8 几何变换

Geometric Transformation

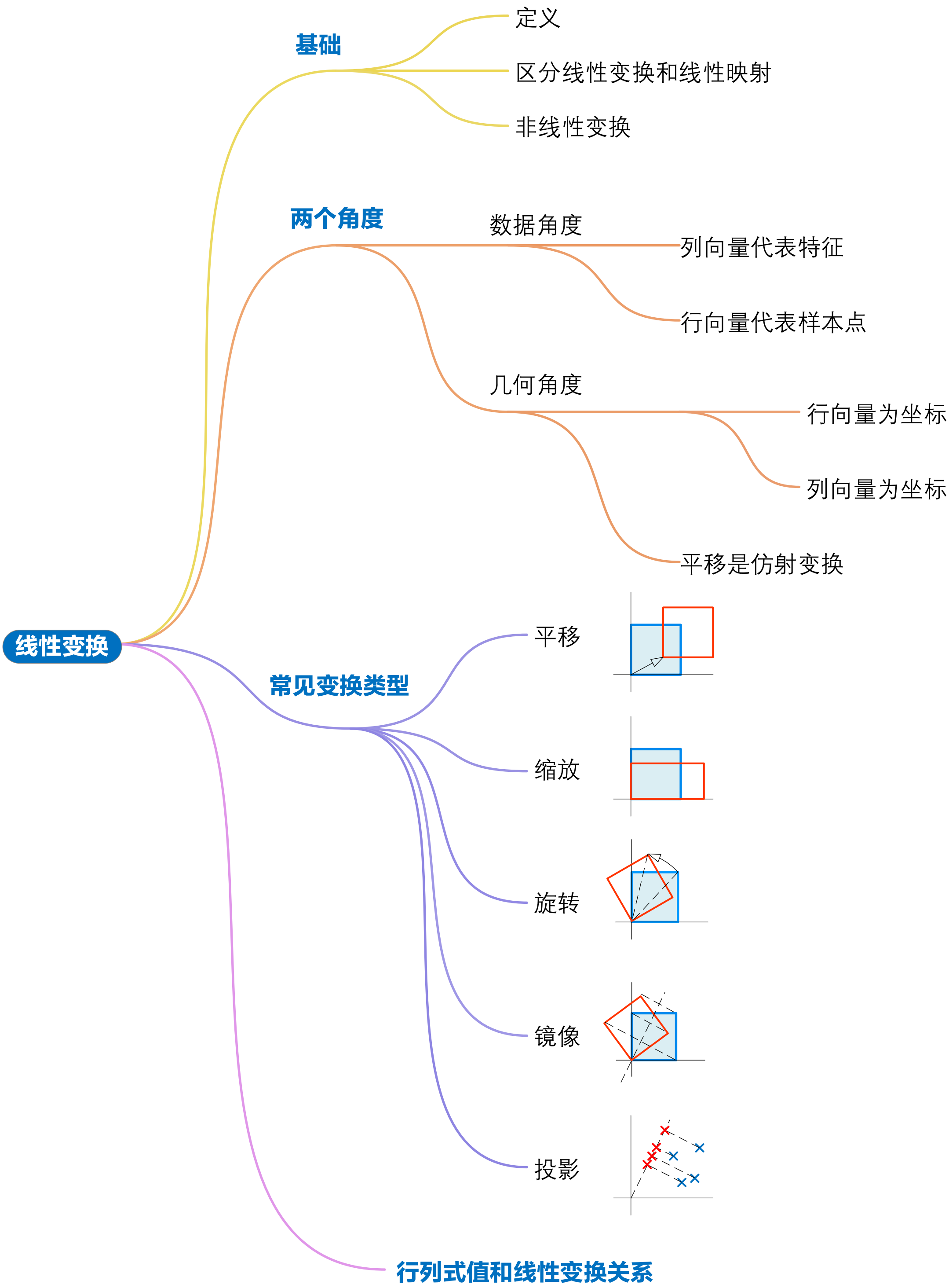
线性变换的特征是平行且等距的网格

矩阵向来大有所为，矩阵从不游手好闲。

Matrices act. They don't just sit there.

—— 吉尔伯特·斯特朗 (Gilbert Strang) | MIT 数学教授 | 1934 ~

- `numpy.array()` 构造多维矩阵/数组
- `numpy.linalg.inv()` 矩阵逆运算
- `numpy.matrix()` 构造二维矩阵
- `numpy.multiply()` 矩阵逐项积
- `tranpose()` 矩阵转置，比如 `A.transpose()`，等同于 `A.T`


线性变换
基础
定义
区分线性变换和线性映射
非线性变换
两个角度
数据角度
列向量代表特征
行向量代表样本点
几何角度
行向量为坐标
列向量为坐标
平移是仿射变换
常见变换类型
平移
缩放
旋转
镜像
投影
行列式值和线性变换关系

8.1 线性变换：线性空间到自身的线性映射

本章开始之前，我们先区分两个概念：**线性映射** (linear mapping) 和**线性变换** (linear transformation)。

线性映射是指从一个空间到另外一个空间的映射，且保持加法和数量乘法运算。比如，映射 L 将向量空间 V 映射到向量空间 W，对于所有的 $\boldsymbol{v}_1$、$\boldsymbol{v}_2 \in V$ 及所有的标量 α 和 β，满足

$$L(\alpha\boldsymbol{v}_1+\beta\boldsymbol{v}_2)=\alpha L(\boldsymbol{v}_1)+\beta L(\boldsymbol{v}_2) \tag{1}$$

白话来说，线性映射把一个空间的点或几何形体映射到另外一个空间。比如图 1 所示的三维物体投影到一个平面上，得到这个杯子在平面上的映像。

图 1 所示的"降维"过程显然不可逆，也就是不能通过杯子在平面的"映像"获得杯子在三维空间所有的形体信息。

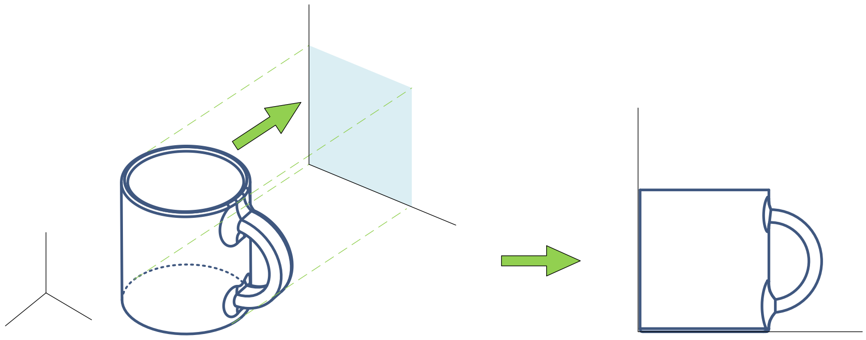


图 1. 线性映射

线性变换是线性空间到自身的线性映射，是一种特殊的线性映射。白话说，线性变换是在同一个坐标系中完成的图形变换。从几何角度来看，线性变换产生"平行且等距"的网格，并且原点保持固定，如图 2 所示。原点保持固定，这一性质很重要，因为大家马上就会看到"平移"不属于线性变换。

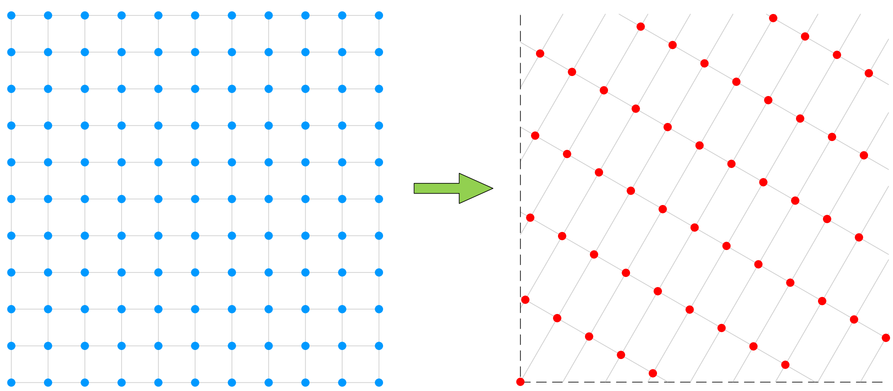


图 2. 线性变换产生平行且等距的网格

但是，请大家注意很多参考资料混用线性映射和线性变换。

非线性变换

与线性变换相对的就是**非线性变换** (nonlinear transformation)。

图 3 和图 4 给出两个非线性变换的例子。图 3 所示为通过非线性变换产生平行不等距网格。图 4 所示产生的网格甚至出现“扭曲”。

有了这两幅图做对比，相信读者能够更好地理解图 2 所展示的“平行且等距”的网格所代表的的线性变换。

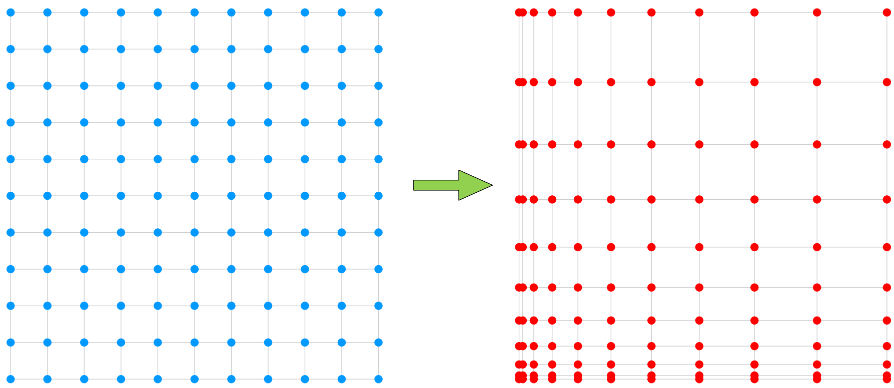


图 3. 非线性变换产生平行但不等距网格

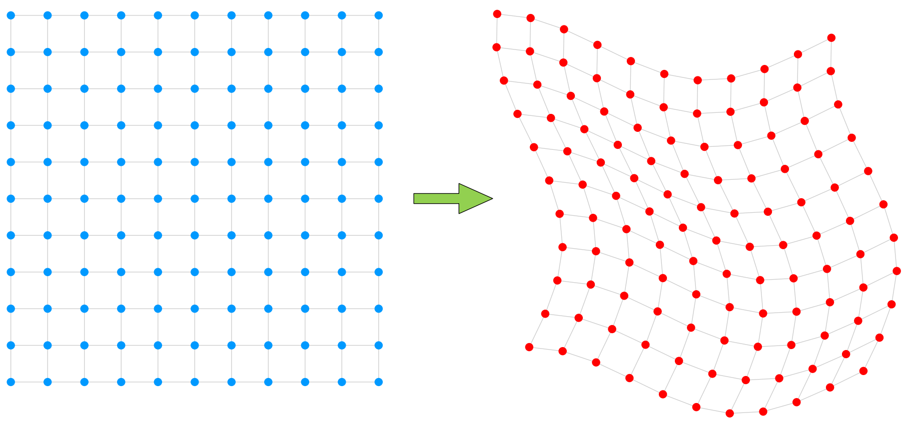


图 4. 非线性变换产生“扭曲”网格

常见平面几何变换

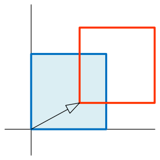
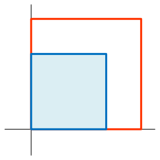
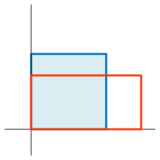
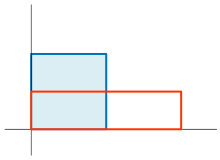
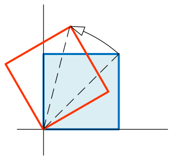
本章下一节开始就是要从几何角度去探讨线性变换。表 1 总结本章将要介绍的常用二维几何变换。表 1 的第二列和第三列矩阵运算互为转置关系。

