

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN  
KHOA CÔNG NGHỆ THÔNG TIN**



## **BÁO CÁO BÀI TẬP CÁ NHÂN LAB 1 MÔN TRỰC QUAN HÓA DỮ LIỆU**



**Giảng viên:** Bùi Tiến Lên  
Lê Nhựt Nam  
Lê Ngọc Thành  
Nguyễn Thị Thu Hằng

**Lớp:** 21\_21

**MSSV:** 21120201

**Họ và tên:** Bùi Đình Bảo

**Học kỳ 1 – Năm học 2023–2024**

# Mục lục

|                                      |    |
|--------------------------------------|----|
| TỰ ĐÁNH GIÁ MỨC ĐỘ HOÀN THÀNH: ..... | 3  |
| CƠ SỞ LÝ THUYẾT CỦA PCA: .....       | 4  |
| 1. Động lực nghiên cứu về PCA: ..... | 4  |
| 2. Đặt vấn đề: .....                 | 4  |
| 3. Cơ sở toán học của PCA: .....     | 5  |
| 4. Thuật toán PCA: .....             | 9  |
| TRIỂN KHAI THUẬT TOÁN PCA: .....     | 11 |
| 1. Minh họa bằng số: .....           | 11 |
| 2. Minh họa bằng code: .....         | 14 |
| REFERENCES .....                     | 15 |

## TỰ ĐÁNH GIÁ MỨC ĐỘ HOÀN THÀNH:

| Công việc   | Mức độ hoàn thành |
|---|-------------------|
| Studying PCA  |                   |
| - Motivation  | 100%              |
| - Problem statement                                 | 100%              |
| - PCA algorithms                                    | 100%              |
| Implementation PCA                                  | 100%              |
| Overall comprehension of the submitted source code. | 100%              |
| Bonus points  |                   |
| - Mathematics behind PCA                            | 100%              |
| - Numerical demo                                    | 100%              |
| <b>Tổng</b>   | <b>110%</b>       |

# CƠ SỞ LÝ THUYẾT CỦA PCA:

## 1. Động lực nghiên cứu về PCA:

Principal Components Analysis (PCA) - Phân tích thành phần chính: Là phương pháp phân tích nhằm giảm chiều dữ liệu (giữ lại những thành phần/biến quan trọng) mà vẫn giữ được các đặc trưng cơ bản của dữ liệu.

Phân tích PCA sẽ giúp chúng ta:

- Giảm chiều dữ liệu (loại bỏ biến nhiễu và dư thừa) mà vẫn giữ được đặc trưng cơ bản của dữ liệu.
- Tiết kiệm nguồn lực, thời gian phân tích dữ liệu, đặc biệt là tối ưu hóa thời gian chạy các mô hình học máy.
- Thuận lợi trong phân tích dữ liệu và vẫn cung cấp cái nhìn trực quan, tổng quát.
- Xác định các biến tương quan.

Trong Machine Learning, Dimension Reduction (giảm chiều ma trận) là quá trình biến đổi dữ liệu từ không gian có số chiều cao về không gian có số chiều thấp, nhưng vẫn giữ được những đặc tính có ý nghĩa của dữ liệu. Trường hợp lý tưởng là sau khi thực hiện dimension reduction, số chiều sau khi biến đổi sẽ bằng hoặc gần bằng số chiều tối thiểu để biểu diễn dữ liệu đó.

Làm việc với dữ liệu ở không gian cao chiều có nhiều nhược điểm: dữ liệu thô thì thường sẽ thừa, do hậu quả của curse of dimensionality (lời nguyền đa chiều), và quá trình phân tích dữ liệu thường sẽ mất rất nhiều thời gian để xử lý dữ liệu. Giảm chiều dữ liệu phổ biến trong các lĩnh vực có số lượng bản ghi lớn và/hoặc số lượng features lớn, chẳng hạn như xử lý tín hiệu, nhận dạng tiếng nói, thông tin học thần kinh (tin học thần kinh, neuroinformatics), và tin sinh học.

Các phương pháp giảm chiều dữ liệu thông thường được chia thành 2 loại là tuyến tính và phi tuyến tính. Các cách tiếp cận cũng được chia thành lựa chọn đặc trưng (feature selection) và trích chọn đặc trưng (feature extraction). Giảm chiều dữ liệu có thể được sử dụng cho giảm nhiễu (noise reduction), trực quan hóa dữ liệu (data visualization), phân tích cụm (cluster analysis), hoặc là một bước trung gian để tạo điều kiện thuận lợi cho các phân tích khác.

## 2. Đặt vấn đề:

Dimensionality Reduction, nói một cách đơn giản, là việc đi tìm một hàm số, hàm số này lấy đầu vào là một điểm dữ liệu ban đầu  $x \in \mathbb{R}^D$  với  $D$  rất lớn, và tạo ra một điểm dữ liệu mới  $z \in \mathbb{R}^K$  có số chiều  $K < D$ .

Phương pháp phân tích thành phần chính dựa trên quan sát rằng dữ liệu thường không phân bố ngẫu nhiên trong không gian mà thường phân bố gần các đường/mặt đặc biệt nào đó.

PCA xem xét một trường hợp đặc biệt khi các mặt đặc biệt đó có dạng tuyến tính là các không gian con (subspace).

