

# HUST

**ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.



**ĐẠI HỌC**  
**BÁCH KHOA HÀ NỘI**  
HANOI UNIVERSITY  
OF SCIENCE AND TECHNOLOGY

# Một số thuật toán giảm chiều dữ liệu trong học máy

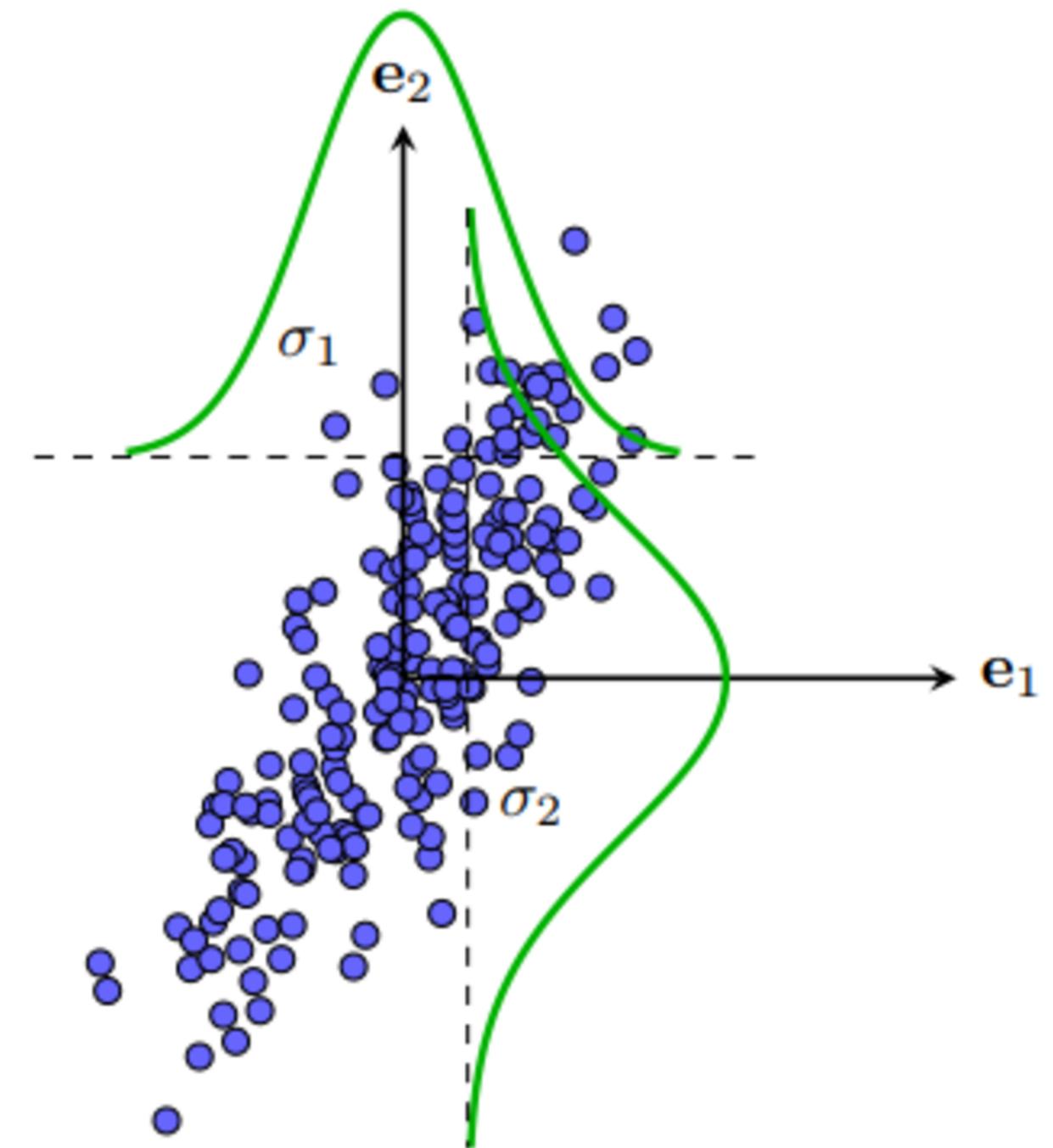
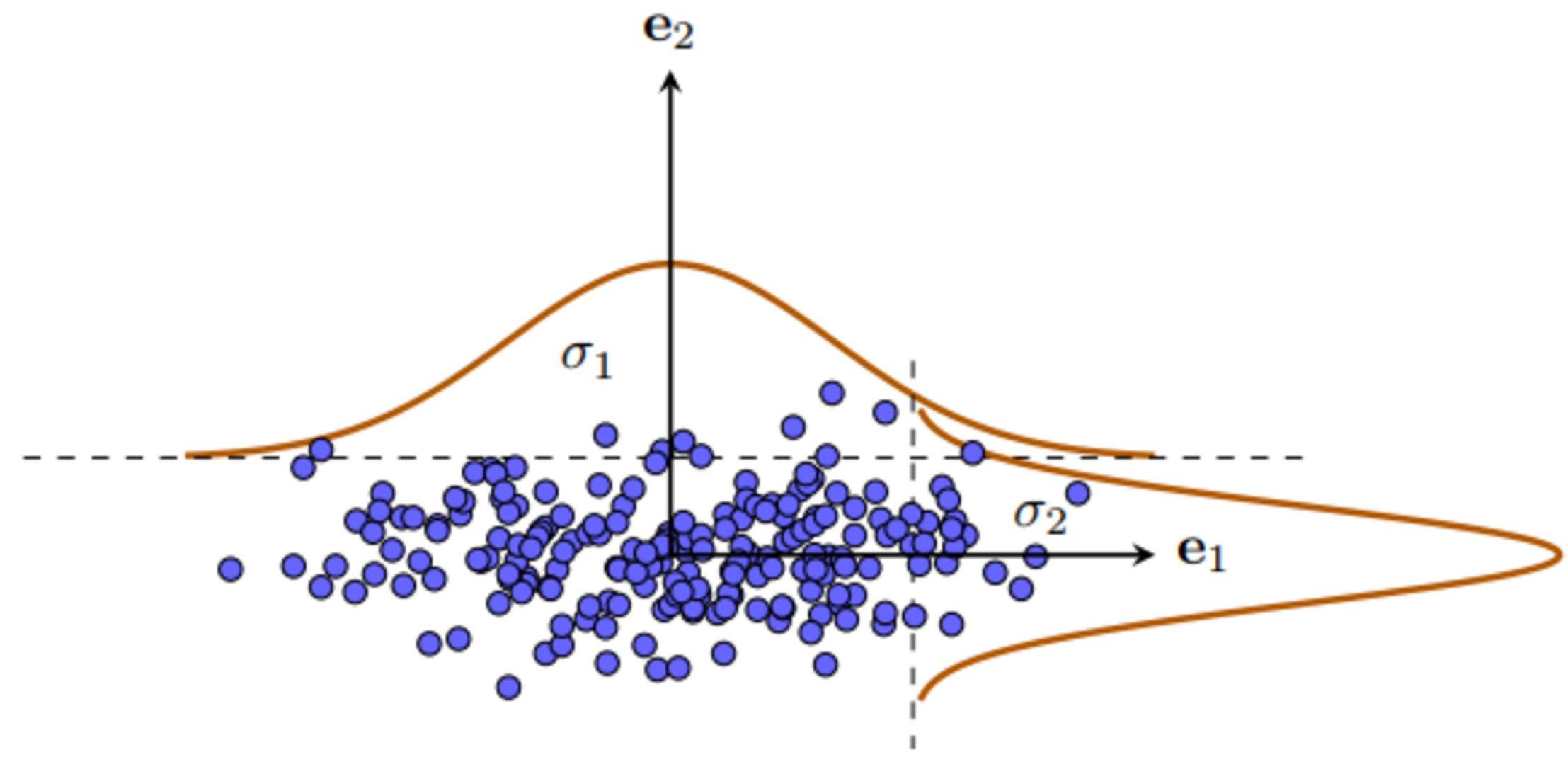
ONE LOVE. ONE FUTURE.

