14 Dive into Eigen Decomposition 深入特征值分解

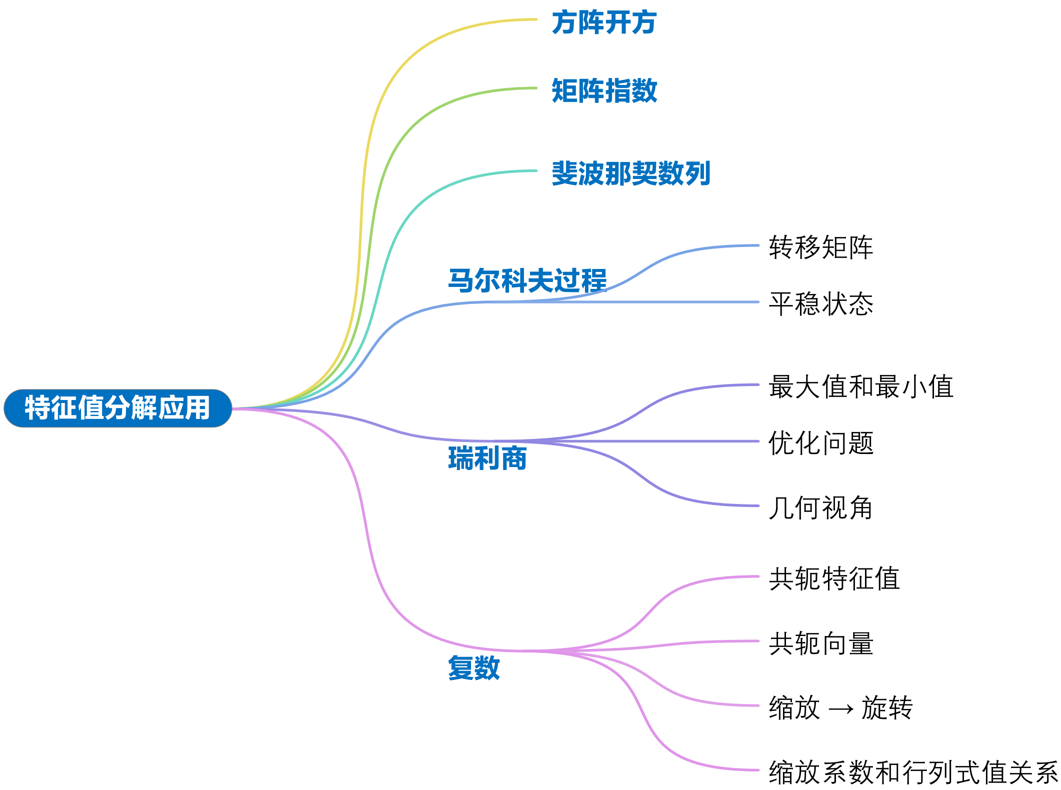
无处不在的特征值分解

生命之殇，并非求其上，却得其中；而是求其下，必得其下。

The greater danger for most of us lies not in setting our aim too high and falling short; but in setting our aim too low, and achieving our mark.

—— 米开朗琪罗 (Michelangelo) | 文艺复兴三杰之一 | 1475 ~ 1564

- `numpy.meshgrid()` 产生网格化数据
- `numpy.prod()` 指定轴的元素乘积
- `numpy.linalg.inv()` 矩阵求逆
- `numpy.linalg.eig()` 特征值分解
- `numpy.diag()` 以一维数组的形式返回方阵的对角线元素，或将一维数组转换成对角阵
- `seaborn.heatmap()` 绘制热图


特征值分解应用
方阵开方
矩阵指数
斐波那契数列
马尔科夫过程
转移矩阵
平稳状态
瑞利商
最大值和最小值
优化问题
几何视角
复数
共轭特征值
共轭向量
缩放 → 旋转
缩放系数和行列式值关系

14.1 方阵开方

本章是上一章的延续，继续探讨特征值及其应用。

这一节介绍利用特征值分解完成方阵开方。

如果方阵 $\boldsymbol{A}$ 可以写作：

$$\boldsymbol{A} = \boldsymbol{B}\boldsymbol{B} \tag{1}$$

$\boldsymbol{B}$ 是 $\boldsymbol{A}$ 的平方根。利用特征值分解，可以求得 $\boldsymbol{A}$ 的平方根。

首先对矩阵 $\boldsymbol{A}$ 特征值分解：

$$\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1} \tag{2}$$

令：

$$\boldsymbol{B} = \boldsymbol{V}\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{V}^{-1} \tag{3}$$

$\boldsymbol{B}^2$ 可以写成：

$$\boldsymbol{B}^2 = \left(\boldsymbol{V}\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{V}^{-1}\right)^2 = \boldsymbol{V}\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{V}^{-1}\boldsymbol{V}\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{V}^{-1} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1} = \boldsymbol{A} \tag{4}$$

注意，能特征值分解的矩阵存在平方根矩阵。

举个例子

给定如下方阵 $\boldsymbol{A}$，求解如下矩阵的平方根：

$$\boldsymbol{A} = \begin{bmatrix} 1.25 & -0.75 \\ -0.75 & 1.25 \end{bmatrix} \tag{5}$$

对 $\boldsymbol{A}$ 进行特征值分解得到：

$$\boldsymbol{A} = \begin{bmatrix} 1.25 & -0.75 \\ -0.75 & 1.25 \end{bmatrix} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1} = \begin{bmatrix} \sqrt{3}/2 & \sqrt{3}/2 \\ -\sqrt{3}/2 & \sqrt{3}/2 \end{bmatrix}\begin{bmatrix} 2 & 0 \\ 0 & 1/2 \end{bmatrix}\begin{bmatrix} \sqrt{3}/2 & -\sqrt{3}/2 \\ \sqrt{3}/2 & \sqrt{3}/2 \end{bmatrix} \tag{6}$$

矩阵 $\boldsymbol{B}$ 为：

$$\begin{aligned} \boldsymbol{B} = \boldsymbol{V}\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{V}^{-1} &= \begin{bmatrix} \sqrt{2}/2 & \sqrt{2}/2 \\ -\sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix}\begin{bmatrix} \sqrt{2} & 0 \\ 0 & \sqrt{2}/2 \end{bmatrix}\begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 \\ \sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1/2 \\ -1 & 1/2 \end{bmatrix}\begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 \\ \sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix} = \begin{bmatrix} 3\sqrt{2}/4 & -\sqrt{2}/4 \\ -\sqrt{2}/4 & 3\sqrt{2}/4 \end{bmatrix} \end{aligned} \tag{7}$$

Bk4_Ch14_01.py 求解上述例子中 $\boldsymbol{A}$ 的平方根。

```
# Bk4_Ch14_01.py

import numpy as np

A = np.matrix([[1.25, -0.75],
               [-0.75, 1.25]])

LAMBDA, V = np.linalg.eig(A)

B = V@np.diag(np.sqrt(LAMBDA))@np.linalg.inv(V)

A_reproduced = B@B

print(A_reproduced)
```

14.2 矩阵指数：幂级数的推广

给定一个标量 a，指数 e^a 可以用幂级数展开：

$$\mathrm{e}^a = \exp(a) = 1 + a + \frac{1}{2!}a^2 + \frac{1}{3!}a^3 + \cdots \tag{8}$$

