

6

Block Matrix

分块矩阵

将大矩阵切成小块，简化运算



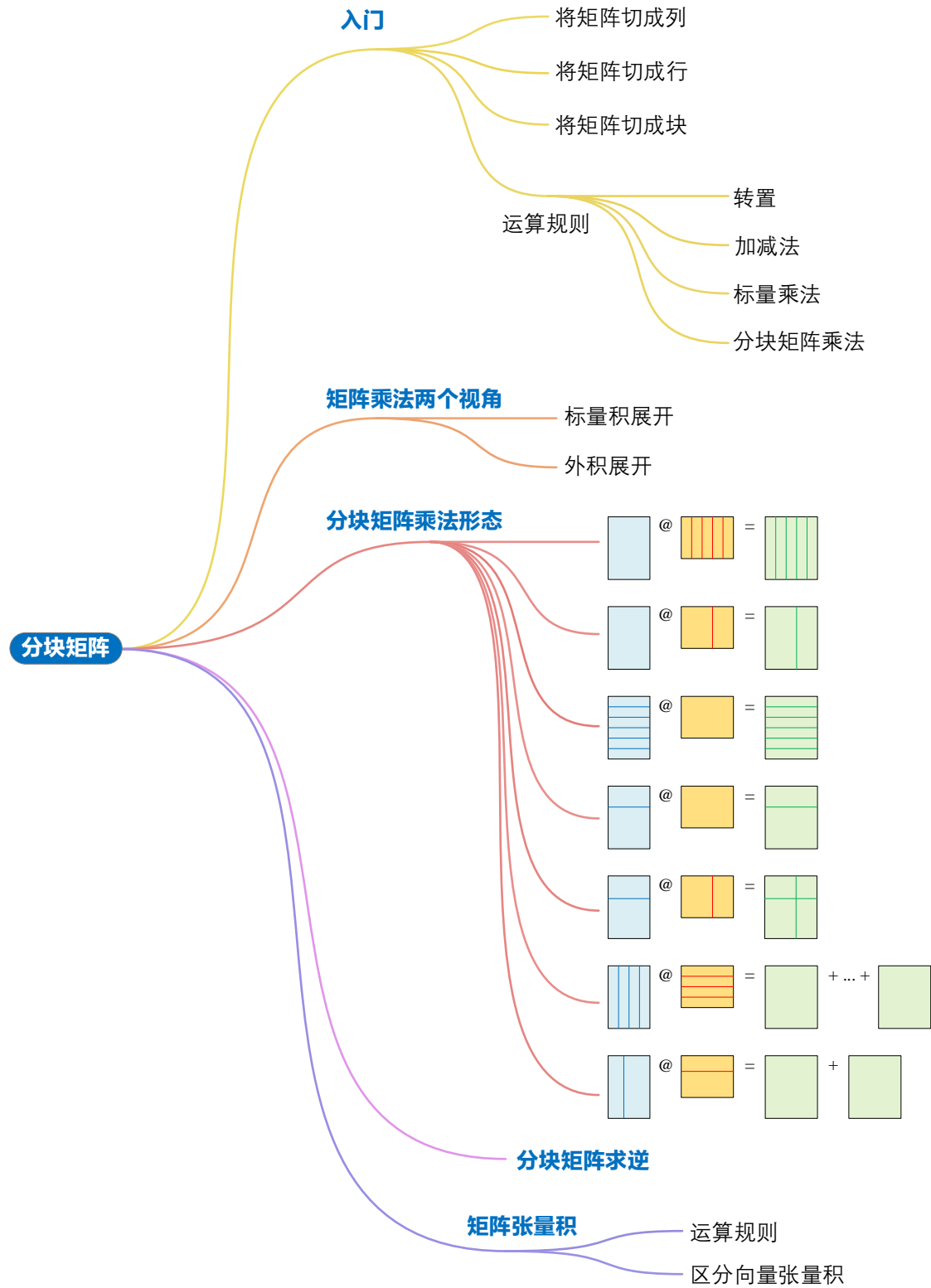
数学的精髓在于自由。

The essence of mathematics is in its freedom.

—— 格奥尔格·康托尔 (Georg Cantor) | 德国数学家 | 1845 ~ 1918



- ◀ `numpy.kron()` 计算矩阵张量积
- ◀ `numpy.random.random_integers()` 生成随机整数
- ◀ `numpy.zeros_like()` 用来生成和输入矩阵形状相同的零矩阵
- ◀ `seaborn.heatmap()` 绘制热图



6.1 分块矩阵：横平竖直切豆腐

分块矩阵 (block matrix 或 partitioned matrix) 将一个矩阵用若干条横线和竖线分割成多个**子块矩阵** (submatrices)。矩阵分块后可以简化运算，同时让运算过程变得更加清晰。

白话讲，矩阵分块好比横平竖直切豆腐；但是下刀的手法很有讲究，这是本章后文要着重探讨的内容。

切丝、切条

实际上，本书一开始就已经不知不觉地使用了分块矩阵这一重要工具。

大家已经清楚知道，如图 1 所示，矩阵 X 可以看做是由一系列行向量或列向量按照一定规则构造而成。这实际上体现的就是分块矩阵的思想。

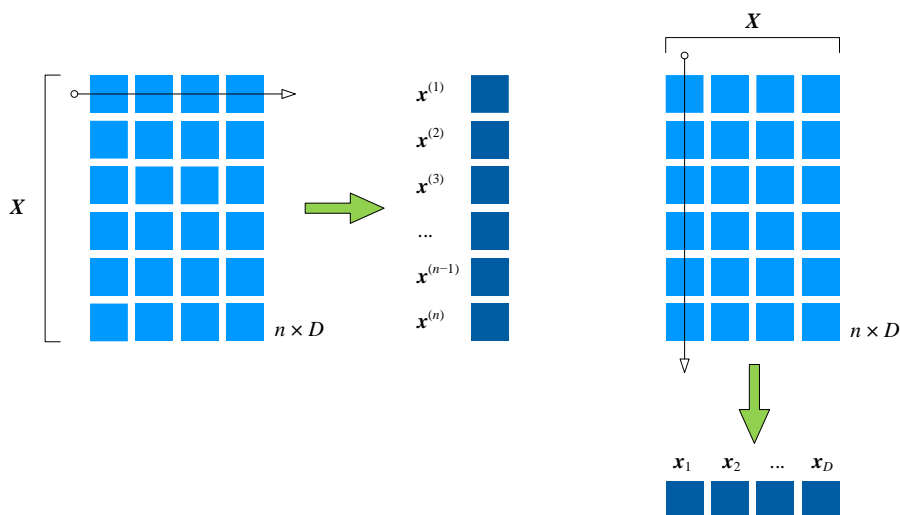


图 1. 矩阵可以写成一系列行向量或列向量

矩阵 X 每行之间切一刀，得到一组行向量：

$$X_{n \times D} = \begin{bmatrix} \mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \\ \vdots \\ \mathbf{x}^{(n)} \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,D} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,D} \end{bmatrix}$$

(1)

矩阵 X 在每列之间切一刀，将 X 切成一组列向量：

$$\mathbf{X}_{n \times D} = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_D] = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,D} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,D} \end{bmatrix} \quad (2)$$

切块

下面介绍分块矩阵其他切法。给出如下矩阵 \mathbf{A} ：

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 & 0 & 0 \\ 4 & 5 & 6 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

我们把矩阵 \mathbf{A} 横竖都切一刀，得到四个子矩阵：

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 & 0 & 0 \\ 4 & 5 & 6 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

给每个子矩阵起个名字，矩阵 \mathbf{A} 记做：

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} \end{bmatrix} \quad (5)$$

也就是，

$$\begin{aligned} \mathbf{A}_{1,1} &= \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, \quad \mathbf{A}_{1,2} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \\ \mathbf{A}_{2,1} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{A}_{2,2} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned} \quad (6)$$

本书后文也会用行、列数来命名分块矩阵，比如：

$$\mathbf{X}_{n \times D} = \begin{bmatrix} \mathbf{X}_{r \times q} & \mathbf{X}_{r \times (D-q)} \\ \mathbf{X}_{(n-r) \times q} & \mathbf{X}_{(n-r) \times (D-q)} \end{bmatrix} \quad (7)$$



