PHƯƠNG PHÁP RÚT GỌN SỐ CHIỀU DỮ LIỆU (PCA)

Nội dung chính

- **Các khái niệm toán học có liên quan.**
- Mô hình PCA.
- Eigenface

Độ lệch chuẩn (standard deviation)

- Giá trị trung bình(mean) $\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n}$

- X = [0 8 12 20] mean = 10

- Y = [8 9 11 12] mean = 10

Độ lệch chuẩn (tt)

- Độ lệch chuẩn:

- Công thức $$s = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)}}$$

- SD(X) = 8.3266

- SD(Y) = 1.8257

Phương sai (covariance)

- Hình thức khác để đo độ phân tán của dữ liệu
- Công thức: $$s^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)}$$

Ví dụ

X_i	$(X_i - \bar{X})$	$(X_i - \bar{X})^2$
8	-2	4
9	-1	1
11	1	1
12	2	4
Total		10
Divided by (n-1)		3.333
Square Root		1.8257

Hiệp phương sai (covariance)

- Độ lệch chuẩn và phương sai tính trên dữ liệu một chiều (1 biến).
- Hiệp phương sai tính trên dữ liệu 2 chiều (2 biến).
- Ý nghĩa là thể hiện mối quan hệ giữa 2 chiều của dữ liệu.

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

Các tính chất của hiệp phương sai

- Nếu giá trị của hiệp phương sai là dương, 2 chiều dữ liệu tỉ lệ thuận với nhau.
- Nếu giá trị của hiệp phương sai là âm, 2 chiều dữ liệu tỉ lệ nghịch với nhau.
- Nếu bằng 0, 2 chiều dữ liệu là độc lập với nhau.
- $\text{cov}(x, y) = \text{cov}(y, x)$.

Ví dụ

H	M	$(H_i - \bar{H})$	$(M_i - \bar{M})$	$(H_i - \bar{H})(M_i - \bar{M})$
9	39	-4.92	-23.42	115.23
15	56	1.08	-6.42	-6.93
25	93	11.08	30.58	338.83
14	61	0.08	-1.42	-0.11
10	50	-3.92	-12.42	48.69
18	75	4.08	12.58	51.33
0	32	-13.92	-30.42	423.45
16	85	2.08	22.58	46.97
5	42	-8.92	-20.42	182.15
19	70	5.08	7.58	38.51
16	66	2.08	3.58	7.45
20	80	6.08	17.58	106.89
Total				1149.89
Average				104.54

Ma trận hiệp phương sai (covariance matrix)

- Covariance là độ do trên 2 chiều dữ liệu. Nếu dữ liệu là nhiều hơn 2 chiều ?

- Ma trận cov là một cách hiệu quả

$$C^{n \times n} = (c_{i,j}, c_{i,j} = cov(Dim_i, Dim_j))$$

- Ví dụ trên dữ liệu có 3 chiều (x, y, z), ma trận có dạng:

$$C = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix}$$

Eigenvector (vector riêng)

- 2 ma trận đại số có thể nhân với nhau khi có kích thước thích hợp. Eigenvector là một trường hợp đặc biệt.

- Eigenvector của một ma trận là một vector mà sau khi tích với một ma trận biến đổi thì được một vector tỉ lệ với chính nó

