25 鸡兔同笼 3

Probability Meets Linear Algebra

鸡兔互变之马尔科夫奇妙夜

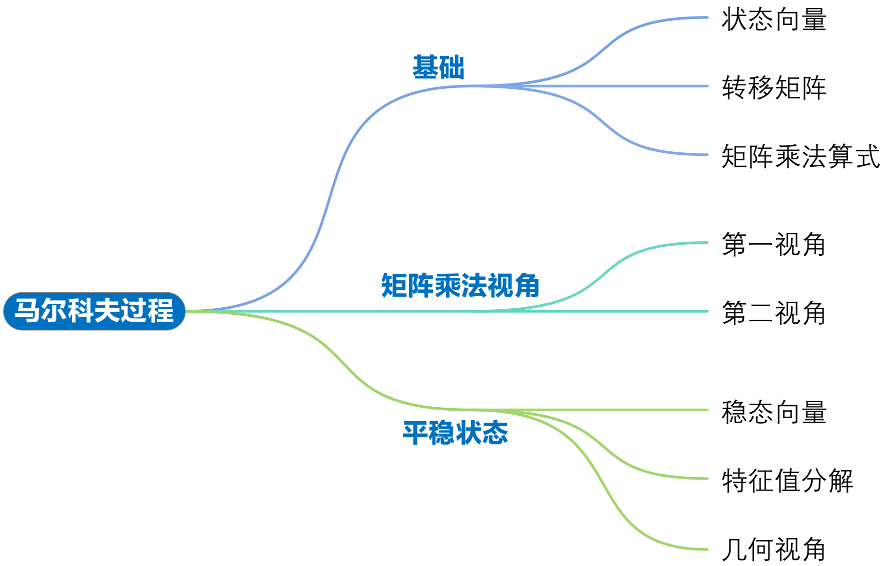
我们必须知道，我们终将知道。

Wir müssen wissen. Wir werden wissen.

We must know. We shall know.

—— 大卫·希尔伯特 (David Hilbert) | 德国数学家 | 1862 ~ 1943

- ◄ `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
- ◄ `numpy.linalg.eig()` 特征值分解
- ◄ `numpy.linalg.inv()` 矩阵求逆
- ◄ `numpy.matrix()` 构造二维矩阵
- ◄ `numpy.meshgrid()` 产生网格化数据
- ◄ `numpy.vstack()` 垂直堆叠数组
- ◄ `seaborn.heatmap()` 绘制热图


马尔科夫过程
基础
状态向量
转移矩阵
矩阵乘法算式
矩阵乘法视角
第一视角
第二视角
平稳状态
稳态向量
特征值分解
几何视角

25.1 鸡兔互变奇妙夜

怪哉，怪哉！

接连数月，村民发现一件奇事——夜深人静时，同笼鸡兔竟然互变！一些小兔变成小鸡，而一些小鸡变成小兔。

村民奔走相告，大家都惊呼，“我们都疯了！”

而一众村民中，农夫则显得处变不惊。在农夫眼里，村里发生的鸡兔互变，像极了老子说的“祸兮，福之所倚；福兮，祸之所伏。”

农夫对村民说，“大家不要怕，恐惧都是来自于未知。我们必须知道，我们终将知道！福祸相生，是福不是祸，是祸躲不过。”

面对这个鸡兔互变的怪相，农夫决定用线性代数这个利器探究一番。

鸡兔互变过程图

农夫先是连续几日统计村里的鸡兔数量，他有个意想不到的发现——每晚有 30%的小鸡变成小兔，其他小鸡不变；与此同时，每晚有 20%小兔变成小鸡，其余小兔不变。变化前后鸡兔总数不变。

他先画了图 1 这幅图，用来描述鸡兔互变的比例。这个比例也就是概率值，即发生变化的可能性。

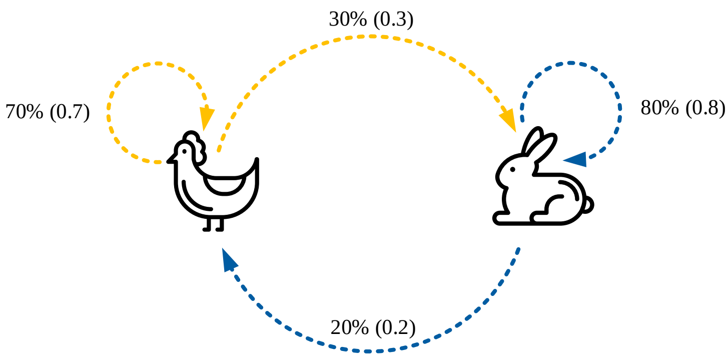


图 1. 鸡兔互变的比例

矩阵乘法

农夫想试试用矩阵乘法来描述这一过程。

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，$\boldsymbol{\pi}$ 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。鸡兔互变的比例写成方阵 $\boldsymbol{T}$，这样 $k \to k+1$ 变化过程可以写成。

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

农夫翻阅线性代数典籍时发现 $\boldsymbol{T}$ 和 $\boldsymbol{\pi}$ 都有自己专门的名称：$\boldsymbol{T}$ 叫转移矩阵 (transition matrix)；列向量 $\boldsymbol{\pi}$ 叫做状态向量 (state vector)。而整个鸡兔互变的过程也有自己的名称——马尔可夫过程 (Markov process)。

转移矩阵

鸡兔互变中，转移矩阵 $\boldsymbol{T}$ 为。

$$\boldsymbol{T} = \begin{bmatrix} 0.7 & 0.2 \\ 0.3 & 0.8 \end{bmatrix} \tag{2}$$