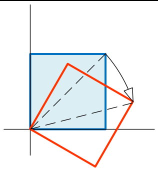
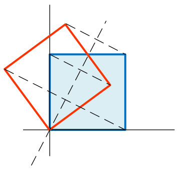
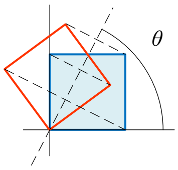
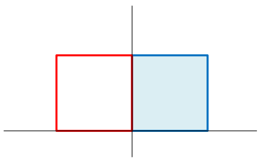
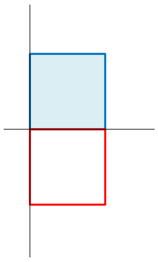
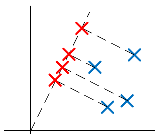
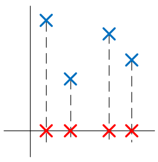
请读者注意，平移并不是线性变换，平移是**仿射变换** (affine transformation)。几何角度，仿射变换是一个向量空间进行线性变换并叠加平移，变换为另一个向量空间。平移时，原点位置发生变化。

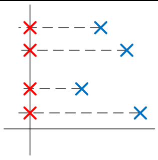
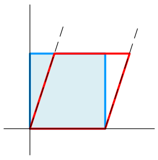
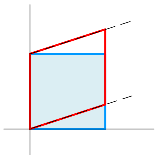
表 1 中所有操作统称几何变换，这便于将这些线性代数概念和本系列丛书《数学要素》中介绍的几何变换联系起来。这也正是本章题目叫“几何变换”的原因。

除了平移以外，表 1 中的几何变换都是从 $\mathbb{R}^2$ 到自身。值得注意的是，正交投影相当于降维，结果在 $\mathbb{R}^2$ 的子空间中。本章后续就一一展开讲解这些几何变换。

表 1. 常用几何变换总结

几何变换	列向量坐标	行向量坐标
平移 (translation) 	$\begin{bmatrix}z_1\\z_2\end{bmatrix}=\begin{bmatrix}x_1\\x_2\end{bmatrix}+\begin{bmatrix}t_1\\t_2\end{bmatrix}$	$\begin{bmatrix}z_1 & z_2\end{bmatrix}=\begin{bmatrix}x_1 & x_2\end{bmatrix}+\begin{bmatrix}t_1 & t_2\end{bmatrix}$ $\boldsymbol{Z}_{n\times 2}=\boldsymbol{X}_{n\times 2}+\begin{bmatrix}t_1 & t_2\end{bmatrix}$
等比例缩放 s 倍 (scaling) 	$\begin{bmatrix}z_1\\z_2\end{bmatrix}=s\begin{bmatrix}x_1\\x_2\end{bmatrix}=\begin{bmatrix}s & 0\\0 & s\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}$	$\begin{bmatrix}z_1 & z_2\end{bmatrix}=s\begin{bmatrix}x_1 & x_2\end{bmatrix}=\begin{bmatrix}x_1 & x_2\end{bmatrix}\begin{bmatrix}s & 0\\0 & s\end{bmatrix}$ $\boldsymbol{Z}_{n\times 2}=\boldsymbol{X}_{n\times 2}\begin{bmatrix}s & 0\\0 & s\end{bmatrix}$
非等比例缩放 (unequal scaling) 	$\begin{bmatrix}z_1\\z_2\end{bmatrix}=\begin{bmatrix}s_1 & 0\\0 & s_2\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}$	$\begin{bmatrix}z_1 & z_2\end{bmatrix}=\begin{bmatrix}x_1 & x_2\end{bmatrix}\begin{bmatrix}s_1 & 0\\0 & s_2\end{bmatrix}$ $\boldsymbol{Z}_{n\times 2}=\boldsymbol{X}_{n\times 2}\begin{bmatrix}s_1 & 0\\0 & s_2\end{bmatrix}$
挤压 s 倍 (squeeze) 	$\begin{bmatrix}z_1\\z_2\end{bmatrix}=\begin{bmatrix}s & 0\\0 & 1/s\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}$	$\begin{bmatrix}z_1 & z_2\end{bmatrix}=\begin{bmatrix}x_1 & x_2\end{bmatrix}\begin{bmatrix}s & 0\\0 & 1/s\end{bmatrix}$ $\boldsymbol{Z}_{n\times 2}=\boldsymbol{X}_{n\times 2}\begin{bmatrix}s & 0\\0 & 1/s\end{bmatrix}$
逆时针旋转 θ (counterclockwise rotation) 	$\begin{bmatrix}z_1\\z_2\end{bmatrix}=\begin{bmatrix}\cos(\theta) & -\sin(\theta)\\\sin(\theta) & \cos(\theta)\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}$	$\begin{bmatrix}z_1 & z_2\end{bmatrix}=\begin{bmatrix}x_1 & x_2\end{bmatrix}\begin{bmatrix}\cos(\theta) & \sin(\theta)\\-\sin(\theta) & \cos(\theta)\end{bmatrix}$ $\boldsymbol{Z}_{n\times 2}=\boldsymbol{X}_{n\times 2}\begin{bmatrix}\cos(\theta) & \sin(\theta)\\-\sin(\theta) & \cos(\theta)\end{bmatrix}$
顺时针旋转 θ (clockwise rotation)	$\begin{bmatrix}z_1\\z_2\end{bmatrix}=\begin{bmatrix}\cos(\theta) & \sin(\theta)\\-\sin(\theta) & \cos(\theta)\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}$	$\begin{bmatrix}z_1 & z_2\end{bmatrix}=\begin{bmatrix}x_1 & x_2\end{bmatrix}\begin{bmatrix}\cos(\theta) & -\sin(\theta)\\\sin(\theta) & \cos(\theta)\end{bmatrix}$

		$Z_{n\times 2} = X_{n\times 2}\begin{bmatrix}\cos(\theta) & -\sin(\theta)\\ \sin(\theta) & \cos(\theta)\end{bmatrix}$
关于通过原点、切向量为 $\boldsymbol{\tau}$ $[\tau_1, \tau_2]^{\mathrm{T}}$ 直线镜像 (reflection) 	$\begin{bmatrix}z_1\\ z_2\end{bmatrix} = \frac{1}{\lVert\boldsymbol{\tau}\rVert^2}\begin{bmatrix}\tau_1^2-\tau_2^2 & 2\tau_1\tau_2\\ 2\tau_1\tau_2 & \tau_2^2-\tau_1^2\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}$	$\begin{bmatrix}z_1 & z_2\end{bmatrix} = \begin{bmatrix}x_1 & x_2\end{bmatrix}\frac{1}{\lVert\boldsymbol{\tau}\rVert^2}\begin{bmatrix}\tau_1^2-\tau_2^2 & 2\tau_1\tau_2\\ 2\tau_1\tau_2 & \tau_2^2-\tau_1^2\end{bmatrix}$ $Z_{n\times 2} = X_{n\times 2}\frac{1}{\lVert\boldsymbol{\tau}\rVert^2}\begin{bmatrix}\tau_1^2-\tau_2^2 & 2\tau_1\tau_2\\ 2\tau_1\tau_2 & \tau_2^2-\tau_1^2\end{bmatrix}$
关于通过原点、方向和水平轴夹角为 θ 直线镜像；等同于上例，切向量相当于 $(\cos\theta, \sin\theta)$ 	$\begin{bmatrix}z_1\\ z_2\end{bmatrix} = \begin{bmatrix}\cos 2\theta & \sin 2\theta\\ \sin 2\theta & -\cos 2\theta\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}$	$\begin{bmatrix}z_1 & z_2\end{bmatrix} = \begin{bmatrix}x_1 & x_2\end{bmatrix}\begin{bmatrix}\cos 2\theta & \sin 2\theta\\ \sin 2\theta & -\cos 2\theta\end{bmatrix}$ $Z_{n\times 2} = X_{n\times 2}\begin{bmatrix}\cos 2\theta & \sin 2\theta\\ \sin 2\theta & -\cos 2\theta\end{bmatrix}$
关于横轴镜像对称 (reflection) 	$\begin{bmatrix}z_1\\ z_2\end{bmatrix} = \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}$	$\begin{bmatrix}z_1 & z_2\end{bmatrix} = \begin{bmatrix}x_1 & x_2\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$ $Z_{n\times 2} = X_{n\times 2}\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$
关于纵轴镜像对称 (reflection) 	$\begin{bmatrix}z_1\\ z_2\end{bmatrix} = \begin{bmatrix}-1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}$	$\begin{bmatrix}z_1 & z_2\end{bmatrix} = \begin{bmatrix}x_1 & x_2\end{bmatrix}\begin{bmatrix}-1 & 0\\ 0 & 1\end{bmatrix}$ $Z_{n\times 2} = X_{n\times 2}\begin{bmatrix}-1 & 0\\ 0 & 1\end{bmatrix}$
向通过原点、切向量为 $\boldsymbol{\tau}$ $[\tau_1, \tau_2]^{\mathrm{T}}$ 直线投影 (projection) 	$\begin{bmatrix}z_1\\ z_2\end{bmatrix} = \frac{1}{\lVert\boldsymbol{\tau}\rVert^2}\begin{bmatrix}\tau_1^2 & \tau_1\tau_2\\ \tau_1\tau_2 & \tau_2^2\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}$	$\begin{bmatrix}z_1 & z_2\end{bmatrix} = \begin{bmatrix}x_1 & x_2\end{bmatrix}\frac{1}{\lVert\boldsymbol{\tau}\rVert^2}\begin{bmatrix}\tau_1^2 & \tau_1\tau_2\\ \tau_1\tau_2 & \tau_2^2\end{bmatrix}$ $Z_{n\times 2} = X_{n\times 2}\frac{1}{\lVert\boldsymbol{\tau}\rVert^2}\begin{bmatrix}\tau_1^2 & \tau_1\tau_2\\ \tau_1\tau_2 & \tau_2^2\end{bmatrix}$
向横轴投影 (projection) 	$\begin{bmatrix}z_1\\ z_2\end{bmatrix} = \begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}$	$\begin{bmatrix}z_1 & z_2\end{bmatrix} = \begin{bmatrix}x_1 & x_2\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix}$ $Z_{n\times 2} = X_{n\times 2}\begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix}$
向纵轴投影 (projection)	$\begin{bmatrix}z_1\\ z_2\end{bmatrix} = \begin{bmatrix}0 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}$	$\begin{bmatrix}z_1 & z_2\end{bmatrix} = \begin{bmatrix}x_1 & x_2\end{bmatrix}\begin{bmatrix}0 & 0\\ 0 & 1\end{bmatrix}$ $Z_{n\times 2} = X_{n\times 2}\begin{bmatrix}0 & 0\\ 0 & 1\end{bmatrix}$