类似地，对于方阵 $\boldsymbol{A}$，可以定义**矩阵指数** (matrix exponential) $\mathrm{e}^{\boldsymbol{A}}$ 为一个收敛幂级数：

$$\mathrm{e}^{\boldsymbol{A}} = \exp(\boldsymbol{A}) = \boldsymbol{I} + \boldsymbol{A} + \frac{1}{2!}\boldsymbol{A}^2 + \frac{1}{3!}\boldsymbol{A}^3 + \cdots \tag{9}$$

如果 $\boldsymbol{A}$ 可以特征值分解得到如下等式，计算 (9) 则容易很多：

$$\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1} \tag{10}$$

其中，

$$\boldsymbol{\Lambda} = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_D \end{bmatrix} \tag{11}$$

利用特征值分解，$\boldsymbol{A}^k$ 可以写作：

$$\boldsymbol{A}^k = \boldsymbol{V}\boldsymbol{\Lambda}^k\boldsymbol{V}^{-1} \tag{12}$$

其中，k 为非负整数。

将 (12) 代入 (9)，得到：

$$
\begin{aligned}
\mathrm{e}^{\boldsymbol{A}} = \exp(\boldsymbol{A}) &= \boldsymbol{V}\boldsymbol{V}^{-1} + \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1} + \frac{1}{2!}\boldsymbol{V}\boldsymbol{\Lambda}^2\boldsymbol{V}^{-1} + \frac{1}{3!}\boldsymbol{V}\boldsymbol{\Lambda}^3\boldsymbol{V}^{-1} + \cdots \\
&= \boldsymbol{V}\left(\boldsymbol{I} + \boldsymbol{\Lambda} + \boldsymbol{\Lambda}^2 + \boldsymbol{\Lambda}^3 + \cdots\right)\boldsymbol{V}^{-1}
\end{aligned} \tag{13}
$$

特别地，对角方阵 $\boldsymbol{\Lambda}$ 矩阵指数为：

$$
\mathrm{e}^{\boldsymbol{\Lambda}} = \exp(\boldsymbol{\Lambda}) = \boldsymbol{I} + \boldsymbol{\Lambda} + \frac{1}{2!}\boldsymbol{\Lambda}^2 + \frac{1}{3!}\boldsymbol{\Lambda}^3 + \cdots \tag{14}
$$

容易计算对角阵 $\boldsymbol{\Lambda}$ 矩阵指数 $\mathrm{e}^{\boldsymbol{\Lambda}}$：

$$
\begin{aligned}
\mathrm{e}^{\boldsymbol{\Lambda}} = \exp(\boldsymbol{\Lambda}) &= \boldsymbol{I} + \boldsymbol{\Lambda} + \frac{1}{2!}\boldsymbol{\Lambda}^2 + \frac{1}{3!}\boldsymbol{\Lambda}^3 + \cdots \\
&= \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} + \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_D \end{bmatrix} + \frac{1}{2!}\begin{bmatrix} \lambda_1^2 & & & \\ & \lambda_2^2 & & \\ & & \ddots & \\ & & & \lambda_D^2 \end{bmatrix} + \cdots \\
&= \lim_{n\to\infty}\begin{bmatrix} \sum_{k=1}^{n}\frac{1}{k!}\lambda_1^k & & & \\ & \sum_{k=1}^{n}\frac{1}{k!}\lambda_2^k & & \\ & & \ddots & \\ & & & \sum_{k=1}^{n}\frac{1}{k!}\lambda_D^k \end{bmatrix} \\
&= \begin{bmatrix} \mathrm{e}^{\lambda_1} & & & \\ & \mathrm{e}^{\lambda_2} & & \\ & & \ddots & \\ & & & \mathrm{e}^{\lambda_D} \end{bmatrix}
\end{aligned} \tag{15}
$$

将 (14) 代入 (13)，得到：

$$
\mathrm{e}^{\boldsymbol{A}} = \boldsymbol{V}\,\mathrm{e}^{\boldsymbol{\Lambda}}\boldsymbol{V}^{-1} \tag{16}
$$

将 (15) 代入 (16)，得到：

$$
\mathrm{e}^{\boldsymbol{A}} = \boldsymbol{V}\begin{bmatrix} \mathrm{e}^{\lambda_1} & & & \\ & \mathrm{e}^{\lambda_2} & & \\ & & \ddots & \\ & & & \mathrm{e}^{\lambda_D} \end{bmatrix}\boldsymbol{V}^{-1} \tag{17}
$$

可以用 `scipy.linalg.expm()` 计算矩阵指数。

14.3 斐波那契数列：求通项式

本系列丛书《数学要素》介绍过**斐波那契数列** (Fibonacci number)，本节介绍如何使用特征值分解推导得到斐波那契数列通项解析式。

斐波那契数列可以通过如下**递归** (recursion) 方法获得：

$$\begin{cases} F_0 = 0 \\ F_1 = F_2 = 1 \\ F_n = F_{n-1} + F_{n-2}, \quad n > 2 \end{cases} \tag{18}$$

包括第 0 项，斐波那契数列的前 10 项为：

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55 \tag{19}$$

构造列向量

将斐波那契数列每连续两项写成列向量：

$$\boldsymbol{x}_0 = \begin{bmatrix} F_0 \\ F_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \boldsymbol{x}_1 = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{x}_2 = \begin{bmatrix} F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad \boldsymbol{x}_3 = \begin{bmatrix} F_3 \\ F_4 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \quad \boldsymbol{x}_4 = \begin{bmatrix} F_4 \\ F_5 \end{bmatrix} = \begin{bmatrix} 3 \\ 5 \end{bmatrix}, \cdots \tag{20}$$

图 1 所示为列向量连续变化过程，能够看到它们逐渐收敛到一条直线上。这条直线通过原点，斜率就是**黄金分割** (golden ratio)：

$$\varphi = \frac{\sqrt{5}+1}{2} \approx 1.61803 \tag{21}$$

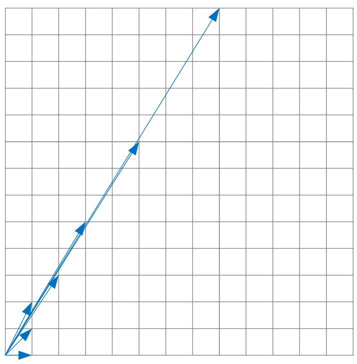


图 1. 斐波那契数列列向量连续变化过程

连续列向量间关系

数列的第 $k+1$ 项 $\boldsymbol{x}_{k+1}$ 和第 k 项 $\boldsymbol{x}_k$ 之间的关系可以写成如下矩阵运算：

$$\boldsymbol{x}_{k+1} = \begin{bmatrix} F_{k+1} \\ F_{k+2} \end{bmatrix} = \boldsymbol{A}\boldsymbol{x}_k = \boldsymbol{A}\begin{bmatrix} F_k \\ F_{k+1} \end{bmatrix} \tag{22}$$

其中

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \tag{23}$$

因此 $\boldsymbol{x}_k$ 可以写成：

$$\begin{aligned} \boldsymbol{x}_k &= \boldsymbol{A}\boldsymbol{x}_{k-1} \\ &= \boldsymbol{A}^2\boldsymbol{x}_{k-2} \\ &= \boldsymbol{A}^3\boldsymbol{x}_{k-3} \\ &\cdots \\ &= \boldsymbol{A}^k\boldsymbol{x}_0 \end{aligned} \tag{24}$$

