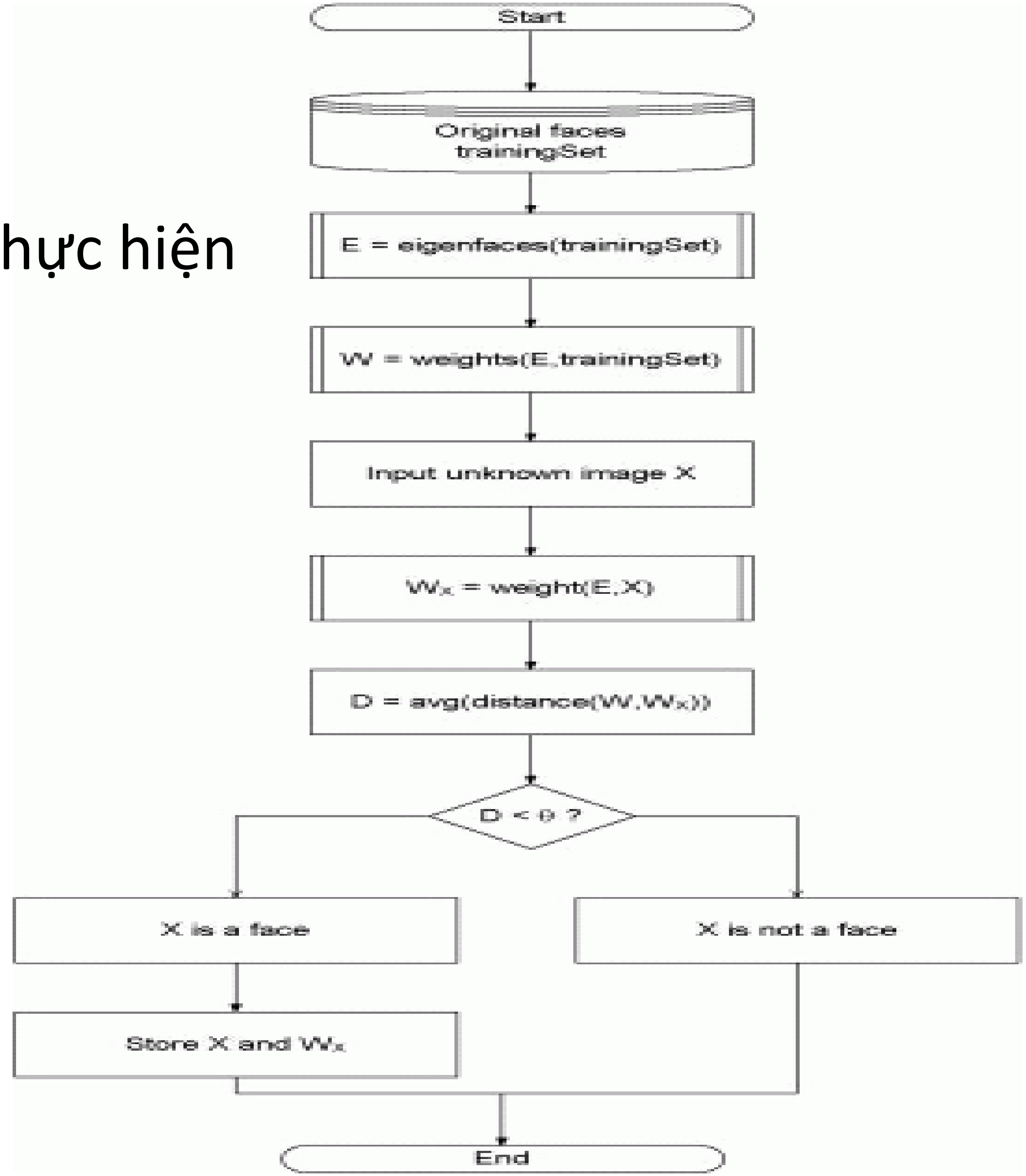
Không chuẩn hoá: trọng lượng tỷ lệ theo độ lệch chuẩn

# Mục tiêu

- Khảo sát ứng dụng nhận dang khuôn mặt theo phương pháp PCA để thấy được ứng dụng của phép biến đổi KL, PCA.

- Mục tiêu của ứng dụng: nhận dạng chính xác nhất khuôn mặt dựa trên những khuôn mặt đã có sẵn.

# Các bước thực hiện



# Các bước thực hiện

- Khởi tạo bao gồm các ảnh khuôn mặt.
- Tính toán các khuôn mặt riêng từ tập đã có, từ đó xác định không gian mặt – face space.
- Tính toán trọng số không gian của các nhóm khuôn mặt tương ứng trong cơ sở dữ liệu bằng cách chiếu lên không gian mặt.
- Tính toán tập trọng số của khuôn mặt cần nhận dạng bằng cách chiếu lên những khuôn mặt riêng đã có.
- Dựa vào trọng số, xác định có thuộc các nhóm khuôn mặt đã biết hay không.

# Tính toán các vectơ khuôn mặt riêng:

- Ý tưởng cơ bản là tìm những vectors có thể biểu diễn tốt nhất các đặc tính của khuôn mặt trong không gian ảnh (eigenvector). Các vector này xác định một không gian khuôn mặt riêng (eigenface space). Những vector này là vector riêng của ma trận hiệp phương sai , được gọi là các khuôn mặt riêng.
- Sau khi có M trị riêng, chỉ giữ lại M’ (< M) trị riêng lớn nhất.
- Các vectơ riêng ứng với các trị riêng có giá trị lớn mô tả các đặc trưng tốt hơn các trị riêng nhỏ.

# Tính toán các vectơ khuôn mặt riêng:

- Tính ma trận hiệp phương sai
  ($A = [\Phi_1\ \Phi_2 \ldots \Phi_M]$)

$$C = \frac{1}{M}\sum_{n=1}^{M} \Phi_n \Phi_n^T = AA^T$$

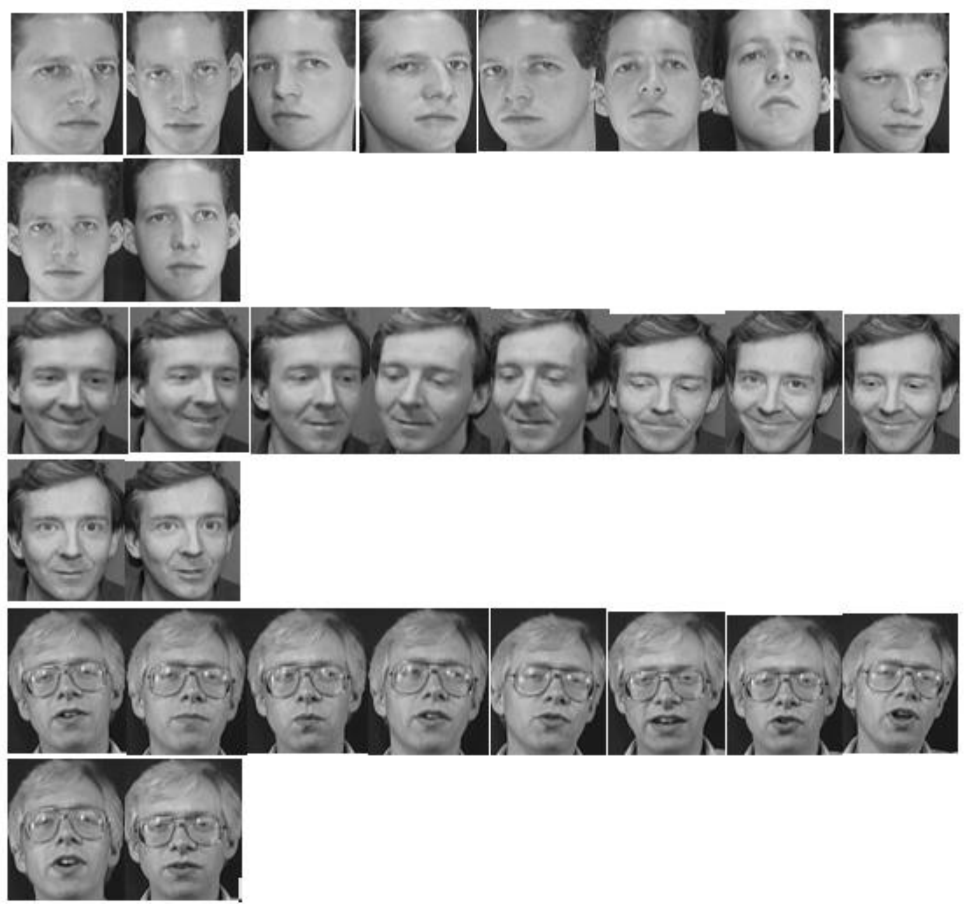
- $A^TAv_i = \mu_i v_i$ ($\mu_i$ là các trị riêng của $A^TA$)
- $Av_i$ là vector riêng của $C = AA^T$. Nhân 2 vế của phương trình với ma trận A

$$A\,A^TAv_i = A\mu_i v_i = \mu_i A v_i$$

- Như vậy, bài toán quy về việc tính các trị riêng $\mu_i$ và véctơ riêng $v_i$ của ma trận $A^TA$. Véctơ riêng của ma trận $C=AA^T$ là $u_i = Av_i$

