活动代码页: 936

D:\workspace\zhtest\dm\machinelearninginaction\Ch14>

D:\workspace\zhtest\dm\machinelearninginaction\Ch14>python

Python 2.7.10 (default, May 23 2015, 09:44:00) [MSC v.1500 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license" for more information.

>>>

>>>

>>> ls

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

NameError: name 'ls' is not defined

>>> import svdRec as svd

>>>

>>> from numpy import \*

>>>

>>> u, s, v = linalg.svd([1,1],[7,7])

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

File "C:\programs\python2.7\lib\site-packages\numpy\linalg\linalg.py", line 1369, in svd

\_assertRankAtLeast2(a)

File "C:\programs\python2.7\lib\site-packages\numpy\linalg\linalg.py", line 202, in \_assertRankAtLeast2

'at least two-dimensional' % a.ndim)

numpy.linalg.linalg.LinAlgError: 1-dimensional array given. Array must be at least two-dimensional

>>> u, s, v = linalg.svd([[1,1],[7,7]])

>>>

>>>

>>> u

array([[-0.14142136, -0.98994949],

[-0.98994949, 0.14142136]])

>>> s

array([ 1.00000000e+01, 2.82797782e-16])

>>> v

array([[-0.70710678, -0.70710678],

[ 0.70710678, -0.70710678]])

>>> u \* v

array([[ 0.1, 0.7],

[-0.7, -0.1]])

>>> u \* s \* v

array([[ 1.00000000e+00, 1.97958447e-16],

[ -7.00000000e+00, -2.82797782e-17]])

>>> transpose(v)

array([[-0.70710678, 0.70710678],

[-0.70710678, -0.70710678]])

>>> u \* s \* transpose(v)

array([[ 1.00000000e+00, -1.97958447e-16],

[ 7.00000000e+00, -2.82797782e-17]])

>>> s1 = [[1.00000000e+01, 0],[0, 2.82797782e-16]]

>>>

>>> u \* s1 \* transpose(v)

array([[ 1.00000000e+00, -0.00000000e+00],

[ 0.00000000e+00, -2.82797782e-17]])

>>>

>>> s1 = [[10, 0],[0, 0.0001]]

>>> u \* s1 \* transpose(v)

array([[ 1.00000000e+00, -0.00000000e+00],

[ 0.00000000e+00, -1.00000000e-05]])

>>> s1 = [[10, 0],[0, 0]]

>>> u \* s1 \* transpose(v)

array([[ 1., -0.],

[ 0., -0.]])

>>>

>>> data = svd.loadExData()

>>> data

[[0, 0, 0, 2, 2], [0, 0, 0, 3, 3], [0, 0, 0, 1, 1], [1, 1, 1, 0, 0], [2, 2, 2, 0, 0], [5, 5, 5, 0, 0], [1, 1, 1, 0, 0]]

>>>

>>> u2, s2, v2 = linalg.svd(data)

>>>

>>> u2

array([[ 8.33888363e-17, -5.34522484e-01, -8.06954481e-01,

-1.67694476e-01, -1.51945515e-01, -1.07860771e-01,

-1.63510668e-02],

[ -5.27910020e-33, -8.01783726e-01, 5.27281973e-01,

-1.66867932e-01, 2.24670188e-01, -2.78768644e-02,

-4.22597083e-03],

[ 0.00000000e+00, -2.67261242e-01, 3.20630432e-02,

8.35992746e-01, -3.70119532e-01, 2.99352135e-01,

4.53800461e-02],

[ -1.79605302e-01, 4.71157956e-17, 2.28998849e-02,

3.75352655e-01, 1.11826091e-01, -8.91930871e-01,

-1.35211543e-01],

[ -3.59210604e-01, 6.12887329e-17, 2.13303915e-01,

-2.82658715e-01, -7.26957442e-01, -6.22837460e-02,

-4.61764925e-01],

[ -8.98026510e-01, -1.04127789e-17, -1.11231934e-01,

6.62588262e-02, 3.41113503e-01, 2.43042631e-01,

3.68438521e-02],

[ -1.79605302e-01, 3.06443665e-17, 1.06651957e-01,

-1.41329357e-01, -3.63478721e-01, -1.98714793e-01,

8.74522132e-01]])