图 2 所示为转移矩阵 $\boldsymbol{T}$ 每个元素的具体含义。

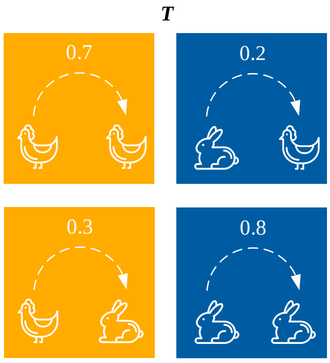


图 2. 转移矩阵 $\boldsymbol{T}$

图 3 所示为用矩阵运算描述 $k \to k+1$ 鸡兔互变过程。

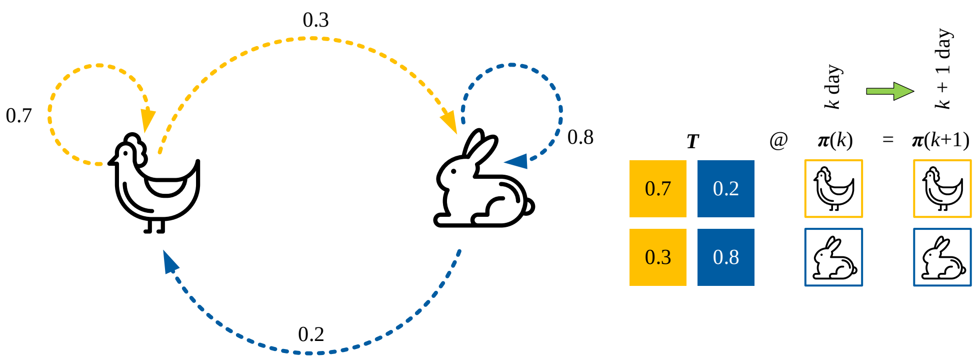


图 3. 用矩阵运算描述鸡兔互变

农夫注意到 $\boldsymbol{T}$ 矩阵的每一列概率值相加为 1。也就是，这个 2×2 的方阵 $\boldsymbol{T}$ 还可以写成。

$$T = \begin{bmatrix} p & q \\ 1-p & 1-q \end{bmatrix} \tag{3}$$

其中，$p = 0.7$，$q = 0.2$。

代入具体数值

农夫假设，第 k 天鸡兔的比例为 60%和 40%，$\boldsymbol{\pi}(k)$ 为：

$$\boldsymbol{\pi}(k) = \begin{bmatrix} 0.6 \\ 0.4 \end{bmatrix} \tag{4}$$

第 $k + 1$ 天，鸡兔比例为：

$$k \to k+1: \quad \boldsymbol{T\pi}(k) = \underbrace{\begin{bmatrix} 0.7 & 0.2 \\ 0.3 & 0.8 \end{bmatrix}}_{\boldsymbol{T}} \underbrace{\begin{bmatrix} 0.6 \\ 0.4 \end{bmatrix}}_{\boldsymbol{\pi}(k)} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} = \boldsymbol{\pi}(k+1) \tag{5}$$

农夫想到这一计算可以用热图表达，于是他画了图 4。

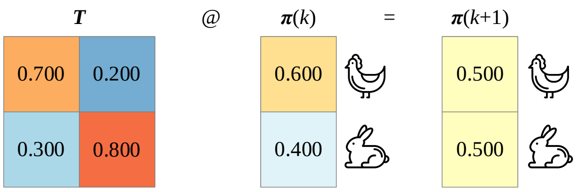


图 4. 第 k 天 → 第 $k + 1$ 天，状态转换运算热图

而第 $k + 2$ 天状态向量 $\boldsymbol{\pi}(k + 2)$ 和第 $k + 1$ 天状态向量 $\boldsymbol{\pi}(k + 1)$ 关系为：

$$k+1 \to k+2: \quad \boldsymbol{T\pi}(k+1) = \boldsymbol{\pi}(k+2) \tag{6}$$

联立 (5) 和 (6)，得到第 $k + 2$ 天状态向量 $\boldsymbol{\pi}(k + 2)$ 和第 k 天状态向量 $\boldsymbol{\pi}(k)$ 关系。

$$k \to k+2: \quad \boldsymbol{T}^2\boldsymbol{\pi}(k) = \boldsymbol{\pi}(k+2) \tag{7}$$

图 5 所示为，第 k 天 → 第 $k + 2$ 天，状态转换运算热图。

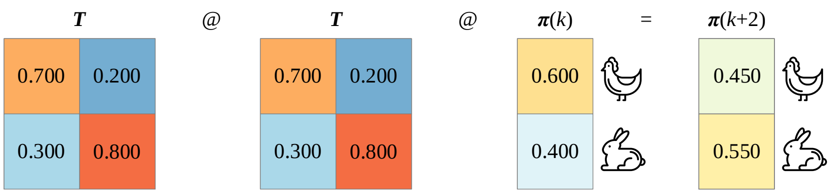


图 5. 第 k 天 → 第 $k+2$ 天，状态转换运算热图

另一种形式

农夫在查找参考书时发现，也有很多典籍用行向量表达状态向量，即对等式 (1) 左右转置：

$$\boldsymbol{\pi}(k)^{\mathrm{T}}\boldsymbol{T}^{\mathrm{T}}=\boldsymbol{\pi}(k+1)^{\mathrm{T}} \tag{8}$$

这样，(5) 可以写成：

$$\boldsymbol{\pi}(k+1)^{\mathrm{T}}=\begin{bmatrix}0.6 & 0.4\end{bmatrix}\begin{bmatrix}0.7 & 0.3\\ 0.2 & 0.8\end{bmatrix}=\begin{bmatrix}0.5 & 0.5\end{bmatrix} \tag{9}$$