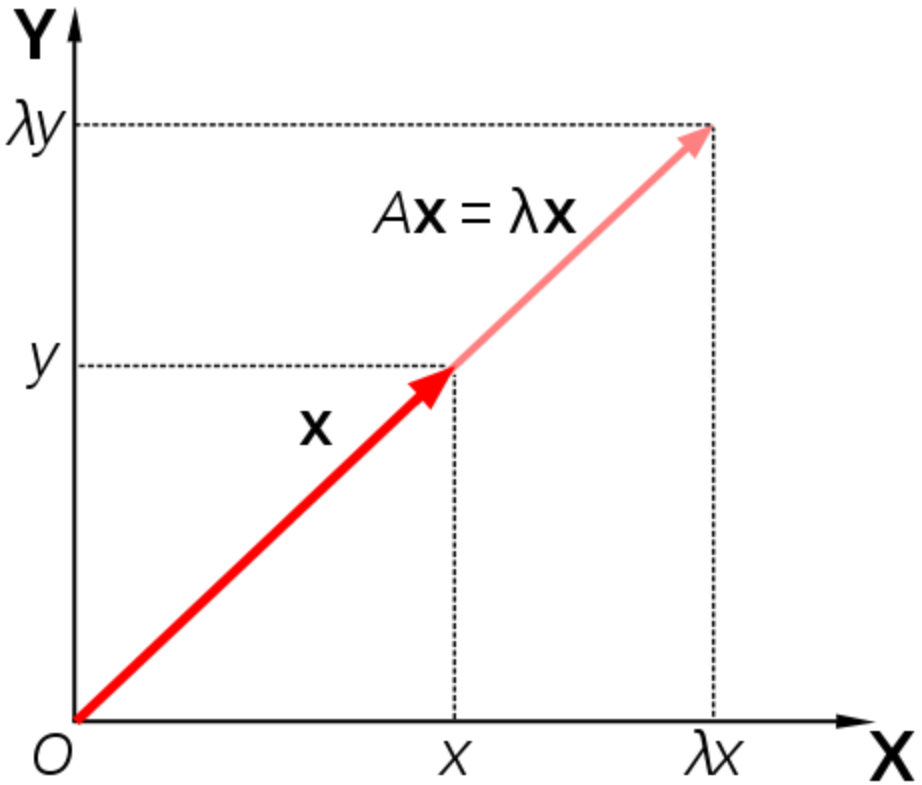
- Biểu diễn toán học: $\boldsymbol{A}\,\boldsymbol{v} = \boldsymbol{\lambda}\,\boldsymbol{v}$

Ví dụ

Eigenvectors v

$$\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix} \times \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 12 \\ 8 \end{pmatrix} = 4 \times \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

Matrix A



Tính chất của eigenvector

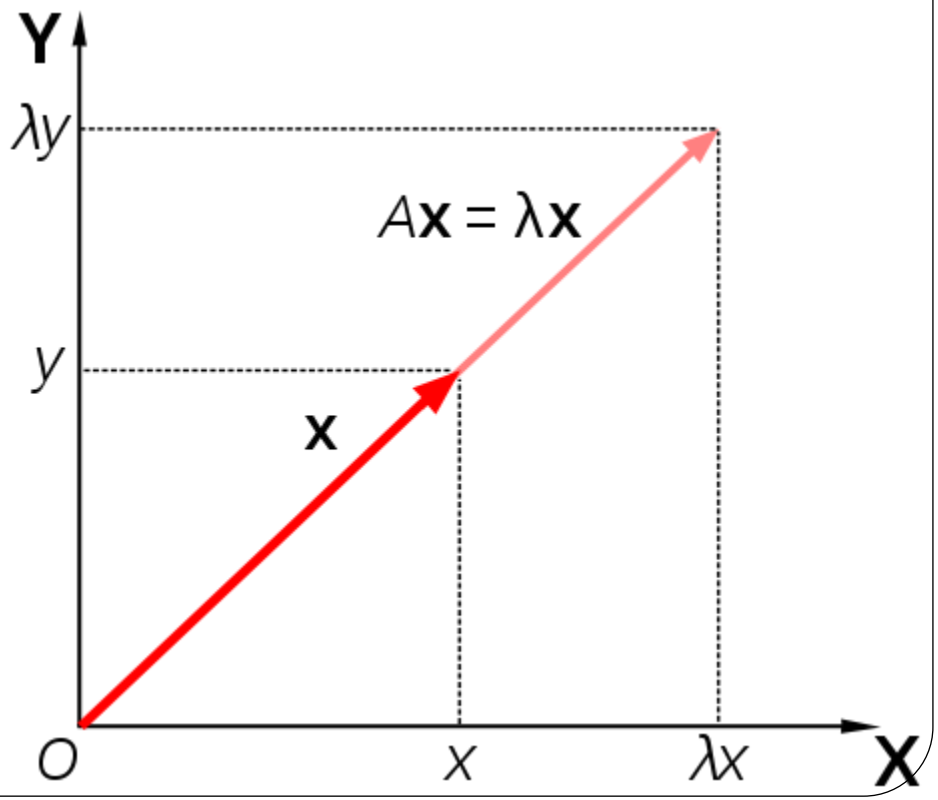
- Eigenvector chỉ được tìm thấy trong một ma trận vuông. Nhưng không phải ma trận vuông nào cũng có eigenvectors.
- Nếu ma trận vuông có kích thước n x n và có eigenvectors thì số eigenvector tìm được là n.
- Nếu scale eigenvector thì sau khi tích với ma trận biến đổi thì bội số vẫn giữ nguyên – eigenvalue
- Tất cả eigenvector của một ma trận là vuông góc với nhau từng đôi một, bất kể là nó có bao nhiêu chiều, trong toán gọi là trực giao ➔ áp dụng trong PCA
- Ngoài ra, người ta thường tìm các eigenvectors có độ dài bằng 1.