Numpy 中矩阵分块可以用指定行、列序号就做到。numpy.block() 函数可以用子块矩阵结合得到原矩阵。请大家参考 Bk4_Ch6_01.py。

鸢尾花数据为例

如图2，将鸢尾花数据矩阵 X 上下切两刀，均匀分成三块。这三个分块矩阵的大小都是 50×4 。本书第1章提到，鸢尾花数据有三个亚属，即三类标签——**山鸢尾** (setosa)、**变色鸢尾** (versicolor) 和**维吉尼亚鸢尾** (virginica)。图2右侧的每个分块代表一类鸢尾花的样本数据子集，每个子集各有50条记录。利用图2右侧的分块矩阵，我们可以分析某一类鸢尾花样本子集的均值、质心(列均值构成的向量)、方差、均方差、协方差、协方差矩阵、相关性系数、相关性系数矩阵等等。

大家将会在本书第22章，以及本系列丛书《概率统计》和《数据科学》两册中看到图2这种分块方式的用途。

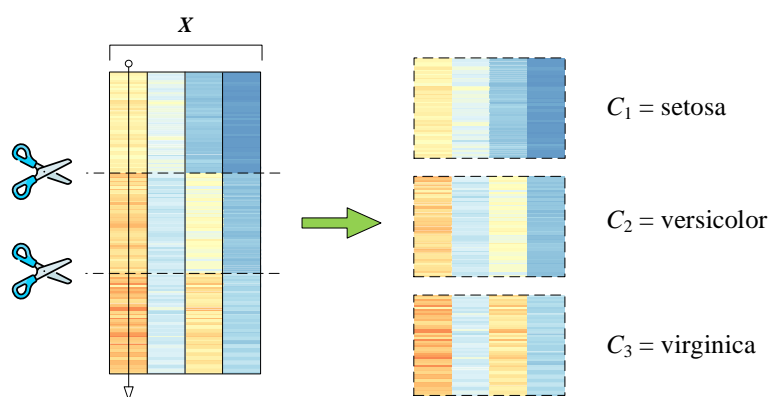


图2. 鸢尾花数据矩阵上下切2刀分成3块

如图3所示，将鸢尾花数据矩阵 X 左右切3刀，得到4个分块矩阵，即4个列向量，形状都为 150×1 。这4个分块矩阵分别代表**花萼长度** (sepal length)、**花萼宽度** (sepal width)、**花瓣长度** (petal length) 和**花瓣宽度** (petal width) 四个特征的样本数据。

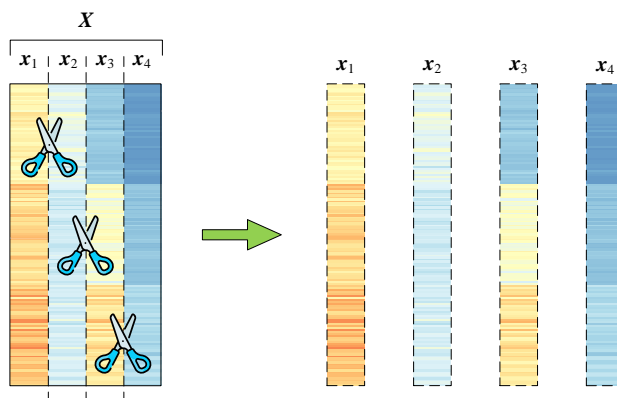


图3. 鸢尾花数据矩阵左右切3刀分成4块

转置

一般情况， $A_{i,j}$ 的行数记做 n_i ，列数为 D_j ；如果矩阵 A 的形状为 $n \times D$ ，按 (5) 分割得到的子块矩阵的行、列数满足：

$$n_1 + n_2 = n, \quad D_1 + D_2 = D \quad (8)$$

对 (5) 中 A 求转置，得到：

$$A^T = \begin{bmatrix} A_{1,1}^T & A_{2,1}^T \\ A_{1,2}^T & A_{2,2}^T \end{bmatrix} \quad (9)$$

上式相当于由两层转置运算构成。第一层把子块当成元素，进行转置；第二层是子块矩阵转置运算。代入具体值，得到：

$$A^T = \begin{bmatrix} 1 & 4 & 0 & 0 \\ 2 & 5 & 0 & 0 \\ 3 & 6 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

请大家仔细对比 (4) 和 (10)，分析转置前后子块矩阵的变化。

标量乘法

(5) 中分块矩阵标量乘法规则如下：

$$kA = \begin{bmatrix} kA_{1,1} & kA_{1,2} \\ kA_{2,1} & kA_{2,2} \end{bmatrix} \quad (11)$$

加减法

给定矩阵 B ，它的形状和 (5) 中 A 相同，采用相同的分块法分割 B ，得到：

$$B = \begin{bmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{bmatrix} \quad (12)$$

矩阵 A 和 B 的相同位置的子块矩阵形状相同， A 和 B 相加为对应位置子块分别相加：

$$A + B = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} + \begin{bmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{bmatrix} = \begin{bmatrix} A_{1,1} + B_{1,1} & A_{1,2} + B_{1,2} \\ A_{2,1} + B_{2,1} & A_{2,2} + B_{2,2} \end{bmatrix} \quad (13)$$

上述规则也适用于减法。

矩阵乘法

分块矩阵乘法规则也基于矩阵乘法规则。 A 和 B 相乘时，首先保证 A 的列数等于 B 的行数。 A 和 B 分块时，保证 A 的每一个子块矩阵的列数分别等于对应位置 B 的每个子块的行数。这样 A 和 B 相乘可以展开写成：

$$AB = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} \begin{bmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{bmatrix} = \begin{bmatrix} A_{1,1}B_{1,1} + A_{1,2}B_{2,1} & A_{1,1}B_{1,2} + A_{1,2}B_{2,2} \\ A_{2,1}B_{1,1} + A_{2,2}B_{2,1} & A_{2,1}B_{1,2} + A_{2,2}B_{2,2} \end{bmatrix} \quad (14)$$

上式中分块矩阵的乘法有两层运算。第一层矩阵乘法将子块视作元素来完成矩阵乘法，第二层是子块矩阵之间矩阵乘法。本章后会深入讲解不同形态的分块矩阵乘法。

6.2 矩阵乘法第一视角：标量积展开

本书前文以两个 2×2 矩阵相乘为例讲解过观察矩阵乘法的两个视角。本节和下一节回顾这两个视角的同时，进一步从分块矩阵视角理解矩阵乘法规则。

本节讨论矩阵乘法的常规视角——**标量积展开** (scalar product expansion)。

首先回顾矩阵乘法规则。

当矩阵 A 的列数等于矩阵 B 的行数时， A 与 B 可以相乘。比如下例中，矩阵 A 的形状为 n 行 D 列，矩阵 B 的形状为 D 行 m 列。 A 与 B 相乘时，相当于 D 被消去。

⚠ 再次强调，一般情况，矩阵乘法不满足交换律，即 $AB \neq BA$ 。

A 与 B 相乘得到的矩阵 C 的行数等于矩阵 A 的行数， C 的列数等于 B 的列数，即 AB 结果的形状为 n 行 m 列：

$$C_{n \times m} = A_{n \times D} B_{D \times m} = A_{n \times D} @ B_{D \times m} = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,m} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n,1} & c_{n,2} & \cdots & c_{n,m} \end{bmatrix} \quad (15)$$

其中，

$$A_{n \times D} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,D} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,D} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,D} \end{bmatrix}, \quad B_{D \times m} = \begin{bmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,m} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ b_{D,1} & b_{D,2} & \cdots & b_{D,m} \end{bmatrix} \quad (16)$$

将矩阵 A 写成一组行向量：

$$A_{n \times D} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,D} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,D} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,D} \end{bmatrix}_{n \times D} = \begin{bmatrix} \mathbf{a}^{(1)} \\ \mathbf{a}^{(2)} \\ \vdots \\ \mathbf{a}^{(n)} \end{bmatrix}_{n \times 1} \quad (17)$$

将矩阵 B 写成一组列向量：

$$B_{D \times m} = \begin{bmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,m} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ b_{D,1} & b_{D,2} & \cdots & b_{D,m} \end{bmatrix}_{D \times m} = [\mathbf{b}_1 \quad \mathbf{b}_2 \quad \cdots \quad \mathbf{b}_m]_{1 \times m} \quad (18)$$