### 3. Cơ sở toán học của PCA:

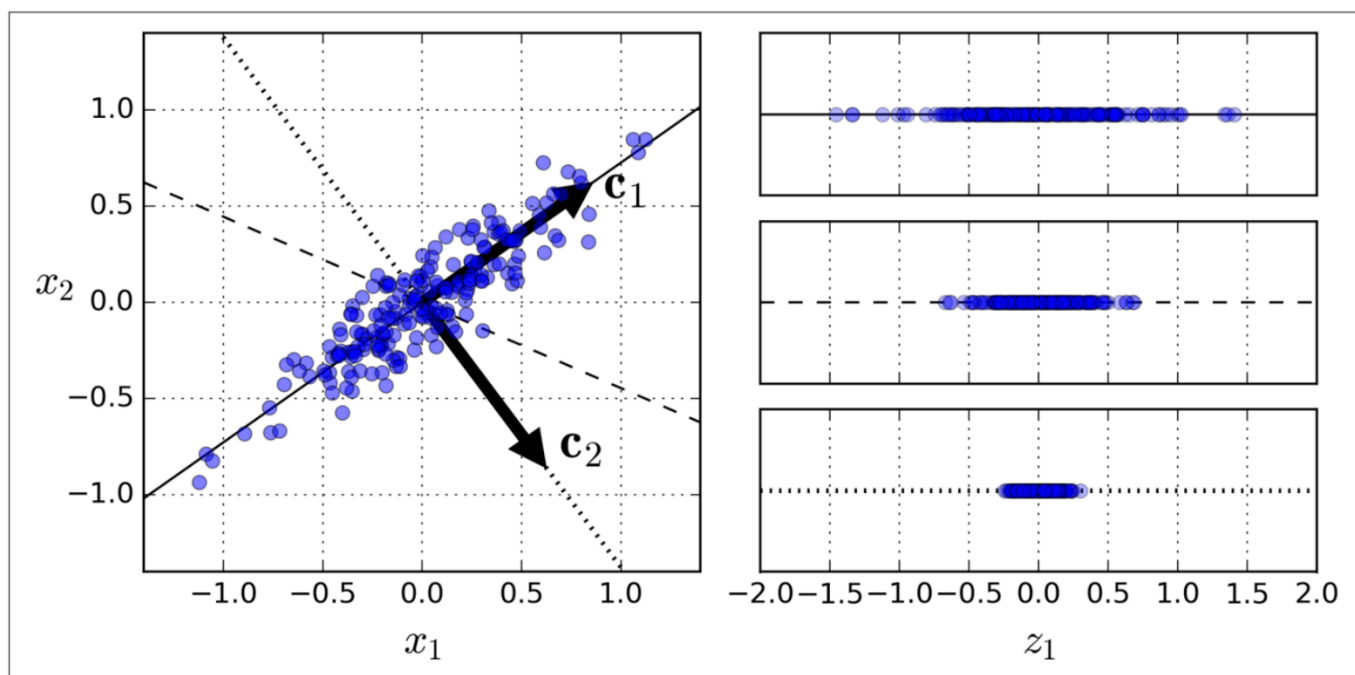
PCA là phương pháp tuyến tính thông dụng nhất để giảm chiều dữ liệu. PCA thực hiện một phép biến đổi tuyến tính, chuyển dữ liệu sang không gian thấp chiều hơn. Khi thực hiện biến đổi, PCA tối đa hóa variance dữ liệu ở không gian thấp chiều.

Trong thực tế, ta sử dụng ma trận covariance (và đôi khi cả ma trận tương quan) và tính các eigenvector của ma trận. Eigenvector có eigenvalue lớn nhất (thành phần chính - principle component) được chọn để tái tạo một lượng variance lớn của dữ liệu. Những eigenvectors tiếp theo còn có thể được sử dụng để mô tả các tính chất và biểu hiện lý học của không gian chứa dữ liệu.

Phép biến đổi tạo ra những ưu điểm sau đối với dữ liệu:

- Giảm số chiều của không gian chứa dữ liệu khi nó có số chiều lớn.
- Xây dựng những trục tọa độ mới, thay vì giữ lại các trục của không gian cũ, nhưng lại có khả năng biểu diễn dữ liệu tốt tương đương, và đảm bảo độ biến thiên của dữ liệu trên mỗi chiều mới.
- Tạo điều kiện để các liên kết tiềm ẩn của dữ liệu có thể được khám phá trong không gian mới, mà nếu đặt trong không gian cũ thì khó phát hiện vì những liên kết này không thể hiện rõ.
- Đảm bảo các trục tọa độ trong không gian mới luôn trực giao đôi một với nhau, mặc dù trong không gian ban đầu các trục có thể không trực giao.
- Chúng ta hãy tiếp cận một ví dụ để có thể hiểu được thuật toán của PCA.

Trước khi ta có thể chiếu tập dữ liệu về một không gian có chiều thấp hơn, ta cần chọn mặt phẳng phù hợp để chiếu dữ liệu.



Ảnh bên dưới hiển thị một tập dữ liệu 2 chiều đơn giản. Trên tập dữ liệu này, có 3 trục (mặt phẳng 1 chiều), và phép chiếu dữ liệu lên 3 trục này sẽ cho ra 3 thể hiện khác nhau của cùng tập dữ liệu đó.

Có thể dễ thấy rằng trục đầu tiên, ký hiệu  $c_1$ , sẽ bảo toàn được variance của tập dữ liệu cũ một cách tốt nhất, do phép chiếu lên trục này sẽ mất đi ít thông tin hơn so với 2 trục còn lại. Hoặc nói theo cách khác, trục  $c_1$  là trục tối thiểu hóa mất mát MSE giữa bộ dữ liệu gốc và dữ liệu đã chiếu. Đây chính là ý tưởng cơ bản của PCA.

