HCMUT EE MACHINE LEARNING & IOT LAB

Buổi 11

Dimensionality Reduction

Presented By: Văn Thịnh

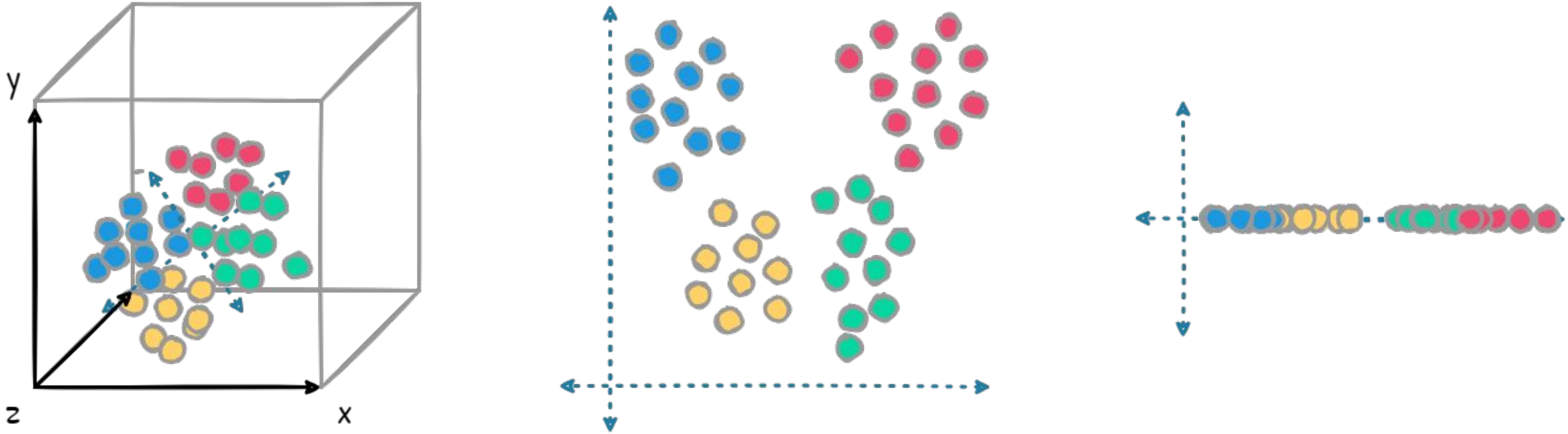
Ho Chi Minh City – 08/2025

Table of Content

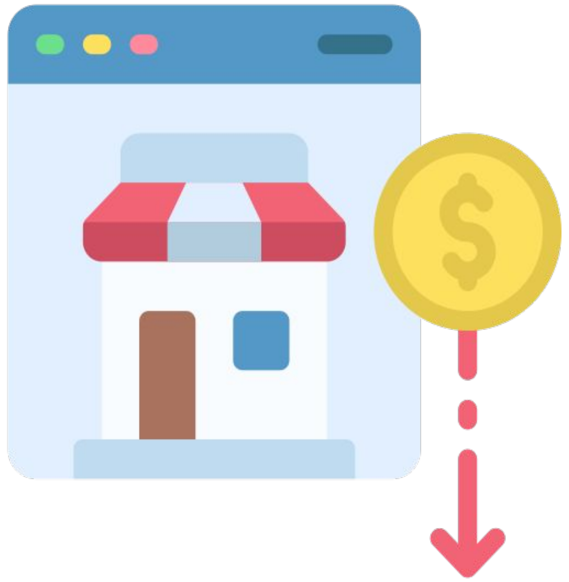
I. Giới thiệu

I. Giới thiệu

Dimensionality Reduction

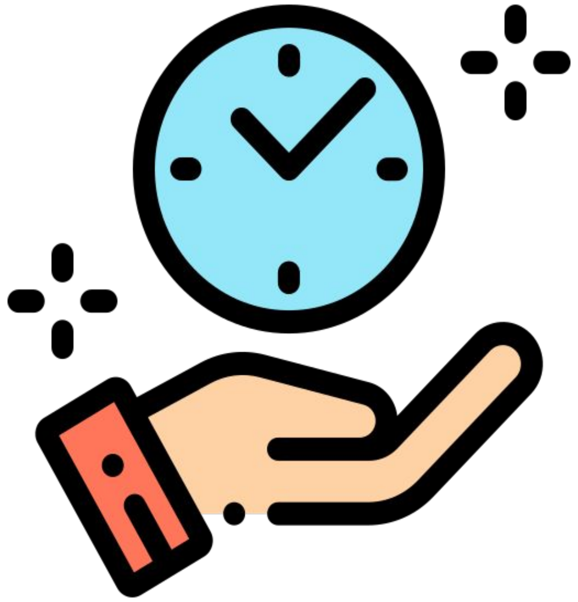


I. Giới thiệu



Tối ưu lưu trữ

Giảm kích thước mô hình



Giảm thời gian huấn luyện/ triển khai

II. Singular Value Decomposition

I. Singular Value Decomposition

1. Cơ sở toán

- Trị riêng và vector riêng

Cho một ma trận vuông $\mathbf{A} \in \mathbb{R}^{n \times n}$, nếu số vô hướng λ và vector $\mathbf{x} \neq \mathbf{0} \in \mathbb{R}^n$ thoả mãn:

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$$

thì λ được gọi là một trị riêng của $\mathbf{A}$ và $\mathbf{x}$ được gọi là vector riêng tương ứng với trị riêng đó.

Một vài tính chất:

1. Nếu $\mathbf{x}$ là một vector riêng của $\mathbf{A}$ ứng với λ thì $k\mathbf{x}, k \neq 0$ cũng là vector riêng ứng với trị riêng đó.
2. Mọi ma trận vuông bậc n đều có n trị riêng (kể cả lặp) và có thể là các số phức.
3. Với ma trận đối xứng, tất cả các trị riêng đều là các số thực.
4. Với *ma trận xác định dương*, tất cả các trị riêng của nó đều là các số thực dương. Với *ma trận nửa xác định dương*, tất cả các trị riêng của nó đều là các số thực không âm.