利用 (17) 和 (18)，矩阵乘积 AB 可以写作：

$$C = AB = \begin{bmatrix} \mathbf{a}^{(1)} \\ \mathbf{a}^{(2)} \\ \vdots \\ \mathbf{a}^{(n)} \end{bmatrix}_{n \times 1} [\mathbf{b}_1 \quad \mathbf{b}_2 \quad \cdots \quad \mathbf{b}_m]_{1 \times m} = \begin{bmatrix} \mathbf{a}^{(1)}\mathbf{b}_1 & \mathbf{a}^{(1)}\mathbf{b}_2 & \cdots & \mathbf{a}^{(1)}\mathbf{b}_m \\ \mathbf{a}^{(2)}\mathbf{b}_1 & \mathbf{a}^{(2)}\mathbf{b}_2 & \cdots & \mathbf{a}^{(2)}\mathbf{b}_m \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}^{(n)}\mathbf{b}_1 & \mathbf{a}^{(n)}\mathbf{b}_2 & \cdots & \mathbf{a}^{(n)}\mathbf{b}_m \end{bmatrix}_{n \times m} \quad (19)$$

上式便是矩阵乘法的常规视角，规则如图 4 所示。

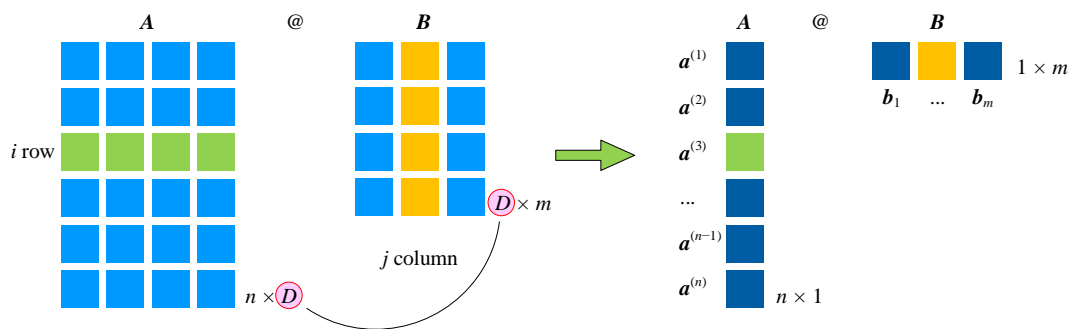
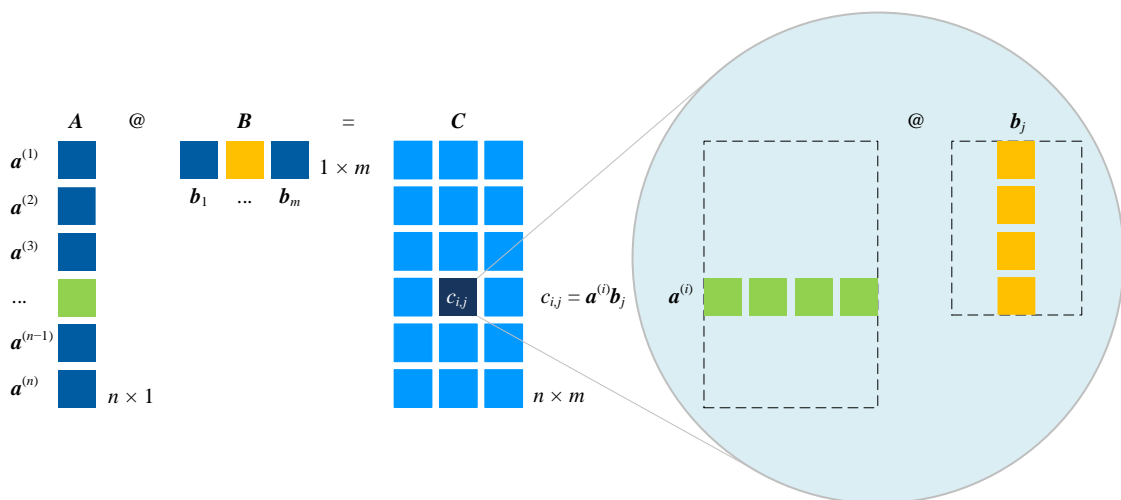


图 4. 矩阵乘法的常规视角

如图 5 所示，矩阵乘积 C 的 (i,j) 元素 $c_{i,j}$ 为矩阵 A 的第 i 行行向量 $\mathbf{a}^{(i)}$ 和矩阵 B 的第 j 列列向量 \mathbf{b}_j 的乘积：

$$c_{i,j} = \mathbf{a}^{(i)}\mathbf{b}_j \quad (20)$$

白话说，矩阵乘法的常规视角是，左侧矩阵的每个行向量，按规则分别乘右侧矩阵每个列向量。

图 5. 矩阵乘法的常规视角中，矩阵乘积 C 的 (i,j) 元素

6.3 矩阵乘法第二视角：外积展开

本节回顾矩阵乘法规则的第二视角——**外积展开** (outer product expansion)。

与上一节介绍的矩阵乘法常规视角不同，我们将矩阵 A 写成一组列向量：

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,D} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,D} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,D} \end{bmatrix} = [a_1 \quad a_2 \quad \cdots \quad a_D] \quad (21)$$

矩阵 B 则写成一组行向量：

$$B = \begin{bmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,m} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ b_{D,1} & b_{D,2} & \cdots & b_{D,m} \end{bmatrix} = \begin{bmatrix} b^{(1)} \\ b^{(2)} \\ \vdots \\ b^{(D)} \end{bmatrix} \quad (22)$$

这样，在计算矩阵乘积 AB 时，我们便得到如图 6 所示这个全新的视角。

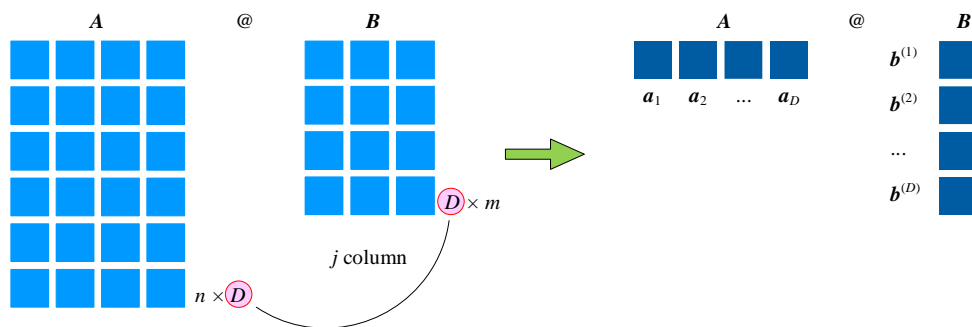
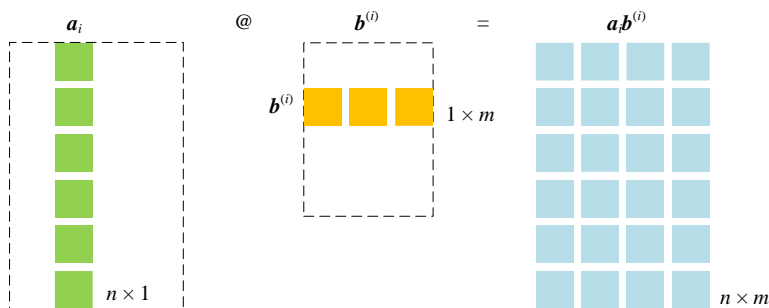


图 6. 矩阵乘法的第二视角

利用 (21) 和 (22)，矩阵乘积 AB 展开写成：

$$C = AB = [a_1 \ a_2 \ \cdots \ a_D]_{n \times D} \begin{bmatrix} b^{(1)} \\ b^{(2)} \\ \vdots \\ b^{(D)} \end{bmatrix}_{D \times 1} = a_1 b^{(1)} + a_2 b^{(2)} + \cdots + a_D b^{(D)} = \sum_{i=1}^D a_i b^{(i)} \quad (23)$$