# Tính toán các vectơ khuôn mặt riêng:

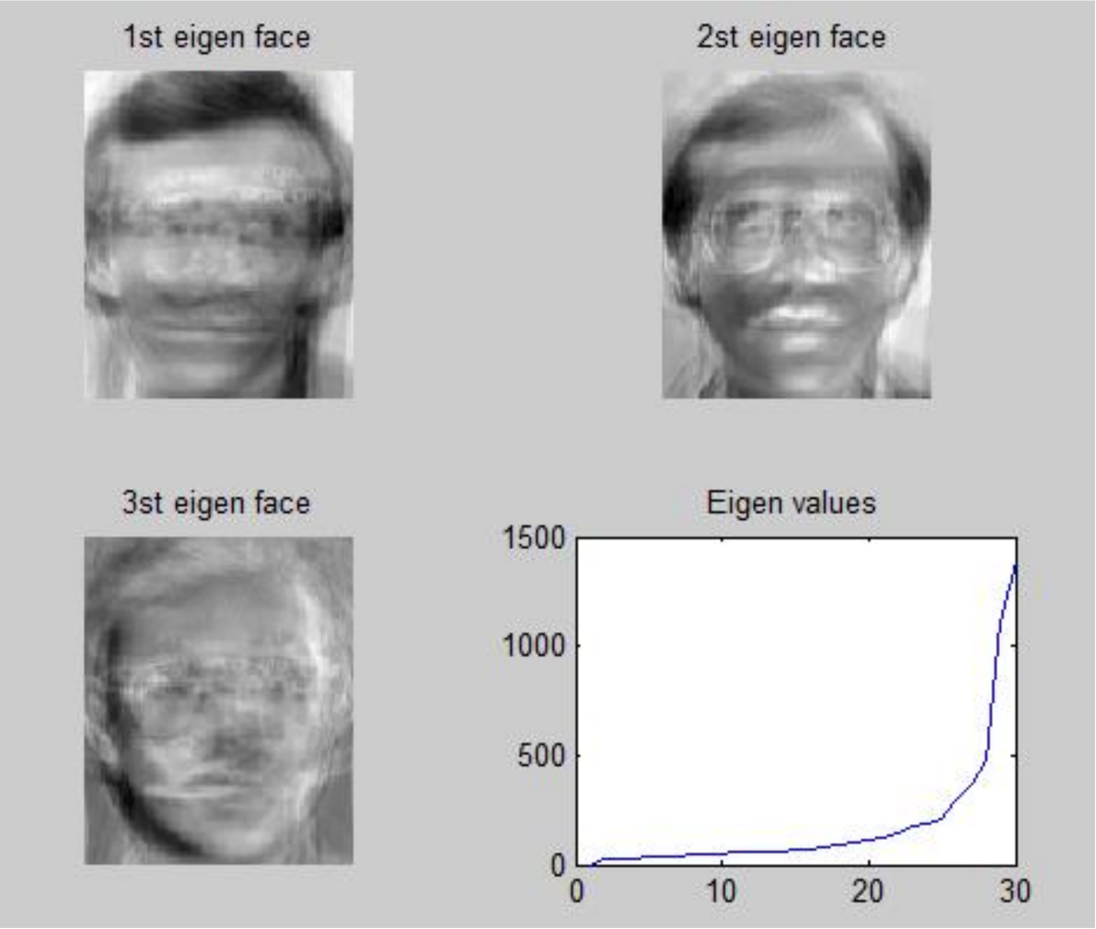
- Tập dữ liệu học



# Tính toán các vectơ khuôn mặt riêng:

- Vectơ khuôn mặt trung bình

# Tính toán các vectơ khuôn mặt riêng:



# Nhận diện

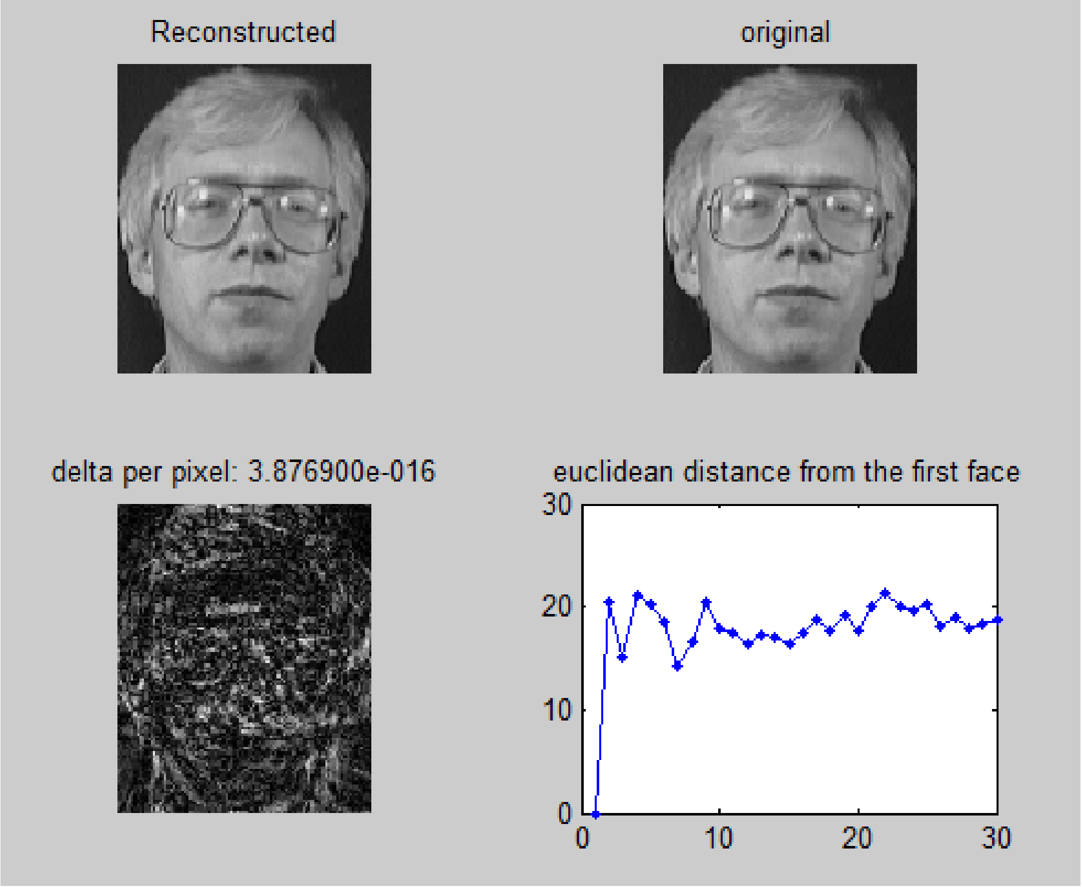
- Một ảnh mặt mới (Γ) được biến đổi thành các thành phần khuôn mặt riêng (chiếu vào không gian ảnh mặt) bằng công thức:

$$\omega_k = u_k^T(\Gamma - \Psi)$$

- Với k = 1,…, M’. $u^T_k$ là các vectơ riêng ta đã tính ở trên.
- Các trọng số của một vector $\Omega^T = [\omega_1, \omega 2, \ldots, \omega_{M'}]$ mô tả các phần liên quan của mỗi khuôn mặt riêng trong biểu diễn ảnh khuôn mặt nhập vào. Để nhận dạng, ta dựa vào phương pháp cực tiểu hóa khoảng cách:

$$\varepsilon_k = \left\| \Omega - \Omega_k \right\|^2$$

# Nhận diện


Reconstructed
original
delta per pixel: 3.876900e-016
euclidean distance from the first face
30
20
10
0
0
10
20
30

# Nhận diện

- Gần không gian mặt và gần một lớp khuôn mặt. → Một cá nhân được nhận dạng và xác định.
- Gần không gian mặt nhưng không gần một lớp khuôn mặt. → Phát hiện một cá nhân lạ.
- Xa không gian mặt và gần một lớp mặt. → Ảnh đã cho không diễn tả gương mặt.
- Xa không gian mặt và không gần lớp mặt nào. → Ảnh đã cho không diễn tả khuôn mặt.

# Ứng dụng của PCA

- Nhiễu đặc trưng bởi phương sai, dư thừa đặc trưng bởi phương sai, chéo hóa ma trận hiệp phương sai cho ta một ma trận mang các đặc trưng sau:
  – Phương sai lớn cho thấy nhiều thông tin chứa đựng trong thành phần đó.
  – Phương sai nhỏ cho thấy có thể là nhiễu.

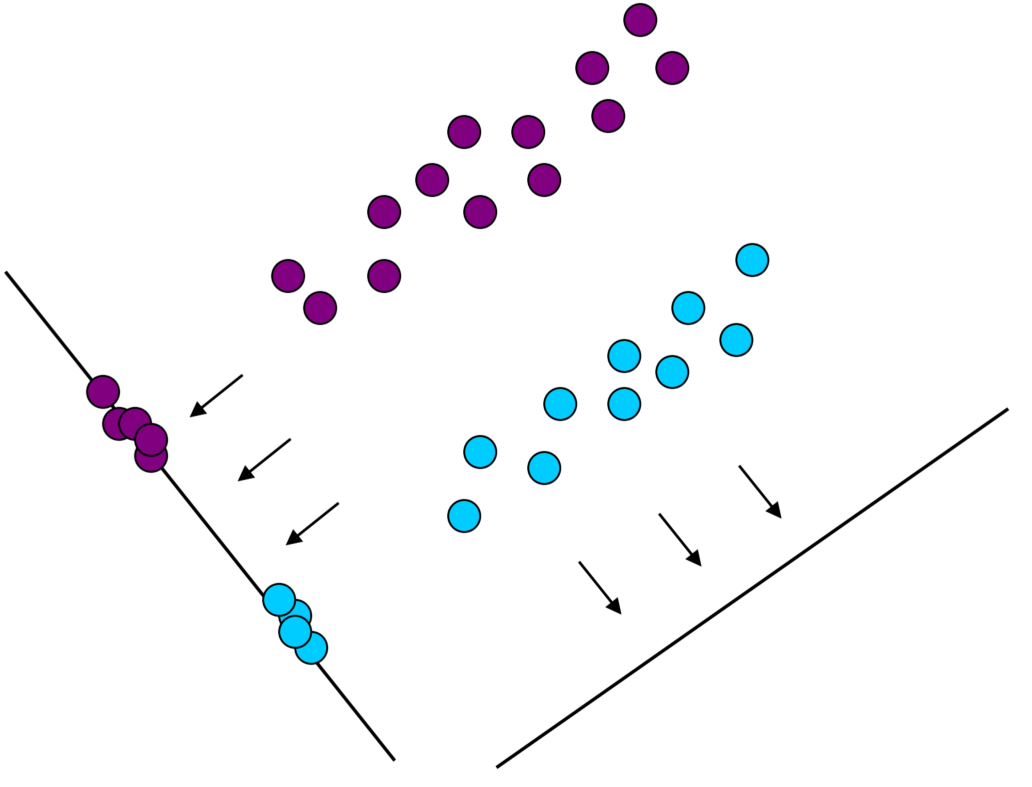
  Dựa vào đó, ta có thể loại bỏ bớt các thành phần không quan trọng và chỉ giữ lại những thành phần quan trọng.

– Xác định các trị riêng và vectơ riêng tổ hợp nên khuôn mặt.

# 4.2. Phân tích khác biệt tuyến tính LDA (Linear Discriminant Analysis)

# LDA

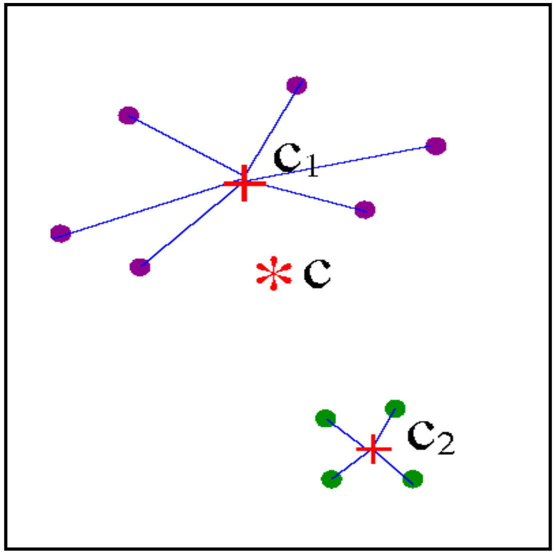
Tăng khả năng tách lớp dữ liệu bằng cách giảm chiều không gian dữ liệu



# LDA – Methodology

Cho tập dữ liệu $\{a_1, \cdots, a_n\}$

Centroids : $c_i = \dfrac{1}{n_i} \sum_{a \in class\ i} a \qquad c = \dfrac{1}{n} \sum_{i=1}^{n} a_i$




- Within-class scatter matrix

$$S_w = \sum_{i=1}^{r} \sum_{a \in class\ i} (a - c_i)(a - c_i)^T$$

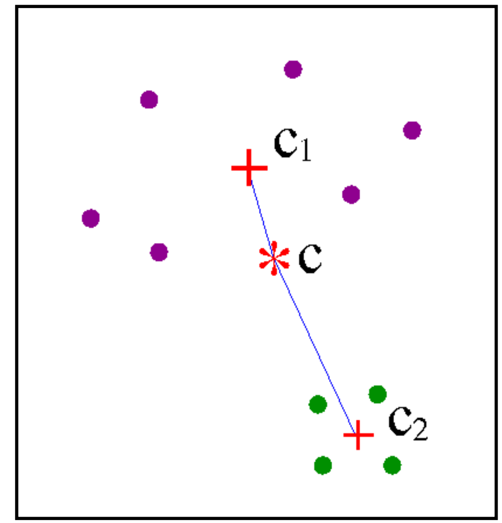
- trace$(S_w)$ $= \sum_{i=1}^{r} \sum_{a \in class\ i} \| a - c_i \|^2$

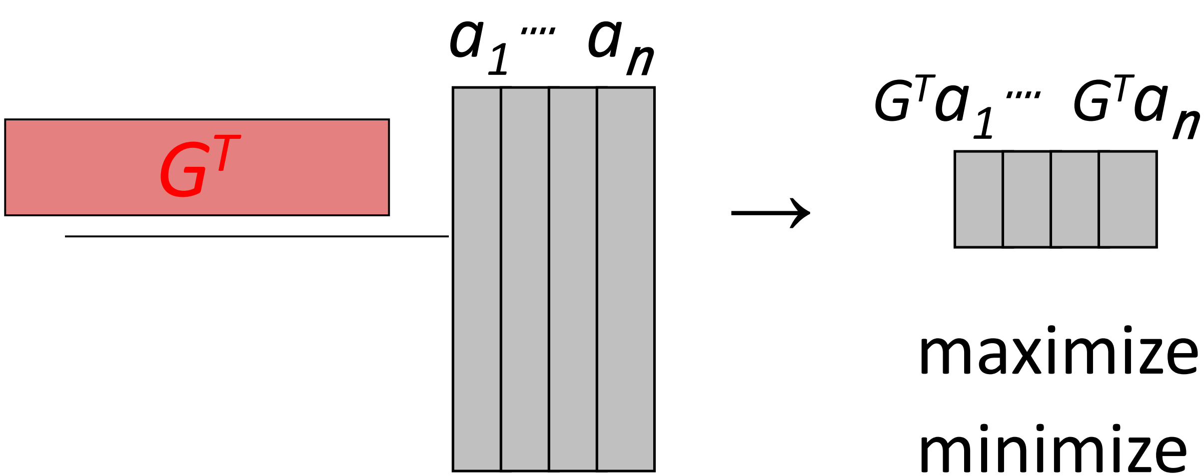
# LDA – Methodology

- Between-class scatter matrix

$$S_b = \sum_{i=1}^{r} n_i (c_i - c)(c_i - c)^T$$

- trace($S_b$) $= \sum_{i=1}^{r} \| c_i - c \|^2$



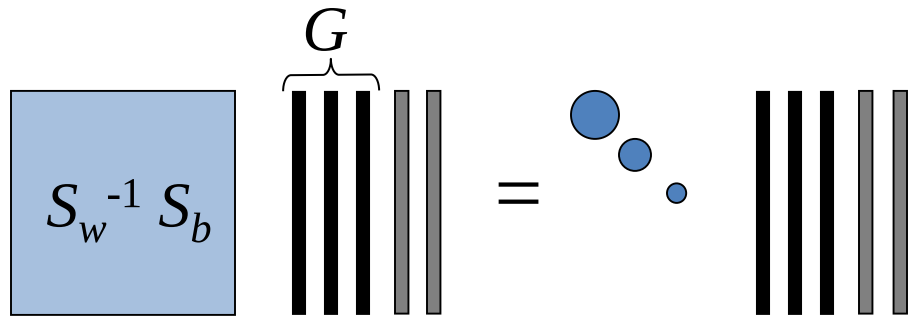

maximize trace($G^T S_b G$)

minimize trace($G^T S_w G$)