PCA sẽ nhận diện trục có variance lớn nhất khi dữ liệu chiếu lên đó. Trong ảnh trên, đó là trục được vẽ bằng đường nét liền. Đồng thời, PCA cũng tìm trục thứ 2, vuông góc với trục đầu tiên, mà giữ lại được lượng variance lớn nhất còn lại. Ví dụ của chúng ta là dữ liệu 2 chiều, nên ngoài đường nét chấm là lựa chọn duy nhất của ta.

Nếu PCA được áp dụng lên bộ dữ liệu có số chiều cao hơn, PCA sẽ tiếp tục tìm trục thứ 3, thứ 4, và tiếp tục cho đến khi số trục bằng số chiều dữ liệu.

Làm thế nào để ta có thể tìm các thành phần chính của dữ liệu? Ta sử dụng một kỹ thuật phân rã ma trận là Singular Value Decomposition (phân rã giá trị đơn). SVD phân bất kỳ một ma trận  $X$  thành tích của 3 ma trận, là  $U, \sigma, V^T$ , trong đó có  $V^T$  chứa tất cả các thành phần chính ta cần tìm.

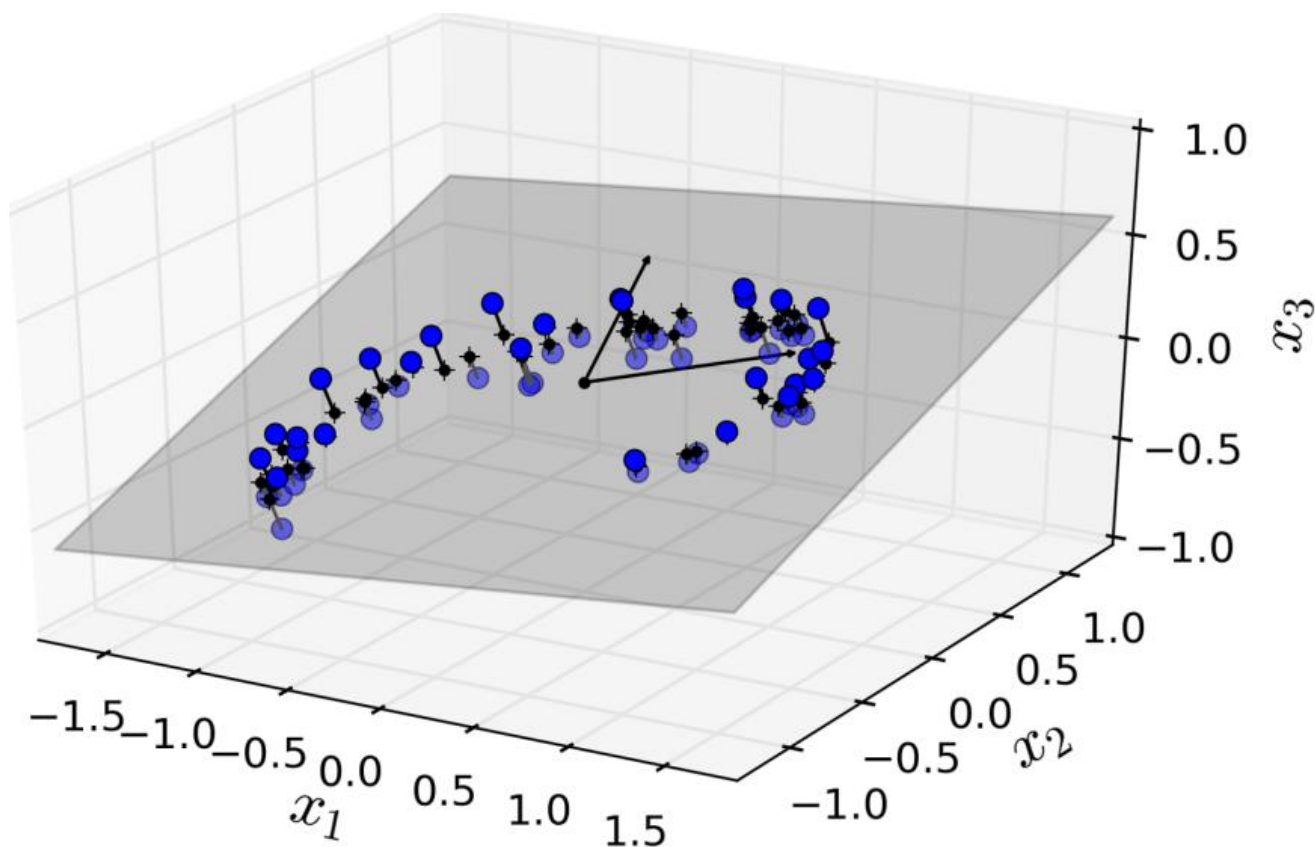
$$V^T = \begin{pmatrix} | & | & \dots & | \\ c_1 & c_2 & \dots & c_n \\ | & | & \dots & | \end{pmatrix}$$