# Nội dung chính

- Các thuật toán giảm chiều dữ liệu:
  - PCA
  - t-SNE
  - LDA
- Ứng dụng trong bài toán thực tế



# PCA



$$\begin{matrix} N \\ D \end{matrix} \quad \mathbf{X} = \begin{matrix} K \\ D \mathbf{U}_K \end{matrix} \quad \begin{matrix} D - K \\ \widehat{\mathbf{U}}_K \end{matrix} \times \begin{matrix} N \\ K \\ D - K \end{matrix} \quad \mathbf{Z} \\ \mathbf{Y} \end{matrix}$$

$$= \begin{matrix} K \\ D \mathbf{U}_K \end{matrix} \times \begin{matrix} N \\ K \\ D \end{matrix} \quad \mathbf{Z} + \begin{matrix} \widehat{\mathbf{U}}_K \end{matrix} \times \begin{matrix} \mathbf{Y} \end{matrix}$$

$$(\mathbf{b}\mathbf{1}^T - \widehat{\mathbf{U}}_K^T \mathbf{X})\mathbf{1} = 0 \Rightarrow N\mathbf{b} = \widehat{\mathbf{U}}_K^T \mathbf{X}\mathbf{1} \Rightarrow \mathbf{b} = \widehat{\mathbf{U}}_K^T \bar{\mathbf{x}}$$

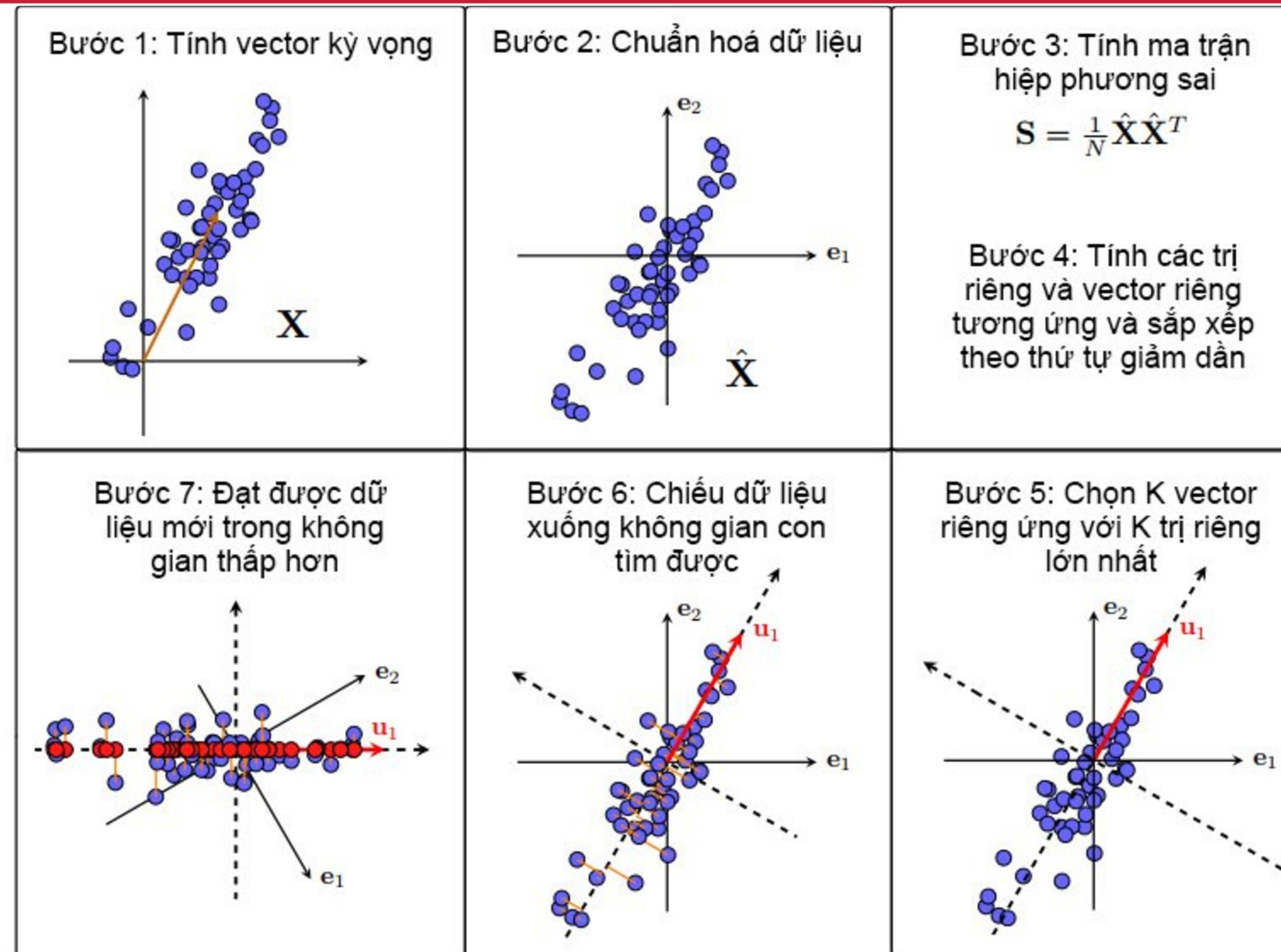
$$\mathbf{X} = \mathbf{U}_K \mathbf{Z} + \widehat{\mathbf{U}}_k \mathbf{Y} \approx \mathbf{U}_K \mathbf{Z} + \widehat{\mathbf{U}}_k \mathbf{b} \mathbf{1}^T = \mathbf{U}_K \mathbf{Z} + \widehat{\mathbf{U}}_K \widehat{\mathbf{U}}_K^T \bar{\mathbf{x}} \mathbf{1}^T \triangleq \tilde{\mathbf{X}}$$

- Hàm mất mát: 
$$\begin{aligned} J &= \frac{1}{N} \|\mathbf{X} - \tilde{\mathbf{X}}\|_F^2 = \frac{1}{N} \|\widehat{\mathbf{U}}_k \widehat{\mathbf{U}}_k^T (\mathbf{X} - \bar{\mathbf{x}}\mathbf{1}^T)\|_F^2 \\ &= \frac{1}{N} \|\widehat{\mathbf{U}}_K^T \widehat{\mathbf{X}}\|_F^2 = \frac{1}{N} \|\widehat{\mathbf{X}}^T \widehat{\mathbf{U}}_K\|_F^2 = \frac{1}{N} \sum_{i=K+1}^D \|\widehat{\mathbf{X}}^T \mathbf{u}_i\|_2^2 \\ &= \frac{1}{N} \sum_{i=K+1}^D \mathbf{u}_i^T \widehat{\mathbf{X}} \widehat{\mathbf{X}}^T \mathbf{u}_i = \sum_{i=K+1}^D \mathbf{u}_i^T \mathbf{S} \mathbf{u}_i \end{aligned}$$

- Với ma trận  $\mathbf{U}$  trực giao bất kỳ ta có:

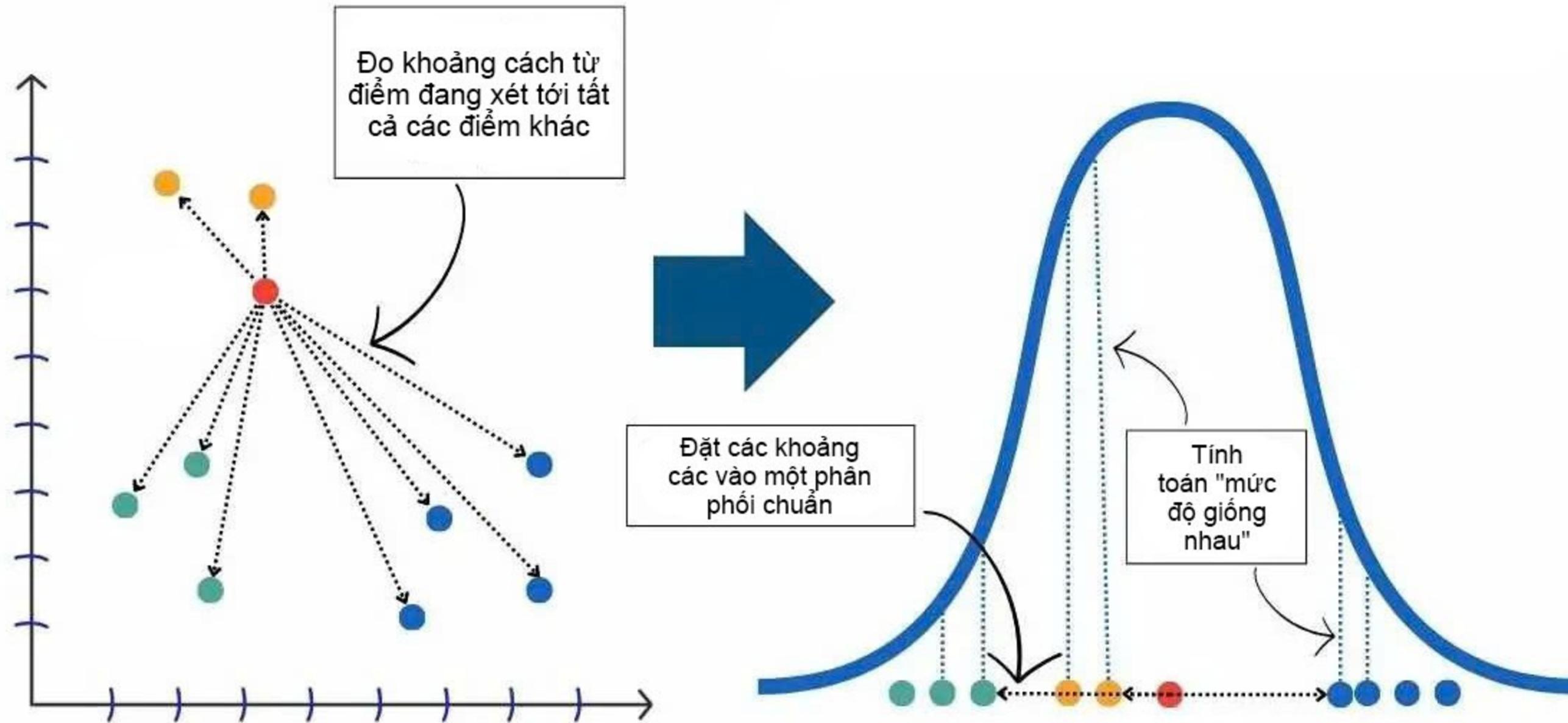
$$\begin{aligned} L &= \sum_{i=1}^D \mathbf{u}_i^T \mathbf{S} \mathbf{u}_i = \frac{1}{N} \|\widehat{\mathbf{X}}^T \mathbf{U}\|_F^2 = \frac{1}{N} \text{trace}(\widehat{\mathbf{X}}^T \mathbf{U} \mathbf{U}^T \widehat{\mathbf{X}}) \\ &= \frac{1}{N} \text{trace}(\widehat{\mathbf{X}}^T \widehat{\mathbf{X}}) = \frac{1}{N} \text{trace}(\widehat{\mathbf{X}} \widehat{\mathbf{X}}^T) = \text{trace}(\mathbf{S}) = \sum_{i=1}^D \lambda_i \end{aligned}$$

