ĐẠI HỌC QUỐC GIA TPHCM

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

Báo cáo Bài tập thực hành

ILab-01: Principle Components Analysis Visualization

Môn học: Trực quan hóa dữ liệu

Sinh viên thực hiện:

Lê Thị Minh Phương

MSSV: 21120533

Giảng viên hướng dẫn:

Lê Nhựt Nam

Ngày 5 tháng 5 năm 2024

LỜI CẢM ƠN

Em xin chân thành cảm ơn thầy Lê Nhựt Nam đã cung cấp những tài liệu bổ ích, tận tình hướng dẫn và hỗ trợ cho chúng em trong suốt toàn bộ môn học nói chung và trong bài tập thực hành này nói riêng.

Mục lục

1 Tổng quan

1.1 Chủ đề bài tập

Principal Component Analysis (PCA)

1.2 Yêu cầu bài tập

Với một tập dữ liệu tự chọn:

- Nghiên cứu về PCA: Động lực, phát biểu vấn đề, giải thích toán học đằng sau PCA (bonus), thuật toán PCA, demo tính toán số (bonus).
- Áp dụng PCA trên tập dữ liệu đã chọn.

1.3 Các tiêu chí đánh giá

Danh sách các tiêu chí đánh giá của bài tập được trình bày trong bảng dưới đây. Phần tự đánh giá về mức độ hoàn thành của bài tập này được trình bày trong phần 9.

Yêu cầu	Điểm
Nghiên cứu về PCA	45%
Cài đặt PCA	45%
Hiểu tổng quan mã nguồn đã nộp	10%
Điểm cộng: Giải thích toán & Demo tính toán số	10%

2 Giới thiệu về PCA

2.1 Nguồn gốc

Trong lịch sử, PCA đã được phát minh nhiều lần và nó được biết đến với những tên gọi khác nhau trong tùy lĩnh vực.

Người đầu tiên phát minh ra PCA là Karl Pearson. Năm 1901, trong một bài báo khoa học của mình, Pearson đưa ra công thức cho PCA là tìm ra "các đường thẳng và mặt phẳng khớp nhất (best fit) với các hệ điểm trong không gian" [6]. Cách giải thích này nhấn mạnh các đặc tính mô hình hóa của PCA và bắt nguồn chủ yếu từ tư duy hồi quy.

Sau đó, vào những năm 1930, PCA được phát minh độc lập bởi Harold Hotelling với ý tưởng lấy tổ hợp tuyến tính của các biến và variation của các thành phần chính. Cách tiếp cận của Hotelling là một cách tiếp cận thống kê đa biến.
Sau này, người ta nhận ra rằng hai cách tiếp cận này rất giống nhau. Phần 3 sẽ đề cập chi tiết hơn về hai cách tiếp cận này.

2.2 Định nghĩa

Một số định nghĩa hoàn chỉnh: Principal Component Analysis (PCA), hay phân tích thành phần chính, là:

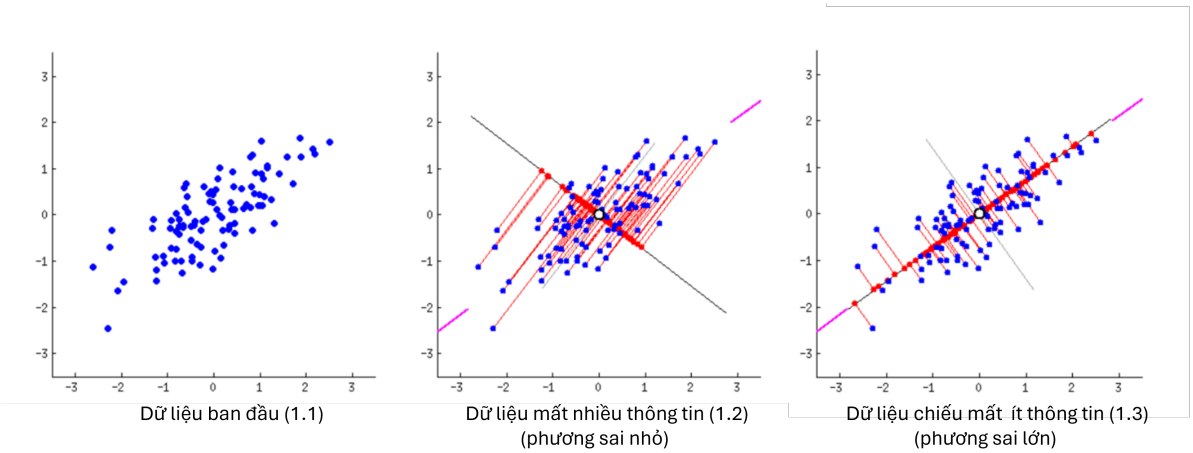
- Một kỹ thuật tìm một tổ hợp tuyến tính của các biến sao cho giải thích tốt nhất covariation structure giữa các biến. [5]
- Một kỹ thuật đa biến (multivariate) để phân tích một bảng dữ liệu trong đó các quan sát được mô tả bằng một số biến phụ thuộc (dependent) định lượng (quantitative) có tương quan với nhau (inter-correlated). [3]

Tựu trung lại, PCA là một kỹ thuật giảm chiều tuyến tính (linear dimensionality reduction)[2] sử dụng sự phụ thuộc giữa các biến của dữ liệu nhiều chiều để biểu diễn nó ở ít chiều hơn hơn mà không làm mất quá nhiều thông tin. [9]
Mục tiêu của PCA:

- Data reduction: Giảm chiều dữ liệu.
- Data interpretation: Giữ lại nhiều thông tin nhất có thể.

3 Tiếp cận bài toán

Khi làm việc với dữ liệu nhiều chiều (mỗi chiều là một biến), ta sẽ gặp khó khăn trong việc hiểu, phân tích và trực quan dữ liệu. Chi phí lưu trữ và tính toán cao cũng là một khó khăn gặp phải đối với dữ liệu nhiều chiều. Vì vậy, ta sẽ có nhu cầu làm thế nào để số chiều dữ liệu giảm xuống (một số lượng chiều nhỏ hơn mà ta mong muốn) nhưng vẫn giữ lại được nhiều thông tin nhất có thể.