# LDA – Methodology

$$J(G) = \max_G \operatorname{trace}\left((G^T S_w G)^{-1} (G^T S_b G)\right)$$

Eigenvalue ? $S_w^{-1} S_b x = \lambda x$




$$S_w^{-1} S_b \quad X \quad = \quad \Sigma \quad X$$

rank($S_b$) $\leq$ number of classes - *1*

# LDA – Ví dụ

**<u>Cho bài toán ví dụ:</u>**

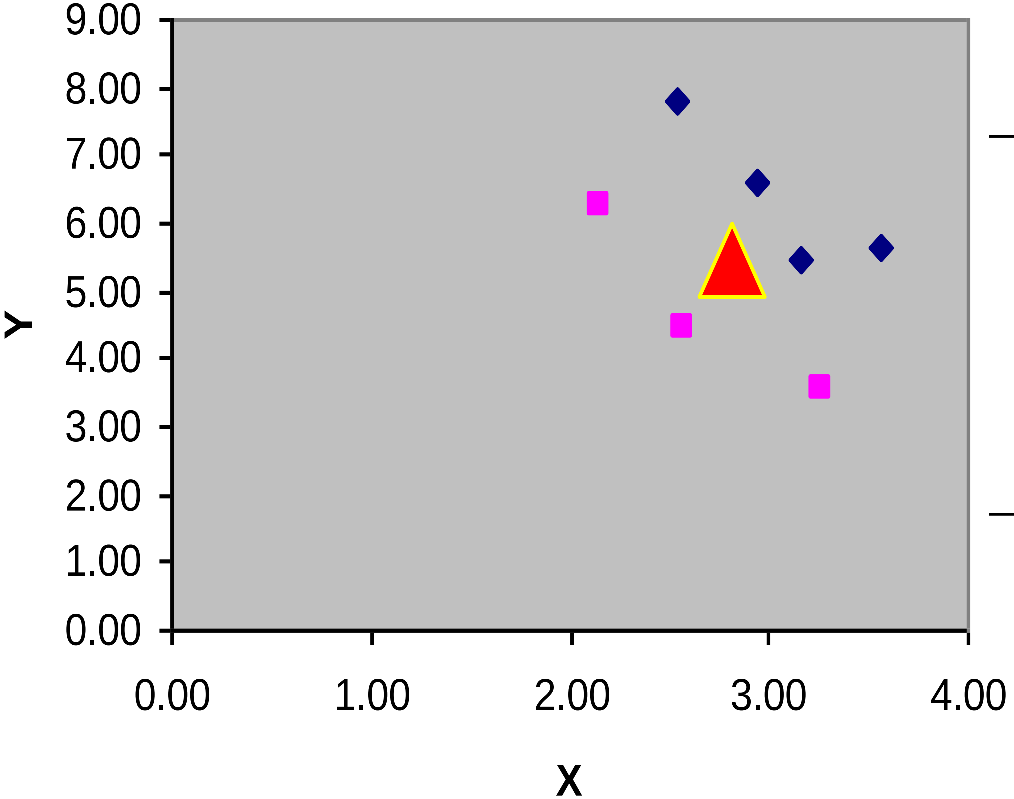
Hãng sản xuất ABC chuyên sản xuất sản phẩm Z với chất lượng sản phẩm chủ yếu dựa trên các thông số về độ cong và đường kính sản phẩm Z. Kết quả kiểm thử từ phong thí nghiệm cho biết chất lượng sản phẩm như bảng sau đây

| Độ cong | Đường kính | Quality Control Result |
|---|---|---|
| [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] |
| 2.16 | 6.22 | Not Passed |
| 3.27 | 3.52 | Not Passed |

**Độ cong : 2.81**
**Đường kính 5. 46**

**?**

# LDA – Ví dụ




| Class | X1 | X2 |
|---|---|---|
| 1 | 2.95 | 6.63 |
| 1 | 2.53 | 7.79 |
| 1 | 3.57 | 5.65 |
| 1 | 3.16 | 5.47 |
| 2 | 2.58 | 4.46 |
| 2 | 2.16 | 6.22 |
| 2 | 3.27 | 3.52 |
| **Dự đoán ?** | 2.81 | 5.46 |

# LDA – Ví dụ

x = đặc trưng (biến độc lập )của dữ liệu

y = nhóm các phần từ của dữ liệu (biến phụ thuộc) của dữ liệu

Trong ví dụ: $\mathbf{x} = \begin{bmatrix} 2.95 & 6.63 \\ 2.53 & 7.79 \\ 3.57 & 5.65 \\ 3.16 & 5.47 \\ 2.58 & 4.46 \\ 2.16 & 6.22 \\ 3.27 & 3.52 \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 2 \\ 2 \\ 2 \end{bmatrix}$

# LDA – Ví dụ

- $x_k$= dữ liệu hàng k $\mathbf{x}_3 = \begin{bmatrix} 3.57 & 5.65 \end{bmatrix}$ $\mathbf{x}_7 = \begin{bmatrix} 3.27 & 3.52 \end{bmatrix}$
- g = số nhóm của y. Trong ví dụ : g = 2
- $x_i$ = đặc trưng dữ liệu của nhóm I

$$\mathbf{x}_1 = \begin{bmatrix} 2.95 & 6.63 \\ 2.53 & 7.79 \\ 3.57 & 5.65 \\ 3.16 & 5.47 \end{bmatrix} \quad \mathbf{x}_2 = \begin{bmatrix} 2.58 & 4.46 \\ 2.16 & 6.22 \\ 3.27 & 3.52 \end{bmatrix}$$

- $\mu_i$ = trung bình của đặc trưng trong nhóm I, với giá trị là trung bình cộng của $x_i$ $\boldsymbol{\mu}_1 = \begin{bmatrix} 3.05 & 6.38 \end{bmatrix}$ $\boldsymbol{\mu}_2 = \begin{bmatrix} 2.67 & 4.73 \end{bmatrix}$
- $\mu$ = vector trung bình tổng (global mean vector) $\boldsymbol{\mu} = \begin{bmatrix} 2.88 & 5.676 \end{bmatrix}$

# LDA – Ví dụ

$\mathbf{x}_i^o$ = phương sai của dữ liệu (mean corrected of data) bằng dữ liệu đặc trừng của nhóm I trừ đi vector trung bình tổng

$$\mathbf{x}_1^o = \begin{bmatrix} 0.060 & 0.951 \\ -0.357 & 2.109 \\ 0.679 & -0.025 \\ 0.269 & -0.209 \end{bmatrix} \quad \mathbf{x}_2^o = \begin{bmatrix} -0.305 & -1.218 \\ -0.732 & 0.547 \\ 0.386 & -2.155 \end{bmatrix}$$

$$\mathbf{c}_i = \frac{\left(\mathbf{x}_i^o\right)^T \mathbf{x}_i^o}{n_i}$$ = Ma trận hiệp phương sai (covariance matrix ) nhóm i

$$\mathbf{c}_1 = \begin{bmatrix} 0.166 & -0.192 \\ -0.192 & 1.349 \end{bmatrix} \quad \mathbf{c}_2 = \begin{bmatrix} 0.259 & -0.286 \\ -0.286 & 2.142 \end{bmatrix}$$

# LDA – Ví dụ

$C(r,s) = \frac{1}{n}\sum_{i=1}^{g} n_i \cdot c_i(r,s)$ = tổ hợp các ma trận hiệp phương sai từ các nhóm. Nó được tính dựa trên các nhóm trong ma trận.

$$\frac{3}{7}\cdot 0.166 + \frac{4}{7}\cdot 0.259 = 0.206$$

$$\frac{3}{7}(-0.192) + \frac{4}{7}(-0.286) = -0.233 \longrightarrow \mathbf{C} = \begin{bmatrix} 0.206 & -0.233 \\ -0.233 & 1.689 \end{bmatrix}$$

$$\frac{3}{7}\cdot 0.1349 + \frac{4}{7}\cdot 2.142 = 1.689$$

# LDA – Ví dụ

Nghịch đảo ma trận C là $\mathbf{C}^{-1} = \begin{bmatrix} 5.745 & 0.791 \\ 0.791 & 0.701 \end{bmatrix}$

p = vector tiền xác suất (mô tả tỉ lệ xác suất ứng với mỗi nhóm sản phẩm). Nếu không được biết trước tỉ lệ này, chúng ta có thể tính bằng cách lấy số lượng sản phẩm của nhóm chia cho số lượng toàn bộ sản phẩm

$$p_i = \frac{n_i}{N} \longrightarrow \mathbf{p} = \begin{bmatrix} 0.571 \\ 0.429 \end{bmatrix} = \begin{bmatrix} 4/7 \\ 3/7 \end{bmatrix}$$

Hàm Discriminant

$$f_i = \boldsymbol{\mu}_i \mathbf{C}^{-1} \mathbf{x}_k^T - \tfrac{1}{2} \boldsymbol{\mu}_i \mathbf{C}^{-1} \boldsymbol{\mu}_i^T + \ln(p_i)$$

Chúng ta sẽ xếp lớp các sản phẩm mới với hàm f là lớn nhất

# LDA – Ví dụ

| Tra | Discriminant function | | |
|---|---|---|---|
| clas | f1 | f2 | |
| | 55.220 | 53.071 | 6.63 |
| | 53.774 | 51.394 | 7.79 |
| | 62.476 | 59.589 | 5.65 |
| | 51.953 | 50.764 | 5.47 |
| | 32.028 | 34.313 | 4.46 |
| | 34.554 | 35.757 | 6.22 |
| | 41.174 | 42.414 | 3.52 |
| pre | 44.049 | 44.085 | 5.46 |

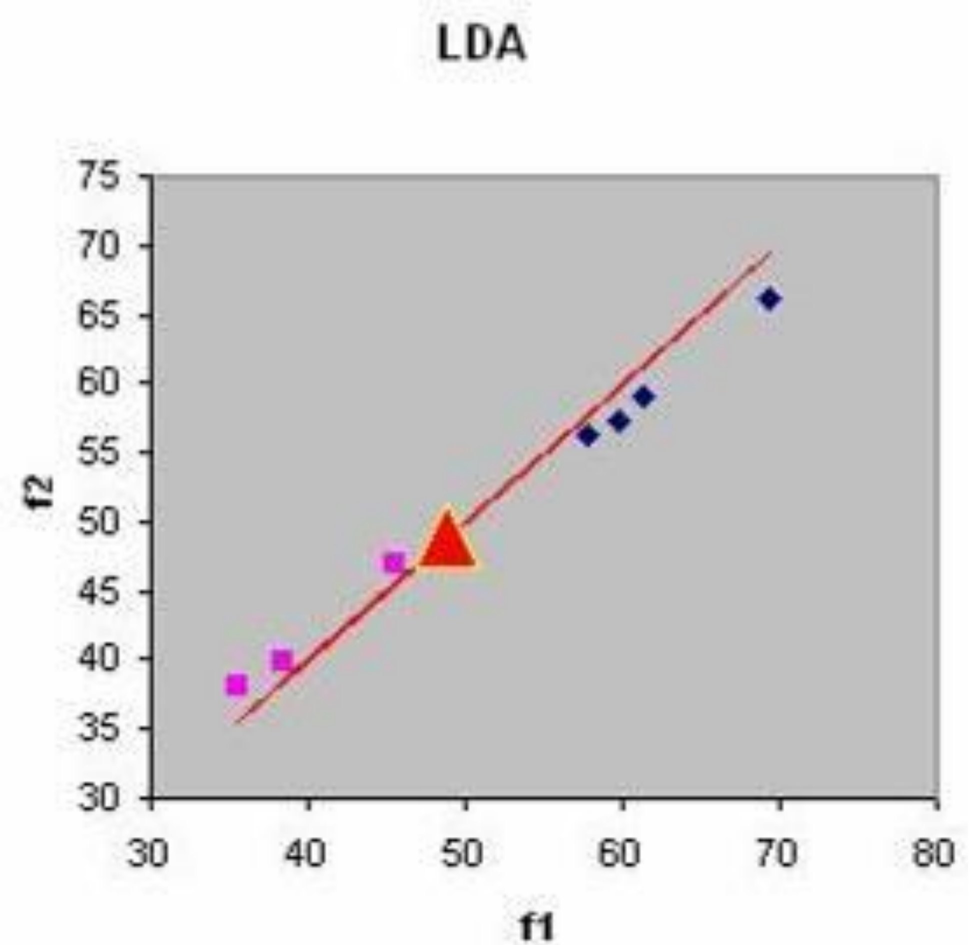
| Mea | Results | Data |
|---|---|---|
| X1o | Classificati | |
| 0 | 1 | |
| -0 | 1 | |
| 0 | 1 | |
| 0 | 1 | |
| -0 | 2 | |
| -0 | 2 | |
| 0 | 2 | |
| -0 | 2 | |

# LDA – Ví dụ

Hàm f được xem như là một luật phân loạt để xác nhận các cá thể và các nhóm riêng biệt. Trong ví dụ trên với sản phẩm Z có độ cong là 2.81 và bán kính là 5.46 ta có

- f(1)=44.049
- f(2)=44.085

Với f(1) < f(2) → sản phẩm Z không đạt yêu cầu (nhóm 2)

