

# Application of Algebra to Data Analysis: PCA

# Bài toán visualization

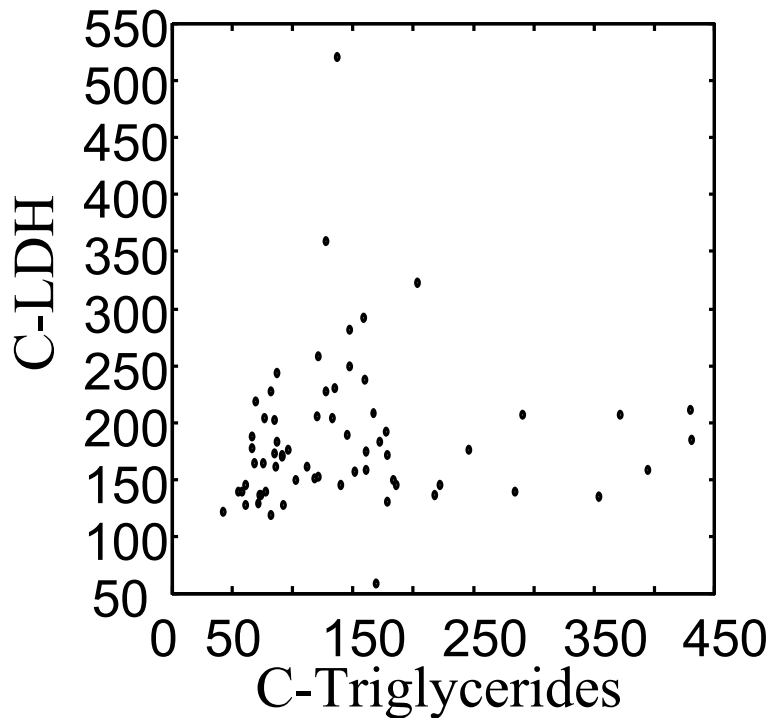
- Có 53 mẫu xét nghiệm từ 65 người
- Làm cách nào để visualize dữ liệu

	H-WBC	H-RBC	H-Hgb	H-Hct	H-MCV	H-MCH	H-MCHC
A1	8.0000	4.8200	14.1000	41.0000	85.0000	29.0000	34.0000
A2	7.3000	5.0200	14.7000	43.0000	86.0000	29.0000	34.0000
A3	4.3000	4.4800	14.1000	41.0000	91.0000	32.0000	35.0000
A4	7.5000	4.4700	14.9000	45.0000	101.0000	33.0000	33.0000
A5	7.3000	5.5200	15.4000	46.0000	84.0000	28.0000	33.0000
A6	6.9000	4.8600	16.0000	47.0000	97.0000	33.0000	34.0000
A7	7.8000	4.6800	14.7000	43.0000	92.0000	31.0000	34.0000
A8	8.6000	4.8200	15.8000	42.0000	88.0000	33.0000	37.0000
A9	5.1000	4.7100	14.0000	43.0000	92.0000	30.0000	32.0000

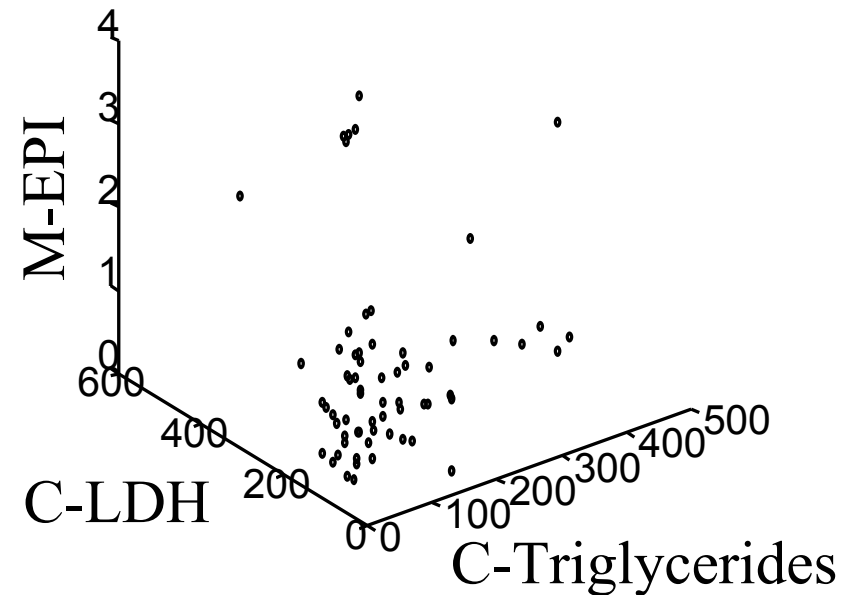
# Visualization

Làm sao để visualization: khi vector nhiều hơn 4 chiều ?

