

第09讲 基底变换 扩展练习

作者：欧新宇 (Xinyu OU)

本文档所展示的测试结果，均运行于：Intel Core i7-7700K CPU 4.2GHz

1. 【计算题】任意基的转移矩阵

令 $v_1 = (3, 3)^T$, $v_2 = (3, 5)^T$ 对有序基 $u_1 = (2, 2)^T$, $u_2 = (-2, 3)^T$ ，求从 $[v_1, v_2]$ 到 $[u_1, u_2]$ 的转移矩阵。

- Python实现

```
import numpy as np
from scipy import linalg

v = np.array(____(1)____)
U = np.array([[2, -2], [2, 3]])
S = ____ (2) ____

np.set_printoptions(precision=3, suppress=True) # 设置输出矩阵保留三位小数
print('S=\n{}'.format(S))
```

答案及解析

```
(1) [[3, 3], [3, 5]]
(2) np.dot(linalg.inv(U), v)
```

问题分析：从特定基 $[u_1, u_2]$ 到特定基 $[v_1, v_2]$ 的基底变换（坐标变换）。此处，求 $[v_1, v_2]$ 到 $[u_1, u_2]$ 的转移矩阵，可以直接套用公式 $S = U^{-1}V$

$$S_1 = U^{-1}V = \begin{bmatrix} 2 & -2 \\ 2 & 3 \end{bmatrix}^{-1} \begin{bmatrix} 3 & 3 \\ 3 & 5 \end{bmatrix}$$

```
import numpy as np
from scipy import linalg

v = np.array([[3, 3], [3, 5]])
U = np.array([[2, -2], [2, 3]])
S = np.dot(linalg.inv(U), v)

np.set_printoptions(precision=3, suppress=True) # 设置保留三位小数
print('S=\n{}'.format(S))
```

```
S=
[[1.5 1.9]
 [0.  0.4]]
```

2. 【计算题】任意基间的转换

令 $E = [(2, 4)^T, (3, 4)^T]$, 并令 $u = (5, 4)^T, v = (2, 3)^T$ 。计算 $[u]_E, [v]_E$ 。

- Python实现

```
import numpy as np
from scipy import linalg

E = np.array(____(1)____)
u = np.array([[5],[4]])
v = np.array([[2],[3]])

np.set_printoptions(precision=3, suppress=True) # 设置保留三位小数
print('uE=\n{}'.format(____(2)____))
print('vE=\n{}'.format(____(3)____))
```

答案及解析

```
(1) [[2,3],[4,4]]
(2) np.dot(linalg.inv(E), u)
(3) np.dot(linalg.inv(E), v)
```

问题分析：从标准基 $[e_1, e_2]$ 到特定基 $[u_1, u_2]$ 的基底变换（坐标变换）。根据坐标变换公式，基坐标 $x = Uc$ ，我们可以得到从标准基坐标向特定基坐标的变换公式： $c = U^{-1}x$ ，其中， U^{-1} 就是标准基 $[e_1, e_2]$ 到特定基 $[u_1, u_2]$ 的转移矩阵。

$$1). [u]_E = E^{-1}u = \begin{bmatrix} 2 & 3 \\ 4 & 4 \end{bmatrix}^{-1} \begin{bmatrix} 5 \\ 4 \end{bmatrix}$$

$$2). [v]_E = E^{-1}v = \begin{bmatrix} 2 & 3 \\ 4 & 4 \end{bmatrix}^{-1} \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

```
import numpy as np
from scipy import linalg

E = np.array([[2,3],[4,4]])
u = np.array([[5],[4]])
v = np.array([[2],[3]])

np.set_printoptions(precision=3, suppress=True) # 设置保留三位小数
print('uE=\n{}'.format(np.dot(linalg.inv(E), u)))
print('vE=\n{}'.format(np.dot(linalg.inv(E), v)))
```

```
uE=
[[-2.]
 [ 3.]]
vE=
[[0.25]
 [0.5 ]]
```

3. 【计算题】任意基间的基底变换

令 $v_1 = (1, 3, 5)^T, v_2 = (2, 4, 6)^T, v_3 = (2, 5, 3)^T$, 并令 $u_1 = (9, -7, 6)^T, u_2 = (8, 6, -3)^T, u_3 = (-2, 4, 6)^T$, 求:

(1). 求从有序基 $[v_1, v_2, v_3]$ 到有序基 $[u_1, u_2, u_3]$ 的转移矩阵。

(2). 若 $x = 3v_1 + 2v_2 - 4v_3$, 确定向量 x 相应于 u_1, u_2, u_3 的坐标。

• Python实现

```
import numpy as np
from scipy import linalg

U = np.array([[9, 8, -2], [-7, 6, 4], [6, -3, 6]])
V = np.array([[1, 2, 2], [3, 4, 5], [5, 6, 3]])
xv = np.array(____(1)____)

S = ____ (2) ____
xu = ____ (3) ____

np.set_printoptions(precision=3, suppress=True) # 设置保留三位小数
print('S=\n {}'.format(S))
print('xu=\n {}'.format(xu))
```

答案及解析

```
(1) [[3], [2], [-4]]
(2) np.dot(linalg.inv(U), V)
(2) np.dot(S, xv)
```

问题分析：从有序基 $[v_1, v_2, v_3]$ 到有序基 $[u_1, u_2, u_3]$ 的基底变换（坐标变换）。根据坐标变换的通用公式 $Vc = Ud$, 即可求得坐标和转移矩阵。

(1). 要获得 v 到 u 的转移矩阵, 实际上就是将 v 到 e 和 e 到 u 的转移矩阵进行合成, 那么可以得到:

$$S = U^{-1}V = \begin{bmatrix} 9 & 8 & -2 \\ -7 & 6 & 4 \\ 6 & -3 & 6 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 2 & 2 \\ 3 & 4 & 5 \\ 5 & 6 & 3 \end{bmatrix}$$

(2). 基于 v 的坐标 x 转换为基于 u 的坐标, 可以通过通用公式的变形获得, 即:

$$Vc = Ud \Rightarrow d = U^{-1}Vc \Rightarrow x_{new} = U^{-1}Vx = Sx = S \begin{bmatrix} 3 \\ 2 \\ -4 \end{bmatrix}$$

```
import numpy as np
from scipy import linalg

U = np.array([[9, 8, -2], [-7, 6, 4], [6, -3, 6]])
V = np.array([[1, 2, 2], [3, 4, 5], [5, 6, 3]])
xv = np.array([[3], [2], [-4]])

S = np.dot(linalg.inv(U), V)
xu = np.dot(S, xv)

np.set_printoptions(precision=3, suppress=True) # 设置保留三位小数
print('S=\n {}'.format(S))
print('xu=\n {}'.format(xu))
```

```
S=
[[0.143 0.194 0.018]
 [0.156 0.267 0.4   ]
 [0.768 0.939 0.682]]
xu=
[[ 0.745]
 [-0.6   ]
 [ 1.455]]
```

4. 【计算题】任意基间的转移矩阵

给定 $v_1 = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$, $v_2 = \begin{bmatrix} -3 \\ -4 \end{bmatrix}$, $S = \begin{bmatrix} 2 & -2 \\ 3 & 5 \end{bmatrix}$, 求: u_1, u_2 , 使得 S 为从 $[v_1, v_2]$ 到 $[u_1, u_2]$ 的转移矩阵。

- Python实现

```
import numpy as np
from scipy import linalg

S = ____ (1) ____
V = np.array([[3, -3], [5, -4]])
U = ____ (2) ____

np.set_printoptions(precision=3, suppress=True) # 设置保留三位小数
print('U1=\n {}'.format(U[:,0]))
print('U2=\n {}'.format(U[:,1]))
```

答案及解析

```
(1) np.array([[2, -2], [3, 5]])
(2) np.dot(V, linalg.inv(S))
```

问题分析: 从特定基 $[u_1, u_2]$ 到特定基 $[v_1, v_2]$ 的基底变换 (坐标变换)。由于 S 是 v 到 u 的转移矩阵, 所以有 $S = U^{-1}V$, 由此可以得到 $V = US \Rightarrow U = VS^{-1}$

$$U = VS^{-1} = \begin{bmatrix} 3 & -3 \\ 5 & -4 \end{bmatrix} \begin{bmatrix} 2 & -2 \\ 3 & 5 \end{bmatrix}^{-1}$$

```
import numpy as np
from scipy import linalg

S = np.array([[2, -2], [3, 5]])
V = np.array([[3, -3], [5, -4]])
U = np.dot(V, linalg.inv(S))

np.set_printoptions(precision=3, suppress=True) # 设置保留三位小数
print('U1=\n {}'.format(U[:,0]))
print('U2=\n {}'.format(U[:,1]))
```

```
U1=  
[1.5  2.312]  
U2=  
[0.   0.125]
```