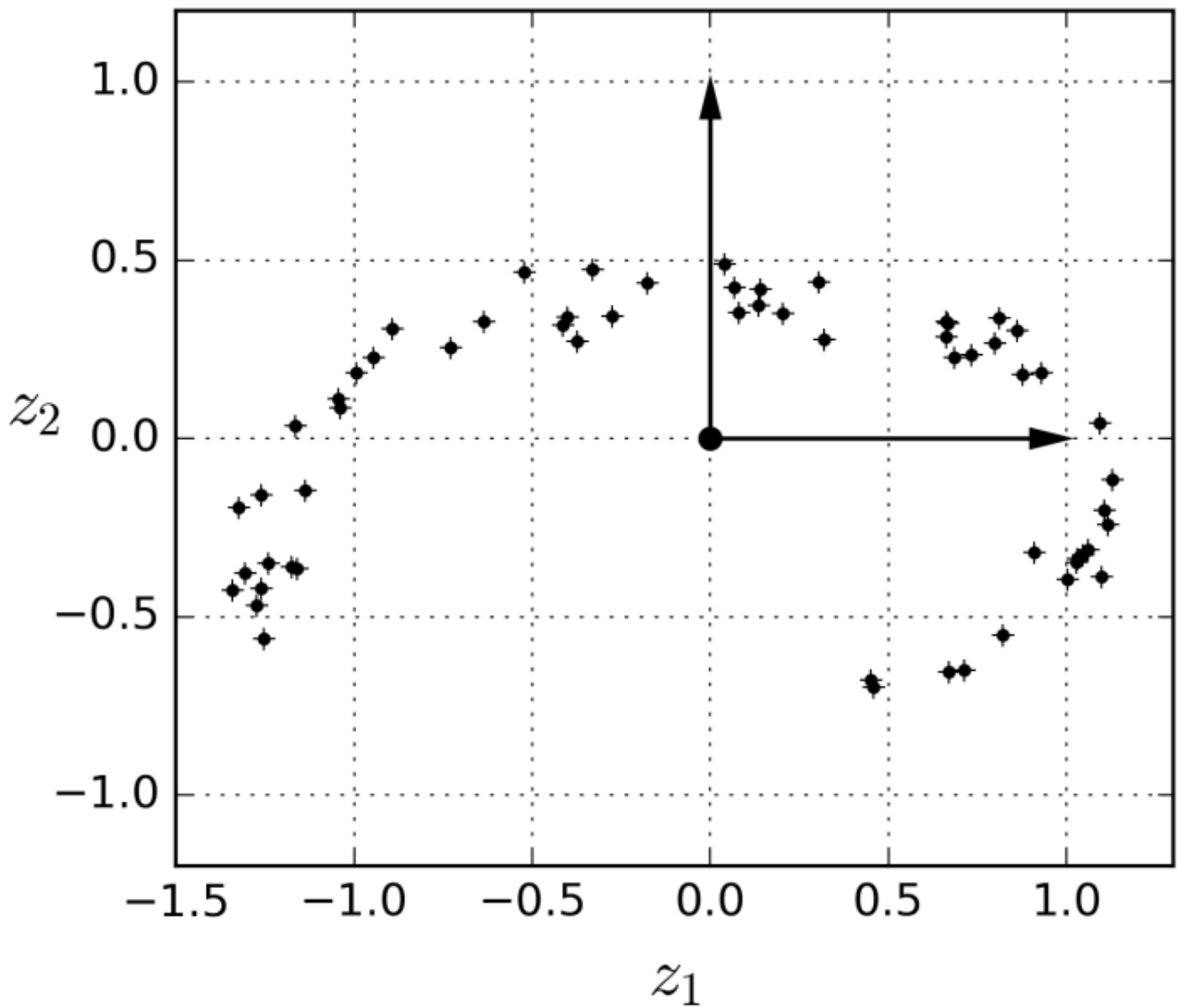
Khi thực hiện PCA, bắt buộc phải center dữ liệu về điểm  $O(0,0)$ . Hàm PCA của Scikit-learn sẽ tự động khi người lập trình sử dụng, nhưng nếu người dùng tự reimplement lại PCA, cần phải center dữ liệu.

Đoạn code Python sau đây sử dụng hàm `svd()` của numpy để lấy tất cả các thành phần chính của tập `X`, rồi lấy ra 2 thành phần cơ bản đầu tiên:

```
X_centered = X - X.mean(axis=0)
U, s, V = np.linalg.svd(X_centered)
c1 = V.T[:, 0]
c2 = V.T[:, 1]
```

Sau khi đã xác định được tất cả các thành phần cơ bản, ta có thể giảm chiều ma trận xuống còn  $d$  chiều, bằng cách chiếu tập dữ liệu vào không gian  $d$ -chiều được xác lập bởi  $d$  thành phần chính. Chọn hyperplane này sẽ đảm bảo phép chiếu bảo toàn được tối đa lượng variance có thể. Ví dụ cụ thể ở 2 hình dưới đây:





Dataset 3D được chiếu xuống mặt phẳng 2D, xác lập bởi 2 thành phần chính, bảo toàn được một lượng variance lớn. Chính vì vậy, dữ liệu được chiếu có hình dạng rất giống dữ liệu gốc trên đồ thị.

Để có thể chiếu dữ liệu sang siêu mặt phẳng thấp chiều, ta có thể sử dụng phép nhân ma trận  $X$  với ma trận  $W_d$ , được định nghĩa là ma trận chứa  $d$  cột đầu tiên của  $V^T$ .

$$X_{d-proj} = X \cdot W_d$$

Scikit-learn cũng hỗ trợ sử dụng PCA, trong thư viện `sklearn.decomposition`. Ta có thể sử dụng nó như sau:

```
from sklearn.decomposition import PCA
pca = PCA(n_components = 2)
X2D = pca.fit_transform(X)
```



## 4. Thuật toán PCA:

Các bước thực hiện của thuật toán PCA:

*Bước 1:* Tính vector kỳ vọng của toàn bộ dữ liệu:

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

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

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

*Bước 3:* Tính ma trận hiệp phương sai:

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

*Bước 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.