**Bi-variate**



**Tri-variate**

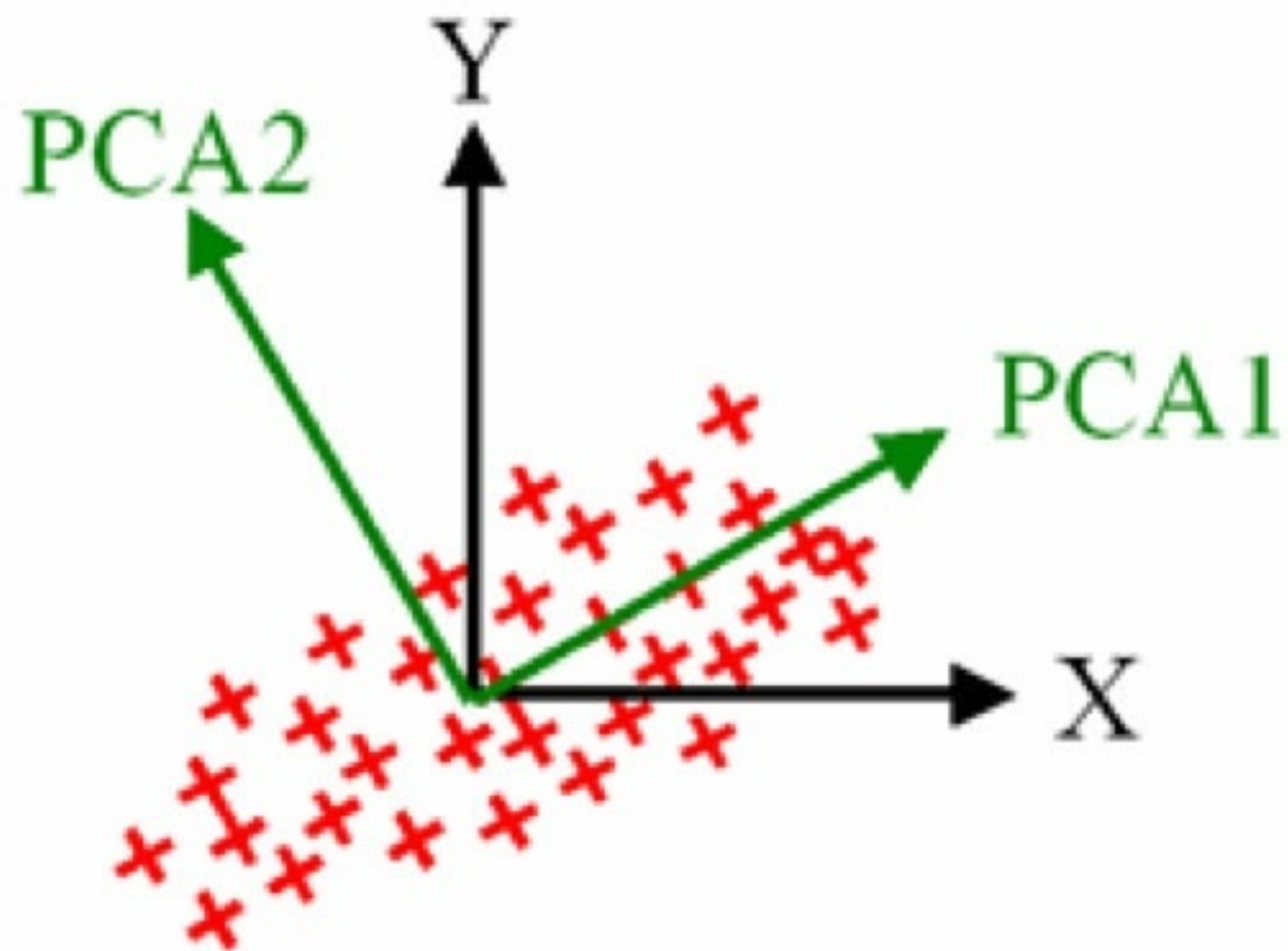


# Vấn đề (problem)

- Có cách biểu diễn dữ liệu nào tốt hơn việc vẽ hết các trục ?
- Có cần thiết phải vẽ hết 53 chiều ?
- Nếu có mối tương quan giữa các đặc trưng (feature),
  - ví dụ xét nghiệm 1 và xét nghiệm 2 kết quả tương đương Liệu chúng ta có thể tìm
- Giải pháp: Principal Component Analysis (PCA)