# PCA



- PCA được sử dụng để giảm chiều dữ liệu, giảm dung lượng lưu trữ, tiền xử lý cho các mô hình học máy,...
- Ưu điểm của PCA
  - Loại bỏ nhiễu, giảm overfit
  - Trực quan hóa dữ liệu
- Nhược điểm của PCA:
  - Khó giải thích
  - Không hoạt động tốt với quan hệ phi tuyến
  - Không quan tâm đến nhãn dữ liệu

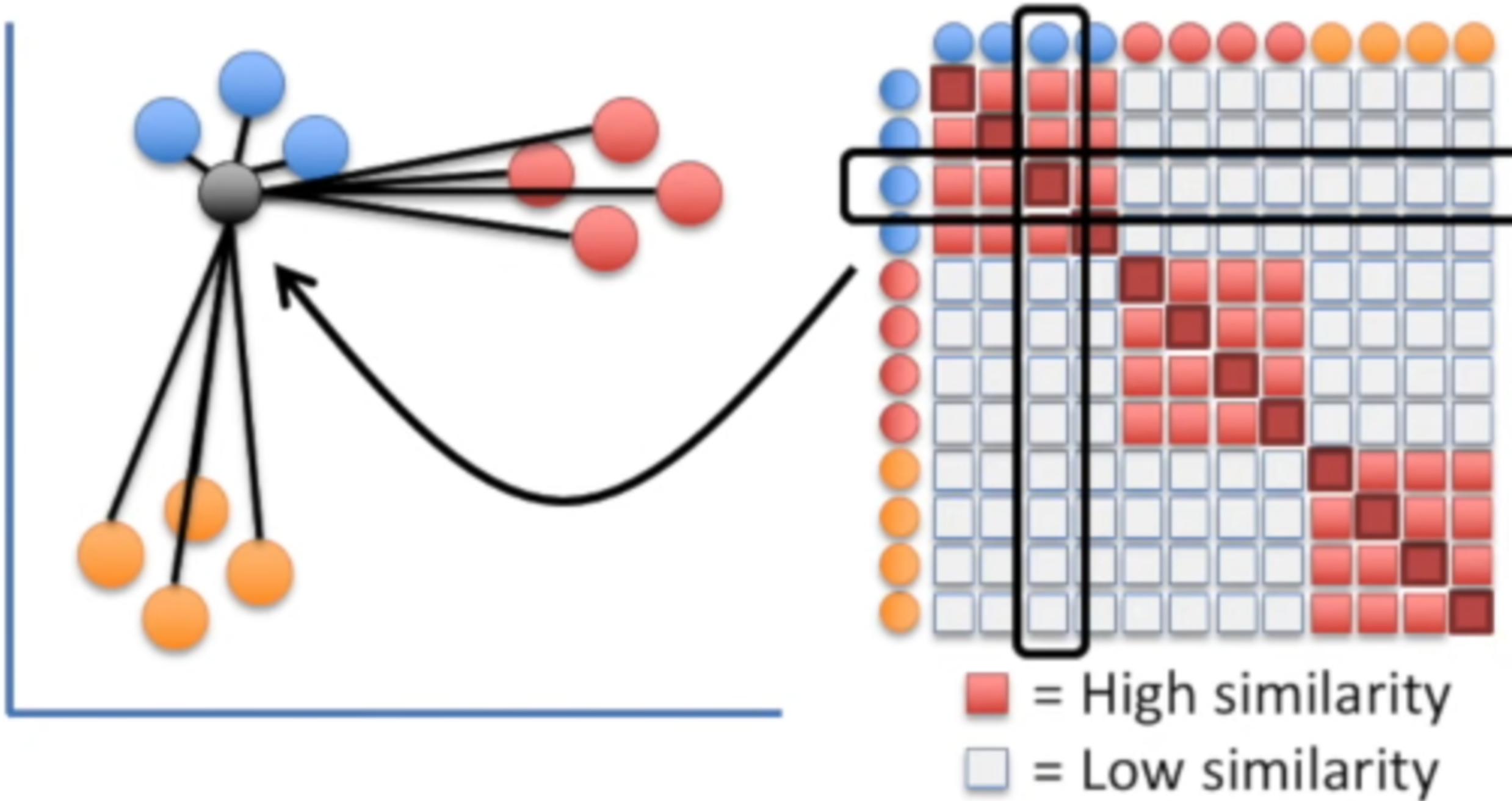
# t-SNE



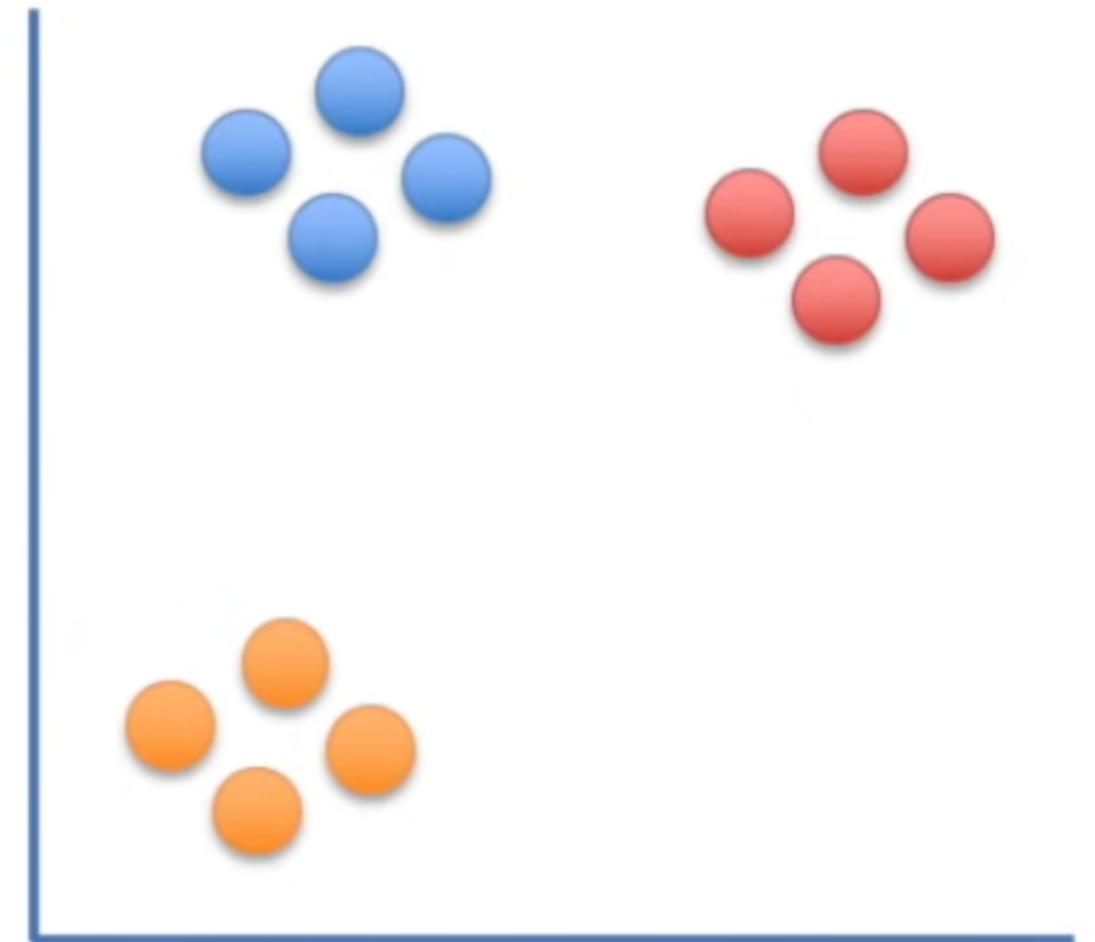
$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / 2\sigma_i^2)}$$

