

線性代數初探：從影像前後景分離學起

Introduction to Linear Algebra via Foreground-Background Separation

Cheng-Fang Su 蘇承芳

Contents

1	Fundamentals of Linear Algebra	1
1.1	從影像到矩陣表示	1
1.1.1	矩陣的定義與符號	1
1.1.2	矩陣與影像的關係	2
1.1.3	影像轉為矩陣的實作	4
1.2	矩陣加法與純量乘法	7
1.2.1	矩陣加法	7
1.2.2	矩陣加法在影像處理中的應用	8
1.2.3	矩陣的純量乘法	9
1.2.4	純量乘法在影像處理中的應用	10
1.3	矩陣乘法	11
1.3.1	矩陣乘法的定義	11
1.3.2	矩陣與向量相乘	12
1.3.3	矩陣乘法的性質	15
1.3.4	單位矩陣與零矩陣	17
1.4	向量的線性組合	18
1.4.1	線性組合的數學形式與操作	19
1.4.2	線性組合在影像處理中的應用	21
1.4.3	向量空間的定義	22
1.4.4	矩陣空間	24
1.4.5	向量空間的維度	25
1.4.6	子空間及其判斷條件	26
1.4.7	子空間與影像處理	28

2	An Introduction to Matrix Analysis	30
2.1	線性相依與線性獨立	30
2.1.1	線性相依與線性獨立的定義	31
2.1.2	線性相依與影像資料	33
2.2	基底與維度：如何描述影像資料的方向性	34
2.2.1	什麼是張成？從向量組合看空間的生成	35
2.2.2	什麼是基底與維度？	36
2.2.3	在影像資料中理解維度	37
2.3	什麼是秩？從直觀談起	39
2.3.1	秩的直觀意義	39
2.3.2	秩為 1 的幾何觀點與實作	40
2.3.3	向量的 Outer Product	42
2.4	矩陣的秩	44
2.4.1	矩陣秩的定義與計算方式	45
2.4.2	低秩背景與稀疏前景模型	47
2.4.3	低秩矩陣的意義	50
2.4.4	稀疏矩陣觀念介紹	52

Chapter 1

Fundamentals of Linear Algebra

線性代數 (Linear Algebra) 是數學中最基本也最具廣泛應用性的課程之一，無論是科學計算、最佳化理論、工程應用、機器學習或量子科技，它都為這些領域提供了極其強大的數學工具。在本門課中，第一章將從一個貼近現實世界的資料形態出發，也就是我們生活中隨處可見的影像，透過一些具體的例子，引導同學們逐步建構出矩陣、矩陣運算、向量結構與子空間等基本概念，為後續的廣泛應用打下良好的數學基礎。

1.1 從影像到矩陣表示

在現代資訊科學與工程技術中，影像 (Image) 已成為數據資料中極為重要的一種形式。舉凡醫學影像、衛星監測、人工智慧、電腦視覺，甚至社交媒體上的圖片分享，我們日常所接觸到的圖像訊息無不以數位影像的形式儲存與傳遞。然而，要對影像進行處理、分析與理解，就必須先將其轉化為數學物件加以研究，而矩陣 (Matrix) 正是描述影像最直接也最自然的數學模型。首先，我們期待學生能夠理解一張影像如何轉換為矩陣，進而開啟後續使用線性代數工具進行影像壓縮、降噪、特徵提取等應用的基礎能力。

在我們開始處理影像之前，為了進行數學處理，首先需要一種能夠系統地儲存數值的方法；在線性代數中，這樣的表示方式就是「矩陣」。

1.1.1 矩陣的定義與符號

首先，我們必須瞭解什麼是「矩陣」，本節將從矩陣的基本定義開始，逐步建立起與影像資料的對應關係，並延伸至數位影像的數學模型表示。

Definition 1.1 (矩陣, Matrix). 令 $m, n \in \mathbb{N}$ 。一個 $m \times n$ 的實數矩陣是一個由 m 列 n 行的實數所構成的二維數表，記為

$$A = [a_{ij}] \in \mathbb{R}^{m \times n},$$

其中 $a_{ij} \in \mathbb{R}$ 表示矩陣 A 的第 i 列第 j 行的元素， $1 \leq i \leq m, 1 \leq j \leq n$ 。

舉例而言，考慮以下矩陣

$$A = \begin{bmatrix} 2 & -1 & 0 \\ 4 & 3 & 5 \end{bmatrix} \in \mathbb{R}^{2 \times 3},$$

顯然矩陣 A 有 2 列 3 行，屬於 2×3 矩陣。矩陣的元素值分別為

$$a_{11} = 2, \quad a_{12} = -1, \quad a_{13} = 0, \quad a_{21} = 4, \quad a_{22} = 3, \quad a_{23} = 5.$$

★ 數學符號複習：

(1) $\mathbb{N}$ ：自然數集、正整數集； (2) $\mathbb{R}$ ：實數集； (3) $\in$ ：屬於。

(4) $A \in \mathbb{R}^{2 \times 3}$ 意即「 A 為 2×3 的實數矩陣」； (5) $a_{ij} \in \mathbb{R}$ 意即「 a_{ij} 為一個實數」。

Exercise 1.2. 給定矩陣：

$$B = \begin{bmatrix} 7 & 8 \\ -2 & 0 \\ 5 & 3 \end{bmatrix},$$

請回答下列問題：

(1) 若 B 是 $m \times n$ 矩陣，請問數對 (m, n) 為何？

(2) 試寫出元素 b_{31} 與 b_{22} 的值。

1.1.2 矩陣與影像的關係

有了矩陣的概念，我們可以介紹影像的數位結構。一張影像就是一個由許多像素 (Pixel) 組成的數值網格，每一格的數值就代表其亮度或顏色資訊；這恰好與矩陣的結構完全對應，所以我們可以利用矩陣的形式來表達一張影像的資訊，在做任何影像處理之前，都需要先將一張影像轉換成矩陣後再操作。

由於數位影像本質上是一個由像素組成的數值陣列，而每個像素值都可以用一個實數來表示。因此，我們可以自然地將一張影像表示為矩陣，通常我們會考慮以下兩種不同的影像：

- (1) 一張灰階影像 (Grayscale Image) 以矩陣 $I \in \mathbb{R}^{m \times n}$ 表示，其中 I_{ij} 為像素亮度值 (如 0-255)，代表第 i 列第 j 行的像素。例如我們給定以下矩陣

$$A = \begin{bmatrix} 2 & 1 & 0 \\ 4 & 3 & 5 \end{bmatrix} \in \mathbb{R}^{2 \times 3},$$

若將其視為一張灰階影像所對應的像素矩陣，每個元素可對應為一個像素的灰階值。通常灰階影像的像素值範圍為 $[0, 255]$ ，因此矩陣 A 可視為一張 2×3 像素的灰階影像。

- (2) 一張彩色影像 (Color Image) 通常表示為三個矩陣所構成的三維陣列，即

$$I \in \mathbb{R}^{m \times n \times 3}, \quad I = (R, G, B),$$

其中 $R, G, B \in \mathbb{R}^{m \times n}$ 分別代表紅色、綠色與藍色通道的亮度矩陣。這三個通道矩陣沿著第三維 (通道維) 堆疊，形成一個 $m \times n$ 的空間影像，在幾何上可以想像為一個具有三層的「矩陣長方體」或三維陣列 (3D array)。

注意到使用 I 做為矩陣的名稱，是取 image 第一個字母的大寫來表達，不要與後面會提到的單位矩陣 $\mathbb{I}$ 混淆。

Example 1.3. 考慮一張 2×3 的灰階影像，其對應矩陣如下：

$$I_{\text{gray}} = \begin{bmatrix} 100 & 150 & 200 \\ 50 & 75 & 125 \end{bmatrix},$$

以及一張 2×3 的彩色影像，其紅、綠、藍三個通道分別如下：

$$R = \begin{bmatrix} 255 & 128 & 64 \\ 0 & 0 & 32 \end{bmatrix}, \quad G = \begin{bmatrix} 100 & 150 & 100 \\ 90 & 100 & 80 \end{bmatrix}, \quad B = \begin{bmatrix} 20 & 30 & 40 \\ 50 & 60 & 70 \end{bmatrix}.$$

試問：

- (1) 灰階影像與彩色影像的矩陣形狀 (shape) 分別為何？
- (2) 在彩色影像中，像素位置 (2, 3) 的 RGB 值為何？

它的答案蠻明顯的，灰階影像為 $\mathbb{R}^{2 \times 3}$ ，彩色影像為 $\mathbb{R}^{2 \times 3 \times 3}$ ，且彩色影像第 2 列第 3 行的 RGB 值為：

$$R_{23} = 32, \quad G_{23} = 80, \quad B_{23} = 70,$$

所以第 2 列第 3 行的 RGB 值為 (32, 80, 70)。

□

然而，為什麼彩色影像需要三維陣列才能表示呢？因為此時影像的每個像素不再是一個實數，而是三個實數組成的向量 (R, G, B) ，因此必須為每一個像素額外多一個維度來容納這三個通道的值，最後就形成了三維陣列。

Exercise 1.4. 下列是某張 3×2 彩色影像的三個通道矩陣（僅列出部分影像的像素）：

$$R = \begin{bmatrix} 100 & 200 \\ 150 & 100 \\ 50 & 80 \end{bmatrix}, \quad G = \begin{bmatrix} 80 & 100 \\ 120 & 90 \\ 60 & 75 \end{bmatrix}, \quad B = \begin{bmatrix} 70 & 90 \\ 100 & 110 \\ 80 & 60 \end{bmatrix}.$$

試問：

- (1) 此影像的尺寸為何？
- (2) 請寫出第 1 列第 2 行的 RGB 值。

1.1.3 影像轉為矩陣的實作

假設我們要把灰階影像轉為矩陣，現在手邊有一張取樣自某張貓貓照片的 4×4 像素影像，每個像素紀錄的亮度值如下：

$$I = \begin{bmatrix} 0 & 64 & 128 & 255 \\ 0 & 64 & 128 & 255 \\ 0 & 64 & 128 & 255 \\ 0 & 64 & 128 & 255 \end{bmatrix}$$

這個矩陣說明了每行都呈現亮度由暗至亮（由左至右遞增），它也正是這張影像的灰階像素矩陣。我們也可以試著寫程式呈現它，程式及效果如圖 1.1 所示。

```
import numpy as np
import matplotlib.pyplot as plt

# 建立 4x4 灰階矩陣 (左暗右亮)
I = np.array([
    [ 0, 64, 128, 255],
    [ 0, 64, 128, 255],
    [ 0, 64, 128, 255],
    [ 0, 64, 128, 255]
], dtype=np.uint8)

# 顯示為灰階影像
plt.imshow(I, cmap='gray', vmin=0, vmax=255)
plt.title("模擬灰階影像 (左暗右亮)")
plt.axis('off')
plt.show()
```



(a) 模擬灰階影像圖（左暗右亮）

(b) 對應的程式碼片段

Figure 1.1: 灰階影像與其生成程式碼對照圖

接下來，我們嘗試將彩色影像做 RGB 三通道分離。同樣考慮貓貓影像 (example.jpg)，我們可寫程式分離出三個矩陣 $R, G, B \in \mathbb{R}^{m \times n}$ ，如圖 1.2 所示。

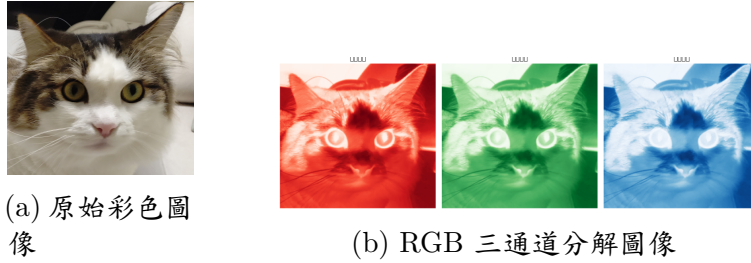


Figure 1.2: 原始影像與通道分解結果比較

更進一步地，我們可以讀取出彩色貓貓影像中的灰階影像並轉為矩陣形式，如圖 1.3 所示。

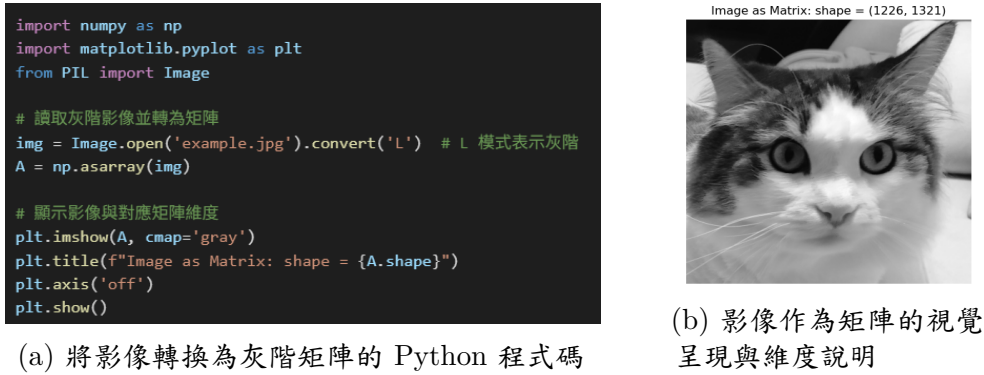


Figure 1.3: 從影像轉換為矩陣表示的程式與輸出結果

注意到圖 1.3 (b) 上方有一排文字：

Image as Matrix: shape = (1226, 1321).

它意謂著我們已把灰階影像轉成了一個二維數值矩陣 $A \in \mathbb{R}^{1226 \times 1321}$ ，也就是說，每個像素會被記成一個「灰階數值」，範圍通常在 0-255；影像中的每一個像素，對應矩陣中的一個元素，最終我們可以得到一個大小為 1226 列、1321 行的二階矩陣。以數學的方式來表達，這張影像現在對應到以下的矩陣：

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1,1321} \\ a_{21} & a_{22} & \cdots & a_{2,1321} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1226,1} & a_{1226,2} & \cdots & a_{1226,1321} \end{bmatrix},$$

其中每個 $a_{ij} \in [0, 255]$ ，代表第 i 列第 j 行像素的亮度。

Example 1.5 (影像轉矩陣). 給定以下 4×4 的灰階影像片段，其像素亮度值如下：

$$I = \begin{bmatrix} 34 & 67 & 120 & 255 \\ 12 & 45 & 80 & 200 \\ 23 & 60 & 100 & 180 \\ 0 & 30 & 90 & 170 \end{bmatrix}.$$

請說明此矩陣如何代表影像？若將此影像縮放為 0.5 倍亮度，請寫出新的矩陣表示。

Solution. 此矩陣為 $\mathbb{R}^{4 \times 4}$ 中的一個元素，每個元素表示對應像素的灰階亮度值。縮放亮度其實就是計算： $\tilde{I} = 0.5 \cdot I$ ，它的運算法則是把矩陣的每個元素都乘以 0.5，然後可得到：

$$\tilde{I} = 0.5 \begin{bmatrix} 34 & 67 & 120 & 255 \\ 12 & 45 & 80 & 200 \\ 23 & 60 & 100 & 180 \\ 0 & 30 & 90 & 170 \end{bmatrix} = \begin{bmatrix} 17 & 33.5 & 60 & 127.5 \\ 6 & 22.5 & 40 & 100 \\ 11.5 & 30 & 50 & 90 \\ 0 & 15 & 45 & 85 \end{bmatrix}.$$

□

由上述的討論，我們得到一個結論：「影像可以表示成矩陣的樣子」，而且，「操作矩陣就等於操作影像」。

Exercise 1.6. 給定下列灰階影像矩陣：

$$I = \begin{bmatrix} 20 & 40 & 60 & 80 \\ 30 & 50 & 70 & 90 \\ 40 & 60 & 80 & 100 \\ 50 & 70 & 90 & 110 \end{bmatrix},$$

- (1) 試說明這是幾乘幾的影像？總共有幾個像素？
- (2) 將此矩陣視為一張影像的亮度矩陣，哪一個元素對應最亮？哪一個最暗？

Exercise 1.7. 某張灰階影像的 3×3 區塊如下：

$$I = \begin{bmatrix} 30 & 60 & 90 \\ 120 & 150 & 180 \\ 210 & 240 & 255 \end{bmatrix}.$$