# Ý tưởng của PCA

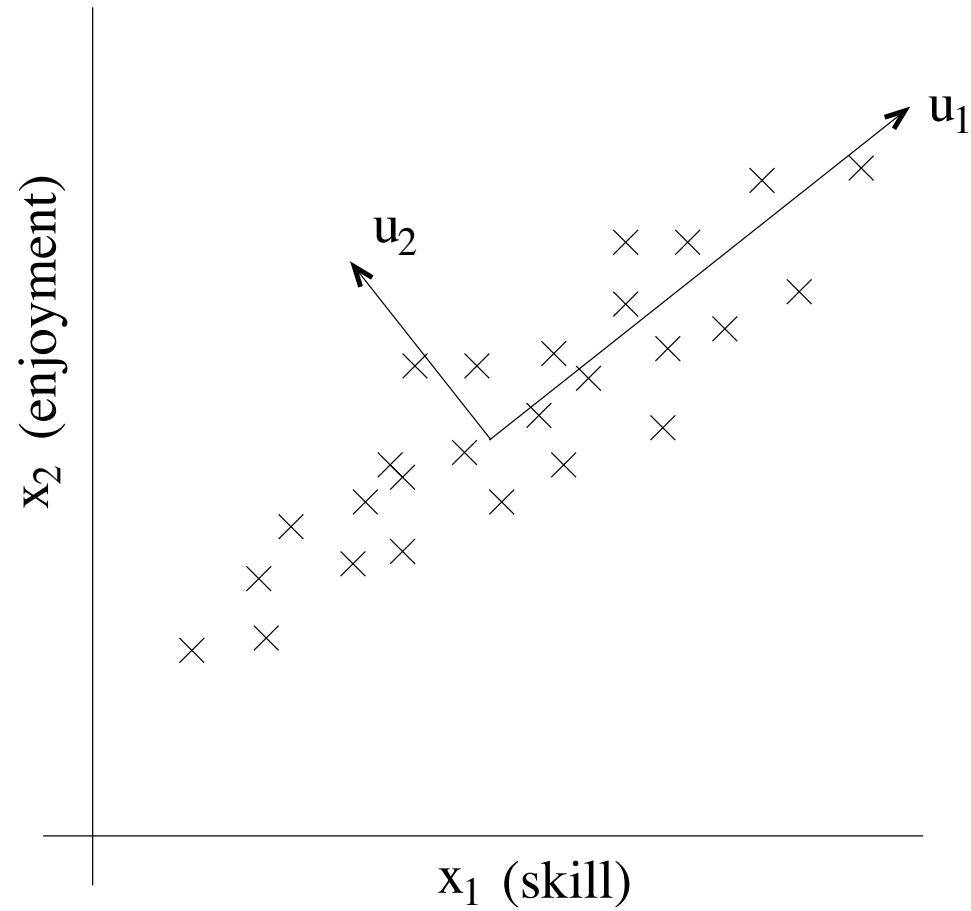
- Ý tưởng PCA:
  - Chiếu dữ liệu xuống không gian ít chiều hơn mà sao vẫn dữ được nhiều thông tin nhất có thể:
    - Tìm mặt phẳng 3-D xấp xỉ của dữ liệu
    - Tìm mặt phẳng 12-D xấp xỉ cho dữ liệu không gian 1000 chiều
- Cụ thể:
  - Chọn phép chiếu xuống không gian mà đảm bảo cực tiểu hoá sự mất mát thông tin khi khôi phục lại dữ liệu



# Mục đích của PCA

- Giả sử ta có tập dữ liệu  $m$  vector  $x_1 \rightarrow x_m$  trong  $R^n$
- Mục đích:  $x \rightarrow x'$  trong  $R^k$  với  $k < n$  và trong không gian này việc phân tích có vẻ “dễ hơn”
- Các vấn đề dẫn đến sự biến đổi
  - Giảm chiều để nén hoặc để dễ phân tích hơn
  - Giảm nhiễu: có các cặp thuộc tính  $x_i, x_j$  liên quan nhiều đến nhau
    - Ví dụ:  $x_i$ : vận tốc tối đa tính theo km,  $x_j$ : vận tốc tối đa tính theo met
    - Ví dụ:  $x_i$ : kĩ năng của 1 người trong công việc,  $x_j$ : độ yêu thích công việc của người đó
  - Muốn visualize dữ liệu về không gian 2 chiều hay 3 chiều để quan sát

- Ví dụ





# Tiền xử lý trước khi PCA

- Ta thực hiện tiền xử lý:

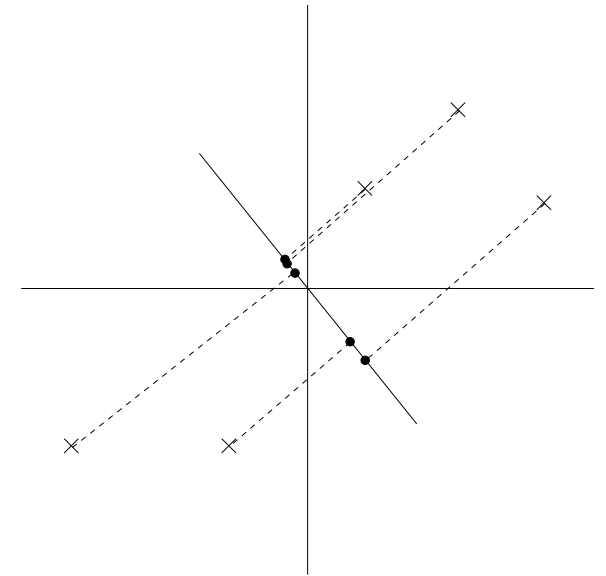
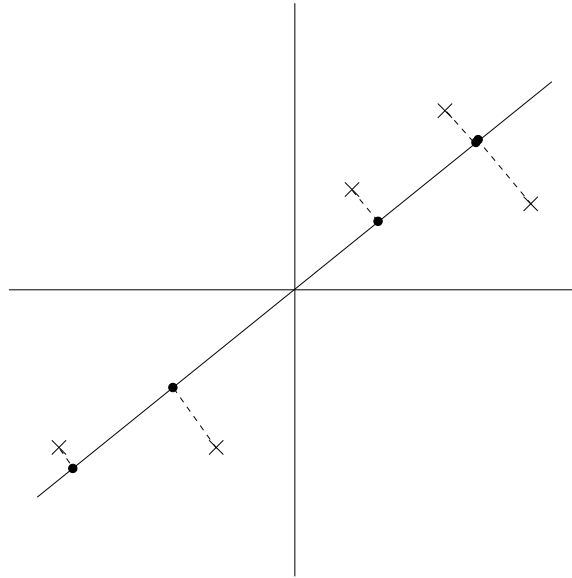
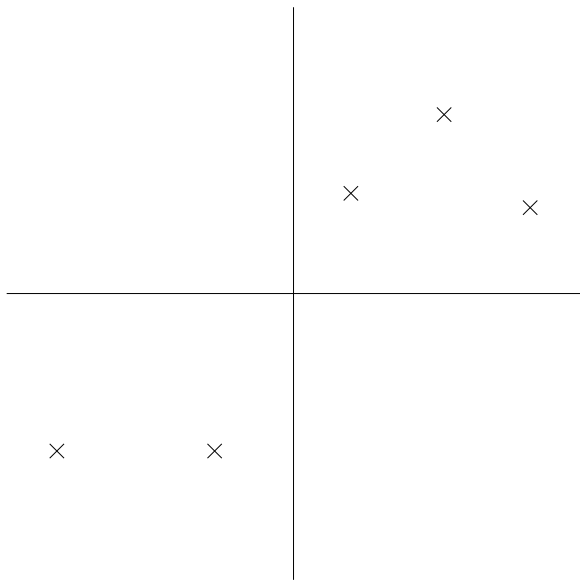
1. Let  $\mu = \frac{1}{m} \sum_{i=1}^m x^{(i)}$ .
2. Replace each  $x^{(i)}$  with  $x^{(i)} - \mu$ .
3. Let  $\sigma_j^2 = \frac{1}{m} \sum_i (x_j^{(i)})^2$
4. Replace each  $x_j^{(i)}$  with  $x_j^{(i)} / \sigma_j$ .

- Giải thích:

- Sau bước 1-2:  $\text{mean}(x) = 0$  ( $E(x) = 0$ )
- Sau bước 3-4: đưa các thuộc tính xi về cùng thước đo (scale)
  - Bước này có thể không cần khi các thuộc tính đã cùng một thước đo
    - $x_i$  là bức ảnh, Mỗi  $x_{ij}$  là 1 pixel trong bức ảnh với grayscale là 1 giá trị trong (0, 1, ..., 255)

# PCA

- Ý tưởng: Chiếu các vectors nên một hướng, tại hướng đó các điểm cách xa nhau nhất (variance lớn nhất)

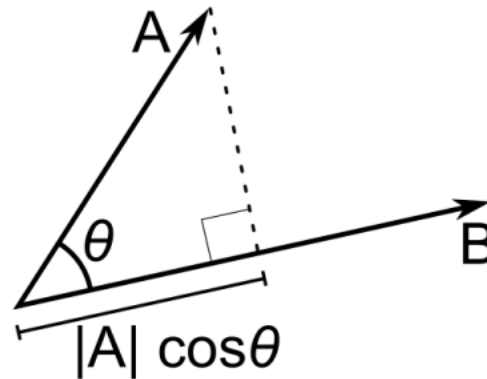


# Chọn vector đơn vị tối ưu

- Tích vô hướng chính là hình chiếu
- Điểm x chiếu lên vector đơn vị  $\mathbf{v}$  ( $\|\mathbf{v}\| = 1$ ), khi đó

