

Miller's ihome

One number,one world ! One heart , one world !

Numpy中矩阵对象（matrix）

发表于 六月 5, 2013

numpy模块中的矩阵对象为numpy.matrix，包括矩阵数据的处理，矩阵的计算，以及基本的统计功能，转置，可逆性等，包括对复数的处理，均在matrix对象中。class numpy.matrix(data,dtype,copy):返回一个矩阵，其中data为ndarray对象或者字符形式；dtype:为data的type；copy:为bool类型。

```
>>> a = np.matrix('1 2 7; 3 4 8; 5 6 9')
>>> a
matrix([[1, 2, 7],
        [3, 4, 8],
        [5, 6, 9]])
#矩阵的换行必须是用分号(;)隔开，内部数据必须为字符串形式(' '),矩阵的元素之间必须以空格隔开。

>>> b=np.array([[1,5],[3,2]])
>>> x=np.matrix(b)
>>> x
matrix([[1, 5],
        [3, 2]])
#矩阵中的data可以为数组对象。
```

矩阵对象的属性：

matrix.T transpose：返回矩阵的转置矩阵
matrix.H hermitian (conjugate) transpose：返回复数矩阵的共轭元素矩阵
matrix.I inverse：返回矩阵的逆矩阵
matrix.A base array：返回矩阵基于的数组

矩阵对象的方法：

all([axis, out]) :沿给定的轴判断矩阵所有元素是否为真(非0即为真)
any([axis, out]) :沿给定轴的方向判断矩阵元素是否为真，只要一个元素为真则为真。
argmax([axis, out]) :沿给定轴的方向返回最大元素的索引（最大元素的位置）。
argmin([axis, out]) :沿给定轴的方向返回最小元素的索引（最小元素的位置）
argsort([axis, kind, order]) :返回排序后的索引矩阵
astype(dtype[, order, casting, subok, copy]) :将该矩阵数据复制，且数据类型为指定的数据类型
byteswap(inplace) Swap the bytes of the array elements
choose(choices[, out, mode]) :根据给定的索引得到一个新的数据矩阵（索引从choices给定）
clip(a_min, a_max[, out]) :返回新的矩阵，比给定元素大的元素为a_max，小的为a_min
compress(condition[, axis, out]) :返回满足条件的矩阵
conj() :返回复数的共轭复数
conjugate() :返回所有复数的共轭复数元素
copy([order]) :复制一个矩阵并赋给另外一个对象，b=a.copy()
cumprod([axis, dtype, out]) :返回沿指定轴的元素累积矩阵
cumsum([axis, dtype, out]) :返回沿指定轴的元素累积和矩阵
diagonal([offset, axis1, axis2]) :返回矩阵中对角线的数据
dot(b[, out]) :两个矩阵的点乘
dump(file) :将矩阵存储为指定文件,可以通过pickle.loads()或者numpy.loads()如:a.dump('d:\a.txt')
dumps() :将矩阵的数据转存为字符串。
fill(value) :将矩阵中的所有元素填充为指定的value
flatten([order]) :将矩阵转化为一个一维的形式，但是还是matrix对象
getA() :返回自己，但是作为ndarray返回

