

SciPy 科學計算函式庫－基礎篇

蔡尚融

2018-05-07

擴充程式庫 SciPy

- Python 的一組擴充函式庫，提供科學計算的函式庫。
<https://www.scipy.org/>
- NumPy、SciPy、Matplotlib。
- NumPy 主要透過多維陣列（n-dimensional array）物件處理；其類別名稱為 ndarray。
- 提供線性代數（linear algebra）運算、傅立葉轉換（Fourier transform）。
- 載入 NumPy 模組

```
import numpy as np
```

■ 向量與矩陣

向量：

```
>>> vu = np.array([1, 2, 3])
```

矩陣：

```
>>> ma = np.array([[4, 5, 6], [7, 8, 9]])
```

■ ndarray.ndim 取得維度 (dimension)。

```
>>> vu.ndim
```

```
1
```

```
>>> ma.ndim
```

```
2
```

■ ndarray.shape 取得每個維度陣列長度。

```
>>> vu.shape
```

```
(3,)
```

```
>>> ma.shape
```

```
(2, 3)
```

- `ndarray.size` 取得陣列元素總數。

```
>>> vu.size
```

```
3
```

```
>>> ma.size
```

```
6
```

- `ndarray.dtype` 取得元素資料格式。

```
>>> ma.dtype
```

```
dtype('int64')
```

- `ndarray.data` 取得記憶體對應位置。

■ 浮點數矩陣。

```
>>> mb = np.array([[1.0, 2.3, 4.5], [2.3, 3.5, 4.0]])  
>>> mb.dtype  
dtype('float64')
```

■ 複數 (complex) 浮點數矩陣。

```
mc = np.array([[1, 2], [3, 4]], dtype=complex)  
>>> mc  
array([[ 1.+0.j,  2.+0.j],  
       [ 3.+0.j,  4.+0.j]])
```

■ 取值與設值類似串列 (list)。

```
>>> mb[1][1] = mb[1][1] + 2.3  
>>> mb[1][1]  
5.8
```

- `zeros()` 產生值全為零的陣列。

```
>>> np.zeros(3)
array([ 0., 0., 0.])
>>> np.zeros((2, 3))
array([[ 0., 0., 0.],
       [ 0., 0., 0.]])
```

- `ones()` 產生值全為一的陣列。

```
>>> np.ones((3, 4))
array([[ 1., 1., 1., 1.],
       [ 1., 1., 1., 1.],
       [ 1., 1., 1., 1.]])
```

- `eye()` 單位矩陣 (identity matrix) 。

```
>>> np.eye(5)
array([[ 1.,  0.,  0.,  0.,  0.],
       [ 0.,  1.,  0.,  0.,  0.],
       [ 0.,  0.,  1.,  0.,  0.],
       [ 0.,  0.,  0.,  1.,  0.],
       [ 0.,  0.,  0.,  0.,  1.]])

>>> np.eye(3, 4)
array([[ 1.,  0.,  0.,  0.],
       [ 0.,  1.,  0.,  0.],
       [ 0.,  0.,  1.,  0.]])
```

隨機矩陣

- 產生 $m \times n$ 的隨機矩陣，其元素值介於 0 與 1 之間。

```
np.random.rand(m, n)
```

```
>>> np.random.rand(3, 4)
```

```
array([[ 0.5438721 ,  0.93418492,  0.71879328,  0.25593889],  
       [ 0.87653749,  0.23116355,  0.05269307,  0.68404487],  
       [ 0.66943055,  0.37236087,  0.34052332,  0.95161537]])
```

- 產生常態分佈的隨機矩陣。

```
np.random.randn(m, n)
```


- NumPy 提供類似 range 的功能，可產生一序列指定範圍與步長的數字。

```
>>> np.arange(1, 9, 2)
```

```
array([1, 3, 5, 7])
```

可支援浮點數：

```
>>> np.arange(1.1, 3.7, 0.7)
```

```
array([ 1.1, 1.8, 2.5, 3.2])
```

- linspace 將指定範圍分成數段。

```
>>> from numpy import pi
```

```
>>> np.linspace(0.5, 2.5, 5)
```

```
array([ 0.5, 1. , 1.5, 2. , 2.5])
```

- `diag()` 取對角線值。

```
>>> x = np.arange(25).reshape((5,5))
>>> np.diag(x)
array([ 0,  6, 12, 18, 24])
>>> np.diag(x, k=1)
array([ 1,  7, 13, 19])
>>> np.diag(x, k=-2)
array([10, 16, 22])
```

- 用 `diag()` 將向量轉對角矩陣。

```
>>> np.diag(np.arange(1.1, 3.5, 1.1))
array([[ 1.1,  0. ,  0. ],
       [ 0. ,  2.2,  0. ],
       [ 0. ,  0. ,  3.3]])
```

基本運算

- 運算子 $+$ $-$ $*$ $/$ $**$ $==$ 為元素運算。

```
>>> ma ** 2  
array([[16, 25, 36],  
       [49, 64, 81]])
```

- 矩陣乘法：dot

```
>>> MA = np.array([[1, 2], [3, 4]])  
>>> MB = np.array([[4, -2], [-3, 1]])  
>>> MA.dot(MB)  
array([[-2, 0],  
       [ 0, -2]])
```

或是

```
>>> np.dot(MA, MB)  
array([[-2, 0],  
       [ 0, -2]])
```

NumPy 的線性代數物件

- `from numpy import linalg`
- 矩陣與向量範數 (norm) : `linalg.norm()`
- 矩陣條件數 (condition number) : `linalg.cond()`
- 矩陣行列式 (determinant) : `linalg.det()`
- 矩陣特徵值 (eigenvalue) :
`linalg.eig()` , `linalg.eigvals()` 。
- 解線性系統 (system of linear equations) : `linalg.solve()`
- 求反矩陣 : `linalg.inv()`

隨堂練習：

求下列矩陣的行列式值與反矩陣。

$$\begin{bmatrix} 2 & 3 & 4 & 5 \\ 0 & 3 & 4 & 5 \\ 0 & 0 & 4 & 5 \\ 0 & 0 & 0 & 5 \end{bmatrix}$$