- (1) 試以文字說明此區域影像的視覺感受變化：從左上到右下的變化趨勢為何？
- (2) 若將影像進行反相 (inversion) 處理，即每個像素值 I_{ij} 變成 $255 - I_{ij}$ ，請寫出轉換後的新矩陣。
- (3) 轉換後的影像呈現為明亮還是暗調？為什麼？

1.2 矩陣加法與純量乘法

矩陣加法與純量乘法是線性代數中最基本的兩種運算，也是後續所有複雜操作的基礎。在影像處理中，矩陣加法常用來進行影像融合、遮罩與修補，而純量乘法則與亮度調整（如 Example 1.5）、對比增強密切相關。理解這兩種操作後，我們就可以開始動手改變一張影像的外觀，進一步感受數學操作如何影響圖像的呈現。

1.2.1 矩陣加法

首先我們定義兩個矩陣的加法。

Definition 1.8 (矩陣加法, Matrix Addition). 設 $A = [a_{ij}] \in \mathbb{R}^{m \times n}$ 與 $B = [b_{ij}] \in \mathbb{R}^{m \times n}$ 為兩個同型的實矩陣。則它們的和定義為：

$$A + B = [a_{ij} + b_{ij}] \in \mathbb{R}^{m \times n},$$

其中對於所有 $1 \leq i \leq m, 1 \leq j \leq n$ ，矩陣 $A + B$ 的第 (i, j) 元素為 $a_{ij} + b_{ij}$ 。

換句話說，矩陣加法是將兩個同型矩陣中對應位置的元素相加，比方說：

$$A = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 5 & 7 \\ 6 & 8 \end{bmatrix} \Rightarrow A + B = \begin{bmatrix} 6 & 10 \\ 8 & 12 \end{bmatrix},$$

從這裡我們很容易理解矩陣加法的操作。

★ 特別的，給定三個同型矩陣 $A, B, C \in \mathbb{R}^{m \times n}$ ，則矩陣加法滿足以下四個基本性質：

- (1) 交換律： $A + B = B + A$ 。
- (2) 結合律： $(A + B) + C = A + (B + C)$ 。
- (3) 加法單位元存在：存在零矩陣 $0 \in \mathbb{R}^{m \times n}$ ，使得 $A + 0 = A$ 。
- (4) 加法逆元存在：對每個 $A \in \mathbb{R}^{m \times n}$ ，存在 $-A \in \mathbb{R}^{m \times n}$ ，使得 $A + (-A) = 0$ 。

Exercise 1.9. 給定兩矩陣

$$A = \begin{bmatrix} 2 & 5 \\ 3 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 4 & -1 \\ 6 & 2 \end{bmatrix}, \quad C = \begin{bmatrix} 4 & -1 & 0 \\ 6 & 2 & 3 \end{bmatrix}.$$

請計算 $A + B$ 與 $A + C$ 。若無法運算，請說明原因。

1.2.2 矩陣加法在影像處理中的應用

矩陣加法在影像處理中的應用，就是「加入光源或背景 (blending)」。它的數學式可表達為：

$$I_{\text{new}} = I + B,$$

其中 B 是一個亮度偏移矩陣（可能是常數或區域光源），比方說增加亮度偏移（全部加上某值），或者是疊加另一張圖（背景或光源）。舉一個影像處理的實例，我們嘗試著加入光源遮罩，如圖 1.4 所示：

```
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt

# 讀取灰階影像並正規化
img = Image.open("example.jpg").convert('L').resize((300, 300))
I = np.asarray(img).astype(np.float64) / 255

# 製作一個光源遮罩矩陣
B = np.zeros_like(I)
B[120:180, 120:180] = 0.4 # 中央區域加亮

I_added = I + B
I_added = np.clip(I_added, 0, 1)

# 顯示疊加效果
fig, axs = plt.subplots(1, 3, figsize=(12, 4))
axs[0].imshow(I, cmap='gray'); axs[0].set_title("原始影像")
axs[1].imshow(B, cmap='gray'); axs[1].set_title("加亮遮罩 B")
axs[2].imshow(I_added, cmap='gray'); axs[2].set_title("I + B")
for ax in axs: ax.axis('off')
plt.tight_layout(); plt.show()
```

Figure 1.4: 矩陣加法：光源遮罩

然後可得到結果如圖 1.5 所示：

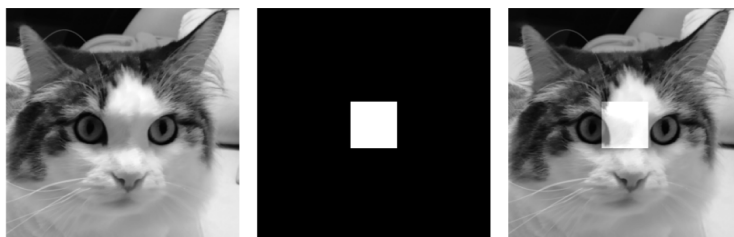


Figure 1.5: 矩陣加法：光源遮罩的結果

我們再給一個例子，利用矩陣加法將影像加亮影像畫素：

```
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image

# 讀取影像，轉為灰階並轉為 numpy 矩陣
img = Image.open('example.jpg').convert('L')
A = np.array(img)

# 建立與 A 同形狀的常數亮度補強矩陣（模擬遮罩加亮）
B = np.full_like(A, 30) # 每個像素加亮 30 單位

# 加法運算（需注意避免超出 255）
C = np.clip(A + B, 0, 255)

# 視覺化顯示兩張影像
fig, axs = plt.subplots(1, 2, figsize=(12, 4))
axs[0].imshow(A, cmap='gray'); axs[0].set_title('原始影像')
axs[1].imshow(C, cmap='gray'); axs[1].set_title('加法處理後')
for ax in axs: ax.axis('off')
plt.tight_layout(); plt.show()
```

Figure 1.6: 影像矩陣的加法處理與視覺化（加亮影像畫素）

然後可得到結果，如圖 1.7 所示：

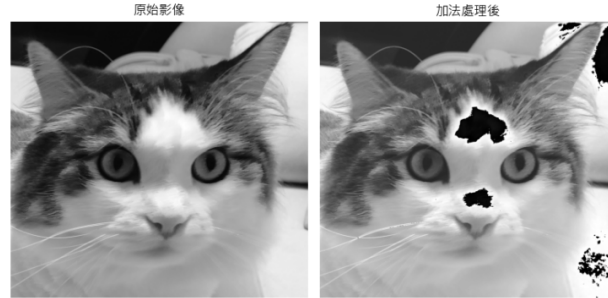


Figure 1.7: 原始影像與加法處理後的亮度過曝現象

1.2.3 矩陣的純量乘法

我們定義矩陣的純量乘法：

Definition 1.10 (純量乘法, Scalar Multiplication). 設 $A = [a_{ij}] \in \mathbb{R}^{m \times n}$ 為一 $m \times n$ 的實矩陣， $\lambda \in \mathbb{R}$ 為一純量 (scalar)。則 λ 與 A 的純量乘法定義為：

$$\lambda A = [\lambda a_{ij}] \in \mathbb{R}^{m \times n},$$

其中對所有 $1 \leq i \leq m$ ， $1 \leq j \leq n$ ， λA 的第 (i, j) 項為 λa_{ij} 。

換言之，純量乘法是將矩陣中每一個元素同時乘以實數 λ 。比方說

$$\begin{aligned}\lambda = 2, \quad A = \begin{bmatrix} 3 & -1 \\ 0 & 4 \end{bmatrix} &\Rightarrow 2A = \begin{bmatrix} 6 & -2 \\ 0 & 8 \end{bmatrix}, \\ \lambda = \pi, \quad A = \begin{bmatrix} 3 & -1 \\ 0 & 4 \end{bmatrix} &\Rightarrow \pi A = \begin{bmatrix} 3\pi & -\pi \\ 0 & 4\pi \end{bmatrix}\end{aligned}$$

★ 特別的，給定 $\lambda, \mu \in \mathbb{R}$, $A, B \in \mathbb{R}^{m \times n}$ ，則純量乘法滿足以下三個性質：

(1) 分配律： $\lambda(A + B) = \lambda A + \lambda B$ 。

(2) 結合律： $(\lambda\mu)A = \lambda(\mu A)$ 。

(2) 單位元純量： $1 \cdot A = A$ 。

這些性質可以方便我們進行矩陣的基本運算。

Exercise 1.11. 給定兩矩陣

$$A = \begin{bmatrix} 2 & 5 \\ 3 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 4 & -1 \\ 6 & 2 \end{bmatrix}.$$

請計算 (1) $3A$ ；(2) $A + 2B$ 。

Exercise 1.12. 設

$$E = \begin{bmatrix} 1 & 0 & 2 \\ -1 & 3 & 1 \end{bmatrix}, \quad F = \begin{bmatrix} 2 & 1 & -1 \\ 0 & -2 & 4 \end{bmatrix}.$$

請計算：(1) $E + F$ ；(2) $2E - 3F$ 。

1.2.4 純量乘法在影像處理中的應用

矩陣的純量乘法在影像處理中就是在控制整張影像的亮度 (brightness adjustment)，它的數學意義是，若原影像為矩陣 $I \in \mathbb{R}^{m \times n}$ ，整體的亮度調整等於乘上常數 $\alpha > 1$ (變亮)，或 $\alpha < 1$ (變暗)：

$$I_{\text{bright}} = \alpha \cdot I.$$

舉一個影像處理的實例，我們嘗試進行影像亮度調整，然後可得到結果如圖 1.8 所示，由左至右分別為 $\alpha = 1$ ， $\alpha = 0.5$ 及 $\alpha = 1.5$ 的結果。

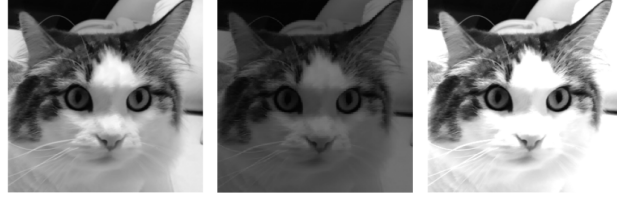


Figure 1.8: 純量乘法：影像亮度調整結果

Exercise 1.13. 給定下列灰階影像矩陣：

$$I = \begin{bmatrix} 20 & 40 & 60 \\ 30 & 50 & 70 \\ 40 & 60 & 80 \end{bmatrix}.$$

若將影像亮度增加為原本的 1.5 倍，試寫出新的矩陣。

Exercise 1.14. 請同學嘗試寫程式完成以下兩個任務：

- (1) 將一張灰階影像亮度整體分別乘以 1.5、0.5、-1，以觀察圖像變化。
- (2) 設計一段程式碼，自動判斷輸入影像是否過曝（意思是像素過多集中在高亮區域），並嘗試利用純量乘法與加法計算的技術來修正。

1.3 矩陣乘法

矩陣乘法是線性代數中最核心的運算之一。不同於加法與純量乘法的「逐個元素」操作，矩陣乘法引入了向量內積的概念，進一步揭示線性變換的本質。對影像處理來說，許多空間轉換（如模糊、旋轉、投影等）都可以形式化地表達為矩陣乘法的應用。

1.3.1 矩陣乘法的定義

首先我們來看矩陣乘法的定義。

Definition 1.15 (矩陣乘法, Matrix Multiplication). 設 $A = [a_{ik}] \in \mathbb{R}^{m \times n}$ 與 $B = [b_{kj}] \in \mathbb{R}^{n \times p}$ ，則它們的乘積 $C = A \times B = AB \in \mathbb{R}^{m \times p}$ 定義為：

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}, \quad \text{對於所有 } 1 \leq i \leq m, 1 \leq j \leq p,$$

其中 c_{ij} 表示矩陣 C 的第 i 列第 j 行元素。

換言之， C 為 $m \times p$ 矩陣，其中每一元素是 A 的第 i 列與 B 的第 j 行的內積。比方說下面這個例子， A 、 B 都是 2×2 矩陣，注意到 $n \times n$ 矩陣亦可稱為 n 階方陣。我們計算 AB 的結果：

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} \Rightarrow AB = \begin{bmatrix} 1 \cdot 5 + 2 \cdot 7 & 1 \cdot 6 + 2 \cdot 8 \\ 3 \cdot 5 + 4 \cdot 7 & 3 \cdot 6 + 4 \cdot 8 \end{bmatrix} = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}.$$

Example 1.16. 我們考慮以下兩個矩陣：

$$A = \begin{bmatrix} 1 & 2 & 0 \\ 3 & -1 & 4 \end{bmatrix} \in \mathbb{R}^{2 \times 3}, \quad B = \begin{bmatrix} 2 & 1 \\ 0 & -1 \\ 3 & 2 \end{bmatrix} \in \mathbb{R}^{3 \times 2}.$$

這兩個矩陣的維度符合定義，所以可以相乘，計算得到的結果為 2×2 矩陣：

$$\begin{aligned} AB &= \begin{bmatrix} 1 \cdot 2 + 2 \cdot 0 + 0 \cdot 3 & 1 \cdot 1 + 2 \cdot (-1) + 0 \cdot 2 \\ 3 \cdot 2 + (-1) \cdot 0 + 4 \cdot 3 & 3 \cdot 1 + (-1) \cdot (-1) + 4 \cdot 2 \end{bmatrix} \\ &= \begin{bmatrix} 2 & -1 \\ 6 + 0 + 12 & 3 + 1 + 8 \end{bmatrix} = \begin{bmatrix} 2 & -1 \\ 18 & 12 \end{bmatrix}. \end{aligned}$$

□

也有一個比較趣味的例子供同學們參考：

$$\begin{bmatrix} \text{弱} & \text{強} \\ \text{燒} & \text{定} \end{bmatrix} \times \begin{bmatrix} \text{肉} & \text{肉} \\ \text{食} & \text{食} \end{bmatrix} = \begin{bmatrix} \text{弱肉} + \text{強食} & \text{弱肉} + \text{強食} \\ \text{燒肉} + \text{定食} & \text{燒肉} + \text{定食} \end{bmatrix}.$$

希望這個例子能讓同學們更快理解矩陣乘法是如何運算的。

Exercise 1.17. 設有矩陣 $A = \begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix}$ 、 $B = \begin{bmatrix} 1 & 4 \\ 2 & -1 \end{bmatrix}$ ，請計算 AB 和 BA 。

Exercise 1.18. 設有矩陣

$$A = \begin{bmatrix} 1 & 2 & -1 \\ 0 & 3 & 4 \end{bmatrix} \in \mathbb{R}^{2 \times 3}, \quad B = \begin{bmatrix} 2 & 0 \\ -1 & 1 \\ 3 & 2 \end{bmatrix} \in \mathbb{R}^{3 \times 2}, \quad C = \begin{bmatrix} -3 & 1 \\ 0 & 1 \\ 1 & -2 \end{bmatrix} \in \mathbb{R}^{3 \times 2}.$$

請計算：(1) $3AB$ ；(2) $A(B - 2C)$ 。

1.3.2 矩陣與向量相乘

向量是線性代數的基本元素，在高中時是數學、物理學科的重要知識之一。向量同時具有大小和方向，一般我們會寫為

$$\vec{u} = (u_1, u_2, \dots, u_n) \in \mathbb{R}^n$$

的形式，它可視為起點為原點，終點指向點 $(u_1, u_2, \dots, u_n)$ ，且長度恰為

$$\sqrt{u_1^2 + u_2^2 + \dots + u_n^2}$$

的幾何對象。但通常向量的起點和終點並不那麼重要，兩個起點不一樣的向量，只要大小相等，方向相同，就可以稱為是同一個向量。

正因為向量有這樣的靈活性，所以在物理及各個應用領域中也都是極為重要的概念。

Definition 1.19 (向量, Vector). 在 $\mathbb{R}^n$ 中，一個向量是由 n 個實數組成的有序數列，常記為：

$$\vec{v} = (v_1, v_2, \dots, v_n) \quad (\equiv \mathbf{v}).$$

它同時具有方向與大小，而這樣的記法稱為向量的坐標表示法。特別地，向量的長度可定義為

$$\sqrt{v_1^2 + v_2^2 + \dots + v_n^2},$$

通常記為 $|\vec{v}|$ 或是 $\|\vec{v}\|$ 。

$\mathbb{R}^n$ 中的向量其實也可以寫成「矩陣的形式」；然而，如果我們要把向量寫成矩陣形式後進行運算，通常會把向量寫成直的「行向量」，也就是

$$\vec{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \in \mathbb{R}^{n \times 1}, \quad (1.3.1)$$