		
沿水平方向剪切，θ 为剪切角 (shear) 	$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 1 & \cot(\theta) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$	$\begin{bmatrix} z_1 & z_2 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \cot(\theta) & 1 \end{bmatrix}$ $\boldsymbol{Z}_{n\times 2} = \boldsymbol{X}_{n\times 2} \begin{bmatrix} 1 & 0 \\ \cot(\theta) & 1 \end{bmatrix}$
沿竖直方向剪切，θ 为剪切角 (shear) 	$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \cot(\theta) & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$	$\begin{bmatrix} z_1 & z_2 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 1 & \cot(\theta) \\ 0 & 1 \end{bmatrix}$ $\boldsymbol{Z}_{n\times 2} = \boldsymbol{X}_{n\times 2} \begin{bmatrix} 1 & \cot(\theta) \\ 0 & 1 \end{bmatrix}$

8.2 平移：仿射变换

再次强调，平移并不是线性变换，平移是仿射变换，因为原点发生改变。

列向量

用列向量表达坐标时，平移可以写成：

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \boldsymbol{t} \tag{2}$$

其中，$\boldsymbol{t}$ 为平移向量：

$$\boldsymbol{t} = \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} \tag{3}$$

(3) 代入 (2) 得到：

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} x_1 + t_1 \\ x_2 + t_2 \end{bmatrix} \tag{4}$$

行向量

用行向量表达坐标时，相当于对 (4) 左右转置：

$$\begin{bmatrix} z_1 & z_2 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 \end{bmatrix} + \begin{bmatrix} t_1 & t_2 \end{bmatrix} = \begin{bmatrix} x_1 + t_1 & x_2 + t_2 \end{bmatrix} \tag{5}$$

图 5 所示为平移的两个例子。

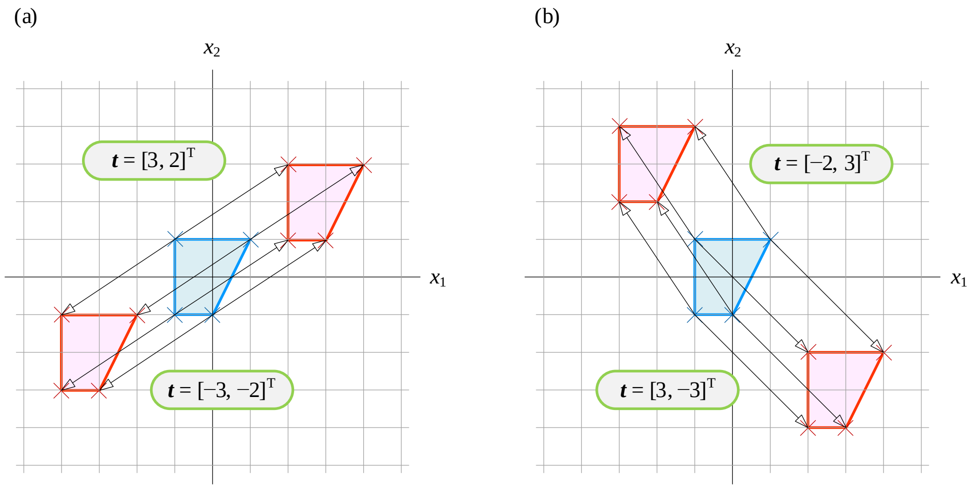


图 5. 平移

如图 6 所示，数据**中心化** (centralize)，也叫**去均值** (demean)，实际上就是一种平移。前文提到数据矩阵中一般用行向量表达坐标点。

对数据矩阵 $\boldsymbol{X}$ 去均值处理，并得到 $\boldsymbol{Y}$：

$$\boldsymbol{Y}_{n\times 2} = \boldsymbol{X}_{n\times 2} - \mathrm{E}\left(\boldsymbol{X}_{n\times 2}\right) \tag{6}$$

注意，前文提到过行向量 $\mathrm{E}(\boldsymbol{X})$ 叫做质心，它的每个元素是数据矩阵 $\boldsymbol{X}$ 每一列数据的均值。显然，$\boldsymbol{Y}$ 的质心位于原点，也就是说 $\mathrm{E}(\boldsymbol{Y}) = [0, 0]$。

利用广播原则，展开(6) 得到：

$$\begin{bmatrix} \boldsymbol{z}_1 & \boldsymbol{z}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 \end{bmatrix} - \begin{bmatrix} \mathrm{E}\left(\boldsymbol{x}_1\right) & \mathrm{E}\left(\boldsymbol{x}_2\right) \end{bmatrix} \tag{7}$$

(7) 对应的统计运算表达为：

$$\begin{cases} Y_1 = X_1 - \mathrm{E}\left(X_1\right) \\ Y_2 = X_2 - \mathrm{E}\left(X_2\right) \end{cases} \tag{8}$$

其中，X_1、X_2、Y_1、Y_2 为随机变量，注意字母大写斜体。从几何角度，平移运算将数据质心移动到原点，如图 6 所示。

大家应该已经注意到了图 6 中的椭圆，通过高斯二元分布可以建立数据和椭圆的联系。也就是说，从几何视角，椭圆可以用来代表散点数据。这是本系列丛书《概率统计》要重点讲解的内容。

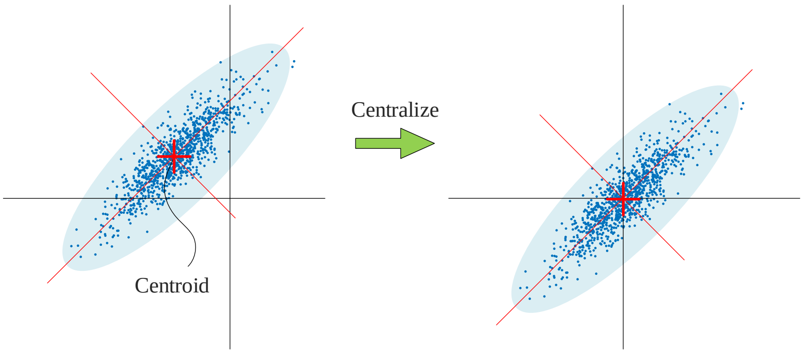


图 6. 数据中心化相当于平移

Bk4_Ch8_01.py 绘制图 5。

```
# Bk4_Ch8_01.py

import matplotlib.pyplot as plt
import numpy as np

def plot_shape(X,copy = False):
    if copy:
        fill_color = np.array([255,236,255])/255
        edge_color = np.array([255,0,0])/255
    else:
        fill_color = np.array([219,238,243])/255
        edge_color = np.array([0,153,255])/255

    plt.fill(X[:,0], X[:,1],
             color = fill_color,
             edgecolor = edge_color)

    plt.plot(X[:,0], X[:,1],marker = 'x',
             markeredgecolor = edge_color*0.5,
             linestyle = 'None')

X = np.array([[1,1],
              [0,-1],
              [-1,-1],
              [-1,1]])

# visualizations

fig, ax = plt.subplots()

plot_shape(X)       # plot original

# translation
t1 = np.array([3,2]);
Z = X + t1
plot_shape(Z,True) # plot copy

t2 = np.array([-3,-2]);
Z = X + t2
plot_shape(Z,True) # plot copy

t3 = np.array([-2,3]);
Z = X + t3
plot_shape(Z,True) # plot copy
```

```
t4 = np.array([3,-3]);
Z = X + t4
plot_shape(Z,True) # plot copy

# Decorations
ax.grid(linestyle='--', linewidth=0.25, color=[0.5,0.5,0.5])
plt.axis('equal'); plt.axis('square')
plt.axhline(y=0, color='k', linewidth = 0.25)
plt.axvline(x=0, color='k', linewidth = 0.25)
plt.xticks(np.arange(-5, 6)); plt.yticks(np.arange(-5, 6))
ax.set_xlim(-5,5); ax.set_ylim(-5,5)
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.spines['bottom'].set_visible(False)
ax.spines['left'].set_visible(False)
plt.xlabel('$x_1$'); plt.ylabel('$x_2$')
```