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

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



# PCA

- Variance: thể hiện độ lệch từ các phần tử tới mean của nó, variance lớn thì các phần tử cách xa tâm
- Sau khi chuẩn hoá mean của data = tâm
- Bài toán trở thành tối ưu:

$$\begin{aligned}\frac{1}{m} \sum_{i=1}^m (x^{(i)T} u)^2 &= \frac{1}{m} \sum_{i=1}^m u^T x^{(i)} x^{(i)T} u \\ &= u^T \left( \frac{1}{m} \sum_{i=1}^m x^{(i)} x^{(i)T} \right) u.\end{aligned}$$

# PCA

- Đặt  $\Sigma = \frac{1}{m} \sum_{i=1}^m x^{(i)} x^{(i)T}$
- Ma trận  $\Sigma$  là đối xứng
- Bài toán tối ưu giống bài toán:
  - Hàm số  $f(\mathbf{x}) = \mathbf{x}^T A \mathbf{x}$  trong đó  $\|\mathbf{x}\|_2 = 1$ 
    - Hàm số đạt giá trị lớn nhất khi  $\mathbf{x}$  = vector riêng của trị riêng lớn nhất
    - Hàm số đạt giá trị nhỏ nhất khi  $\mathbf{x}$  = vector riêng của trị riêng nhỏ nhất

# Ma trận $\Sigma$

- Ma trận  $\Sigma$  chính là ma trận covariance giữa các thuộc tính

$$\text{covariance}(X, Y) = \frac{\sum_{i=1}^n (\bar{X}_i - \bar{X})(\bar{Y}_i - \bar{Y})}{(n-1)}$$

- $V = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$  thì  $vv^T = \begin{bmatrix} v_1*v_1, v_1*v_2, v_1*v_3 \\ v_2*v_1, v_2*v_2, v_2*v_3 \\ v_3*v_1, v_3*v_2, v_3*v_3 \end{bmatrix}$

- Lưu ý là ta đã chuẩn hoá để  $\text{mean} = 0$

# Ma trận Covariance

- Ma trận

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

where

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

- Viết gọn lại  $\underline{\mathbf{X}} = N \times m$  data matrix,

$$\Sigma \leftarrow \underline{\mathbf{X}} \underline{\mathbf{X}}^T$$

# PCA

- Giảm data xuống k chiều: chọn k trị riêng lớn nhất

$$\lambda_1 \geq \lambda_2 \geq \dots \lambda_n$$

$$y^{(i)} = \begin{bmatrix} u_1^T x^{(i)} \\ u_2^T x^{(i)} \\ \vdots \\ u_k^T x^{(i)} \end{bmatrix} \in \mathbb{R}^k.$$

- Nói cách khác ta có thể lấy  $XV$  ( $V$  là k vector riêng ứng với k trị riêng lớn nhất) để biểu diễn data



# PCA cũng có nghĩa là minimize lỗi khôi phục dữ liệu ban đầu (minimize reconstruction error)

Let  $\mathbf{x} \in \mathbb{R}^N$

Let  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_m] \in \mathbb{R}^{N \times m}$ ,  
 $m$ : number of instances,  $N$ : dimension

Let  $\mathbf{U} = \begin{pmatrix} \mathbf{u}_1^T \\ \vdots \\ \mathbf{u}_N^T \end{pmatrix} \in \mathbb{R}^{N \times N}$  orthogonal matrix,  $\mathbf{U}\mathbf{U}^T = \mathbf{I}_N$

$$\mathbf{y} \doteq \mathbf{U}\mathbf{x}, \mathbf{x} = \mathbf{U}^T\mathbf{y} = \sum_{i=1}^N \mathbf{u}_i y_i$$

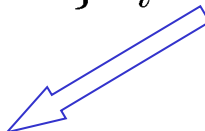
$\hat{\mathbf{x}} \doteq \sum_{i=1}^M \mathbf{u}_i y_i, (M \leq N)$  approximation of  $\mathbf{x}$   
using  $M$  basis vectors only.

$$\epsilon^2 \doteq \mathbb{E}\{\|\mathbf{x} - \hat{\mathbf{x}}\|^2\} = \frac{1}{m} \sum_{j=1}^m \|\mathbf{x}_j - \hat{\mathbf{x}}_j\|^2, \text{ average error}$$

**GOAL:**

$$\arg \min_{\mathbf{U}} \epsilon^2, \text{ s.t } \mathbf{U}^T \mathbf{U} = \mathbf{I}_N$$