Hình 1: Phương sai và sự mất mát thông tin

Nguồn: PCA - Sagor Saha

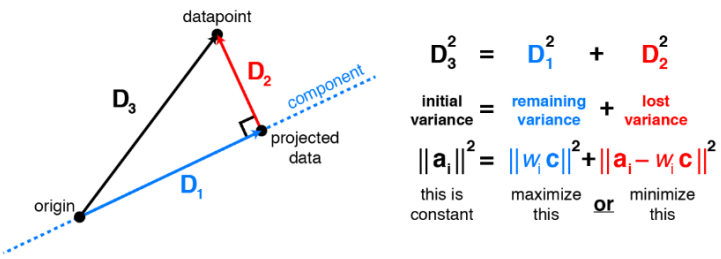
Ta có thể tiếp cận bài toán trên theo hai hướng khác nhau:

3.1 Hướng tiếp cận của Pearson (góc nhìn hồi quy)

Tìm một đường thẳng, một mặt phẳng hay tổng quát hơn là một không gian con của không gian ban đầu sao cho không gian con này khớp nhất với dữ liệu gốc. Tức là tìm không gian con có độ lỗi nhỏ nhất so với dữ liệu gốc.

3.2 Hướng tiếp cận của Hotelling (góc nhìn thống kê đa biến)

Tìm không gian con sao cho khi chiếu dữ liệu gốc lên đó thì thông tin được giữ lại nhiều nhất có thể bằng cách giữa lại lượng variation của dữ liệu nhiều nhất có thể (đo bằng phương sai) [1]. Tức là tìm không gian con sao cho phương sai của dữ liệu ảnh là lớn nhất (hình 1.3).



Hình 2: Mối quan hệ giữa độ lỗi và phương sai

Nguồn: PCA - Ethen

Dù tiếp cận từ hai bài toán tối ưu khác nhau nhưng hai hàm mục tiêu lại có mối quan hệ chặt chẽ với nhau. Trong hình 2, ta thấy độ lỗi ở đây chính là lost variance (lượng phương sai mất đi khi chiếu dữ liệu lên không gian con); và phương sai của dữ liệu thu được bởi phép chiếu chính là remaining variance (phương sai giữ lại được bởi phép chiếu này). Theo định lý Pythagoras, với initial variance (phương sai của dữ liệu ban đầu) không đổi, lost variance càng lớn thì remaining variance càng nhỏ và ngược lại. Hay nói cách khác, bài toán minimize độ lỗi của không gian chiếu có cùng nghiệm với bài toán maximize phương sai của dữ liệu ảnh (hai bài toán tương đương nhau). Chứng minh toán của mối quan hệ giữa độ lỗi chiếu và phương sai được trình bày trong phần 5.3.

4 Giả định

Để có thể áp dụng được PCA, dữ liệu cần phải có một số tính chất nhất định. Dữ liệu trong đời thực phức tạp và thường không thỏa mãn tính chất này, tuy nhiên chúng ta có thể thực hiện PCA với một số giả định sau đây:

4.1 Tính tuyến tính (Linearity)

Tính tuyến tính (linearity) có thể hiểu là tính chất có thể biểu diễn các điểm trong tập dữ liệu dưới dạng một tổ hợp tuyến tính, mỗi vector cơ sở ứng với một thuộc tính.

Hầu hết dữ liệu trong thực tế là phi tuyến, biểu hiện ở các thuộc tính định tính (qualitative). Tuy nhiên, xấp xỉ tuyến tính cục bộ thường có thể cung cấp một xấp xỉ tốt vì ở gần điểm cân bằng ổn định, các số hạng phi tuyến thường trở nên không đáng kể, cho phép chúng ta biểu diễn gần đúng dữ liệu bằng các tổ hợp tuyến tính. [10]

4.2 Phân phối Gauss

Điểm cốt lõi của giả định này đó là phân phối của dữ liệu được mô tả đầy đủ thông qua giá trị trung bình và phương sai. Loại phân phối duy nhất thỏa mãn là phân phối Gauss. [10]

4.3 Signal-to-Noise Ratio cao

Điểm cốt lõi của giả định này đó là các tín hiệu quan trọng trong dữ liệu mạnh hơn nhiều (tức là có phương sai cao hơn) so với nhiễu. Điều này đồng nghĩa với việc các phương sai lớn có đủ ý nghĩa trong việc thể hiện pattern của dữ liệu. [10]

4.4 Các thành phần chính trực giao

Tính trực giao thể hiện các thành phần chính không tương quan với nhau. Đây là một giả định quan trọng vì nó giúp giới hạn phạm vi chọn các vector cơ sở, từ đó đơn giản hóa phép toán của PCA và làm cho nó hiệu quả về mặt tính toán

Tuy nhiên giả định này đồng nghĩa với việc áp đặt cấu trúc tuyến tính trên dữ liệu, điều này có thể không phải lúc nào cũng phù hợp. Đối với dữ liệu có cấu trúc phi tuyến, các kỹ thuật khác như kernel PCA hoặc manifold learning có thể phù hợp hơn. [10]

5 Phát triển bài toán

Phần này sẽ mô tả quá trình phát triển bài toán Phân tích thành phần chính bằng các chứng minh toán học đơn giản. **Hướng tiếp cận thống kê đa biến** mang nhiều ý nghĩa quan trọng và sẽ được chọn làm hướng tiếp cận chính trong phần này. Tuy nhiên, để dễ hiểu và tự nhiên, trước tiên ta sẽ bắt đầu phần này với hướng tiếp cận hồi quy của Pearson. Sau khi phát biểu bài toán, chứng minh mối liên hệ về mặt toán học giữa độ lỗi chiếu này với phương sai, ta sẽ quay lại phát triển bài toán theo hướng tiếp cận thống kê.