8.3 缩放：对角阵

等比例缩放 (equal scaling) 是指在缩放时各个维度采用相同缩放比例。

举个例子，如图 7 所示，横纵坐标等比例放大 2 倍，等比例缩放得到的图形和原图形相似。

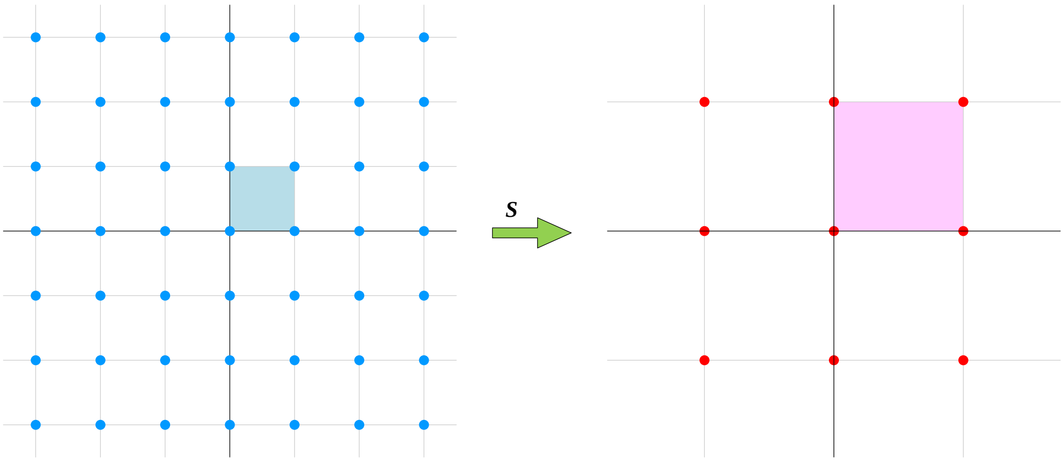


图 7. 等比例扩大 2 倍网格变化

列向量

等比例缩放对应的矩阵运算：

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{9}$$

可以发现等比例缩放矩阵为对角阵，对角线元素相同。(9) 整理得到：

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} sx_1 \\ sx_2 \end{bmatrix} = s \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{10}$$

等比例缩放矩阵和单位向量存在以下关系:

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = s\boldsymbol{I} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{11}$$

行向量

数据坐标为行向量时:

$$\begin{bmatrix} z_1 & z_2 \end{bmatrix} = s\begin{bmatrix} x_1 & x_2 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}\begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} \tag{12}$$

对于数据矩阵，等比例缩放为:

$$\boldsymbol{Z}_{n\times 2} = \boldsymbol{X}_{n\times 2}\begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} \tag{13}$$

行列式值

计算行列式值转化矩阵的行列式值:

$$\det\begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} = s^2 \tag{14}$$

可以发现对于二维空间，等比例缩放对应图形面积变化 s^2 倍。

图 8 (a) 所示的缩放系数为 $s = 3$，图形放大，面积放大 9 倍。图 8 (b) 所示的缩放系数为 $s = 0.75$，图形缩小，面积缩小为原来 9/16。

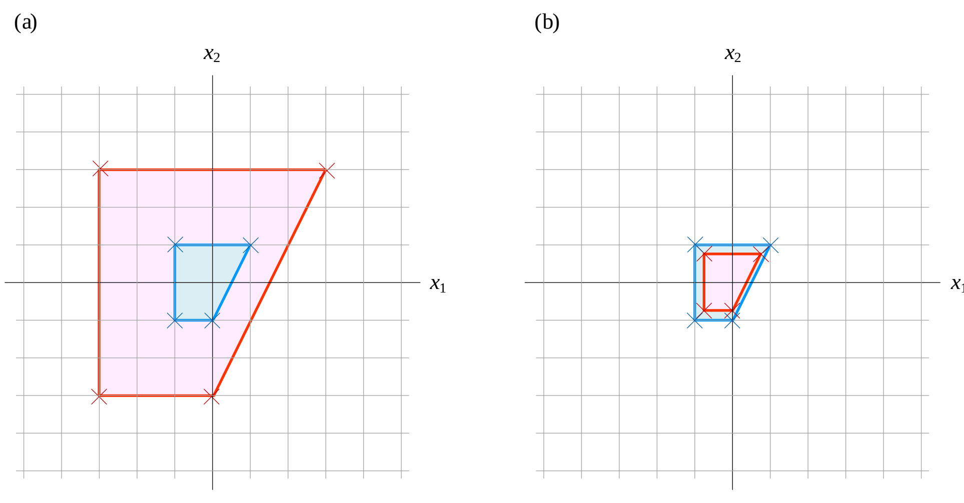


图 8. 等比例缩放两个例子

非等比例缩放

图 9 所示为**非等比例缩放** (unequal scaling) 的例子。

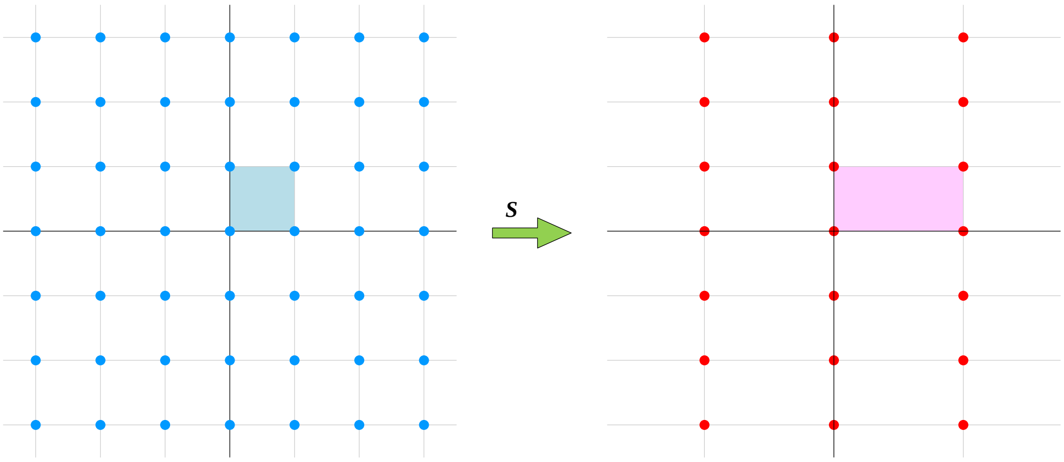


图 9. 非等比例缩放网格变化

非等比例缩放矩阵为:

$$\boldsymbol{S} = \boldsymbol{S}^{\mathrm{T}} = \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix} \tag{15}$$

数据点为列向量时,

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \boldsymbol{S}^{\mathrm{T}} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{16}$$

数据点为行向量时,

$$\begin{bmatrix} z_1 & z_2 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \boldsymbol{S} = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix} \tag{17}$$

请大家根据图 10 两幅子图中图形缩放前后横轴轴比例，来推断矩阵 $\boldsymbol{S}$ 的值分别是多少。

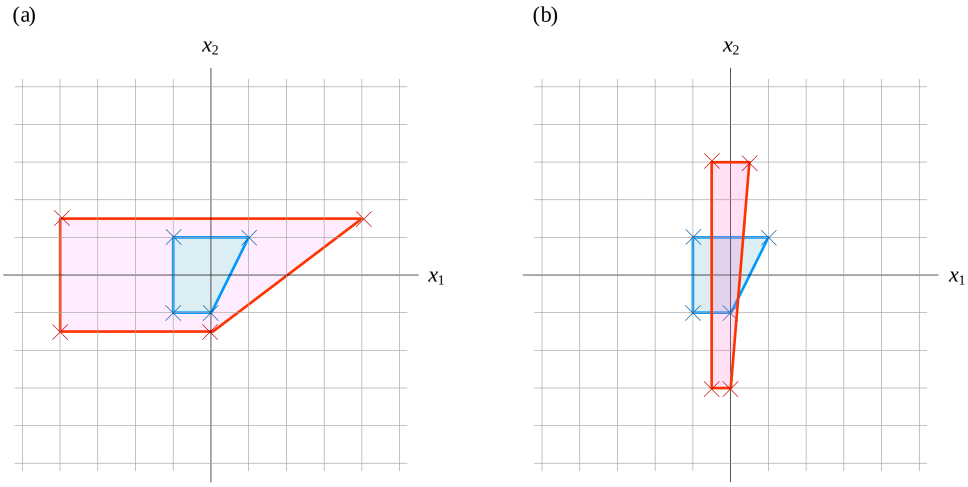


图 10. 非等比例缩放

逆矩阵

现在回过头来再看矩阵 $\boldsymbol{S}$ 的逆。

从线性变换角度，矩阵的逆 $\boldsymbol{S}^{-1}$ 无非就是矩阵 $\boldsymbol{S}$ 对应的几何变换“逆操作”。如图 11 所示，缩放操作的逆运算就是将缩放后图形再还原成原图形。

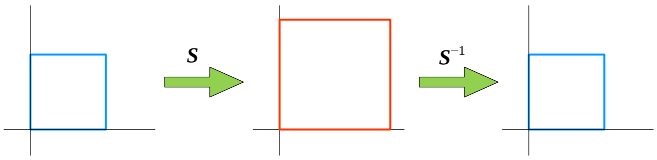


图 11. 缩放的逆运算

特别地，如果缩放时将图形完全压扁，比如：

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}}_{S} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{18}$$

(18) 中矩阵 $\boldsymbol{S}$ 的行列式值为 0，也就是说变换矩阵不可逆。容易发现，(18) 造成的形变也是不可逆的。

这样，我们从几何图形变换角度，解释为什么只有行列式值不为 0 的方阵才存在逆矩阵。

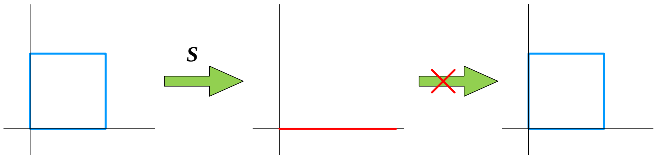


图 12. 不可逆缩放

联想到数据**标准化** (standardization) 这一概念。实际上，数据标准化就相当于是一种不同特征的数据先平移，然后再用标准差进行比例缩放。每个特征采用的缩放系数为均标准差的倒数。

数据矩阵 $\boldsymbol{X}$ 标准化得到数据矩阵 $\boldsymbol{Z}$，对应运算过程如下：

$$\boldsymbol{Z}_{n\times 2} = \left(\boldsymbol{X}_{n\times 2} - \mathrm{E}\left(\boldsymbol{X}_{n\times 2}\right)\right)\begin{bmatrix} 1/\sigma_1 & 0 \\ 0 & 1/\sigma_2 \end{bmatrix} \tag{19}$$

展开得到：

$$\begin{bmatrix} \boldsymbol{z}_1 & \boldsymbol{z}_2 \end{bmatrix} = \begin{bmatrix} \dfrac{\boldsymbol{x}_1 - \mathrm{E}\left(\boldsymbol{x}_1\right)}{\sigma_1} & \dfrac{\boldsymbol{x}_2 - \mathrm{E}\left(\boldsymbol{x}_2\right)}{\sigma_2} \end{bmatrix} \tag{20}$$

对应的统计运算则是：

$$\begin{cases} Z_1 = \dfrac{X_1 - \mathrm{E}\left(X_1\right)}{\sigma_1} \\ Z_2 = \dfrac{X_2 - \mathrm{E}\left(X_2\right)}{\sigma_2} \end{cases} \tag{21}$$