这种情况，转移矩阵的每一行概率相加为 1。对应的矩阵运算热图为图 6。

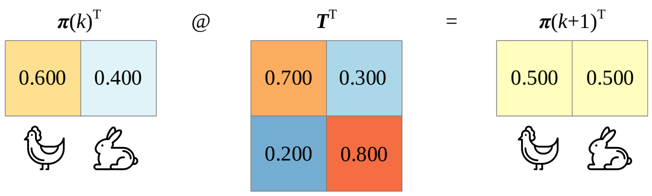


图 6. 第 k 天 → 第 $k+1$ 天，状态转换运算热图，注意状态向量为行向量

以下代码计算状态向量转化，并绘制图 4 和图 5 两幅热图。

```
# Bk3_Ch25_1

import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

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

# pi(i), state vector
pi_i = np.matrix([[0.6],
                  [0.4]])

#%% pi(i) >>> pi(i + 1): pi(i + 1) = T@pi(i)

pi_i_1 = T@pi_i

fig, axes = plt.subplots(1, 5, figsize=(12, 3))

all_max = 1
all_min = 0

plt.sca(axes[0])
ax = sns.heatmap(T,cmap='RdYlBu_r',vmax = all_max,vmin = all_min,
                 cbar_kws={"orientation": "horizontal"},
                 annot = True,fmt=".3f")
ax.set_aspect("equal")
```

```
plt.title('$T$')
plt.yticks(rotation=0)

plt.sca(axes[1])
plt.title('$@$')
plt.axis('off')

plt.sca(axes[2])
ax = sns.heatmap(pi_i,cmap='RdYlBu_r',vmax = all_max,vmin = all_min,
                 cbar_kws={"orientation": "horizontal"},
                 annot = True,fmt=".3f")
ax.set_aspect("equal")
plt.title('$\pi_{i}$')
plt.yticks(rotation=0)

plt.sca(axes[3])
plt.title('$=$')
plt.axis('off')

plt.sca(axes[4])
ax = sns.heatmap(pi_i_1,cmap='RdYlBu_r',vmax = all_max,vmin = all_min,
                 cbar_kws={"orientation": "horizontal"},
                 annot = True,fmt=".3f")
ax.set_aspect("equal")
plt.title('$\pi_{i+1}$')
plt.yticks(rotation=0)

#%% pi(i) >>> pi(i + 2): pi(i + 2) = T^2@pi(i)

pi_i_2 = T@T@pi_i

fig, axes = plt.subplots(1, 7, figsize=(12, 3))

plt.sca(axes[0])
ax = sns.heatmap(T,cmap='RdYlBu_r',vmax = all_max,vmin = all_min,
                 cbar_kws={"orientation": "horizontal"},
                 annot = True,fmt=".3f")
ax.set_aspect("equal")
plt.title('$T$')
plt.yticks(rotation=0)

plt.sca(axes[1])
plt.title('$@$')
plt.axis('off')

plt.sca(axes[2])
ax = sns.heatmap(T,cmap='RdYlBu_r',vmax = all_max,vmin = all_min,
                 cbar_kws={"orientation": "horizontal"},
                 annot = True,fmt=".3f")
ax.set_aspect("equal")
plt.title('$T$')
plt.yticks(rotation=0)

plt.sca(axes[3])
plt.title('$@$')
plt.axis('off')

plt.sca(axes[4])
ax = sns.heatmap(pi_i,cmap='RdYlBu_r',vmax = all_max,vmin = all_min,
                 cbar_kws={"orientation": "horizontal"},
                 annot = True,fmt=".3f")
ax.set_aspect("equal")
plt.title('$\pi_{i}$')
plt.yticks(rotation=0)

plt.sca(axes[5])
plt.title('$=$')
plt.axis('off')

plt.sca(axes[6])
ax = sns.heatmap(pi_i_2,cmap='RdYlBu_r',vmax = all_max,vmin = all_min,
```

```
                    cbar_kws={"orientation": "horizontal"},
                    annot = True,fmt=".3f")
ax.set_aspect("equal")
plt.title('$\pi_{i+2}$')
plt.yticks(rotation=0)
```

25.2 第一视角：“鸡/兔→鸡”和“鸡/兔→兔”

农夫想到自己学习矩阵乘法时，书上讲过矩阵乘法有两个主要视角。他想先用矩阵乘法第一视角来分析 (1) 矩阵运算式。

他把 $\boldsymbol{T}$ 写成两个行向量 $\boldsymbol{t}^{(1)}$ 和 $\boldsymbol{t}^{(1)}$ 上下叠加，代入 (1) 得到。

$$\underbrace{\begin{bmatrix} \boldsymbol{t}^{(1)} \\ \boldsymbol{t}^{(2)} \end{bmatrix}}_{\boldsymbol{T}} \boldsymbol{\pi}(k) = \begin{bmatrix} \boldsymbol{t}^{(1)}\boldsymbol{\pi}(k) \\ \boldsymbol{t}^{(2)}\boldsymbol{\pi}(k) \end{bmatrix} = \underbrace{\begin{bmatrix} \pi_1(k+1) \\ \pi_2(k+1) \end{bmatrix}}_{\boldsymbol{\pi}(k+1)} \tag{10}$$

鸡/兔→鸡

农夫发现只看 (10) 第一行运算的话，它代表的转化是“鸡/兔 → 鸡”，如图 7 所示：

$$\begin{bmatrix} \boldsymbol{t}^{(1)} \\ \end{bmatrix} \boldsymbol{\pi}(k) = \begin{bmatrix} \boldsymbol{t}^{(1)}\boldsymbol{\pi}(k) \\ \end{bmatrix} = \begin{bmatrix} \pi_1(k+1) \\ \end{bmatrix} \tag{11}$$

也就是说，上式代表第 k 天的鸡、兔，在第 $k+1$ 天变为鸡。

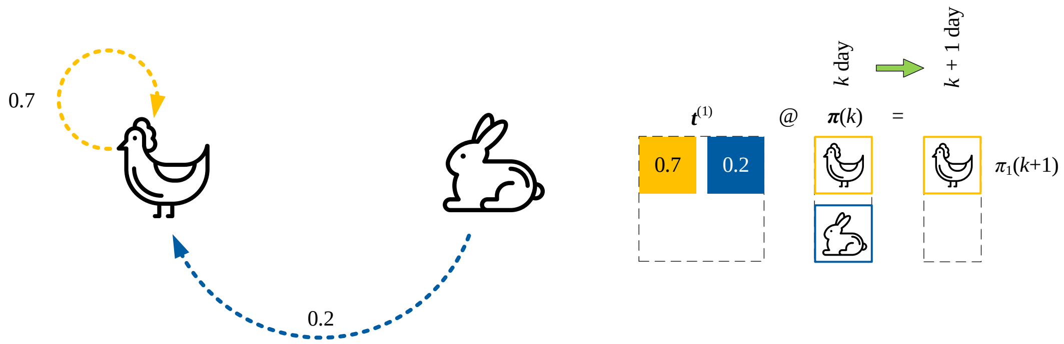


图 7. 鸡/兔→鸡

代入具体值，得到：

$$\begin{bmatrix} 0.7 & 0.2 \\ & \end{bmatrix} @ \underbrace{\begin{bmatrix} 0.6 \\ 0.4 \end{bmatrix}}_{\boldsymbol{\pi}(k)} = \begin{bmatrix} 0.5 \\ \end{bmatrix} \tag{12}$$

