矩阵计算 python 笔记

王梦圆

2020-02

1 创建一个向量

在 python 中可以用 numpy.array() 来创建一个向量，例如：

```
import numpy as np
x = np.array([1,2,3,4])
print(x)
```

2 创建一个矩阵

在 python 里面使用 numpy 创建矩阵有两种方法：第一种：直接法——np.matrix() 创建矩阵

```
x = np.matrix([[1,2,3,4],[5,6,7,8],[9,10,11,12]])
x
```

第二种：利用 np.arange().reshape(shape,order) 创建矩阵

```
x1 = np.arange(12).reshape(3,4)#3行4列矩阵，默认按行排列
x2 = np.arange(12).reshape(4,3)#4行3列矩阵，按行排列
x3 = np.arange(12).reshape((4,3),order='F')#按列排列
```

二维数组不能直接定义行名和列名，可以先转化成 DataFrame 形式，再利用 index 和 columns 定义行名和列名

```
import pandas as pd
index=['r1','r2','r3']
columns=['c1','c2','c3','c4']
x1 = pd.DataFrame(x1,index=index,columns=columns)
x1
```

3 矩阵转置

A 为 $m \times n$ 矩阵，求 A 的转置在 python 中可用 A.T，例如：

```
A = np.arange(1,13).reshape(3,4)
A
```

```
A.T
```

在 python 里面若想要得到一个行向量，例如：

```
x = np.array([1,2,3,4,5,6,7,8,9,10]).reshape((1,-1))
type(x)
```

利用 x.T, 可以得到一个列向量，例如

```
x.T
```

```
(x.T).shape
```

在 python 里面若想要得到一个列向量，例如：

```
x = np.array([1,2,3,4,5,6,7,8,9,10]).reshape((-1,1))
x.shape
```

4 矩阵相加减

在 python 中对同行同列矩阵相加减，可用符号：“+”、“−”，例如：

```
A = B = np.arange(1,13).reshape((3,4),order='F')
A+B
```

```
A-B
```

5 数与矩阵相乘

A 为 $m \times n$ 矩阵，c>0，在 python 中求 c**A** 可用符号：“*”，例如：

```
c =2
c*A
```

6 矩阵相乘

A 为 m×n 矩阵，**B** 为 n×k 矩阵，在 python 中求 **AB** 可用 np.dot()，例如：

```
A = np.arange(1,13).reshape((3,4),order='F')
B = np.arange(1,13).reshape((4,3),order='F')
np.dot(A,B)
```

7 矩阵对角元素相关运算

例如要取一个方阵的对角元素，

```
A = np.arange(1,17).reshape((4,4),order='F')
np.diag(A)
```

对一个向量应用 np.diag() 函数将产生以这个向量为对角元素的对角矩阵，例如：

```
A = np.arange(1,17).reshape((4,4),order='F')
np.diag(np.diag(A))
```

对一个正整数 z 应用 np.identity() 函数将产生以 z 维单位矩阵，例如：

```
np.identity(3)
```

8 矩阵求逆

矩阵求逆可用函数 np.linalg.inv()，例如：

```
A = np.random.normal(0,1,(4,4))
np.linalg.inv(A)
```

```
np.dot(np.linalg.inv(A),A)
```

9 矩阵的特征值与特征向量

矩阵 A 的谱分解为 **A=UΛU'**, 其中 Λ 是由 A 的特征值组成的对角矩阵，U 的列为 A 的特征值对应的特征向量，在 python 中可以用函数 np.linalg.eig() 函数得到 U 和 Λ

```
A = np.identity(4)+1
A_eig = np.linalg.eig(A)
Λ = np.diag(A_eig[0])
U = A_eig[1]
Λ,U
```

```
np.dot(np.dot(U,Λ),U.T)
```

```
np.dot(U.T,U)
```

10 矩阵的 Choleskey 分解

对于正定矩阵 **A**，可对其进行 Choleskey 分解，即：**A=PP'**，其中 P 为下三角矩阵，在 python 中可以用函数 np.linalg.cholesky() 进行 Choleskey 分解，例如：

```
A = np.identity(4)+1
cho = np.linalg.cholesky(A)
np.dot(cho,cho.T)
```