式 (1.3.1) 中的表示法就是向量的矩陣形式，向量寫成矩陣後是一個 $n \times 1$ 矩陣。

事實上，把向量寫成直的很佔版面，為了讓版面美觀，有些書會把式 (1.3.1) 中的行向量寫成

$$\vec{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = [v_1 \ v_2 \ \dots \ v_n]^\top,$$

其中，符號 $\top$ 的操作稱為轉置 (transpose)。

注意到矩陣 $[v_1 \ v_2 \ \dots \ v_n]$ 的大小為 $1 \times n$ 。

Remark 1.20. 若矩陣 $A = [a_{ij}] \in \mathbb{R}^{m \times n}$ ，則 A 的轉置矩陣 $A^\top$ 可寫為

$$A^\top = [a_{ji}] \in \mathbb{R}^{n \times m}.$$

比方說， $A = \begin{bmatrix} 2 & 0 \\ 1 & 3 \\ -1 & -1 \end{bmatrix} \in \mathbb{R}^{3 \times 2}$ ，其轉置矩陣 $A^\top$ 為

$$A^\top = \begin{bmatrix} 2 & 1 & -1 \\ 0 & 3 & -1 \end{bmatrix} \in \mathbb{R}^{2 \times 3}.$$

□

Example 1.21. 給定矩陣 $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \in \mathbb{R}^{2 \times 2}$ ，向量（矩陣形式） $\vec{v} = \begin{bmatrix} a \\ b \end{bmatrix} \in \mathbb{R}^{2 \times 1}$ ，則

$$A \vec{v} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} a + 2b \\ 3a + 4b \end{bmatrix} \in \mathbb{R}^{2 \times 1}.$$

□

注意到在線性代數裡，我們通常都將向量寫成行向量（即矩陣形式）來與其他矩陣做運算。

Example 1.22. 設

$$A = \begin{bmatrix} 1 & -1 & 2 \\ 0 & 3 & 1 \end{bmatrix} \in \mathbb{R}^{2 \times 3}, \quad \vec{v} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \in \mathbb{R}^{3 \times 1},$$

則矩陣與向量相乘為：

$$A \vec{v} = \begin{bmatrix} 1 & -1 & 2 \\ 0 & 3 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x - y + 2z \\ 3y + z \end{bmatrix} \in \mathbb{R}^{2 \times 1},$$

這表示一個 3 維的向量經由 A 映射到 2 維的空間。

□

Exercise 1.23. 令矩陣

$$A = \begin{bmatrix} 2 & -1 \\ 0 & 3 \end{bmatrix}, \quad \vec{u} = \begin{bmatrix} 4 \\ 5 \end{bmatrix}$$

請計算 $A \vec{u}$ 。

Exercise 1.24. 令矩陣

$$A = \begin{bmatrix} 1 & 2 \\ -1 & 0 \\ 3 & 1 \end{bmatrix}, \quad \vec{v} = \begin{bmatrix} 2 \\ -1 \end{bmatrix},$$

請計算 $A \vec{v}$ 。

1.3.3 矩陣乘法的性質

學習完矩陣乘法的基本定義與計算方法後，我們需要理解矩陣乘法在結構上具備哪些性質，例如，矩陣乘法滿足結合律與分配律，卻不滿足交換性。

Proposition 1.25 (矩陣乘法的結合律與分配律). 設 A, B, C 為適當維度的矩陣，則：

- (1) 結合律： $(AB)C = A(BC)$ 。
- (2) 分配律： $A(B + C) = AB + AC$ ， $(A + B)C = AC + BC$ 。

下面的例子我們可以驗證 Proposition 1.25。

Example 1.26. 假設

$$A = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 3 & 0 \\ 1 & 2 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

請依序計算：

- (1) 請比較 $(AB)C$ 與 $A(BC)$ 兩者是否相等。
- (2) 請比較 $A(B + C)$ 與 $AB + AC$ 兩者是否相等。
- (3) 請比較 $B + C$ 與 $C + B$ 是否相等，並請試著說明它所代表的現象。

注意到矩陣 A, B 相乘有「非交換性」，意思是，若 AB, BA 都存在，則 AB 不一定等於 BA 。所謂的「非交換性」不僅是數學上的概念，在應用時，它也會造成影像在不同操作順序下結果的差異。

矩陣乘法可視為對向量進行線性變換，這些變換包括旋轉、縮放、平移等操作。當兩個矩陣依序作用在一個向量 $\vec{u}$ 上時，實際上是將 $\vec{u}$ 先經過第一個變換 S ，再經過第二個變換 T ，即

$$\vec{u} \xrightarrow{\text{經過第一個變換}} S \vec{u} \xrightarrow{\text{經過第二個變換}} T(S \vec{u}) = TS \vec{u}.$$

因此，改變 S 與 T 的乘法順序就可能改變最終的變換結果。

比方說，假設 A 是一個水平拉伸矩陣， B 是一個旋轉矩陣。若先拉伸再旋轉 (BA)，與先旋轉再拉伸 (AB) 將得到完全不同的結果。這反映出矩陣乘法的非交換性能直接影響空間幾何的結構與應用。以程式來解釋的話，程式如圖 ?? 所示：

實際跑程式的結果如圖 1.9 所示：



Figure 1.9: 影像輸出後的比較

接下來，我們可以回到數學的角度來驗證矩陣乘法會有非交換性。

Example 1.27 (非交換性). 設：

$$A = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 3 & 0 \\ 1 & 2 \end{bmatrix}$$

則：

$$AB = \begin{bmatrix} 1 \cdot 3 + 2 \cdot 1 & 1 \cdot 0 + 2 \cdot 2 \\ 0 \cdot 3 + 1 \cdot 1 & 0 \cdot 0 + 1 \cdot 2 \end{bmatrix} = \begin{bmatrix} 5 & 4 \\ 1 & 2 \end{bmatrix}, \quad BA = \begin{bmatrix} 3 \cdot 1 + 0 \cdot 0 & 3 \cdot 2 + 0 \cdot 1 \\ 1 \cdot 1 + 2 \cdot 0 & 1 \cdot 2 + 2 \cdot 1 \end{bmatrix} = \begin{bmatrix} 3 & 6 \\ 1 & 4 \end{bmatrix},$$

因此 $AB \neq BA$ ，矩陣乘法不具交換性。 $\square$

Example 1.28. 令：

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \in \mathbb{R}^{2 \times 3}, \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & -1 \\ 2 & 1 \end{bmatrix} \in \mathbb{R}^{4 \times 2},$$

因為 AB 是 2×3 乘以 4×2 ，中間維度 $3 \neq 4$ ，無法相乘，因此 AB 不符合矩陣乘法的前提，即 AB 無定義。然而， BA 是 4×2 乘以 2×3 ，中間維度皆為 2，所以可進行矩陣乘法，結果為 4×3 矩陣。

實際計算 BA ：

$$\begin{aligned} BA &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & -1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \\ &= \begin{bmatrix} 1 \cdot 1 + 0 \cdot 4 & 1 \cdot 2 + 0 \cdot 5 & 1 \cdot 3 + 0 \cdot 6 \\ 0 \cdot 1 + 1 \cdot 4 & 0 \cdot 2 + 1 \cdot 5 & 0 \cdot 3 + 1 \cdot 6 \\ 1 \cdot 1 + (-1) \cdot 4 & 1 \cdot 2 + (-1) \cdot 5 & 1 \cdot 3 + (-1) \cdot 6 \\ 2 \cdot 1 + 1 \cdot 4 & 2 \cdot 2 + 1 \cdot 5 & 2 \cdot 3 + 1 \cdot 6 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ -3 & -3 & -3 \\ 6 & 9 & 12 \end{bmatrix}. \end{aligned}$$

所以，即使 AB 可以計算， BA 卻可能無法被定義，所以矩陣乘法不具有交換性。 $\square$

Exercise 1.29. 設

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}.$$

請判斷：

- (1) $(A + B) + C$ 與 $A + (B + C)$ 是否相等？
- (2) 是否存在矩陣加法下不滿足結合律的情形？請計算結果並解釋。

Exercise 1.30. 設

$$A = \begin{bmatrix} 1 & 2 \\ 0 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 \\ -1 & 2 \end{bmatrix},$$

請分別計算 AB 與 BA 的結果，並說明 AB 與 BA 是否相等。

Exercise 1.31. 設

$$A = \begin{bmatrix} 1 & 2 & -1 \\ 0 & -1 & 3 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ -1 & 2 \end{bmatrix},$$

請分別計算 AB 與 BA 的結果，並說明 AB 與 BA 是否相等。

Exercise 1.32. 已知矩陣 T_1 是將影像向右平移， T_2 是將影像放大兩倍：

- (1) 試說明為何 $T_1 T_2 \neq T_2 T_1$ 。
- (2) 若影像先縮放再平移，與先平移再縮放，影像中物體位置會有何差異？

1.3.4 單位矩陣與零矩陣

此外，我們也需要認識單位矩陣 (Identity Matrix) 與零矩陣 (Zero Matrix)，它們在矩陣乘法中所扮演的角色可以幫助我們建立起更完整的矩陣觀念。

Definition 1.33 (單位矩陣). 對任意正整數 n ， $n \times n$ 的單位矩陣 (Identity Matrix) 記為 $\mathbb{I}_n$ ，定義為對角線元素皆為 1，其餘元素皆為 0 的正方形矩陣，即：

$$(\mathbb{I}_n)_{ij} = \begin{cases} 1 & \text{若 } i = j, \\ 0 & \text{若 } i \neq j, \end{cases} \quad \text{其中 } 1 \leq i, j \leq n.$$

單位矩陣 $\mathbb{I}_n$ 通常也寫為 $\mathbb{I}_n = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}_{n \times n}$ 。

Definition 1.34 (零矩陣). 對任意 $m, n \in \mathbb{N}$, $m \times n$ 的零矩陣 (Zero Matrix) 記為 $\mathbf{0}_{m \times n}$, 定義為所有元素皆為 0 的矩陣:

$$(\mathbf{0}_{m \times n})_{ij} = 0, \quad \text{對所有 } 1 \leq i \leq m, 1 \leq j \leq n.$$

若不特別指定大小, 常簡記為 $\mathbf{0}$ 。

單位矩陣與零矩陣是線性代數中最基本也最具結構意義的對象之一, 同學們可以透過以下的性質理解它們在矩陣運算中所扮演的角色。

Proposition 1.35. 單位矩陣與零矩陣的乘法性質分別如下:

- (1) 單位矩陣 $\mathbb{I}_n$: 對所有 $A \in \mathbb{R}^{n \times n}$, 有 $A\mathbb{I}_n = \mathbb{I}_n A = A$ 。
- (2) 零矩陣 $\mathbf{0}$: 對所有 A , 有 $A\mathbf{0} = \mathbf{0}A = \mathbf{0}$ 。

舉個簡單的例題, 設 $A = \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}$, $I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $\mathbf{0} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$, 則可由計算驗證出:

$$(1) A\mathbb{I}_2 = \mathbb{I}_2 A = A; \quad (2) A + \mathbf{0} = \mathbf{0} + A = A; \quad (3) A \cdot \mathbf{0} = \mathbf{0} \cdot A = \mathbf{0}.$$

Exercise 1.36. 對於任意符合維度的矩陣 A 與單位矩陣 $\mathbb{I}$ 、零矩陣 $\mathbf{0}$, 判斷以下四個敘述是否正確, 並簡述理由:

- (1) $A\mathbb{I} = \mathbb{I}A = A$; (2) $A + \mathbf{0} = A$; (3) $A \cdot \mathbf{0} = \mathbf{0} \cdot A = A$ 。
- (4) 若 $AB = \mathbb{I}$, 則必有 $BA = \mathbb{I}$ 。

Exercise 1.37. 填入正確的矩陣使等式成立:

$$(1) A + \boxed{} = A; \quad (2) \boxed{} \cdot A = \mathbf{0}; \quad (3) A \cdot \boxed{} = A.$$

1.4 向量的線性組合

我們先前提過向量是線性代數的基本元素, 而線性組合則是理解向量空間結構的基礎觀念。影像中的每一行、每一列、甚至一張圖都可以視為高維向量的一部分, 能否掌握向量的加權疊加概念, 直接影響我們對於影像特徵、壓縮與分解的理解。我們接下來會從直觀的幾何觀點出發, 引導同學們理解兩個或多個向量如何組成新的向量, 線性組合如何用於資料近似, 並進一步鋪墊後續介紹向量空間與主成分分析的理論基礎。

1.4.1 線性組合的數學形式與操作

Definition 1.38 (線性組合, Linear Combination). 若給定向量 $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_k \in \mathbb{R}^n$, 以及實數係數 $c_1, c_2, \dots, c_k$, 則向量 :

$$\vec{w} = c_1 \vec{v}_1 + c_2 \vec{v}_2 + \dots + c_k \vec{v}_k$$

稱為這些向量的線性組合。

以下我們來看一個簡單的例題：

Example 1.39. 設 $\vec{v}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ 、 $\vec{v}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $c_1 = 3$, $c_2 = -2$, 則：

$$\vec{w} = 3 \vec{v}_1 - 2 \vec{v}_2 = 3 \begin{bmatrix} 1 \\ 0 \end{bmatrix} - 2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ -2 \end{bmatrix} .$$

此向量 $\vec{w}$ 是 $\vec{v}_1$ 與 $\vec{v}_2$ 的線性組合，也就是從原點指向平面中坐標 $(3, -2)$ 的箭頭。 □

從幾何意義來看，向量的線性組合可視為將不同方向與長度的向量進行加權「拉伸與疊加」，然後產生新的向量。例如在二維平面中，兩個不共線的向量所張成的所有線性組合會構成一個平面區域。

Example 1.40. 令

$$\vec{v}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \vec{v}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

則任意向量 $\vec{w} = a \vec{v}_1 + b \vec{v}_2$ 可表示為 $\begin{bmatrix} a \\ b \end{bmatrix}$ ，對應到 $\mathbb{R}^2$ 中的向量 (a, b) 。若 $\vec{v}_1$ 與 $\vec{v}_2$ 呈倍數關係（例如 $\vec{v}_2 = 2 \vec{v}_1$ ），所有線性組合明顯地將只落在一條直線上，無法填滿平面。 □

Example 1.41. 考慮三個三維空間中的向量：

$$\vec{v}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \vec{v}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \vec{v}_3 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix},$$

如果我們對這三個向量做線性組合，會產生哪些新的向量？由觀察可知：

$$\vec{v}_3 = \vec{v}_1 + \vec{v}_2,$$

所以 $\vec{v}_3$ 其實沒有提供新的方向，它只是前兩個方向的加總，也就是說：

- $\vec{v}_1$ 和 $\vec{v}_2$ 可以帶我們走 xy 平面的「東西向」和「南北向」；

- $\vec{v}_3$ 只是在 xy 平面走東再走北，跟 $\vec{v}_1, \vec{v}_2$ 本身的方向沒差別；
- 因此，用這三個向量做任何線性組合，都無法讓我們在 z 軸方向有任何的移動。

所以這三個向量的線性組合只會落在水平的 xy 平面中，無法離開這個平面，即使原本題目給定的三個向量都是落在 $\mathbb{R}^3$ 空間裡。 $\square$

我們同樣可以以 Python 程式碼來說明視覺化線性組合範圍：

```
import numpy as np
import matplotlib.pyplot as plt

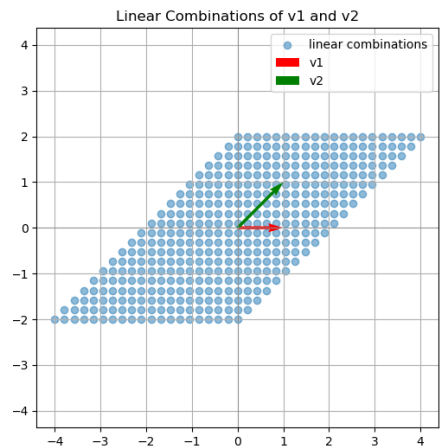
# 定義兩個基底向量
v1 = np.array([1, 0])
v2 = np.array([0, 1])

# 建立係數範圍
c1_vals = np.linspace(-2, 2, 20)
c2_vals = np.linspace(-2, 2, 20)

# 建立所有線性組合點
points = [c1 * v1 + c2 * v2 for c1 in c1_vals for c2 in c2_vals]
points = np.array(points)

# 繪圖
plt.figure(figsize=(8, 8))
plt.scatter(points[:, 0], points[:, 1], alpha=0.5, label='linear combinations')
plt.quiver(0, 0, v1[0], v1[1], angle='xy', scale_units='xy', scale=1, color='r', label='v1')
plt.quiver(0, 0, v2[0], v2[1], angle='xy', scale_units='xy', scale=1, color='g', label='v2')
plt.axhline(0, color='gray', lw=0.5)
plt.axvline(0, color='gray', lw=0.5)
plt.axis('equal')
plt.legend()
plt.title('Linear Combinations of v1 and v2')
plt.grid(True)
plt.show()
```

(a) 對應的 Python 程式碼



(b) v_1 與 v_2 的線性組合所構成的平面

Figure 1.10: 線性組合視覺化與程式說明

Exercise 1.42. 設 $\vec{v}_1 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$, $\vec{v}_2 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$, $c_1 = 2$, $c_2 = 4$, 請計算向量：

$$\vec{w} = 2\vec{v}_1 + 4\vec{v}_2,$$

並以坐標表示法寫出 $\vec{w}$ 。

Exercise 1.43. 設 $\vec{v}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $\vec{v}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, 請問以下哪一個向量是它們的線性組合？

(A) $\begin{bmatrix} 2 \\ 3 \end{bmatrix}$; (B) $\begin{bmatrix} -1 \\ 0 \end{bmatrix}$; (C) $\begin{bmatrix} 0 \\ -4 \end{bmatrix}$; (D) 以上皆是。

Exercise 1.44. 令

$$\vec{v}_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad \vec{v}_2 = \begin{bmatrix} 3 \\ -1 \end{bmatrix}$$

試問以下哪些向量可以寫成 $a\vec{v}_1 + b\vec{v}_2$ 的形式？請說明理由並找出對應的係數。

(1) $\begin{bmatrix} 4 \\ 3 \end{bmatrix}$, (2) $\begin{bmatrix} 2 \\ 2 \end{bmatrix}$, (3) $\begin{bmatrix} 5 \\ 0 \end{bmatrix}$.