I. Singular Value Decomposition

1. Cơ sở toán

- Hệ trực giao và trực chuẩn

Một hệ cơ sở $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m \in \mathbb{R}^m$ được gọi là *trực giao* (orthogonal) nếu mỗi vector là khác 0 và tích của hai vector khác nhau bất kỳ bằng 0:

$$\mathbf{u}_i \neq \mathbf{0}; \quad \mathbf{u}_i^T \mathbf{u}_j = 0 \; \forall \, 1 \leq i \neq j \leq m$$

Một hệ cơ sở $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m \in \mathbb{R}^m$ được gọi là *trực chuẩn* (orthonormal) nếu nó là một hệ *trực giao* và độ dài Euclidean (norm 2) của mỗi vector bằng 1:

$$\mathbf{u}_i^T \mathbf{u}_j = \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{otherwise} \end{cases} \qquad (4)$$

Gọi $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m]$ với $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m \in \mathbb{R}^m$ là *trực chuẩn*, thế thì từ (4) có thể suy ra ngay:

$$\mathbf{U}\mathbf{U}^T = \mathbf{U}^T\mathbf{U} = \mathbf{I}$$

trong đó $\mathbf{I}$ là ma trận đơn vị bậc m. Ta gọi $\mathbf{U}$ là *ma trận trực giao* (orthogonal matrix). *Ma trận loại này không được gọi là ma trận trực chuẩn, không có định nghĩa cho ma trận trực chuẩn.*

I. Singular Value Decomposition

1. Cơ sở toán

- Hệ trực giao và trực chuẩn

Một vài tính chất:

1. $\mathbf{U}^{-1} = \mathbf{U}^T$: nghịch đảo của một ma trận trực giao chính là chuyển vị của nó.
2. Nếu $\mathbf{U}$ là ma trận trực giao thì chuyển vị của nó $\mathbf{U}^T$ cũng là một ma trận trực giao.
3. Định thức (determinant) của ma trận trực giao bằng 1 hoặc -1. Điều này có thể suy ra từ việc $\det(\mathbf{U}) = \det(\mathbf{U}^T)$ và $\det(\mathbf{U})\det(\mathbf{U}^T) = \det(\mathbf{I}) = 1$.
4. Ma trận trực giao thể hiện cho phép xoay (rotate) một vector. Giả sử có hai vector $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$ và ma trận trực giao $\mathbf{U} \in \mathbb{R}^{m \times m}$. Dùng ma trận này để xoay hai vector trên ta được $\mathbf{Ux}, \mathbf{Uy}$. Tích vô hướng của hai vector mới là:

$$(\mathbf{Ux})^T(\mathbf{Uy}) = \mathbf{x}^T\mathbf{U}^T\mathbf{Uy} = \mathbf{x}^T\mathbf{y}$$

   như vậy *phép xoay không làm thay đổi tích vô hướng giữa hai vector.*

5. Giả sử $\hat{\mathbf{U}} \in \mathbb{R}^{m \times r}, r < m$ là môt ma trận con của ma trận trực giao $\mathbf{U}$ được tạo bởi r cột của $\mathbf{U}$, ta sẽ có $\hat{\mathbf{U}}^T\hat{\mathbf{U}} = \mathbf{I}_r$. Việc này có thể được suy ra từ (4).

I. Singular Value Decomposition

1. Cơ sở toán

- Chéo hóa ma trận

Giả sử $\mathbf{x}_1, \ldots, \mathbf{x}_n \neq \mathbf{0}$ là các vector riêng của một ma trận vuông $\mathbf{A}$ ứng với các trị riêng $\lambda_1, \ldots, \lambda_n$ (có thể lặp hoặc là các số phức) của nó. Tức là $\mathbf{A}\mathbf{x}_i = \lambda_i \mathbf{x}_i, \ \forall i = 1, \ldots, n$.

Đặt $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$, và $\mathbf{X} = \left[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\right]$, ta sẽ có $\mathbf{A}\mathbf{X} = \mathbf{X}\boldsymbol{\Lambda}$. Hơn nữa, nếu các trị riêng $\mathbf{x}_1, \ldots, \mathbf{x}_n$ là độc lập tuyến tính, ma trận $\mathbf{X}$ là một ma trận khả nghịch. Khi đó ta có thể viết $\mathbf{A}$ dưới dạng tích của ba ma trận:

$$\mathbf{A} = \mathbf{X}\boldsymbol{\Lambda}\mathbf{X}^{-1} \tag{1.28}$$

Các vector riêng $\mathbf{x}_i$ thường được chọn sao cho $\mathbf{x}_i^T\mathbf{x}_i = 1$. Cách biểu diễn một ma trận như (1.28) được gọi là *eigendecomposition* vì nó tách ra thành tích của các ma trận đặc biệt dựa trên vector riêng (eigenvectors) và trị riêng (eigenvalues). Ma trận các trị riêng $\boldsymbol{\Lambda}$ là một ma trận đường chéo. Vì vậy, cách khai triển này cũng có tên gọi là chéo hoá ma trận.

I. Singular Value Decomposition

2. Phân tích ma trận

Singular value decomposition

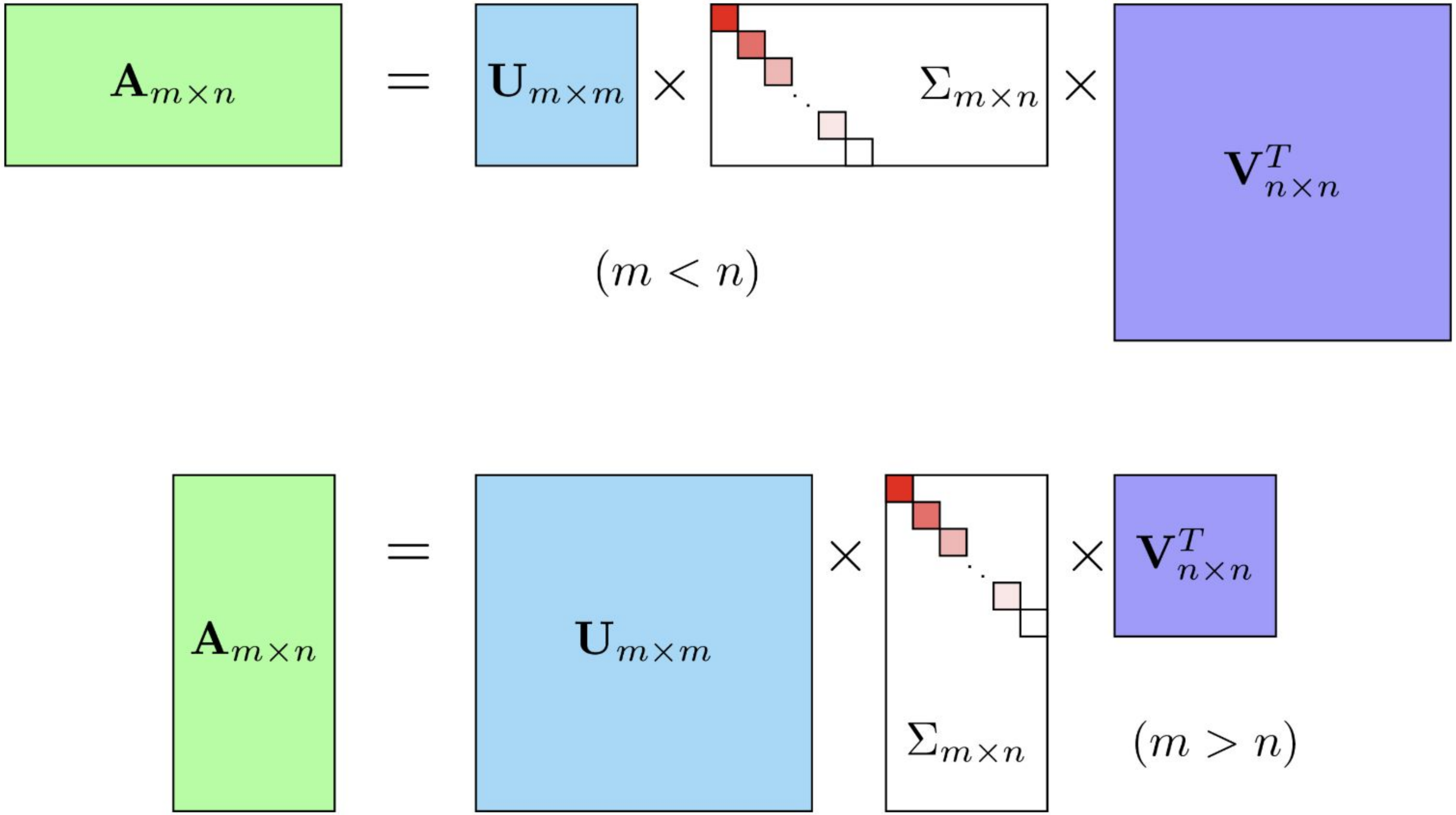
Một ma trận $\mathbf{A}_{m\times n}$ *bất kỳ đều có thể phân tích thành dạng:*