若矩阵为对称正定矩阵，可以利用 Choleskey 分解求行列式的值，如：

```
np.prod(np.diag(cho)**2)
```

```
np.linalg.det(A)
```

11 矩阵奇异值分解

A 为 $m \times n$ 矩阵，rank(A)= r, 可以分解为：**A=UDV'**, 其中 **U'U=V'V=I**。在 python 中可以用函数 scd() 进行奇异值分解，例如：

```
A = np.arange(1,19).reshape((3,6),order='F')
A
```

```
d = np.linalg.svd(A,full_matrices=0)[1]
d
```

```
u = np.linalg.svd(A,full_matrices=0)[0]
u
```

```
v = np.linalg.svd(A,full_matrices=0)[2]
v
```

```
np.dot(np.dot(u,np.diag(d)),v)
```

```
np.dot(u.T,u,out=None,)
```

```
np.dot(v,v.T)
```

12 矩阵 QR 分解

A 为 $m \times n$ 矩阵可以进行 QR 分解，**A=QR**，其中：**Q'Q = I**，在 python 中可以用函数 qr() 进行 QR 分解，得到 Q 矩阵和 R 矩阵，例如：

```
A = np.arange(1,17).reshape((4,4),order='F')
qr = np.linalg.qr(A)
Q = qr[0]
R = qr[1]
np.dot(Q,R)
```

```
np.dot(Q.T,Q)
```

13 矩阵广义逆（Moore-Penrose）

$n \times m$ 矩阵 $\mathbf{A}+$ 称为 $m \times n$ 矩阵 $\mathbf{A}$ 的 Moore-Penrose 逆，如果它满足下列条件：$\mathbf{A\ A+A=A}$；$\mathbf{A+A\ A+=\ A+}$；$\mathbf{(A\ A+)H=A\ A+}$；$\mathbf{(A+A)H=\ A+A}$ 在 python 中可用 np.linalg.pinv() 函数，例如：

```
A = np.arange(1,17).reshape((4,4),order='F')
MP = np.linalg.pinv(A)#广义逆
```

验证性质 1：

```
np.dot(np.dot(A,MP),A)
```

验证性质 2：

```
np.dot(np.dot(MP,A),MP)
```

验证性质 3：

```
np.dot(A,MP).T
```

```
np.dot(A,MP)
```

验证性质 4：

```
np.dot(MP,A).T
```

```
np.dot(MP,A)
```

14 矩阵 Kronecker 积

n×m 矩阵 A 与 h×k 矩阵 B 的 kronecker 积为一个 nh×mk 维矩阵，公式为：

$$\mathbf{A}_{m\times n} \otimes \mathbf{B}_{h\times k} = \begin{pmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \vdots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{pmatrix}_{mh\times nh}$$

在 python 中 kronecker 积可以用函数 np.kron() 来计算，例如：

```
A = np.arange(1,5).reshape((2,2),order='F')
B = np.ones((2,2))
A
```

```
B
```

```
np.kron(A,B)
```

15 矩阵的维数

在 python 中可以直接利用 (matrix).shape 得到矩阵的维数

```
A = np.arange(1,13).reshape((3,4),order='F')
A
```

```
A.shape
```

16 矩阵的行和、列和、行平均与列平均

在 python 中很容易求得一个矩阵的各行的和、平均数与列的和、平均数，例如：

```
A = np.arange(1,13).reshape((3,4),order='F')
A
```

```
A.sum(axis=1)#行和
```

```
A.mean(axis=1)#行平均
```

```
A.sum(axis=0)#列和
```

```
A.mean(axis=0)#列平均
```

17 矩阵 X'X 的逆

在统计计算中，我们常常需要计算这样矩阵的逆，如 OLS 估计中求系数矩阵。R 中的包 "strucchange" 提供了有效的计算方法。

```
A = np.arange(1,13).reshape((3,4),order='F')
A
```

```
A.sum(axis=1)#行和
```

```
A.mean(axis=1)#行平均
```

```
A.sum(axis=0)#列和
```

```
A.mean(axis=0)#列平均
```