Eigenvalue (trị riêng)

Eigenvalue λ

$$\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix} \times \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 12 \\ 8 \end{pmatrix} = 4 \times \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

Eigenvectors A



Cách tìm eigenvalues

- Eigenvalue **λ** của ma trận A
- Công thức ***det(A – λI) = 0***

- Ví dụ: $A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$ $\det(A - \lambda I) = \begin{vmatrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{vmatrix}$
- λ1 = -1 ➔ v1 = (1, -1)
- λ2 = 3 ➔ v2 = (1, 1)

Nội dung chính

- Các khái niệm toán học có liên quan.
- **Mô hình PCA.**
- Eigenface.

PCA

Principle component analysis

Nội dung chính

- Các khái niệm toán học có liên quan.
- **Mô hình PCA**.
- Eigenface

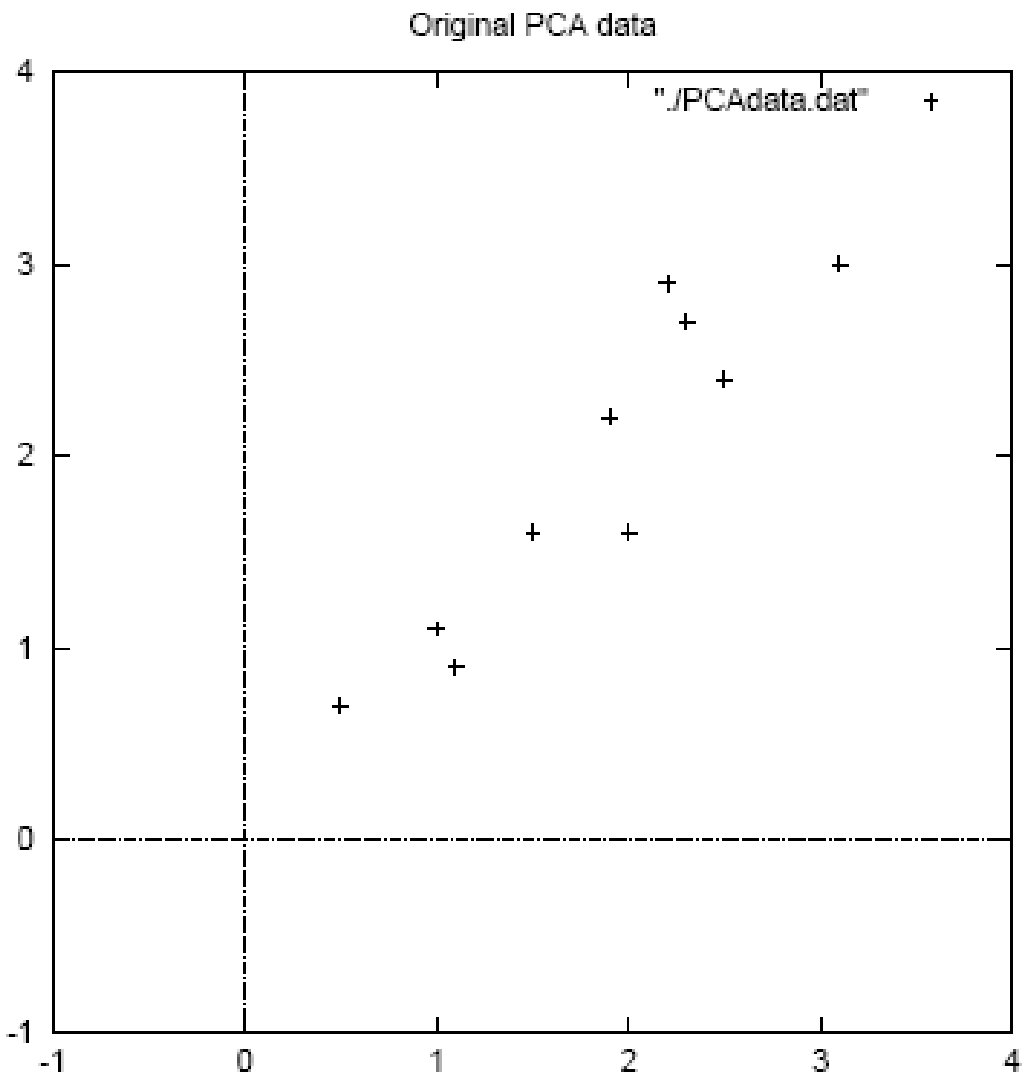
Mô hình PCA