*Bước 5:* Chọn K vector riêng ứng với K trị riêng lớn nhất để xây dựng ma trận  $U_K$  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.

*Bước 6:* Chiếu dữ liệu ban đầu đã chuẩn hoá  $\hat{X}$  xuống không gian con tìm được.

*Bước 7:* Dữ liệu mới chính là tọa độ của các điểm dữ liệu trên không gian mới.

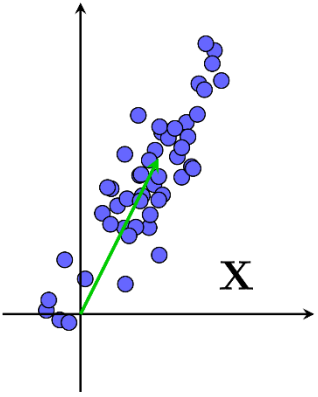
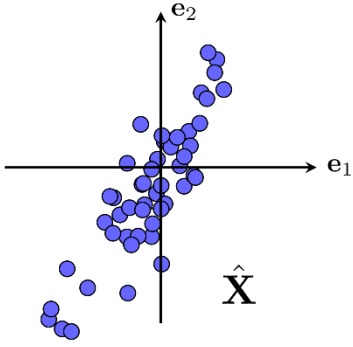
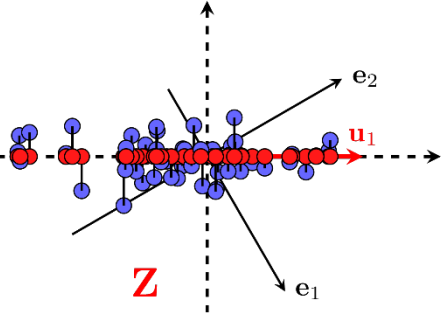
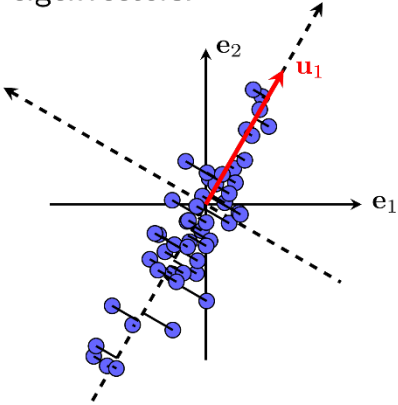
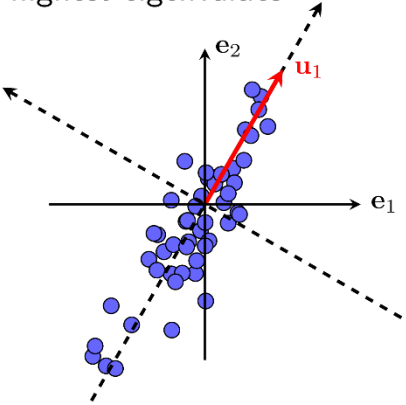
$$Z = U_K^T \hat{X}$$

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

$$x \approx U_K Z + \bar{x}$$

Các bước thực hiện PCA có thể được xem trong hình dưới đây:

# PCA procedure

|  |   |  |
|--|---|--|
| <p>1. Find mean vector</p>  <p>A scatter plot of blue data points in a 2D space. A green line segment represents the mean vector, starting from the origin and pointing towards the center of the data cluster. The label <b>X</b> is placed near the bottom right of the plot.</p> | <p>2. Subtract mean</p>  <p>The data points are now centered around the origin. The axes are labeled <math>e_1</math> and <math>e_2</math>. The label <math>\hat{\mathbf{X}}</math> is placed near the bottom right of the plot.</p>                                       | <p>3. Compute covariance matrix:<br/> <math display="block">\mathbf{S} = \frac{1}{N} \hat{\mathbf{X}} \hat{\mathbf{X}}^T</math></p> <p>4. Compute eigenvalues and eigenvectors of <math>\mathbf{S}</math>:<br/> <math>(\lambda_1, \mathbf{u}_1), \dots, (\lambda_D, \mathbf{u}_D)</math><br/> Remember the orthonormality of <math>\mathbf{u}_i</math>.</p>                    |
| <p>7. Obtain projected points in low dimension.</p>  <p>The data points are projected onto the <math>e_1</math> axis, represented by a dashed line. The projected points are shown as red dots along this axis. The label <b>Z</b> is placed near the bottom left of the plot.</p> | <p>6. Project data to selected eigenvectors.</p>  <p>The data points are shown with a red arrow labeled <math>\mathbf{u}_1</math> indicating the direction of the first principal component. Dashed lines show the projection of the data points onto this direction.</p> | <p>5. Pick <math>K</math> eigenvectors w. highest eigenvalues</p>  <p>The data points are shown with a red arrow labeled <math>\mathbf{u}_1</math> indicating the direction of the first principal component. Dashed lines show the projection of the data points onto this direction.</p> |