特征值分解

对 $\boldsymbol{A}$ 进行特征值分解：

$$\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1} \tag{25}$$

其中，

$$\boldsymbol{\Lambda} = \begin{bmatrix} \lambda_1 & \\ & \lambda_2 \end{bmatrix}, \quad \boldsymbol{V} = \begin{bmatrix} 1 & 1 \\ \lambda_1 & \lambda_2 \end{bmatrix}, \quad \boldsymbol{V}^{-1} = \frac{1}{\lambda_2 - \lambda_1}\begin{bmatrix} \lambda_2 & -1 \\ -\lambda_1 & 1 \end{bmatrix} \tag{26}$$

$\boldsymbol{A}$ 的特征方程为：

$$\lambda^2 - \lambda - 1 = 0 \tag{27}$$

求解 (27)，可以得到两个特征值：

$$\lambda_1 = \frac{1-\sqrt{5}}{2}, \quad \lambda_2 = \frac{1+\sqrt{5}}{2} \tag{28}$$

$\boldsymbol{x}_k$ 可以写成：

$$\boldsymbol{x}_k = \boldsymbol{V}\boldsymbol{\Lambda}^k\boldsymbol{V}^{-1}\boldsymbol{x}_0 \tag{29}$$

将 (26) 代入 (29)，得到：

$$\begin{aligned} \boldsymbol{x}_k &= \frac{1}{\lambda_2 - \lambda_1}\begin{bmatrix} 1 & 1 \\ \lambda_1 & \lambda_2 \end{bmatrix}\begin{bmatrix} \lambda_1^k & \\ & \lambda_2^k \end{bmatrix}\begin{bmatrix} \lambda_2 & -1 \\ -\lambda_1 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ &= \frac{1}{\lambda_2 - \lambda_1}\begin{bmatrix} \lambda_2^k - \lambda_1^k \\ \lambda_2^{k+1} - \lambda_1^{k+1} \end{bmatrix} \end{aligned} \tag{30}$$

即，

$$\begin{bmatrix} F_k \\ F_{k+1} \end{bmatrix} = \frac{1}{\lambda_2 - \lambda_1} \begin{bmatrix} \lambda_2^k - \lambda_1^k \\ \lambda_2^{k+1} - \lambda_1^{k+1} \end{bmatrix} \tag{31}$$

确定通项式

因此 F_k 可以写成：

$$F_k = \frac{\lambda_2^k - \lambda_1^k}{\lambda_2 - \lambda_1} \tag{32}$$

将 (28) 代入 (32) 得到 F_k 解析式：

$$F_k = \frac{\left(\frac{1+\sqrt{5}}{2}\right)^k - \left(\frac{1-\sqrt{5}}{2}\right)^k}{\sqrt{5}} \tag{33}$$

至此，我们通过特征值分解得到斐波那契数列通项式解析式。

14.4 马尔科夫过程的平稳状态

本系列丛书在《数学要素》中介绍过一个“鸡兔互变”的有趣例子。例子中，鸡兔之间存在一定比例的相互转化。图 2 描述鸡兔互变的比例，每晚有 30%的小鸡变成小兔，其他小鸡不变；同时，每晚有 20%小兔变成小鸡，其余小兔不变。这个转化的过程叫做**马尔科夫过程** (Markov process)。

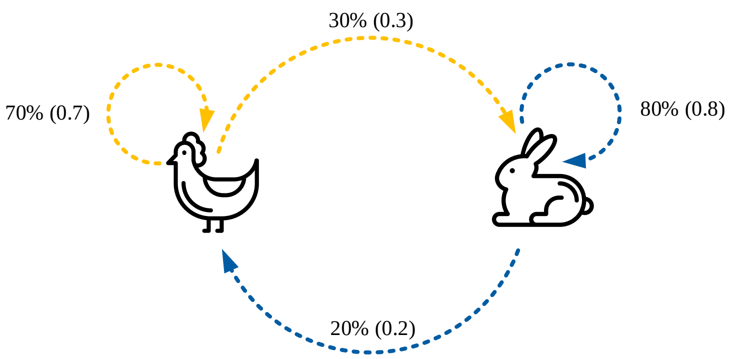


图 2. 鸡兔互变的比例

马尔科夫过程满足以下三个性质：(1) 可能输出状态有限；(2) 下一步输出的概率仅仅依赖上一步的输出状态；(3) 概率值相对于时间为常数。

“鸡兔互变”这个例子中，第 k 天，鸡兔的比例用列向量 $\boldsymbol{\pi}(k)$ 表示；其中，$\boldsymbol{\pi}(k)$ 第一行元素代表小鸡的比例，第二行元素代表小兔的比例。第 $k+1$ 天，鸡兔的比例用列向量 $\boldsymbol{\pi}(k+1)$ 表示。

变化的比例写成方阵 $\boldsymbol{T}$，$\boldsymbol{T}$ 通常叫做**转移矩阵** (transition matrix)。

这样 $k \to k+1$ 变化过程可以写成：

$$k \to k+1: \quad \boldsymbol{T}\boldsymbol{\pi}(k) = \boldsymbol{\pi}(k+1) \tag{34}$$

对于鸡兔互变，$\boldsymbol{T}$ 为：

$$\boldsymbol{T} = \begin{bmatrix} 0.7 & 0.2 \\ 0.3 & 0.8 \end{bmatrix} \tag{35}$$