# Minimizing Reconstruction Error

$$\begin{aligned}\varepsilon^2 &= \mathbb{E}\{\|\mathbf{x} - \hat{\mathbf{x}}\|^2\} = \mathbb{E}\{\|\sum_{i=1}^N \mathbf{u}_i y_i - \sum_{i=1}^M \mathbf{u}_i y_i\|^2\} \\&= \mathbb{E}\{\sum_{i=M+1}^N y_i \mathbf{u}_i^T \mathbf{u}_i y_i\} = \sum_{i=M+1}^N \mathbb{E}\{y_i^2\} \\&= \sum_{i=M+1}^N \mathbb{E}\{(\mathbf{u}_i^T \mathbf{x})(\mathbf{x}^T \mathbf{u}_i)\} \\&= \sum_{i=M+1}^N \mathbf{u}_i^T \mathbb{E}\{\mathbf{x} \mathbf{x}^T\} \mathbf{u}_i \quad \mathbf{x} \text{ is centered!} \\&= \sum_{i=M+1}^N \mathbf{u}_i^T \Sigma \mathbf{u}_i\end{aligned}$$


# PCA

- Tổng hợp lại các bước tính PCA:
  - Tiền xử lý dữ liệu
  - Tính ma trận covariance của dữ liệu
  - Tính trị riêng và vector của ma trận
  - Chọn ra  $k$  trị riêng lớn nhất
  - Biểu diễn các điểm dữ liệu theo  $k$  vector riêng ứng với  $k$  trị riêng này

# PCA và SVD

- Nhớ lại SVD:
  - Với ma trận  $X$  bất kì ta có thể phân tích ma trận này bằng tích ba ma trận  $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ ,
  - Trong đó  $U, V$  là hai ma trận trực giao ( $UU^T = I, VV^T = I$ ,  $S$  là ma trận chéo)
- $\Sigma = XX^T = \mathbf{U}\mathbf{S}\mathbf{V}^T \mathbf{V}\mathbf{S}\mathbf{U}^T = \mathbf{U}\mathbf{S}^2\mathbf{U}^T$  Lưu ý  $S$  là ma trận chéo nên mới nhân đc như thế này
- Từ trên ta thấy Các cột của  $U$  cũng là các vector riêng của  $\Sigma$

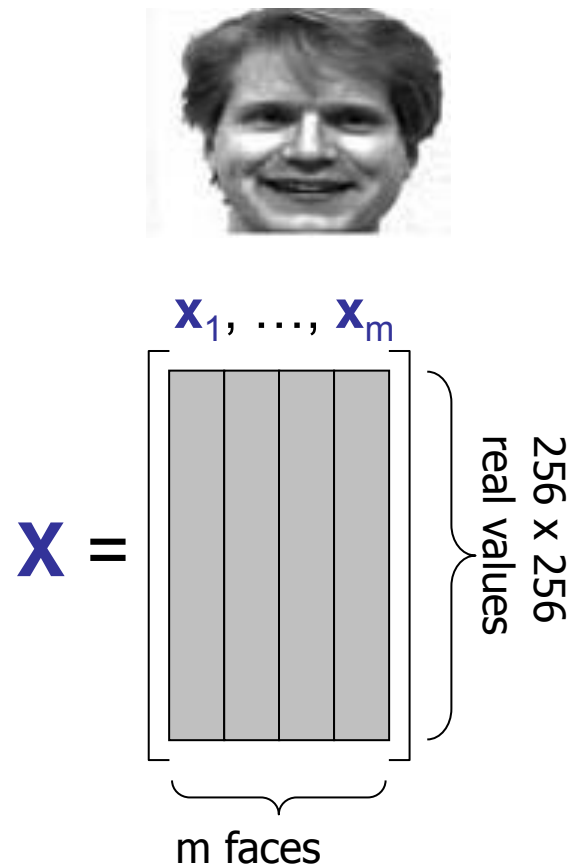
# PCA với số chiều dữ liệu lớn

- Giả sử làm việc với dữ liệu ảnh:  $256 \times 256$



# PCA với dữ liệu lớn

- Không thể thực hiện
  - Mỗi khuôn mặt  $x$  thuộc vào  $\mathbb{R}^{256 \times 256}$
  - $256 \times 256 \sim 64k$
  - Khi đó  $\Sigma = XX^T$  kích thước  $64k \times 64k \rightarrow$  lớn