利用第二视角，矩阵乘法运算转化成求和运算。如图 7 所示，列向量 a_i 和行向量 $b^{(i)}$ 乘积的结果的形状为 $n \times m$ ，即乘积 C 矩阵的形状。

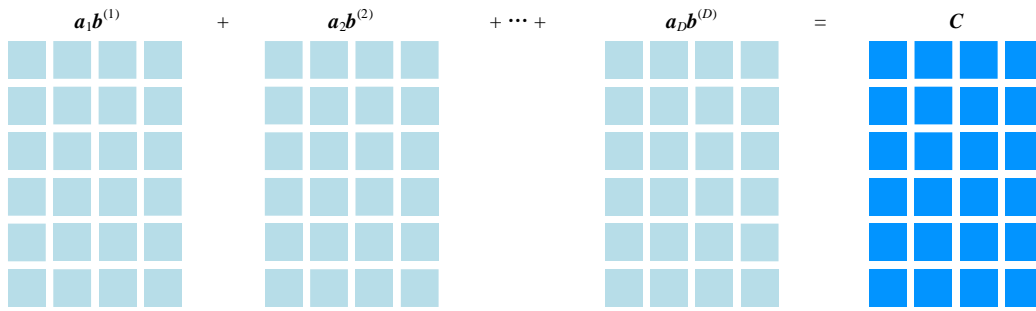
图 7. 列向量 a_i 和行向量 $b^{(i)}$ 乘积的结果

令，

$$C_i = a_i b^{(i)} \quad (24)$$

通过观察 (23)，可以发现乘积 C 矩阵相当于 D 个矩阵 C_i 叠加之和：

$$C = C_1 + C_2 + \cdots + C_D = \sum_{i=1}^D C_i \quad (25)$$

图 8. 乘积 C 矩阵相当于 D 个矩阵叠加之和

张量积

用张量积运算规则，把 (23) 中矩阵 C 写成一系列张量积之和：

$$C = a_1 \otimes (b^{(1)})^T + a_2 \otimes (b^{(2)})^T + \cdots + a_D \otimes (b^{(D)})^T$$

$$= \sum_{i=1}^D a_i \otimes (b^{(i)})^T \quad (26)$$

⚠ 请大家格外注意 (26) 中的转置运算。

矩阵乘法的第二视角不仅仅是常规视角的补充。在很多数据科学和机器学习算法中，矩阵乘法第二视角扮演至关重要的角色。

热图示例

下面我们用具体数字和热图可视化矩阵乘法外积展开。

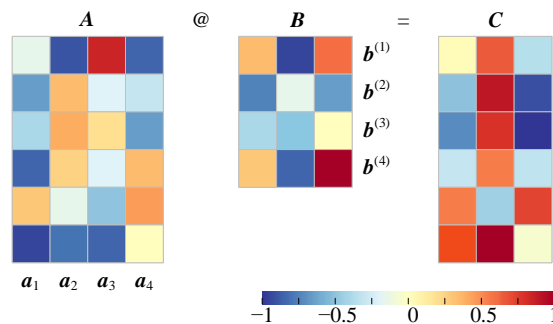


图 9. 矩阵乘法热图

图 9 所示为 A 和 B 矩阵乘法热图。将矩阵 A 拆解为一组列向量，矩阵 B 拆解为一组行向量。按照 (23)，得到如图 10 所示 4 幅热图。

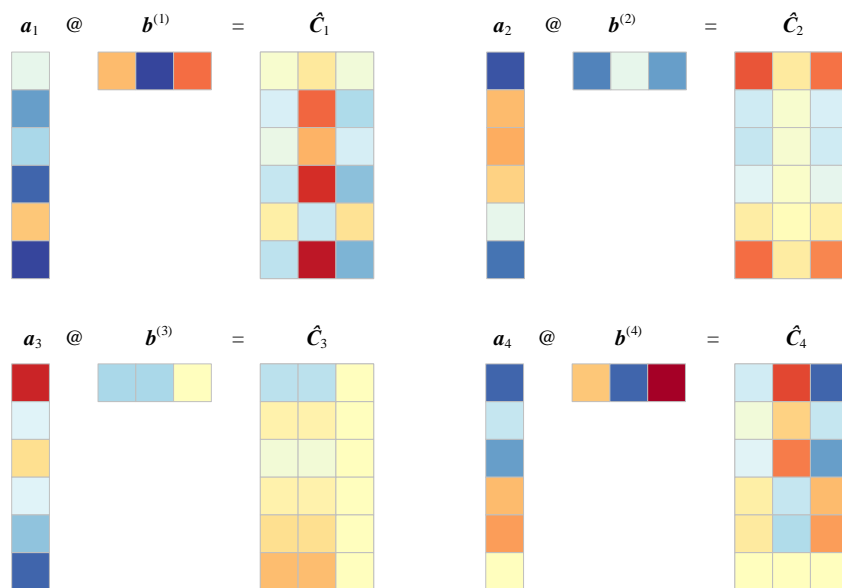


图 10. 四副列向量乘行向量结果热图

同样，也可以用张量积来计算得到这 4 幅热图，如图 11 所示。

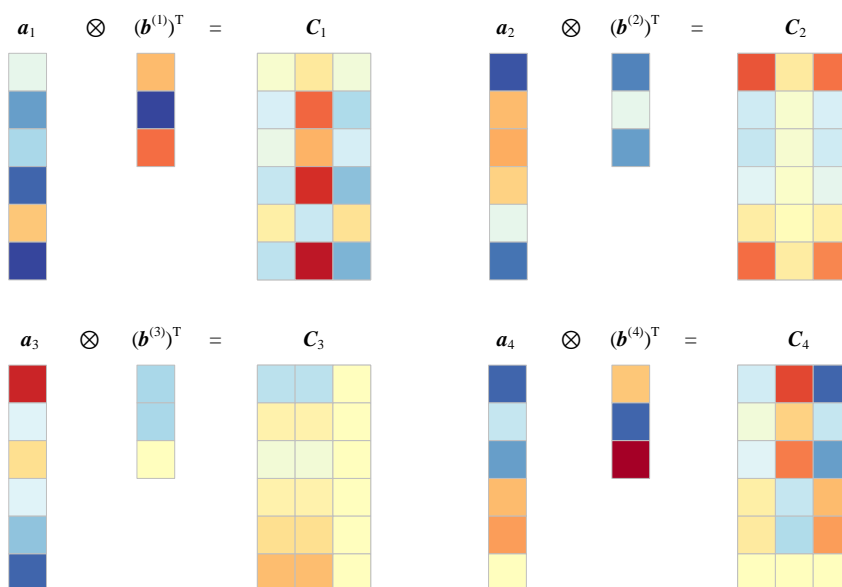


图 11. 四副张量积热图

如图 12 所示，将这 4 幅热图叠加，我们可以得到乘积结果矩阵 C 。

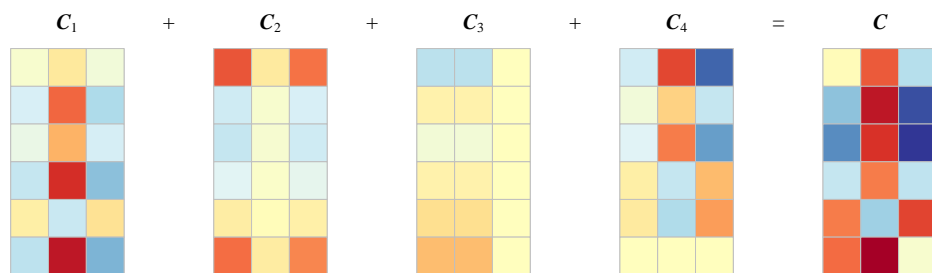


图 12. 四幅热图叠加

图 12 这个思路对于**特征值分解** (Eigen Decomposition)、**奇异值分解** (Singular Value Decomposition, SVD)、**主成分分析** (Principal Component Analysis, PCA) 非常重要。本书第 13、14 章将专门讲解特征值分解原理和应用，第 15、16 章专门介绍奇异值分解原理和应用。学好特征值分解、奇异值分解的关键就是“多视角”——数据视角、向量视角、几何视角、空间视角、统计视角等等。本书第 18 章专门介绍理解特征值分解、奇异值分解的优化视角。本书第 23 章则用奇异值分解介绍“四个空间”。