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$$

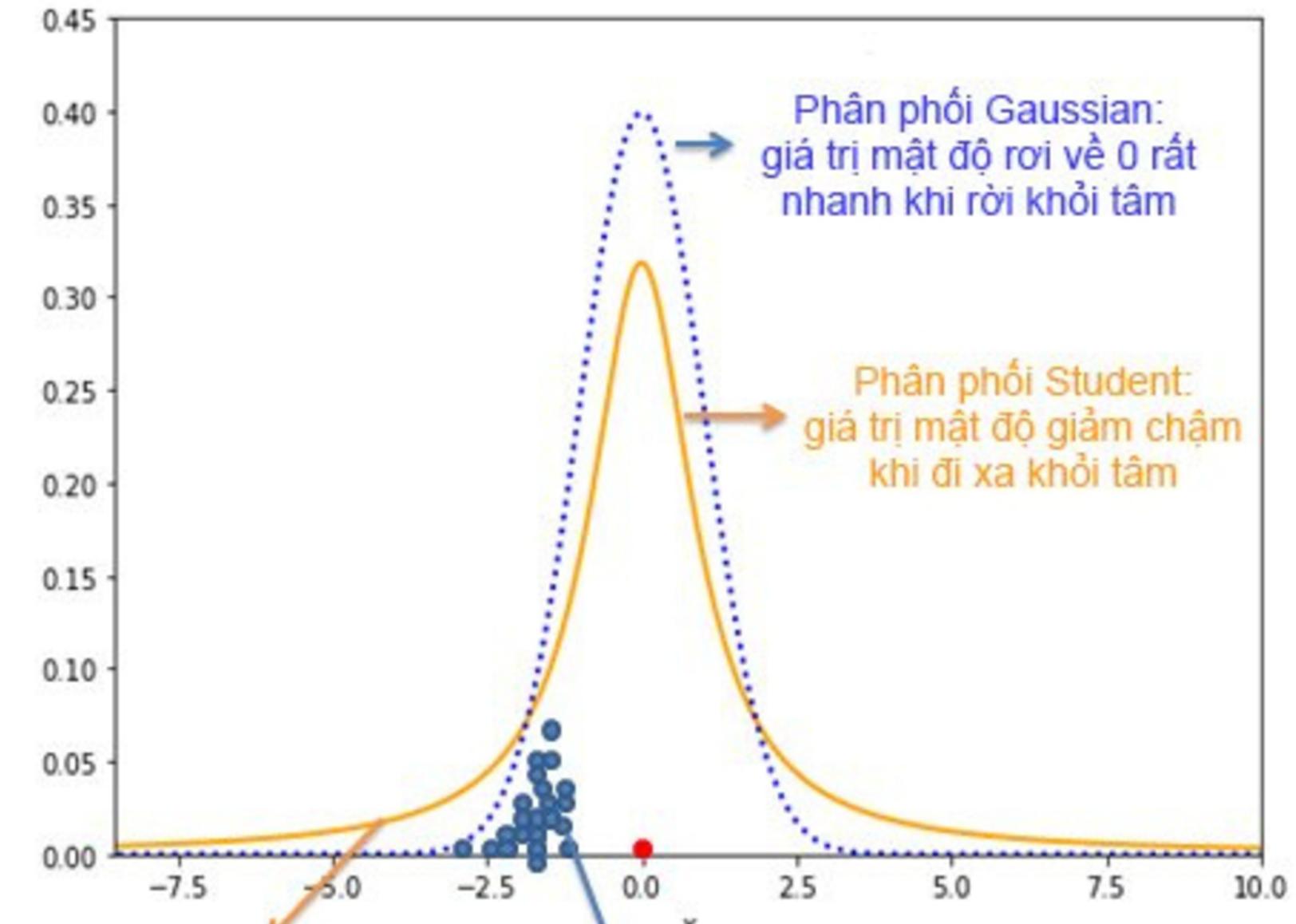
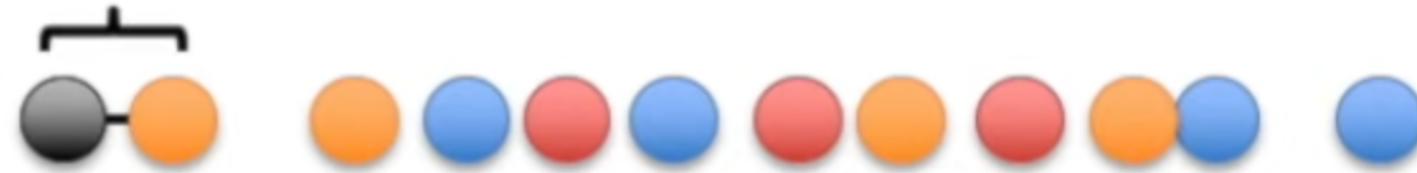
# t-SNE



# t-SNE



$$q_{ij} = \frac{(1 + \| \mathbf{y}_i - \mathbf{y}_j \|^2)^{-1}}{\sum_k \sum_{l \neq k} (1 + \| \mathbf{y}_k - \mathbf{y}_l \|^2)^{-1}}$$