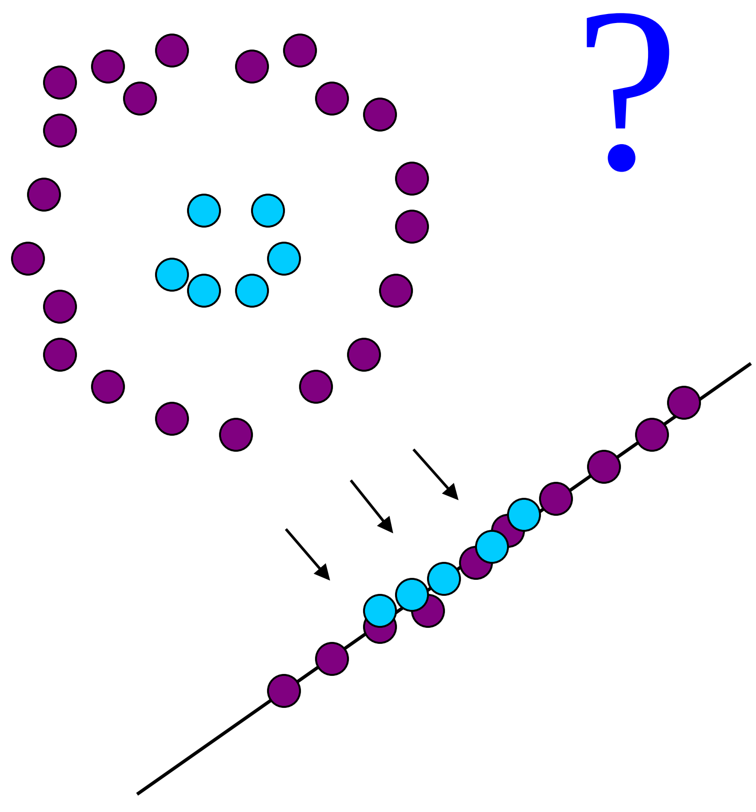


# LDA – so sánh với PCA

- LDA có tốt hơn PCA không?
  - Có một xu hướng trong cộng đồng máy tính là thích LDA hơn PCA về khả năng dễ sử dụng của nó
  - Có một khác biệt chính là LDA xác định rõ về hàm phân biệt giữa các lớp trong khi PCA không rõ ràng về việc xác định rõ cấu trúc lớp
  - Xét về các mẫu test với kích thước nhỏ PCA cho kết quả tốt hơn LDA những với những mẫu test có kích thước lớn và cần sự biểu diễn riêng biệt cho mỗi lớp thì LDA tỏ ra vượt trội so với PCA

# KLDA – Nonlinear Dimension Reduction

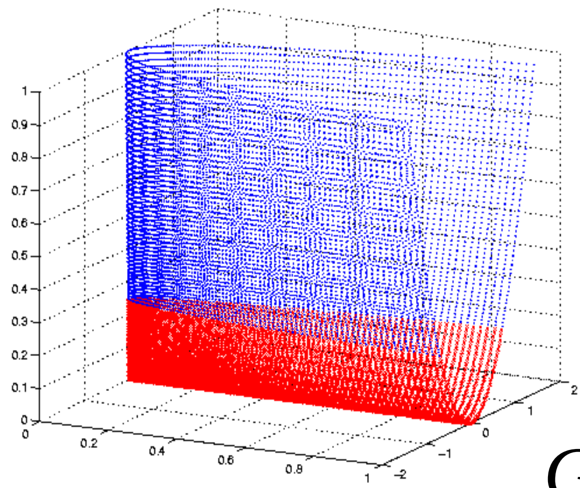
Áp dụng LDA vào dữ liệu phi tuyến tính



# KLDA – Nonlinear Dimension Reduction

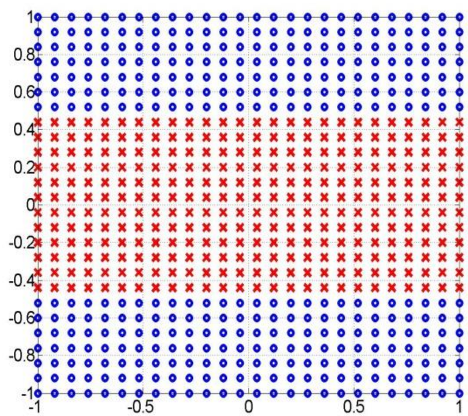
**1 - Ánh xạ phi tuyến**

$$\Phi(x_1, x_2) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$$



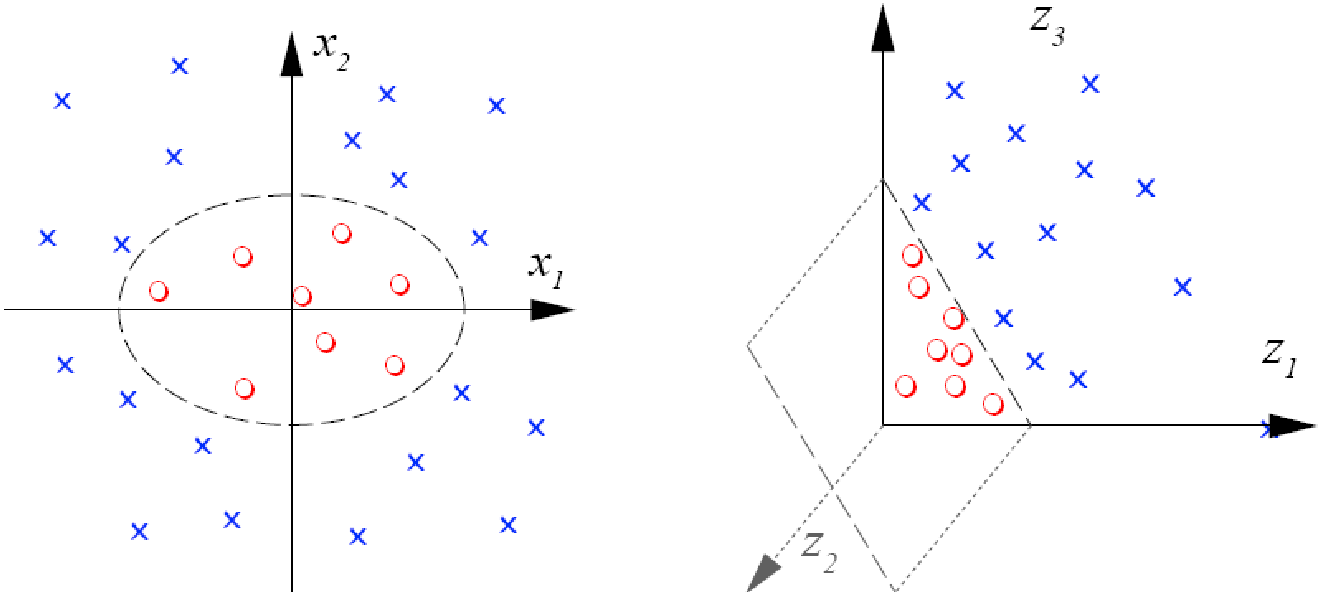

**2 - LDA**

$G^T$



# KLDA – Nonlinear Dimension Reduction

$$\Phi : \mathbb{R}^2 \to \mathbb{R}^3$$
$$(x_1, x_2) \mapsto (z_1, z_2, z_3) := (x_1^2, \sqrt{2}\, x_1 x_2, x_2^2)$$




$$\left(\frac{x_1}{a}\right)^2 + \left(\frac{x_2}{b}\right)^2 = 1 \longrightarrow \frac{z_1}{a^2} + \frac{z_3}{b^2} = 1$$

# KLDA – Nonlinear Dimension Reduction by Kernel Methods

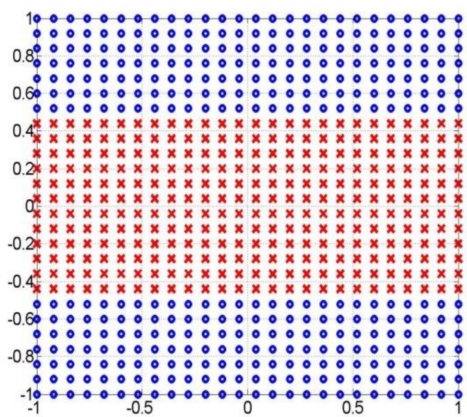
**Sử dụng hàm biến đổi Kernel *k*(*x*,*y*)**

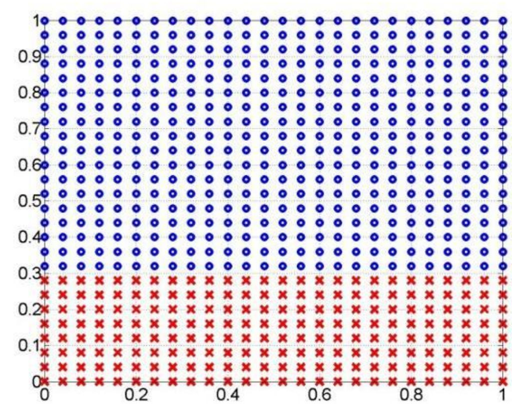
$$< \Phi(x), \Phi(y) > = k(x, y)$$

$\Phi$

LDA

$G^T$



# KLDA – HÀM BIẾN ĐỔI

- Gaussian kernel

$$k(x,y) = \exp(-\|x-y\|^2 / \sigma)$$

- Polynomial kernel

$$k(x,y) = (\alpha_1 < x, y > +\alpha_2)^d \quad (d > 0,\ \alpha_1, \alpha_2 \in R)$$

# KLDA – NHẬN DẠNG PHI TUYẾN SỬ DỤNG KERNEL

$$\{\Phi(a_1),\ldots,\Phi(a_n)\}$$

Want to apply LDA

$\Phi$

- $\{a1,a2,\ldots,an\}$

$$S_b\, x = \lambda\, S_w\, x$$

$<\Phi(x),\Phi(y)>= k(x,y)$

# KLDA – NHẬN DẠNG PHI TUYẾN SỬ DỤNG KERNEL (2)

$$\{\Phi(a_1),\ldots,\Phi(a_n)\}$$

$\Phi$

- $\{a1, a2, \ldots, an\}$

$$S_b\, x = \lambda\, S_w\, x$$

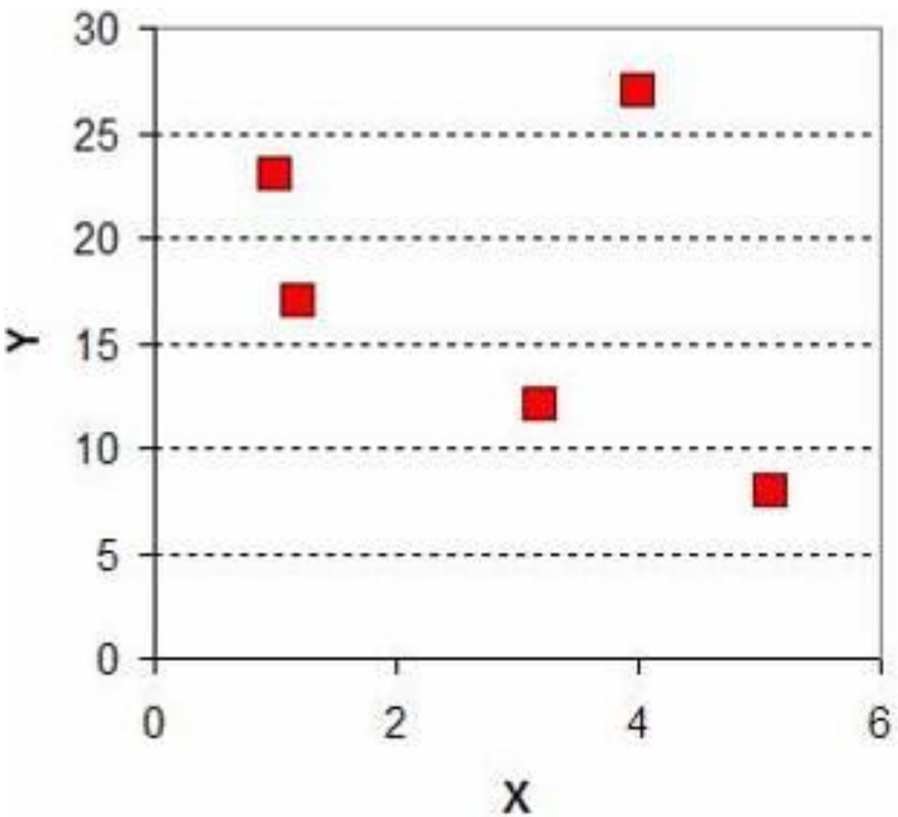
$$\left\{ \begin{bmatrix} k(a_1,a_1) \\ \cdots \\ k(a_n,a_1) \end{bmatrix}, \ldots, \begin{bmatrix} k(a_1,a_n) \\ \cdots \\ k(a_n,a_n) \end{bmatrix} \right\}$$

$$\boldsymbol{S_b}\, u = \lambda\, \boldsymbol{S_w}\, u$$

Áp dụng thuật toán LDA

# KLDA – VÍ DỤ

Cho dữ liệu huấn luyện như sau. Tìm cách để nối các dữ liệu rời rạc lại với nhau ?