第 k 天的鸡兔的比例分别为 60%和 40%，到了 $k+1$ 天，鸡的比例为 50%。图 8 所示为上述运算热图。

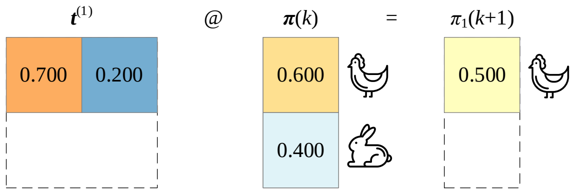


图 8. 第 k 天 → 第 $k+1$ 天，鸡/兔→鸡

鸡/兔→兔

图 9 所示为 (10) 第二行运算，它代表“鸡/兔 → 兔”。也就是说，第 k 天的鸡、兔，第 $k+1$ 天变为兔。

$$\begin{bmatrix} \\ t^{(2)} \end{bmatrix} \pi(k) = \begin{bmatrix} \\ t^{(2)} \pi(k) \end{bmatrix} = \begin{bmatrix} \\ \pi_2(k+1) \end{bmatrix} \tag{13}$$

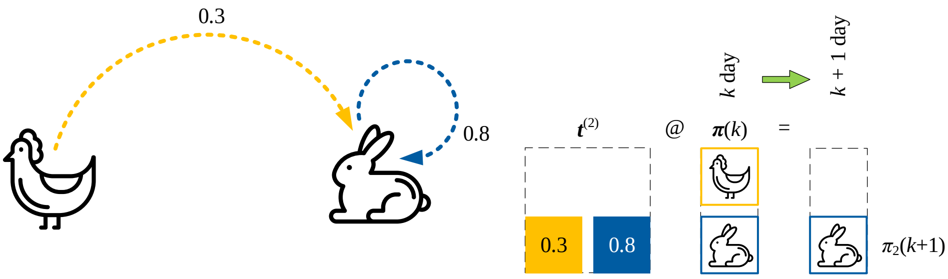


图 9. 鸡/兔 → 兔

图 10 所示为第 k 天的鸡兔的比例分别为 60%和 40%，到了 $k+1$ 天，兔的比例也为 50%。

$$\begin{bmatrix} \\ 0.3 & 0.8 \end{bmatrix} @ \underbrace{\begin{bmatrix} 0.6 \\ 0.4 \end{bmatrix}}_{\pi(k)} = \begin{bmatrix} \\ 0.5 \end{bmatrix} \tag{14}$$

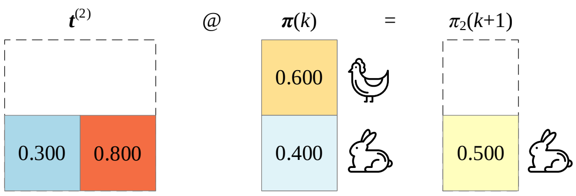


图 10. 第 k 天 → 第 $k+1$ 天，鸡/兔→鸡

这就是利用矩阵乘法第一视角来分析状态转化运算。

25.3 第二视角："鸡→鸡/兔"和"兔→鸡/兔"

农夫继续用矩阵乘法第二视角分析 (1) 矩阵运算式。

他将转移矩阵 $\boldsymbol{T}$ 写成左右排列列向量 $\boldsymbol{t}_1$ 和 $\boldsymbol{t}_2$，代入 (1) 展开得到。

$$\underbrace{\begin{bmatrix} \boldsymbol{t}_1 & \boldsymbol{t}_2 \end{bmatrix}}_{\boldsymbol{T}} \underbrace{\begin{bmatrix} \pi_1(k) \\ \pi_2(k) \end{bmatrix}}_{\boldsymbol{\pi}(k)} = \pi_1(k)\boldsymbol{t}_1 + \pi_2(k)\boldsymbol{t}_2 = \underbrace{\begin{bmatrix} \pi_1(k+1) \\ \pi_2(k+1) \end{bmatrix}}_{\boldsymbol{\pi}(k+1)} \tag{15}$$

其中，π_1 代表鸡的比例，π_2 代表鸡的比例。

矩阵乘法第二视角将矩阵乘法 $\boldsymbol{T\pi}(k) = \boldsymbol{\pi}(k+1)$ 转化为矩阵加法 $\pi_1(k)\boldsymbol{t}_1 + \pi_2(k)\boldsymbol{t}_2$。农夫下面考虑分别分析，$\pi_1(k)\boldsymbol{t}_1$ 和 $\pi_2(k)\boldsymbol{t}_2$ 代表的具体含义。

(15) 这个式子让农夫看着头大，他决定代入具体鸡兔数值。

鸡→鸡/兔

假设第 k 天，鸡兔的比例仍为 60%、40%。

$$\boldsymbol{\pi}(k) = \begin{bmatrix} \pi_1(k) \\ \pi_2(k) \end{bmatrix} = \begin{bmatrix} 0.6 \\ 0.4 \end{bmatrix} \tag{16}$$

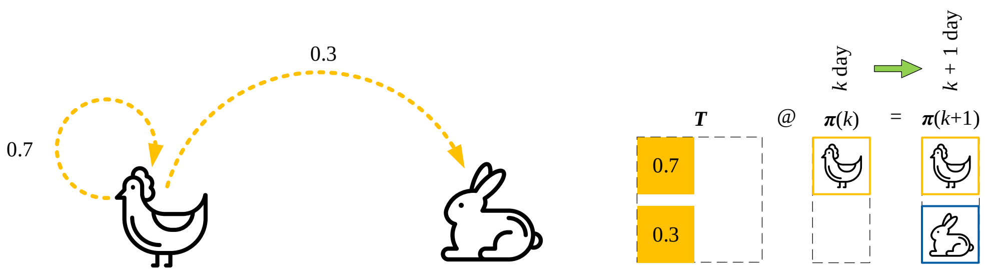


图 11. 鸡→鸡/兔

如图 11 所示，$\pi_1(k)\boldsymbol{t}_1$ 代表"鸡 → 鸡/兔"；第 k 天，鸡的比例为 0.6，这些鸡在第 $k+1$ 天变成占总体比例 0.42 的鸡和 0.18 的兔。

$$\pi_1(k)\boldsymbol{t}_1 = 0.6 \times \begin{bmatrix} 0.7 \\ 0.3 \end{bmatrix} = \begin{bmatrix} 0.42 \\ 0.18 \end{bmatrix} \tag{17}$$