Chú ý: *Những chứng minh dưới đây phần lớn được tham khảo từ bài giảng của giáo sư Laurenz Wiskott [4], đại học Stanford.*

5.1 Phát biểu bài toán

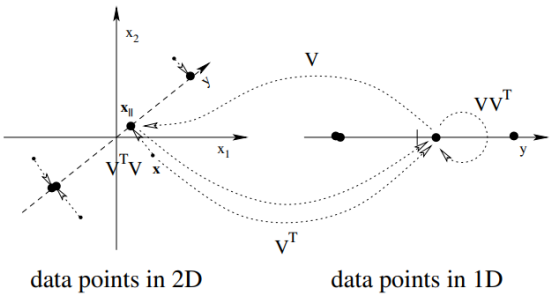
Định nghĩa 5.1 *Cho một tập* $\{x^\mu : \mu = 1, ..., M\}$ *của tập dữ liệu* I *chiều* $x^\mu = (x_1^\mu, x_2^\mu, ..., x_I^\mu)$ *có giá trị trung bình bằng 0:* $\langle x^\mu \rangle_\mu = 0_I$*. Tìm ma trận trực giao* $\mathbf{U}$ *có định thức* $|\mathbf{U}| = +1$ *là ma trận biến đổi* $x'^\mu = \mathbf{U}^T x^\mu$ *sao cho với* P *là số chiều cho trước, dữ liệu khi chiếu lên* P *chiều đầu tiên của nó* $x'^\mu_\parallel := (x'^\mu_1, x'^\mu_2, ..., x'^\mu_P, 0, ..., 0)^T$ *có* ***độ lỗi chiếu*** *nhỏ nhất*

$$E := \langle \|x'^\mu - x'^\mu_\parallel\|^2 \rangle_\mu$$

trong số tất cả các phép chiếu có thể có lên không gian con P *chiều. Các vector dòng của ma trận* $\mathbf{U}$ *đại diện cho các trục mới được gọi là các thành phần chính (principal components).*

Chú ý:

- x^μ: Tập hợp M điểm dữ liệu được chỉ mục bởi $\mu = 1, ..., M$.
- $\langle x^\mu \rangle_\mu = 0_I$: Giá trị trung bình của M điểm dữ liệu chỉ định bởi μ. Để đơn giản, từ nay ta dùng kí hiệu $\langle . \rangle$ để chỉ giá trị trung bình.
- Nếu dữ liệu có giá trị trung bình khác 0 thì thực hiện chuẩn hóa (standardize, hay center) để đưa dữ liệu về dạng có giá trị trung bình 0 bằng cách trừ tất cả các điểm dữ liệu cho giá trị trung bình này.
- Ma trận $\mathbf{U}$ với định thức bằng $+1$ tương đương với một phép quay (**rotation**, khác với **reflection** khi định thức bằng -1). Trong phép quay này, các điểm dữ liệu x có hình dạng của dữ liệu không thay đổi, chỉ có góc nhìn thay đổi. Có thể hiểu phép nhân với ma trận $\mathbf{U}^T$ là một phép quay dữ liệu hoặc phép quay hệ tọa độ.



Hình 3: Các phép ánh xạ giữa các hệ tọa độ

Nguồn: Lecture Notes on PCA, Laurenz Wiskott [4]

5.2 Ánh xạ giữa các hệ tọa độ

5.2.1 Ma trận $\mathbf{V}^T$: ánh xạ từ hệ tọa độ ban đầu I chiều vào hệ tọa độ mới P chiều

Xét các điểm dữ liệu $\mathbf{x}$ trong không gian I chiều trong hệ tọa độ ban đầu. Không gian con tuyến tính được sinh bởi P vector trực giao

$$\mathbf{v}_p := (v_{1_p}, v_{2_p}, ..., v_{I_p})^T \text{ với } \mathbf{v}_p^T v_q = \delta_{pq} := \begin{cases} 1 & \text{nếu } p = q \\ 0 & \text{nếu } p \neq q \end{cases}$$

Khi đó ta có ma trận

$$\mathbf{V} := (v_1, v2, ..., v_p)$$

Ma trận này ánh xạ $\mathbf{x}$ từ không gian I chiều trong hệ tọa độ ban đầu thành $\mathbf{y}$ trong không gian con P chiều trong hệ tọa độ mới $(P \leq I)$ sinh bởi các vector $\mathbf{v}_p$

$$\mathbf{y} := \mathbf{V}^T\mathbf{x}$$

5.2.2 Ma trận V: ánh xạ từ hệ tọa độ mới P chiều vào hệ tọa độ ban đầu I chiều

Bởi vì các vector $\mathbf{v}_p$ trực giao nên $\mathbf{V}$ có thể được dùng để chuyển ngược lại từ không gian mới về không gian ban đầu

$$\mathbf{x}_{\|} := \mathbf{V}\mathbf{y} = \mathbf{V}\mathbf{V}^T\mathbf{x}.$$

5.2.3 Ma trận $\mathbf{V}^T\mathbf{V}$: ánh xạ đồng nhất trong hệ tọa độ mới

Ma trận $\mathbf{V}^T\mathbf{V}$ thực hiện ánh xạ từ không gian con I chiều trong hệ tọa độ mới vào không gian P chiều trong hệ tọa độ ban đầu ($\mathbf{V}$), rồi ánh xạ ngược lại vào không gian con P chiều trong hệ tọa độ mới. Kết quả của phép ánh xạ này là một biểu diễn của dữ liệu trong không gian con P chiều trong hệ tọa độ mới. Ánh xạ này đã đồng nhất từ không gian con P chiều trong hệ tọa độ mới vào chính không gian P chiều trong hệ tọa độ mới (hình 3). Do đó, ánh xạ này còn được gọi là một **ánh xạ đồng nhất** (identity mapping) trong *hệ tọa độ mới*, và $\mathbf{V}^T\mathbf{V}$ được gọi là **ma trận đồng nhất** (identity matrix)

$$(\mathbf{V}^T\mathbf{V})_{pq} = \mathbf{v}_p^T\mathbf{V}_q \iff \mathbf{V}^T\mathbf{V} = \mathbf{1}_P$$

với $\mathbf{1_P}$ chỉ ma trận đồng nhất P chiều. Ví dụ với $P = 2$, ta có

$$\mathbf{V}^T\mathbf{V} = \begin{pmatrix} \mathbf{v_1}^T \\ \mathbf{v_2}^T \end{pmatrix} \begin{pmatrix} \mathbf{v_1} & \mathbf{v_2} \end{pmatrix} \begin{pmatrix} \mathbf{v_1}^T\mathbf{v_1} & \mathbf{v_1}^T\mathbf{v_2} \\ \mathbf{v_2}^T\mathbf{v_1} & \mathbf{v_2}^T\mathbf{v_2} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Chú ý: Gọi là ma trận đồng nhất vì khi nhân một ma trận bất kỳ với ma trận đồng nhất thì ma trận đó không đổi. Ma trận đồng nhất còn được gọi là ma trận đơn vị (unit matrix).