$$\mathbf{A}_{m\times n} = \mathbf{U}_{m\times m}\mathbf{\Sigma}_{m\times n}(\mathbf{V}_{n\times n})^T \tag{20.4}$$

Trong đó, $\mathbf{U}, \mathbf{V}$ *là các ma trận trực giao,* $\mathbf{\Sigma}$ *là một ma trận đường chéo cùng kích thước với* $\mathbf{A}$*. Các phần tử trên đường chéo chính của* $\mathbf{\Sigma}$ *là không âm và được theo thứ tự giảm dần* $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0 = 0 = \cdots = 0$*. Số lượng các phần tử khác 0 trong* $\mathbf{\Sigma}$ *chính là rank của ma trận* $\mathbf{A}$*:* $r = rank(\mathbf{A})$*.*

I. Singular Value Decomposition

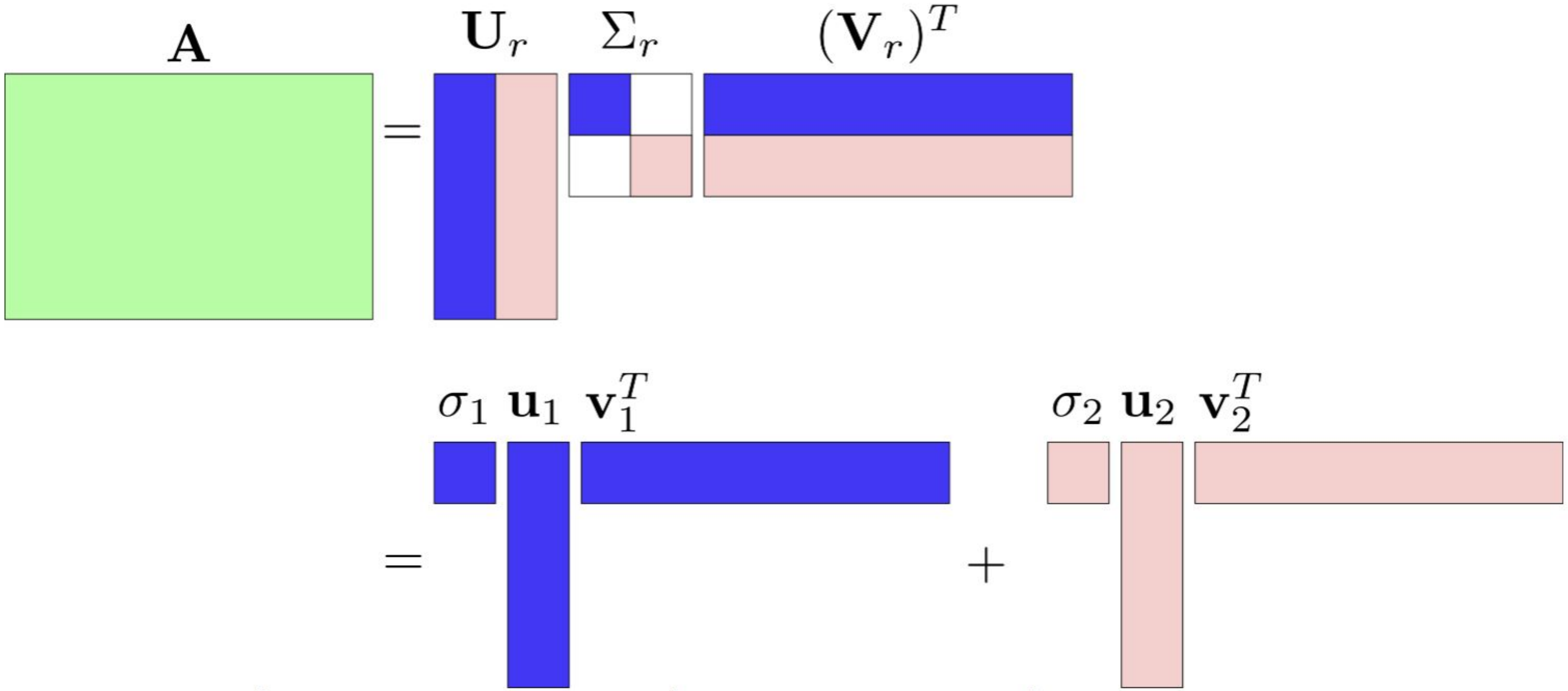
2. Phân tích ma trận



I. Singular Value Decomposition

2. Phân tích ma trận

$$\mathbf{A} = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^2 + \cdots + \sigma_r \mathbf{u}_r \mathbf{v}_r^T$$



I. Singular Value Decomposition

2. Phân tích ma trận

Chú ý rằng trong ma trận Σ, các giá trị trên đường chéo là không âm và giảm dần $\sigma_1 \geq \sigma_2 \geq \ldots, \geq \sigma_r \geq 0 = 0 = \cdots = 0$. Thông thường, chỉ một lượng nhỏ các σ_i mang giá trị lớn, các giá trị còn lại thường nhỏ và gần 0. Khi đó ta có thể xấp xỉ ma trận $\mathbf{A}$ bằng tổng của $k < r$ ma trận có rank 1:

$$\mathbf{A} \approx \mathbf{A}_k = \mathbf{U}_k \Sigma_k (\mathbf{V}_k)^T = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^2 + \cdots + \sigma_k \mathbf{u}_k \mathbf{v} k^T \quad (7)$$