图 12 所示为 (17) 运算热图。

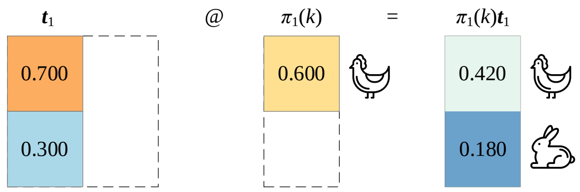


图 12. 第 k 天 → 第 $k+1$ 天，鸡→鸡/兔

兔→鸡/兔

如图 13 所示，$\pi_2(k)\boldsymbol{t}_2$ 代表“兔 → 鸡/兔”。第 k 天，兔的比例为 0.4，这些兔在第 $k+1$ 天变成占总体比例 0.08 的鸡和 0.32 的兔：

$$\pi_2(k)\boldsymbol{t}_2 = 0.4 \times \begin{bmatrix} 0.2 \\ 0.8 \end{bmatrix} = \begin{bmatrix} 0.08 \\ 0.32 \end{bmatrix} \tag{18}$$

图 14 热图对应上述运算。

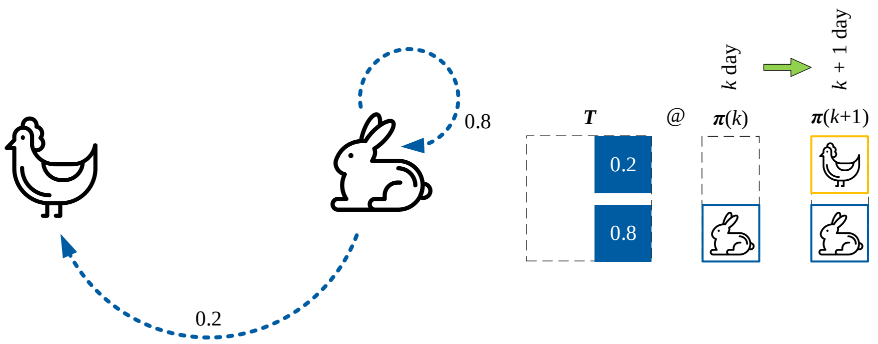


图 13. 兔→鸡/兔

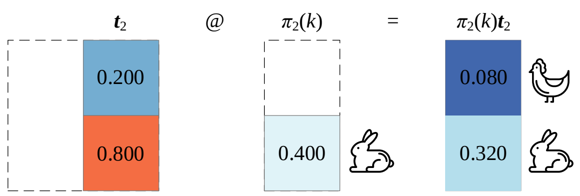


图 14. 第 k 天 → 第 $k+1$ 天，兔→鸡/兔

如图 15 热图所示，将 (17) 和 (18) 相加，得到第 $k+1$ 天状态向量 $\boldsymbol{\pi}(i+1)$。

$$\boldsymbol{\pi}(k+1)=\begin{bmatrix}0.42\\0.18\end{bmatrix}+\begin{bmatrix}0.08\\0.32\end{bmatrix}=\begin{bmatrix}0.5\\0.5\end{bmatrix}\tag{19}$$

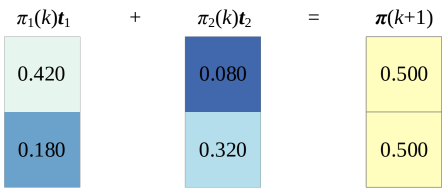


图 15. 第 k 天 → 第 $k+1$ 天，鸡/兔→鸡/兔

这就是利用矩阵乘法第二视角来分析状态转化运算。

25.4 连续几夜鸡兔转换

农夫把自己所学所想和村民分享后，大家都觉得线性代数有趣，认为这个分析有道理。大家纷纷加入农夫成立的“线代探秘小组”，学线代，用线代，继续探究鸡兔互变这个疑难杂症。

有细心村民发现，虽然连日来各家鸡兔互变没有停止，但是全村的鸡兔比例似乎达到了某种平衡。真是丈二和尚摸不着头脑!

农夫想用线性代数方法来看看连续几晚鸡兔互变有何有趣特征。

第 0 天，为初始状态，记做 $\boldsymbol{\pi}(0)$。

第 1 天，状态向量 $\boldsymbol{\pi}(1)$ 为:

$$0\to1:\quad \boldsymbol{T}\boldsymbol{\pi}(0)=\boldsymbol{\pi}(1)\tag{20}$$

第 2 天，状态向量 $\boldsymbol{\pi}(2)$ 和 $\boldsymbol{\pi}(0)$ 关系为:

$$0\to2:\quad \boldsymbol{T}\boldsymbol{\pi}(1)=\boldsymbol{T}^2\boldsymbol{\pi}(0)=\boldsymbol{\pi}(2)\tag{21}$$

第 3 天，状态向量 $\boldsymbol{\pi}(3)$ 和 $\boldsymbol{\pi}(0)$ 关系为。

$$0\to3:\quad \boldsymbol{T}\boldsymbol{\pi}(2)=\boldsymbol{T}^3\boldsymbol{\pi}(0)=\boldsymbol{\pi}(3)\tag{22}$$

这样 $0\to k+1$ 变化过程可以写成:

$$0\to k:\quad \boldsymbol{T}^k\boldsymbol{\pi}(0)=\boldsymbol{\pi}(k)\tag{23}$$

12 夜

农夫想算算连续 12 夜，在不同鸡兔初始比例 $\boldsymbol{\pi}(0)$ 条件下，鸡兔达到平衡时比例特点。

图 16 所示的五种情况为鸡的初始比例更高，经过连续 12 夜的变化，农夫发现鸡兔的比例都达到了 40%、60%，也就是 4:6。

这个结果让农夫和“线代探秘小组”组员都眼前一亮！

而图 17 对应的一种情况是，鸡兔的初始比例相同，都是 50%；12 夜之后，鸡兔比例还是 40%、60%。

图 18 所示的五种情况是，初始状态 $\boldsymbol{\pi}(0)$ 时，兔的比例更高。有趣的是，12 夜之后，鸡兔比例最终还是达到 40%、60%。