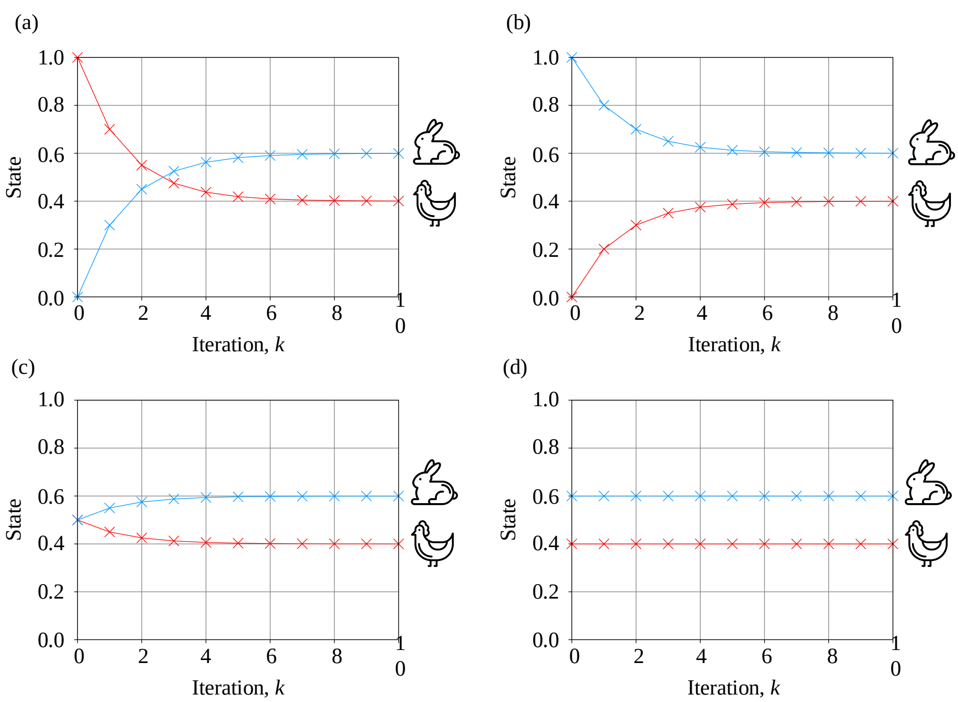


图 3. 不同初始状态条件下平稳状态

求平稳状态

如图 3 所示，我们初步得出结论不管初始状态向量 ($k = 0$) 如何，鸡兔比例最后都达到了一定的平衡，也就是：

$$\boldsymbol{T}\boldsymbol{\pi} = \boldsymbol{\pi} \tag{36}$$

有了本书特征值分解相关的知识，相信大家一眼就看出来 (36) 代表的关系就是特征值分解。

看过本系列丛书《数学要素》一册的读者应该还记得图 4 这幅图，它从几何视角描述了不同初始状态向量条件下，经过连续 12 次变化，向量都收敛于同一方向。

对 $\boldsymbol{T}$ 进行特征值分解得到两个特征向量：

$$\boldsymbol{v}_1 = \begin{bmatrix} -0.707 \\ 0.707 \end{bmatrix}, \quad \boldsymbol{v}_2 = \begin{bmatrix} 0.5547 \\ 0.8321 \end{bmatrix} \tag{37}$$

显然，鸡兔总比例之和为 1。因此 $\boldsymbol{\pi}$ 的两个元素之和为 1，且元素取值均非负，因此选择 $\boldsymbol{v}_2$ 来计算 $\boldsymbol{\pi}$：

$$\boldsymbol{\pi} = \frac{1}{0.5547 + 0.8321}\boldsymbol{v}_2 = \frac{1}{0.5547 + 0.8321}\begin{bmatrix} 0.5547 \\ 0.8321 \end{bmatrix} = \begin{bmatrix} 0.4 \\ 0.6 \end{bmatrix} \tag{38}$$

这个 $\boldsymbol{\pi}$ 叫做**平稳状态** (steady state)。

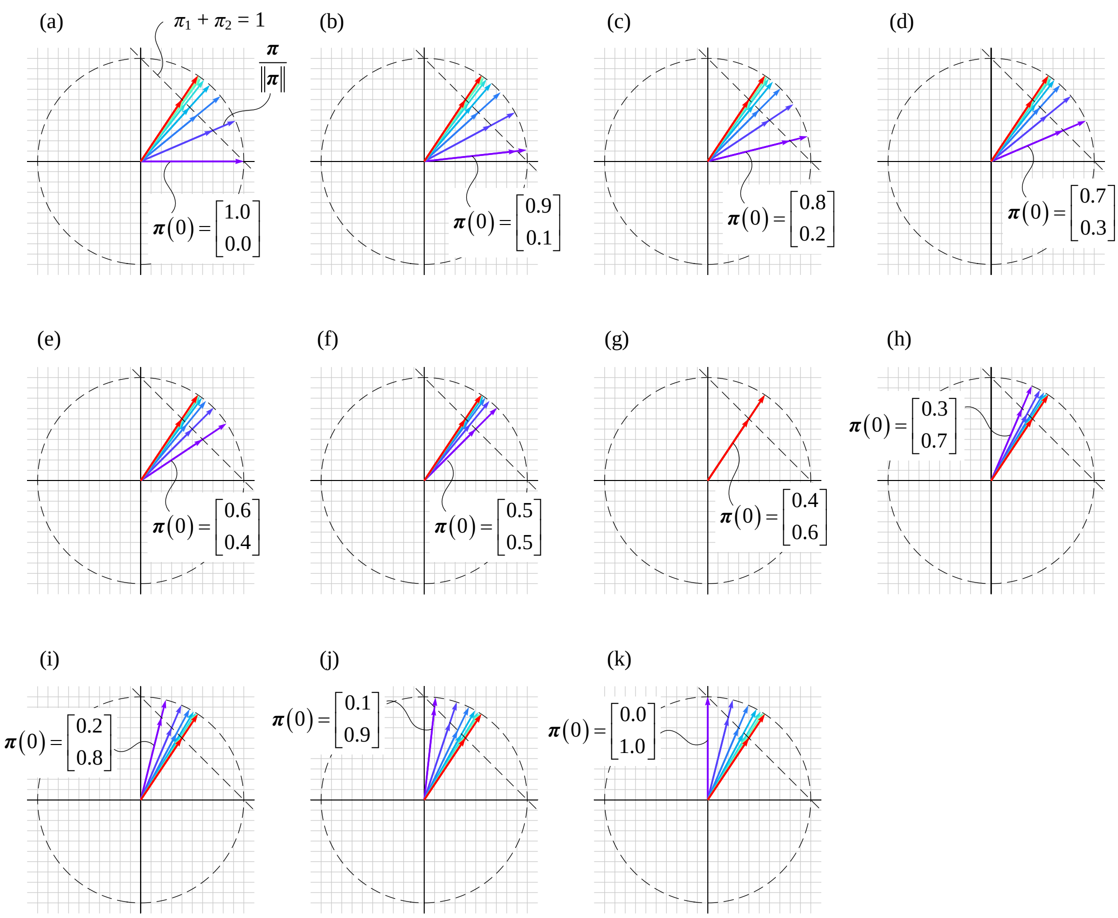


图 4. 连续 12 夜鸡兔互变比例，几何视角，图片来自《数学要素》

`Bk4_Ch14_02.py` 绘制图 3。

```python
# Bk4_Ch14_02.py

import numpy as np
import matplotlib.pyplot as plt

# transition matrix
T = np.matrix([[0.7, 0.2],
               [0.3, 0.8]])

# steady state
sstate = np.linalg.eig(T)[1][:,1]
sstate = sstate/sstate.sum()
print(sstate)

# initial states
initial_x_array = np.array([[1, 0, 0.5, 0.4],  # Chicken
                            [0, 1, 0.5, 0.6]]) # Rabbit

num_iterations = 10;

for i in np.arange(0,4):

    initial_x = initial_x_array[:,i][:, None]

    x_i = np.zeros_like(initial_x)
    x_i = initial_x
    X =   initial_x.T;

    # matrix power through iterations

    for x in np.arange(0,num_iterations):
        x_i = T@x_i;
        X = np.concatenate([X, x_i.T],axis = 0)

    fig, ax = plt.subplots()

    itr = np.arange(0,num_iterations+1);
    plt.plot(itr,X[:,0],marker = 'x',color = (1,0,0))
    plt.plot(itr,X[:,1],marker = 'x',color = (0,0.6,1))

    ax.grid(linestyle='--', linewidth=0.25, color=[0.5,0.5,0.5])
    ax.set_xlim(0, num_iterations)
    ax.set_ylim(0, 1)
    ax.set_xlabel('Iteration, k')
    ax.set_ylabel('State')
```

14.5 瑞利商

瑞利商 (Rayleigh quotient) 在很多机器学习算法中扮演重要角色，瑞利商和特征值分解有着密切的关系。本节利用几何视角可视化瑞利商，让大家深入理解瑞利商这个概念。