## TRIỂN KHAI THUẬT TOÁN PCA:

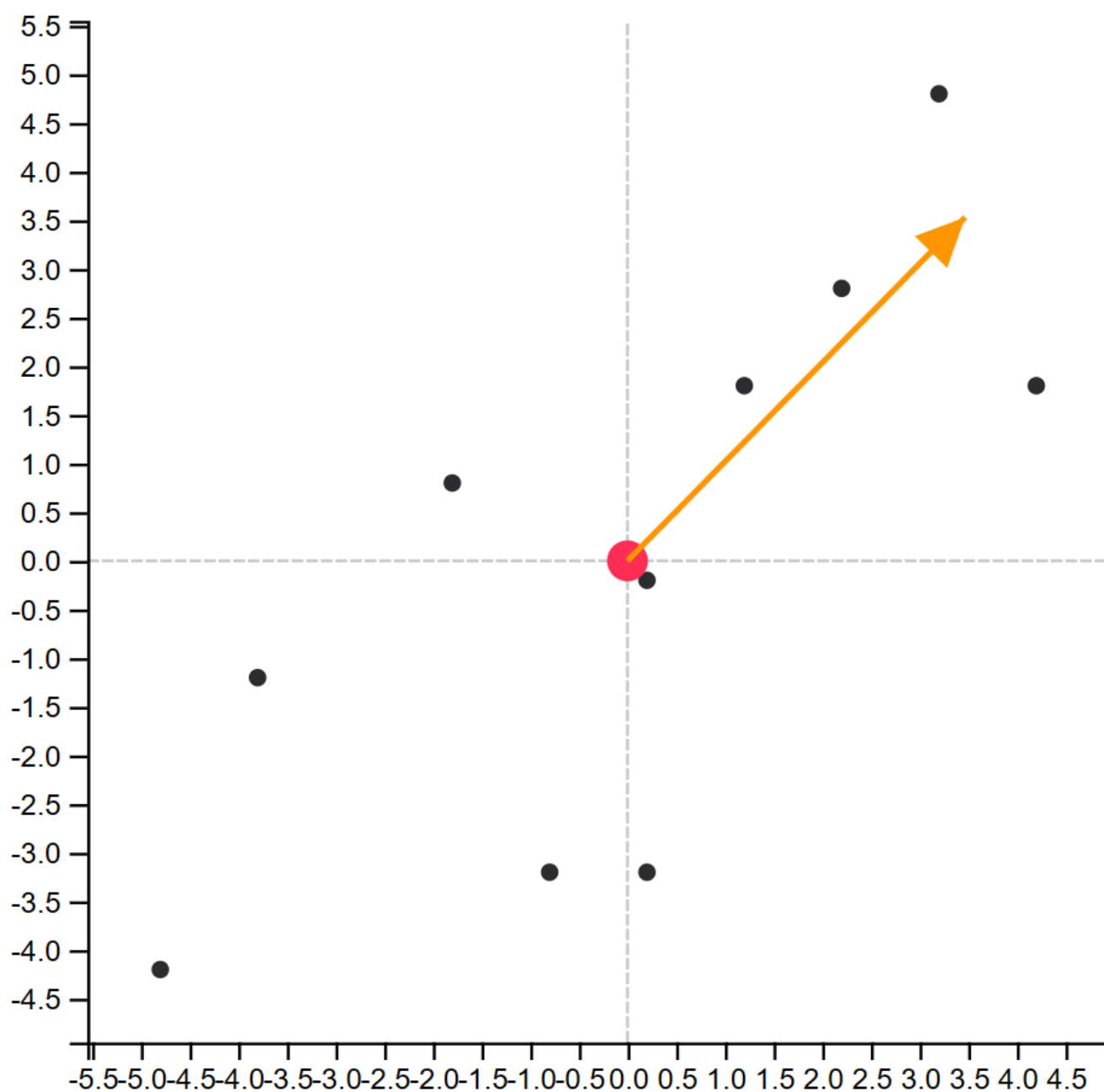
### 1. Minh họa bằng số:

Triển khai thuật toán PCA trên bộ dữ liệu điểm số đơn giản như sau:

|        |    |   |   |   |   |   |   |   |   |   |
|--------|----|---|---|---|---|---|---|---|---|---|
| Toán   | 8  | 6 | 4 | 5 | 9 | 5 | 0 | 3 | 7 | 1 |
| Vật lý | 10 | 7 | 2 | 5 | 7 | 2 | 1 | 6 | 8 | 4 |

*Bước 1:* Tính vector kỳ vọng của toàn bộ bộ dữ liệu:

$$\bar{x} = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix} = \frac{1}{n} \sum_{i=1}^n x_i = \begin{bmatrix} 4.800 \\ 5.200 \end{bmatrix}$$



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

$$\hat{x}_i = x_i - \bar{x}$$

*Bước 3:* Tính ma trận hiệp phương sai:

$$S = \frac{1}{n-1} \sum_{i=1}^n \hat{x}_i \hat{x}_i^T = \begin{bmatrix} 8.400 & 6.267 \\ 6.267 & 8.622 \end{bmatrix}$$

*Bước 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:

Tính các cặp trị riêng, vector riêng  $(\lambda, e)$  của ma trận  $S$  với  $e$  được chuẩn hóa. Sắp xếp theo thứ tự giảm dần của trị riêng, trong trường hợp này ta có:

$$\lambda_1 = 14.799; e_1 = \begin{bmatrix} 0.701 \\ 0.713 \end{bmatrix}$$

$$\lambda_2 = 2.243; e_2 = \begin{bmatrix} -0.713 \\ 0.701 \end{bmatrix}$$

*Bước 5:* Chọn  $K$  vector riêng ứng với  $K$  trị riêng lớn nhất để xây dựng ma trận  $U_K$  có các cột tạo thành một hệ trục 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:

Vì dữ liệu hiện tại là 2 chiều nên  $k$  có hai sự lựa chọn  $k=1$  hoặc  $k=2$ .

$$\lambda_1 = 14.799; e_1 = \begin{bmatrix} 0.701 \\ 0.713 \end{bmatrix}$$

Tỉ lệ diễn giải xu hướng biến thiên của chiều vừa mới chọn (tỉ lệ đóng góp vào diễn giải tổng phương sai của các biến ngẫu nhiên ban đầu):

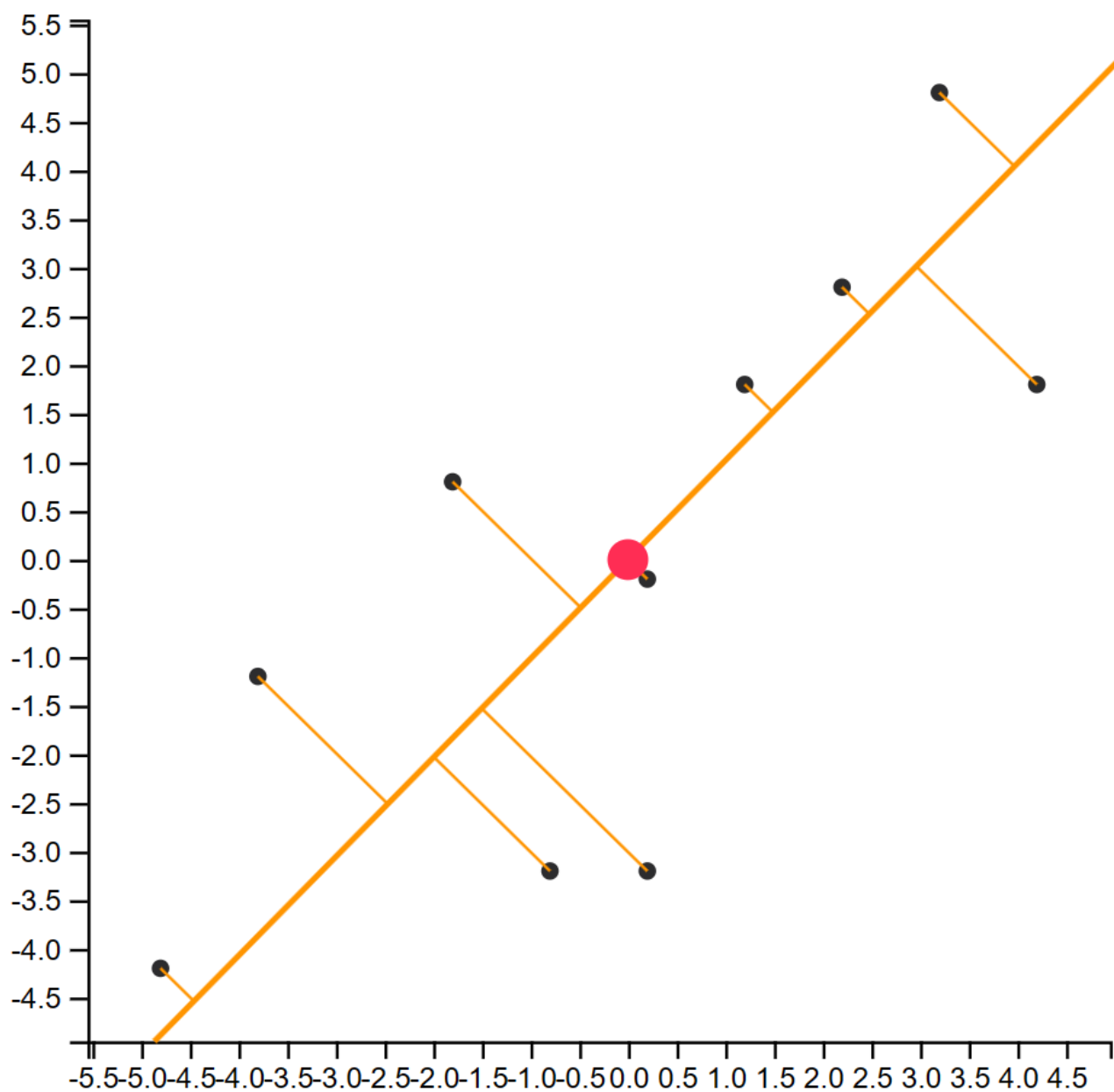
$$\frac{\lambda_1}{\lambda_1 + \lambda_2} = 0.868$$

*Bước 6:* Chiếu dữ liệu ban đầu đã chuẩn hoá  $\hat{X}$  xuống không gian con tìm được:

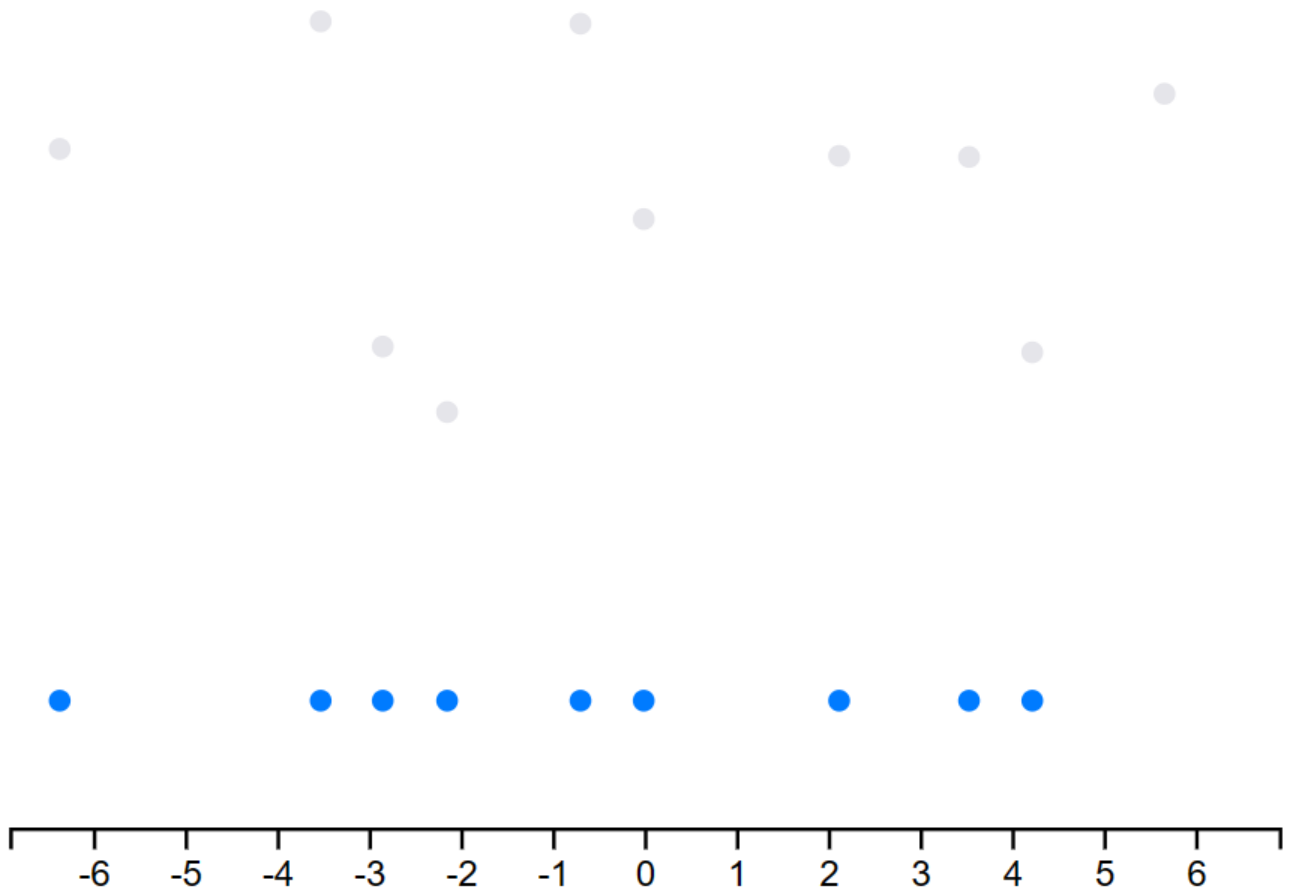
Mỗi tọa độ trong không gian mới được tính bởi công thức:

$$y_i = e_i^T x$$

$$y_i = e_i^T x = [0.701 \quad 0.713] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0.701x_1 + 0.713x_2$$



*Bước 7:* Dữ liệu mới chính là tọa độ của các điểm dữ liệu trên không gian mới:



## 2. Minh họa bằng code:

Được minh chứng trong source code của bài nộp, bao gồm:

- PCA áp dụng cho bộ dữ liệu điểm số đơn giản ở trên.
- PCA áp dụng cho bộ dữ liệu phân lớp các loại rượu vang.

Link dataset: <https://www.kaggle.com/datasets/brynja/wineuci>

## REFERENCES

1. Sách tham khảo Mathematics for Machine Learning (được cho sẵn trong Lab).
2. Trang machine learning cơ bản: <https://machinelearningcoban.com/2017/06/15/pca/>
3. Tổng quan về PCA của trang tek4: <https://tek4.vn/khoa-hoc/machine-learning-co-ban/giam-chieu-du-lieu>
4. Hỗ trợ trực quan hóa thuật toán PCA của trang thetalog: [https://thetalog.com/thetaflow/pca2d-visualizer/?fbclid=IwAR3Ck\\_bcJzcdP8sYsy5bbR42gOuxSN0J3ctakVLFTv39s-Tz8uXSB2uXWE](https://thetalog.com/thetaflow/pca2d-visualizer/?fbclid=IwAR3Ck_bcJzcdP8sYsy5bbR42gOuxSN0J3ctakVLFTv39s-Tz8uXSB2uXWE)
5. Tính trị riêng và vector riêng của ma trận: [https://elearning.tcu.edu.vn/1151/51\\_gi\\_tr\\_ring\\_vector\\_ring\\_ca\\_ma\\_trn.html](https://elearning.tcu.edu.vn/1151/51_gi_tr_ring_vector_ring_ca_ma_trn.html)
6. Về ma trận hiệp phương sai: [https://vi.wikipedia.org/wiki/Ma\\_tr%E1%BA%ADn\\_hi%E1%BB%87p\\_ph%C6%B0%C6%A1ng\\_sai](https://vi.wikipedia.org/wiki/Ma_tr%E1%BA%ADn_hi%E1%BB%87p_ph%C6%B0%C6%A1ng_sai)
7. Sử dụng ChatGPT để giải đáp thắc mắc.

-----HẾT-----