农夫觉得可以初步得出结论，在给定的转移矩阵 $\boldsymbol{T}$ 前提下，不管鸡兔初始比例 $\boldsymbol{\pi}(0)$ 如何，结果都达到了一定的平衡，也就是：

$$\boldsymbol{T\pi} = \boldsymbol{\pi} \tag{24}$$

$\pi(0)$	$\pi(1)$	$\pi(2)$	$\pi(3)$	$\pi(4)$	$\pi(5)$	$\pi(6)$	$\pi(7)$	$\pi(8)$	$\pi(9)$	$\pi(10)$	$\pi(11)$	$\pi(12)$
1.000	0.700	0.550	0.475	0.437	0.419	0.409	0.405	0.402	0.401	0.401	0.400	0.400
0.000	0.300	0.450	0.525	0.563	0.581	0.591	0.595	0.598	0.599	0.599	0.600	0.600

$\pi(0)$	$\pi(1)$	$\pi(2)$	$\pi(3)$	$\pi(4)$	$\pi(5)$	$\pi(6)$	$\pi(7)$	$\pi(8)$	$\pi(9)$	$\pi(10)$	$\pi(11)$	$\pi(12)$
0.900	0.650	0.525	0.462	0.431	0.416	0.408	0.404	0.402	0.401	0.400	0.400	0.400
0.100	0.350	0.475	0.538	0.569	0.584	0.592	0.596	0.598	0.599	0.600	0.600	0.600

$\pi(0)$	$\pi(1)$	$\pi(2)$	$\pi(3)$	$\pi(4)$	$\pi(5)$	$\pi(6)$	$\pi(7)$	$\pi(8)$	$\pi(9)$	$\pi(10)$	$\pi(11)$	$\pi(12)$
0.800	0.600	0.500	0.450	0.425	0.412	0.406	0.403	0.402	0.401	0.400	0.400	0.400
0.200	0.400	0.500	0.550	0.575	0.588	0.594	0.597	0.598	0.599	0.600	0.600	0.600

$\pi(0)$	$\pi(1)$	$\pi(2)$	$\pi(3)$	$\pi(4)$	$\pi(5)$	$\pi(6)$	$\pi(7)$	$\pi(8)$	$\pi(9)$	$\pi(10)$	$\pi(11)$	$\pi(12)$
0.700	0.550	0.475	0.438	0.419	0.409	0.405	0.402	0.401	0.401	0.400	0.400	0.400
0.300	0.450	0.525	0.562	0.581	0.591	0.595	0.598	0.599	0.599	0.600	0.600	0.600

$\pi(0)$	$\pi(1)$	$\pi(2)$	$\pi(3)$	$\pi(4)$	$\pi(5)$	$\pi(6)$	$\pi(7)$	$\pi(8)$	$\pi(9)$	$\pi(10)$	$\pi(11)$	$\pi(12)$
0.600	0.500	0.450	0.425	0.412	0.406	0.403	0.402	0.401	0.400	0.400	0.400	0.400
0.400	0.500	0.550	0.575	0.588	0.594	0.597	0.598	0.599	0.600	0.600	0.600	0.600

图 16. 连续 12 夜鸡兔互变比例，鸡的初始比例更高

$\pi(0)$	$\pi(1)$	$\pi(2)$	$\pi(3)$	$\pi(4)$	$\pi(5)$	$\pi(6)$	$\pi(7)$	$\pi(8)$	$\pi(9)$	$\pi(10)$	$\pi(11)$	$\pi(12)$
0.500	0.450	0.425	0.412	0.406	0.403	0.402	0.401	0.400	0.400	0.400	0.400	0.400
0.500	0.550	0.575	0.588	0.594	0.597	0.598	0.599	0.600	0.600	0.600	0.600	0.600

图 17. 连续 12 夜鸡兔互变比例，鸡和兔的初始比例一样高

$\pi(0)$	$\pi(1)$	$\pi(2)$	$\pi(3)$	$\pi(4)$	$\pi(5)$	$\pi(6)$	$\pi(7)$	$\pi(8)$	$\pi(9)$	$\pi(10)$	$\pi(11)$	$\pi(12)$
0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400
0.600	0.600	0.600	0.600	0.600	0.600	0.600	0.600	0.600	0.600	0.600	0.600	0.600

$\pi(0)$	$\pi(1)$	$\pi(2)$	$\pi(3)$	$\pi(4)$	$\pi(5)$	$\pi(6)$	$\pi(7)$	$\pi(8)$	$\pi(9)$	$\pi(10)$	$\pi(11)$	$\pi(12)$
0.300	0.350	0.375	0.387	0.394	0.397	0.398	0.399	0.400	0.400	0.400	0.400	0.400
0.700	0.650	0.625	0.613	0.606	0.603	0.602	0.601	0.600	0.600	0.600	0.600	0.600

$\pi(0)$	$\pi(1)$	$\pi(2)$	$\pi(3)$	$\pi(4)$	$\pi(5)$	$\pi(6)$	$\pi(7)$	$\pi(8)$	$\pi(9)$	$\pi(10)$	$\pi(11)$	$\pi(12)$
0.200	0.300	0.350	0.375	0.387	0.394	0.397	0.398	0.399	0.400	0.400	0.400	0.400
0.800	0.700	0.650	0.625	0.613	0.606	0.603	0.602	0.601	0.600	0.600	0.600	0.600

$\pi(0)$	$\pi(1)$	$\pi(2)$	$\pi(3)$	$\pi(4)$	$\pi(5)$	$\pi(6)$	$\pi(7)$	$\pi(8)$	$\pi(9)$	$\pi(10)$	$\pi(11)$	$\pi(12)$
0.100	0.250	0.325	0.362	0.381	0.391	0.395	0.398	0.399	0.399	0.400	0.400	0.400
0.900	0.750	0.675	0.638	0.619	0.609	0.605	0.602	0.601	0.601	0.600	0.600	0.600

$\pi(0)$	$\pi(1)$	$\pi(2)$	$\pi(3)$	$\pi(4)$	$\pi(5)$	$\pi(6)$	$\pi(7)$	$\pi(8)$	$\pi(9)$	$\pi(10)$	$\pi(11)$	$\pi(12)$
0.000	0.200	0.300	0.350	0.375	0.388	0.394	0.397	0.398	0.399	0.400	0.400	0.400
1.000	0.800	0.700	0.650	0.625	0.613	0.606	0.603	0.602	0.601	0.600	0.600	0.600

