Compressing Data via Dimensionality Reduction

Nguyễn Quốc Bảo - 18110053
Nguyễn Minh Hoàng - 18110095

Faculty of Mathematics and Computer Science
University of Science

Table of Contents

Table of Contents

Vì sao phải giảm số chiều dữ liệu ?

Khó khăn:

- Khó khăn về tính toán, thời gian thực thi
- Hạn chế về không gian lưu trữ
- Các tính chất của dữ liệu trong số chiều nhỏ có thể không còn đúng trong trường hợp số chiều lớn, dẫn đến việc phân tích dữ liệu khó khăn

Vì sao phải giảm số chiều ?

Khó khăn:

- Việc chiếu dữ liệu từ không gian có số chiều lớn xuống không gian có số chiều thấp dẫn đến các thông tin quan trọng của dữ liệu bị mất một phần (hoặc toàn bộ) (Giải quyết trong phương pháp PCA và LDA)
- Việc giảm chiều của dữ liệu có thể làm cho bài toán ban đầu có thể dẫn đến không giải quyết được

Vì sao phải giảm số chiều ?

Thuận lợi:

- Việc chiếu dữ liệu xuống không gian thấp hơn có thể dẫn đến bài toán trở nên dễ dàng hơn
- Thời gian tính toán nhanh chóng, cho ra kết quả nhanh hơn
- Tiết kiệm bộ nhớ lưu trữ dữ liệu

Vì sao phải giảm số chiều ?

Ý tưởng chung của các thuật toán sẽ giới thiệu:
Cho vectơ $\boldsymbol{X} \in \mathbb{R}^n$, ta cần chuyển dữ liệu về vectơ $\boldsymbol{Y} \in \mathbb{R}^m$ với $m << n$
Xây dựng ma trận chiếu $\boldsymbol{W} \in \mathbb{R}^{m \times n}$
Khi đó: $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{W} \in \mathbb{R}^m$

Table of Contents

Mở đầu về PCA

PCA là phương pháp giúp xác định mẫu (pattern) của dữ liệu, truyền tải dữ liệu thông qua việc làm nổi bật các đặc trưng giống và khác nhau của chúng.
Vì việc xác định mẫu hoặc mô hình của dữ liệu với số lượng chiều lớn thông qua các phương pháp trực quan băng đồ họa thông thường sẽ rất khó khăn, do vậy PCA sẽ là công cụ hữu ích giúp chúng ta giải quyết vấn đề này.

Thuật toán PCA

Các bước của thuật toán PCA:

- Chuẩn hóa dữ liệu (d-chiều)
- Xây dựng ma trận hiệp phương sai từ dữ liệu đã được chuẩn hoá
- Tìm các vectơ riêng, trị riêng của ma trận hiệp phương sai
- Xây dựng tập trực chuẩn từ tập hợp các vecto trên
- Sắp xếp các trị riêng theo chiều giảm dần
- Chọn k vectơ riêng đầu tiên ứng với k trị riêng đầu tiên trong bộ trị riêng có thứ tự ở bước trên ($k < d$)
- Xây dựng ma trận chiếu $\boldsymbol{W}$ từ các vectơ riêng được chọn.
- Tìm hình chiếu của dữ liệu đã chuẩn hóa trong không gian mới sinh bởi ma trận $\boldsymbol{W}$.

Demo thuật toán PCA

Cho tập dữ liệu sau:

	f1	f2	f3	f4
0	1	2	3	4
1	5	5	6	7
2	1	4	2	3
3	5	3	2	1
4	8	1	2	2

Demo thuật toán PCA

Bước 1: Chuẩn hóa dữ liệu qua phép biến đổi

$$x_{new} = \frac{x - \mu}{\sigma}$$

Demo thuật toán PCA

Ta tính μ, σ của từng cột :

	f1	f2	f3	f4
$\mu =$	4	3	3	3.4
$\sigma =$	3	1.58	1.73	2.30

Sau đó ta thực hiện chuẩn hóa ra được kết quả như sau :

f1	f2	f3	f4
-1	-0.632	0	0.26
0.33	1.26	1.73	1.56
-1	0.63	-0.57	-0.173
0.33	0	-0.57	-1.04
1.33	-1.26	-0.57	-0.6

Demo thuật toán PCA

Bước 2: Xây dựng ma trận hiệp phương sai trên mẫu đã chuẩn hóa

Ma trận hiệp phương sai của $X_1, X_2, ..X_n$ có công thức là :

$$\begin{bmatrix} Cov(X_1^2) & \cdots & Cov(X_1X_n) \\ \vdots & & \\ Cov(X_1X_n) & \cdots & Cov(X_n^2) \end{bmatrix}$$