图 6 所示为数据标准化过程，值得注意的是旋转椭圆变成了正圆。此外，数据标准化并不改变相关性系数大小。

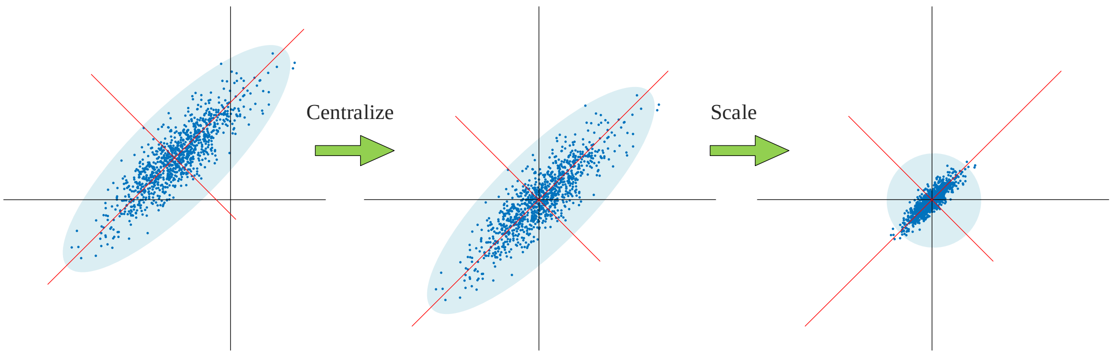


图 13. 数据标准化

挤压

还有一种特殊的缩放叫做**挤压** (squeeze)，比如竖直方向或水平方向压扁。图 14 所示为挤压对应的网格变化。

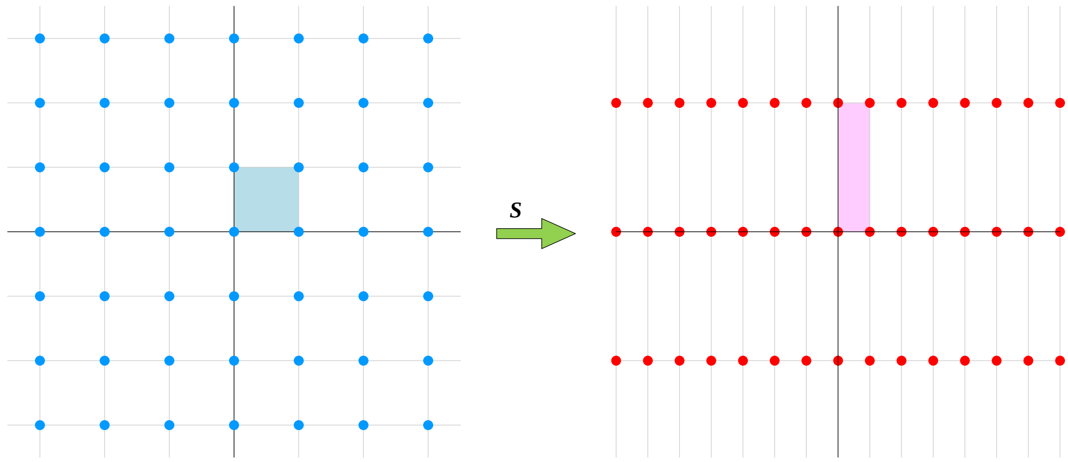


图 14. 挤压所对应的网格图变化

列向量

坐标为列向量时，挤压对应的矩阵运算为：

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} s & 0 \\ 0 & 1/s \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{22}$$

计算上式方阵行列式值，发现结果为 1，这说明挤压前后面积没有变化：

$$\det \begin{bmatrix} s & 0 \\ 0 & 1/s \end{bmatrix} = 1 \tag{23}$$

图 15 所示为几何图形挤压的两个例子，也请大家根据图形自行推断完成几何变换矩阵的具体数值，并计算行列式值。

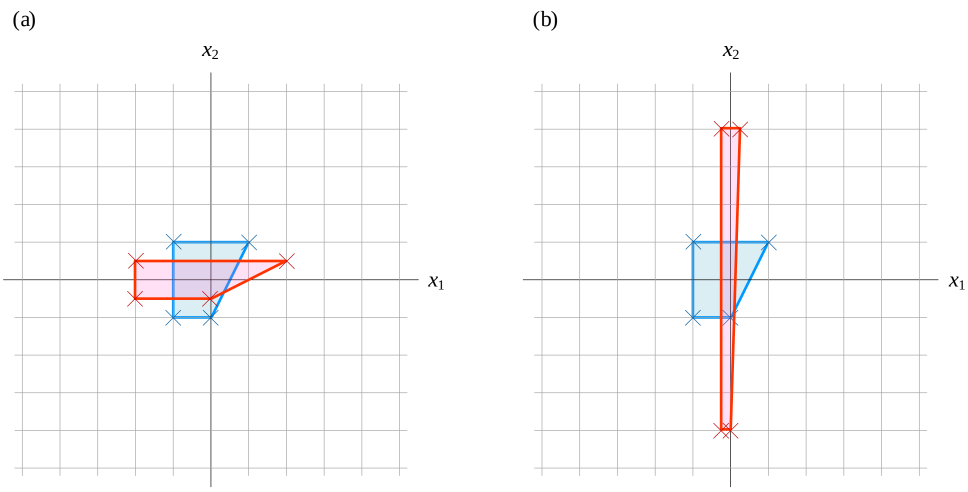


图 15. 挤压变换两个例子

8.4 旋转：行列式值为 1

本节介绍旋转，图 16 所示为经过旋转变化前后的网格。旋转是非常重要的几何变换，我们会在本书后续很多内容看到它。

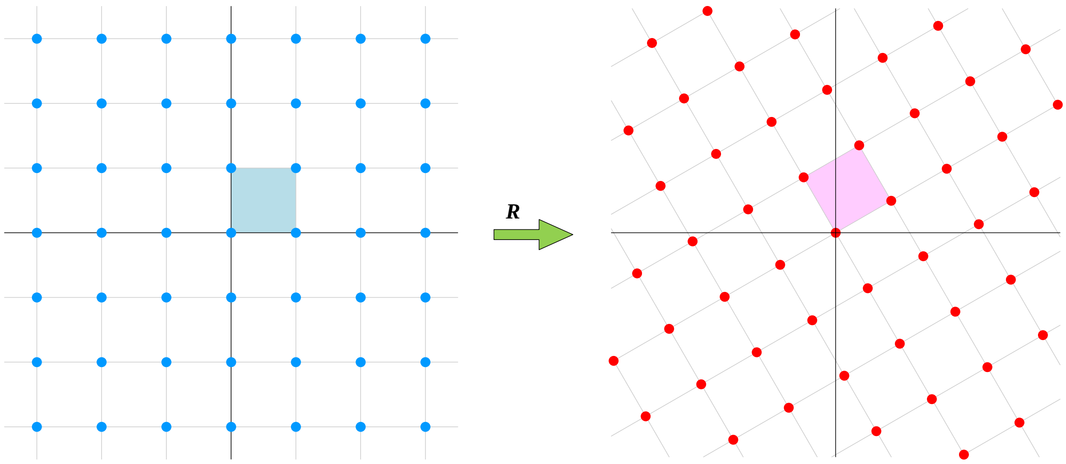


图 16. 旋转变换的网格

列向量坐标 $\boldsymbol{x}$ 逆时针旋转 θ 得到 $\boldsymbol{z}$：

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \boldsymbol{R}^{\mathrm{T}} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{24}$$

其中 $\boldsymbol{R}$ 为，

$$\boldsymbol{R} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \tag{25}$$

(25) 代入 (24)，得到下式：

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{26}$$

$\boldsymbol{R}$ 的行列式值为 1，也就是说旋转之后的网格面积不变：

$$\det \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} = 1 \tag{27}$$

行向量

坐标点为行向量，逆时针旋转 θ 对应的运算为：

$$\begin{bmatrix} z_1 & z_2 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \boldsymbol{R} \tag{28}$$

(25) 代入 (28) 得到:

$$\begin{bmatrix} z_1 & z_2 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \tag{29}$$

(29) 相当于对 (26) 等式左右转置。

对于数据矩阵情况，逆时针旋转的运算规则如下:

$$\boldsymbol{Z}_{n\times 2} = \boldsymbol{X}_{n\times 2}\boldsymbol{R} = \boldsymbol{X}_{n\times 2} \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \tag{30}$$

图 17 所示为几何形状旋转操作的两个例子。

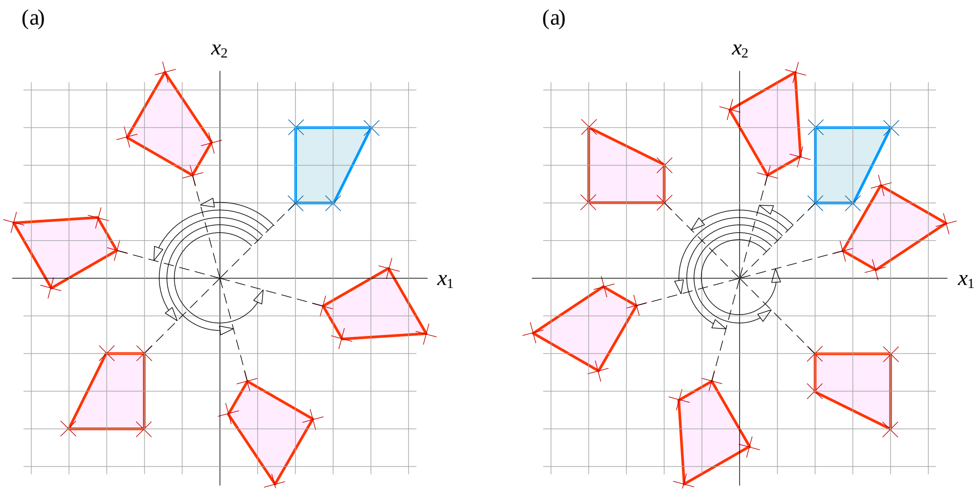


图 17. 旋转的两个例子

逆矩阵

旋转矩阵 $\boldsymbol{R}$ 求逆得到 $\boldsymbol{R}^{-1}$；如图 18 所示，几何角度，$\boldsymbol{R}^{-1}$ 是朝着相反方向旋转。