1.4.2 線性組合在影像處理中的應用

線性組合的概念放在一組矩陣 $\{A_1, A_2, \dots, A_k\}$ 中也適用，矩陣的線性組合相當於把一組矩陣拿來做純量乘法後相加，即：

$$c_1 A_1 + c_2 A_2 + \dots + c_k A_k,$$

所以，當兩張影像轉換成矩陣後做線性組合時，我們實際上是在計算類似以下的公式：

$$I = \alpha \cdot I_1 + (1 - \alpha) \cdot I_2, \quad \alpha \in [0, 1],$$

這種操作在影像處理中常見於影像混合 (image blending)、影像變形與過渡 (morphing) 與動態前後景建構 (background modeling)，請見以下的有關影像混合的例題：

Example 1.45. 假設有兩張圖像 $I_1, I_2 \in \mathbb{R}^{m \times n}$ ，定義影像混合後的結果 I_{blend} 為：

$$I_{\text{blend}} = \alpha I_1 + (1 - \alpha) I_2,$$

其中 $\alpha \in [0, 1]$ ，可以控制兩圖之間混合的比例，這就是影像混合常用的方式。同學可執行以下的程式驗證：

```
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt

# 載入並轉為灰階 + 統一尺寸
img1 = Image.open("image1.jpg").convert('L').resize((300, 300))
img2 = Image.open("image2.jpg").convert('L').resize((300, 300))

I1 = np.asarray(img1).astype(np.float64) / 255
I2 = np.asarray(img2).astype(np.float64) / 255

# 線性組合 (α = 0.3)
alpha = 0.3
I_blend = alpha * I1 + (1 - alpha) * I2
```

線性組程式 (圖像疊加)

```
# 顯示結果
fig, axs = plt.subplots(1, 3, figsize=(12, 4))
axs[0].imshow(I1, cmap='gray')
axs[0].set_title("Image 1")

axs[1].imshow(I2, cmap='gray')
axs[1].set_title("Image 2")

axs[2].imshow(I_blend, cmap='gray')
axs[2].set_title(f"Linear Blend (α = {alpha})")

for ax in axs:
    ax.axis('off')

plt.tight_layout()
plt.show()
```

線性組合結果展示程式碼

Figure 1.11: 將兩張灰階圖進行線性組合

針對 image1.jpg 及 image2.jpg，可用 $\alpha = 0.3$ 達到以下的效果：



Figure 1.12: 影像輸出後的比較

□

Exercise 1.46. 灰階影像有另一種數學表達的方式，我們可以將一張 $m \times n$ 像素的灰階影像視為長度為 mn 的行向量，如果我們手中有 k 張 $m \times n$ 像素的灰階影像，每一張都表示成長度為 mn 的行向量，則這 k 張影像可排列成一個 $mn \times k$ 的大矩陣 M ，且矩陣 M 中的每一行都代表了一張圖片的資訊。試解釋若矩陣 M 中所有行向量都可用某一個特定向量的線性組合來表示，這對這張圖片而言代表什麼？

1.4.3 向量空間的定義

由 Exercise 1.46，我們得到了一個把灰階影像轉成行向量的方法，也就是說，我們可以將一張張 $m \times n$ 像素的灰階影像

$$A^{(1)} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}^{(1)} & a_{m2}^{(1)} & \cdots & a_{mn}^{(1)} \end{bmatrix}_{m \times n}, \quad A^{(2)} = \begin{bmatrix} a_{11}^{(2)} & a_{12}^{(2)} & \cdots & a_{1n}^{(2)} \\ a_{21}^{(2)} & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}^{(2)} & a_{m2}^{(2)} & \cdots & a_{mn}^{(2)} \end{bmatrix}_{m \times n}, \dots$$

視為長度為 mn 的行向量：

$$\vec{v}_1 = \begin{bmatrix} a_{11}^{(1)} \\ \vdots \\ a_{1n}^{(1)} \\ a_{21}^{(1)} \\ \vdots \\ a_{2n}^{(1)} \\ \vdots \\ a_{mn}^{(1)} \end{bmatrix}_{mn \times 1}, \quad \vec{v}_2 = \begin{bmatrix} a_{11}^{(2)} \\ \vdots \\ a_{1n}^{(2)} \\ a_{21}^{(2)} \\ \vdots \\ a_{2n}^{(2)} \\ \vdots \\ a_{mn}^{(2)} \end{bmatrix}_{mn \times 1}, \dots \vec{v}_i = \begin{bmatrix} a_{11}^{(i)} \\ \vdots \\ a_{1n}^{(i)} \\ a_{21}^{(i)} \\ \vdots \\ a_{2n}^{(i)} \\ \vdots \\ a_{mn}^{(i)} \end{bmatrix}_{mn \times 1}, \dots$$

如果我們手中有 k 張 $m \times n$ 像素的灰階影像，每一張都表示成長度為 mn 的行向量，則這 k 張影像可排列成一個 $mn \times k$ 的矩陣 I_{total} 如下：

$$I_{\text{total}} = [\vec{v}_1 \ \vec{v}_2 \ \cdots \ \vec{v}_k]_{mn \times k} = \begin{bmatrix} a_{11}^{(1)} & a_{11}^{(2)} & \cdots & a_{11}^{(k)} \\ \vdots & \vdots & \vdots & \vdots \\ a_{1n}^{(1)} & a_{1n}^{(2)} & \cdots & a_{1n}^{(k)} \\ a_{21}^{(1)} & a_{21}^{(2)} & \cdots & a_{21}^{(k)} \\ \vdots & \vdots & \vdots & \vdots \\ a_{2n}^{(1)} & a_{2n}^{(2)} & \cdots & a_{2n}^{(k)} \\ \vdots & \vdots & \vdots & \vdots \\ a_{mn}^{(1)} & a_{mn}^{(2)} & \cdots & a_{mn}^{(k)} \end{bmatrix}_{mn \times k}.$$

矩陣中的每一行都代表了一張圖片的資訊。

既然每個影像矩陣都可以視為一個行向量，那麼我們就需要一個數學結構來描述「一組向量的合法集合」，這個結構就是「向量空間」。

Definition 1.47 (向量空間, Vector Space). 設 V 為一個非空集合，其上的元素稱為向量，且定義了兩個運算：向量加法與純量乘法。若這兩個運算滿足以下所有的條件，則稱 V 為定義在體 $\mathbb{R}$ 上的向量空間 (vector space over $\mathbb{R}$)：

1. 加法封閉性：對任意 $\vec{u}, \vec{v} \in V$ ，有 $\vec{u} + \vec{v} \in V$ 。
2. 加法交換律：對任意 $\vec{u}, \vec{v} \in V$ ，有 $\vec{u} + \vec{v} = \vec{v} + \vec{u}$ 。
3. 加法結合律：對任意 $\vec{u}, \vec{v}, \vec{w} \in V$ ，有 $(\vec{u} + \vec{v}) + \vec{w} = \vec{u} + (\vec{v} + \vec{w})$ 。
4. 存在加法單位元：存在 $\vec{0} \in V$ ，使得對任意 $\vec{v} \in V$ ，皆有 $\vec{v} + \vec{0} = \vec{v}$ 。
5. 存在加法反元素：對任意 $\vec{v} \in V$ ，存在 $-\vec{v} \in V$ ，使得 $\vec{v} + (-\vec{v}) = \vec{0}$ 。
6. 純量乘法封閉性：對任意 $c \in \mathbb{R}$ 與 $\vec{v} \in V$ ，有 $c\vec{v} \in V$ 。
7. 純量乘法結合律：對任意 $a, b \in \mathbb{R}$ 與 $\vec{v} \in V$ ，有 $a(b\vec{v}) = (ab)\vec{v}$ 。
8. 分配律：
 - 向量上的分配律：對任意 $a \in \mathbb{R}$ 、 $\vec{u}, \vec{v} \in V$ ， $a(\vec{u} + \vec{v}) = a\vec{u} + a\vec{v}$ 。
 - 純量上的分配律：對任意 $a, b \in \mathbb{R}$ 、 $\vec{v} \in V$ ， $(a + b)\vec{v} = a\vec{v} + b\vec{v}$ 。
9. 單位純量作用：對任意 $\vec{v} \in V$ ， $1 \cdot \vec{v} = \vec{v}$ 。

為什麼影像處理可以被視為發生在向量空間中的操作？因為在向量空間中，影像即向

量。前面提過我們可以將一張 $m \times n$ 的影像視為長度為 mn 的行向量，所有這種「拉直後的影像」，將構成 $\mathbb{R}^{mn \times 1}$ 的向量空間，而且，在此空間中，與加法、純量乘法相關的動作全都合法，所以，影像經過處理後仍會是合法的影像，請見以下的例題。

Example 1.48. 設 $I_1, I_2 \in \mathbb{R}^{m \times n}$ 為兩張影像，將它們向量化後，得到：

$$\vec{i}_1 = \text{vec}(I_1), \quad \vec{i}_2 = \text{vec}(I_2) \in \mathbb{R}^{mn \times 1}.$$

對任意純量 $a, b \in \mathbb{R}$ ，我們可定義新的向量：

$$\vec{i} = a \vec{i}_1 + b \vec{i}_2.$$

則 $\vec{i} \in \mathbb{R}^{mn \times 1}$ ，說明向量化後的影像在此向量空間中對加法與純量乘法封閉。若我們再將 $\vec{i}$ 還原為 $m \times n$ 的影像矩陣，仍然是一張合法的影像。 $\square$

Exercise 1.49. 設有兩張影像：

$$I_1 = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad I_2 = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} \in \mathbb{R}^{2 \times 2}.$$

- (1) 分別將 I_1 與 I_2 向量化，並寫出 $\vec{i}_1 = \text{vec}(I_1)$ 、 $\vec{i}_2 = \text{vec}(I_2)$ 。
- (2) 計算 $\vec{i} = \frac{1}{2} \vec{i}_1 + \frac{1}{2} \vec{i}_2$ ，並寫出其向量形式。
- (3) 將 $\vec{i}$ 還原為 2×2 的影像矩陣，並說明該影像是否仍屬於 $\mathbb{R}^{2 \times 2}$ 。
- (4) 試說明此操作為何符合向量空間的封閉性。

1.4.4 矩陣空間

在數學中，原本向量是指 $\mathbb{R}^n$ 中的元素，例如 $\vec{v} = (1, 2, 3) \in \mathbb{R}^3$ ，而我們對向量做的操作（包括加法與純量乘法），要求這些運算滿足一系列的條件，這使得 $\mathbb{R}^n$ 成為一個向量空間。然而，向量空間不一定只能是 $\mathbb{R}^n$ ，任何集合只要定義了加法與純量乘法，且這兩個運算滿足定義中所有的條件，這個集合就可以構成一個向量空間。例如，所有由 $m \times n$ 的實矩陣構成的集合：

$$M_{m \times n}(\mathbb{R}) = \{A \in \mathbb{R}^{m \times n}\},$$

這個集合在矩陣加法與純量乘法下也構成向量空間，因為它滿足向量空間定義裡所有的條件。

所以，矩陣可以視為某個向量空間中的元素，只是我們把它的形狀維持成 $m \times n$ ，而不是把它拉直成 $\mathbb{R}^{mn \times 1}$ 的行向量。

Example 1.50. 若 $I_1, I_2 \in \mathbb{R}^{m \times n}$ ，則對任何 $a, b \in \mathbb{R}$ ，有

$$I = a \cdot I_1 + b \cdot I_2 \in \mathbb{R}^{m \times n}.$$

矩陣 I 仍然在 $\mathbb{R}^{m \times n}$ 中，這顯示了影像矩陣在像素加權上構成封閉的向量空間。□

回過頭觀察上一節的圖 1.12，我們可從視覺化的角度觀察到影像的線性組合仍是一張合法的影像，這就是所謂的向量空間的封閉性。

Exercise 1.51. 設三個矩陣 $A, B, C \in \mathbb{R}^{2 \times 3}$ 分別為：

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, \quad B = \begin{bmatrix} -1 & 0 & 1 \\ 0 & -1 & 2 \end{bmatrix}, \quad C = \begin{bmatrix} 2 & 1 & 0 \\ -1 & 3 & 1 \end{bmatrix}.$$

- (1) 計算線性組合 $D = 2A - 3B + C$ ，並寫出結果矩陣。
- (2) 證明矩陣 D 仍屬於 $\mathbb{R}^{2 \times 3}$ 。
- (3) 試說明：為何 $\mathbb{R}^{2 \times 3}$ 可視為一個向量空間？我們在此題中驗證了哪些向量空間的性質？

Exercise 1.52. 設兩張灰階影像矩陣為：

$$I_1 = \begin{bmatrix} 100 & 120 \\ 80 & 90 \end{bmatrix}, \quad I_2 = \begin{bmatrix} 30 & 60 \\ 70 & 110 \end{bmatrix}.$$

若 $a = 0.5, b = 0.5$ ，請計算線性組合影像：

$$I = aI_1 + bI_2.$$

並說明 I 是否仍屬於 $\mathbb{R}^{2 \times 2}$ 。更進一步地，請問這個例子是否說明了向量空間的封閉性？

1.4.5 向量空間的維度

每個向量空間都有一個稱為「維度 (dimension)」的重要性質。這個維度代表的是：「我們需要幾個彼此獨立的方向，才能完整地描述空間中的所有向量」。這些方向不只是幾何上的箭頭，也可以理解為生成整個空間的「基本構成單位」。更具體地說，維度就是「能夠產生該空間所有向量所需的最少向量數量」。

例如，在 $\mathbb{R}^2$ 中：

- 一條通過原點的直線只需要一個非零向量就能張成整條線（例如 $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$ 所張成的所有倍數），因此它是一個一維空間；

- 平面 $\mathbb{R}^2$ 需要兩個不共線的向量（如 $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ 和 $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ ）才能描述 $\mathbb{R}^2$ 中所有的向量，故稱為二維空間。