图 18. 连续 12 夜鸡兔互变比例，兔的初始比例更高

求解平衡状态

农夫把 (24) 代入 (3)，得到：

$$\begin{bmatrix} p & q \\ 1-p & 1-q \end{bmatrix} \begin{bmatrix} \pi_1 \\ \pi_2 \end{bmatrix} = \begin{bmatrix} \pi_1 \\ \pi_2 \end{bmatrix} \tag{25}$$

另外，状态向量本身元素相加为 1，由此农夫得到两个等式。

$$\begin{cases} p\pi_1 + q\pi_2 = \pi_1 \\ \pi_1 + \pi_2 = 1 \end{cases} \tag{26}$$

求解二元一次线性方程组得到：

$$\begin{cases} \pi_1 = \dfrac{q}{1-p+q} \\ \pi_2 = \dfrac{1-p}{1-p+q} \end{cases} \tag{27}$$

农夫记得他假设 $p = 0.7$，$q = 0.2$，代入 (27) 得到：

$$\begin{cases} \pi_1 = 0.4 \\ \pi_2 = 0.6 \end{cases} \tag{28}$$

也就鸡兔互变平衡时，稳态向量 $\boldsymbol{\pi}$ 为：

$$\boldsymbol{\pi} = \begin{bmatrix} \pi_1 \\ \pi_2 \end{bmatrix} = \begin{bmatrix} 0.4 \\ 0.6 \end{bmatrix} \tag{29}$$

这和农夫之前做的模拟实验结果完全一致！真可谓“山重水复疑无路，柳暗花明又一村。”

也就是说，$\boldsymbol{T}$ 乘上 (29) 中的稳态向量 $\boldsymbol{\pi}$，结果还是稳态向量 $\boldsymbol{\pi}$：

$$\boldsymbol{T\pi} = \boldsymbol{\pi} \quad \Rightarrow \quad \underbrace{\begin{bmatrix} 0.7 & 0.2 \\ 0.3 & 0.8 \end{bmatrix}}_{\boldsymbol{T}} \underbrace{\begin{bmatrix} 0.4 \\ 0.6 \end{bmatrix}}_{\boldsymbol{\pi}} = \underbrace{\begin{bmatrix} 0.4 \\ 0.6 \end{bmatrix}}_{\boldsymbol{\pi}} \tag{30}$$

农夫突然记起这就是前几日他读到的特征值分解 (eigen decomposition)！书上反复提到特征值分解的重要性，农夫今天也见识到这个数学利器的伟力。

以下代码绘制本节 11 幅热图。

```python
# Bk3_Ch25_2

import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

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

pi_i = np.matrix([[0.6],
                  [0.4]])

all_max = 1
all_min = 0


pi_array = np.vstack((np.linspace(1,0,11),1 - np.linspace(1,0,11)))
pi_array=np.matrix(pi_array)
num_steps = 12
```

```python
for ini in np.arange(0,np.shape(pi_array)[1]):

    pi = pi_array[:,ini]


    fig, axes = plt.subplots(1, num_steps + 1, figsize=(12, 3))

    for i in np.arange(0,num_steps + 1):

        plt.sca(axes[i])
        ax = sns.heatmap(pi,cmap='RdYlBu_r',vmax = all_max,vmin = all_min,
                         annot = True,fmt=".3f",cbar=False,
                         xticklabels=False, yticklabels=False)
        ax.set_aspect("equal")
        plt.title('$\pi(' + str(i) + ')$')
        ax.tick_params(left=False, bottom=False)

        pi = T@pi
```

25.5 有向量的地方，就有几何

农夫学习线性代数时，总结了几句真经。其中一句就是——有向量的地方，就有几何。

他决定透过几何这个视角来看看状态向量的变化。

农夫把图 16、图 17、图 18 对应的 11 种状态向量的初始值画在平面直角坐标系中，用"有方向的线段"代表具体向量数值。

在他画的图 19 这 11 幅子图中，紫色向量代表鸡兔初始比例 $\boldsymbol{\pi}(0)$，红色向量代表经过 12 夜鸡兔互变后 $\boldsymbol{\pi}(12)$ 的位置。

农夫发现不管初始比例 $\boldsymbol{\pi}(0)$ 如何，也就是紫色向量位于任何方位；经过 12 夜持续变化，红色向量 $\boldsymbol{\pi}(12)$ 的位置几乎完全一致。特别地，如图 19 (g) 所示，当初始比例 $\boldsymbol{\pi}(0)$ 就是稳态向量时：

$$\boldsymbol{\pi}(0)=\begin{bmatrix}0.4\\0.6\end{bmatrix} \tag{31}$$

转移矩阵 $\boldsymbol{T}$ 没有改变 $\boldsymbol{\pi}(0)$ 的方向；农夫查阅典籍发现，这个向量也有自己的名字，它叫做 $\boldsymbol{T}$ 的特征向量 (eigenvector)。