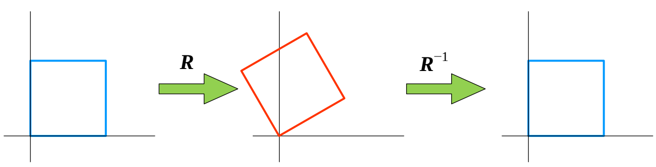


图 18. 旋转的逆运算

从数据角度看旋转操作，如图 19 所示。

数据按照特定的角度绕原点旋转后，让椭圆的长轴位于横轴。也就是说，旋转椭圆变成正椭圆。图 19 中，从左到右过程是主成分分析 (principal component analysis, PCA) 的思路。而从右到左的过程代表着，利用线性不相关随机数产生具有指定相关性系数的随机数。

这些内容，我们会在《概率统计》和《数据科学》两册书中深入讲解。

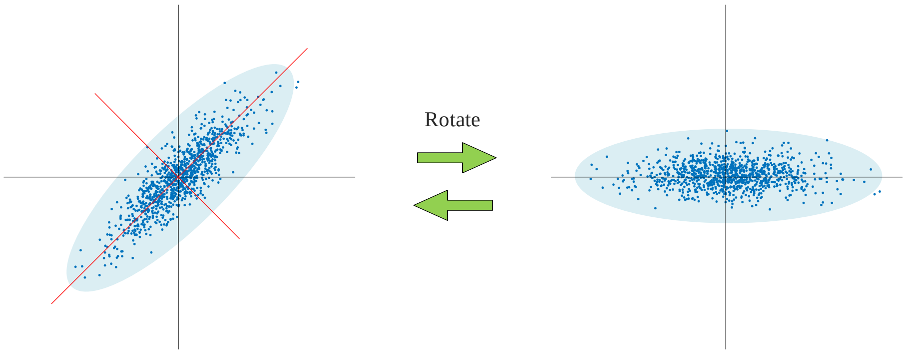


图 19. 数据视角下的旋转

矩阵乘法不满足交换律

前文讲过，一般来说，矩阵乘法不满足交换律，即，

$$\boldsymbol{AB} \neq \boldsymbol{BA} \tag{31}$$

现在我们用图形的线性变换来说明这一点。

图 20 所示左侧方格，先经过 $\boldsymbol{S}$ 缩放，再通过 $\boldsymbol{R}$ 旋转得到右侧红色网格。图 20 红色网格显然不同于图 21。因为图 21 红色网格是先通过 $\boldsymbol{R}$ 旋转再经过 $\boldsymbol{S}$ 缩放得到的。

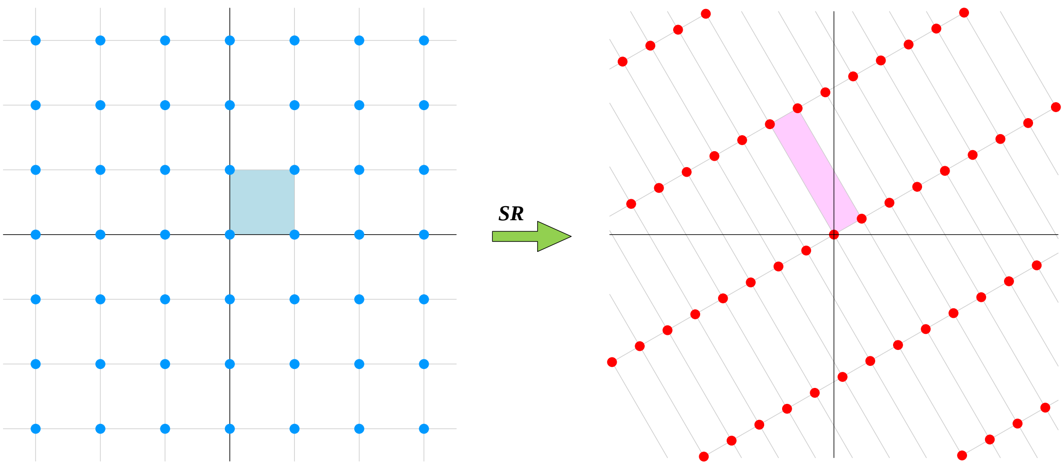


图 20. 先缩放再旋转

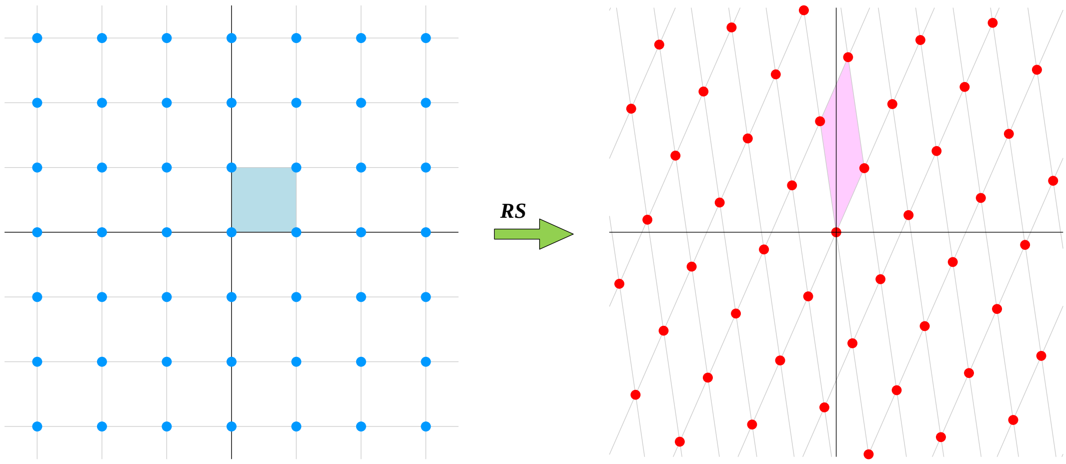


图 21. 先旋转再缩放

但是，两个 2 × 2 缩放矩阵相乘满足交换律，因为它们都是对角阵。下式的 $\boldsymbol{S}_1$ 和 $\boldsymbol{S}_2$ 均为缩放矩阵，相乘时交换顺序不影响结果：

$$\boldsymbol{S}_1\boldsymbol{S}_2 = \boldsymbol{S}_2\boldsymbol{S}_1 \tag{32}$$

采用极坐标推导本节给出的旋转变换。

图 22 给出的是向量 $\boldsymbol{a}$ 在极坐标系坐标为 (r, α)，在正交系中向量 $\boldsymbol{a}$ 的横纵坐标为：

$$\boldsymbol{a} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} r\cos\alpha \\ r\sin\alpha \end{bmatrix} \tag{33}$$

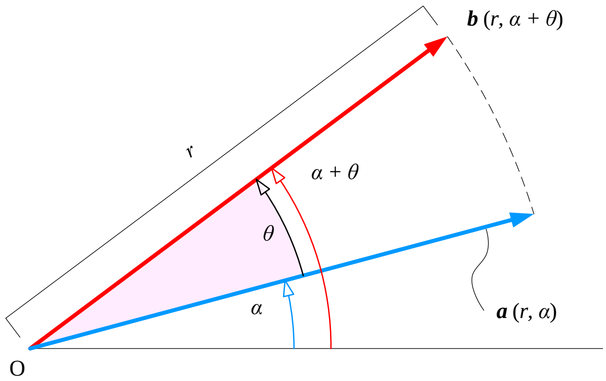


图 22. 极坐标中解释旋转

向量 $\boldsymbol{a}$ 逆时针旋转 θ 后，得到向量 $\boldsymbol{b}$。$\boldsymbol{b}$ 对应极坐标为 $(r, \alpha + \theta)$。向量 $\boldsymbol{b}$ 对应的横纵坐标为：

$$
\boldsymbol{b} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} r\cos(\alpha+\theta) \\ r\sin(\alpha+\theta) \end{bmatrix} \tag{34}
$$

(34) 展开得到：

$$
\boldsymbol{b} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} r\cos(\alpha+\theta) \\ r\sin(\alpha+\theta) \end{bmatrix} = \begin{bmatrix} \underbrace{r\cos\alpha}_{x_1}\cos\theta - \underbrace{r\sin\alpha}_{x_2}\sin\theta \\ \underbrace{r\sin\alpha}_{x_2}\cos\theta + \underbrace{r\cos\alpha}_{x_1}\sin\theta \end{bmatrix} \tag{35}
$$

将 (33) 中 x_1 和 x_2 代入 (35)，得到：

$$
\boldsymbol{b} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} x_1\cos\theta - x_2\sin\theta \\ x_2\cos\theta + x_1\sin\theta \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \cos\theta & \sin\theta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{36}
$$

Bk4_Ch8_02.py 绘制图 17。

```python
# Bk4_Ch8_02.py

import matplotlib.pyplot as plt
import numpy as np

def plot_shape(X,copy = False):
    if copy:
        fill_color = np.array([255,236,255])/255
        edge_color = np.array([255,0,0])/255
    else:
        fill_color = np.array([219,238,243])/255
        edge_color = np.array([0,153,255])/255

    plt.fill(X[:,0], X[:,1],
             color = fill_color,
             edgecolor = edge_color)

    plt.plot(X[:,0], X[:,1],marker = 'x',
             markeredgecolor = edge_color*0.5,
             linestyle = 'None')

X = np.array([[1,1],
              [0,-1],
              [-1,-1],
              [-1,1]]) + np.array([3,3])

# visualizations

thetas = np.linspace(30, 330, num=11)

for theta in thetas:

    fig, ax = plt.subplots()

    theta = theta/180*np.pi;
    # rotation
    R = np.array([[np.cos(theta),  np.sin(theta)],
                  [-np.sin(theta), np.cos(theta)]])

    Z = X@R;
    plot_shape(Z,True) # plot copy
    plot_shape(X)      # plot original

    # Decorations
    ax.grid(linestyle='--', linewidth=0.25, color=[0.5,0.5,0.5])
```

```
plt.axis('equal'); plt.axis('square')
plt.axhline(y=0, color='k', linewidth = 0.25)
plt.axvline(x=0, color='k', linewidth = 0.25)
plt.xticks(np.arange(-5, 6)); plt.yticks(np.arange(-5, 6))
ax.set_xlim(-5,5); ax.set_ylim(-5,5)
ax.spines['top'].set_visible(False); ax.spines['bottom'].set_visible(False)
ax.spines['left'].set_visible(False); ax.spines['right'].set_visible(False)
plt.xlabel('$x_1$'); plt.ylabel('$x_2$')
```