同理，在 $\mathbb{R}^{m \times n}$ 中，每一個影像矩陣（例如一張 $m \times n$ 的灰階影像）都可以拉平成長度為 mn 的行向量，使它變成 $\mathbb{R}^{mn \times 1}$ 空間中的一個點。這表示所有大小為 $m \times n$ 的影像矩陣，經過向量化之後，共同構成了一個維度為 mn 的向量空間 $\mathbb{R}^{mn \times 1}$ 。

在應用方面，若大量影像資料主要集中在少數幾個變化方向上，那麼我們就可以只保留這些主要方向所需的資訊來近似整體的影像。所以，哪些方向是主要的，這些方向共有多少個（這就是維度），我們都會需要知道。

Example 1.53. 一張 28×28 的 MNIST 手寫數字影像，是一張灰階影像矩陣，可視為：

$$I \in \mathbb{R}^{28 \times 28}.$$

將此影像拉直成行向量後，會變成長度為 $28 \times 28 = 784$ 的向量：

$$\vec{x} \in \mathbb{R}^{784 \times 1}.$$

因此，所有 MNIST 影像經過向量化後，共同位於一個 784 維的向量空間 $\mathbb{R}^{784 \times 1}$ 中。 □

Exercise 1.54. 一張 3×4 的灰階影像可以視為 $\mathbb{R}^{m \times n}$ 中的矩陣。請問：

- (1) 此影像拉直成行向量後會落在哪個向量空間中？
- (2) 此向量空間的維度為多少？

1.4.6 子空間及其判斷條件

接著，我們定義所謂的子空間。

Definition 1.55 (子空間, Subspace). 設 V 為向量空間，且 $W \subseteq V$ ，若 W 在繼承 V 的運算下仍構成向量空間，則稱 W 是 V 的一個子空間。

在向量空間中，子空間是一個封閉於加法與純量乘法的集合，每個子空間也都具有自己的維度（dimension），也就是「能夠用多少個方向來描述這個空間中的所有向量」。

在數學上，如果要確認 W 是否為 V 的子空間，從定義來驗證不容易，還是需要比較明確的判斷法則。所以， W 為子空間的判別方式如下：

- (1) $\vec{0} \in W$ (包含零向量);
- (2) 若 $\vec{u}, \vec{v} \in W$, 則 $\vec{u} + \vec{v} \in W$ (加法封閉);
- (3) 若 $c \in \mathbb{R}$ 且 $\vec{v} \in W$, 則 $c\vec{v} \in W$ (純量乘法封閉)。

三者必須同時成立, 才能稱 W 為向量空間 V 的子空間。以下我們從 $\mathbb{R}^2$ 與 $\mathbb{R}^3$ 中的直線與平面為例, 幫助同學建立更清晰的空間概念。

Example 1.56. 從幾何的角度來舉例說明：

- (1) 在 $\mathbb{R}^2$ 中, 通過原點的直線為 $\mathbb{R}^2$ 的子空間, 但不通過原點的平移線不是 $\mathbb{R}^2$ 的子空間。
- (2) 在 $\mathbb{R}^3$ 中, 通過原點的平面為 $\mathbb{R}^3$ 的子空間, 但不通過原點的平面不是 $\mathbb{R}^3$ 的子空間。
- (3) 零子空間 $\{\vec{0}\}$ 與整個空間 $\mathbb{R}^n$ 均為 $\mathbb{R}^n$ 的子空間。 □

同學亦可練習以下的練習題, 檢查自己對子空間的瞭解程度有多高。

Exercise 1.57. 下列集合是否為 $\mathbb{R}^2$ 的子空間? 請分別檢驗三條件是否成立。

- (1) 所有通過原點的直線。
- (2) 所有滿足 $x + y = 1$ 的點所構成的集合。
- (3) 所有滿足 $y = 2x$ 的點所構成的集合。

Exercise 1.58. 下列哪些集合是 $\mathbb{R}^3$ 的子空間? 請說明原因。

- (1) 所有 $z = 0$ 的點所構成的集合;
- (2) 所有 $x + y + z = 1$ 的點所構成的集合;
- (3) 零向量;
- (4) 所有 $x = y$ 的點所構成的集合。

1.4.7 子空間與影像處理

然而，子空間這個數學結構可以如何幫助我們處理或理解影像資料？

在處理多張來自監視錄影或連續拍攝的靜態場景影像時，通常會觀察到一個現象：雖然每張影像的像素數可能非常龐大（例如 $640 \times 480 = 307200$ 維），但在多數情況下，這些影像的「整體變化趨勢」其實非常有限。例如，若拍攝的場景為一間辦公室，那麼牆壁、地板、桌子、窗戶等背景結構通常不會改變，真正變動的只是人走進畫面、手部動作、光線微幅變化等局部現象。

從數學的角度來看，這代表：當我們把這些影像轉換成向量後會發現，它們之間其實非常接近，彼此的差異只集中在某些特定的變動方向上。換句話說，這些影像大致上都「落在」一個較小的區域內，而這個區域的形狀與方向，可以用少數幾個代表性的向量來描述。這樣的現象，在數學上對應到一種低維的結構，也就是說這些向量並不需要很多維度就能描述清楚它們的變化。

注意到這裡提到的低維結構就是一個低維度的子空間。

在數學建模中，背景佔據整體影像的大部分，因此會在向量空間中形成一個低維的主結構，也就是「主要資訊」，注意到所謂的主要資訊指的是在影像資料中佔據了主要的變異模式與空間結構的背景。但在應用上，我們的重點是找出那些違反這個主結構的變化，也就是前景，這些變化通常稀疏而明顯，雖然在數量上少，卻是我們真正關心的目標。根據這樣的理解，我們可以將一張影像（經向量化後）表示為：

$$\vec{x} = \vec{x}_{\text{bg}} + \vec{x}_{\text{fg}},$$

其中：

- (1) $\vec{x}_{\text{bg}}$ 落在背景所張成的低維子空間（也稱後景子空間）；
- (2) $\vec{x}_{\text{fg}}$ 為偏離該子空間的成分，通常對應於「異常」、「前景」、「雜訊」或「移動物體」。

這就是本課程將要深入討論的「前後景分離（foreground-background separation）」之基本假設與動機。

Example 1.59. 想像你用固定的攝影機拍攝一間教室，錄下連續 100 張畫面。因為教室的背景不會變，所以這 100 張影像看起來幾乎一樣，只有偶爾有學生經過時畫面才會稍微改變。

如果我們把每張影像攤平成一個長長的向量，就會發現：這些向量都很接近某幾個「固定的模式」，例如：牆壁的亮度、地板的紋路、講桌的位置等等。這些穩定出現的模式，數學上就會構成一個「低維的空間」，可以簡單地用少數幾個基本圖像來描述。

而當某一張影像有學生走進畫面，這張向量就會在某些地方出現「不符合這些基本模式」的部分，也就是說，它和其他背景向量「不太一樣」。這種不同，就是我們感興趣的「前景資訊」或「異常區域」。

這個想法是我們之後要學的「前後景分離」的核心。 $\square$

Example 1.60. 假設我們用定點攝影機拍攝一間教室連續 100 張畫面，大多數影像中的背景幾乎完全一樣，只有少數幾張出現學生走動。若我們將這 100 張影像向量化後，組成一個影像資料矩陣 $M \in \mathbb{R}^{10000 \times 100}$ ，可以發現：大多數影像之間的差異非常小，它們幾乎都位在某個低維子空間 $S \subset \mathbb{R}^{10000}$ 上，這個子空間代表「穩定的背景」。

當一張影像 $\vec{x} \in \mathbb{R}^{10000}$ 中有前景物體（例如學生）出現時，它就會偏離這個子空間。此時我們可以計算：

$$\vec{x}_{\text{bg}} = P_S(\vec{x}), \quad \vec{x}_{\text{fg}} = \vec{x} - \vec{x}_{\text{bg}}$$

來將影像拆解為背景部分（接近穩定）與前景變化部分（可能是人物或異常）。這樣的分離方式，讓我們能有效抓出動態區域，用於行人偵測或異常行為辨識。 $\square$

總結來說，當我們從一堆相似的影像中尋找其中穩定不變的部分（通常是背景），數學上就可以用「子空間」來描述這種穩定結構。這些子空間可以說是影像資料中最重要的共通模式，而那些偏離這些穩定模式的成分，就可能是我們要找的前景、異常、雜訊或動態變化。

Exercise 1.61. 設一張影像 $\vec{x} \in \mathbb{R}^{mn}$ ，已知某子空間 S 由背景影像張成， P_S 為對 S 的正交投影，請說明：

- (1) 如何從 $\vec{x}$ 中分離出背景與前景部分？
- (2) 若 $\vec{x}_{\text{fg}} = \vec{x} - P_S(\vec{x})$ 很小，這表示什麼？
- (3) 若 $\vec{x}_{\text{fg}}$ 很大，又表示什麼？

Exercise 1.62. 考慮一組 100 張影像 $\vec{x}_1, \dots, \vec{x}_{100} \in \mathbb{R}^{3000 \times 1}$ ，其背景幾乎完全相同，僅少部分影像含有前景物體，那麼，這 100 張影像大致會落在哪種維度的子空間？

Chapter 2

An Introduction to Matrix Analysis

2.1 線性相依與線性獨立

在空間中，我們會想知道，一組向量究竟能帶來幾種「不同的方向」？有些向量看起來不同，但其實是沿著同一條線，只是長度不一樣、方向相反或者是上下左右平移。這樣的向量並沒有新增資訊，因為它們可以由某些向量「疊加」或「拉伸」後得到。

給定一組向量，如果當中有的向量可以透過其他向量的線性組合得到，我們說這組向量是線性相依的（Linear Dependent）；相反地，如果每一個向量都提供了新方向，無法由其他向量拼湊出來，我們就說這組向量是線性獨立的（Linear Independent）。

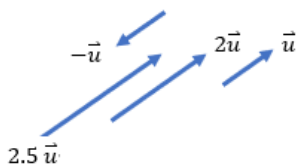


Figure 2.1: 所有向量皆平行（同向或反向），所以這四個向量為線性相依

線性獨立的向量是「擴展空間」的基礎，它們所能張成的空間維度會等於向量的數量；而線性相依的向量雖然看起來多，但其實在空間中只是「重複方向」，無法真正擴張空間的維度。

想像自己要和另兩位朋友一起旅遊，每人提供一個導航方向，規劃你們的移動路線。如果三個人提供的方向彼此之間可以由「加減與倍數」計算出來，例如第二人說「我走的方向就是第一人的兩倍快」，第三人說「我其實是第一人的反方向」，那麼這三個方向之間並沒有真正提供新的資訊。雖然表面上有三個向量，但實際上我們只得到了一個方向的變化方式，其餘只是它的重複或變形，這就是線性相依的概念。

但如果是兩個人去旅遊，規畫移動路線時分別指向「東」與「北」，那麼這兩個方向就各自提供了不同的資訊，它們是線性獨立的。這種情況下，你可以藉由這兩個基本方向，組合出許多其他方向，甚至幾乎到達平面上的任何一點，這也就是線性獨立的優勢所在。

2.1.1 線性相依與線性獨立的定義

請思考以下兩種情況：

- (a) 向量 $\vec{v}_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ 和 $\vec{v}_2 = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$ 指向同一條直線，只是長度不同。此時， $\vec{v}_2 = 2 \vec{v}_1$ ，彼此無法展開出新方向，因此它們是線性相依的。
- (b) 向量 $\vec{v}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ 和 $\vec{v}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ 則分別指向 x 軸正向與 y 軸正向，能張成整個 $\mathbb{R}^2$ ，因此是線性獨立的。

這樣的幾何觀念將是我們理解矩陣秩與子空間結構的重要基礎。

Definition 2.1 (線性相依與獨立). 設 $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_k \in \mathbb{R}^n$ 。若存在非全為零的常數 $c_1, \dots, c_k$ ，使得：

$$c_1 \vec{v}_1 + \dots + c_k \vec{v}_k = \vec{0}, \quad (2.1.1)$$

則稱 $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_k$ 線性相依。反之，若只有當所有 $c_i = 0$ 時 (2.1.1) 式才成立，則稱 $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_k$ 線性獨立。

Example 2.2. 考慮以下三個向量：

$$\vec{v}_1 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \quad \vec{v}_2 = \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix}, \quad \vec{v}_3 = \begin{bmatrix} -1 \\ -2 \\ -3 \end{bmatrix}$$

我們想判斷這三個向量是否線性相依。於是令

$$a \vec{v}_1 + b \vec{v}_2 + c \vec{v}_3 = \vec{0}.$$

將 $\vec{v}_1, \vec{v}_2, \vec{v}_3$ 代入可得

$$a \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + b \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix} + c \begin{bmatrix} -1 \\ -2 \\ -3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

化簡後可得到聯立方程式：

$$\begin{cases} a + 2b - c = 0 \\ 2a + 4b - 2c = 0 \\ 3a + 6b - 3c = 0 \end{cases}$$

這是一組退化的線性方程組，有無限多組非零解，例如：

$$a = 1, \quad b = 0, \quad c = 1 \quad \Rightarrow \quad \vec{v}_1 + \vec{v}_3 = \mathbf{0}.$$

事實上，我們可看出 $\vec{v}_2 = 2 \vec{v}_1$ ，且 $\vec{v}_3 = -1 \vec{v}_1$ ，因此三個向量皆在 $\vec{v}_1$ 所張成的一條線上，故此向量組 $\{\vec{v}_1, \vec{v}_2, \vec{v}_3\}$ 是線性相依的。□

注意到線性相依表示某些向量之間存在重複方向，而線性獨立的向量才真正能提供額外資訊，然後拓展空間。當然，我們也可以利用程式來檢查向量是否線性相依：

```
import numpy as np

# 三個向量作為矩陣的欄向量
V = np.array([[1, 2, -1],
              [2, 4, -2],
              [3, 6, -3]])

# 計算秩 (rank)：若 rank < 向量數，則相依
rank = np.linalg.matrix_rank(V)
print("向量秩為：", rank)

if rank < V.shape[1]:
    print("這些向量是線性相依的。")
else:
    print("這些向量是線性獨立的。")
```

(a) 線性相依性檢查程式碼

```
向量秩為：1
這些向量是線性相依的。
```

(b) 程式執行結果

Figure 2.2: 使用 NumPy 判斷向量是否線性相依的程式與結果

上述程式輸出為「秩為 1」，代表三個向量其實都在同一條線上。從程式執行的結果，我們可看出「秩」是檢查線性相依性的有效工具，之後會有更多深入的討論。

Exercise 2.3. 判斷以下兩小題的向量組合是否線性相依，並說明理由：

$$(1) \begin{bmatrix} 2 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}; \quad (2) \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix}, \begin{bmatrix} 3 \\ 5 \\ 7 \end{bmatrix}.$$

Exercise 2.4. 判斷下列向量組合是否線性相依： $\mathbf{u}_1 = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$ ， $\mathbf{u}_2 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$ ， $\mathbf{u}_3 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$ 。

2.1.2 線性相依與影像資料

先前曾提過，在影像前後景分離的情境中，若多張影像的背景部分幾乎不變，那麼將這些影像向量化後，得到的一群向量彼此之間通常會非常接近，甚至可能互為比例或線性組合。換句話說，這些向量會集中在少數幾個方向上，而不是分散地指向各種不同方向。這樣的特性意味著，我們可以只用少數幾個基本方向就描述出這些背景影像。

在應用上，下面再舉一個可視化線性相依或線性獨立的影像處理實例；這個例題是從影片出發，我們把影片視為許多張靜態圖像（frame）依時間順序連續播放而成，每一張圖像稱為「一幀」，如果影片的播放速率是 30 fps（frames per second），表示每秒會顯示 30 張不同的幀。通常，在處理監視器錄影時，我們常會說「從影片中擷取出 10 幀影像作為分析資料」，意思是，我們從中抽出 10 張靜止畫面來進行更進一步的分析。

Example 2.5（靜態攝影機下的背景建模）。假設用固定攝影機拍攝一個走廊，得到多張連續畫面（例如有 n 張影像，且每張影像為 100×100 灰階圖像，轉為向量為長度 10000 的行向量），我們可以將這些影像整理為一個矩陣 $A \in \mathbb{R}^{10000 \times n}$ ， A 中的每行皆為一幀影像向量。