5.2.4 Ma trận $\mathbf{V}\mathbf{V}^T$: phép chiếu từ không gian I chiều vào không gian con P chiều trong hệ tọa độ ban đầu

Ma trận $\mathbf{V}\mathbf{V}^T$ thực hiện ánh xạ từ không gian I chiều trong hệ tọa độ ban đầu vào không gian con P chiều trong hệ tọa độ mới, sau đó ánh xạ ngược lại vào hệ tọa độ ban đầu. Kết quả của ánh xạ này là một biểu diễn trong không gian con P chiều trong hệ tọa độ ban đầu, do đó nó được gọi là một phép chiếu $\mathbf{P}$ từ không gian I chiều vào không gian con P chiều trong hệ tọa độ ban đầu

$$\mathbf{P} := \mathbf{V}\mathbf{V}^T.$$

Chú ý:

- Khi thực hiện liên tiếp các phép chiếu P, ta thấy kết quả tương đương với một phép chiếu P duy nhất

$$\mathbf{P}\mathbf{P} = \mathbf{V}\mathbf{V}^T\mathbf{V}\mathbf{V}^T = \mathbf{V}\mathbf{1}_P\mathbf{V}^T = \mathbf{V}\mathbf{V}^T = \mathbf{P}.$$

- Ma trận $\mathbf{P}$ có kích thước $I \times I$. Nếu số chiều $P = I$ thì $\mathbf{P} = \mathbf{1}_P$, phép chiếu tương đương với ánh xạ đồng nhất, tức là không xảy ra mất mát dữ liệu.

5.3 Mối liên hệ giữa độ lỗi chiếu và phương sai

5.3.1 Phương sai

Định nghĩa 5.2 *Phương sai của một tập dữ liệu nhiều chiều được định nghĩa là tổng phương sai của các thành phần của nó. Với tập dữ liệu có giá trị trung bình bằng 0, ta có*

$$var(\mathbf{x}) := \sum_{i=1}^{I} \langle x_i^2 \rangle = \left\langle \sum_{i=1}^{I} x_i^2 \right\rangle = \langle \mathbf{x}^T \mathbf{x} \rangle$$

5.3.2 Độ lỗi chiếu

Độ lỗi chiếu E được định nghĩa là giá trị trung bình của bình phương khoảng cách giữa các điểm ban đầu $\mathbf{x}$ và điểm chiếu $\mathbf{x}_\parallel$ như trong phần 5.1:

$$E := \langle \| x'^{\mu} - x'^{\mu}_{\parallel} \|^2 \rangle_\mu$$

5.3.3 Mối liên hệ

Để phân tích mối quan hệ giữa độ lỗi chiếu và phương sai, trước hết ta định nghĩa

$$\mathbf{x}_\perp = \mathbf{x} - \mathbf{x}_\parallel.$$

Khi đó ta biểu diễn độ lỗi chiếu như sau

$$
\begin{aligned}
E &= \langle \mathbf{x}_\perp^T \mathbf{x}_\perp \rangle \\
&= \langle (\mathbf{x} - \mathbf{x}_\parallel)^T (\mathbf{x} - \mathbf{x}_\parallel) \rangle \\
&= \langle (\mathbf{x} - \mathbf{V}\mathbf{V}^T\mathbf{x})^T (\mathbf{x} - \mathbf{V}\mathbf{V}^T\mathbf{x}) \rangle \\
&= \langle \mathbf{x}^T\mathbf{x} - 2\mathbf{x}^T\mathbf{V}\mathbf{V}^T\mathbf{x} + \mathbf{x}^T\mathbf{V}\mathbf{V}^T\mathbf{V}\mathbf{V}^T\mathbf{x} \rangle \\
&= \langle \mathbf{x}^T\mathbf{x} \rangle - \langle \mathbf{x}^T\mathbf{V}\mathbf{V}^T\mathbf{V}\mathbf{V}^T\mathbf{x} \rangle \\
&= \langle \mathbf{x}^T\mathbf{x} \rangle - \langle \mathbf{x}_\parallel^T \mathbf{x}_\parallel \rangle \\
&= \langle \mathbf{x}^T\mathbf{x} \rangle - \langle \mathbf{y}^T\mathbf{y} \rangle
\end{aligned}
$$

Phép biến đổi trên cho thấy rằng độ lỗi chiếu cũng chính là độ sai khác giữa phương sai của dữ liệu ban đầu và phương sai của dữ liệu chiếu, hay có thể gọi là lượng phương sai mất mát khi chiếu dữ liệu như trong hình 2. Do vậy, có thể kết luận được rằng bài toán minimize độ lỗi chiếu cũng tương đương với bài toán maximize phương sai của dữ liệu chiếu.

5.4 Giải bài toán tối ưu theo hướng tiếp cận thống kê

Ta đã chứng minh sự tương đương của hai bài toán tối ưu. Từ phần này, ta quay lại phát triển bài toán dưới góc nhìn thống kê đa biến: maximize phương sai của dữ liệu chiếu.