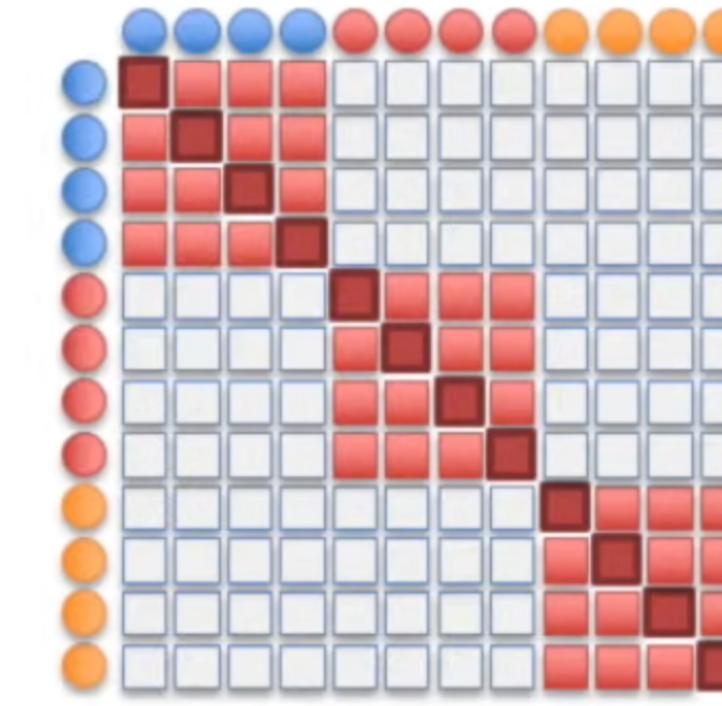
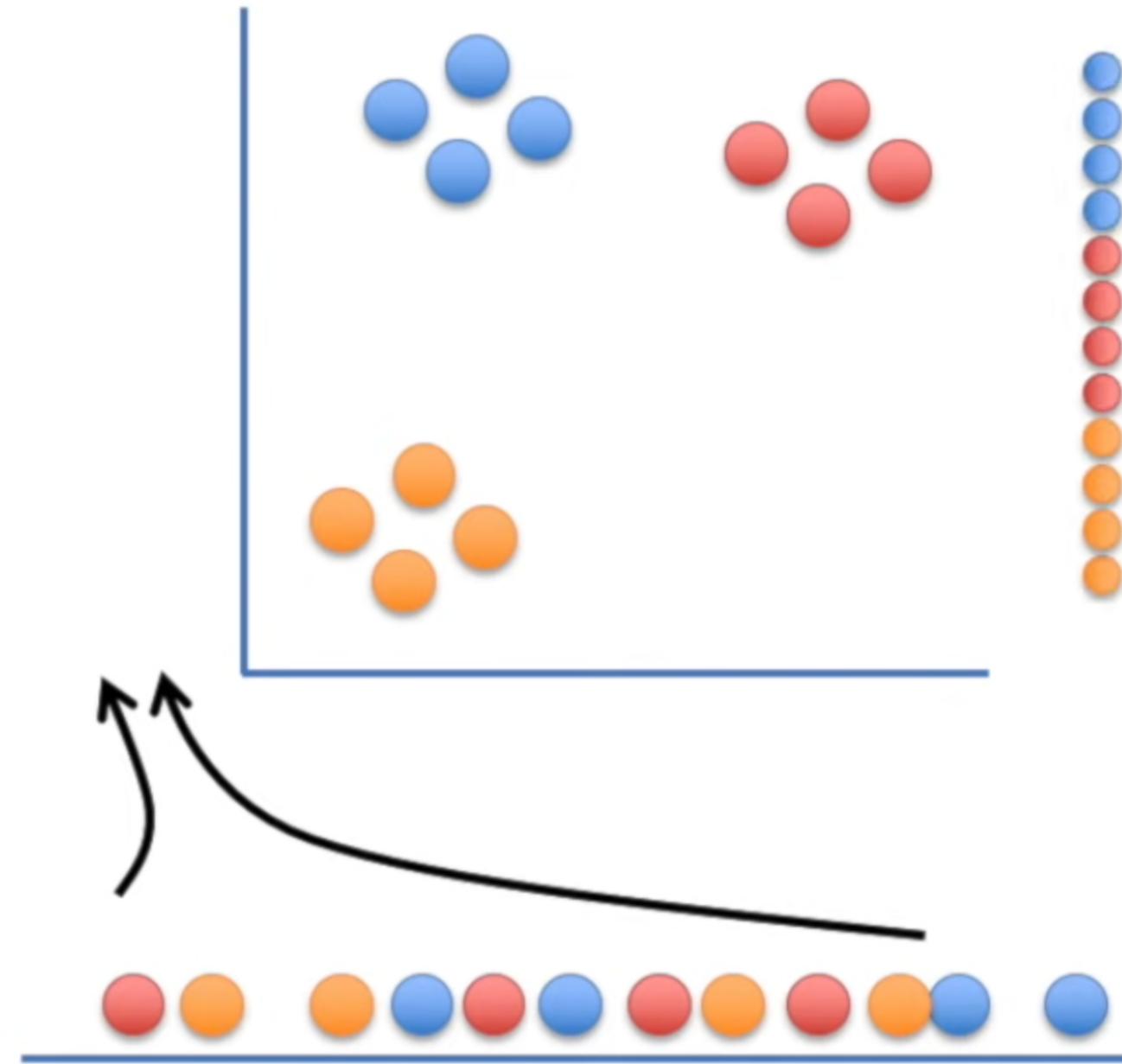
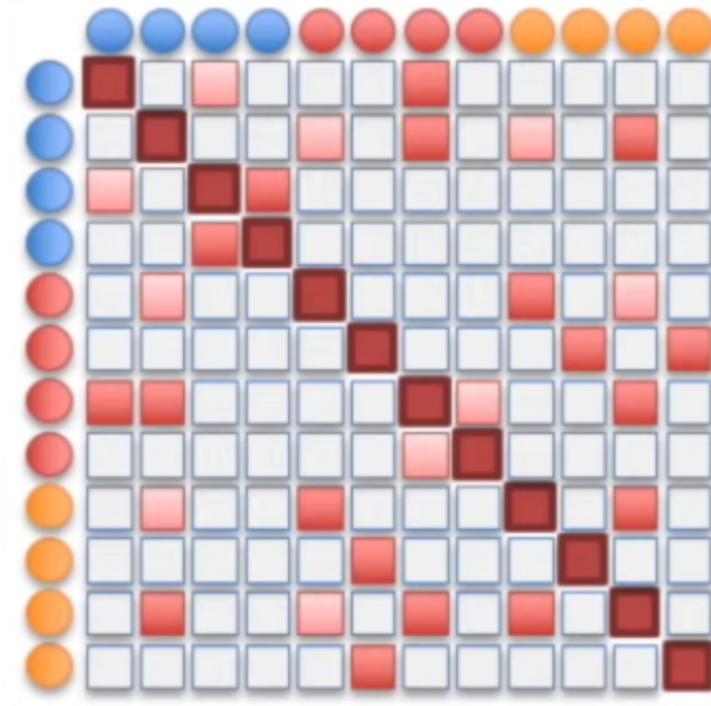
t-Student giúp cấu  
trúc lân cận được  
bảo toàn tốt hơn

Phân phối Gaussian:  
giá trị mật độ rơi về 0 rất  
nhanh khi rời khỏi tâm

Phân phối Student:  
giá trị mật độ giảm chậm  
khi đi xa khỏi tâm

Gaussian hầu như “không dành  
chỗ” cho các điểm ở khoảng  
cách từ trung bình đến xa

# t-SNE



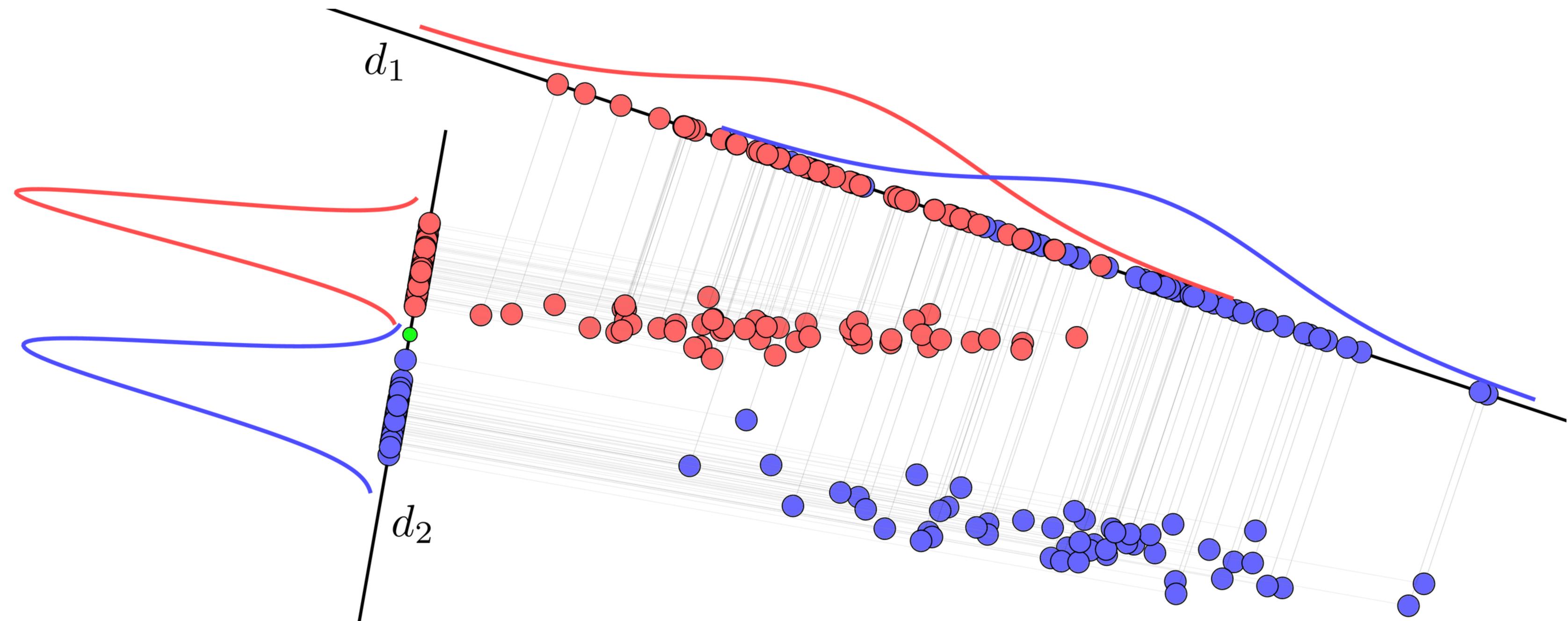
■ = High similarity  
□ = Low similarity

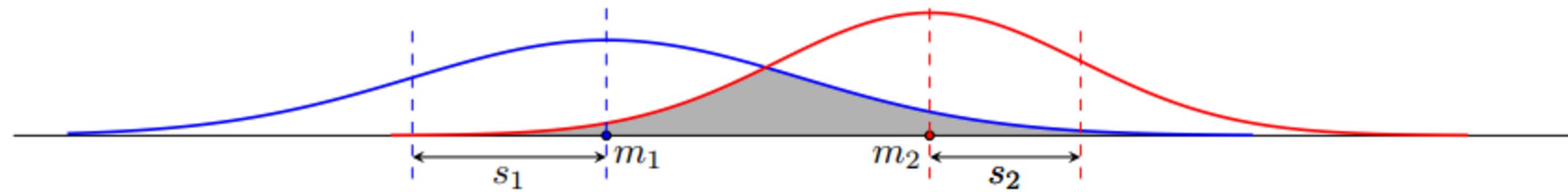
$$y_i^{(k+1)} = y_i^{(k)} + hD_i^{(k)}, \quad \text{for } i = 1, \dots, n.$$

$$D_i^{(k)} = 4 \sum_{1 \leq j \leq n, j \neq i} (y_j^{(k)} - y_i^{(k)}) S_{ij}^{(k)}$$