若背景完全不變（例如空無一人），則所有影像非常相似，這些影像的向量幾乎都是沿著同一個方向，只是亮度略有不同，這樣的向量組合是線性相依的。若有人走過，只會造成部分畫面不同，這使得有些影像變得不再完全是前面幾張的組合，但整體而言，大多數影像仍然看起來差不多，所以向量變化的程度仍然不大。所以，這兩種情形，許多影像向量之間都仍是線性相依的。

以程式來判斷這些影像向量之間是否線性相依的話，可以參考圖 2.3 中的程式碼：

```
import numpy as np
from numpy.linalg import matrix_rank
from skimage.io import imread_collection
from skimage.color import rgb2gray
from skimage.transform import resize

# 讀取多張影像
images = imread_collection('path/*.jpg') # 假設已經是對齊的固定攝影機畫面

# 每張影像轉成向量，建立資料矩陣
A = np.column_stack([
    resize(rgb2gray(img), (100, 100)).flatten()
    for img in images
])

# 判斷向量間是否相依
r = matrix_rank(A)
print(f"影像資料矩陣的秩為 {r}，共 {A.shape[1]} 張影像。")
```

Figure 2.3: 判斷影像向量之間是否線性相依

特別地，若秩遠小於影像張數，表示有很多影像向量線性相依。

□

我們可以把這一節學到的觀念做成以下的表格，方便同學們理解中間的差別：

現象	數學概念	代表意義
多張影像幾乎一樣	向量線性相依	資訊重複，沒有新內容
畫面出現變化（如有人走動）	向量線性獨立	帶來新方向，增加資訊

Table 2.1: 影像變化與線性關係的對應

Exercise 2.6. 假設你有以下三個影像的行向量（每列代表一張簡化過的灰階影像，畫素取值為 0 255）：

$$\mathbf{I}_1 = \begin{bmatrix} 100 \\ 120 \\ 130 \\ 110 \end{bmatrix}, \quad \mathbf{I}_2 = \begin{bmatrix} 200 \\ 240 \\ 260 \\ 220 \end{bmatrix}, \quad \mathbf{I}_3 = \begin{bmatrix} 150 \\ 180 \\ 195 \\ 165 \end{bmatrix}$$

請問這三個行向量是否線性相依？試說明它們之間的關係在影像資料中的意義。

★ 同學們可觀察到上題的 $\mathbf{I}_2 = 2 \cdot \mathbf{I}_1$ ，想一想， $\mathbf{I}_3$ 是否也能由 $\mathbf{I}_1, \mathbf{I}_2$ 組合出來？

Exercise 2.7. 若使用固定攝影機拍攝一段空無一人的走廊影像，連續擷取了 10 張畫面，並將每張影像轉成長度為 10000 的向量。你發現這些向量彼此之間非常相似，幾乎可以用其中一張影像的向量來表示其餘影像。

- (1) 請問這表示這些影像向量之間是線性相依還是線性獨立？為什麼？
- (2) 若第 11 張影像中出現了一位走過走廊的人，你認為這張影像的向量與前面 10 張會是線性相依的嗎？請解釋你的推論。

2.2 基底與維度：如何描述影像資料的方向性

我們在前面討論了向量的線性相依與獨立，接下來的問題是，如果我們手上有一堆向量，那麼要如何判斷這些向量到底「涵蓋了多少不同的方向」？要解決這個問題就需要先介紹一些新的觀念。

2.2.1 什麼是張成？從向量組合看空間的生成

如果我們手上有幾個向量，透過它們的線性組合，究竟可以「走到」多少地方？它們能不能「覆蓋整個平面」或「填滿整個空間」？這個問題的答案，就與「張成 (span)」這個觀念密切相關。

從幾何的角度來看，向量的張成就是：所有可以由這些向量線性組合得到的向量所構成的集合。也就是說，如果你能把一個新向量寫成這幾個向量的加權結果（線性組合），那它就屬於這個「張成的空間」裡。例如：

- (1) 一個向量可以張成一條線；
- (2) 兩個不共線的向量可以張成一個平面；
- (3) 三個不共面的向量可以張成三維空間。

Definition 2.8 (張成, Span). 對向量集合 $\{\vec{v}_1, \vec{v}_2, \dots, \vec{v}_k\}$ ，其張成空間定義為所有線性組合所構成的集合：

$$\text{span}\{\vec{v}_1, \dots, \vec{v}_k\} = \left\{ a_1 \vec{v}_1 + \dots + a_k \vec{v}_k \mid a_i \in \mathbb{R} \right\}.$$

這表示你可以用這些向量「加加乘乘」產生出整個空間裡的所有向量。

Example 2.9. 比方說，考慮向量：

$$\vec{v}_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad \vec{v}_2 = \begin{bmatrix} 3 \\ 4 \end{bmatrix},$$

則 $\vec{v}_1$ 與 $\vec{v}_2$ 所張成的空間 $\text{span}\{\vec{v}_1, \vec{v}_2\}$ 為

$$\text{span}\{\vec{v}_1, \vec{v}_2\} = \left\{ a \begin{bmatrix} 1 \\ 2 \end{bmatrix} + b \begin{bmatrix} 3 \\ 4 \end{bmatrix} \mid a, b \in \mathbb{R} \right\} = \left\{ \begin{bmatrix} a + 3b \\ 2a + 4b \end{bmatrix} \mid a, b \in \mathbb{R} \right\}.$$

這個集合包含所有 $\mathbb{R}^2$ 中的向量，因為 $\vec{v}_1$ 和 $\vec{v}_2$ 是線性獨立的，張成整個平面 $\mathbb{R}^2$ 。如果同學想驗證這個集合是否包含所有 $\mathbb{R}^2$ 中的向量，我們可以從 $\mathbb{R}^2$ 任取一個向量 $\begin{bmatrix} \alpha \\ \beta \end{bmatrix}$ ，然後令

$$\begin{bmatrix} a + 3b \\ 2a + 4b \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix},$$

計算可得到 $a = \frac{2\alpha - \beta}{2}$, $b = \frac{3\beta - 4\alpha}{2}$ ，一定都能找到 a 與 b 的值，所以 $\mathbb{R}^2$ 中所有的向量都可以用 $\vec{v}_1$ 與 $\vec{v}_2$ 的線性組合來表示，這也說明了 $\vec{v}_1$ 和 $\vec{v}_2$ 張成了整個平面 $\mathbb{R}^2$ 。□

Exercise 2.10. 已知向量

$$\vec{v}_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad \vec{v}_2 = \begin{bmatrix} 2 \\ 4 \end{bmatrix},$$

請判斷以下向量是否屬於 $\text{span}\{\vec{v}_1, \vec{v}_2\}$:

$$\vec{w}_1 = \begin{bmatrix} 3 \\ 6 \end{bmatrix}, \quad \vec{w}_2 = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

並說明理由。

Exercise 2.11. 設 $\vec{v}_1 = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}$, $\vec{v}_2 = \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}$, $\vec{v}_3 = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}$ 。

- (1) 試判斷 $\vec{v}_3$ 是否在 $\text{span}\{\vec{v}_1, \vec{v}_2\}$ 中。
- (2) 若是，請找出對應的係數組 (a, b) 使得 $\vec{v}_3 = a \vec{v}_1 + b \vec{v}_2$ 。

2.2.2 什麼是基底與維度？

我們可以這樣想：在一個向量空間中，若每個向量都可以由少數幾個固定方向的向量之線性組合所表示，那麼這些固定方向就扮演了描述整個空間的角色，稱為這個空間的基底 (basis)。而這組基底所含的向量個數，正是這個空間的維度 (dimension)，也就是描述這個空間所需的「獨立方向」的數量。不過，這裡的說法還是籠統了些，我們可以仔細看一下數學定義怎麼說。

Definition 2.12 (基底, Basis). 若向量集合 $\{\vec{v}_1, \vec{v}_2, \dots, \vec{v}_k\}$ 同時滿足以下兩條件：

- (1) 線性獨立；
- (2) 張成整個空間 V ，即 $\text{span}\{\vec{v}_1, \dots, \vec{v}_k\} = V$ ；

則此集合稱為空間 V 的一組基底。

比方說，考慮三個向量：

$$\vec{v}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \vec{v}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \vec{v}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

這三個向量彼此正交、線性獨立，並能張成 $\mathbb{R}^3$ ，因此 $\{\vec{v}_1, \vec{v}_2, \vec{v}_3\}$ 是 $\mathbb{R}^3$ 的一組基底，我們也將這組基底稱為 $\mathbb{R}^3$ 的標準基底 (standard basis)。

接下來再以向量：

$$\vec{v}_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad \vec{v}_2 = \begin{bmatrix} 3 \\ 4 \end{bmatrix},$$

為例，因為 Example 2.9 中已說明了 $\vec{v}_1$ 和 $\vec{v}_2$ 張成了整個平面 $\mathbb{R}^2$ 且線性獨立，所以由定義可知 $\{\vec{v}_1, \vec{v}_2\}$ 為一組 $\mathbb{R}^2$ 中的基底。

Definition 2.13 (維度, Dimension). 向量空間 V 的維度定義為其任一組基底中向量的個數，記為 $\dim(V)$ 。

Example 2.14. 考慮以下三個向量：

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}, \quad \mathbf{v}_2 = \begin{bmatrix} 2 \\ 0 \\ 4 \end{bmatrix}, \quad \mathbf{v}_3 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}.$$

觀察可知， $\mathbf{v}_2 = 2\mathbf{v}_1$ ，所以它不是新的方向，只是原方向的放大。而 $\mathbf{v}_1$ 與 $\mathbf{v}_3$ 則彼此線性獨立，提供兩個獨立方向。因此空間的維度為 2；且一組基底可以是 $\{\mathbf{v}_1, \mathbf{v}_3\}$ 。□

Example 2.15. 我們來看另一組例子：

$$\mathbf{b}_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{b}_2 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{b}_3 = \begin{bmatrix} 1 \\ 2 \\ 1 \\ 0 \end{bmatrix}.$$

我們能發現 $\mathbf{b}_3 = \mathbf{b}_1 + \mathbf{b}_2$ 因此它並非新的方向，這表示只需要兩個向量 $\{\mathbf{b}_1, \mathbf{b}_2\}$ 就可以張成這三個向量所生成的向量子空間，且維度為 2，一組基底為 $\{\mathbf{b}_1, \mathbf{b}_2\}$ 。□

Exercise 2.16. 請判斷下列向量所張成的空間的維度為何？並找出一組基底。

$$\mathbf{w}_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{w}_2 = \begin{bmatrix} 2 \\ 2 \\ 0 \end{bmatrix}, \quad \mathbf{w}_3 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}.$$

同學可以觀察向量間是否存在比例關係，或檢查哪些向量是線性獨立的。

2.2.3 在影像資料中理解維度

在前後景分離的問題中，我們會把每張影像轉換成一個行向量，然後觀察這些行向量之間的關係。如果這些影像彼此很相似，代表它們其實只在少數幾個方向上有變化，所以，

這些影像可以用一小組「代表性方向」來描述，這些方向就是所謂的基底。其他影像只是這些基底的不同的線性組合，就像我們可以用幾個積木就拼出許多不同的圖案一樣。

舉例來說，假設我們把 1 張 100×100 灰階影像切成 100 張 10×10 灰階影像，每一張 10×10 灰階影像都向量化成長度為 100 的行向量，然後把這 100 張影像的行向量排成一個矩陣 $A \in \mathbb{R}^{100 \times 100}$ 。如果這些向量彼此很相似，那麼只需要少數幾個具代表性的向量來描述其他所有的向量，此時也表示影像資料中的變化不多，這些向量所張成的空間維度也會比較小。

```
from skimage import data, color
from skimage.transform import resize
import numpy as np
import matplotlib.pyplot as plt

# 讀取太空人圖像並轉為灰階
img = color.rgb2gray(data.astronaut())

# 縮放成 100x100
img = resize(img, (100, 100))

# 切成 10x10 區塊
patches = []
for i in range(0, 100, 10):
    for j in range(0, 100, 10):
        patch = img[i:i+10, j:j+10].flatten()
        patches.append(patch)

# 將所有 patch 組成矩陣
A = np.column_stack(patches)

print("總共區塊數量:", A.shape[1])
print("基底維度 (rank):", np.linalg.matrix_rank(A))

# 顯示圖片
plt.imshow(img, cmap='gray')
plt.title('轉換後的太空人影像 (100x100)')
plt.axis('off')
plt.show()
```



Python 切割區塊並分析基底維度

太空人原始影像與輸出結果

Figure 2.4: 從圖像切割區塊並觀察其所組成向量的基底維度

Exercise 2.17. 以下是四個向量，分別代表從影像中擷取的四個小區塊，每個區塊都已轉為長度為 5 的行向量：

$$\vec{v}_1 = \begin{bmatrix} 10 \\ 20 \\ 30 \\ 40 \\ 50 \end{bmatrix}, \quad \vec{v}_2 = \begin{bmatrix} 20 \\ 40 \\ 60 \\ 80 \\ 100 \end{bmatrix}, \quad \vec{v}_3 = \begin{bmatrix} 9 \\ 21 \\ 29 \\ 41 \\ 49 \end{bmatrix}, \quad \vec{v}_4 = \begin{bmatrix} 15 \\ 30 \\ 45 \\ 60 \\ 75 \end{bmatrix}$$

(1) 請問這些向量是否有明顯的相似性？是否可能落在某幾個共同方向上？

(2) 根據你的觀察，這些向量可能是線性相依的嗎？請說明理由。

(3) 這樣的現象在影像前景與背景資料中，對維度大小可能有什麼影響？

Exercise 2.18. 我們擷取了一段走廊影片中的 10 張影像，並將每張 100×100 灰階圖像轉為長度 10,000 的向量，組成資料矩陣 $A \in \mathbb{R}^{10000 \times 10}$ 。若我們觀察到這些影像的內容非常相似（例如背景完全一樣，僅亮度略有不同），請問這 10 張影像所構成的空間維度應該會大還是小？請說明理由。若有一兩張影像中突然出現前景物體（如移動中的人），請問維度是否會改變？為什麼？

2.3 什麼是秩？從直觀談起

在正式進入矩陣秩的數學定義前，我們先從直觀的例子來理解「秩 (rank)」的含義，並聚焦在最基礎也最重要的情形：**Rank = 1**，也就是「秩為 1 的矩陣」所代表的意義。

2.3.1 秩的直觀意義

矩陣的「秩」(rank) 可以理解為：這個矩陣中到底包含了多少個彼此真正獨立、不重複的「方向」。換句話說，若矩陣的所有行（或列）都只是某一個固定向量的倍數，那整體其實只沿著同一個方向在變動，這就是秩為 1 的情形。

Example 2.19. 請觀察下列兩個矩陣：

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 4 \\ 2 & 1 & 1 \end{bmatrix}.$$

在矩陣 A 中，第二行是第一行的兩倍，第三行是第一行的三倍，也就是說所有行都只是第一行的倍數，沒有新增的「方向」。這樣的矩陣秩為 1，因為資料只集中在一個方向上。

但在矩陣 B 中，三行之間彼此無法用倍數關係表達，因此它們提供了多個獨立的方向，也就是說， B 的秩大於 1。□

最後我們補充矩陣秩的數學定義。

Definition 2.20 (秩, Rank). 給定一個矩陣 $A \in \mathbb{R}^{m \times n}$ ，其秩 (rank) 可記為 $\text{rank}(A)$ ，其定義為：

$$\text{rank}(A) = \max \{r \in \mathbb{N} \mid A \text{ 有 } r \text{ 個線性獨立的行向量}\}$$

也就是矩陣 A 中所有行向量所張成空間的維度。

事實上，把定義中的行向量全改成列向量來計算矩陣的秩，答案也還是會相同。同一個矩陣不會因為你使用行向量的角度，或是列向量的角度得到不同的秩。

Exercise 2.21. 下列每個矩陣都包含了數個向量（每一行代表一個向量）。請觀察它們之間是否彼此「太相似」，也就是：是否看起來只是某一向量的倍數關係？請選出你認為幾乎都在同一個方向上的情形，並簡要說明理由。

$$(1) \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{bmatrix} \quad (2) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3) \begin{bmatrix} 5 & 5 & 5 \\ 10 & 10 & 10 \\ -5 & -5 & -5 \end{bmatrix} \quad (4) \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 4 \\ 2 & 1 & 1 \end{bmatrix}.$$