Trong đó Covariance được tính bằng :

$$\text{Cov(X,Y)} = \frac{1}{n-1}\sum(x_i - \bar{x})(y_i - \bar{y})$$

Demo thuật toán PCA

Áp dụng công thức trên ma trận hiệp phương sai có dạng như sau:

$$\begin{bmatrix} Cov(f1,f1) & Cov(f1,f2) & Cov(f1,f3) & Cov(f1,f4) \\ Cov(f2,f1) & Cov(f2,f2) & Cov(f2,f3) & Cov(f2,f4) \\ Cov(f3,f1) & Cov(f3,f2) & Cov(f3,f3) & Cov(f1,f4) \\ Cov(f4,f1) & Cov(f4,f2) & Cov(f4,f3) & Cov(f4,f4) \end{bmatrix}$$

$$\text{Cov(f1,f1)} = \frac{1}{n-1}\sum(f_{1i} - \bar{f_1})(f_{1i} - \bar{f_1}) = 0.8$$

$$\text{Cov(f1,f2)} = \frac{1}{n-1}\sum(f_{1i} - \bar{f_1})(f_{2i} - \bar{f_2}) = -0.25$$

Demo thuật toán PCA

Ta ra được ma trận hiệp phương sai như sau :

	f1	f2	f3	f4
f1	0.8	-0.25	0.03	-0.144
f2	-0.25	0.8	0.51	0.49
f3	0.038	0.51	0.8	0.75
f4	-0.144	0.49	0.75	0.8

Bước 3: Tìm các vectơ riêng, trị riêng của ma trận hiệp phương sai mẫu

Ta có công thức tính trị riêng và vecto riêng như sau :

$$\boxed{\Sigma v = \lambda v} \tag{1}$$

trong đó :

Σ là ma trận hiệp phương sai

v là vecto riêng

λ là trị riêng

Demo thuật toán PCA

Trước tiên ta sẽ tính trị riêng :
$\Sigma - \lambda I$ có dạng như sau :

	f1	f2	f3	f4
f1	0.8 - λ	-0.25	0.03	-0.144
f2	-0.25	0.8 - λ	0.51	0.49
f3	0.038	0.51	0.8 - λ	0.75
f4	-0.144	0.49	0.75	0.8 - λ

Giải phương trình $\det(\Sigma - \lambda I) = 0$ ta tính được trị riêng :

$$\lambda = \begin{bmatrix} 2.51 \\ 1.06 \\ 0.39 \\ 0.02 \end{bmatrix}$$

Demo thuật toán PCA

Tiếp theo ta sẽ tính vecto riêng bằng cách giải:

$$(\Sigma - \lambda I)v = 0 \tag{2}$$

$$\begin{bmatrix} 0.8-\lambda & -0.25 & 0.038 & -0.144 \\ -0.25 & 0.8-\lambda & 0.51 & 0.49 \\ 0.03 & 0.511 & 0.8-\lambda & 0.75 \\ -0.144 & 0.494 & 0.75 & 0.8-\lambda \end{bmatrix} \begin{bmatrix} v1 \\ v2 \\ v3 \\ v4 \end{bmatrix} = 0$$

Demo thuật toán PCA

Với $\lambda = 2.51$, giải phương trình bên trên áp dụng hệ thức Crammer, giá trị của vecto riệng là :

$$v = \begin{bmatrix} 0.161 \\ -0.52 \\ -0.58 \\ -0.59 \end{bmatrix}$$

Áp dụng tương tự như trên cho các giá trị λ còn lại ta được vecto riêng tương ứng với tất cả trị riêng là :

e_1	e_2	e_3	e_4
0.161	-0.91	-0.3	0.196
-0.52	0.2	-0.81	0.12
-0.58	-0.32	0.18	0.449
-0.59	0.11	0.449	0.654

Demo thuật toán PCA

Bước 5: Sắp xếp các trị riêng theo chiều giảm dần
Bên trên các trị riêng đã được sắp xếp nên ta không cần thay đổi gì.
Bước 6 : Chọn k vectơ riêng đầu tiên ứng với k trị riêng đầu tiên trong bộ trị riêng có thứ tự ở bước trên ($k < d$)
Bước 7 : Xây dựng ma trận chiếu W từ các vectơ riêng được chọn.
Ta chọn 2 cột đầu của các trị riêng.

e_1	e_2
0.161	-0.91
-0.52	0.2
-0.58	-0.32
-0.59	0.11

Demo thuật toán PCA

Bước 8 :

Tìm hình chiếu của dữ liệu đã chuẩn hóa trong không gian mới sinh bởi ma trận W

Áp dụng công thức :

$$\boxed{Y = XW} \tag{3}$$

$$\begin{bmatrix} -1 & -0.63 & 0 & 0.26 \\ 0.33 & 1.2 & 1.3 & 1.5 \\ -1 & 0.63 & -0.57 & -0.17 \\ 0.33 & 0 & -0.57 & -1.04 \\ 1.33 & -1.26 & -0.57 & -0.6 \end{bmatrix} \begin{bmatrix} 0.16 & -0.91 \\ -0.52 & 0.2 \\ -0.58 & -0.32 \\ -0.59 & 0.11 \end{bmatrix} = \begin{bmatrix} 0.01 & 0.25 \\ -2.55 & -0.78 \\ -0.05 & 1.25 \\ 1.04 & 0.01 \\ 1.5 & -1.2 \end{bmatrix}$$

Demo thuật toán PCA - Python

Bắt đầu phần code

https://github.com/hoangchiro0210/Py4DS/tree/main/Lab08-GK