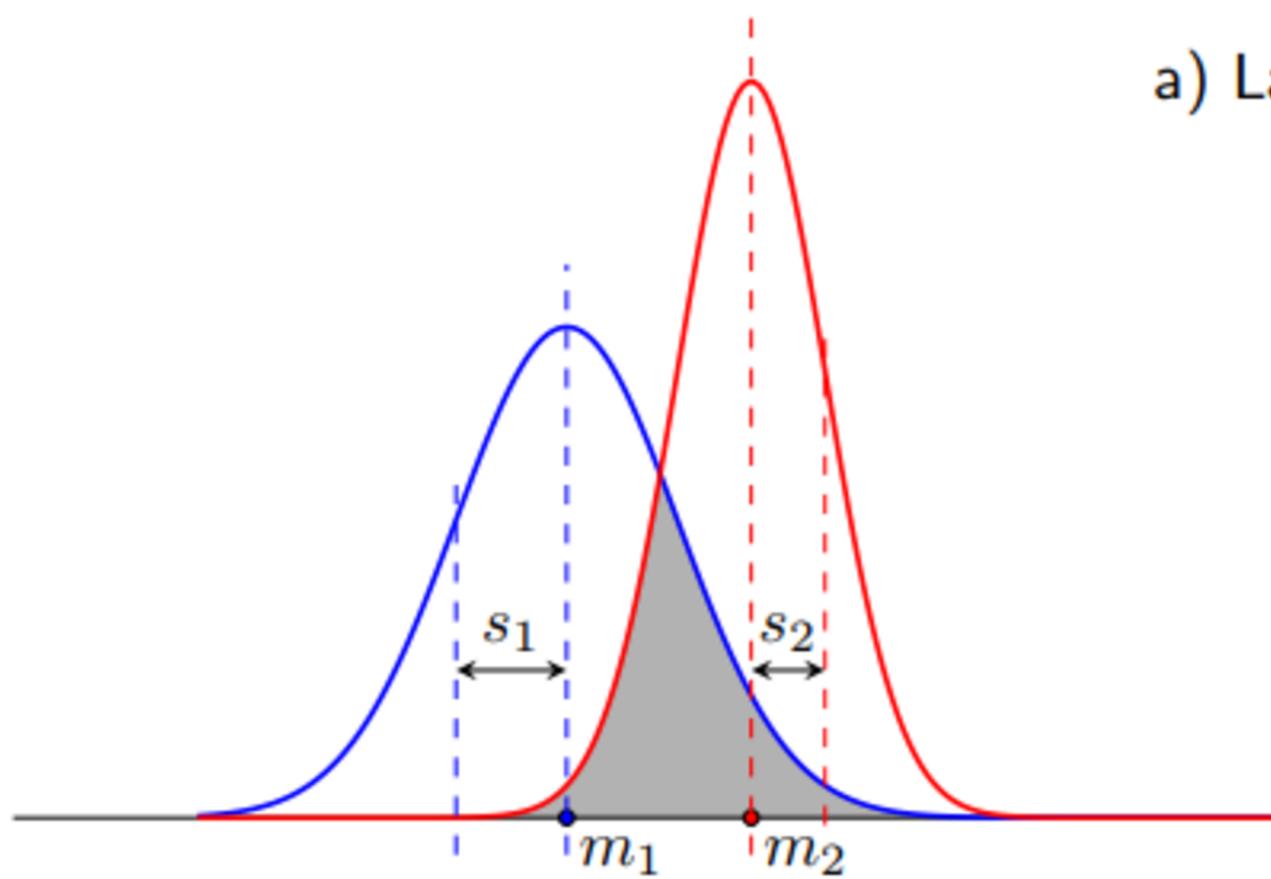
$$S_{ij}^{(k)} = (p_{ij} - q_{ij}^{(k)}) / (1 + \|y_i^{(k)} - y_j^{(k)}\|_2^2)$$

- t-SNE chủ yếu phục vụ cho khám phá dữ liệu và trực quan hóa, giúp nhận biết các cụm, biên giới các lớp hay điểm bất thường
- **Ưu điểm của t-SNE:**
  - Giữ tốt cấu trúc cục bộ
  - Trực quan hóa tốt, hữu ích trong phân tích dữ liệu
- **Nhược điểm của t-SNE:**
  - Tốn kém thời gian tính toán khi dữ liệu lớn
  - Không bảo toàn cấu trúc toàn cục
  - Nhạy với tham số đầu vào, kết quả không ổn định
  - Chỉ để trực quan, không để suy luận định lượng

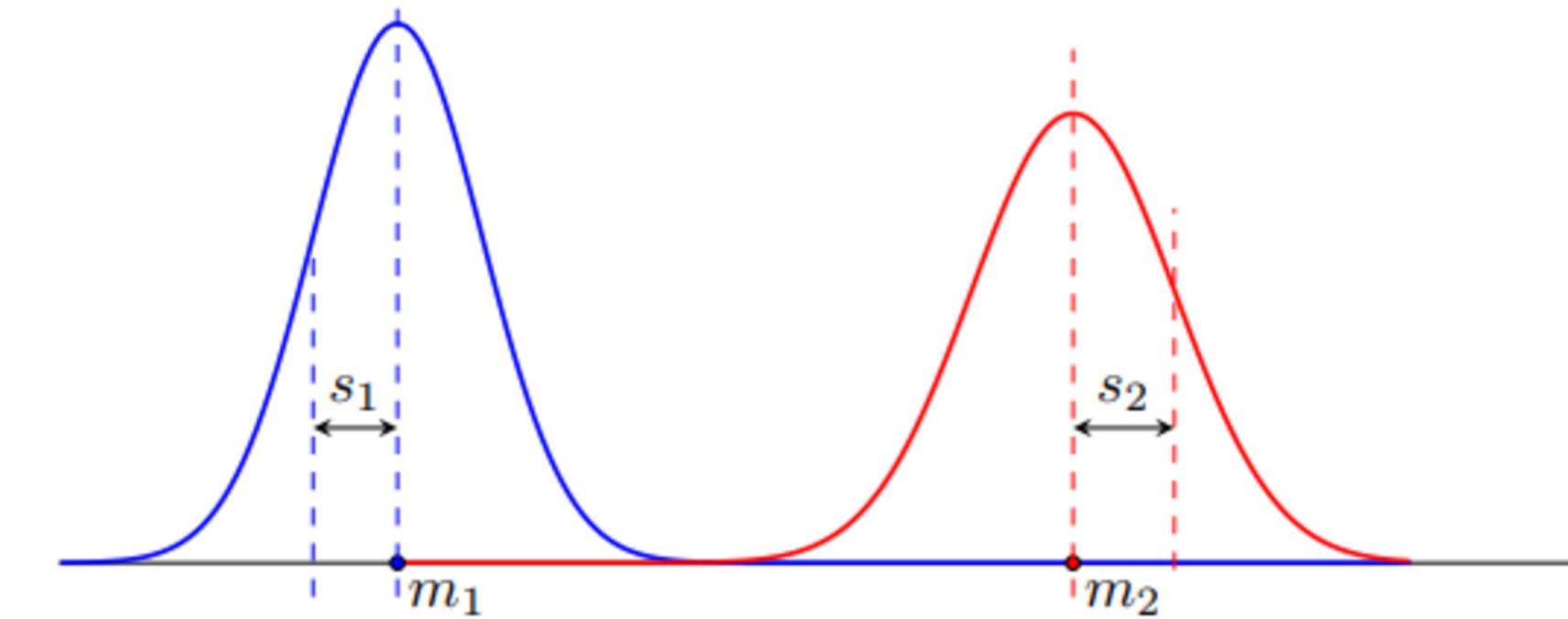




a) Large  $(m_1 - m_2)^2$ , large  $s_1^2 + s_2^2$



b) Small  $(m_1 - m_2)^2$ , small  $s_1^2 + s_2^2$



c) Large  $(m_1 - m_2)^2$ , small  $s_1^2 + s_2^2$

- Một vài ký hiệu

- Ma trận dữ liệu lớp thứ k ở không gian ban đầu và không gian sau:

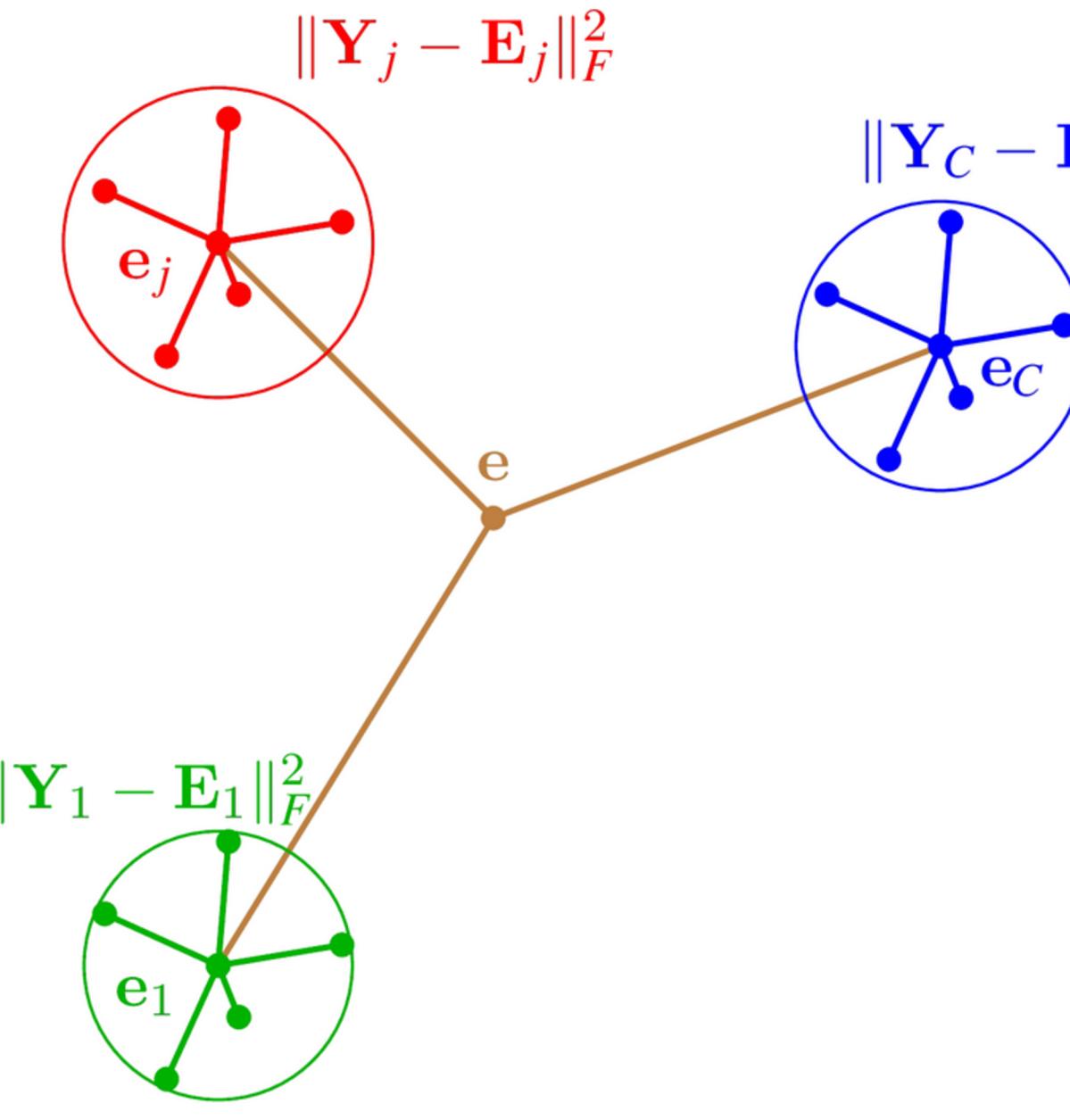
$$\mathbf{X}_k, \mathbf{Z}_k = \mathbf{W}^T \mathbf{X}_k$$

- Vector kỳ vọng của lớp thứ k trong không gian ban đầu:

$$\mathbf{m}_k = \frac{1}{N_k} \sum_{n \in \mathcal{C}_k} \mathbf{x}_k$$

- Vector kỳ vọng của lớp thứ k trong không gian mới:

$$\mathbf{e}_k = \frac{1}{N_k} \sum_{n \in \mathcal{C}_k} \mathbf{z}_n = \mathbf{W}^T \mathbf{m}_k$$



$$\begin{aligned}
 \sigma_k^2 &= \sum_{n \in \mathcal{C}_k} \|\mathbf{z}_n - \mathbf{e}_k\|_F^2 = \|\mathbf{Z}_k - \mathbf{E}_k\|_2^2 = \|\mathbf{W}^T(\mathbf{X}_k - \mathbf{M}_k)\|_F^2 \\
 &= \text{trace} (\mathbf{W}^T(\mathbf{X}_k - \mathbf{M}_k)(\mathbf{X}_k - \mathbf{M}_k)^T \mathbf{W}) \\
 s_W &= \sum_{k=1}^C \sigma_k^2 = \sum_{k=1}^C \text{trace} (\mathbf{W}^T(\mathbf{X}_k - \mathbf{M}_k)(\mathbf{X}_k - \mathbf{M}_k)^T \mathbf{W}) = \text{trace} (\mathbf{W}^T \mathbf{S}_W \mathbf{W}) \\
 \mathbf{S}_W &= \sum_{k=1}^C \|\mathbf{X}_k - \mathbf{M}_k\|_F^2 = \sum_{k=1}^C \sum_{n \in \mathcal{C}_k} (\mathbf{x}_n - \mathbf{m}_k)(\mathbf{x}_n - \mathbf{m}_k)^T \\
 s_B &= \sum_{k=1}^C N_k \|\mathbf{e}_k - \mathbf{e}\|_F^2 = \sum_{k=1}^C \|\mathbf{E}_k - \mathbf{E}\|_F^2 = \text{trace} (\mathbf{W}^T \mathbf{S}_B \mathbf{W}) \\
 \mathbf{S}_B &= \sum_{k=1}^C (\mathbf{M}_k - \mathbf{M})(\mathbf{M}_k - \mathbf{M})^T = \sum_{k=1}^C N_k (\mathbf{m}_k - \mathbf{m})(\mathbf{m}_k - \mathbf{m})^T
 \end{aligned}$$

- Bài toán tối ưu

$$\mathbf{W} = \arg \max_{\mathbf{W}} J(\mathbf{W}) = \arg \max_{\mathbf{W}} \frac{\text{trace}(\mathbf{W}^T \mathbf{S}_B \mathbf{W})}{\text{trace}(\mathbf{W}^T \mathbf{S}_W \mathbf{W})}$$

$$\nabla_{\mathbf{W}} J(\mathbf{W}) = \frac{2 \left( \mathbf{S}_B \mathbf{W} \text{trace}(\mathbf{W}^T \mathbf{S}_W \mathbf{W}) - \text{trace}(\mathbf{W}^T \mathbf{S}_B \mathbf{W}) \mathbf{S}_W \mathbf{W} \right)}{(\text{trace}(\mathbf{W}^T \mathbf{S}_W \mathbf{W}))^2} = 0$$

$$\Leftrightarrow \mathbf{S}_W^{-1} \mathbf{S}_B \mathbf{W} = J(\mathbf{W}) \mathbf{W}$$

- LDA thường được dùng khi dữ liệu có nhãn lớp và mục tiêu chính là phân loại hoặc trực quan hóa dữ liệu
- Ưu điểm của LDA:
  - Giảm chiều có giám sát, tận dụng thông tin nhãn
  - Giảm nhiễu, tăng hiệu quả cho các mô hình phân loại tuyến tính
- Nhược điểm của LDA:
  - Không phù hợp cho dữ liệu phi tuyến
  - Số chiều mới không thể vượt quá số nhãn