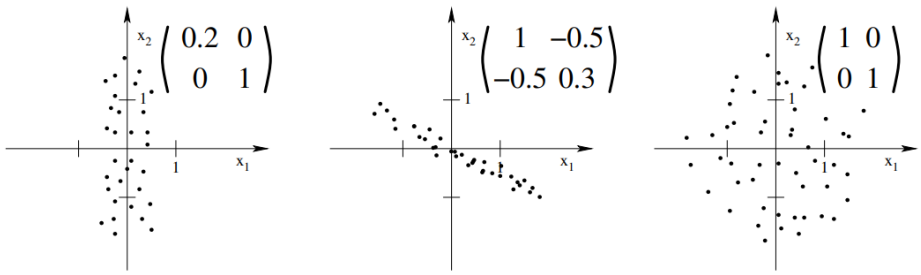
5.4.1 Ma trận hiệp phương sai

Để đơn giản, ta xét phép chiếu trong hệ tọa độ từ không gian 2 chiều lên một không gian con 1 chiều, tức là tìm một trục (được xác định bởi 1 vector) trong hệ tọa độ ban đầu sao cho phương sai của dữ liệu chiếu trên trục đó là lớn nhất.

Khi đó, ta viết $\mathbf{x} = (x_1, x_2)^T$. Phương sai của thành phần thứ nhất và thứ hai lần lượt là $\text{var}(x_1) = \langle x_1 x_1 \rangle$ và $\text{var}(x_1) = \langle x_2 x_2 \rangle$ (nhắc lại, giả định tập dữ liệu có giá trị trung bình 0 và kí hiệu $\langle . \rangle$ để chỉ phép tính giá trị trung bình khi xét trên toàn tập dữ liệu).

Trong ví dụ này, nếu $\text{var}(x_1)$ lớn hơn so với $\text{var}(x_1)$ thì hướng của trục chiếu "gần" với $(1, 0)^T$ (tức thành phần thứ hai) hơn (ví dụ 1 và 2, hình 4).

Tuy nhiên, nếu C_{11} và C_{22} là như nhau (ví dụ 3, hình 4) thì lúc này phương sai của các thành phần 1 và 2 không còn đủ hữu ích để lựa chọn trục chiếu. Đây là lúc cần đến một khái niệm gọi là **hiệp**



Hình 4: Ma trận hiệp phương sai

Nguồn: Lecture Notes on PCA, Laurenz Wiskott [4]

phương sai. Hiệp phương sai giữa hai thành phần i và j được biểu diễn trong ma trận hiệp phương sai như sau

$$C_{ij} := \langle x_i x_j \rangle.$$

Giá trị C_{12} dương và lớn thể hiện sự tương quan thuận mạnh giữa thành hai thành phần x_1 và x_2. Ngược lại, C_{12} âm và lớn thể hiện sự tương quan nghịch mạnh giữa thành hai thành phần x_1 và x_2. Giá trị C_{12} gần 0 thể hiện tương quan thấp hoặc không tương quan giữa hai thành phần.
Ta biểu diễn ma trận hiệp phương sai dưới dạng vector

$$C_x := \langle \mathbf{x}\mathbf{x}^T \rangle = \frac{1}{M} \sum_{\mu} x^{\mu} x^{\mu T}.$$

Chú ý: Ma trận hiệp phương sai là ma trận đối xứng, nghĩa là $C_x^T = C_x$.

5.4.2 Phân tích riêng của ma trận PSD

Một ma trận vuông $\mathbf{M}$ được gọi là đối xứng nếu $\mathbf{M}^T = \mathbf{M}$.
Ma trận hiệp phương sai không chỉ là một ma trận đối xứng, mà còn là một **ma trận đối xứng nửa xác định dương** (PSD, viết tắt cho positive semi-definite), nghĩa là luôn có thể phân tích thành tích $\mathbf{U}\Lambda\mathbf{U}^T$. Ma trận $\mathbf{U}$ được gọi là ma trận xoay, gồm các cột là các vector riêng của ma trận PSD. Ma trận Λ là ma trận đường chéo với các phần tử trên đường chéo là các giá trị riêng tương ứng với các vector riêng của ma trận PSD. Phép phân tích trên được gọi là một phép **phân**

tích riêng (eigenvalue decomposition) của ma trận đối xứng nửa xác định dương $\mathbf{M}$

$$\mathbf{M} = \mathbf{U}\Lambda\mathbf{U}^T.$$

Phép phân tích riêng được thực hiện trên ma trận hiệp phương sai để phân tích thành phần chính

$$C_x = \mathbf{U}\Lambda\mathbf{U}^T.$$

Kết quả thu được sau khi phân tích bao gồm ma trận $\mathbf{U}$ chứa các **thành phần chính**; ma trận Λ chứa **phương sai giải thích** bởi các thành phần chính.

5.4.3 Mối quan hệ giữa tổng phương sai và tổng trị riêng

Nhắc lại một số tính chất của hàm *trace*, kí hiệu $tr()$:

- Với mọi scalar $s : s = tr(s)$.
- Với mọi ma trận $\mathbf{A}, \mathbf{B} : tr(\mathbf{AB}) = tr(\mathbf{BA})$.
- Với mọi ma trận $\mathbf{A}, \mathbf{B}, \mathbf{C} : tr(\mathbf{ABC}) = tr(\mathbf{BCA})$.

Ta xét phương sai

$$\begin{aligned}
\langle \mathbf{x}^T\mathbf{x} \rangle &= \langle tr(\mathbf{x}^T\mathbf{x}) \rangle \\
&= \langle tr(\mathbf{x}\mathbf{x}^T) \rangle \\
&= tr(\langle \mathbf{x}\mathbf{x}^T \rangle) \\
&= tr(\mathbf{C}_\mathbf{x}) \\
&= tr(\mathbf{U}\mathbf{U}^T\mathbf{C}_x) \\
&= tr(\mathbf{U}^T\mathbf{C}_x\mathbf{U}) \\
&= tr(\Lambda) \\
&= \sum_i \lambda_i
\end{aligned}$$

Do đó, tổng phương sai của dữ liệu bằng tổng các trị riêng của ma trận hiệp phương sai.