getA1() : 返回一个扁平（一维）的数组（ndarray）

getH() :返回自身的共轭复数转置矩阵
 getI() :返回本身的逆矩阵
 getT() :返回本身的转置矩阵
 max([axis, out]) : 返回指定轴的最大值
 mean([axis, dtype, out]) :沿给定轴方向，返回其均值
 min([axis, out]) : 返回指定轴的最小值
 nonzero() :返回非零元素的索引矩阵
 prod([axis, dtype, out]) :返回指定轴方型上，矩阵元素的乘积。
 ptp([axis, out]) :返回指定轴方向的最大值减去最小值。
 put(indices, values[, mode]) :用给定的value替换矩阵本身给定索引 (indices) 位置的值
 ravel([order]) :返回一个数组，该数组是一维数组或平数组
 repeat(repeats[, axis]) :重复矩阵中的元素，可以沿指定轴方向重复矩阵元素，repeats为重复次数
 reshape(shape[, order]) :改变矩阵的大小,如：reshape([2,3])
 resize(new_shape[, refcheck]) :改变该数据的尺寸大小
 round([decimals, out]) :返回指定精度后的矩阵，指定的位数采用四舍五入，若为1，则保留一位小数
 searchsorted(v[, side, sorter]) :搜索V在矩阵中的索引位置
 sort([axis, kind, order]) :对矩阵进行排序或者按轴的方向进行排序
 squeeze([axis]) :移除长度为1的轴
 std([axis, dtype, out, ddof]) :沿指定轴的方向，返回元素的标准差。
 sum([axis, dtype, out]) : 沿指定轴的方向，返回其元素的总和
 swapaxes(axis1, axis2):交换两个轴方向上的数据。
 take(indices[, axis, out, mode]) :提取指定索引位置的数据,并以一维数组或者矩阵返回(主要取决axis)
 tofile(fid[, sep, format]) :将矩阵中的数据以二进制写入到文件
 tolist() :将矩阵转化为列表形式
 tostring([order]):将矩阵转化为python的字符串。
 trace([offset, axis1, axis2, dtype, out]):返回对角线元素之和
 transpose(*axes) :返回矩阵的转置矩阵，不改变原有矩阵
 var([axis, dtype, out, ddof]) : 沿指定轴方向，返回矩阵元素的方差
 view([dtype, type]) :生成一个相同数据，但是类型为指定新类型的矩阵。
 ü All方法

```

>>> a = np.asmatrix('0 2 7; 3 4 8; 5 0 9')
>>> a.all()
False
>>> a.all(axis=0)
matrix([[False, False,  True]], dtype=bool)
>>> a.all(axis=1)
matrix([[False],
 [ True],
 [False]], dtype=bool)

```

ü Astype方法

```

>>> a.astype(float)
matrix([[ 12.,   3.,   5.],
 [ 32.,  23.,   9.],
 [ 10., -14.,  78.]])

```

ü Argsort方法

```

>>> a=np.matrix('12 3 5; 32 23 9; 10 -14 78')
>>> a.argsort()
matrix([[1, 2, 0],
 [2, 1, 0],
 [1, 0, 2]])

```

ü Clip方法

```

>>> a
matrix([[ 12,   3,   5],
 [ 32,  23,   9],
 [ 10, -14,  78]])
>>> a.clip(12,32)
matrix([[12, 12, 12],
 [32, 23, 12],
 [12, 12, 32]])

```

ü Cumprod方法

```

>>> a.cumprod(1)
matrix([[ 12,  36, 180],
 [ 32,  896, 8064],
 [ 10, -126, 9720]])

```

```

>>> a.cumprod(axis=1)
matrix([[ 12,    36,   180],
 [ 32,   736,  6624],
 [ 10,   -140, -10920]])

ü Cumsum方法
>>> a.cumsum(axis=1)
matrix([[12, 15, 20],
 [32, 55, 64],
 [10, -4, 74]])

ü Tolist方法
>>> b.tolist()
[[12, 3, 5], [32, 23, 9], [10, -14, 78]]

ü Tofile方法
>>> b.tofile('d:\\b.txt')

ü compress()方法
>>> from numpy import *
>>> a = array([10, 20, 30, 40])
>>> condition = (a > 15) & (a < 35)
>>> condition
array([False, True, True, False], dtype=bool)
>>> a.compress(condition)
array([20, 30])
>>> a[condition]                                # same effect
array([20, 30])
>>> compress(a >= 30, a)                        # this form a
so exists
array([30, 40])
>>> b = array([[10,20,30],[40,50,60]])
>>> b.compress(b.ravel() >= 22)
array([30, 40, 50, 60])
>>> x = array([3,1,2])
>>> y = array([50, 101])
>>> b.compress(x >= 2, axis=1)                  # illustrates
the use of the axis keyword
array([[10, 30],
 [40, 60]])
>>> b.compress(y >= 100, axis=0)
array([[40, 50, 60]])

```

0

此条目由Miller Wu发表在Python分类目录，并贴了matrix、numpy标签。将[固定链接](http://www.idataskys.com/numpy%E4%B8%AD%E7%9F%A9%E9%98%B5%E5%AF%B9%E8%B1%A1%Ef%bc%88matrix%Ef%bc%89/)
<http://www.idataskys.com/numpy%E4%B8%AD%E7%9F%A9%E9%98%B5%E5%AF%B9%E8%B1%A1%Ef%bc%88matrix%Ef%bc%89/>
 加入收藏夹。