I. Singular Value Decomposition

2. Phân tích ma trận

Định lý:

$$||\mathbf{A} - \mathbf{A}_k||_F^2 = \sum_{i=k+1}^{r} \sigma_i^2 \quad (8)$$

Sai số tương đối

$$\frac{||\mathbf{A} - \mathbf{A}_k||_F^2}{||\mathbf{A}||_F^2} = \frac{\sum_{i=k+1}^{r} \sigma_i^2}{\sum_{j=1}^{r} \sigma_j^2} \quad (17)$$

Ví dụ, nếu ta muốn giữ lại ít nhất 90% lương thông tin trong $\mathbf{A}$, trước hết ta tính $\sum_{j=1}^{r} \sigma_j^2$, sau đó chọn k là số nhỏ nhất sao cho:

$$\frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_{j=1}^{r} \sigma_j^2} \geq 0.9$$

I. Singular Value Decomposition

3. Hệ quả

Để lưu ảnh với Truncated SVD, ta sẽ lưu các ma trận $\mathbf{U}_k \in \mathbb{R}^{m\times k}, \Sigma_k \in \mathbb{R}^{k\times k}, \mathbf{V}_k \in \mathbb{R}^{n\times k}$. Tổng số phần tử phải lưu là $k(m+n+1)$ với chú ý rằng ta chỉ cần lưu các giá trị trên đường chéo của Σ_k. Giả sử mỗi phần tử được lưu bởi một số thực 4 byte, thế thì số byte cần lưu trữ là $4k(m+n+1)$. Nếu so giá trị này với ảnh gốc có kích thước mn, mỗi giá trị là 1 số nguyên 1 byte, tỉ lệ nén là:

$$\frac{4k(m+n+1)}{mn}$$

Khi $k \ll m, n$, ta được một tỉ lệ nhỏ hơn 1. Trong ví dụ của chúng ta $m = 960, n = 1440, k = 100$, ta có tỉ lệ nén xấp xỉ 0.69, tức đã tiết kiệm được khoảng 30% bộ nhớ.

III. Principal Component Analysis

1.Principal Component Analysis

1. Cơ sở toán

- Biểu diễn vector trong hệ các cơ sở khác nhau

Mỗi vector cột $\mathbf{x} = [x_1, x_2, \dots, x_D] \in \mathbb{R}^D$, biểu diễn của nó trong hệ đơn vị là:

$$\mathbf{x} = x_1\mathbf{e}_1 + x_2\mathbf{e}_2 + \dots + x_D\mathbf{e}_D$$

Giả sử có một hệ cơ sở khác $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_D$ (các vector này độc lập tuyến tính), vậy thì biểu diễn của vector $\mathbf{x}$ trong hệ cơ sở mới này có dạng:

$$\mathbf{x} = y_1\mathbf{u}_1 + y_2\mathbf{u}_2 + \dots + y_D\mathbf{u}_D = \mathbf{U}\mathbf{y}$$

$\mathbf{U}$ là ma trận mà cột thứ d của nó chính là vector $\mathbf{u}_d$. Lúc này, vector $\mathbf{y}$ chính là biểu diễn của $\mathbf{x}$ trong hệ cơ sở mới. Bộ các số $y_d, d = 1, 2, \dots, D$ là duy nhất vì $\mathbf{y}$ có thể tính được bằng:

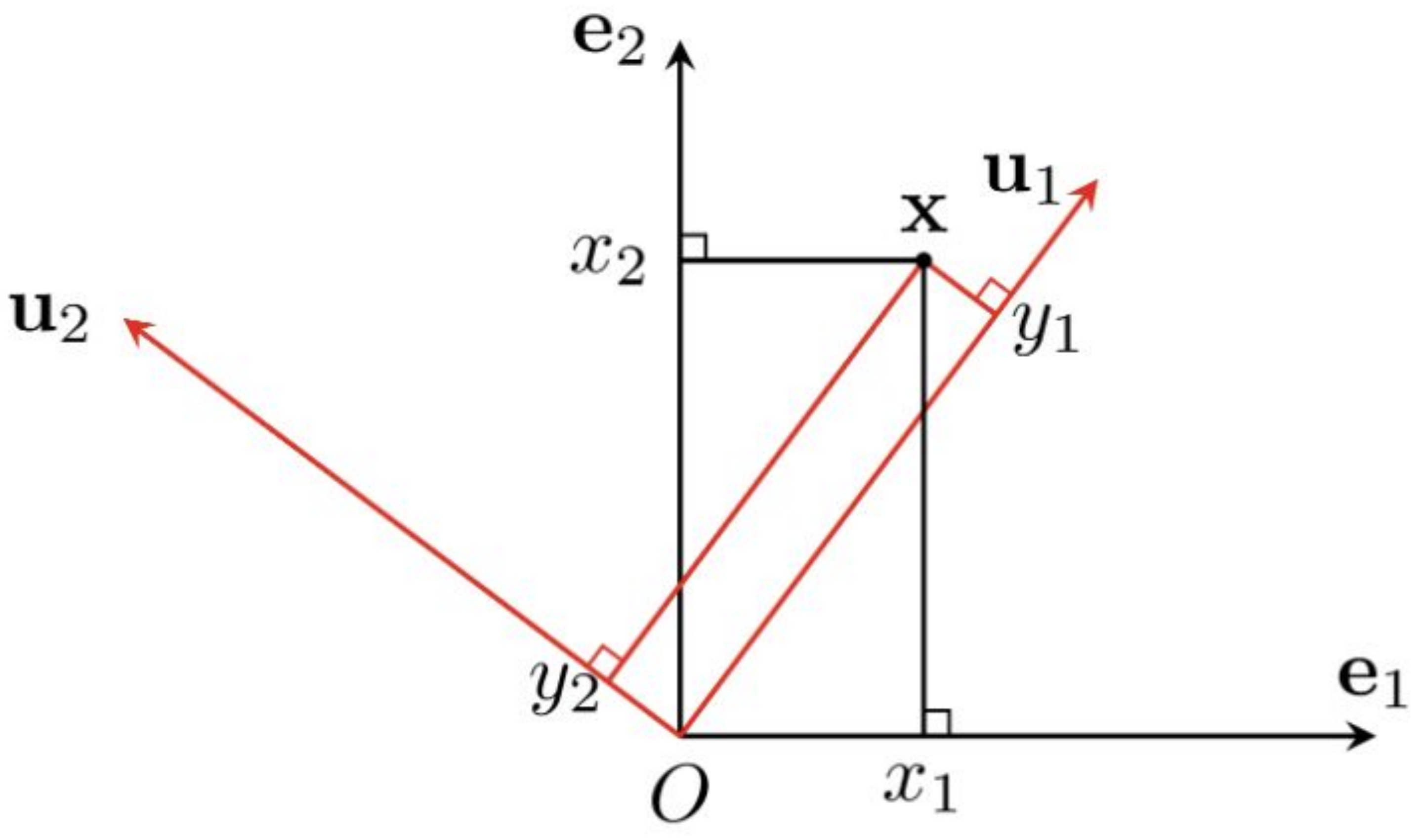
$$\mathbf{y} = \mathbf{U}^{-1}\mathbf{x} \quad (7)$$

với chú ý rằng $\mathbf{U}$ là ma trận khả nghịch vì các cột của nó độc lập tuyến tính.