定义

给定实数对称矩阵 $\boldsymbol{A}$，它的瑞利商定义为：

$$R(\boldsymbol{x})=\frac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}} \tag{39}$$

其中，$\boldsymbol{x} = [x_1, x_2, \dots, x_D]^{\mathrm{T}}$。注意，$\boldsymbol{x}$ 不能为零向量，也就是说，$x_1, x_2, \dots, x_D$ 不能同时为 0。

先给出结论，瑞利商 $R(\boldsymbol{x})$ 的取值范围：

$$\lambda_{\min} \le R(\boldsymbol{x}) \le \lambda_{\max} \tag{40}$$

其中，$\lambda_{\min}$ 和 $\lambda_{\max}$ 分别为矩阵 $\boldsymbol{A}$ 的最小和最大特征值。

最大值和最小值

求解 $R(\boldsymbol{x})$ 的最大、最小值，等价于 $R(\boldsymbol{x})$ 分母为定值条件下，求解分子的最大值和最小值。一般情况下，给定如下分母条件：

$$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}=\|\boldsymbol{x}\|_2^2=1 \quad \Leftrightarrow \quad \|\boldsymbol{x}\|_2=1 \tag{41}$$

$\boldsymbol{A}$ 为对称矩阵，对其特征值分解得到：

$$\boldsymbol{A}=\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{\mathrm{T}} \tag{42}$$

$R(\boldsymbol{x})$ 的分子可以写成：

$$\left(\boldsymbol{V}^{\mathrm{T}}\boldsymbol{x}\right)^{\mathrm{T}}\boldsymbol{\Lambda}\left(\boldsymbol{V}^{\mathrm{T}}\boldsymbol{x}\right)=\left(\boldsymbol{V}^{\mathrm{T}}\boldsymbol{x}\right)^{\mathrm{T}}\begin{bmatrix}\lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_D\end{bmatrix}\left(\boldsymbol{V}^{\mathrm{T}}\boldsymbol{x}\right) \tag{43}$$

令

$$\boldsymbol{y}=\boldsymbol{V}^{\mathrm{T}}\boldsymbol{x} \tag{44}$$

这样，(44) 可以写成：

$$\boldsymbol{y}^{\mathrm{T}}\begin{bmatrix}\lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_D\end{bmatrix}\boldsymbol{y}=\begin{bmatrix}y_1\\ y_2\\ \vdots\\ y_D\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}\lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_D\end{bmatrix}\begin{bmatrix}y_1\\ y_2\\ \vdots\\ y_D\end{bmatrix}=\lambda_1 y_1^2+\lambda_2 y_2^2+\cdots+\lambda_D y_D^2 \tag{45}$$

类似地，$R(\boldsymbol{x})$ 的分母可以写成：

$$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}=\left(\boldsymbol{V}^{\mathrm{T}}\boldsymbol{x}\right)^{\mathrm{T}}\left(\boldsymbol{V}^{\mathrm{T}}\boldsymbol{x}\right)=\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}=y_1^2+y_2^2+\cdots+y_D^2=1 \tag{46}$$

这样，瑞利商就可以简洁地写成以 $\boldsymbol{y}$ 为自变量的函数 $R(\boldsymbol{y})$：

$$R(\boldsymbol{y})=\frac{\lambda_1 y_1^2+\lambda_2 y_2^2+\cdots+\lambda_D y_D^2}{y_1^2+y_2^2+\cdots+y_D^2} \tag{47}$$

优化问题

以下两个优化问题等价，而且都可以用本节介绍的瑞利商求解：

$$\max \frac{\|\boldsymbol{A}\boldsymbol{x}\|}{\|\boldsymbol{x}\|} \quad \Rightarrow \quad \max \left(\frac{\|\boldsymbol{A}\boldsymbol{x}\|}{\|\boldsymbol{x}\|}\right)^2 = \frac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}} \tag{48}$$

式中，对 $\boldsymbol{A}$ 是否为对称矩阵没有限制，因为 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ 为对称矩阵。求解 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ 的特征值最大值和最小值，便可以解决这个优化问题。本书后续讲解拉格朗日乘子法时，会再深入探讨这个优化问题。

举个例子

下面，我们以 2 × 2 矩阵为例，讲解如何求解瑞利商。给定 $\boldsymbol{A}$ 为

$$\boldsymbol{A} = \begin{bmatrix} 1.5 & 0.5 \\ 0.5 & 1.5 \end{bmatrix} \tag{49}$$

$R(\boldsymbol{x})$ 为：

$$R(\boldsymbol{x}) = \frac{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} 1.5 & 0.5 \\ 0.5 & 1.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}} = \frac{1.5x_1^2 + x_1x_2 + 1.5x_2^2}{x_1^2 + x_2^2} \tag{50}$$

$\boldsymbol{A}$ 的两个特征值分别为 $\lambda_1 = 2$，$\lambda_2 = 1$。$R(\boldsymbol{x})$ 等价于 $R(\boldsymbol{y})$，根据 (47) $R(\boldsymbol{y})$ 写成：

$$R(\boldsymbol{y}) = \frac{y_1^2 + 2y_2^2}{y_1^2 + y_2^2} \tag{51}$$

推导最值

求解 $R(\boldsymbol{y})$ 的最大、最小值，等价于 $R(\boldsymbol{y})$ 分母为 1 条件下，分子的最大值和最小值。

简单推导 $R(\boldsymbol{y})$ 最大值：

$$R(\boldsymbol{y}) = y_1^2 + 2y_2^2 \le 2\underbrace{\left(y_1^2 + y_2^2\right)}_{1} = 2 \tag{52}$$

推导 $R(\boldsymbol{y})$ 最小值：

$$R(\boldsymbol{y}) = y_1^2 + 2y_2^2 \ge \underbrace{\left(y_1^2 + y_2^2\right)}_{1} = 1 \tag{53}$$

几何视角

下面我们用几何方法来解释。

(50) 的分母为 1，意味着分母代表的几何图形是个单位圆，即，

$$x_1^2 + x_2^2 = 1 \tag{54}$$

(50) 分子对应二次函数：

$$f\left(x_1, x_2\right) = 1.5x_1^2 + x_1x_2 + 1.5x_2^2 \tag{55}$$

这个二次函数对应的等高线图如所示图 5 (a) 所示。$f(x_1, x_2)$ 等高线和单位圆相交的交点中找到 $f(x_1, x_2)$ 获取最大值和最小值点。最大特征值 λ_1 对应的特征向量 $\boldsymbol{v}_1$，$\boldsymbol{v}_1$ 这个方向上做一条直线，直线和单位圆交点 (x_1, x_2) 对应的就是瑞利商的最大值点；此时，瑞利商的最大值为 λ_1。