- Là một phương pháp xác định các mẫu đặc trưng trong dữ liệu, biểu diễn lại dữ liệu để làm nổi rõ sự tương đồng và khác biệt.
- Những mẫu đặc trưng khó tìm trong dữ liệu nhiều chiều nên PCA làm giảm chiều của chúng mà không làm mất mát nhiều thông tin.
- Được ứng dụng nhiều nhất trong nhận dạng mặt người.

Bước 1: Chuẩn bị dữ liệu

Data =

x	y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9



Bước 2: Trừ cho giá trị trung bình

Data =

x	y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9

DataAdjust =

x	y
.69	.49
-1.31	-1.21
.39	.99
.09	.29
1.29	1.09
.49	.79
.19	-.31
-.81	-.81
-.31	-.31
-.71	-1.01

Bước 3: Tạo ma trận hiệp phương sai

- Vì dữ liệu có 2 chiều nên kết quả tính toán được thể hiện trên ma trận có kích thước 2 x 2

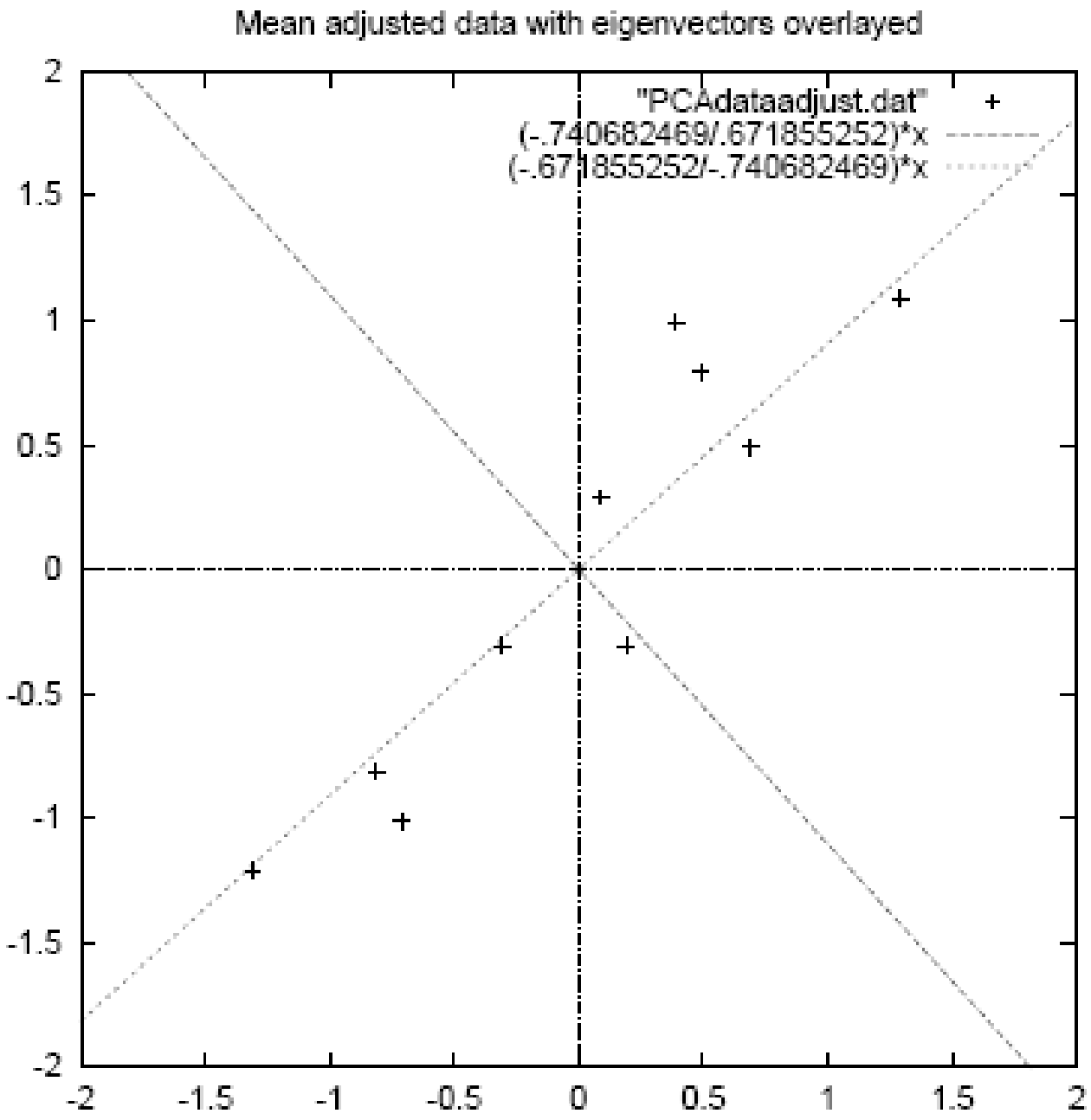
	X	Y
X	0.616555556	0.615444444
Y	0.615444444	0.716555556