>>> s2

array([ 9.64365076e+00, 5.29150262e+00, 6.13509323e-16,

1.36249353e-16, 1.66081800e-18])

>>> v2

array([[ -5.77350269e-01, -5.77350269e-01, -5.77350269e-01,

0.00000000e+00, 0.00000000e+00],

[ -1.70408510e-16, 8.52042552e-17, 8.52042552e-17,

-7.07106781e-01, -7.07106781e-01],

[ -6.48041655e-01, 4.30125877e-01, 2.17915777e-01,

-4.16861163e-01, 4.16861163e-01],

[ -4.71240570e-01, 3.20846381e-01, 1.50394188e-01,

5.71126743e-01, -5.71126743e-01],

[ -1.56974540e-01, -6.15399054e-01, 7.72373594e-01,

6.40426005e-03, -6.40426005e-03]])

>>> s3 = [[s2[0], 0, 0],[0, s2[1], 0],[0, 0, s2[2]]]

>>>

>>> u \* s1 \* v

array([[ 1., 0.],

[-0., -0.]])

>>>

>>> u2[:,:3]

array([[ 8.33888363e-17, -5.34522484e-01, -8.06954481e-01],

[ -5.27910020e-33, -8.01783726e-01, 5.27281973e-01],

[ 0.00000000e+00, -2.67261242e-01, 3.20630432e-02],

[ -1.79605302e-01, 4.71157956e-17, 2.28998849e-02],

[ -3.59210604e-01, 6.12887329e-17, 2.13303915e-01],

[ -8.98026510e-01, -1.04127789e-17, -1.11231934e-01],

[ -1.79605302e-01, 3.06443665e-17, 1.06651957e-01]])

>>> v2[:3,:]

array([[ -5.77350269e-01, -5.77350269e-01, -5.77350269e-01,

0.00000000e+00, 0.00000000e+00],

[ -1.70408510e-16, 8.52042552e-17, 8.52042552e-17,

-7.07106781e-01, -7.07106781e-01],

[ -6.48041655e-01, 4.30125877e-01, 2.17915777e-01,

-4.16861163e-01, 4.16861163e-01]])

>>> u2[:,:3] \* s3 \* v2[:3, :]

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

ValueError: operands could not be broadcast together with shapes (7,3) (3,3)

>>>

>>> u2[:,:3] \* s3 \* v2[:3, :]

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

ValueError: operands could not be broadcast together with shapes (7,3) (3,3)

>>> v2.t

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

AttributeError: 'numpy.ndarray' object has no attribute 't'

>>> v2.t

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

AttributeError: 'numpy.ndarray' object has no attribute 't'

>>>

>>> numpy.matrix(u2[:,:3]) \* numpy.matrix(s3) \* numpy.matrix(v2[:3, :])

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

NameError: name 'numpy' is not defined

>>> import numpy

>>>

>>> numpy.matrix(u2[:,:3]) \* numpy.matrix(s3) \* numpy.matrix(v2[:3, :])

matrix([[ 3.38527299e-16, -9.18227598e-16, -8.13167874e-16,

2.00000000e+00, 2.00000000e+00],

[ 5.13345525e-16, -2.22348585e-16, -2.90996941e-16,

3.00000000e+00, 3.00000000e+00],

[ 2.28246415e-16, -1.12036017e-16, -1.16210397e-16,

1.00000000e+00, 1.00000000e+00],

[ 1.00000000e+00, 1.00000000e+00, 1.00000000e+00,

-1.82147769e-16, -1.70434560e-16],

[ 2.00000000e+00, 2.00000000e+00, 2.00000000e+00,

-2.83873535e-16, -1.74769346e-16],

[ 5.00000000e+00, 5.00000000e+00, 5.00000000e+00,

6.74084192e-17, 1.05136830e-17],

[ 1.00000000e+00, 1.00000000e+00, 1.00000000e+00,

-1.41936767e-16, -8.73846730e-17]])