1.Principal Component Analysis

1. Cơ sở toán

- Biểu diễn vector trong hệ các cơ sở khác nhau



Hình 1: Chuyển đổi toạ độ trong các hệ cơ sở khác nhau.

1.Principal Component Analysis

2. PCA

- Dữ liệu 1 chiều

Cho N giá trị $x_1, x_2, \ldots, x_N$. *Kỳ vọng* và *phương sai* của bộ dữ liệu này được định nghĩa là:

$$\bar{x} = \frac{1}{N}\sum_{n=1}^{N} x_n = \frac{1}{N}\mathbf{X1}$$

$$\sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x})^2$$

1.Principal Component Analysis

2. PCA

- Dữ liệu nhiều chiều (**D** chiều)

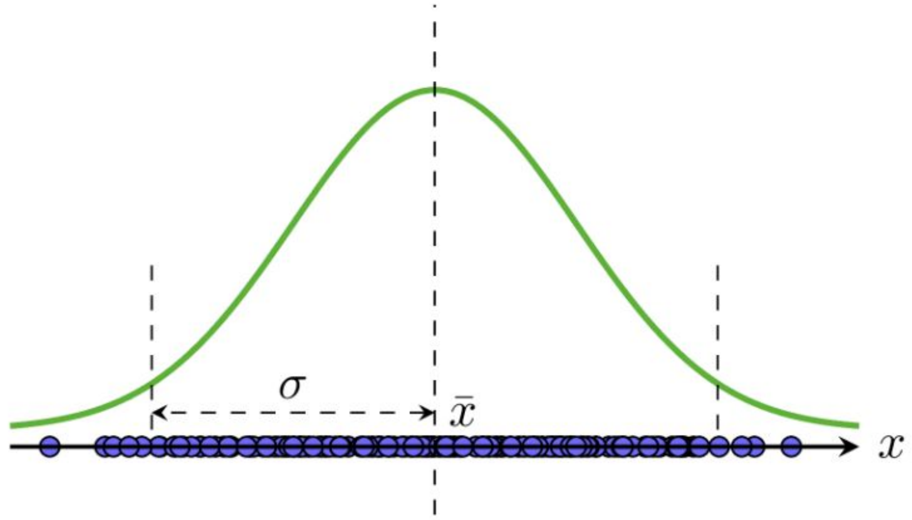
Cho N điểm dữ liệu được biểu diễn bởi các vector cột $\mathbf{x}_1, \dots, \mathbf{x}_N$, khi đó, *vector kỳ vọng* và *ma trận hiệp phương sai* của toàn bộ dữ liệu được định nghĩa là:

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n$$

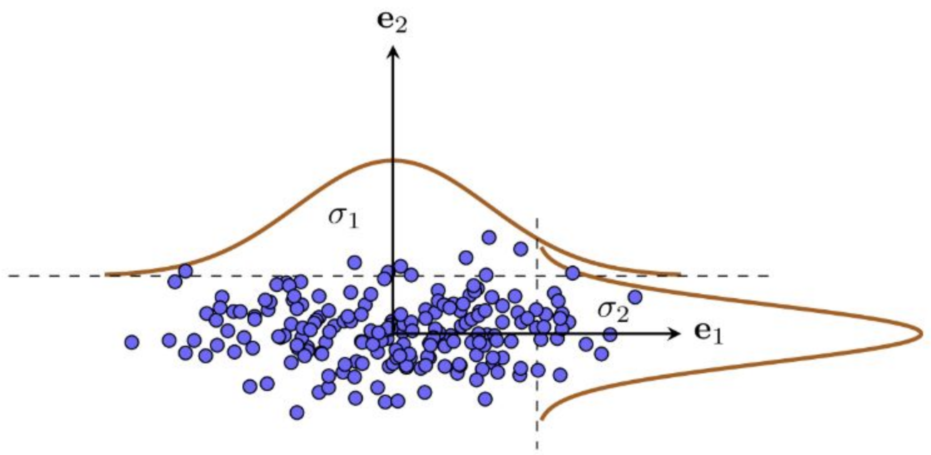
$$\mathbf{S} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^T = \frac{1}{N}\hat{\mathbf{X}}\hat{\mathbf{X}}^T$$