Bước 4: Tìm eigenvectors và eigenvalues

- Eigenvectors và eigenvalues cho biết nhiều thông tin của dữ liệu hơn.
- Eigenvectors có độ dài đều bằng 1.

Eigenvectors	
-0.735178656	-0.677873399
0.677873399	-0.735178656

Eigenvalues
0.0490833989
1.28402771


Mean adjusted data with eigenvectors overlayed
"PCAdataadjust.dat"
(-.740682469/.671855252)*x
(-.671855252/-.740682469)*x
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

Bước 5: Chọn lựa thành phần và tạo vector đặc trưng

- Eigenvectors nào mà có eigenvalues lớn là thành phần chính của tập dữ liệu.
- Eigenvectors tìm được ở bước trên được sắp xếp theo thứ tự giảm dần theo eigenvalues tương ứng và loại bỏ những thành phần ít có ý nghĩa.
- Đây là lúc dữ liệu bị giảm chiều. Dữ liệu ban đầu từ n chiều giảm xuống còn p chiều ($p \leq n$).
- FeatureVector = (eig_1 eig_2 … eig_p)

Eigenvector
-0.677873399
-0.735178656

Bước 6: Phát sinh dữ liệu mới

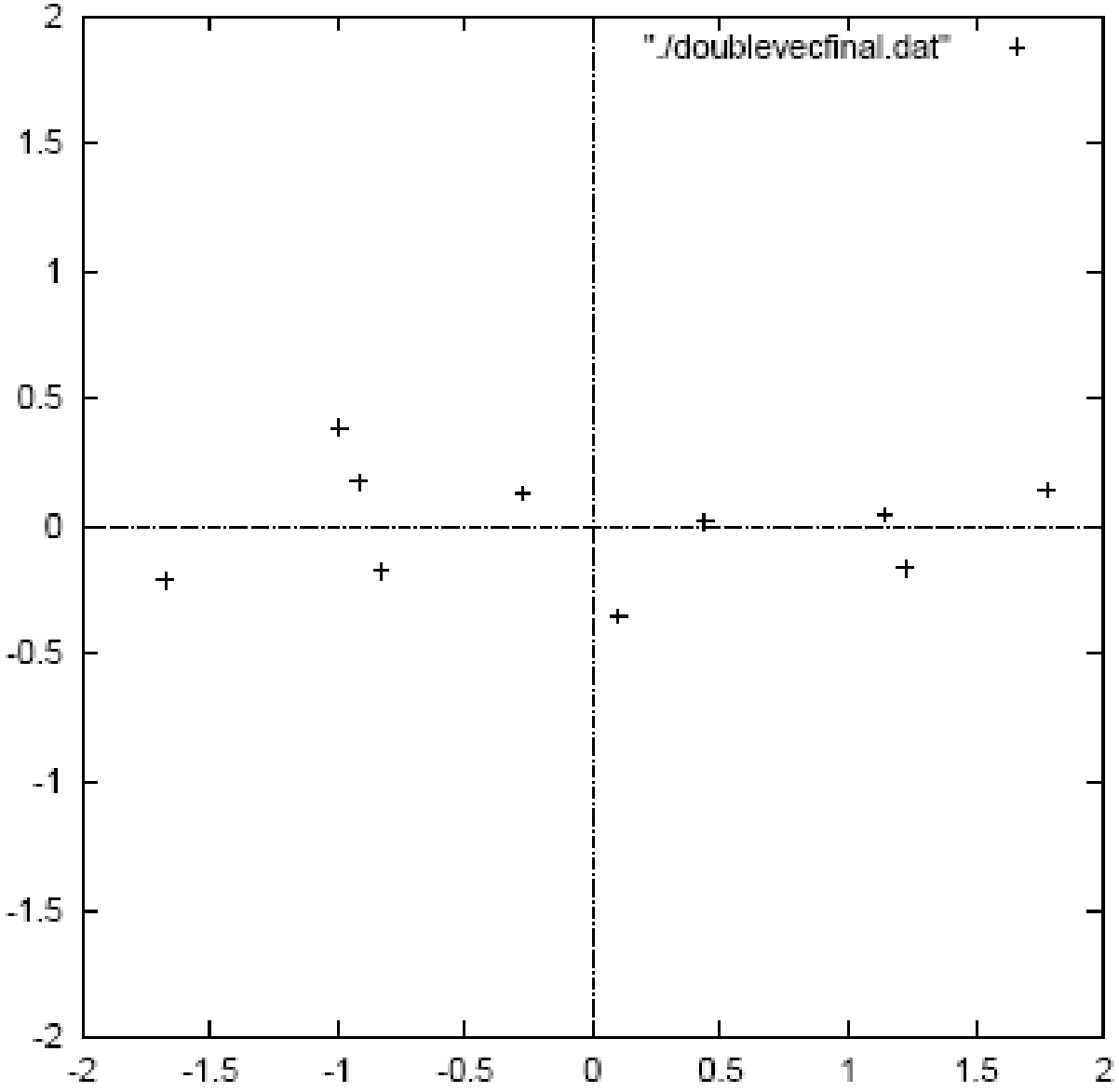
- Sau đi đã chọn được các thành phần chính là các vector đặc trưng, tạo ma trận chuyển vị của ma trận đó.