Bk4_Ch6_02.py 绘制图 12 的每幅热图。

6.4 矩阵乘法更多视角：分块多样化

本节介绍常见几种分块矩阵乘法形态，它们都可以视作观察矩阵乘法的不同视角。

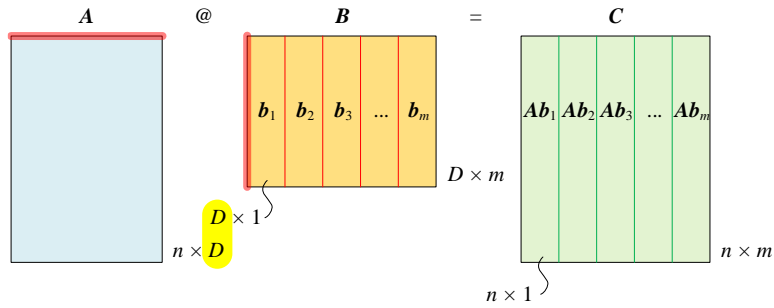
B 切成列向量

A 和 B 矩阵相乘时，将 B 分割成列向量，这样 AB 结果为：

$$C = AB = A[b_1 \ b_2 \ \cdots \ b_m] = [Ab_1 \ Ab_2 \ \cdots \ Ab_m] \quad (27)$$

图 13 所示为上述运算示意图。

▲ 请大家格外注意这个视角，本书之后的投影运算中经常见到这种展开方法。

图 13. A 和 B 矩阵相乘时，将 B 写成一组列向量

反方向来看，如果存在以下一系列矩阵乘法运算：

$$Ab_1 = c_1, \quad Ab_2 = c_2, \quad \dots \quad Ab_m = c_m \quad (28)$$

其中，列向量 b_1, b_2, \dots, b_m 的形状相同。(28) 中 m 个等式可以合成得到：

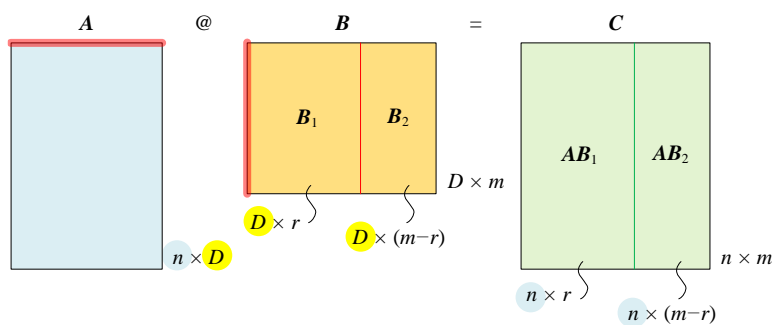
$$A \begin{bmatrix} b_1 & b_2 & \dots & b_m \end{bmatrix} = \begin{bmatrix} c_1 & c_2 & \dots & c_m \end{bmatrix} \quad (29)$$

B 左右切一刀

B 先左右切一刀后，矩阵 A 再左乘 B ，乘积 AB 展开写成：

$$AB = A \begin{bmatrix} B_1 & B_2 \end{bmatrix} = \begin{bmatrix} AB_1 & AB_2 \end{bmatrix} \quad (30)$$

图 14 所示为上述运算示意图。

图 14. 将 B 左右切一刀再右乘 A

A 切成一组行向量

A 和 B 矩阵相乘，将 A 分割成一组行向量，乘积 AB 结果为：

$$C = AB = \begin{bmatrix} a^{(1)} \\ a^{(2)} \\ \vdots \\ a^{(n)} \end{bmatrix}_{n \times 1} @ B = \begin{bmatrix} a^{(1)}B \\ a^{(2)}B \\ \vdots \\ a^{(n)}B \end{bmatrix}_{n \times 1} \quad (31)$$

图 15 所示为上述运算示意图。此外，请大家也试着从“合成”角度，逆向来看上述运算。

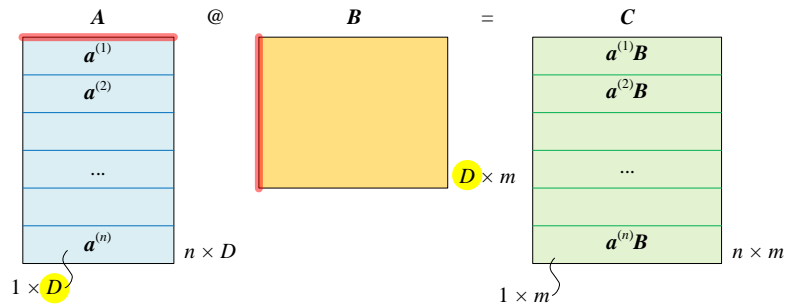


图 15. A 和 B 矩阵相乘，将 A 分割成一组行向量

A 上下切一刀

将 A 先上下切一刀， A 再左乘 B ，乘积 AB 结果为：

$$AB = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} B = \begin{bmatrix} A_1 B \\ A_2 B \end{bmatrix} \quad (32)$$

图 16 所示为上述运算示意图。

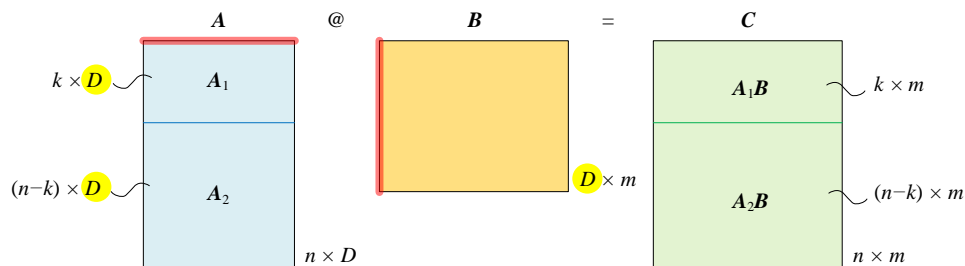


图 16. A 上下切一刀，再左乘 B

A 上下切， B 左右切

上下分块的 A 乘左右分块的 B ，乘积 AB 结果展开为：

$$AB = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \begin{bmatrix} B_1 & B_2 \end{bmatrix} = \begin{bmatrix} A_1 B_1 & A_1 B_2 \\ A_2 B_1 & A_2 B_2 \end{bmatrix} \quad (33)$$

如图 17 所示， A_1 和 A_2 的列数还是 D ， B_1 和 B_2 的行数也是 D 。这个视角类似矩阵乘法的第一视角。我们可以把 A_1 和 A_2 视作矩阵 A 的两个元素， B_1 和 B_2 看成矩阵 B 的两个元素。

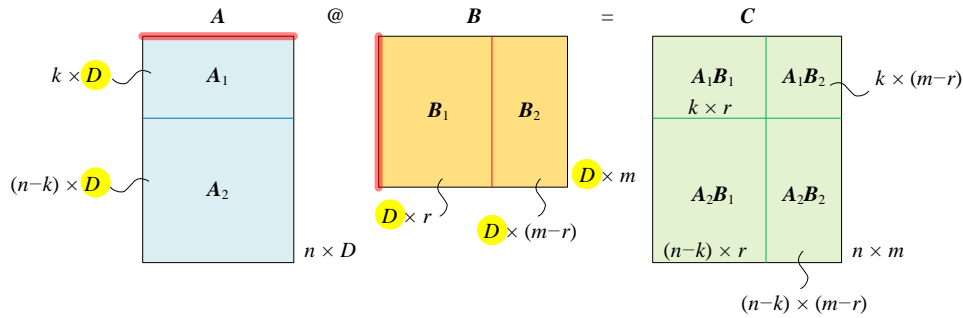


图 17. 上下分块的 A 乘左右分块的 B

A 左右切， B 上下切

左右分块的 A 乘上下分块的 B ，乘积 AB 结果展开为：

$$AB = \begin{bmatrix} A_1 & A_2 \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} = A_1 B_1 + A_2 B_2 \quad (34)$$