5.4.4 Xoay hệ tọa độ với ma trận U

Ta sử dụng ma trận xoay $\mathbf{U}$ để ánh xạ dữ liệu từ hệ tọa độ ban đầu I chiều vào hệ tọa độ mới I chiều xác định bởi các vector cơ sở là các vector riêng trong $\mathbf{U}$.
Dữ liệu sau khi được biến đổi bởi ma trận xoay $\mathbf{U}$, kí hiệu $\mathbf{x}' := \mathbf{U}^T\mathbf{x}$ có ma trận hiệp phương sai

$$\begin{aligned}\mathbf{C}'_x &:= \langle \mathbf{x}'\mathbf{x}'^T \rangle \\ &= \langle (\mathbf{U}^T\mathbf{x})(\mathbf{U}^T\mathbf{x})^T) \rangle \\ &= \mathbf{U}^T \langle \mathbf{x}\mathbf{x}^T \rangle \mathbf{U} \\ &= \mathbf{U}^T \mathbf{C}_x \mathbf{U} \\ &= \Lambda.\end{aligned}$$

Nhận xét: Ma trận hiệp phương sai của dữ liệu sau khi ánh xạ là một **ma trận đường chéo**. Với phép chéo hóa này, phương sai của dữ liệu được quan sát dễ dàng qua ma trận hiệp phương sai, với mỗi phần tử trên đường chéo thể hiện phương sai giải thích bởi các thành phần chính.

5.4.5 Giảm chiều với ma trận V′

Sau khi xoay, ta thực hiện giảm chiều, tức chiếu $\mathbf{x}'$ lên P chiều xác định bởi P vector riêng trong hệ tọa độ mới (đương nhiên, khi giảm chiều $P < I$). Xét tập P vector riêng (trực giao) bất kì $\mathbf{v}'_p$ với $\mathbf{V}' := (\mathbf{v}'_1, \mathbf{v}'_2, ..., \mathbf{v}'_p)$ ta biến đổi $\mathbf{y} := \mathbf{V}'^T\mathbf{x}'$. Khi đó phương sai của $\mathbf{y}$ là

$$\begin{aligned}\langle y^T y \rangle &= \langle x'^T V' V'^T x' \rangle \\ &= \langle tr(x'^T V' V'^T x') \rangle \\ &= \langle tr(V'^T x' x'^T V') \rangle \\ &= tr(V'^T C'_x V') \\ &= tr(V'^T \Lambda V') \\ &= \sum_i \lambda_i \sum_p (v'_{ip})^2.\end{aligned}$$

Mặt khác, vì $\mathbf{V}'$ là một ma trận trực giao nên ta có:

- Các vector cột của $\mathbf{V}'$ có norm bằng 1

$$\sum_i (v'_{ip})^2 = 1.$$

- Tổng bình phương các phần tử của $\mathbf{V}'$ bằng P

$$\sum_{ip} (v'_{ip})^2 = P.$$

- Các vector dòng của $\mathbf{V}'$ có norm nhỏ hơn hoặc bằng 1 (do giảm chiều)

$$\sum_p (v'_{ip})^2 \leq 1.$$

Từ các ràng buộc trên, cùng với $\langle y^T y \rangle = \sum_i \lambda_i \sum_p (v'_{ip})^2$, có thể thấy: ***Với số chiều P cho trước, để maximize phương sai của dữ liệu chiếu $\mathbf{y}$, ta cần chọn không gian con P chiều xác định bởi P thành phần chính sao cho tổng trị riêng λ_i ứng với các thành phần này là lớn nhất.***

5.4.6 Tổng hợp các phép biến đổi

Quá trình phân tích thành phần chính trên đã đi qua các bước:

- Xoay hệ tọa độ với ma trận $\mathbf{U}$ tức $\mathbf{x}' = \mathbf{U}^T\mathbf{x}$.
- Giảm chiều với ma trận $\mathbf{V}'$ tức $\mathbf{y} = \mathbf{V}'^T\mathbf{x}'$.

Đặt $\mathbf{V} = \mathbf{U}\mathbf{V}'$. Khi đó, các bước trên có thể tóm gọn lại thành một phép ánh xạ $\mathbf{V}$ duy nhất từ hệ tọa độ ban đầu I chiều vào hệ tọa độ mới P chiều ứng với P vector riêng có trị riêng lớn nhất:

$$\begin{aligned}\mathbf{y} &= \mathbf{V}'^T\mathbf{x}' \\ &= \mathbf{V}'^T\mathbf{U}^T\mathbf{x}' \\ &= \mathbf{V}^T\mathbf{x}.\end{aligned}$$

Và như đã chứng minh, **y** có phương sai lớn nhất.

6 Thuật toán

Bước 1: Chuẩn hóa dữ liệu

Center dữ liệu bằng cách trừ đi giá trị trung bình để giá trị trung bình của tập dữ liệu bằng 0.

Bước 2: Tính ma trận hiệp phương sai

$$C_x = \langle \mathbf{x}\mathbf{x}^T \rangle = \frac{1}{M} \sum_{\mu} x^{\mu} x^{\mu T}.$$

Bước 3: Phân tích riêng (eigenvalue decomposition) ma trận hiệp phương sai

$$C_x = \mathbf{U}\Lambda\mathbf{U}^T.$$

Bước 4: Sắp xếp các vector riêng theo thứ tự giảm dần trị riêng

Bước 5: Chọn số lượng thành phần chính

Các thành phần chính được chọn nên giải thích được ít nhất 80% phương sai của dữ liệu. Để có thể chọn số lượng thành phần chính phù hợp, có thể sử dụng scree plot để xem lượng phương sai giải thích bởi mỗi thành phần chính; hoặc sử dụng biểu đồ thể hiện tỉ lệ phương sai tích lũy.

Bước 6: Chiếu dữ liệu

Ánh xạ dữ liệu từ không gian nhiều chiều trong hệ tọa độ ban đầu vào không gian ít chiều trong hệ tọa độ mới với ma trận biến đổi

$$\mathbf{V} = \mathbf{U}\mathbf{V}'.$$

7 Cài đặt

7.1 Chuẩn bị dữ liệu

Tập dữ liệu Phần cài đặt PCA được thực hiện trên tập dữ liệu hoa Iris.

Bộ dữ liệu Iris là bộ dữ liệu cổ điển trong lĩnh vực học máy và thống kê. Nó được nhà thống kê và nhà sinh vật học người Anh Ronald Fisher giới thiệu trong bài báo năm 1936 "Việc sử dụng nhiều phép đo trong các vấn đề phân loại". Bộ dữ liệu bao gồm 150 mẫu hoa diên vĩ (iris), mỗi mẫu có bốn đặc điểm (chiều dài đài hoa, chiều rộng đài hoa, chiều dài cánh hoa và chiều rộng cánh hoa) được đo bằng centimet. Mỗi mẫu được dán nhãn bằng một trong ba loài hoa diên vĩ: Setosa,

Versicolor và Virginica.