**Training Data**

| $i$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $X_i$ | 1 | 1.2 | 3.2 | 4 | 5.1 |
| $Y_i$ | 23 | 17 | 12 | 27 | 8 |

# KLDA – VÍ DỤ

Sử dụng hàm Gaussian Kernel :

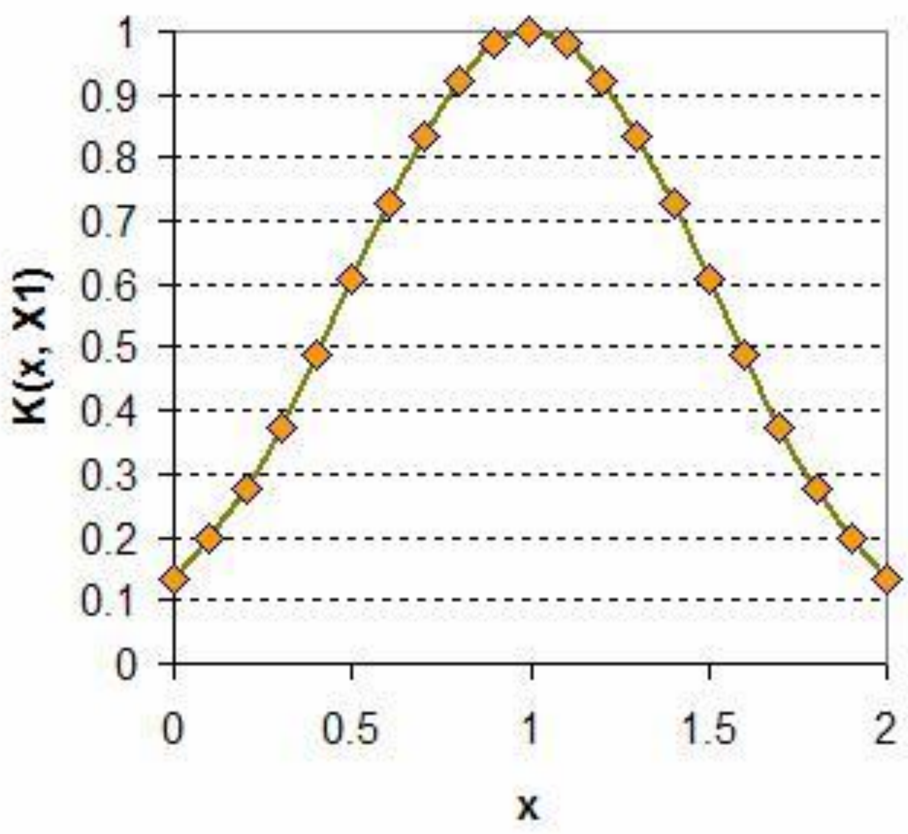
$$K_{\alpha}(x, X) = \exp\left(-\frac{(x - X)^2}{2\alpha^2}\right)$$

$$K_{\alpha=0.5}(x = 0, X_1 = 1) = \exp\left(-\frac{(x - X)^2}{2\alpha^2}\right) = \exp\left(-\frac{(0 - 1)^2}{2(0.5)^2}\right) = 0.13534$$

$$K_{\alpha=0.5}(x = 0.1, X_1 = 1) = \exp\left(-\frac{(0.1 - 1)^2}{2(0.5)^2}\right) = 0.19790$$

# KLDA – VÍ DỤ (2)

| x | K(x, X1) |
|---|---|
| 0 | 0.13534 |
| 0.1 | 0.19790 |
| 0.2 | 0.27804 |
| 0.3 | 0.37531 |
| 0.4 | 0.48675 |
| 0.5 | 0.60653 |
| 0.6 | 0.72615 |
| 0.7 | 0.83527 |
| 0.8 | 0.92312 |
| 0.9 | 0.98020 |
| 1 | 1.00000 |
| 1.1 | 0.98020 |
| 1.2 | 0.92312 |
| 1.3 | 0.83527 |
| 1.4 | 0.72615 |
| 1.5 | 0.60653 |
| 1.6 | 0.48675 |
| 1.7 | 0.37531 |
| 1.8 | 0.27804 |
| 1.9 | 0.19790 |
| 2 | 0.13534 |

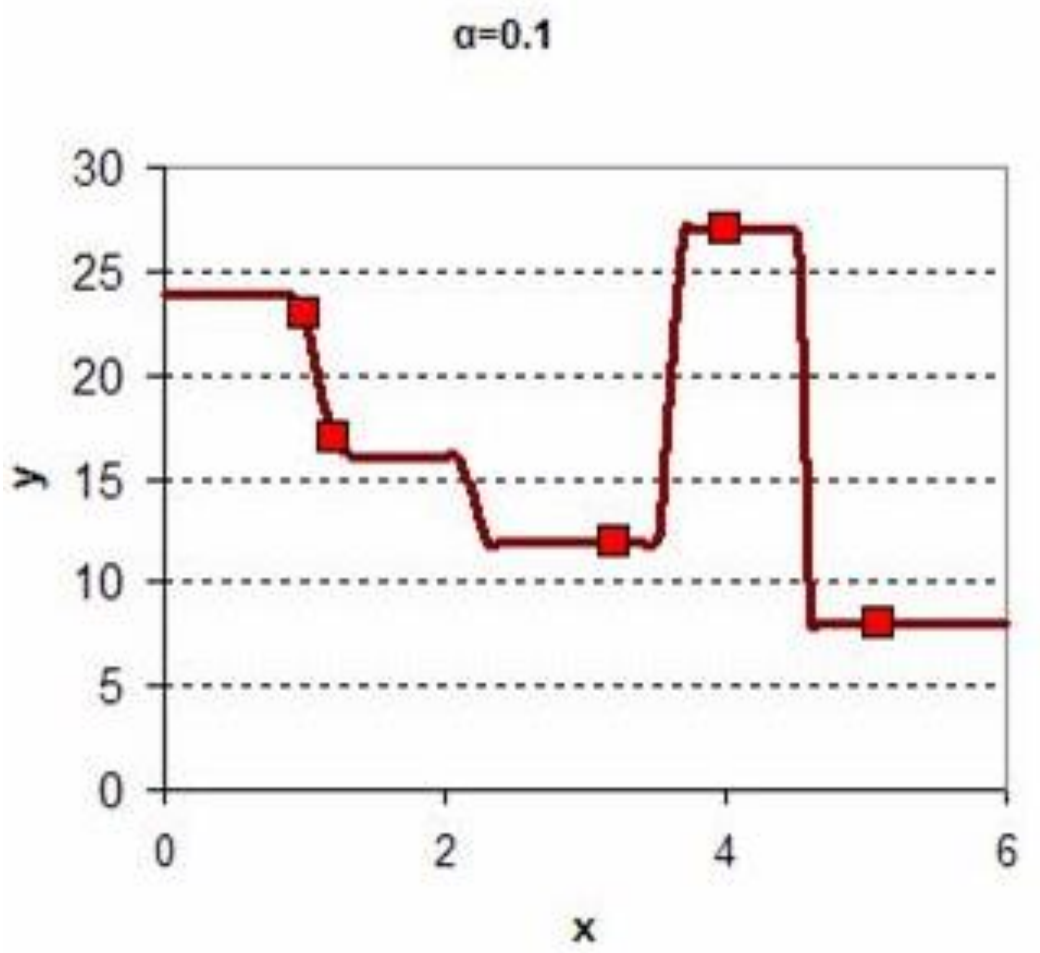


# KLDA – VÍ DỤ (3)

Nếu ta thu nhỏ hàm Kernel với tần số nhỏ lại.

$\alpha = 0.1$

Kết quả thu được với tần số nhỏ là một hàm số "thô" trong liên kết

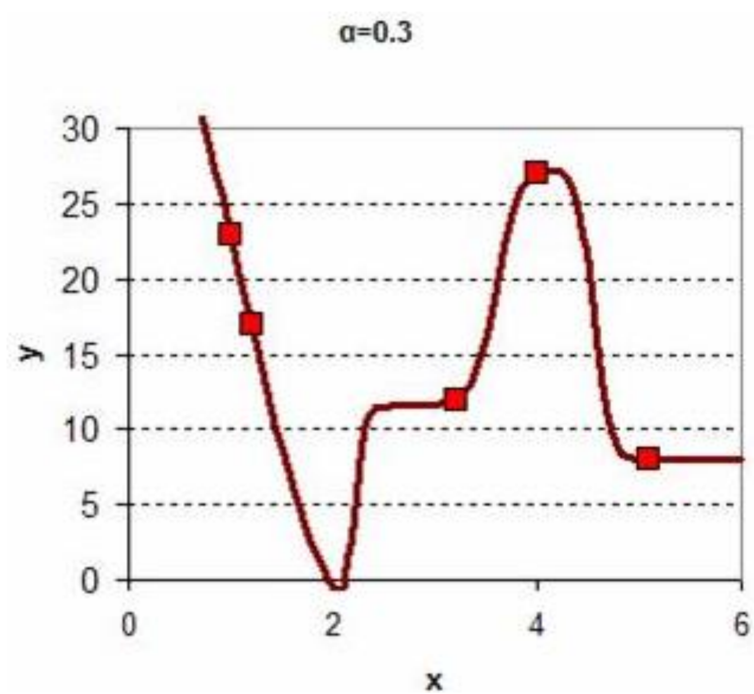


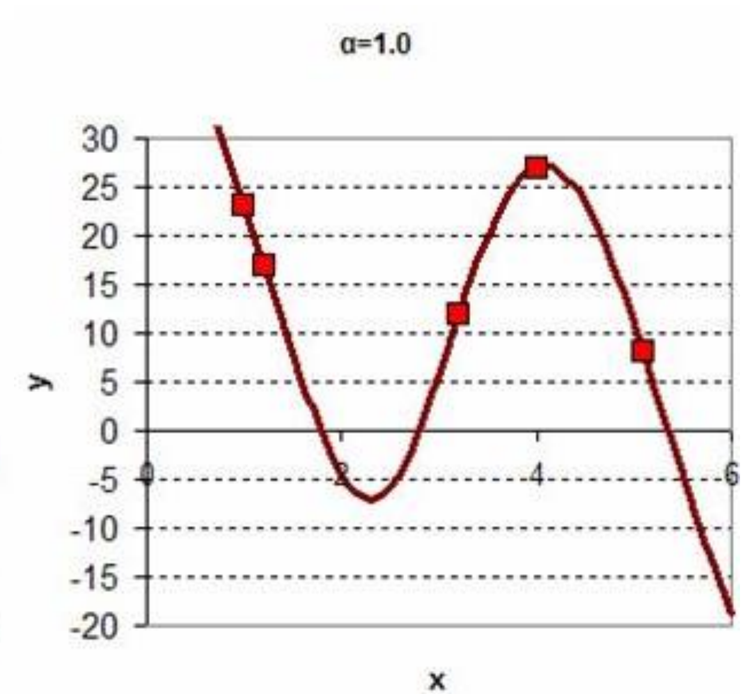
# KLDA – VÍ DỤ (4)

Thay tần số lớn hơn để tìm ra hàm kernel tốt nhất

$\alpha = 0.3$ , và $\alpha = 1.0$

Sẽ thu được một giá trị “trơn” hơn nhưng chi phí phải trả cho việc kéo giản là việc phải xác định giá trị Y < 0 như hình vẽ ở dưới

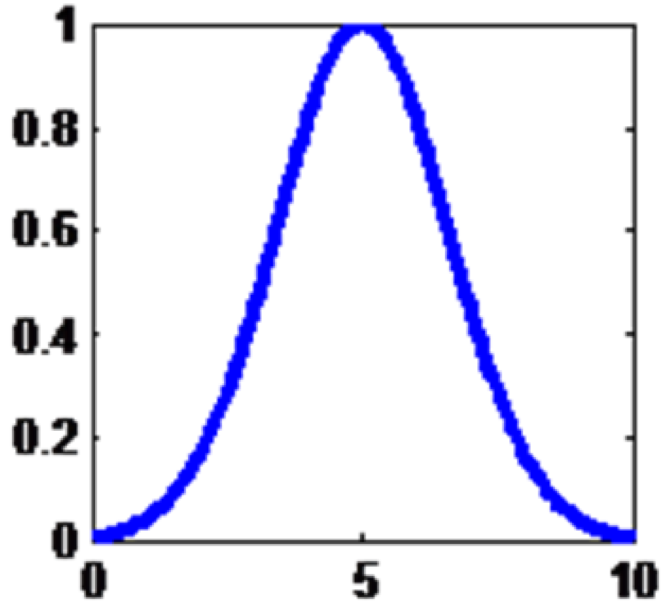


# KLDA – HÀM CƠ BẢN

Gaussian

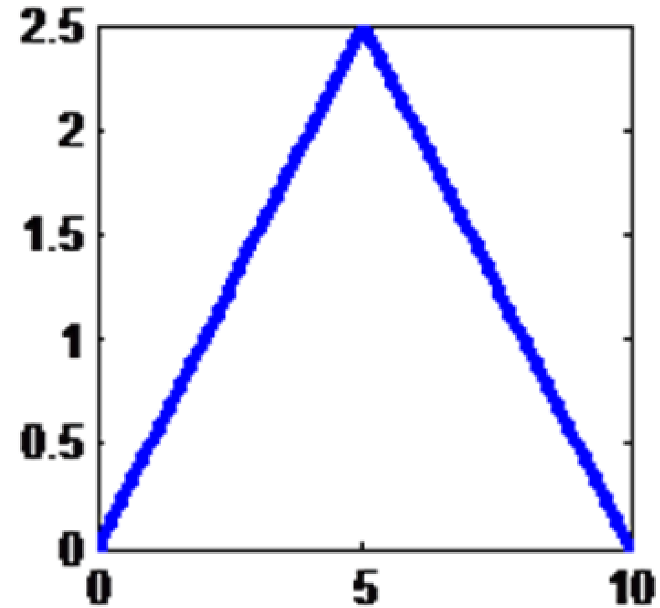
$$K_{\alpha}(x,X)=\exp\left(-\frac{(x-X)^2}{2\alpha^2}\right)$$




