



SciPy

SciPy 是基于 NumPy 创建的 Python 科学计算核心库，提供了众多数学算法与函数。



与 NumPy 交互

[参阅 NumPy](#)

```
>>> import numpy as np
>>> a = np.array([1,2,3])
>>> b = np.array([(1+5j,2j,3j), (4j,5j,6j)])
>>> c = np.array([(1.5,2,3), (4,5,6)], [(3,2,1), (4,5,6)])
```

索引技巧

```
>>> np.mgrid[0:5,0:5]      创建稠密栅格
>>> np.ogrid[0:2,0:2]     创建开放栅格
>>> np.r_[[3,[0]*5,-1:1:10j]]  按行纵向堆叠数组按列
>>> np.c_[b,c]            列横向堆叠数组
```

操控形状

```
>>> np.transpose(b)      转置矩阵
>>> b.flatten()          拉平数组
>>> np.hstack((b,c))     按列横向堆叠数组
>>> np.vstack((a,b))     按行纵向堆叠数组
>>> np.hsplit(c,2)       在索引2横向分割数组
>>> np.vsplit(d,2)       在索引2纵向分割数组
```

多项式

```
>>> from numpy import poly1d
>>> p = poly1d([3,4,5])  创建多项式对象
```

矢量函数

```
>>> def myfunc(a):
    if a < 0:
        return a*2
    else:
        return a/2
>>> np.vectorize(myfunc)  矢量函数
```

类型控制

```
>>> np.real(c)           返回数组元素的实部
>>> np.imag(c)           返回数组元素的虚部
>>> np.real_if_close(c,tol=1000)  如果复数接近0, 返回实部将
>>> np.cast['f'](np.pi)  对象转化为数据类型
```

常用函数

```
>>> np.angle(b,deg=True)  返回复数的角度
>>> g = np.linspace(0,np.pi,num=5)  创建等差数组 (样本数)
>>> g[3:] += np.pi
>>> np.unwrap(g)          解包
>>> np.logspace(0,10,3)   创建等差数组 (对数刻度)
>>> np.select([c<4],[c*2])  根据条件返回数组列表的值

>>> misc.factorial(a)     因子
>>> misc.comb(10,3,exact=True)  取K次N项的组合, 已改为scipy.special.comb
>>> misc.central_diff_weights(3)  NP点中心导数的权重
>>> misc.derivative(myfunc,1.0)  查找函数在某点的第n个导数
```

线性代数

使用 linalg 和 sparse 模块。注意 scipy.linalg 包含了 numpy.linalg, 并扩展了其功能。

```
>>> from scipy import linalg, sparse
```

创建矩阵

```
>>> A = np.matrix(np.random.random((2,2)))
>>> B = np.asmatrix(b)
>>> C = np.mat(np.random.random((10,5)))
>>> D = np.mat([[3,4], [5,6]])
```

基础矩阵例程

```
逆矩阵
>>> A.I          求逆矩阵
>>> linalg.inv(A)  求逆矩阵
>>> A.T          矩阵转置
>>> A.H          共轭转置
>>> np.trace(A)   计算对角线元素的和

范数
>>> linalg.norm(A)  Frobenius 范数
>>> linalg.norm(A,1) L1 范数 (最大列汇总)
>>> linalg.norm(A,np.inf) L 范数 (最大列汇总)

排名
>>> np.linalg.matrix_rank(C)  矩阵排名

行列式
>>> linalg.det(A)           行列式

求解线性问题
>>> linalg.solve(A,b)       求解稠密矩阵
>>> E = np.mat(a).T         求解稠密矩阵
>>> linalg.lstsq(D,E)       用最小二乘法求解线性代数方程

广义逆
>>> linalg.pinv(C)          计算矩阵的伪逆 (最小二乘法求解器)
>>> linalg.pinv2(C)         计算矩阵的伪逆 (SVD)
```

创建稀疏矩阵

```
>>> F = np.eye(3, k=1)      创建2X2单位矩阵
>>> G = np.mat(np.identity(2))  创建2X2单位矩阵
>>> C[C > 0.5] = 0
>>> H = sparse.csr_matrix(C)  压缩稀疏矩阵
>>> I = sparse.csc_matrix(D)  压缩稀疏列矩阵
>>> J = sparse.dok_matrix(A)  DOK矩阵
>>> E.todense()              将稀疏矩阵转为全矩阵
>>> sparse.isspmatrix_csc(A)  单位稀疏矩阵
```

稀疏矩阵例程

```
逆矩阵
>>> sparse.linalg.inv(I)     求逆矩阵

范数
>>> sparse.linalg.norm(I)    范数

解决线性问题
>>> sparse.linalg.spsolve(H,I)  稀疏求解矩阵
```

稀疏矩阵函数

```
>>> sparse.linalg.expm(I)    稀疏矩阵指数
```

调用帮助

```
>>> help(scipy.linalg.diagsvd)
>>> np.info(np.matrix)
```

矩阵函数

```
加法
>>> np.add(A,D)            加法

减法
>>> np.subtract(A,D)       减法

除法
>>> np.divide(A,D)         除法

乘法
>>> np.multiply(D,A)       乘法
>>> np.dot(A,D)           点积
>>> np.vdot(A,D)          向量点积
>>> np.inner(A,D)         内积
>>> np.outer(A,D)         外积
>>> np.tensordot(A,D)     张量点积
>>> np.kron(A,D)          Kronecker 积

指数函数
>>> linalg.expm(A)         矩阵指数
>>> linalg.expm2(A)       矩阵指数 (泰勒级数)
>>> linalg.expm3(D)       矩阵指数 (特征值分解)

对数函数
>>> linalg.logm(A)        矩阵对数

三角函数
>>> linalg.sinm(D)        矩阵正弦
>>> linalg.cosm(D)        矩阵余弦
>>> linalg.tanm(A)        矩阵切线

双曲三角函数
>>> linalg.sinhm(D)       双曲矩阵正弦
>>> linalg.coshm(D)       双曲矩阵余弦
>>> linalg.tanhm(A)       双曲矩阵切线

矩阵符号函数
>>> np.sigm(A)            矩阵符号函数
>>> linalg.sqrtm(A)       矩阵平方根
>>> linalg.sqrtm(A)       矩阵平方根

任意函数
>>> linalg.funm(A, lambda x: x*x)  评估矩阵函数
```

矩阵分解

```
特征值与特征向量
>>> la, v = linalg.eig(A)   求解方阵的普通或广义特征值问题

>>> l1, l2 = la
>>> v[:,0]                  解包特征值
>>> v[:,1]                  第一个特征值
>>> linalg.eigvals(A)      第二个特征值
>>> linalg.eigvals(A)      解包特征值

奇异值分解
>>> U,s,Vh = linalg.svd(B)  奇异值分解 (SVD)
>>> M,N = B.shape
>>> Sig = linalg.diagsvd(s,M,N)  在 SVD 中构建 Sigma 矩阵

LU 分解
>>> P,L,U = linalg.lu(C)    LU 分解
```

解构稀疏矩阵

```
>>> la, v = sparse.linalg.eigs(F,1)  特征值与特征向量
>>> sparse.linalg.svds(H, 2)         奇异值分解 (SVD)
```