>>> numpy.matrix(u2[:,:3]) \* numpy.matrix(s3) \* numpy.matrix(v2[:3, :])

matrix([[ 3.38527299e-16, -9.18227598e-16, -8.13167874e-16,

2.00000000e+00, 2.00000000e+00],

[ 5.13345525e-16, -2.22348585e-16, -2.90996941e-16,

3.00000000e+00, 3.00000000e+00],

[ 2.28246415e-16, -1.12036017e-16, -1.16210397e-16,

1.00000000e+00, 1.00000000e+00],

[ 1.00000000e+00, 1.00000000e+00, 1.00000000e+00,

-1.82147769e-16, -1.70434560e-16],

[ 2.00000000e+00, 2.00000000e+00, 2.00000000e+00,

-2.83873535e-16, -1.74769346e-16],

[ 5.00000000e+00, 5.00000000e+00, 5.00000000e+00,

6.74084192e-17, 1.05136830e-17],

[ 1.00000000e+00, 1.00000000e+00, 1.00000000e+00,

-1.41936767e-16, -8.73846730e-17]])

>>>

>>> svd.ecludSim(data[:,3] , data[:,4])

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

TypeError: list indices must be integers, not tuple

>>> data[:,3]

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

TypeError: list indices must be integers, not tuple

>>> data[:,3]

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

TypeError: list indices must be integers, not tuple

>>> mymat = numpy.matrix(data)

>>> mymat

matrix([[0, 0, 0, 2, 2],

[0, 0, 0, 3, 3],

[0, 0, 0, 1, 1],

[1, 1, 1, 0, 0],

[2, 2, 2, 0, 0],

[5, 5, 5, 0, 0],

[1, 1, 1, 0, 0]])

>>> mymat[:,3]

matrix([[2],

[3],

[1],

[0],

[0],

[0],

[0]])

>>> svd.ecludSim(mymat[:,3] , mymat[:,4])

1.0

>>> svd.ecludSim(mymat[:,3] , mymat[:,1])

0.12973190755680383

>>> svd.cosSim(mymat[:,3] , mymat[:,1])

0.5

>>> svd.cosSim(mymat[:,3] , mymat[:,4])

1.0

>>> svd.pearsSim(mymat[:,3] , mymat[:,4])

0.99999999999999989

>>> svd.pearsSim(mymat[3,:] , mymat[6,:])

1.0

>>> svd.pearsSim(mymat[4,:] , mymat[6,:])

1.0

>>> svd.pearsSim(mymat[0,:] , mymat[6,:])

1.0

>>> svd.cosSim(mymat[0,:] , mymat[6,:])

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

File "svdRec.py", line 39, in cosSim

num = float(inA.T\*inB)

TypeError: only length-1 arrays can be converted to Python scalars

>>> mymat[0,:]

matrix([[0, 0, 0, 2, 2]])

>>> la.norm(mymat[0,:])

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

NameError: name 'la' is not defined

>>> linalg.norm(mymat[0,:])

2.8284271247461903

>>> svd.pearsSim(mymat[4,:] , mymat[6,:])

1.0

>>>

>>> svd.ecludSim(mymat[4,:] , mymat[6,:])

0.36602540378443865

>>> svd.cosSim(mymat[0,:] , mymat[6,:])

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

File "svdRec.py", line 39, in cosSim

num = float(inA.T\*inB)

TypeError: only length-1 arrays can be converted to Python scalars

>>>

>>> nonzero(1)

(array([0], dtype=int64),)

>>> nonzero('a')

(array([0], dtype=int64),)

>>> data.ndim

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

AttributeError: 'list' object has no attribute 'ndim'

>>> array(data).ndim

2

>>> data

[[0, 0, 0, 2, 2], [0, 0, 0, 3, 3], [0, 0, 0, 1, 1], [1, 1, 1, 0, 0], [2, 2, 2, 0, 0], [5, 5, 5, 0, 0], [1, 1, 1, 0, 0]]