$\alpha=5$

Norm

$$K_{\alpha,\beta}(x,X)=-\frac{\|x-X\|}{\alpha}+\beta$$

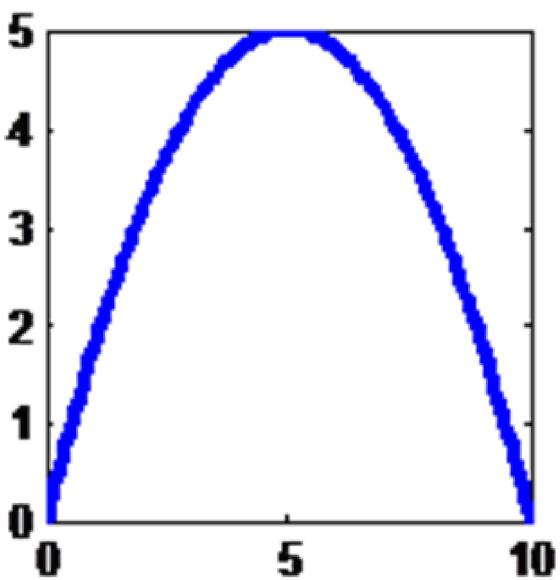



$\alpha=2$, $\beta=2.5$

# KLDA – HÀM CƠ BẢN (2)

Quadratic

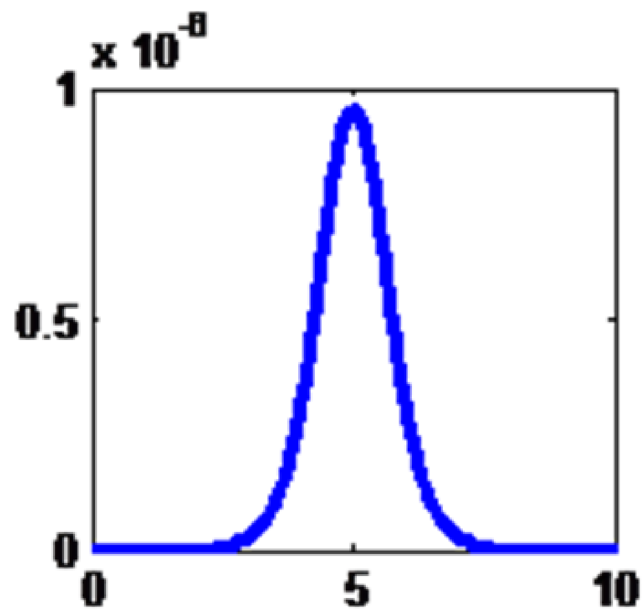
$$K_{\alpha,\beta}(x,X) = -\frac{(x-X)^2}{\alpha^2} + \beta$$




$\alpha = 5$, $\beta = 5$

Multi quadratic

$$K_{\alpha}(x,X) = \left(\frac{(x-X)^2}{\alpha^2} + \beta^2\right)^{-\gamma}$$

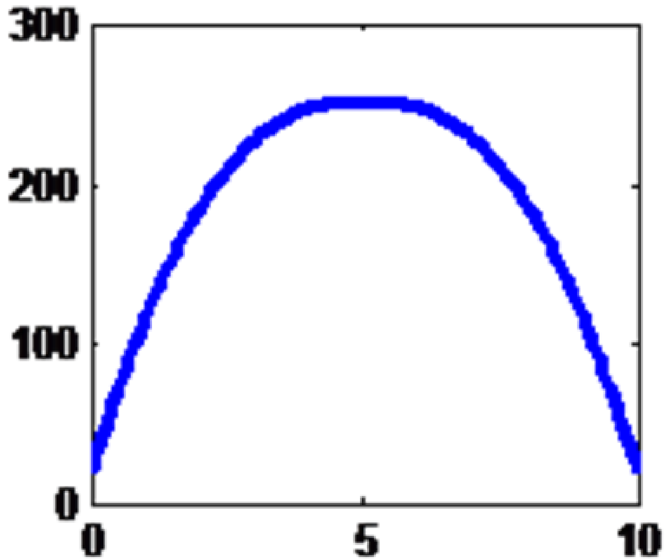



$\alpha = 0.5$, $\beta = 4$, $\gamma = 5$

# KLDA – HÀM CƠ BẢN (3)

Spline

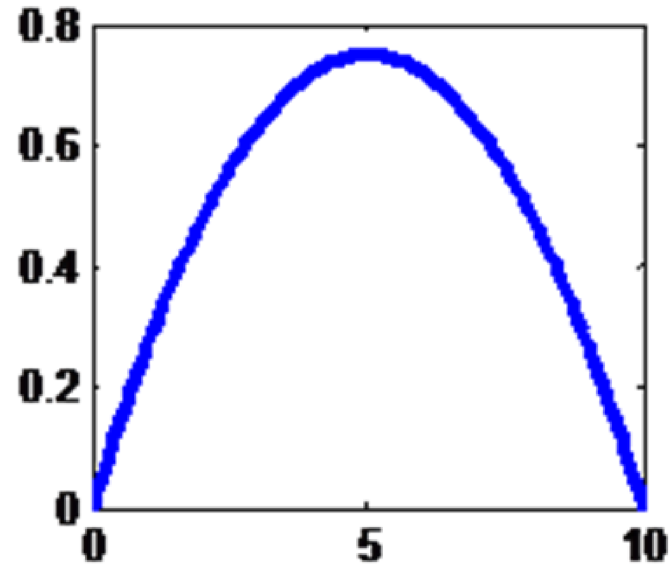
$$K_\alpha(x,X)=\begin{cases}\beta & \text{if } x=X\\ -\left(\frac{(x-X)}{\alpha}\right)^2 \ln\frac{(x-X)}{\alpha}+\beta, & \text{otherwise}\end{cases}$$




$\alpha=\frac{1}{2}$, $\beta=250$

Epanechnikov

$$K_\alpha(x,X)=\begin{cases}\frac{3}{4}\left(1-\frac{(x-X)^2}{\alpha^2}\right), & \text{if } \left(\frac{|x-X|}{\alpha}\right)\le 1\\ 0 & \text{otherwise}\end{cases}$$

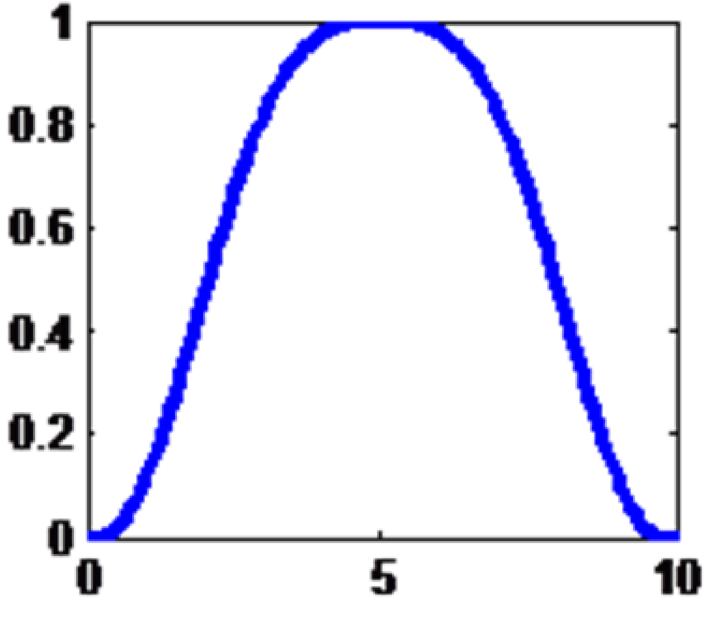



$\alpha=5$

# KLDA – HÀM CƠ BẢN (4)

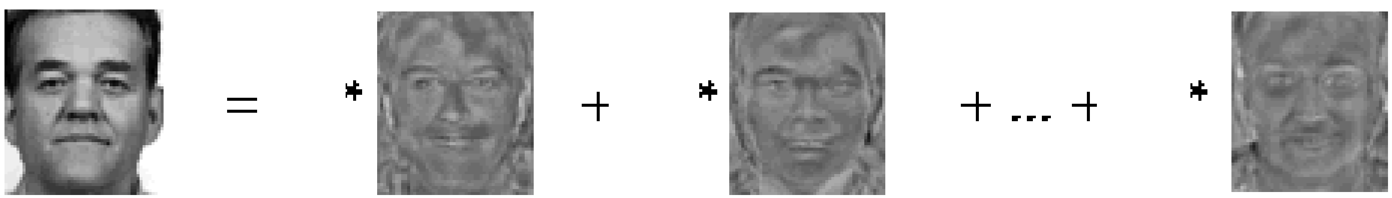
Tri-cube

$$K_\alpha(x,X) = \begin{cases} \left(1-\left|\dfrac{x-X}{\alpha}\right|^3\right)^3, & if \left(\dfrac{|x-X|}{\alpha}\right) \le 1 \\ 0 & otherwise \end{cases}$$




$\alpha = 5$

# KLDA – Face Recognition



**Feature vector=(** $y_1 \quad y_2 \quad \cdots \quad y_N$ **)**

**Face coding**

# AT&T Database

- Pose variation
- 40 classes, 10 images/class, 28 by 23

Set1

Set2

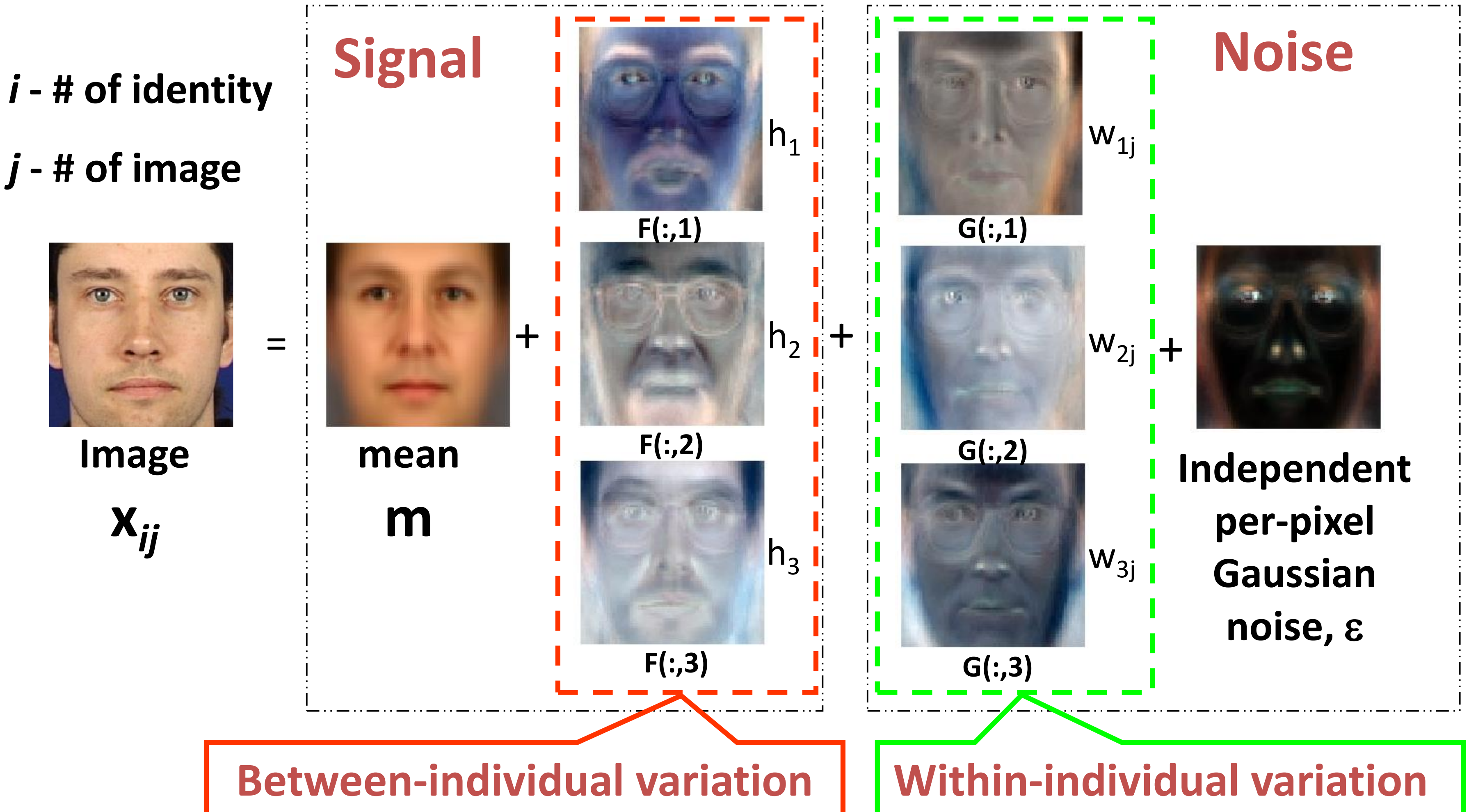
(Mirror of Set1)

# FERET Database

- Facial expression and illumination variation
- 200 classes, 3 images/class, 24 by 21

# KLDA – Face Recognition

$$x_{ij} = \mu + Fh_i + Gw_{ij} + \varepsilon_{ij}$$



# PHẦN THAM KHẢO THÊM LDA

# Linear Discriminant Analysis (LDA)

- Multiple classes and PCA
  - Suppose there are *C* classes in the training data.
  - PCA is based on the sample covariance which characterizes the scatter of the entire data set, *irrespective of class-membership*.
  - The projection axes chosen by PCA might not provide good discrimination power.