Hint: 哪些矩陣的行向量「幾乎只是比例不同，但圖樣一樣」？又，哪些行向量之間差異很大？

2.3.2 秩為 1 的幾何觀點與實作

其實，我們也可以從列的角度來實作，把 3×3 矩陣 A 想像成三張影像（每列是一張 3×1 的灰階影像），它們的圖樣完全相同，只是亮度不同。這表示：這組資料其實只有「一種變化方向」。

Rank

若所有行（或列）都是某個固定向量的倍數，則這些資料實際上只落在一條線上，沒有展開為平面或空間。

Example 2.22 (以列向量的方式來觀察秩的意義). 請觀察下列矩陣：

$$M = \begin{bmatrix} 2 & 4 & 6 \\ 4 & 8 & 12 \\ -2 & -4 & -6 \end{bmatrix},$$

這三列分別是 $\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$ 的倍數（2 倍、4 倍、-2 倍），表示它們其實都在同一個方向上。這表示：這個矩陣只有一種方向變化，因此 **Rank** = 1。同學可參考圖 2.5：

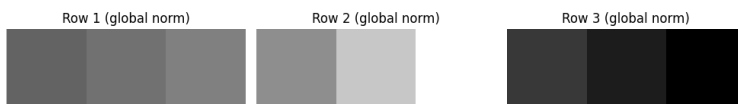


Figure 2.5: 三列為比例關係，轉為灰階影像圖樣完全一致 (Rank = 1)

我們也可以從行向量的角度來看，請參考圖 2.6：

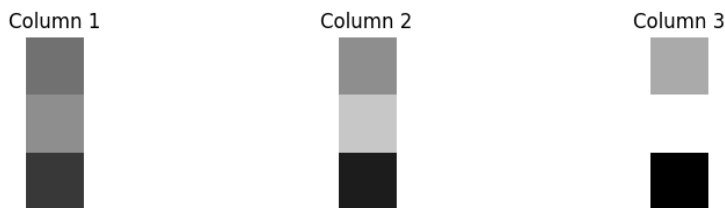


Figure 2.6: 以行向量觀察 Rank = 1 的矩陣，每一行皆為同方向的比例變化

□

Example 2.23. 請判斷矩陣 N 的秩是否為 1，並說明理由。

$$N = \begin{bmatrix} 3 & 6 \\ 6 & 12 \\ 9 & 18 \end{bmatrix}.$$

這個矩陣是 3×2 的非方陣，觀察其兩個行向量：

$$\text{第 2 行} = 2 \times \text{第 1 行} \Rightarrow \begin{bmatrix} 6 \\ 12 \\ 18 \end{bmatrix} = 2 \cdot \begin{bmatrix} 3 \\ 6 \\ 9 \end{bmatrix}$$

所以，所有的行向量都落在同一個方向上，即這個矩陣的所有資訊都集中在一個方向，這就是 **Rank = 1** 的矩陣。

我們也可以觀察列向量的情形：

$$\begin{bmatrix} 3 & 6 \end{bmatrix}, \quad \begin{bmatrix} 6 & 12 \end{bmatrix}, \quad \begin{bmatrix} 9 & 18 \end{bmatrix},$$

它們也彼此成比例，**Rank = 1**，所以不管從哪個角度分析，這個矩陣的行和列都只佔據一個方向，rank 的值不會因為我們習慣使用行向量或是使用列向量來分析而有所不同。□

Exercise 2.24. 請判斷下列矩陣是否為 Rank = 1 的矩陣，並說明它對應的影像現象。

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{bmatrix}.$$

Exercise 2.25. 下列哪一個矩陣是 Rank = 1？

$$(A) \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}; \quad (B) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad (C) \begin{bmatrix} 2 & 2 \\ 3 & 3 \end{bmatrix}; \quad (D) \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}.$$

說明你的選擇，並說明它們分別對應的影像現象。

2.3.3 向量的 Outer Product

我們可以透過程式建立一個簡單的 **Rank = 1** 矩陣。例如，取兩個向量，分別當作「行」與「列」方向，將它們相乘（做 outer product）所形成的矩陣，自然就是 **Rank = 1**。這是因為「整個矩陣都是由同一組向量的組合所張成的」。注意到 outer product 不是矩陣乘法中的點積（dot product），而是行乘列（column-by-row）產生一個完整的矩陣。如果我們寫程式視覺化這個結果（可畫成灰階圖）會發現每一列的圖樣都一致，只是亮度不同。

Example 2.26. 設行向量 $\vec{u}$ 及列向量 $\vec{v}$ 分別如下

$$\vec{u} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \quad \vec{v} = [4 \ 5 \ 6].$$

令 $A = \vec{u} \cdot \vec{v}$ ，也就是做 outer product：

$$\begin{aligned} A &= \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \cdot [4 \ 5 \ 6] = \begin{bmatrix} 1 \cdot [4 \ 5 \ 6] \\ 2 \cdot [4 \ 5 \ 6] \\ 3 \cdot [4 \ 5 \ 6] \end{bmatrix} \\ &= \begin{bmatrix} 1 \cdot 4 & 1 \cdot 5 & 1 \cdot 6 \\ 2 \cdot 4 & 2 \cdot 5 & 2 \cdot 6 \\ 3 \cdot 4 & 3 \cdot 5 & 3 \cdot 6 \end{bmatrix} = \begin{bmatrix} 4 & 5 & 6 \\ 8 & 10 & 12 \\ 12 & 15 & 18 \end{bmatrix}, \end{aligned}$$

這個矩陣的每一列都是 $[4 \ 5 \ 6]$ 的倍數，代表它只有「一個方向」，所以

$$\text{rank}(A) = 1.$$

這是一個由兩個向量用 outer product 生成的 **Rank = 1** 矩陣。 □

注意到 $\vec{u} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$ 負責的是控制亮度， $\vec{v} = [4 \ 5 \ 6]$ 負責的是圖樣本身，然後使用 outer product 就是在把 $\vec{v}$ 的每一列改變亮度，其程式的實作與觀察如下：

```

import numpy as np
import matplotlib.pyplot as plt

# 與圖片一致: u 是行向量 (垂直), v 是列向量 (水平)
u = np.array([[1], [2], [3]]) # shape: (3, 1)
v = np.array([[4, 5, 6]]) # shape: (1, 3)

# Outer product: 行向量 @ 列向量 = 3x3 矩陣
A = u @ v

# 視覺化每一列
fig, axes = plt.subplots(1, 3, figsize=(8, 2))
for i in range(3):
    row_img = A[i, :].reshape(1, 3)
    axes[i].imshow(row_img, cmap='gray', vmin=A.min(), vmax=A.max())
    axes[i].axis('off')
    axes[i].set_title(f'Row {i+1}')

plt.suptitle("由圖中向量生成的外積矩陣 (rank = 1)")
plt.tight_layout()
plt.show()

```

Figure 2.7: 建立 rank = 1 的矩陣

由於每一列都只是 $[4 \ 5 \ 6]$ 的倍數，這些灰階影像只是亮度比例不同，但圖樣完全一致。這正是 Rank = 1 的幾何視覺意義：只有一種「方向變化」。請參考視覺化後的效果，如圖 2.8 所示：



Figure 2.8: 由向量外積產生的 Rank = 1 矩陣（列圖樣為比例關係）

同學們應該會發現 $\vec{u} \cdot \vec{v}$ 就是在算 $\mathbb{R}^{m \times 1}$ 中與 $\mathbb{R}^{1 \times n}$ 中的矩陣相乘，理論上使用第一章學過的矩陣乘法概念就好，為什麼還要特別定義 outer product 的概念呢？原因是，一般矩陣乘法可以是任意維度匹配的乘法，而 outer product 專指「以一個行向量乘上一個列向量，得到一個 Rank = 1 的矩陣」這樣的「乘法」。事實上，因為

$$A = \vec{u} \cdot \vec{v} = \begin{bmatrix} 1 \cdot 4 & 1 \cdot 5 & 1 \cdot 6 \\ 2 \cdot 4 & 2 \cdot 5 & 2 \cdot 6 \\ 3 \cdot 4 & 3 \cdot 5 & 3 \cdot 6 \end{bmatrix} = \begin{bmatrix} 4 & 5 & 6 \\ 8 & 10 & 12 \\ 12 & 15 & 18 \end{bmatrix},$$

這個矩陣的每一列都是 $\vec{v}$ 的倍數，每一行也都是 $\vec{u}$ 的倍數，因此整個矩陣只包含「一個方向」的資訊，也就是 Rank = 1。這樣的 outer product 結構具有高度的規則性：行與列皆成比例，整體變化集中在一個方向上。因此，它經常被用作低秩近似（low-rank approximation）與資料壓縮中的「基本單元」，可以用來近似更高秩矩陣的基本單元。越多個 outer product 疊加，我們就越能重建出原始資料的結構。

最後，雖然還沒有教大家如何用系統性的方式計算矩陣的 rank，我們依然可以補充一個 **Rank = 2** 的例子讓同學們比較。



Figure 2.9: 列向量含有兩個獨立方向 (Rank = 2)

這張圖展示了矩陣

$$M = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 1 & 0 & 1 \end{bmatrix}$$

的三列對應的 1×3 灰階影像。同學們可以發現，前兩列為比例關係，但第三列圖樣與前兩列不同，代表它帶入了一個新的獨立方向。因此這個矩陣所張成的空間不再是一條線，而是一個平面，對應到 **Rank = 2**。

下一節，我們將正式定義「矩陣的秩」，並學會用列簡化 (row echelon form) 的方式來計算它。

Exercise 2.27. 下列哪些矩陣是 **Rank = 1**？請觀察其行或列是否皆為某一向量的倍數（即所有向量共線）。請注意：這些矩陣不一定是方陣。

(1) $\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$; (2) $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$; (3) $\begin{bmatrix} 3 & 3 \\ 6 & 6 \\ 9 & 9 \end{bmatrix}$; (4) $\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \end{bmatrix}$; (5) $\begin{bmatrix} 5 \\ 10 \\ -5 \\ -10 \end{bmatrix}$; (6) $\begin{bmatrix} 1 & 2 \\ 3 & 5 \end{bmatrix}$.

2.4 矩陣的秩

秩 (rank) 是矩陣最重要的數值特徵之一，反映了其列（或行）向量的線性獨立性與所張成的空間維度。理解秩不僅有助於掌握解線性方程組的條件，更能幫助我們辨識一個矩陣在資訊表示上的「有效維度」。

在影像處理中，矩陣的秩可以幫助我們回答以下問題：

- (1) 這些影像畫面彼此「有多大差異」？
- (2) 有哪些重複性高、可壓縮的內容？

(3) 哪些部分是前景中真正的異常變化？

這些觀察都可以透過矩陣的秩來量化，而這也正是「背景為低秩、前景為稀疏」這類模型的數學依據。

2.4.1 矩陣秩的定義與計算方式

我們在 Definition 2.20 提過，矩陣 $A \in \mathbb{R}^{m \times n}$ 的秩 $\text{rank}(A)$ 是其列向量或行向量中線性獨立向量的最大數目；事實上，我們也可以定義為矩陣轉換為列階梯形式（row echelon form）後，其非零列的個數。從先前一些例子，我們觀察到下列的定理成立：

Theorem 2.28 (行秩等於列秩). 對任意矩陣 $A \in \mathbb{R}^{m \times n}$ ，其行向量所張成的子空間與列向量所張成的子空間具有相同維度，即：

$$\text{rank}_{\text{row}}(A) = \text{rank}_{\text{col}}(A).$$

因此，矩陣秩只需計算行或列其中一者即可。

Example 2.29 (以列階梯形式判斷矩陣的秩). 考慮矩陣：

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 1 & 1 & 0 \end{bmatrix},$$

對其進行初等列運算，第 i 列以 R_i 來表示：

(1) 第一列乘以 -2 倍再加入到第二列：

$$\boxed{R_2} \rightarrow -2R_1 + R_2 = [0 \ 0 \ 0].$$

(2) 第一列乘以 -1 倍再加入到第三列：

$$\boxed{R_3} \rightarrow -R_1 + R_3 = [0 \ -1 \ -3].$$

(3) 交換第二列與第三列，得到：

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & -1 & -3 \\ 0 & 0 & 0 \end{bmatrix}. \quad (2.4.1)$$

這個計算過程就叫做「將矩陣轉換為列階梯形式 (row echelon form)」，最後的矩陣 (2.4.1) 有兩列為非零列，因此：

$$\text{Rank}(A) = 2.$$

這表示矩陣 A 的列或行只含有兩個線性獨立的向量，其所張成的空間為一個 $\mathbb{R}^3$ 中的 2 維平面。 $\square$

Example 2.30. 考慮矩陣：

$$B = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 6 & 9 \\ 0 & 1 & 1 & 1 \end{bmatrix} \in \mathbb{R}^{3 \times 4}.$$

我們先將第一列乘以 -2 加到第二列，得到：

$$\boxed{R_2} \rightarrow R_2 - 2R_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}.$$

交換第二列與第三列，得到最終列階梯形式為：

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

有三列為非零列，所以 $\text{Rank}(B) = 3$ ，這表示矩陣 B 中有三個線性獨立的行（或列）向量，其所張成的是一個三維子空間。 $\square$

Exercise 2.31. 請手動對下列矩陣進行初等列運算，並判斷其秩：

$$A = \begin{bmatrix} 1 & 3 & 2 \\ 2 & 6 & 4 \\ 0 & -1 & 2 \end{bmatrix}.$$

Exercise 2.32. 以下三個影像向量：

$$\mathbf{I}_1 = \begin{bmatrix} 50 \\ 80 \\ 100 \end{bmatrix}, \quad \mathbf{I}_2 = \begin{bmatrix} 100 \\ 160 \\ 200 \end{bmatrix}, \quad \mathbf{I}_3 = \begin{bmatrix} 55 \\ 85 \\ 105 \end{bmatrix}.$$

請觀察這三個向量是否線性相依？構成的矩陣秩為多少？說明你的推論。

2.4.2 低秩背景與稀疏前景模型

在每天的監視錄影、醫學影像或自動駕駛中，我們經常需要從畫面中「分離出真正重要的變化」。例如教室的背景牆壁、黑板、桌椅都不會改變，但走進畫面的人，是重要的前景資訊，這些畫面可以視為由兩部分構成：

- (1) 背景：穩定、重複、類似的東西（例如靜止的場景、光線變化，傾向整體的狀況）；
- (2) 前景：出現、移動、擾動的東西（例如人物或車輛，通常是局部現象）。

首先複習一下多張影像的標準矩陣建構方式：假設我們從固定方向的監視錄影中擷取了 k 張灰階影像（每張大小為 $m \times n$ ），則可以對每張影像進行向量化（vectorization），轉成長度為 mn 的行向量 $\vec{I}_i$ （其中 $1 \leq i \leq k$ ），於是我們得到一個由 k 張影像所構成的矩陣

$$M = \begin{bmatrix} \vec{I}_1 & \vec{I}_2 & \dots & \vec{I}_k \end{bmatrix} \in \mathbb{R}^{mn \times k}.$$

在這個表示法下，每一行的元素都對應著「影像中同一個像素位置在不同時間點的變化」。從數學的角度來看，我們可以將一張影像（或一組影像序列）表示為一個矩陣 M ，並試著將它分解成兩個部分：

$$M = L + S, \quad (\text{低秩與稀疏結構分解模型})$$

其中：

- (1) 矩陣 L 代表「背景部分」，畫面內容穩定、變化少，結構重複性高，因此在數學上稱為低秩（low-rank）；
- (2) 矩陣 S 則代表「前景或擾動部分」，例如移動中的人影、閃動的光點，這些只出現在局部位置，因此被視為稀疏（sparse）。

這種分解方法被稱為低秩-稀疏分解（low-rank-sparse decomposition），概念如圖所示：

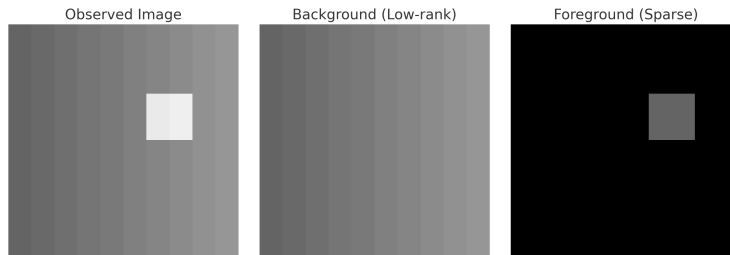


Figure 2.10: 將畫面分解為低秩背景與稀疏前景

其中，Observed Image 是原始觀察到的影像；Background (Low-rank) 是穩定的背景結構；Foreground (Sparse) 則為只有局部變化的前景（例如人物或動態物件）。