>>> data(np.nonzero(data))

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

NameError: name 'np' is not defined

>>> data(nonzero(data))

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

TypeError: 'list' object is not callable

>>>

>>> data(numpy.nonzero(data))

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

TypeError: 'list' object is not callable

>>>

>>>

>>> data[:,1].A

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

TypeError: list indices must be integers, not tuple

>>> data[:,1]

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

TypeError: list indices must be integers, not tuple

>>> mymat[:,1]

matrix([[0],

[0],

[0],

[1],

[2],

[5],

[1]])

>>> mymat[:,1].A

array([[0],

[0],

[0],

[1],

[2],

[5],

[1]])

>>> mymat.ndim

2

>>> mymat.shape

(7L, 5L)

>>> mymat.size

35

>>> ar = np.arange(3\*4\*5).reshape(3,4,5)

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

NameError: name 'np' is not defined

>>> ar = np.arange(3\*4\*5).reshape(3,4,5)

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

NameError: name 'np' is not defined

>>> import numpy as np

>>> ar = np.arange(3\*4\*5).reshape(3,4,5)

>>>

>>> ar

array([[[ 0, 1, 2, 3, 4],

[ 5, 6, 7, 8, 9],

[10, 11, 12, 13, 14],

[15, 16, 17, 18, 19]],

[[20, 21, 22, 23, 24],

[25, 26, 27, 28, 29],

[30, 31, 32, 33, 34],

[35, 36, 37, 38, 39]],

[[40, 41, 42, 43, 44],

[45, 46, 47, 48, 49],

[50, 51, 52, 53, 54],

[55, 56, 57, 58, 59]]])

>>> b2 = np.array([[True, False, True], [True, False, False]])

>>> b2

array([[ True, False, True],

[ True, False, False]], dtype=bool)

>>> nonzero(b2)

(array([0, 0, 1], dtype=int64), array([0, 2, 0], dtype=int64))

>>> ar[b2]

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

IndexError: boolean index did not match indexed array along dimension 0; dimension is 3 but corresponding boolean dimension is 2

>>> a[nonzero(b2)]

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

NameError: name 'a' is not defined

>>> ar[nonzero(b2)]

array([[ 0, 1, 2, 3, 4],

[10, 11, 12, 13, 14],

[20, 21, 22, 23, 24]])

>>> ar[np.matrix(b2)]

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

IndexError: boolean index did not match indexed array along dimension 0; dimension is 3 but corresponding boolean dimension is 2

>>> ar[b2]

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

IndexError: boolean index did not match indexed array along dimension 0; dimension is 3 but corresponding boolean dimension is 2

>>>

>>> ar[nonzero(b2)[0], nonzero(b2)[1]]

array([[ 0, 1, 2, 3, 4],

[10, 11, 12, 13, 14],

[20, 21, 22, 23, 24]])

>>> ar[1:3, nonzero(b2)[0], nonzero(b2)[1]]

array([[20, 22, 25],

[40, 42, 45]])

>>> ar[0:3, nonzero(b2)[0], nonzero(b2)[1]]

array([[ 0, 2, 5],

[20, 22, 25],

[40, 42, 45]])

>>> mymat[:,1].A

array([[0],

[0],

[0],

[1],

[2],

[5],

[1]])

>>> mymat[:,1].A > 0

array([[False],

[False],

[False],

[ True],

[ True],

[ True],

[ True]], dtype=bool)

>>> overlap = nonzero(logical\_and(mymat[:,1].A > 0, mymat[:, 3].A > 0))

>>> overlap

(array([], dtype=int64), array([], dtype=int64))

>>> logical\_and(mymat[:,1].A > 0, mymat[:, 3].A > 0)

array([[False],

[False],

[False],

[False],

[False],

[False],

[False]], dtype=bool)

>>> len(overlap)

2

>>> ol = nonzero(logical\_and(mymat[:,1].A > 0, mymat[:, 2].A > 0))