- What is the goal of LDA?
  - Perform dimensionality reduction while preserving as much of the class discriminatory information as possible.
  - Seeks to find directions along which the classes are best separated.
  - Takes into consideration the scatter *within-classes* but also the scatter *between-classes*.
  - More capable of distinguishing image variation due to identity from variation due to other sources such as illumination and expression.

# Linear Discriminant Analysis (LDA)

- Methodology

- Suppose there are $C$ classes
- Let $\boldsymbol{\mu}_i$ be the mean vector of class $i$, $i = 1, 2, \ldots, C$
- Let $M_i$ be the number of samples within class $i$, $i = 1, 2, \ldots, C$,
- Let $M = \sum_{i=0}^{C} M_i$ be the total number of samples. and

<u>Within-class scatter matrix:</u>

$$S_w = \sum_{i=1}^{C} \sum_{j=1}^{M_i} (y_j - \boldsymbol{\mu}_i)(y_j - \boldsymbol{\mu}_i)^T$$

<u>Between-class scatter matrix:</u>

$$S_b = \sum_{i=1}^{C} (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T$$

$$\boldsymbol{\mu} = 1/C \sum_{i=1}^{C} \boldsymbol{\mu}_i \quad \text{(mean of entire data set)}$$

# Linear Discriminant Analysis (LDA)

- Methodology – cont.
  - LDA computes a transformation that maximizes the between-class scatter while minimizing the within-class scatter:

$$\text{maximize } \frac{det(S_b)}{det(S_w)}$$

  - Such a transformation should retain class separability while reducing the variation due to sources other than identity (e.g., illumination).

# Linear Discriminant Analysis (LDA)

- Linear transformation implied by LDA
  - The linear transformation is given by a matrix *U* whose columns are the eigenvectors of $S_w^{-1} S_b$ (called *Fisherfaces*).

$$\begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_K \end{bmatrix} = \begin{bmatrix} u_1^T \\ u_2^T \\ \dots \\ u_K^T \end{bmatrix} (x - \boldsymbol{\mu}) = U^T (x - \boldsymbol{\mu})$$

  - The eigenvectors are solutions of the <u>*generalized eigenvector problem*</u>:

$$S_B u_k = \boldsymbol{\lambda}_k S_w u_k$$

# Linear Discriminant Analysis (LDA)

- Does $S_w^{-1}$ always exist?
  - If $S_w$ is non-singular, we can obtain a conventional eigenvalue problem by writing:

$$S_w^{-1} S_B u_k = \lambda_k u_k$$

  - In practice, $S_w$ is often singular since the data are image vectors with large dimensionality while the size of the data set is much smaller ($M << N$ )

# Linear Discriminant Analysis (LDA)

- Does $S_w^{-1}$ always exist? – cont.
  - To alleviate this problem, we can perform two projections:

1) PCA is first applied to the data set to reduce its dimensionality.

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

2) LDA is then applied to further reduce the dimensionality.

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

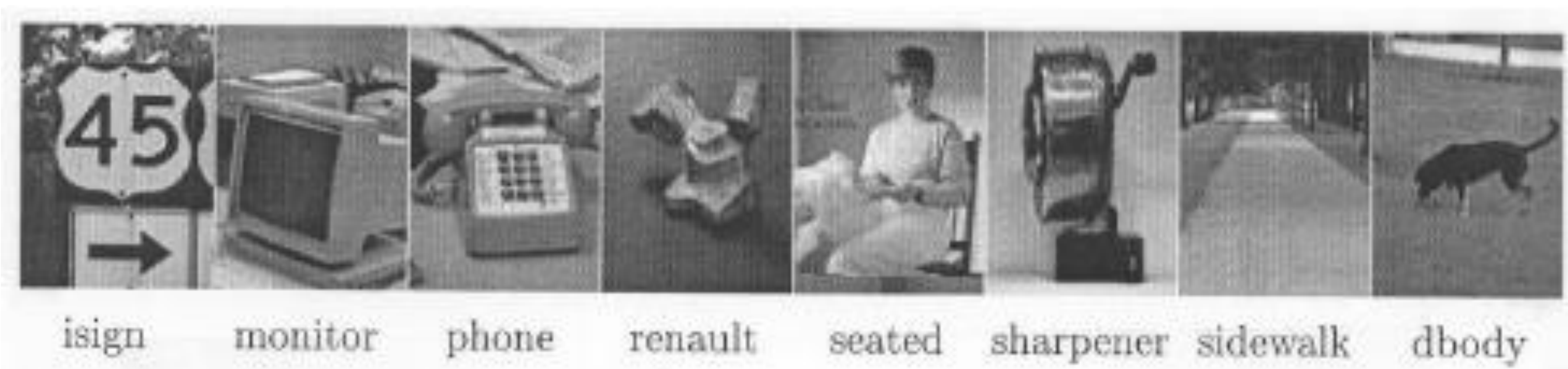
# Linear Discriminant Analysis (LDA)

- Case Study: Using Discriminant Eigenfeatures for Image Retrieval
  - D. Swets, J. Weng, "Using Discriminant Eigenfeatures for Image Retrieval", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 831-836, 1996.

- Content-based image retrieval
  - The application being studied here is query-by-example image retrieval.
  - The paper deals with the problem of selecting a good set of image features for content-based image retrieval.

# Linear Discriminant Analysis (LDA)

- Assumptions
  - "Well-framed" images are required as input for training and query-by-example test probes.
  - Only a small variation in the size, position, and orientation of the objects in the images is allowed.



# Linear Discriminant Analysis (LDA)

- Some terminology
  - Most Expressive Features (MEF): the features (projections) obtained using PCA.
  - Most Discriminating Features (MDF): the features (projections) obtained using LDA.

- Computational considerations
  - When computing the eigenvalues/eigenvectors of $S_w^{-1}S_B u_k = \lambda_k u_k$ numerically, the computations can be unstable since $S_w^{-1}S_B$ is not always symmetric.
  - See paper for a way to find the eigenvalues/eigenvectors in a stable way.
  - Important: Dimensionality of LDA is bounded by C-1 --- this is the rank of $S_w^{-1}S_B$

# Linear Discriminant Analysis (LDA)

- Factors unrelated to classification
  - MEF vectors show the tendency of PCA to capture major variations in the training set such as lighting direction.
  - MDF vectors discount those factors unrelated to classification.

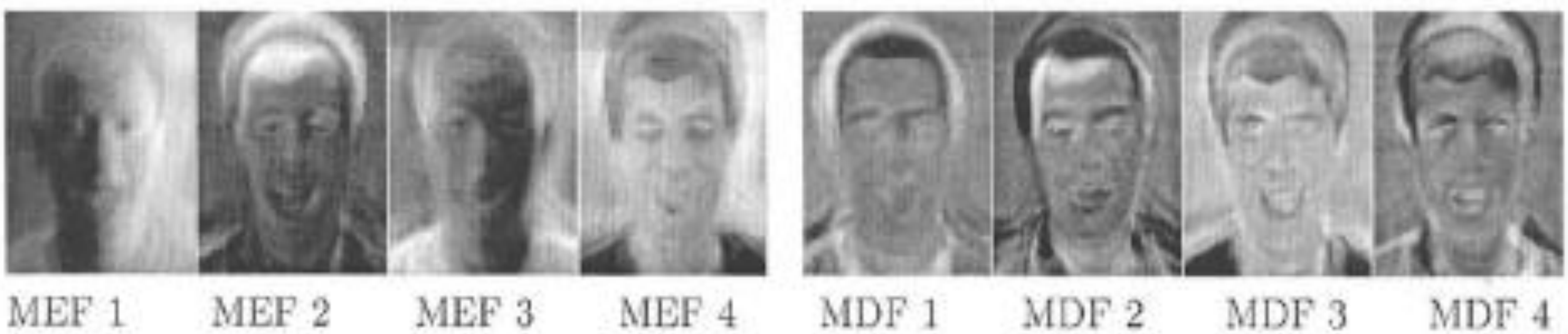
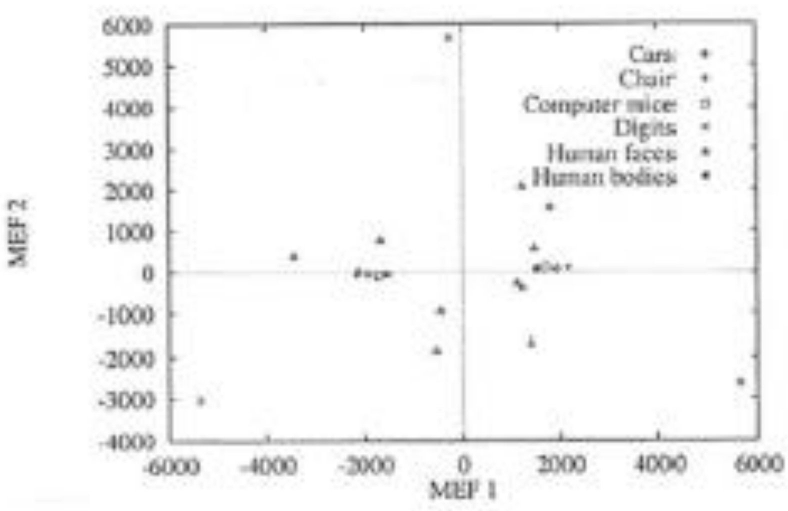
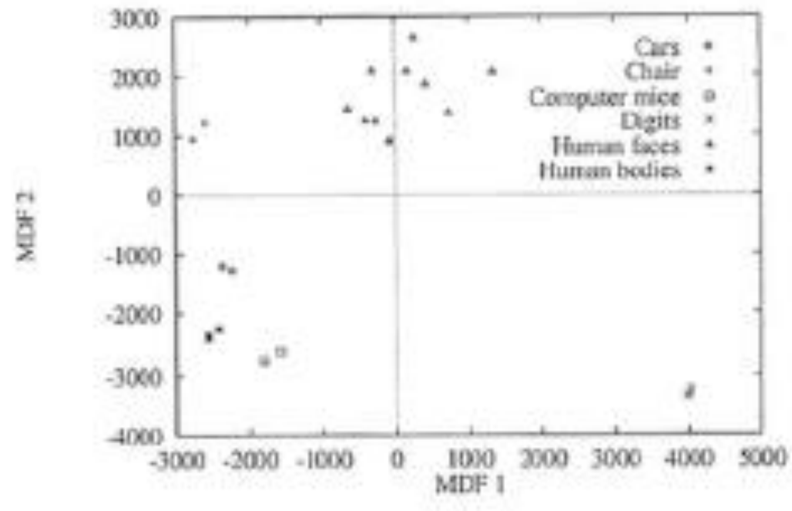



Figure 2. A sample of MEF and MDF vectors treated as images. The MEF vectors show the tendency of the principal components to capture major variations in the training set, such as lighting direction. The MDF vectors show the ability of the MDFs to discount those factors unrelated to classification. The training images used to produce these vectors are courtesy of the Weizmann Institute.