8.5 镜像：行列式值为负

本节介绍两种表达镜像的方法。

切向量

第一种镜像用切向量来完成。切向量 $\boldsymbol{\tau}$ 具体为：

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \tag{37}$$

关于通过原点、切向量为 $\boldsymbol{\tau}$ 直线**镜像** (reflection) 的线性变换操作如下：

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \underbrace{\frac{1}{\|\boldsymbol{\tau}\|^2} \begin{bmatrix} \tau_1^2 - \tau_2^2 & 2\tau_1\tau_2 \\ 2\tau_1\tau_2 & \tau_2^2 - \tau_1^2 \end{bmatrix}}_{\boldsymbol{T}} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{38}$$

对 $\boldsymbol{T}$ 求行列式值：

$$\det\left(\frac{1}{\|\boldsymbol{\tau}\|^2} \begin{bmatrix} \tau_1^2 - \tau_2^2 & 2\tau_1\tau_2 \\ 2\tau_1\tau_2 & \tau_2^2 - \tau_1^2 \end{bmatrix} \right) = \frac{-\left(\tau_1^2 - \tau_2^2\right)^2 - 4\tau_1^2\tau_2^2}{\|\boldsymbol{\tau}\|^4} = \frac{-\left(\tau_1^2 + \tau_2^2\right)^2}{\left(\tau_1^2 + \tau_2^2\right)^2} = -1 \tag{39}$$

$\boldsymbol{T}$ 的行列式值为负数，这说明线性变换前后图形发生翻转。图 23 给出两个镜像的例子。

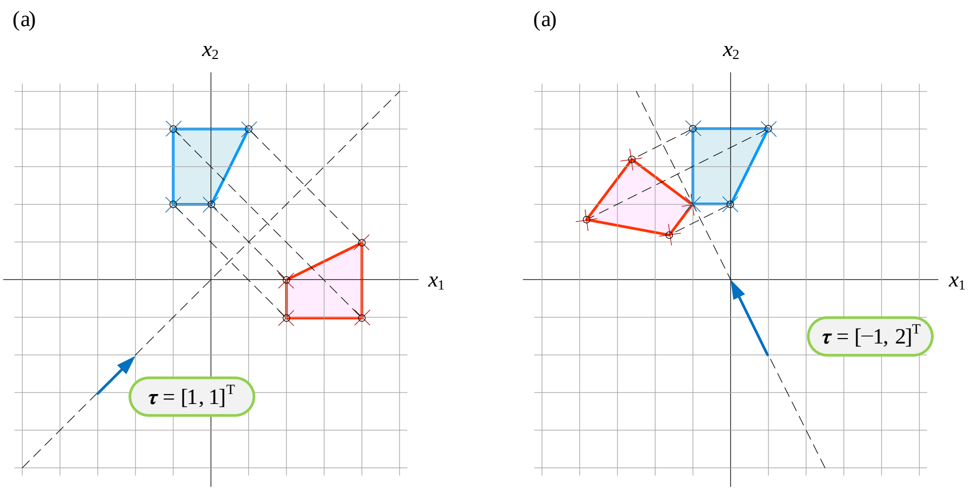


图 23. 两个镜像变换的例子

角度

第二种镜像通过角度定义。关于通过原点、方向和水平轴夹角为 θ 直线镜像，类比 (37)，直线的切向量相当于 $[\cos\theta, \sin\theta]^{\mathrm{T}}$，因此完成镜像的运算为：

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{40}$$

关于横纵轴镜像

关于横轴镜像对称：

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{41}$$

关于纵轴镜像对称：

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{42}$$

图 24 所示为分别关于横轴和纵轴镜像的例子，请大家自行计算矩阵的行列式值。

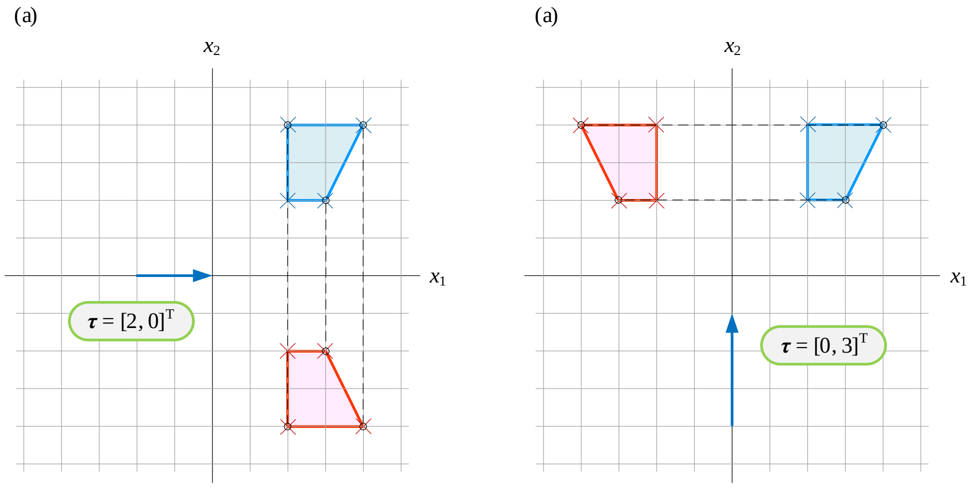


图 24. 分别关于横轴和纵轴镜像的例子

8.6 投影：降维操作

本节从几何角度简单介绍投影。不加特殊说明的话，本书中提到的投影都是正交投影 (orthogonal projection)。

切向量

假设原来某点的坐标为 (x_1, x_2)，向通过原点、切向量为 $\boldsymbol{\tau}$ $[\tau_1, \tau_2]^{\mathrm{T}}$ 直线方向**投影** (projection)，投影点坐标 (z_1, z_2) 为：

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \underbrace{\frac{1}{\|\boldsymbol{\tau}\|^2}\begin{bmatrix} \tau_1^2 & \tau_1\tau_2 \\ \tau_1\tau_2 & \tau_2^2 \end{bmatrix}}_{\boldsymbol{P}}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{43}$$

正交投影的特点是，(x_1, x_2) 和 (z_1, z_2) 两点连线垂直于 $\boldsymbol{\tau}$。

行列式值

上式中矩阵 $\boldsymbol{P}$ 的行列式值为 0：

$$\det\left(\frac{1}{\|\boldsymbol{\tau}\|^2}\begin{bmatrix} \tau_1^2 & \tau_1\tau_2 \\ \tau_1\tau_2 & \tau_2^2 \end{bmatrix}\right) = 0 \tag{44}$$

如图 25 所示，投影是一个降维的过程，平面网格坍塌成一条直线。

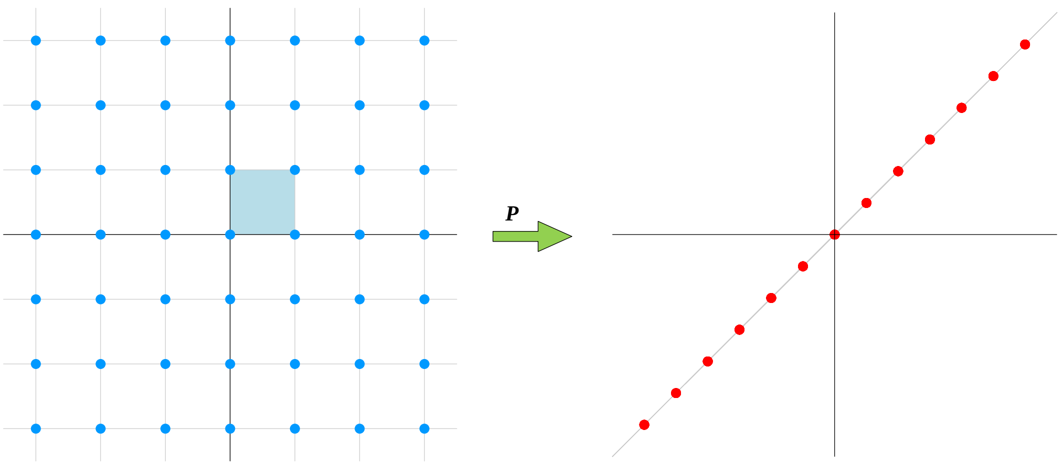


图 25. 投影网格

请大家自行计算图 26 中两个投影矩阵的具体值，并检验它们的行列式值是否为 0。

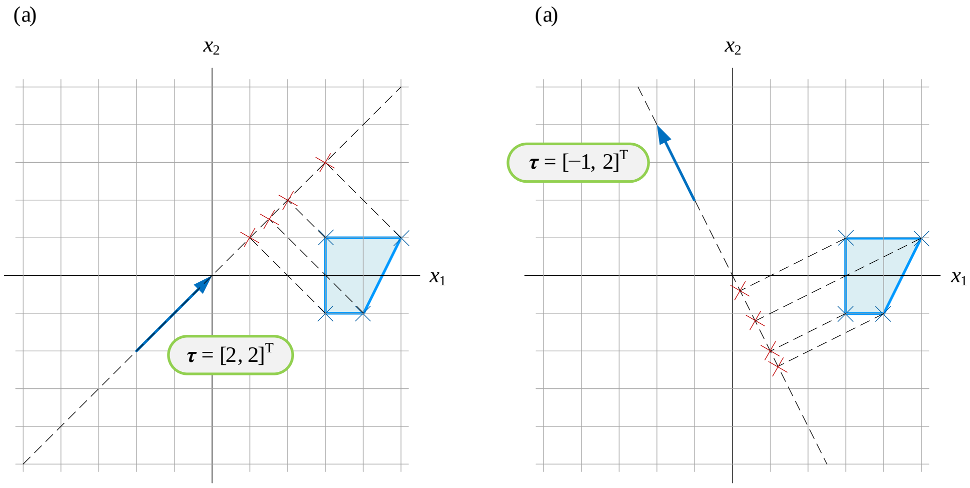


图 26. 两个投影例子

横纵轴

向横轴投影，相当于压扁到横轴：

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{45}$$

显然上式中矩阵的行列式值为 0。

向纵轴投影：

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{46}$$

秩

此外，大家自己可以计算一下矩阵 $\boldsymbol{P}$ 的秩，可以发现 $\operatorname{rank}(\boldsymbol{P}) = 1$。这说明，$\boldsymbol{P}$ 的列向量线性相关。简单整理 $\boldsymbol{P}$ 得到：

$$\boldsymbol{P} = \frac{1}{\|\boldsymbol{\tau}\|^2} \begin{bmatrix} \tau_1 \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} & \tau_2 \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \end{bmatrix} \tag{47}$$