>>> ol

(array([3, 4, 5, 6], dtype=int64), array([0, 0, 0, 0], dtype=int64))

>>> len(ol)

2

>>> mymat[ol,1]

matrix([[1, 2, 5, 1],

[0, 0, 0, 0]])

>>> mymat[ol,3]

matrix([[0, 0, 0, 0],

[2, 2, 2, 2]])

>>> mymat[ol[0],3]

matrix([[0],

[0],

[0],

[0]])

>>> ol

(array([3, 4, 5, 6], dtype=int64), array([0, 0, 0, 0], dtype=int64))

>>> ol[0]

array([3, 4, 5, 6], dtype=int64)

>>> mymat[ol[0],1]

matrix([[1],

[2],

[5],

[1]])

>>> svd.recommend(mymat, 1)

the 0 and 3 similarity is: 0.000000

the 0 and 4 similarity is: 0.000000

the 1 and 3 similarity is: 0.000000

the 1 and 4 similarity is: 0.000000

the 2 and 3 similarity is: 0.000000

the 2 and 4 similarity is: 0.000000

[(0, 0), (1, 0), (2, 0)]

>>> mymat[0,1] = mymat[0,0] = mymat[1,0] = mymat[2,0] = 4

>>> mymat[3,3] = 2

>>> mymat

matrix([[4, 4, 0, 2, 2],

[4, 0, 0, 3, 3],

[4, 0, 0, 1, 1],

[1, 1, 1, 2, 0],

[2, 2, 2, 0, 0],

[5, 5, 5, 0, 0],

[1, 1, 1, 0, 0]])

>>> svd.recommend(mymat, 1)

the 1 and 0 similarity is: 1.000000

the 1 and 3 similarity is: 0.928746

the 1 and 4 similarity is: 1.000000

the 2 and 0 similarity is: 1.000000

the 2 and 3 similarity is: 1.000000

the 2 and 4 similarity is: 0.000000

[(2, 3.5), (1, 3.3414430073352088)]

>>> svd.recommend(mymat, 1, 5)

the 1 and 0 similarity is: 1.000000

the 1 and 3 similarity is: 0.928746

the 1 and 4 similarity is: 1.000000

the 2 and 0 similarity is: 1.000000

the 2 and 3 similarity is: 1.000000

the 2 and 4 similarity is: 0.000000

[(2, 3.5), (1, 3.3414430073352088)]

>>> svd.recommend(mymat, 1, N=5)

the 1 and 0 similarity is: 1.000000

the 1 and 3 similarity is: 0.928746

the 1 and 4 similarity is: 1.000000

the 2 and 0 similarity is: 1.000000

the 2 and 3 similarity is: 1.000000

the 2 and 4 similarity is: 0.000000

[(2, 3.5), (1, 3.3414430073352088)]

>>>

>>> mymat[1,:].A == 0

array([[False, True, True, False, False]], dtype=bool)

>>> mymat[1,:] == 0

matrix([[False, True, True, False, False]], dtype=bool)

>>> nonzero(mymat[1,:].A == 0)

(array([0, 0], dtype=int64), array([1, 2], dtype=int64))

>>> nonzero(mymat[1,:].A == 0)[1]

array([1, 2], dtype=int64)

>>>

>>> mymat[1, nonzero(mymat[1,:].A == 0)[1]]

matrix([[0, 0]])

>>> svd.standEst(mymat, 1, cosSim, 1)

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

NameError: name 'cosSim' is not defined

>>> svd.standEst(mymat, 1, svd.cosSim, 1)

the 1 and 0 similarity is: 1.000000

the 1 and 3 similarity is: 0.928746

the 1 and 4 similarity is: 1.000000

3.3414430073352088

>>> sorted(mymat[1,:])

[matrix([[4, 0, 0, 3, 3]])]

>>> sorted(mymat[1,:], True)

[matrix([[4, 0, 0, 3, 3]])]

>>> sorted(mymat[1,:][0], True)