- Nhân ma trận chuyển vị vào phía bên trái của dữ liệu ban đầu.

  FinalData = RowFeatureVector x RowDataAdjust

x	y
-.827970186	-.175115307
1.77758033	.142857227
-.992197494	.384374989
-.274210416	.130417207
-1.67580142	-.209498461
-.912949103	.175282444
.0991094375	-.349824698
1.14457216	.0464172582
.438046137	.0177646297
1.22382056	-.162675287

Data transformed with 2 eigenvectors

x
-.827970186
1.77758033
-.992197494
-.274210416
-1.67580142
-.912949103
.0991094375
1.14457216
.438046137
1.22382056



Nội dung chính

- Các khái niệm toán học có liên quan.
- Mô hình PCA.
- **Eigenface**

Eigenface

- Là hệ thống nhận dạng mặt người hiệu quả và đơn giản.
- Không phụ thuộc vào mô hình 3 chiều hay các đặc điểm trên khuôn mặt (mắt, mũi, miệng, …).
- Việc phân lớp dựa trên sự kết hợp tuyến tính của các vector đặc trưng.

Các bước huấn luyện

- Chuẩn bị tập ảnh dùng cho việc huấn luyện.
- Tính toán eigenfaces từ tập huấn luyện. Giữ lại M ảnh có eigenvalues cao nhất.
- Tính lại dữ liệu ban đầu trong không gian M chiều.