图 6 (a) 所示为 $f(x_1, x_2)$ 曲面，以及单位圆在曲面上的映射值对应的曲线。

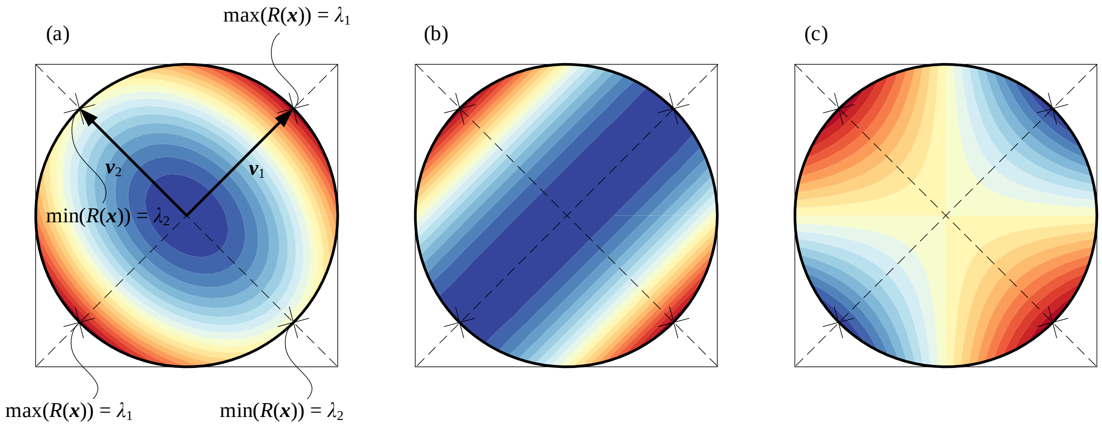


图 5. 平面上可视化 $f(x_1, x_2)$ 和单位圆

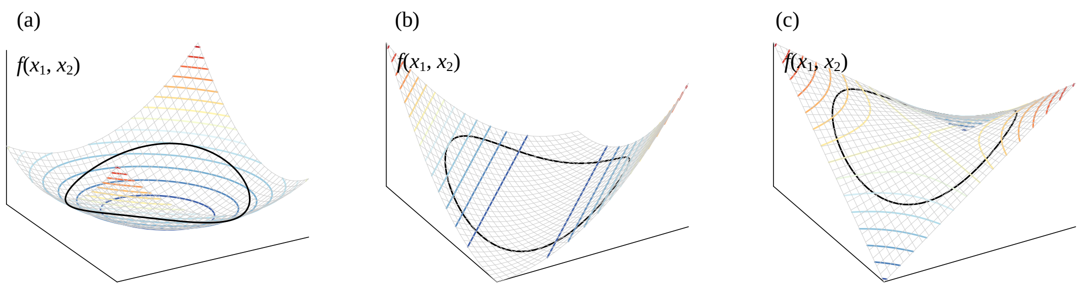


图 6. 三维空间中可视化 $f(x_1, x_2)$ 和单位圆

请读者格外注意，采用单位圆作为限制条件是为了简化瑞利商对应的数学问题，而且单位圆正好是单位向量终点的落点。

但是，实际上满足瑞利商最大值的点 (x_1, x_2) 有无数个，它们都位于特征向量 $\boldsymbol{v}_1$ 所在直线上。我们能从图 7 中一睹瑞利商 $R(x_1, x_2)$ 曲面形状真容，以及瑞利商最大值和最小值对应的 (x_1, x_2)。注意，瑞利商 $R(x_1, x_2)$ 在 $(0, 0)$ 没有定义。

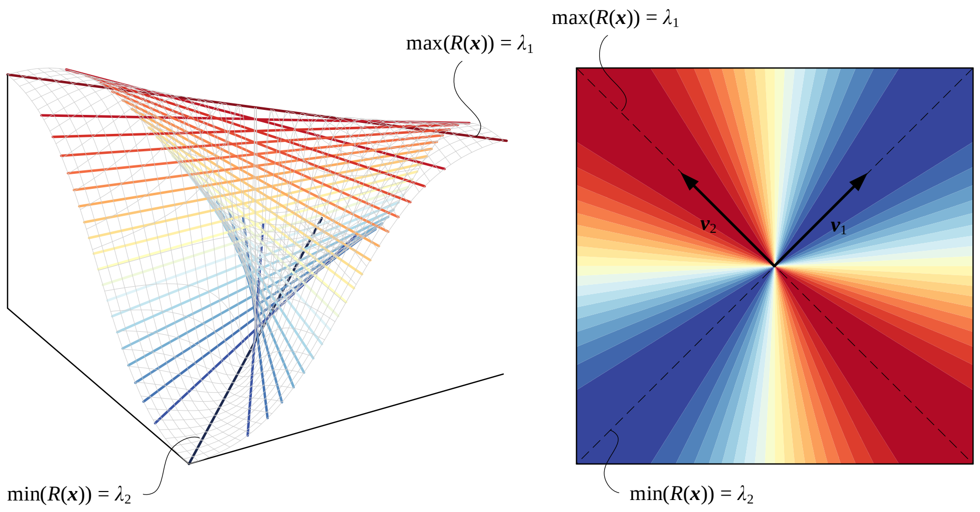


图 7. 三维空间中可视化瑞利商

再举个例子

给定矩阵 $\boldsymbol{A}$:

$$\boldsymbol{A} = \begin{bmatrix} 0.5 & -0.5 \\ -0.5 & 0.5 \end{bmatrix} \tag{56}$$

它的特征值分别为 $\lambda_1 = 1$，$\lambda_2 = 0$。$f(x_1, x_2)$ 等高线和曲面如图 5 (b) 和图 6 (b)所示。

图 5 (c) 等高线对应的矩阵 $\boldsymbol{A}$ 为：

$$\boldsymbol{A} = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} \tag{57}$$

它的特征值分别为 $\lambda_1 = 1$，$\lambda_2 = -1$。图 6 (c) 所示为 $f(x_1, x_2)$ 曲面的形状。

三维空间

以上探讨的三种情况都是以 2×2 矩阵为例。在三维空间中，$D = 3$ 这种情况，(41) 对应的是一个单位圆球体，将 $f(x_1, x_2, x_3)$ 三元函数的数值以等高线的形式映射到单位圆球体，得到图 8。