Ta thực hiện PCA trên tập 4 thuộc tính của bộ dữ liệu Iris:

```
X = iris.data
```

7.2 Bước 1: Chuẩn hóa dữ liệu

Ta có thể sử dụng hàm có sẵn của thư viện `sklearn` để thực hiện chuẩn hóa dữ liệu về giá trị trung bình bằng 0 và phương sai bằng 1:

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

Hoặc cũng có thể tự định nghĩa hàm chuẩn hóa như sau:

```
def standard_scaler(X):
    mean = np.mean(X, axis=0)
    std_dev = np.std(X, axis=0)
    X_scaled = (X - mean) / std_dev
    return X_scaled
```

7.3 Bước 2: Tính ma trận hiệp phương sai

Ta có thể sử dụng hàm có sẵn của thư viện `numpy` để tính ma trận hiệp phương sai:

```
covariance_matrix = np.cov(X_scaled.T)
```

Hoặc cũng có thể tự định nghĩa hàm tính ma trận hiệp phương sai như sau:

```
def cov(X):
    n_samples = X.shape[0]
    covariance_matrix = (X.T @ X) / (n_samples - 1)
    return covariance_matrix
```

7.4 Bước 3: Phân tích riêng ma trận hiệp phương sai

Để đơn giản, ta sử dụng hàm của thư viện `numpy` để tìm vector riêng và trị riêng của ma trận hiệp phương sai:

```
eigenvalues, eigenvectors = np.linalg.eig(covariance_matrix)
```

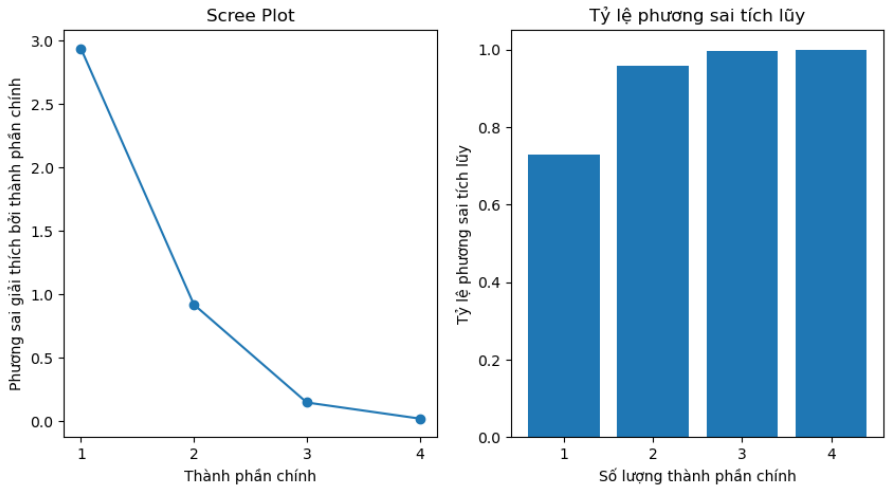
7.5 Bước 4: Sắp xếp các vector riêng theo thứ tự giảm dần trị riêng

Ta xếp các vector riêng (các thành phần chính) theo thứ tự trị riêng (phương sai giải thích) từ cao xuống thấp:

```
    sorted_indices = np.argsort(eigenvalues)[::-1]
    sorted_eigenvalues = eigenvalues[sorted_indices]
    sorted_eigenvectors = eigenvectors[:, sorted_indices]
```

7.6 Bước 5: Chọn số lượng thành phần chính

Các thành phần chính được chọn nên giải thích được ít nhất 80% phương sai của dữ liệu. Để có thể chọn số lượng thành phần chính phù hợp, có thể sử dụng scree plot để xem lượng phương sai giải thích bởi mỗi thành phần chính; hoặc sử dụng biểu đồ thể hiện tỉ lệ phương sai tích lũy. 5



Hình 5: Scree plot và biểu đồ tỷ lệ phương sai tích lũy

7.7 Bước 6: Chiếu dữ liệu

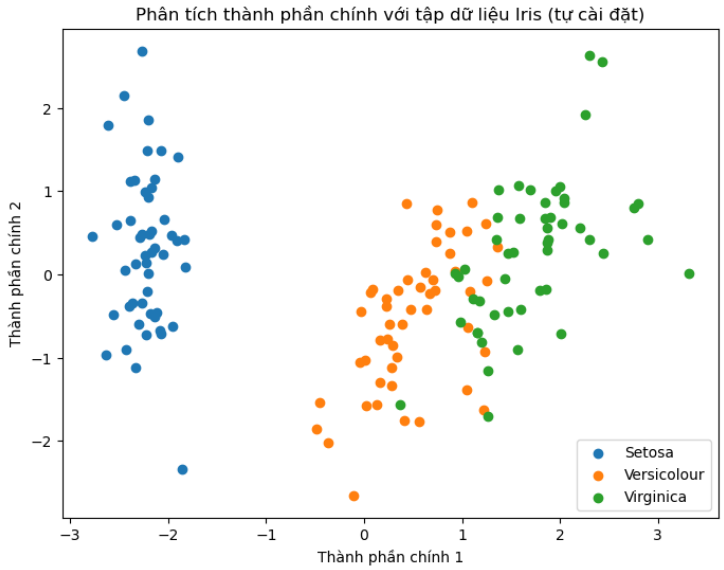
Ánh xạ dữ liệu từ không gian nhiều chiều trong hệ tọa độ ban đầu vào không gian ít chiều trong hệ tọa độ mới với ma trận biến đổi xác định bởi các thành phần chính đã chọn:

```
    X_pca = X_scaled.dot(top_eigenvectors)
```