如图 18 所示， A_1 列数等于 B_1 行数， A_2 列数等于 B_2 行数。这类似前面讲到的矩阵乘法的第二视角。

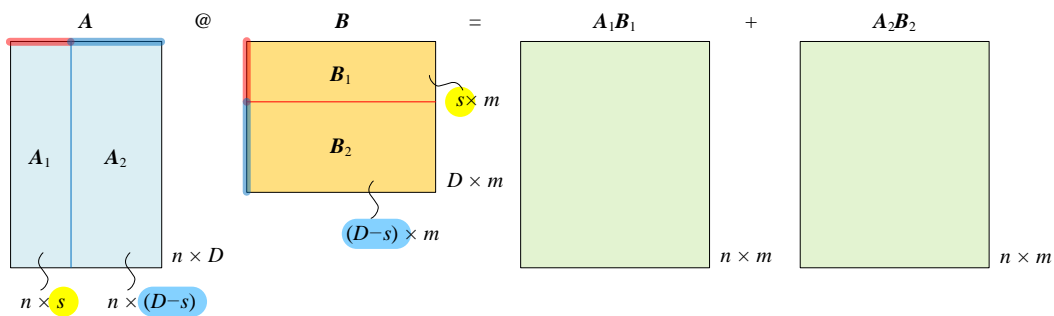


图 18. 左右分块的 A 乘上下分块的 B

A 和 B 都大卸四块

A 和 B 都上下左右分块，乘积 AB 结果为：

$$AB = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} \begin{bmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{bmatrix} = \begin{bmatrix} A_{1,1}B_{1,1} + A_{1,2}B_{2,1} & A_{1,1}B_{1,2} + A_{1,2}B_{2,2} \\ A_{2,1}B_{1,1} + A_{2,2}B_{2,1} & A_{2,1}B_{1,2} + A_{2,2}B_{2,2} \end{bmatrix} \quad (35)$$

如图 19 所示, $A_{1,1}$ 、 $A_{1,2}$ 、 $A_{2,1}$ 、 $A_{2,2}$ 的列数分别等于 $B_{1,1}$ 、 $B_{2,1}$ 、 $B_{1,2}$ 、 $B_{2,2}$ 的行数。图 19 中给出的分块矩阵乘法相当于两个 2×2 矩阵相乘, 结果 C 还是 2×2 。这也相当于矩阵乘法的第一视角。

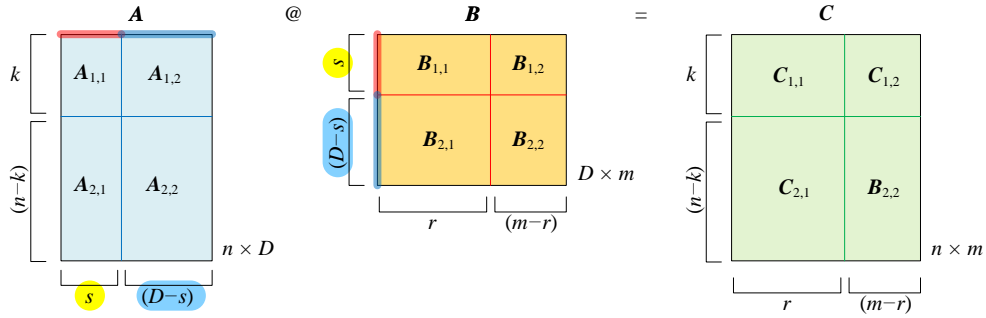


图 19. A 和 B 都上下左右分块

矩阵 C 的四个元素分别为 $C_{1,1}$ 、 $C_{1,2}$ 、 $C_{2,1}$ 、 $C_{2,2}$ 。图 20 到图 23 分别展示如何计算 $C_{1,1}$ 、 $C_{1,2}$ 、 $C_{2,1}$ 、 $C_{2,2}$ 。以 $C_{1,1}$ 为例, $C_{1,1}$ 的行数为 $A_{1,1}$ 的行数, $C_{1,1}$ 的列数为 $B_{1,1}$ 的列数。

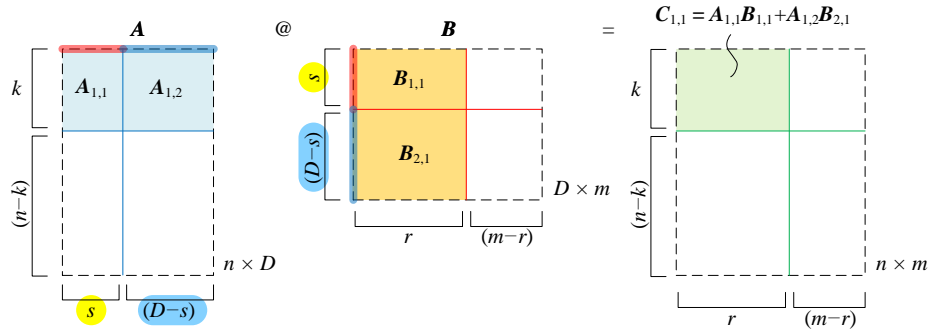


图 20. 计算 $C_{1,1}$

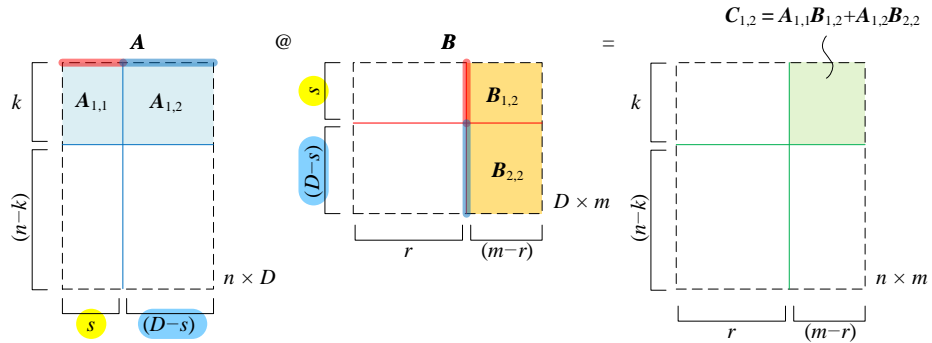
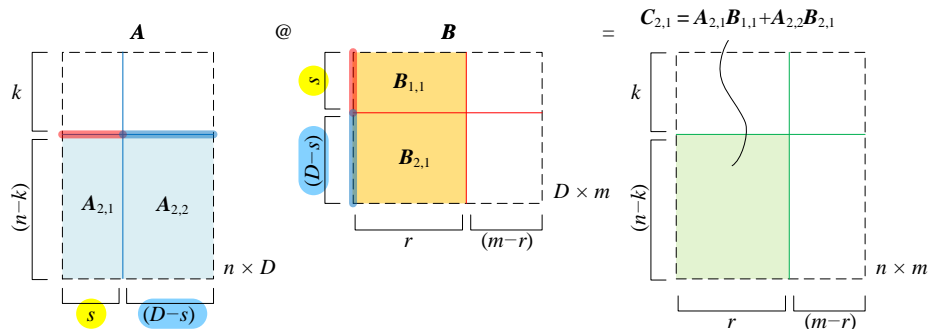
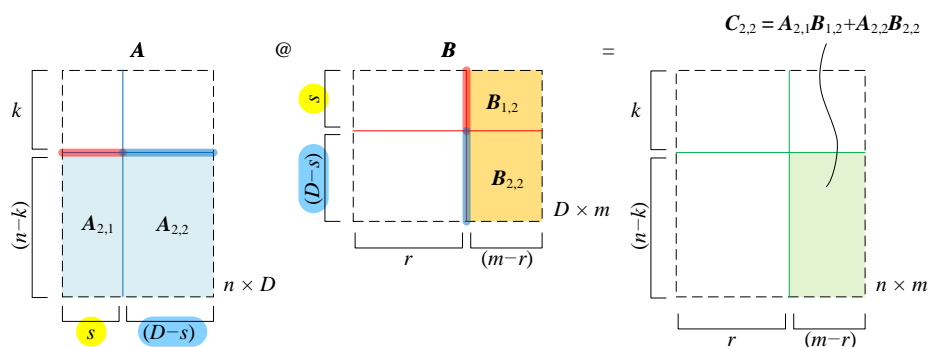
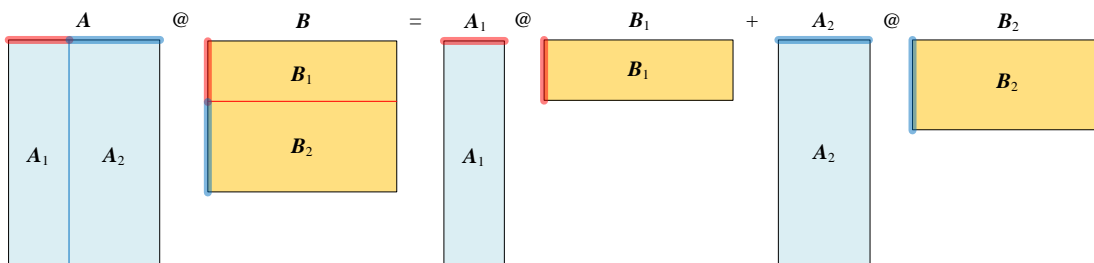