# Vấn đề độ phức tạp tính toán

- Giả sử dữ liệu có  $m$  vector, mỗi vector kích thước  $N$ 
  - $m = 500$  khuôn mặt, mỗi khuôn mặt có kích thước  $N = 64k$
- Ma trận covariance có kích thước  $N \times N$ , tính vector riêng:
  - Tìm  $N$  vector riêng/trị riêng độ phức tạp  $O(N^3)$
  - Tìm  $k$  vector riêng/trị riêng độ phức tạp  $O(kN^2)$ -
- Khi  $N = 64k \rightarrow$  thời gian lâu

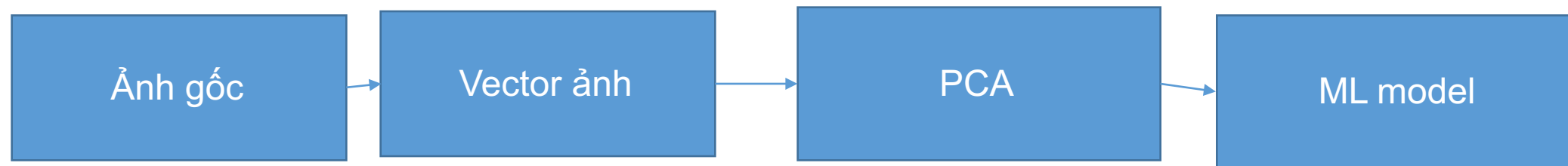
# Giải pháp

- Thấy rằng  $m \ll 64K$
- Sử dụng  $L = X^T X$  thay vì  $XX^T$
- Chứng minh được nếu  $\mathbf{v}$  là vector riêng của  $L$  thì  $X\mathbf{v}$  là vector riêng của  $\Sigma$

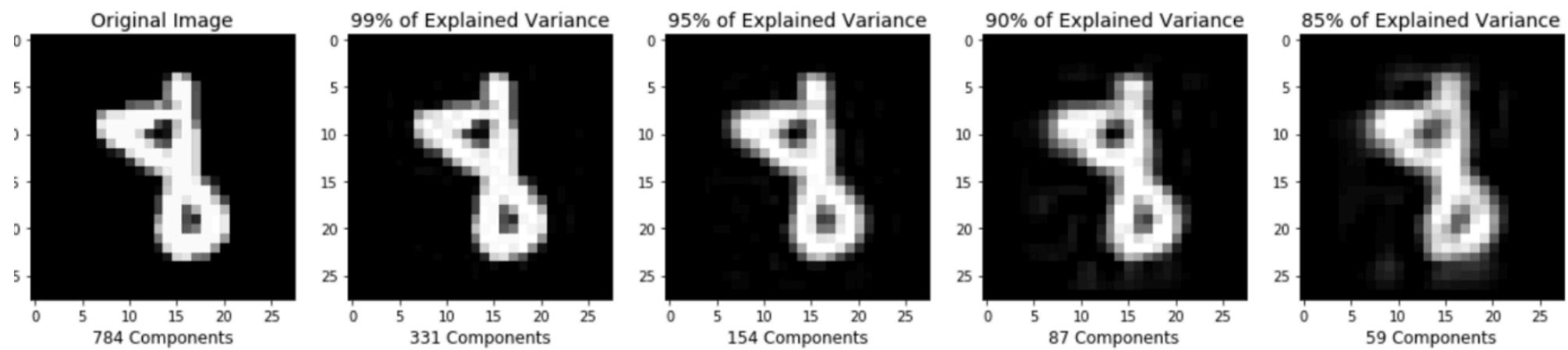
$$\begin{aligned}\text{Proof: } \quad \mathbf{L} \mathbf{v} &= \gamma \mathbf{v} \\ \mathbf{X}^T \mathbf{X} \mathbf{v} &= \gamma \mathbf{v} \\ \mathbf{X} (\mathbf{X}^T \mathbf{X} \mathbf{v}) &= \mathbf{X} (\gamma \mathbf{v}) = \gamma \mathbf{X} \mathbf{v} \\ (\mathbf{X} \mathbf{X}^T) \mathbf{X} \mathbf{v} &= \gamma (\mathbf{X} \mathbf{v}) \\ \Sigma (\mathbf{X} \mathbf{v}) &= \gamma (\mathbf{X} \mathbf{v})\end{aligned}$$