1.Principal Component Analysis

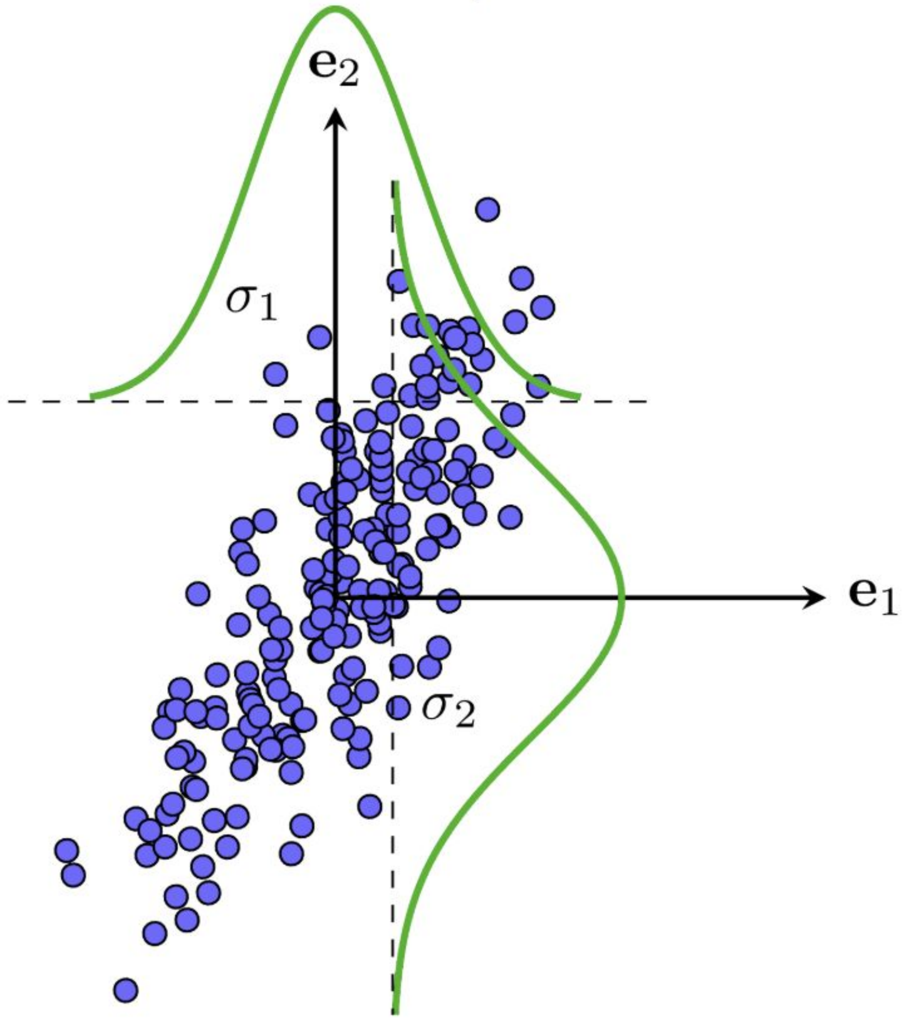
2. PCA



a)



b)

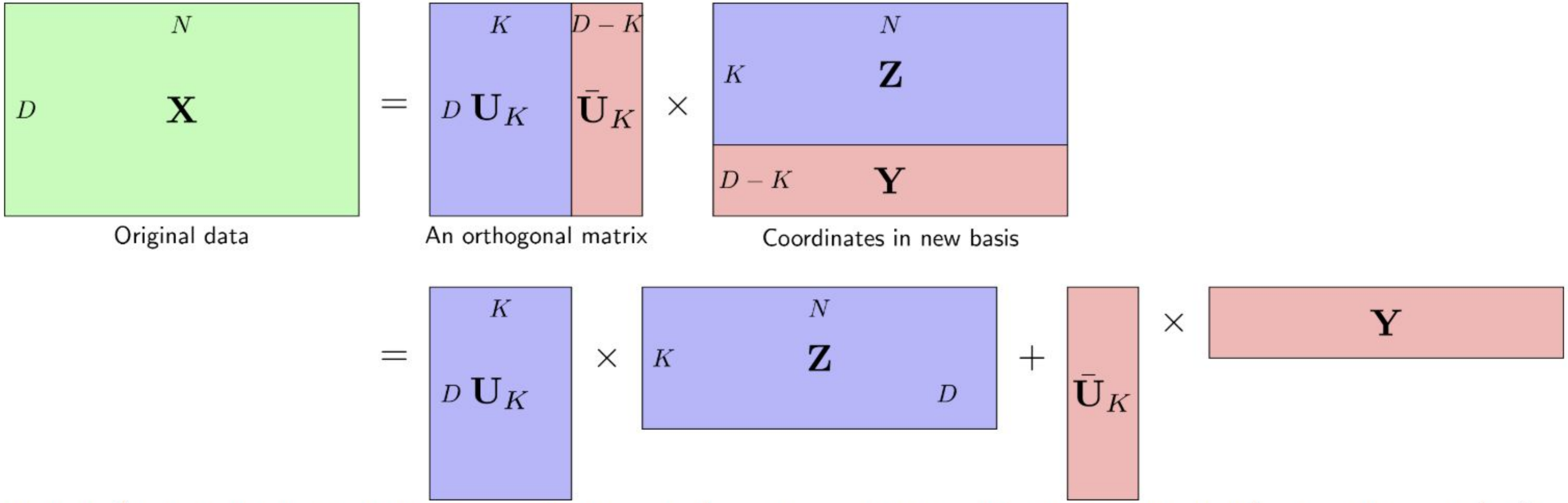


Hình 2: Ví dụ về kỳ vọng và phương sai. a) Trong không gian 1 chiều. b) Không gian 2 chiều mà hai chiều không tương quan. Trong trường hợp này, ma trận hiệp phương sai là ma trận đường chéo với hai phần tử trên đường chéo là σ_1, σ_2, đây cũng chính là hai trị riêng của ma trận hiệp phương sai và là phương sai của mỗi chiều dữ liệu. c) Dữ liệu trong không gian hai chiều có tương quan. Theo mỗi chiều, ta có thể tính được kỳ vọng và phương sai. Phương sai càng lớn thì dữ liệu trong chiều đó càng phân tán. Trong ví dụ này, dữ liệu theo chiều thứ hai phân tán nhiều hơn so so với chiều thứ nhất.

1.Principal Component Analysis

2. PCA

- Gọi **U** là hệ trực chuẩn mới và ta muốn giữ lại **K** tọa độ trong hệ này. Không mất tính tổng quát, giả sử đó là **K** tọa độ đầu.



Hình 3: Ý tưởng chính của PCA: Tìm một hệ trực chuẩn mới sao cho trong hệ này, các thành phần quan trọng nhất nằm trong K thành phần đầu tiên.

$$\mathbf{X} = \mathbf{U}_K\mathbf{Z} + \bar{\mathbf{U}}_K\mathbf{Y} \quad (8)$$

1.Principal Component Analysis

2. PCA

- Xấp xỉ **X**

$$\mathbf{X} \approx \tilde{\mathbf{X}} = \mathbf{U}_K\mathbf{Z} + \bar{\mathbf{U}}_K\bar{\mathbf{U}}_K^T\bar{\mathbf{x}}\mathbf{1}^T \quad (10)$$

1.Principal Component Analysis

2. PCA

- Hàm mất mát

$$J = \frac{1}{N}||\mathbf{X} - \tilde{\mathbf{X}}||_F^2 = \frac{1}{N}||\bar{\mathbf{U}}_K\bar{\mathbf{U}}_K^T\mathbf{X} - \bar{\mathbf{U}}_K\bar{\mathbf{U}}_K^T\bar{\mathbf{x}}\mathbf{1}^T||_F^2 \quad (11)$$

$$\begin{aligned}
J &= \frac{1}{N}||\bar{\mathbf{U}}_K^T(\mathbf{X} - \bar{\mathbf{x}}\mathbf{1})^T||_F^2 = \frac{1}{N}||\bar{\mathbf{U}}_K^T\hat{\mathbf{X}}||_F^2 \\
&= \frac{1}{N}||\hat{\mathbf{X}}^T\bar{\mathbf{U}}_K||_F^2 = \frac{1}{N}\sum_{i=K+1}^{D}||\hat{\mathbf{X}}^T\mathbf{u}_i||_2^2 \\
&= \frac{1}{N}\sum_{i=K+1}^{D}\mathbf{u}_i^T\hat{\mathbf{X}}\hat{\mathbf{X}}^T\mathbf{u}_i \\
&= \sum_{i=K+1}^{D}\mathbf{u}_i^T\mathbf{S}\mathbf{u}_i \quad (12)
\end{aligned}$$