图 21. 计算 $C_{1,2}$ 图 22. 计算 $C_{2,1}$ 图 23. 计算 $C_{2,2}$

逐步分块

还有一个办法解释图 19 所示分块矩阵乘法——逐步分块。

首先将 A 左右分块， B 上下分块， AB 乘积的结果如 (34)，乘积 AB 结果写成 A_1B_1 和 A_2B_2 相加，具体如图 24 所示。

图 24. 首先将 A 左右分块， B 上下分块

然后再对 A_1 和 A_2 上下分块， B_1 和 B_2 左右分块：

$$\mathbf{A}_1 = \begin{bmatrix} \mathbf{A}_{1,1} \\ \mathbf{A}_{2,1} \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} \mathbf{A}_{1,2} \\ \mathbf{A}_{2,2} \end{bmatrix}, \quad \mathbf{B}_1 = \begin{bmatrix} \mathbf{B}_{1,1} & \mathbf{B}_{1,2} \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} \mathbf{B}_{2,1} & \mathbf{B}_{2,2} \end{bmatrix} \quad (36)$$

如图 25 所示, $\mathbf{A}_1\mathbf{B}_1$ 按如下方式计算得到:

$$\mathbf{A}_1\mathbf{B}_1 = \begin{bmatrix} \mathbf{A}_{1,1} \\ \mathbf{A}_{2,1} \end{bmatrix} \begin{bmatrix} \mathbf{B}_{1,1} & \mathbf{B}_{1,2} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{1,1}\mathbf{B}_{1,1} & \mathbf{A}_{1,1}\mathbf{B}_{1,2} \\ \mathbf{A}_{2,1}\mathbf{B}_{1,1} & \mathbf{A}_{2,1}\mathbf{B}_{1,2} \end{bmatrix} \quad (37)$$

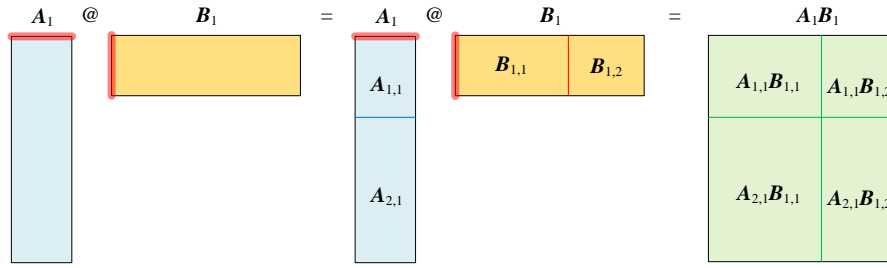


图 25. 计算 $\mathbf{A}_1\mathbf{B}_1$

同理, 如图 26 所示, 计算 $\mathbf{A}_2\mathbf{B}_2$:

$$\mathbf{A}_2\mathbf{B}_2 = \begin{bmatrix} \mathbf{A}_{1,2} \\ \mathbf{A}_{2,2} \end{bmatrix} \begin{bmatrix} \mathbf{B}_{2,1} & \mathbf{B}_{2,2} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{1,2}\mathbf{B}_{2,1} & \mathbf{A}_{1,2}\mathbf{B}_{2,2} \\ \mathbf{A}_{2,2}\mathbf{B}_{2,1} & \mathbf{A}_{2,2}\mathbf{B}_{2,2} \end{bmatrix} \quad (38)$$

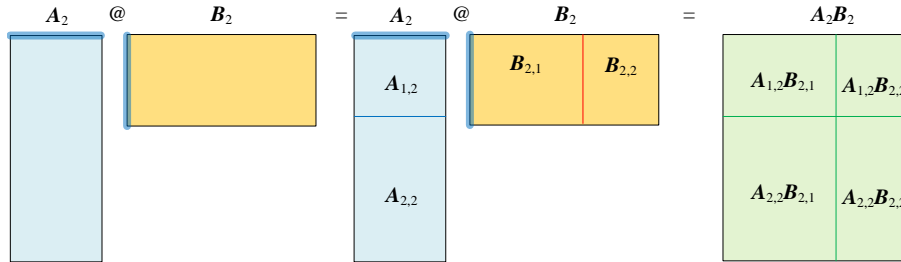


图 26. 计算 $\mathbf{A}_2\mathbf{B}_2$

(37) 和 (38) 相加就可以获得 (35) 结果, 即:

$$\begin{bmatrix} \mathbf{A}_{1,1}\mathbf{B}_{1,1} & \mathbf{A}_{1,1}\mathbf{B}_{1,2} \\ \mathbf{A}_{2,1}\mathbf{B}_{1,1} & \mathbf{A}_{2,1}\mathbf{B}_{1,2} \end{bmatrix} + \begin{bmatrix} \mathbf{A}_{1,2}\mathbf{B}_{2,1} & \mathbf{A}_{1,2}\mathbf{B}_{2,2} \\ \mathbf{A}_{2,2}\mathbf{B}_{2,1} & \mathbf{A}_{2,2}\mathbf{B}_{2,2} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{1,1}\mathbf{B}_{1,1} + \mathbf{A}_{1,2}\mathbf{B}_{2,1} & \mathbf{A}_{1,1}\mathbf{B}_{1,2} + \mathbf{A}_{1,2}\mathbf{B}_{2,2} \\ \mathbf{A}_{2,1}\mathbf{B}_{1,1} + \mathbf{A}_{2,2}\mathbf{B}_{2,1} & \mathbf{A}_{2,1}\mathbf{B}_{1,2} + \mathbf{A}_{2,2}\mathbf{B}_{2,2} \end{bmatrix} \quad (39)$$

实际上, 这个思路便是矩阵乘法第二视角。

本节内容足见矩阵乘法的灵活性, 以及矩阵乘法两个视角的重要性。本书第 11 章讲解 QR 分解、第 16 章讲解四种奇异值分解类型时都会用到分块矩阵乘法。

6.5 分块矩阵的逆

如图 27 所示，将一个方阵分割成四个子块矩阵 A 、 B 、 C 和 D ，其中 A 和 D 为方阵。当原矩阵可逆时，原矩阵的逆可以通过子块矩阵运算得到：

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} (A - BD^{-1}C)^{-1} & -(A - BD^{-1}C)^{-1}BD^{-1} \\ -D^{-1}C(A - BD^{-1}C)^{-1} & D^{-1} + D^{-1}C(A - BD^{-1}C)^{-1}BD^{-1} \end{bmatrix} \quad (40)$$

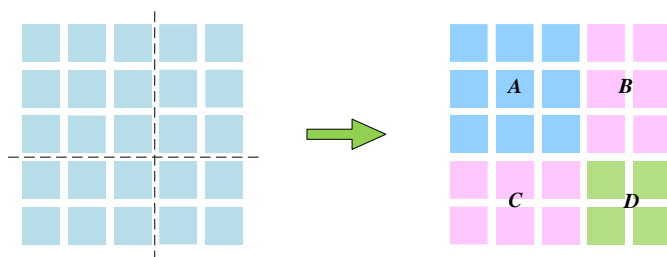


图 27. 分块矩阵求逆

令，

$$H = (A - BD^{-1}C)^{-1} \quad (41)$$

(40) 分块矩阵的逆可以写成：

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} H & -HBD^{-1} \\ -D^{-1}CH & D^{-1} + D^{-1}CHBD^{-1} \end{bmatrix} \quad (42)$$