7.8 Sử dụng kết quả phân tích

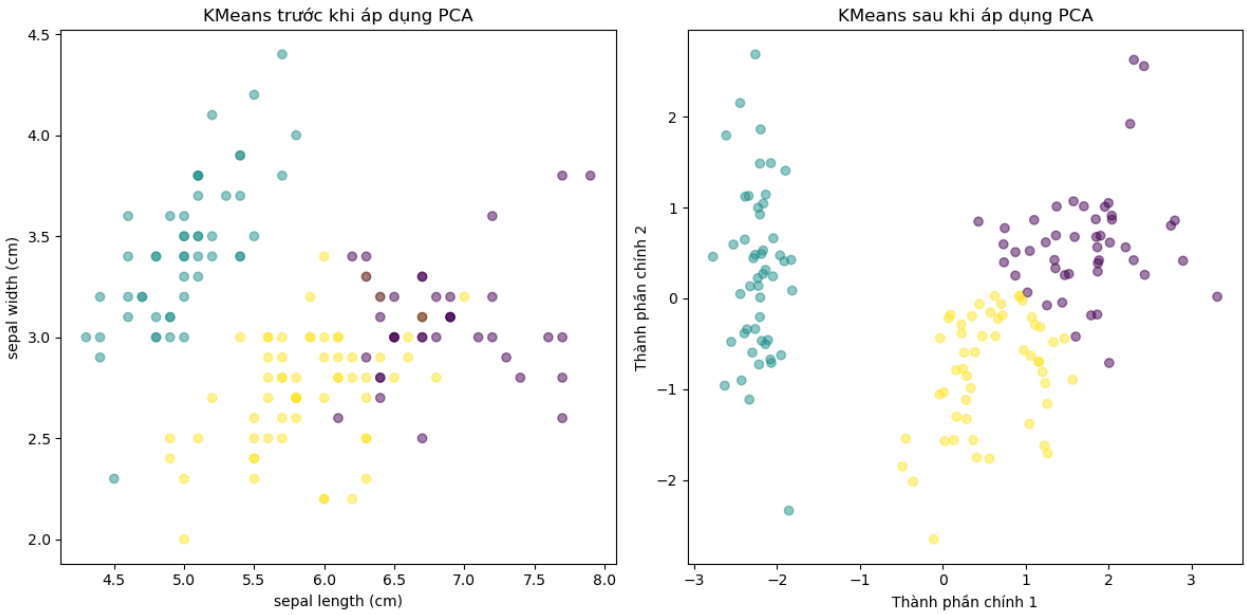
Ta có thể sử dụng kết quả sau khi thực hiện giảm chiều bằng PCA cho các mục đích trực quan hoặc tính toán.

Sử dụng PCA cho mục đích trực quan:



Hình 6: Trực quan dữ liệu Iris trong không gian 2 chiều

Sử dụng PCA cho mục đích tính toán:



Hình 7: Áp dụng kỹ thuật gom cụm KMeans trong không gian 2 chiều

8 Nhận xét về PCA

PCA là một phương pháp giảm chiều đơn giản và dễ tính toán. Một ưu điểm, tuy nhiên đồng thời cũng là nhược điểm của PCA đó là nó **không có tham số** để điều chỉnh và nghiệm của bài toán tối ưu là duy nhất.

Một ưu điểm của PCA là nó **bảo toàn cấu trúc toàn cục** của dữ liệu. Điều này là quan trọng để áp dụng cho các mục đích tính toán sau này, ví dụ như trong học máy. Một số kỹ thuật giảm chiều như t-SNE chỉ bảo toàn cấu trúc cục bộ của dữ liệu, chỉ sử dụng cho mục đích trực quan, khám phá dữ liệu.

Một nhược điểm khác của PCA là không phù hợp cho dữ liệu **phi tuyến**. Để giảm chiều cho dữ liệu có cấu trúc phi tuyến, ta có thể sử dụng Kernel PCA. Khác với PCA, Kernel PCA là một thuật toán có tham số và có chi phí tính toán cao [10]. Ngoài ra còn có nhiều kỹ thuật giảm chiều khác có thể áp dụng cho dữ liệu có cấu trúc phi tuyến.

Với trường hợp các thành phần chính không **trực giao** hoặc dữ liệu không tuân theo **phân phối Gauss** như trong giả định. Lúc này PCA không phù hợp để giảm chiều. Một kỹ thuật áp dụng cho trường hợp này là Independent Component Analysis (ICA). Tuy nhiên, điểm yếu của ICA đó là tính toán phức tạp vì bài toán tối ưu của nó có dạng phi tuyến. [10]

9 Tự đánh giá

Dưới đây là bảng tự đánh giá mức độ hoàn thành theo các yêu chí của bài tập ILab 01: Principle Components Analysis Visualization.

Yêu cầu	Trạng thái
Nghiên cứu về PCA (45%)	Hoàn thành
Cài đặt PCA (45%)	Hoàn thành
Hiểu tổng quan mã nguồn đã nộp (10%)	Hoàn thành
Điểm cộng: Giải thích toán & Demo tính toán số (10%)	Hoàn thành

Tài liệu

[1] Rasmus Bro and Age K Smilde. Principal component analysis. *Analytical methods*, 6(9):2812–2831, 2014.

[2] M.P. Deisenroth, A.A. Faisal, and C.S. Ong. *Mathematics for Machine Learning*. Cambridge University Press, 2020.

[3] Michael Greenacre, Patrick JF Groenen, Trevor Hastie, Alfonso Iodice d'Enza, Angelos Markos, and Elena Tuzhilina. Principal component analysis. *Nature Reviews Methods Primers*, 2(1):100, 2022.

[4] Laurenz Wiskott. Lecture Notes on Principal component analysis (Stanford University), 2004. Accessed: 05/05/2024.

[5] Nathaniel E. Helwig. Principal component analysis Lecture notes (University of Minnesota), 2017. Accessed: 05/05/2024.

[6] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572, 1901.

[7] Prof. Michale Fee. Lecture Notes on Principal component analysis (Massachusetts Institute of Technology), 2017. Accessed: 05/05/2024.

[8] Sagor Saha. PCA for Visualization and Dimension Reduction, ongoing. Accessed: 05/05/2024.

[9] Cosma Shalizi. Advanced data analysis from an elementary point of view. 2013.

[10] Jonathon Shlens. A tutorial on principal component analysis. 2014.