Các bước nhận dạng

- Tính bộ trọng giữa ảnh mới đưa vào và các eigenfaces bằng cách chiếu ảnh vào từng eigenface.
- Xác định ảnh có phải là khuôn mặt hay không bằng cách kiểm tra ảnh có đủ gần với không gian mặt (face space) không.
- Nếu là khuôn mặt, phân lớp dựa vào trọng số để xác định người trong ảnh.
- Cập nhật lại eigenface và những mẫu trọng số.
- Nếu là khuôn mặt mà chưa biết, đưa vào csdl.

Khái niệm eigenfaces

- Xét một tấm ảnh NxN, số chiều của nó là N^2. Ví dụ với tấm ảnh 256x256 thì có được vectơ 65536 chiều.
- Cần tìm một tập các vector đặc trưng được định nghĩa trong không gian con của những ảnh ban đầu được gọi là face space.
- Những vector này là eigenvectors của một ma trận hiệp phương sai của những ảnh gốc được gọi là eigenface.

Các bước tính eigenfaces

- Cho một tập ảnh khuôn mặt $\{\Gamma_1, \Gamma_2, \Gamma_3, \ldots, \Gamma_M\}$.
- Mặt có giá trị trung bình $\psi = \frac{1}{M}\sum_{n=1}^{M}\Gamma_n$
- Độ lệch so với mặt trung bình $\phi_i = \Gamma_i - \psi$.
- Tìm ma trận hiệp phương sai $C = \frac{1}{M}\sum_{n=1}^{M}\phi_n\phi_n^T = AA^T$ với $A = [\phi_1, \phi_2, \phi_3, \ldots, \phi_M]$

Các bước tính eigenfaces

- Tìm M vector trực giao và chọn các vector tốt nhất để biểu diễn sự phân bố của dữ liệu.

$$\lambda_k = \frac{1}{M}\sum_{n=1}^{M}(u_k^T \phi_n)^2$$

- u_k và λ_k là 2 eigenvector và eigenvalue của ma trận hiệp phương sai.

Các bước tính eigenfaces

- Xét $A^TAv_i = \mu_i v_i$.
- Nhân 2 vế cho A ta có $\mathbf{AA^T}Av_i = \mu_i Av_i$.
- Xem Av_i như là eigenvectors của $AA^T = C$.
- Gọi $L = A^TA$, tìm M eigenvector trên ma trận L.
- Vector xác định sự kết hợp tuyến tính của M tập ảnh huấn luyện là

$$u_l = \sum_{k=1}^{M} v_{lk}\phi_k$$

(l = 1, 2, …, M)

Figure 1. (b) The average face Ψ.

Figure 2. Seven of the eigenfaces calculated from the input images of Figure 1.

Sử dụng eigenfaces phân lớp ảnh

- Chiếu một tấm ảnh mới vào không gian eigenfaces.

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

- Các trọng số tạo 1 vector $\Omega^T = [\omega_1, \omega_2, \ldots, \omega_{M'}]$.
- Khoảng cách euclidean đo khoảng cách giữa ảnh mới và từng lớp trong không gian mặt

$$\varepsilon_k^2 = \|\omega - \omega_k\|^2$$

Sử dụng eigenfaces phân lớp ảnh

- Để kiểm tra hình có phải là mặt hay không, chiếu ảnh vào face space, kiểm tra sự khác biệt giữa những ảnh đã có trong không gian mặt và ảnh mới.
- Ảnh được chiếu vào face space: $\phi = \Gamma - \psi$

$$\phi_f = \sum_{i=1}^{M'} \omega_i u_i$$

$$\varepsilon^2 = \|\phi - \phi_f\|^2$$

Sử dụng eigenfaces phân lớp ảnh

- Nếu $\varepsilon_k < \theta_\varepsilon$ và $\varepsilon < \theta_\varepsilon$ thì ảnh mới là mặt và thuộc về 1 lớp k.
- Nếu $\varepsilon_k < \theta_\varepsilon$ và $\varepsilon \geq \theta_\varepsilon$ thì ảnh là dạng nhiễu và không thể phân biệt.
- Nếu $\varepsilon_k \geq \theta_\varepsilon$ và $\varepsilon < \theta_\varepsilon$ thì ảnh là mặt nhưng không thuộc về lớp nào.
- Nếu $\varepsilon_k \geq \theta_\varepsilon$ và $\varepsilon \geq \theta_\varepsilon$ thì ảnh không là mặt.