[matrix([[4, 0, 0, 3, 3]])]

>>> sorted(data[1,:])

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

TypeError: list indices must be integers, not tuple

>>> sorted(data[1][:])

[0, 0, 0, 3, 3]

>>>

>>> sorted(data[2][:])

[0, 0, 0, 1, 1]

>>> sorted(data[:][4])

[0, 0, 2, 2, 2]

>>> sorted(data[:][3])

[0, 0, 1, 1, 1]

>>> sorted(data[0][3])

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

TypeError: 'int' object is not iterable

>>> sorted(data[0][3])

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

TypeError: 'int' object is not iterable

>>>

>>>

>>> data

[[0, 0, 0, 2, 2], [0, 0, 0, 3, 3], [0, 0, 0, 1, 1], [1, 1, 1, 0, 0], [2, 2, 2, 0, 0], [5, 5, 5, 0, 0], [1, 1, 1, 0, 0]]

>>> lambda jj: jj[1]

<function <lambda> at 0x00000000036605F8>

>>> aa = lambda jj: jj[1]

>>> aa([1,2,3])

2

>>> svd.recommend(mymat, 1, N=4)

the 1 and 0 similarity is: 1.000000

the 1 and 3 similarity is: 0.928746

the 1 and 4 similarity is: 1.000000

the 2 and 0 similarity is: 1.000000

the 2 and 3 similarity is: 1.000000

the 2 and 4 similarity is: 0.000000

[(2, 3.5), (1, 3.3414430073352088)]

>>> svd.recommend(mymat, 2, N=4)

the 1 and 0 similarity is: 1.000000

the 1 and 3 similarity is: 0.928746

the 1 and 4 similarity is: 1.000000

the 2 and 0 similarity is: 1.000000

the 2 and 3 similarity is: 1.000000

the 2 and 4 similarity is: 0.000000

[(2, 2.5), (1, 2.0243290220056256)]

>>> svd.recommend(mymat, 3, N=4)

the 4 and 0 similarity is: 0.962910

the 4 and 1 similarity is: 1.000000

the 4 and 2 similarity is: 0.000000

the 4 and 3 similarity is: 1.000000

[(4, 1.3375060272377768)]

>>>

>>> ol2 = nonzero(logical\_and(mymat[:,1].A > 0, mymat[:, 2].A > 0))

>>> ol2

(array([3, 4, 5, 6], dtype=int64), array([0, 0, 0, 0], dtype=int64))

>>> mymat

matrix([[4, 4, 0, 2, 2],

[4, 0, 0, 3, 3],

[4, 0, 0, 1, 1],

[1, 1, 1, 2, 0],

[2, 2, 2, 0, 0],

[5, 5, 5, 0, 0],

[1, 1, 1, 0, 0]])

>>> data2 = svd.loadExData2()

>>> data2

[[0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 5], [0, 0, 0, 3, 0, 4, 0, 0, 0, 0, 3], [0, 0, 0, 0, 4, 0, 0, 1, 0, 4, 0], [3, 3, 4, 0, 0, 0, 0, 2, 2, 0, 0], [5, 4, 5, 0, 0, 0, 0, 5, 5, 0, 0], [0, 0, 0, 0, 5, 0, 1, 0, 0, 5, 0], [4, 3, 4, 0, 0, 0, 0, 5, 5, 0, 1], [0, 0, 0, 4, 0, 4, 0, 0, 0, 0, 4], [0, 0, 0, 2, 0, 2, 5, 0, 0, 1, 2], [0, 0, 0, 0, 5, 0, 0, 0, 0, 4, 0], [1, 0, 0, 0, 0, 0, 0, 1, 2, 0, 0]]

>>> mymat2 = np.matrix(data2)