# Ví dụ của áp dụng PCA vào phân loại ảnh



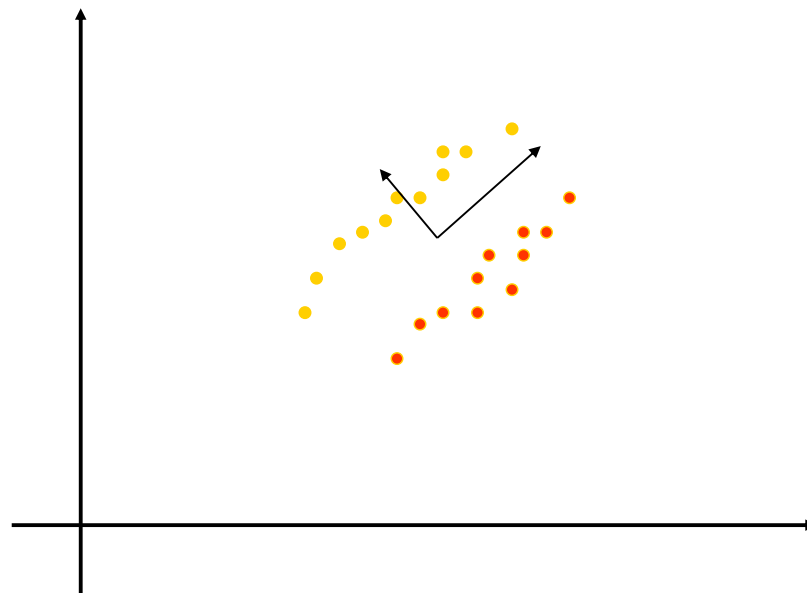
# Ví dụ về áp dụng PCA vào dữ liệu ảnh



Original image (left) with Different Amounts of Variance Retained

# Các vấn đề với PCA

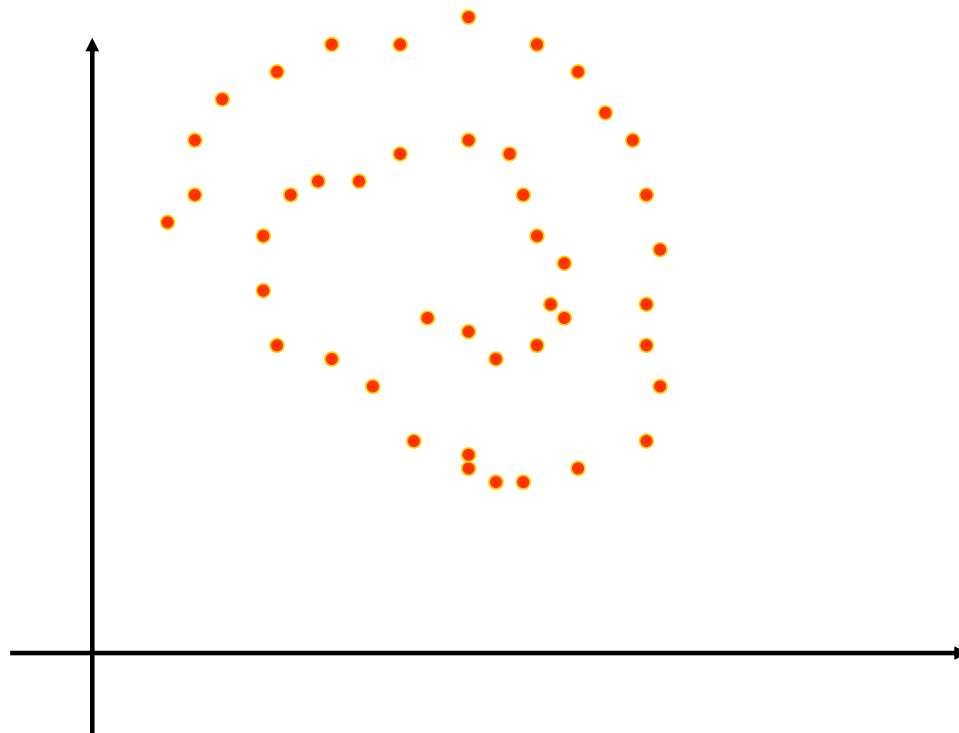
- PCA không sử dụng được thông tin về nhãn, PCA  $\rightarrow$  maximize variance



- Sau khi PCA kết quả chưa chắc tốt hơn so với ban đầu

# Các vấn đề của PCA

- PCA không xử lý được dữ liệu không tuyến tính (non-linear)



# Tổng kết lại PCA

- PCA
  - Tìm ra các vector cơ sở trực chuẩn cho dữ liệu
  - Sắp xếp trọng số các chiều theo độ quan trọng
  - Loại bỏ những chiều ít quan trọng
- Ứng dụng
  - Lọc nhiễu
  - Giảm chiều dữ liệu
  - visualize data
- Hạn chế
  - Không tính đến nhãn của dữ liệu
  - Chỉ xử lý được các mối quan hệ tuyến tính

# Exercises

- Thực hiện PCA Vào dữ liệu ảnh:
  - Bộ dataset cho ảnh: MNIST original
  - Thực hiện PCA trên tập ảnh dùng thư viện của sklearn
  - Hiển thị ảnh sau khi thực hiện PCA để so sánh với ảnh ban đầu