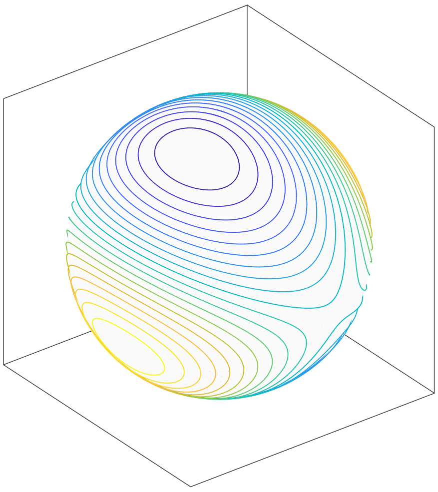


图 8. 三维单位球体表面瑞利商值等高线

Bk4_Ch14_03.py 绘制图 5 和图 6。

```
# Bk4_Ch14_03.py

import sympy
import numpy as np
import matplotlib.pyplot as plt
from numpy import linalg as L


def mesh_circ(c1, c2, r, num):

    theta = np.linspace(0, 2*np.pi, num)
    r     = np.linspace(0,r, num)
    theta,r = np.meshgrid(theta,r)
    xx1 = np.cos(theta)*r + c1
    xx2 = np.sin(theta)*r + c2

    return xx1, xx2


#define symbolic vars, function
x1,x2 = sympy.symbols('x1 x2')

A = np.array([[0.5, -0.5],
              [-0.5, 0.5]])

Lambda, V = L.eig(A)

x = np.array([[x1,x2]]).T

f_x = x.T@A@x
f_x = f_x[0][0]

f_x_fcn = sympy.lambdify([x1,x2],f_x)

xx1, xx2 = mesh_circ(0, 0, 1, 50)

ff_x = f_x_fcn(xx1,xx2)
```

```python
if Lambda[1] > 0:
    levels = np.linspace(0,Lambda[0],21)
else:
    levels = np.linspace(Lambda[1],Lambda[0],21)

t = np.linspace(0,np.pi*2,100)

# 2D visualization
fig, ax = plt.subplots()

ax.plot(np.cos(t), np.sin(t), color = 'k')

cs = plt.contourf(xx1, xx2, ff_x,
                  levels=levels, cmap = 'RdYlBu_r')
plt.show()
ax.set_aspect('equal')
ax.xaxis.set_ticks([])
ax.yaxis.set_ticks([])
ax.set_xlabel('$x_1$')
ax.set_ylabel('$x_2$')
ax.set_xlim(-1,1)
ax.set_ylim(-1,1)
clb = fig.colorbar(cs, ax=ax)
clb.set_ticks(levels)


#%% 3D surface of f(x1,x2)

x1_ = np.linspace(-1.2,1.2,31)
x2_ = np.linspace(-1.2,1.2,31)

xx1_fine, xx2_fine = np.meshgrid(x1_,x2_)

ff_x_fine = f_x_fcn(xx1_fine,xx2_fine)

f_circle = f_x_fcn(np.cos(t), np.sin(t))

# 3D visualization

fig, ax = plt.subplots()
ax = plt.axes(projection='3d')

ax.plot(np.cos(t), np.sin(t), f_circle, color = 'k')
# circle projected to f(x1,x2)

ax.plot_wireframe(xx1_fine,xx2_fine,ff_x_fine,
                  color = [0.8,0.8,0.8],
                  linewidth = 0.25)

ax.contour3D(xx1_fine,xx2_fine,ff_x_fine,15,
             cmap = 'RdYlBu_r')

ax.view_init(elev=30, azim=60)
ax.xaxis.set_ticks([])
ax.yaxis.set_ticks([])
ax.zaxis.set_ticks([])
ax.set_xlim(xx1_fine.min(),xx1_fine.max())
ax.set_ylim(xx2_fine.min(),xx2_fine.max())
plt.tight_layout()
ax.set_proj_type('ortho')
plt.show()
```

14.6 特征值分解中的复数现象

本书前文在对实数矩阵进行特征值分解时，我们偶尔发现特征值、特征向量存在虚数。这一节讨论这个现象。

举个例子

给定如下 2 × 2 实数矩阵 $\boldsymbol{A}$:

$$\boldsymbol{A} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \tag{58}$$

对 $\boldsymbol{A}$ 进行特征值分解，得到两个特征值分别为:

$$\lambda_1 = 1 + i, \quad \lambda_2 = 1 - i \tag{59}$$

共轭复数

这对共轭特征值出现的原因是，方阵 $\boldsymbol{A}$ 特征方程有一对复数解:

$$|\boldsymbol{A} - \lambda \boldsymbol{I}| = 0 \tag{60}$$

注意到两个特征值共轭，因此它们也常被称作**共轭特征值** (conjugate eigenvalues)。当矩阵系数是实数的时候，非实数的特征值会以共轭复数形式对出现。所谓**共轭复数** (complex conjugate)，是指两个实部相等，虚部互为相反数的复数。

λ_1 和 λ_2 对应的特征向量分别是:

$$\boldsymbol{v}_1 = \begin{bmatrix} i \\ 1 \end{bmatrix}, \quad \boldsymbol{v}_2 = \begin{bmatrix} -i \\ 1 \end{bmatrix} \tag{61}$$

我们发现这两个特征向量对应位置如果是复数的话，它们之间的关系也是共轭复数。因此，这样的特征向量，常被称作**共轭特征向量** (conjugate eigenvector)。

展开来说，本书前文讲述的向量矩阵等概念都是建立在实数 $\mathbb{R}^n$ 基础之上，我们可以把同样的数学工具推广到复数空间 $\mathbb{C}^n$ 上。

$\mathbb{C}^n$ 中的任意复向量 $\boldsymbol{x}$ 的共轭向量 $\bar{\boldsymbol{x}}$，也是 $\mathbb{C}^n$ 中的向量。$\bar{\boldsymbol{x}}$ 的分量是 $\boldsymbol{x}$ 对应分量的共轭复数。

比如，给定复数向量 $\boldsymbol{x}$ 和对应的共轭向量 $\bar{\boldsymbol{x}}$ 如下:

$$\boldsymbol{x} = \begin{bmatrix} 1+i \\ 3-2i \end{bmatrix}, \quad \bar{\boldsymbol{x}} = \begin{bmatrix} 1-i \\ 3+2i \end{bmatrix} \tag{62}$$

Re($\boldsymbol{x}$) 和 Im($\boldsymbol{x}$) 分别叫做复向量 $\boldsymbol{x}$ 的实部和虚部，它们分别由 $\boldsymbol{x}$ 向量各个分量的实部和虚部构成。

比如 (62) 中复数向量 $\boldsymbol{x}$ 对应的实部 Re($\boldsymbol{x}$) 和虚部 Im($\boldsymbol{x}$) 分别为：

$$\operatorname{Re}(\boldsymbol{x}) = \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \quad \operatorname{Im}(\boldsymbol{x}) = \begin{bmatrix} i \\ -2i \end{bmatrix} \tag{63}$$

读到这里很多读者可能已经不知所云。为了帮助大家理解，下面介绍一类有趣的 2×2 矩阵的特征值分解，以及它们对应的几何特征。

特征值分解

给定矩阵 $\boldsymbol{A}$ 如下：

$$\boldsymbol{A} = \begin{bmatrix} a & -b \\ b & a \end{bmatrix} \tag{64}$$

其中，a 和 b 均为实数，且不同时等于 0。

容易求得 $\boldsymbol{A}$ 的复数特征值为一对共轭复数：

$$\lambda = a \pm bi \tag{65}$$

两者的关系如图 9 所示。

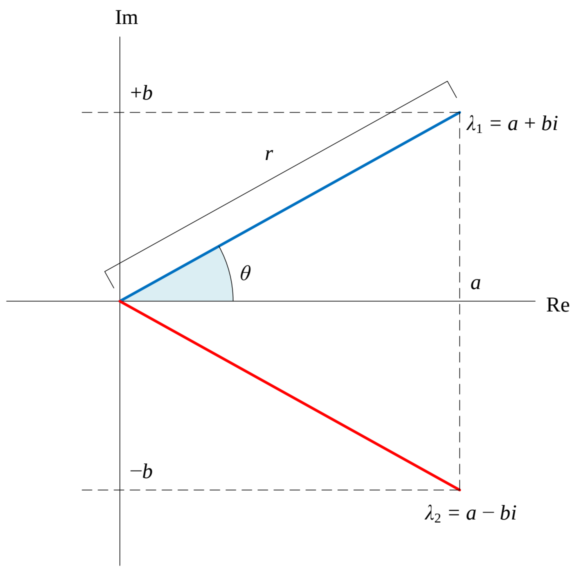


图 9. 一对共轭特征值

两个共轭特征值的模相等，令 r 复数特征值的模：

$$r = |\lambda| = \sqrt{a^2 + b^2} = \sqrt{|\boldsymbol{A}|} \tag{66}$$

容易发现，r 是矩阵 $\boldsymbol{A}$ 行列式值的平方根。这样 $\boldsymbol{A}$ 可以写成：

$$\boldsymbol{A} = \sqrt{a^2 + b^2}\begin{bmatrix} \frac{a}{\sqrt{a^2+b^2}} & \frac{-b}{\sqrt{a^2+b^2}} \\ \frac{b}{\sqrt{a^2+b^2}} & \frac{a}{\sqrt{a^2+b^2}} \end{bmatrix} = r\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}}_{\boldsymbol{R}} \underbrace{\begin{bmatrix} r & 0 \\ 0 & r \end{bmatrix}}_{\boldsymbol{S}} \tag{67}$$