- Trustworthiness

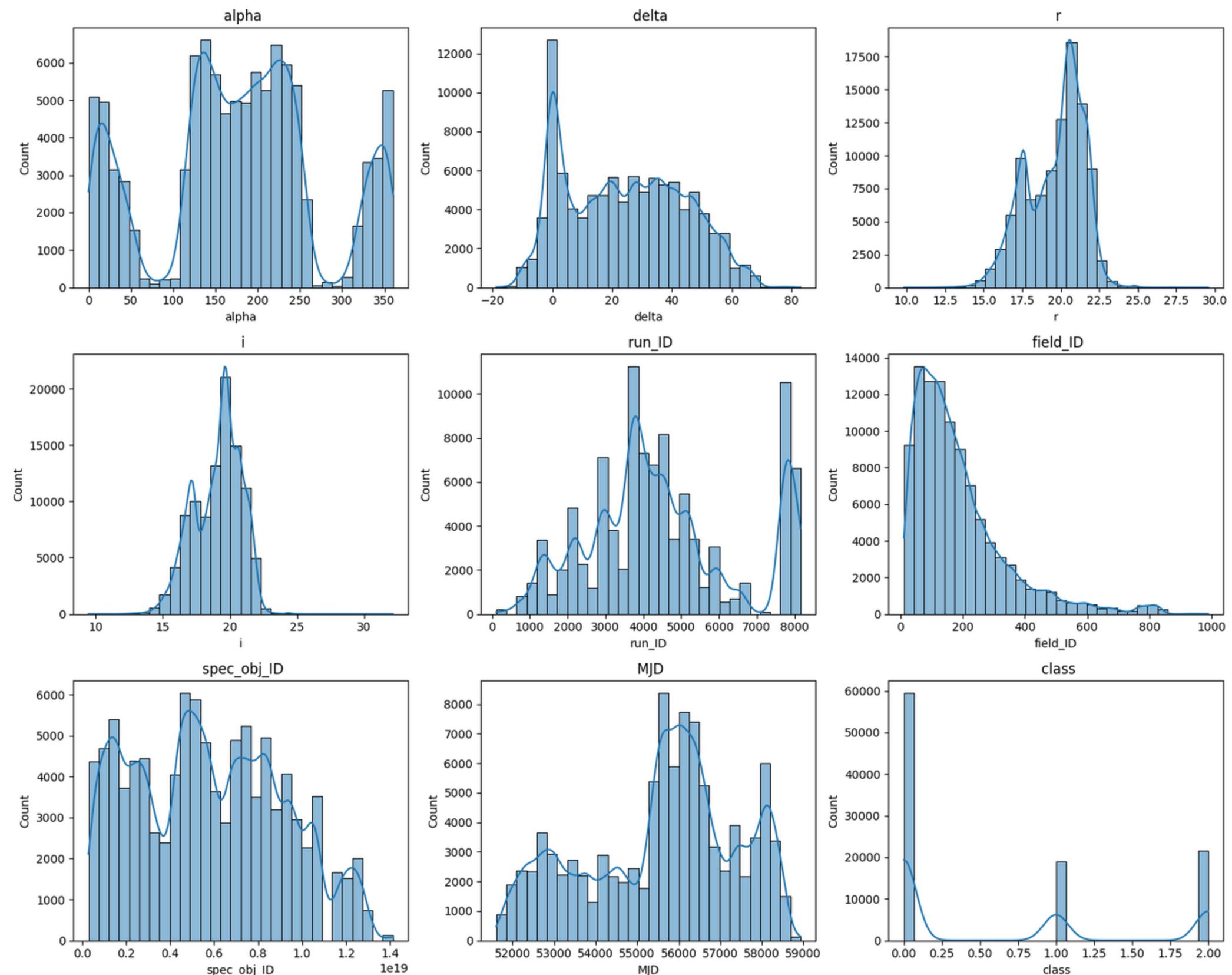
$$TW(k) = 1 - \frac{2}{nk(2n - 3k - 1)} \sum_{i=1}^n \sum_{j \in U_k(i)} (r(i, j) - k)$$

- Neighborhood Hit

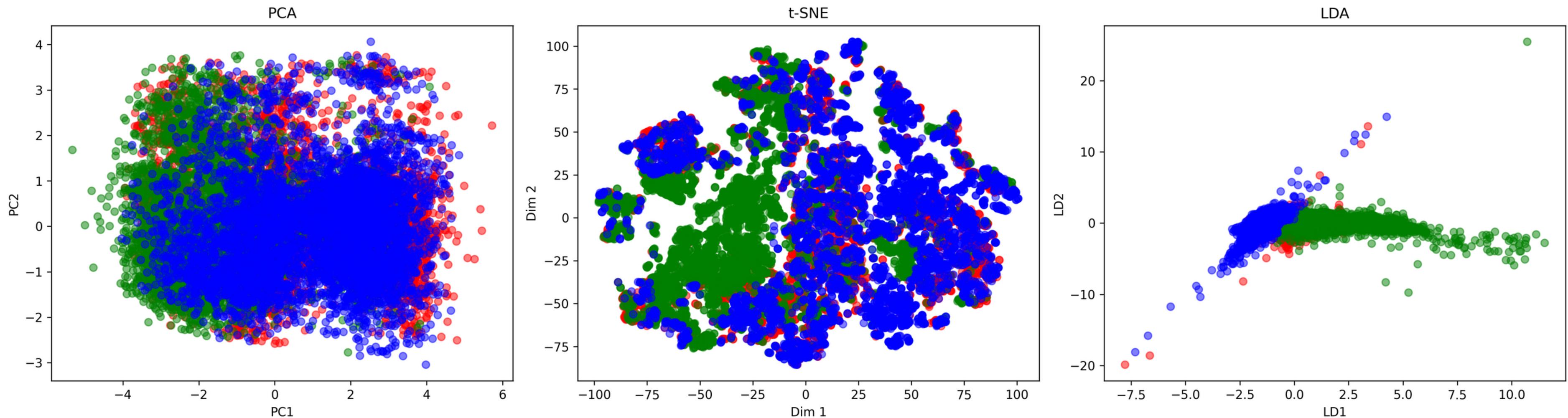
$$NH(k) = \frac{1}{nk} \sum_{i=1}^n \sum_{j \in N_k^{embed}(i)} \delta(y_i, y_j)$$

# Ứng dụng trong bài toán thực tế

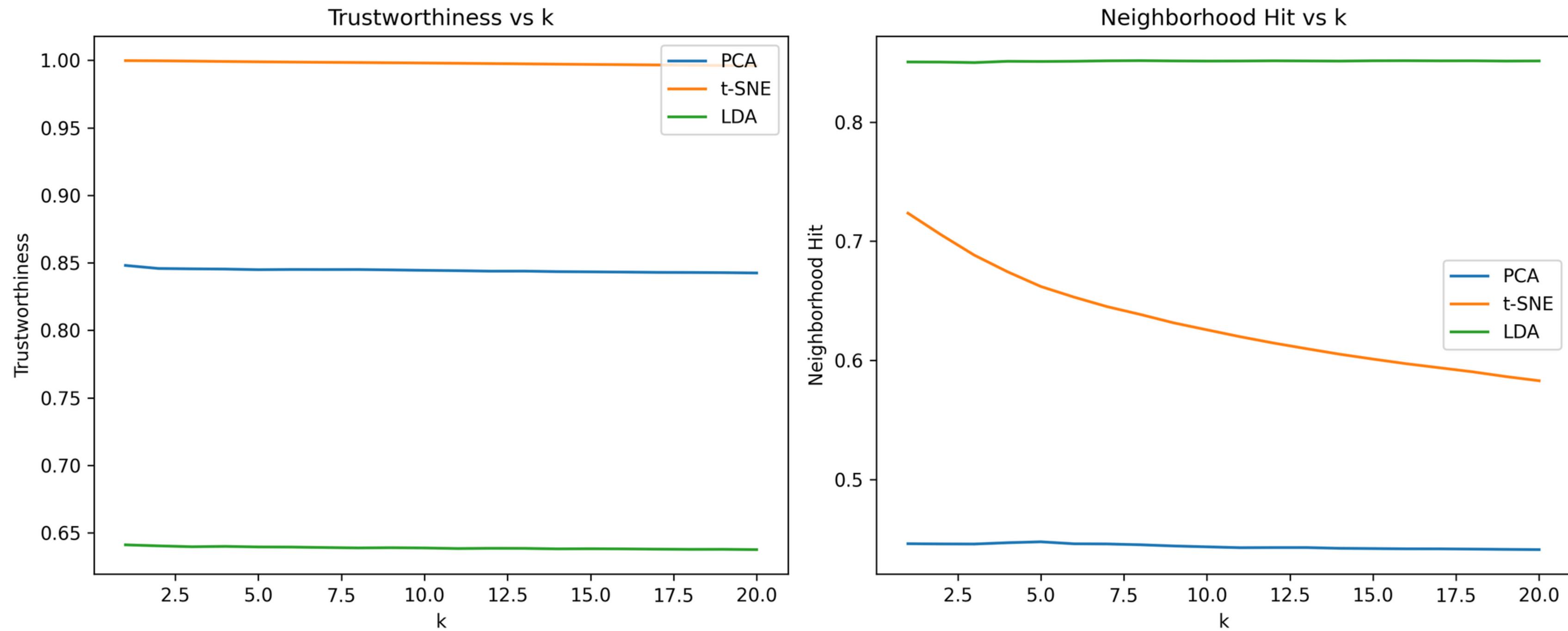
- Bộ dữ liệu: Stellar Classification Dataset
- Dữ liệu của 100000 khảo sát thiên văn của trạm quan sát SDSS, bao gồm:
  - 17 đặc trưng gồm: quang phổ, hồng ngoại, vị trí quét, sợi quang học,...
  - Nhãn phân loại: sao, thiên hà và chuẩn tinh



# Ứng dụng trong bài toán thực tế



# Ứng dụng trong bài toán thực tế





**HUST**

**THANK YOU !**