的确，$\boldsymbol{P}$ 的列向量之间存在倍数关系。大家自行计算 (45) 和 (46) 中矩阵的秩。

张量积

再进一步，我们发现 (47) 可以写成：

$$P = \frac{1}{\|\boldsymbol{\tau}\|^2}\begin{bmatrix}\tau_1\begin{bmatrix}\tau_1\\\tau_2\end{bmatrix} & \tau_2\begin{bmatrix}\tau_1\\\tau_2\end{bmatrix}\end{bmatrix} = \frac{1}{\|\boldsymbol{\tau}\|^2}\begin{bmatrix}\tau_1\\\tau_2\end{bmatrix}@\begin{bmatrix}\tau_1 & \tau_2\end{bmatrix} = \frac{1}{\|\boldsymbol{\tau}\|}\begin{bmatrix}\tau_1\\\tau_2\end{bmatrix}@\left(\frac{1}{\|\boldsymbol{\tau}\|}\begin{bmatrix}\tau_1\\\tau_2\end{bmatrix}\right)^{\mathrm{T}} \tag{48}$$

容易发现，上式中存在本书前文讲过的向量单位化 (vector normalization)，也叫归一化。也就是说 $\boldsymbol{\tau}$ 单位化得到单位向量 (unit vector) $\hat{\boldsymbol{\tau}}$：

$$\hat{\boldsymbol{\tau}} = \frac{1}{\|\boldsymbol{\tau}\|}\begin{bmatrix}\tau_1\\\tau_2\end{bmatrix} \tag{49}$$

(48) 可以进一步写成张量积的形式，具体如下：

$$\boldsymbol{P} = \hat{\boldsymbol{\tau}}\hat{\boldsymbol{\tau}}^{\mathrm{T}} = \hat{\boldsymbol{\tau}} \otimes \hat{\boldsymbol{\tau}} \tag{50}$$

大家可能已经疑惑了，正交投影怎么和张量积联系起来了？卖个关子，我们把这个问题留给下两章回答。

8.7 再谈行列式值：几何视角

有了本章之前的内容，本节总结行列值的几何意义。

对于一个 2×2 矩阵 $\boldsymbol{A}$，它对应一定的线性变换，而 $\boldsymbol{A}$ 的行列式值决定了面积缩放系数。

行列值为正

举个例子，$\boldsymbol{i}$ 和 $\boldsymbol{j}$ 向量经过矩阵线性变换得到 $\boldsymbol{u}$ 和 $\boldsymbol{v}$:

$$\begin{bmatrix}3 & 1\\1 & 4\end{bmatrix}\underbrace{\begin{bmatrix}1\\0\end{bmatrix}}_{\boldsymbol{i}} = \underbrace{\begin{bmatrix}3\\1\end{bmatrix}}_{\boldsymbol{u}}, \quad \begin{bmatrix}3 & 1\\1 & 4\end{bmatrix}\underbrace{\begin{bmatrix}0\\1\end{bmatrix}}_{\boldsymbol{j}} = \underbrace{\begin{bmatrix}1\\4\end{bmatrix}}_{\boldsymbol{v}} \tag{51}$$

如图 27 所示，$\boldsymbol{i}$ 和 $\boldsymbol{j}$ 向量构成的正方形面积为 1。而 $\boldsymbol{u}$ 和 $\boldsymbol{v}$ 向量构成的平行四边形面积为 11，即对应 $|\boldsymbol{A}| = 11$，平面几何形状放大 11 倍。

反之，如果矩阵 $\boldsymbol{A}$ 的行列式值小于 1 大于 0，对应平面几何形状缩小。

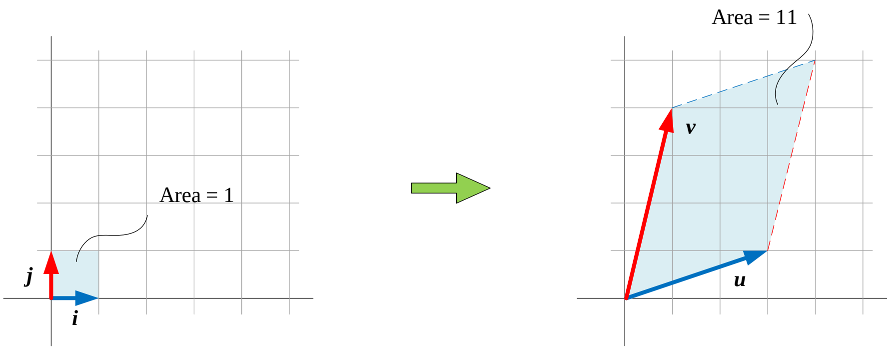


图 27. 行列式值为正

行列式值为 0

当然，行列式值可以为 0，也可以为负数。

如果矩阵 $\boldsymbol{A}$ 行列式值为 0 时，从几何上来讲，$\boldsymbol{A}$ 起到降维效果。我们看这个例子，$\boldsymbol{i}$ 和 $\boldsymbol{j}$ 向量经过矩阵线性变换得到 $\boldsymbol{u}$ 和 $\boldsymbol{v}$：

$$\begin{bmatrix} 2 & 4 \\ 2 & 4 \end{bmatrix} \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{\boldsymbol{i}} = \underbrace{\begin{bmatrix} 2 \\ 2 \end{bmatrix}}_{\boldsymbol{u}}, \quad \begin{bmatrix} 2 & 4 \\ 2 & 4 \end{bmatrix} \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{\boldsymbol{j}} = \underbrace{\begin{bmatrix} 4 \\ 4 \end{bmatrix}}_{\boldsymbol{v}} \tag{52}$$

如图 28 所示，$\boldsymbol{u}$ 和 $\boldsymbol{v}$ 向量共线，夹角为 0，这样它们构成图形的面积也为 0，对应 $|\boldsymbol{A}| = 0$。

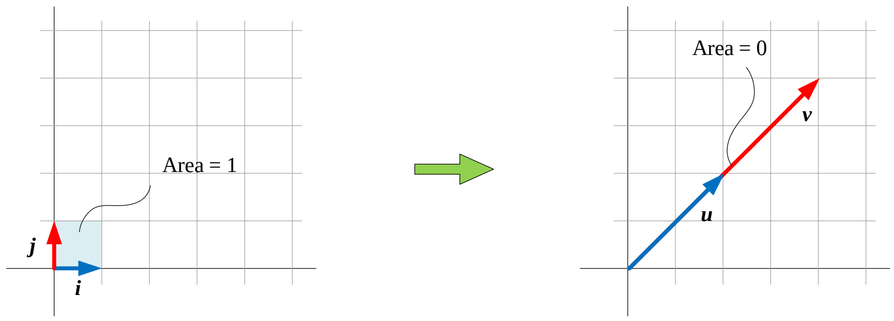


图 28. 行列式值为零

行列式值为负

如果矩阵 $\boldsymbol{A}$ 行列式值为负，几何上来看，几何图形翻转，如图 29 所示。图 29 中图形面积则放大了 10 倍 (行列式值的绝对值为 10)。请大家根据图 29 中 $\boldsymbol{u}$ 和 $\boldsymbol{v}$ 两个向量判断 $\boldsymbol{A}$ 的具体值。

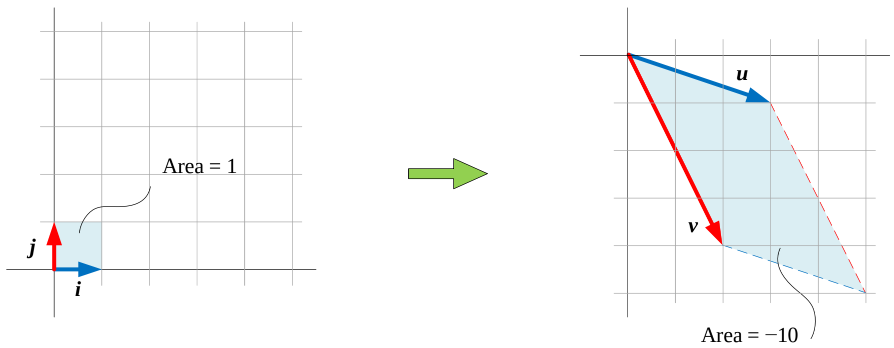


图 29. 行列式值为负

你可能会问，我们在本章几何变换中见到的矩阵绝大多数都不是对角方阵，计算面积或体积显然不容易。有没有一种办法能够将这些矩阵，还原成其原本面貌，也就是对角方阵？

也就是，将奇形怪状的几何体，通过平行移动转化成“立方体”，这样更容易计算面积或体积。举个例子，如图 30 所示，把平行四边形变成一个长方形。

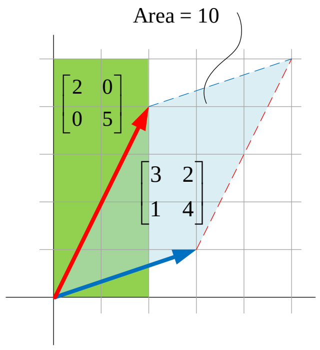


图 30. 把平行四边形变成长方形

答案是肯定的，这就是我们本书后续在矩阵分解中要着重讲解的特征值分解。此外，图 30 的 2 和 5 不是随心所欲定的，大家很快就会发现 2 和 5 叫做矩阵的特征值。

本章讲了很多种几何变换，请大家格外关注平移、缩放、旋转和投影。我们将会在接下来的内容中反复使用这四种几何变换。

此外，本章在讲解几何变换的同时，还和大家从几何角度回顾并探讨了矩阵可逆性、矩阵乘法不满足交换律、秩、行列式值等线性代数概念。请大家特别注意行列式值的几何视角，我们将在特征值分解中再进一步探讨。

用几何视角理解线性代数概念，是学习线性代数的唯一“捷径”。此外，数据视角会让大家看到线性代数的高度实用性，并直接和编程联结起来。

有数据的地方，就有向量！

有向量的地方，就有几何！

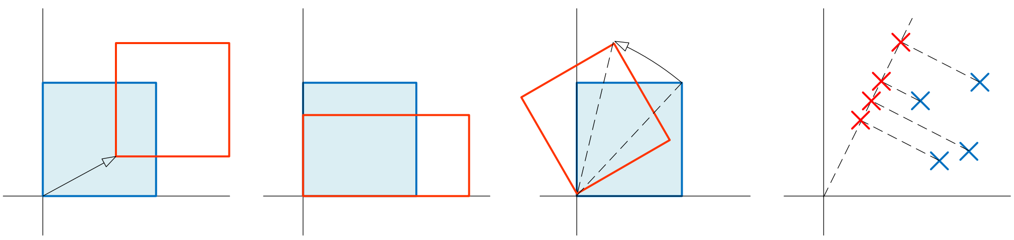


图 31. 总结本章重要内容的四副图