图 9 所示复平面上，θ 为水平轴正方向和 (0, 0) 到 (a, b) 射线的夹角，θ 也称作为复数 $\lambda_1 = a + bi$ 的辐角。

几何视角

有了上述分析，矩阵 $\boldsymbol{A}$ 的几何变换就变得很清楚，$\boldsymbol{A}$ 是缩放 ($\boldsymbol{S}$) 和旋转 ($\boldsymbol{R}$) 的复合。这就解释了上一章旋转矩阵进行特征值分解时，得到的两个特征值为共轭复数。

给平面上某个位置的 $\boldsymbol{x}_0$，用矩阵 $\boldsymbol{A}$ 不断作用在 $\boldsymbol{x}_0$ 上：

$$\boldsymbol{x}_n = \boldsymbol{A}^n \boldsymbol{x}_0 \tag{68}$$

如图 10 (a) 所示，当缩放系数 $r = 1.2 > 1$，我们可以看到，随着 n 增大，向量 $\boldsymbol{x}_n$ 不断旋转向外。

如图 10 (b) 所示，当缩放系数 $r = 0.8 < 1$，我们可以看到，随着 n 增大，向量 $\boldsymbol{x}_n$ 不断旋转向内。

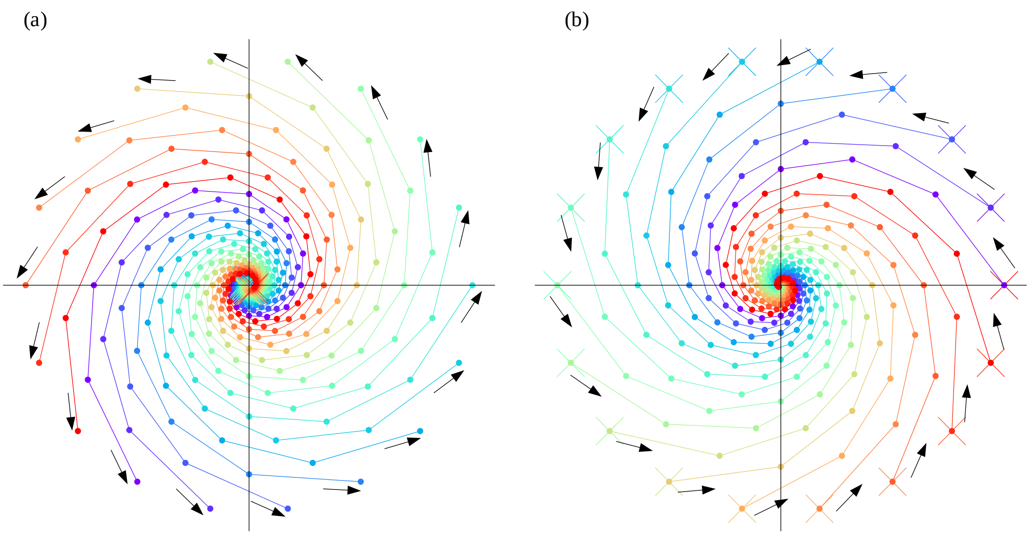


图 10. 在矩阵 $\boldsymbol{A}$ 几何变换重复下，向量的 $\boldsymbol{x}$ 位置变化

Bk4_Ch14_04.py 绘制图 10。

```
# Bk4_Ch14_04.py

import numpy as np
import matplotlib.pyplot as plt

theta = np.deg2rad(30)

r = 0.8 # 1.2, scaling factor

R = np.array([[np.cos(theta), -np.sin(theta)],
              [np.sin(theta),  np.cos(theta)]])

S = np.array([[r, 0],
              [0, r]])

A = R@S

# A = np.array([[1, -1],
#                [1, 1]])

Lamb, V = np.linalg.eig(A)

theta_array = np.arange(0,np.pi*2,np.pi*2/18)


colors = plt.cm.rainbow(np.linspace(0,1,len(theta_array)))


fig, ax = plt.subplots()

for j, theat_i in enumerate(theta_array):

    # initial point
    x = np.array([[5*np.cos(theat_i)],
                  [5*np.sin(theat_i)]])

    plt.plot(x[0],x[1],
             marker = 'x',color = colors_j,
             markersize = 15)
    # plot the initial point

    x_array = x

    for i in np.arange(20):

        x = A@x
        x_array = np.column_stack((x_array,x))


    colors_j = colors[j,:]
    plt.plot(x_array[0,:],x_array[1,:],
             marker = '.',color = colors_j)


plt.axis('scaled')

ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.spines['bottom'].set_visible(False)
ax.spines['left'].set_visible(False)
ax.axvline(x=0,color = 'k')
ax.axhline(y=0,color = 'k')
```

本章主要着墨在特征值分解的应用，比如方阵开方、矩阵指数、斐波那契数列、马尔科夫过程平衡状态等等。

本章特别值得注意的一个知识点是瑞利商，数据科学和机器学习很多算法中都离不开瑞利商。希望大家能从几何视角理解瑞利商的最值。本书还将在拉格朗日乘子法中继续探讨瑞利商。

本章最后以我们在对实数矩阵分解中遇到的复数现象为例，介绍了共轭特征值和共轭特征向量。注意，复数矩阵自有一套体系，比如实数矩阵中有转置，而复数矩阵的转置叫做埃尔米特转置 (Hermitian transpose)。复数矩阵相关内容不在本书范围内，感兴趣的读者可以自行学习。

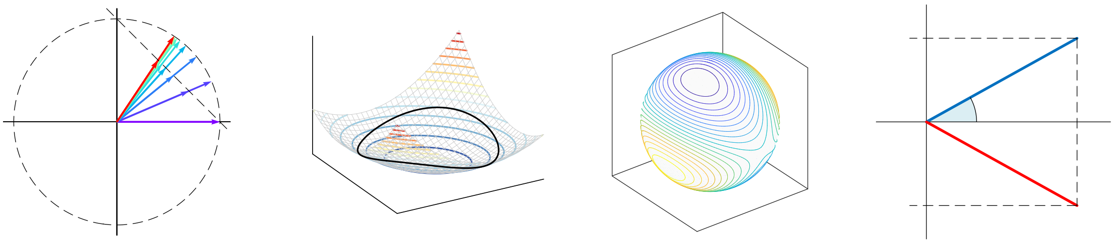


图 11. 总结本章重要内容的四副图