当然，这个分块矩阵的逆还有其他表达方式，本节不一一赘述。

➡ 分块矩阵的逆将会用在协方差矩阵上，特别是在求解条件概率、多元线性回归时。本系列丛书《概率统计》一则会深入探讨这一话题。

6.6 克罗内克积：矩阵张量积

克罗内克积 (Kronecker product)，也叫矩阵张量积，是两个任意大小矩阵之间的运算，运算符为 \otimes 。

矩阵 A 的形状为 $n \times D$ ，矩阵 B 的形状为 $p \times q$ ，那么 $A \otimes B$ 的形状为 $np \times Dq$ ，结果为：

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,D} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,D} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,D} \end{bmatrix} \otimes \mathbf{B} = \begin{bmatrix} a_{1,1}\mathbf{B} & a_{1,2}\mathbf{B} & \cdots & a_{1,D}\mathbf{B} \\ a_{2,1}\mathbf{B} & a_{2,2}\mathbf{B} & \cdots & a_{2,D}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1}\mathbf{B} & a_{n,2}\mathbf{B} & \cdots & a_{n,D}\mathbf{B} \end{bmatrix} \quad (43)$$

上式中每个 $a_{i,j}$ 可以看成是缩放系数。

比如两个 2×2 矩阵 \mathbf{A} 和 \mathbf{B} 的张量积为 4×4 矩阵：

$$\begin{aligned} \mathbf{A} \otimes \mathbf{B} &= \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix} \otimes \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} = \begin{bmatrix} a_{1,1}\mathbf{B} & a_{1,2}\mathbf{B} \\ a_{2,1}\mathbf{B} & a_{2,2}\mathbf{B} \end{bmatrix} \\ &= \begin{bmatrix} a_{1,1} \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} & a_{1,2} \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} \\ a_{2,1} \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} & a_{2,2} \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} a_{1,1}b_{1,1} & a_{1,1}b_{1,2} & a_{1,2}b_{1,1} & a_{1,2}b_{1,2} \\ a_{1,1}b_{2,1} & a_{1,1}b_{2,2} & a_{1,2}b_{2,1} & a_{1,2}b_{2,2} \\ a_{2,1}b_{1,1} & a_{2,1}b_{1,2} & a_{2,2}b_{1,1} & a_{2,2}b_{1,2} \\ a_{2,1}b_{2,1} & a_{2,1}b_{2,2} & a_{2,2}b_{2,1} & a_{2,2}b_{2,2} \end{bmatrix} \end{aligned} \quad (44)$$

`numpy.kron()` 可以用来计算矩阵张量积。

克罗内克积讲究顺序，一般情况 $\mathbf{A} \otimes \mathbf{B} \neq \mathbf{B} \otimes \mathbf{A}$ 。

请大家注意以下有关克罗内克积性质：

$$\begin{aligned} \mathbf{A} \otimes (\mathbf{B} + \mathbf{C}) &= \mathbf{A} \otimes \mathbf{B} + \mathbf{A} \otimes \mathbf{C} \\ (\mathbf{B} + \mathbf{C}) \otimes \mathbf{A} &= \mathbf{B} \otimes \mathbf{A} + \mathbf{C} \otimes \mathbf{A} \\ (k\mathbf{A}) \otimes \mathbf{B} &= \mathbf{A} \otimes (k\mathbf{B}) = k(\mathbf{A} \otimes \mathbf{B}) \\ (\mathbf{A} \otimes \mathbf{B}) \otimes \mathbf{C} &= \mathbf{A} \otimes (\mathbf{B} \otimes \mathbf{C}) \\ \mathbf{A} \otimes \mathbf{0} &= \mathbf{0} \otimes \mathbf{A} = \mathbf{0} \end{aligned} \quad (45)$$

和向量张量积的关系

克罗内克积相当于向量张量积的推广；反过来，向量张量积也可以看做克罗内克积的特例。

但两者稍有不同，为了方便计算，两个 2×1 列向量的张量积定义为 $\mathbf{a} \otimes \mathbf{b} = \mathbf{ab}^T$ ，也就是：

$$\mathbf{a} \otimes \mathbf{b} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \otimes \mathbf{b} = \begin{bmatrix} a_1\mathbf{b}^T \\ a_2\mathbf{b}^T \end{bmatrix} \quad (46)$$

请大家注意 (46) 中的转置运算。而 (43) 中不存在转置。

举个例子

\mathbf{A} 和 \mathbf{B} 分别为：

$$\mathbf{A} = \begin{bmatrix} -1 & 1 \\ 0.7 & -0.4 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0.5 & -0.6 \\ -0.8 & 0.3 \end{bmatrix} \quad (47)$$

A 和 B 的张量积 $A \otimes B$ 为：

$$A \otimes B = \begin{bmatrix} -1 & 1 \\ 0.7 & -0.4 \end{bmatrix} \otimes \begin{bmatrix} 0.5 & -0.6 \\ -0.8 & 0.3 \end{bmatrix} = \begin{bmatrix} -1 \times \begin{bmatrix} 0.5 & -0.6 \\ -0.8 & 0.3 \end{bmatrix} & 1 \times \begin{bmatrix} 0.5 & -0.6 \\ -0.8 & 0.3 \end{bmatrix} \\ 0.7 \times \begin{bmatrix} 0.5 & -0.6 \\ -0.8 & 0.3 \end{bmatrix} & -0.4 \times \begin{bmatrix} 0.5 & -0.6 \\ -0.8 & 0.3 \end{bmatrix} \end{bmatrix} \quad (48)$$

图 28 所示为上述计算的热图。

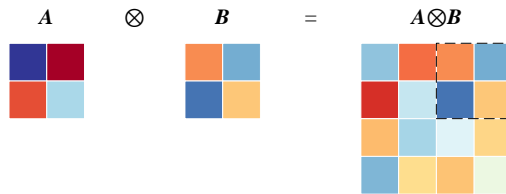


图 28. A 和 B 的张量积 $A \otimes B$

再给出第三个 2×2 矩阵 C ：

$$C = \begin{bmatrix} -1 & 1 \\ 0.7 & -0.4 \end{bmatrix} \quad (49)$$

在 $A \otimes B \otimes C$ 的张量积的运算如图 29 所示。也请大家先计算 $B \otimes C$ ，再计算 $A \otimes B \otimes C$ 。

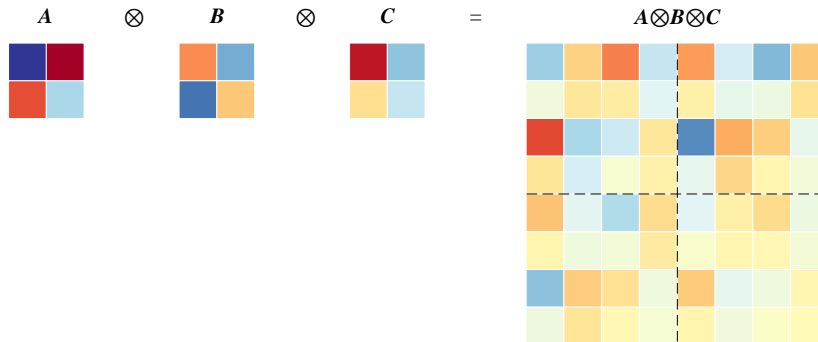
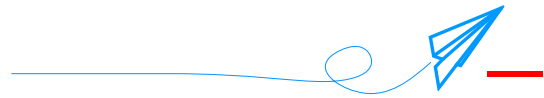


图 29. A 、 B 、 C 的张量积 $A \otimes B \otimes C$



Bk4_Ch6_03.py 计算张量积并绘制图 28。请大家自行绘制图 29。



虽然分块矩阵乘法运算让人看的眼花缭乱；但是，万变不离其宗，大家关键要把握的是矩阵乘法规则，这是根本。其次，同等重要的就是，我们在本书中反复强调的——矩阵乘法两个视角。

此外，大家注意矩阵乘法的“合成”，也就是分块矩阵乘法的逆向运算。掌握这个逆向思维方式有助于理解和简化很多运算，大家将会在本书后文数据投影中看到大量实例。