即使我們尚未學到完整的演算法，也能先從數學結構直觀理解這個模型的核心想法。

Example 2.33. 假設我們有一個簡單的 5×6 灰階影像序列矩陣 M ，且每一行 $\vec{v}_i$ 是一張影像，此矩陣由 6 張影像所構成：

$$M = [\vec{v}_1 \quad \vec{v}_2 \quad \cdots \quad \vec{v}_6] = \begin{bmatrix} 100 & 100 & 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 & 100 & 255 \end{bmatrix}.$$

從 M 中可觀察到，每一行是一張 5×1 的影像，而第 6 張影像（即第 6 行）在第 5 個像素有一個異常值 255，這代表這個位置所對應的影像位置，有前景物體正經過這裡。接著，我們可將 M 分解為 L 與 S ：

(a) 背景矩陣（低秩部分）：

$$L = \begin{bmatrix} 100 & 100 & 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 & 100 & 100 \end{bmatrix} \Rightarrow \text{rank}(L) = 1.$$

這個矩陣代表「每一張影像都幾乎相同」，所有行向量都一樣，所以秩為 1；注意到秩為 1 就是一種低秩的情形。

(b) 稀疏矩陣（前景異常部分）：

$$S = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 155 \end{bmatrix}$$

其中只有一個非零值 $255 - 100 = 155$ ，表示在第 6 張影像的第 5 個像素，有一個前景物體造成的變動，此矩陣其餘位置皆為零，我們將這樣的矩陣稱為稀疏矩陣 (sparse)。

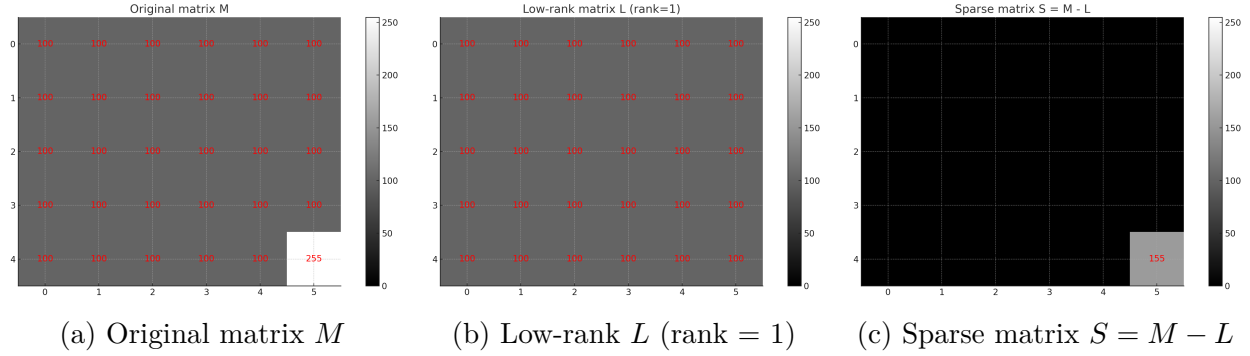


Figure 2.11: Matrix decomposition $M = L + S$ with rank-1 background: original data M , low-rank background L , and sparse anomaly S .

在這個例子中，我們可以看到向量 $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_6$ 是線性相依的，其中 $\vec{v}_1 = \vec{v}_2 = \dots = \vec{v}_5$ ，而 $\vec{v}_6$ 提供了一個與 $\vec{v}_1$ 不同的方向資訊，所以這 6 個向量的張成空間為

$$\begin{aligned} \text{span}\{\vec{v}_1, \dots, \vec{v}_6\} &= \left\{ a_1 \vec{v}_1 + \dots + a_6 \vec{v}_6 \mid a_i \in \mathbb{R}, 1 \leq i \leq 6 \right\} \\ &= \left\{ a \vec{v}_1 + b \vec{v}_6 \mid a, b \in \mathbb{R} \right\}, \end{aligned}$$

向量 $\vec{v}_1$ 與 $\vec{v}_6$ 線性獨立，所以矩陣 M 的秩為 2（這對矩陣來說也是低秩的情形），而 $\text{span}\{\vec{v}_1, \dots, \vec{v}_6\}$ 是一個低維子空間（維度為 2）。 $\square$

Exercise 2.34 (前景擾動與秩的觀察). 假設我們用靜止攝影機錄下一個空教室畫面，連續擷取了 10 張畫面，每張轉成向量後組成資料矩陣 $M \in \mathbb{R}^{10000 \times 10}$ 。

- (1) 如果這 10 張畫面幾乎完全相同，只有亮度微幅變化，則 M 的秩會高還是低？
- (2) 某天第 5 張畫面中突然有人走入教室，會對 M 的秩造成什麼影響？為什麼？
- (3) 若我們用模型 $M = L + S$ ，請說明哪部分代表背景？哪部分代表前景？

Exercise 2.35. 假設我們用靜止攝影機錄下一個空教室畫面（背景穩定），並將 20 張連續畫面向量化後組成矩陣 $M \in \mathbb{R}^{10000 \times 20}$ 。某天其中一張畫面中出現了一名走進教室的人。

- (1) 若 $\text{rank}(L) = 1$ ，你認為這表示什麼？
- (2) 稀疏矩陣 S 中的非零元素會集中在哪些影像畫面？為什麼？

2.4.3 低秩矩陣的意義

從數學上來說，秩代表這個矩陣能被多少組線性獨立的向量「建構」出來。如果某個矩陣的秩為 1，就表示它的所有行（或列）向量都落在一個共同方向上。由於，矩陣不會有秩比 1 還低的情形，所以秩為 1 的矩陣是一種低秩矩陣。

從影像來讀解的話，試想一段背景穩定的監視器影片畫面，我們從影片中擷取出的每張影像通常只會有局部的變化，此時，當我們把這多張畫面向量化後組成一個大矩陣時，這個大矩陣的每一列或每一行往往彼此相似，向量間線性相依的程度很高，矩陣的秩就會很低。

這樣的矩陣我們稱為「低秩矩陣 (low-rank matrix)」，它表示這些影像資料內部具有高度的冗餘性，而在影像處理中，冗餘性常常表現為畫面中大量重複或變化不大的部分，所以，從向量的角度來看，我們可以用較少的方向（特徵）來描述矩陣的主要結構。例如有一個 3×6 灰階影像序列矩陣 M ，且每一行 $\vec{v}_i$ 是一張影像，此矩陣由 6 張影像所構成：

$$M = [\vec{v}_1 \quad \vec{v}_2 \quad \cdots \quad \vec{v}_6] = \begin{bmatrix} 50 & 100 & 120 & 110 & 100 & 100 \\ 50 & 100 & 120 & 100 & 100 & 100 \\ 50 & 100 & 120 & 100 & 100 & 160 \end{bmatrix}.$$

可以看出 $\vec{v}_2 = \vec{v}_5 = 2 \vec{v}_1$ 且 $\vec{v}_3 = 2.4 \vec{v}_1$ ，然後 $\vec{v}_1$ 、 $\vec{v}_4$ 、 $\vec{v}_6$ 三個向量線性獨立，矩陣 M 的每一個行向量都還算相似，6 個向量間線性相依的程度很高，所以這 6 個向量的張成空間可表示為

$$\begin{aligned} \text{span}\{\vec{v}_1, \dots, \vec{v}_6\} &= \left\{ a_1 \vec{v}_1 + \cdots + a_6 \vec{v}_6 \mid a_i \in \mathbb{R}, 1 \leq i \leq 6 \right\} \\ &= \left\{ a \vec{v}_1 + b \vec{v}_4 + c \vec{v}_6 \mid a, b, c \in \mathbb{R} \right\}, \end{aligned}$$

張成空間維度為 3，意味著矩陣 M 的秩為 3，且我們可以進行更進一步的分解：

$$M = L + S = \begin{bmatrix} 50 & 100 & 120 & 100 & 100 & 100 \\ 50 & 100 & 120 & 100 & 100 & 100 \\ 50 & 100 & 120 & 100 & 100 & 100 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 60 \end{bmatrix},$$

矩陣 L 顯然秩為 1（也是低秩），它是背景，而矩陣 S 可以讓我們看到影片在什麼時間點的什麼位置有局部變化。

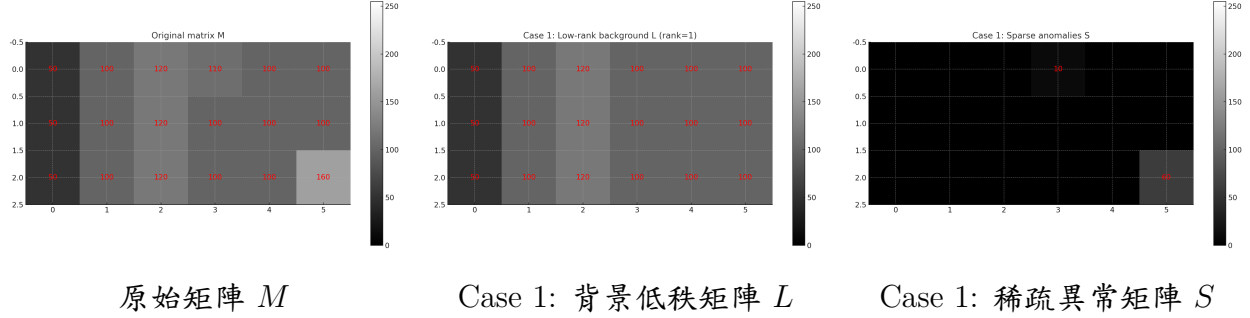


Figure 2.12: 以秩 1 背景模型進行分解後的三張圖比較：原始影像矩陣 M 、低秩背景 L 、稀疏異常 S 。

不過，這個例子有另一種可以解讀的方式，假設我們取 $\vec{v}_4 = [110 \ 100 \ 100]^\top$ ，那個 110 可能只是有窗戶，導致影片在某個時間點只在那個位置光影有微小的變化，此時我們的分解可寫為

$$M = L + S = \begin{bmatrix} 50 & 100 & 120 & 110 & 100 & 100 \\ 50 & 100 & 120 & 100 & 100 & 100 \\ 50 & 100 & 120 & 100 & 100 & 100 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 60 \end{bmatrix},$$

矩陣 L 秩為 2（也仍是低秩），而矩陣 S 還是可以讓我們看到影片在什麼時間點的什麼位置有局部變化，所以，背景 L 怎麼取與我們實際要處理的問題有關，這也就是為什麼有時我們取 L 秩為 1 的情形，有時只單純需要 L 是低秩（不一定要秩為 1）就好，完全看實際的問題來決定。

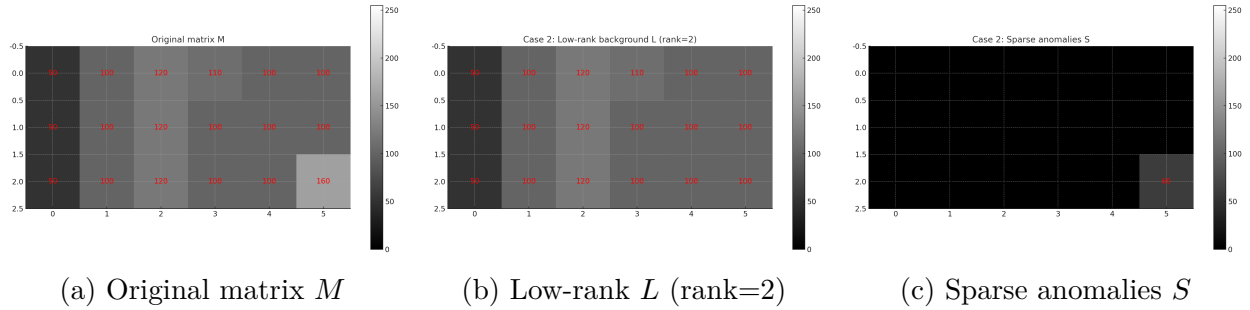


Figure 2.13: Matrix decomposition under Case 2 (assumed background of rank 2): $M = L + S$.

Example 2.36. 假設我們在一分鐘內用同一個角度拍下 20 張空教室的畫面，每張畫面大小為 100×100 像素。我們將這些畫面向量化，組成矩陣 $M \in \mathbb{R}^{10000 \times 20}$ 。在教室背景完全不變的前提下，那麼這 20 張畫面是完全一樣的，矩陣 M 的每一行都是一樣的向量，也就是秩為 1。

即使這一分鐘內教室的光線稍有不同，一般而言秩也不會太高，可能仍小於 5。這意味著：我們可以用很少的「方向」來描述整個教室的樣貌。 □

最後，我們還是列出低秩矩陣的數學定義以供同學們參考：

Definition 2.37 (低秩矩陣, Low-Rank Matrix). 若矩陣 $A \in \mathbb{R}^{m \times n}$ 的秩滿足 $\text{rank}(A) = r \ll \min(m, n)$ ，則稱 A 為低秩矩陣。

Exercise 2.38. 試觀察下列矩陣的行或列是否具有「重複或線性關聯」的現象，並判斷其是否為低秩矩陣。若是，請指出秩是多少。

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{bmatrix}$$

2.4.4 稀疏矩陣觀念介紹

在影像處理或資料科學中，我們常常遇到這樣的情況：「一個矩陣中絕大多數的元素都是 0，只有少數幾個地方是非零數值。」這種矩陣我們稱為稀疏矩陣 (sparse matrix)，也可以想成，稀疏矩陣就是「幾乎都是空白的資料表格」。

Definition 2.39 (稀疏矩陣, Sparse Matrix). 設矩陣 $A \in \mathbb{R}^{m \times n}$ ，若其中非零元素的數量遠小於總元素數 mn ，則稱 A 為稀疏矩陣。

以我們多次提到的監視器影片為例，大部分時間點畫面都只有一個空教室，偶爾有人會在畫面中經過。如果我們把這些畫面中「有改變」的部分（例如有人經過）挑出來並記錄在向量中，會發現大部分像素值仍然是 0，只有「有人的位置」會有數值，擷取多張影像向量化並組合，就會形成一個稀疏的矩陣。比方說，

$$S = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 255 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 128 \end{bmatrix},$$

這是某張影像的前景遮罩：其中非零值表示人物出現的位置，其他地方都是背景（用 0 表示）。整體來看幾乎都是 0，只有「重要的點」才是非零。

那麼，該怎麼量化一個矩陣的「稀疏程度」呢？我們可以定義一個數字來描述這個矩陣有多稀疏，這就是所謂的稀疏度 (sparsity level)：

$$\text{sparsity} = 1 - \frac{\text{非零元素個數}}{\text{總元素個數}}.$$

以上面矩陣為例，它是 4×4 的矩陣，共 16 個元素，只有 2 個非零值，所以：

$$\text{sparsity} = 1 - \frac{2}{16} = 0.875.$$

這代表有 87.5% 的位置都是 0，是一個高度稀疏的矩陣。

Example 2.40.

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

此矩陣中僅有兩個非零元素，因此 sparsity 為 $1 - \frac{2}{16} = 0.875$ ，表示有 87.5% 的元素為零。

稀疏矩陣之所以重要，是因為它們可以幫助我們更有效地處理與儲存資料。首先，由於矩陣中大部分的元素都是 0，只需要記錄少數有意義的位置，就能大幅節省記憶體空間。其次，稀疏矩陣讓我們更容易找出影像中的「重點」，像是人物、物體等通常只佔據畫面的一小部分，對應到的正是那些非零值。最後，稀疏結構也有助於資料壓縮與傳輸，因為我們只需要處理那些真正有資訊的地方。

Exercise 2.41. 下列矩陣中，哪一個是稀疏矩陣？請簡單說明理由。

$$(1) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (2) \begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 0 \\ 4 & 5 & 6 \end{bmatrix}.$$

Exercise 2.42. 一張影像經處理後，得到一個矩陣 A ，其中只有 5% 的像素是非零值。請問此矩陣是否為稀疏矩陣？你如何估算它的稀疏程度？