=> **Uk** là ma trận với các cột tạo bởi K vector riêng của **S** (ứng với **K** trị riêng lớn nhất)

1.Principal Component Analysis

2. PCA

1. Tính vector kỳ vọng của toàn bộ dữ liệu:

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n$$

2. Trừ mỗi điểm dữ liệu đi vector kỳ vọng của toàn bộ dữ liệu:

$$\hat{\mathbf{x}}_n = \mathbf{x}_n - \bar{\mathbf{x}}$$

3. Tính ma trận hiệp phương sai:

$$\mathbf{S} = \frac{1}{N}\hat{\mathbf{X}}\hat{\mathbf{X}}^T$$

1.Principal Component Analysis

2. PCA

1. Tính vector kỳ vọng của toàn bộ dữ liệu:

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n$$

2. Trừ mỗi điểm dữ liệu đi vector kỳ vọng của toàn bộ dữ liệu:

$$\hat{\mathbf{x}}_n = \mathbf{x}_n - \bar{\mathbf{x}}$$

3. Tính ma trận hiệp phương sai:

$$\mathbf{S} = \frac{1}{N}\hat{\mathbf{X}}\hat{\mathbf{X}}^T$$

1.Principal Component Analysis

2. PCA

4. Tính các trị riêng và vector riêng có norm bằng 1 của ma trận này, sắp xếp chúng theo thứ tự giảm dần của trị riêng.
5. Chọn K vector riêng ứng với K trị riêng lớn nhất để xây dựng ma trận $\mathbf{U}_K$ có các cột tạo thành một hệ trực giao. K vectors này, còn được gọi là các thành phần chính, tạo thành một không gian con *gần* với phân bố của dữ liệu ban đầu đã chuẩn hoá.
6. Chiếu dữ liệu ban đầu đã chuẩn hoá $\hat{\mathbf{X}}$ xuống không gian con tìm được.
7. Dữ liệu mới chính là toạ độ của các điểm dữ liệu trên không gian mới.

$$\mathbf{Z} = \mathbf{U}_K^T\hat{\mathbf{X}}$$

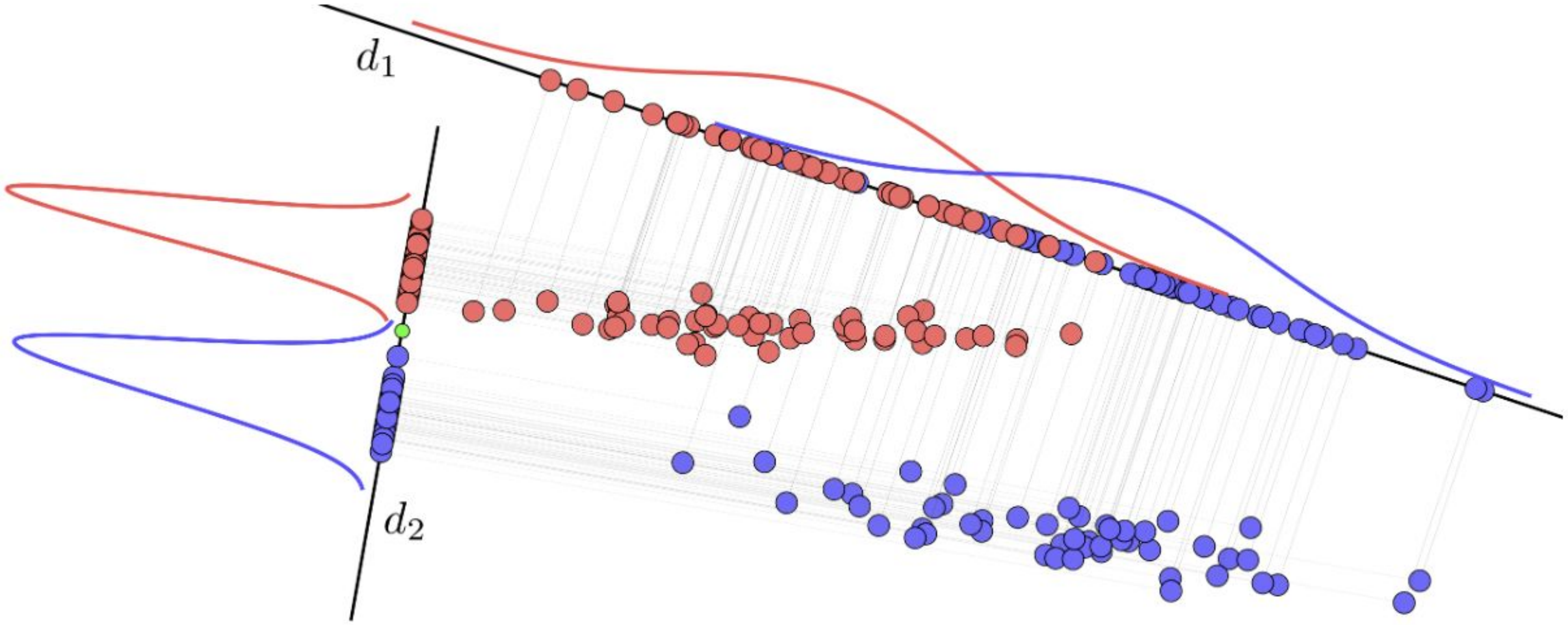
Dữ liệu ban đầu có thể tính được xấp xỉ theo dữ liệu mới như sau:

$$\mathbf{x} \approx \mathbf{U}_K\mathbf{Z} + \bar{\mathbf{x}}$$

IV. Linear Discriminant Analysis

IV. Linear Discriminant Analysis

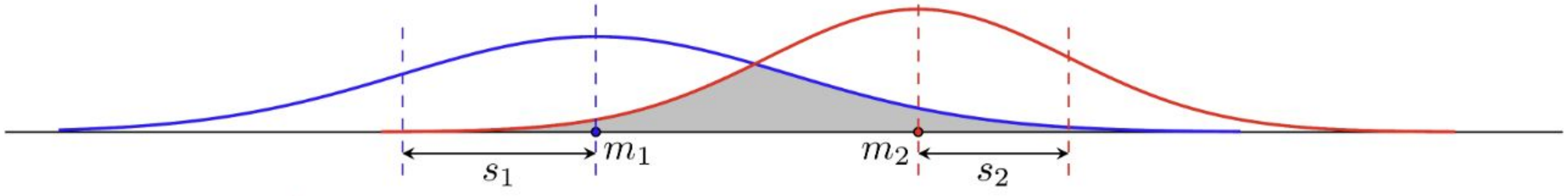
1. Động lực



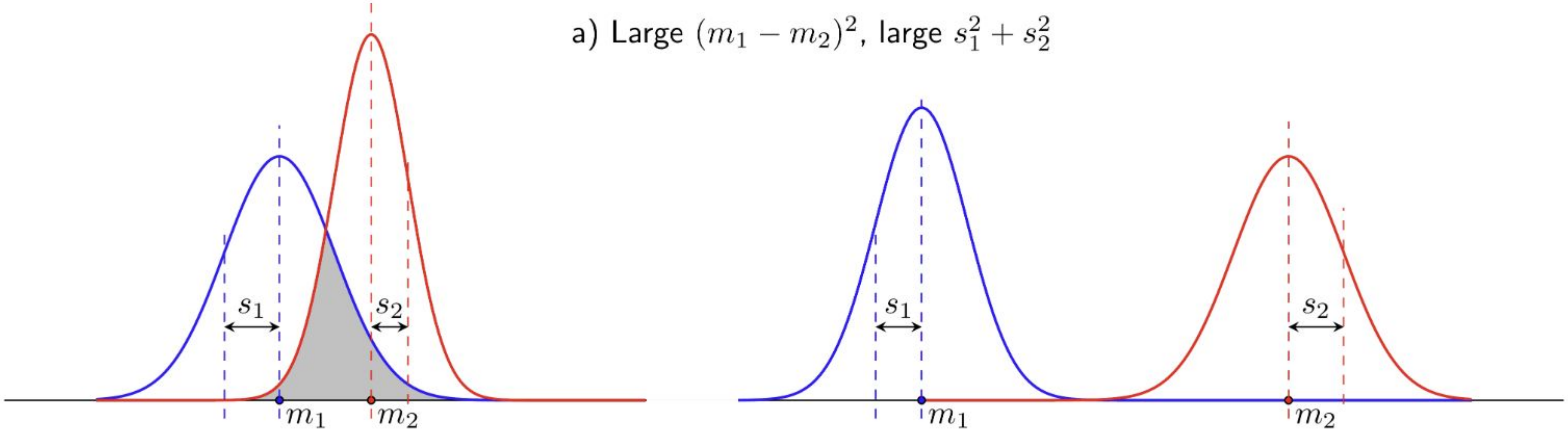
=> Không phải giữ nhiều thông tin nhất luôn là tốt nhất

IV. Linear Discriminant Analysis

2. LDA



a) Large $(m_1 - m_2)^2$, large $s_1^2 + s_2^2$



IV. Linear Discriminant Analysis

2. LDA

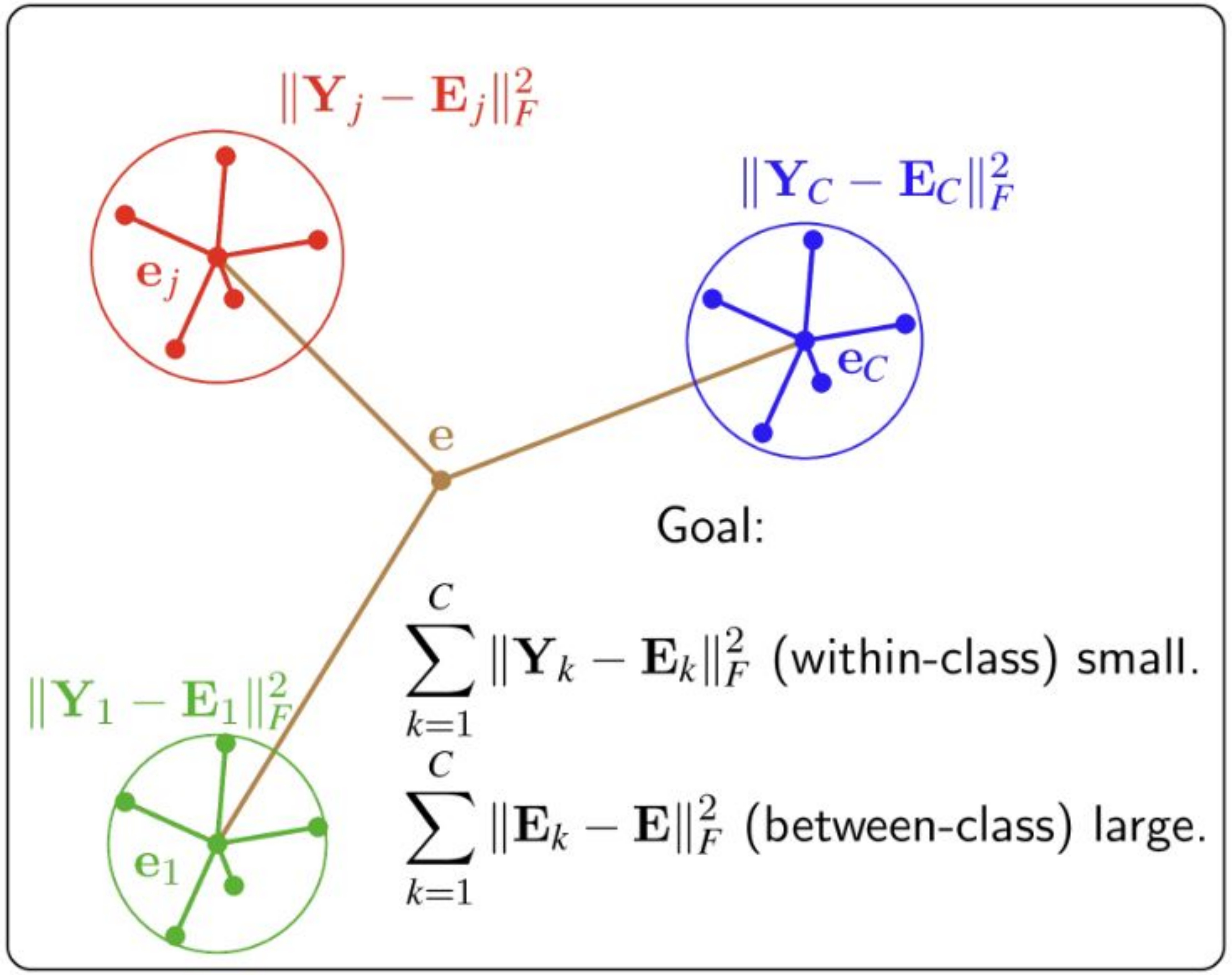
- Hàm mục tiêu (maximize)

$$J(\mathbf{w}) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2} \qquad (4)$$

IV. Linear Discriminant Analysis

2. LDA

Multi-class



THANK YOU

CONTACT US

403.1 H6, BKHCM Campus 2

mlandiotlab@gmail.com

ml-iotlab.com

facebook.com/hcmut.ml.iot.lab

youtube.com/@mliotlab