而且，他发现变化过程，向量终点都落在一条直线上。这条直线代表——鸡、兔比例之和为 1。

农夫在图 19 中还图了另外一组向量，这些向量都是单位向量，对应：

$$\frac{\boldsymbol{\pi}}{\|\boldsymbol{\pi}\|} \tag{32}$$

这一组向量终点都落在单位圆上，因为它们的模都是 1。

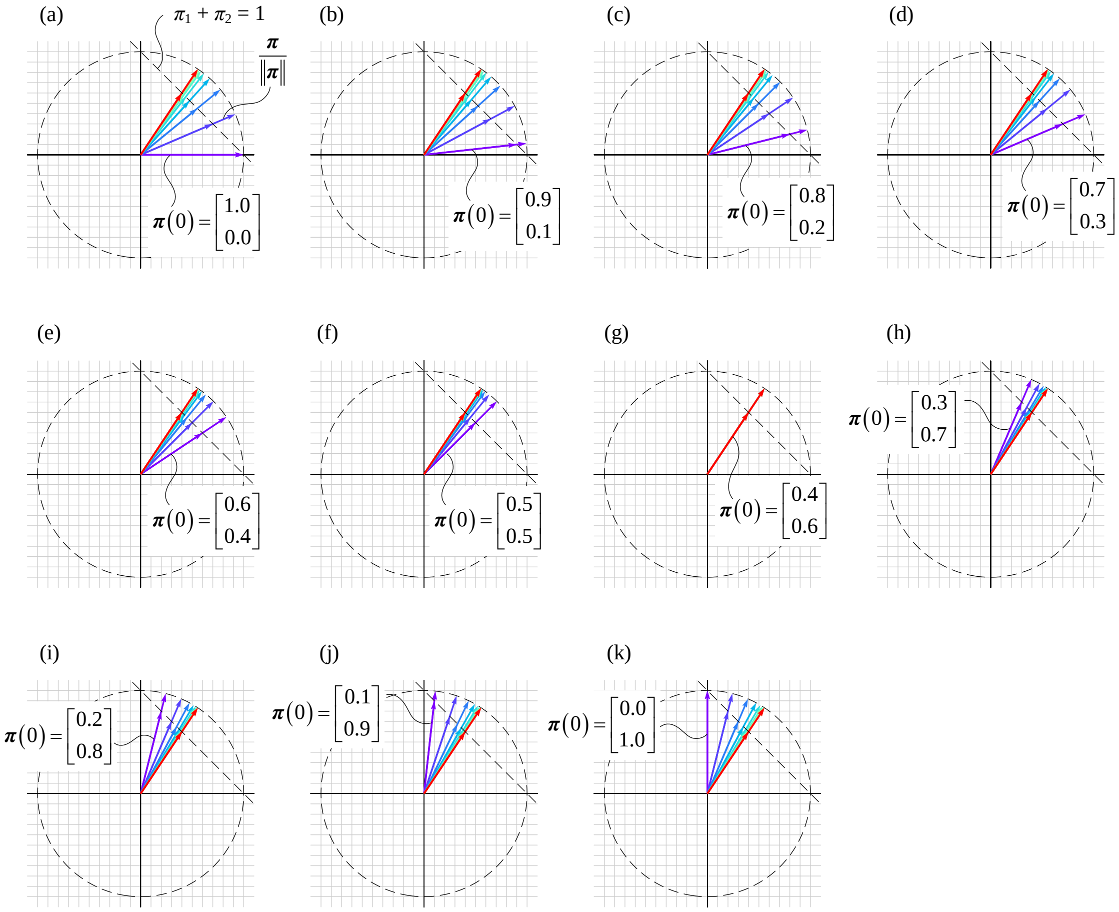


图 19. 连续 12 夜鸡兔互变比例，几何视角

以下代码绘制图 19。

```
# Bk3_Ch25_3

import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

def draw_vector(vector,RBG, ax):

    ax.quiver(0, 0, vector[0], vector[1],
              angles='xy',
              scale_units='xy',
              scale=1,
              color = RBG)

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

all_max = 1
all_min = 0
```

```
x1 = np.linspace(-1.1, 1.1, num=201)
x2 = x1
xx1, xx2 = np.meshgrid(x1,x2)
zz = ((np.abs((xx1))**2) + (np.abs((xx2))**2))**(1./2)

pi_array = np.vstack((np.linspace(1,0,11),1 - np.linspace(1,0,11)))
pi_array=np.matrix(pi_array)
num_steps = 12

colors = plt.cm.rainbow(np.linspace(0,1,num_steps + 1))

for ini in np.arange(0,np.shape(pi_array)[1]):

    pi = pi_array[:,ini]


    fig, ax = plt.subplots(figsize=(10, 10))

    # plot a reference line
    plt.plot(x1,1-x1,color = 'k',
             linestyle = '--')

    # plot a unit circle as reference
    plt.contour(xx1, xx2, zz, levels = [1],
                colors='k', linestyles = ['--'])

    for i in np.arange(0,num_steps + 1):

        # plot normalized vector
        draw_vector(pi/np.linalg.norm(pi), colors[i], ax)

        # plot original vector
        draw_vector(pi, colors[i], ax)

        ax.tick_params(left=False, bottom=False)
        ax.set_xlim(-1.1, 1.1)
        ax.set_ylim(-1.1, 1.1)
        # plt.axis('off')
        ax.axvline(x = 0, color = 'k')
        ax.axhline(y = 0, color = 'k')
        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.grid(color = [0.8,0.8,0.8])
        plt.xticks(np.linspace(-1,1,21))
        plt.yticks(np.linspace(-1,1,21))

        pi = T@pi
        # update pi
```

25.7 彩蛋

至此，小村村民心中一块大石头算是落地了。对于“鸡兔互变”这个奇事，大伙儿也都见怪不怪了！

前后脚的事儿，村民发现鸡兔互变也停了。笑容在大伙儿脸上绽开，农夫把全村老少都邀到自家菜园，要好好欢庆一番！

大伙儿都没闲着，摘果蔬，网肥鱼，蒸米饭，取美酒，摆桌椅，嘉宾纷沓，鼓瑟吹笙，烹羊宰牛且为乐，会须一饮三百杯 …

看着被这阵仗吓坏了的一笼鸡兔，农夫说，“你们这次立了大功，留着过年吧！”

欢言酌春酒，摘我园中蔬。微雨从东来，好风与之俱。

变与不变

书到用时方恨少，腹有诗书气自华，农夫这次让大伙儿理解了这两句话的精髓。

经过这场线性代数风暴之后，小村村民白天田间耕作时都会怀揣一本数学典籍，一得片刻休息，大伙儿分秒必争、手不释卷。夜深人静时，焚膏继晷、挑灯夜读者甚多。大伙儿不再惧怕未知，因为“我们必须知道，我们终将知道。”

渐渐地，这个曾经与世隔绝的小村处处都在变化，村民们也都肉眼可见地变化。你让我说，村民哪里发生了变化？我也说不上。反正，时时刻刻都在变化，感觉一切都在变得更好。

而不变的是，小村还是那个小村，村民还是咱们这五十几户村民。

云山青青，凤泉泠泠。山色依旧可爱，泉声更是可听。

(镜头拉远拉高) 一川松竹任横斜，有人家，被云遮。

东风升，云雾腾。

紫气东来，祥云西至。

这场鸡兔同笼引发的思想风暴，似乎给这个沉睡千年的村庄带了什么，也似乎带走了什么。

好像什么都没有发生，又好像要发生什么。

往时曾发生的，来日终将发生。