# Linear Discriminant Analysis (LDA)

- Clustering effect

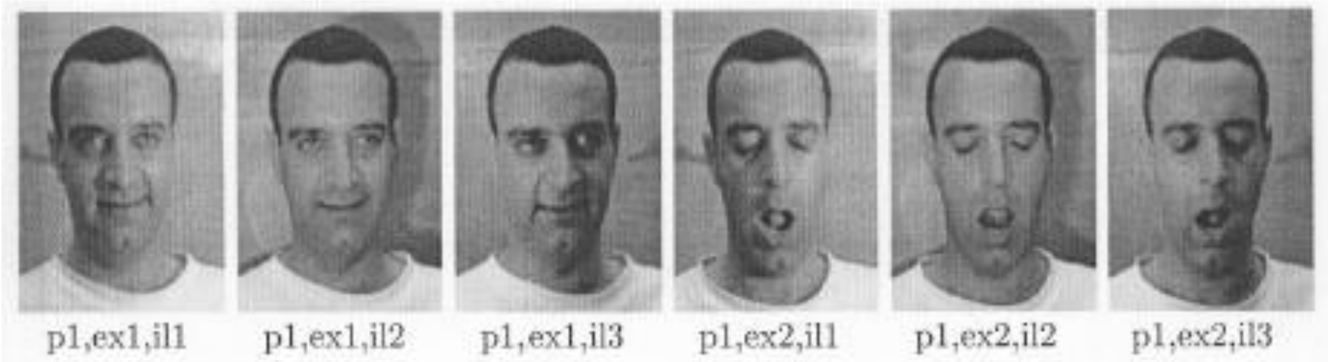


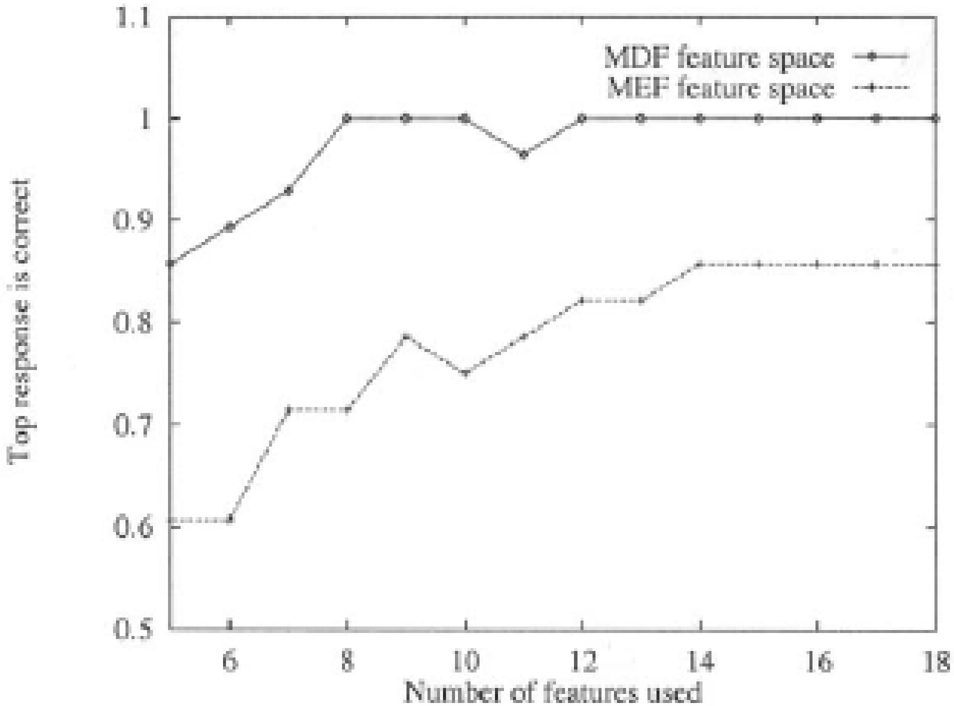

- Methodology
  1) Generate the set of MEFs for each image in the training set.
  2) Given an query image, compute its MEFs using the same procedure.
  3) Find the *k* closest neighbors for retrieval (e.g., using Euclidean distance).

# Linear Discriminant Analysis (LDA)

- Experiments and results
  - <u>Face images</u>
    - A set of face images was used with 2 expressions, 3 lighting conditions.
    - Testing was performed using a disjoint set of images - one image, randomly chosen, from each individual.



# Linear Discriminant Analysis (LDA)




(MDFs can tolerate within-class variability better than MEFs)

# Linear Discriminant Analysis (LDA)

– correct search probes

(a) List of training images

(b) List of search probes

(MDFs can tolerate within-class variability due to facial expressions)

# Linear Discriminant Analysis (LDA)

– failed search probe

(a) Search probe

(b) Training images

(query image corresponds to 3D rotation)

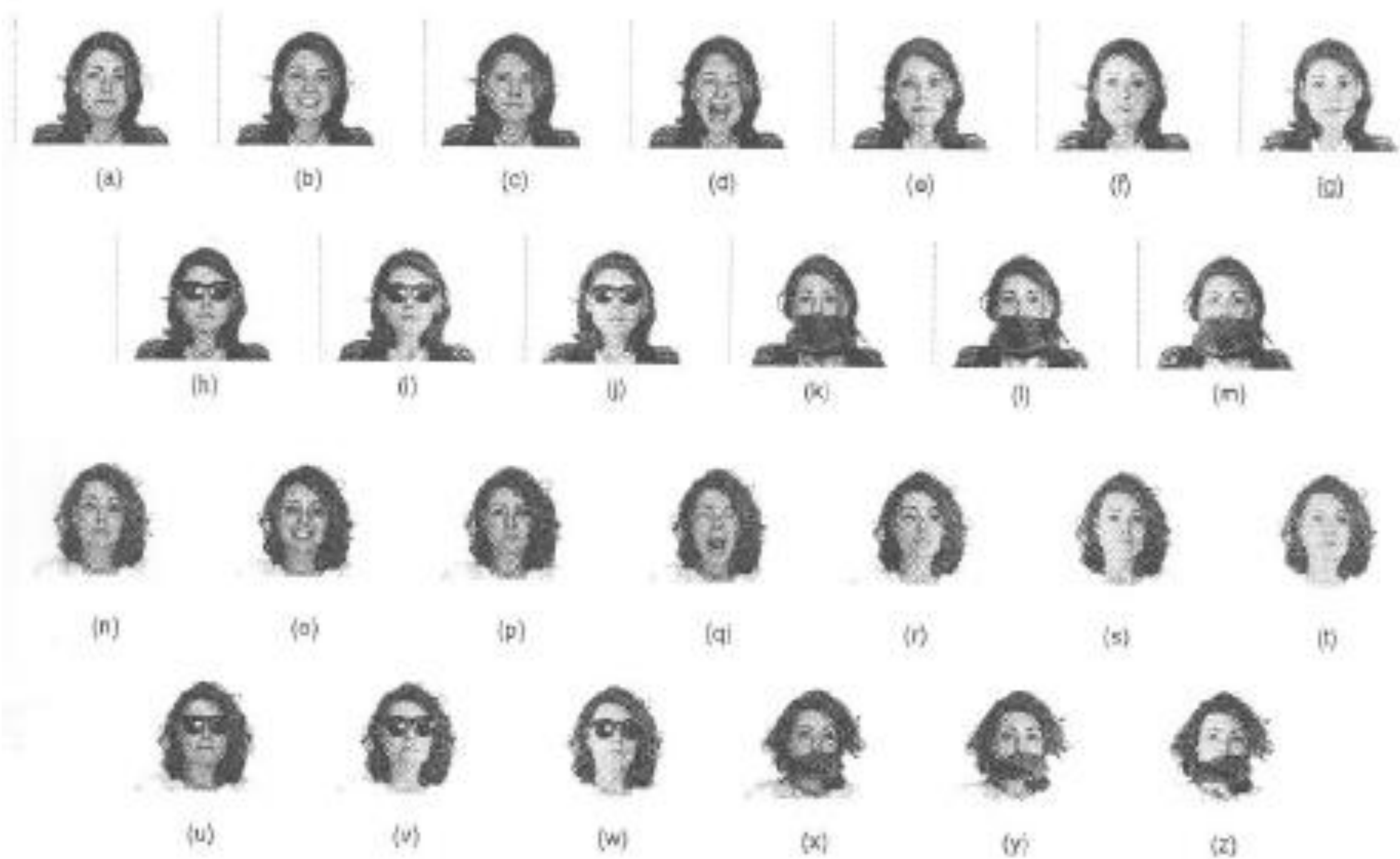
# Linear Discriminant Analysis (LDA)

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

- Is LDA always better than PCA?
  - There has been a tendency in the computer vision community to prefer LDA over PCA.
  - This is mainly because LDA deals directly with discrimination between classes while PCA does not pay attention to the underlying class structure.
  - This paper shows that when the training set is small, PCA can outperform LDA.
  - When the number of samples is large and representative for each class, LDA outperforms PCA.

# Linear Discriminant Analysis (LDA)

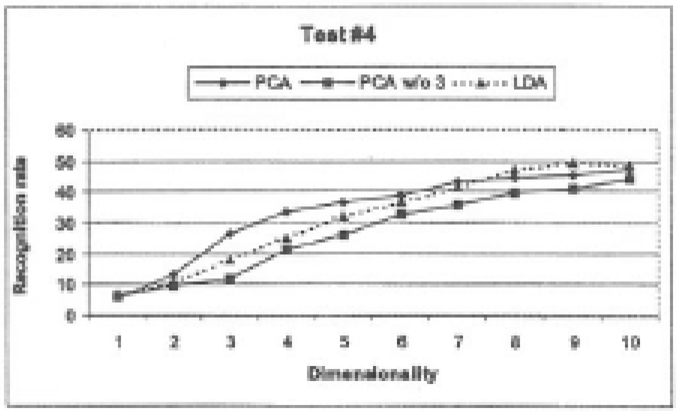
- Is LDA always better than PCA? – cont.

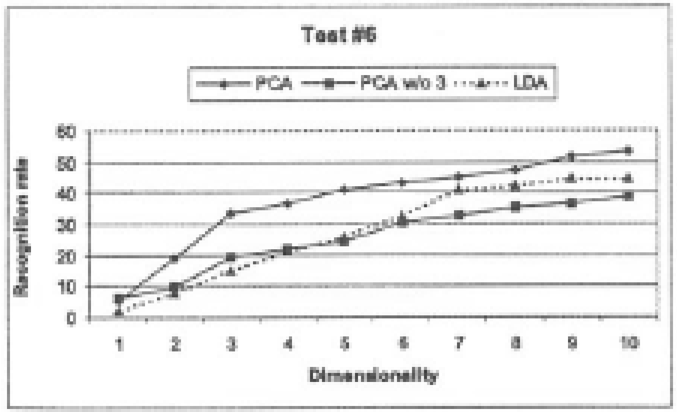


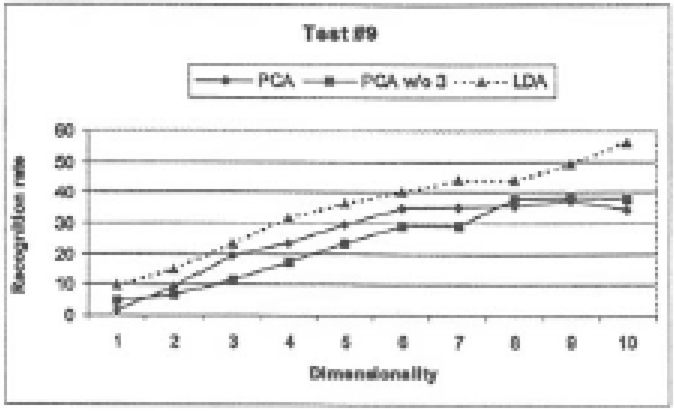
# Linear Discriminant Analysis (LDA)

- Is LDA always better than PCA? – cont.

(LDA is not always better when the sample is small)



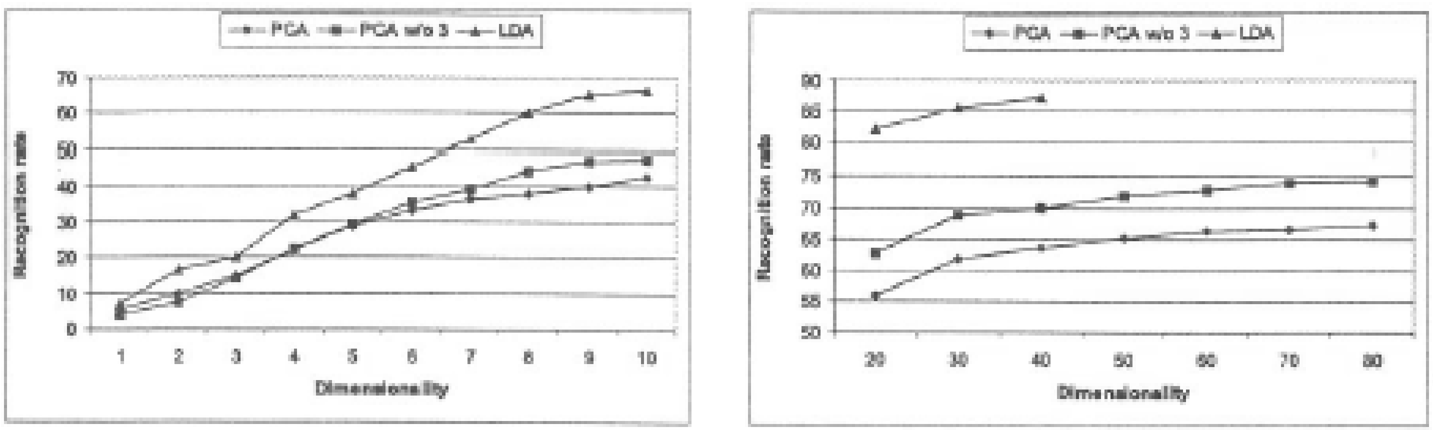



# Linear Discriminant Analysis (LDA)

- Is LDA always better than PCA? – cont.

(LDA outperforms PCA when the sample is large)



# Linear Discriminant Analysis (LDA)

- Critique of LDA
  - Only linearly separable classes will remain separable after applying LDA.
  - It does not seem to be superior to PCA when the training data set is small.

# Bài tiểu luận môn học

Theo danh sách bài tập lớn đã phân