>>> mymat2

matrix([[0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 5],

[0, 0, 0, 3, 0, 4, 0, 0, 0, 0, 3],

[0, 0, 0, 0, 4, 0, 0, 1, 0, 4, 0],

[3, 3, 4, 0, 0, 0, 0, 2, 2, 0, 0],

[5, 4, 5, 0, 0, 0, 0, 5, 5, 0, 0],

[0, 0, 0, 0, 5, 0, 1, 0, 0, 5, 0],

[4, 3, 4, 0, 0, 0, 0, 5, 5, 0, 1],

[0, 0, 0, 4, 0, 4, 0, 0, 0, 0, 4],

[0, 0, 0, 2, 0, 2, 5, 0, 0, 1, 2],

[0, 0, 0, 0, 5, 0, 0, 0, 0, 4, 0],

[1, 0, 0, 0, 0, 0, 0, 1, 2, 0, 0]])

>>> u4, s4, v4 = la.svd(mymat2)

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

NameError: name 'la' is not defined

>>> u4, s4, v4 = la.svd(mymat2)

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

NameError: name 'la' is not defined

>>> from numpy import linalg as la

>>>

>>> u4, s4, v4 = la.svd(mymat2)

>>> u4

matrix([[-0.02173672, -0.41043862, -0.29555566, -0.31993924, 0.77606298,

0.15828951, -0.0435284 , 0.02059981, -0.02594897, 0.10460968,

0.00940588],

[-0.01664767, -0.40868796, -0.29149768, -0.15138979, -0.33933823,

-0.02422168, -0.14675961, 0.578175 , 0.19363235, -0.46295592,

-0.03773245],

[-0.03763173, -0.27302481, 0.4269746 , -0.10978297, -0.0155641 ,

-0.1131373 , 0.57177959, 0.33872886, -0.13586013, 0.17290962,

0.4760405 ],

[-0.3928286 , 0.03215633, 0.00283676, 0.02145489, -0.11989979,

0.7023584 , -0.17084214, -0.12090699, -0.21988993, -0.19766208,

0.45307019],

[-0.68146521, 0.05125169, 0.00962441, 0.02067521, -0.06539204,

0.02800613, -0.09643557, 0.34249592, 0.09728232, 0.53792639,

-0.31932109],

[-0.01031581, -0.35826614, 0.52908411, 0.06256753, 0.03236161,

0.05503535, -0.18939118, -0.01918345, -0.5197144 , -0.20532508,

-0.48489136],

[-0.60364271, -0.00222591, -0.02262313, -0.02723249, 0.13686857,

-0.37576272, 0.31226741, -0.33309364, 0.09742059, -0.50425492,

-0.04690811],

[-0.02078959, -0.4841342 , -0.34503998, -0.16062914, -0.47045568,

-0.07723516, 0.11601606, -0.47642592, -0.209654 , 0.32897717,

-0.0202877 ],

[-0.01290907, -0.35922701, -0.12620599, 0.9008227 , 0.13205031,

-0.00140471, 0.03068934, 0.00331704, 0.11759822, 0.04143523,

0.09716744],

[-0.00900549, -0.30733798, 0.47941858, -0.14085095, -0.04530678,

0.04665223, -0.30466862, -0.28031423, 0.6814256 , 0.06957399,

0.10474517],

[-0.11812788, 0.00805012, 0.00186006, -0.00117811, 0.05552191,

-0.56133398, -0.60872297, 0.0417376 , -0.289553 , 0.07197909,

0.45441173]])

>>> s4

array([ 15.77075346, 11.40670395, 11.03044558, 4.84639758,

3.09292055, 2.58097379, 1.00413543, 0.72817072,

0.43800353, 0.22082113, 0.07367823])

>>> sig2 = s4\*\*2

>>> sig2

array([ 2.48716665e+02, 1.30112895e+02, 1.21670730e+02,

2.34875695e+01, 9.56615756e+00, 6.66142570e+00,

1.00828796e+00, 5.30232598e-01, 1.91847092e-01,

4.87619735e-02, 5.42848136e-03])

>>> sum(sig2)

541.99999999999955

>>> sum(sig2[:2])

378.82955951135784

>>> sum(sig2[:2])/sum(sig2)

0.69894752677372352

>>> sum(sig2[:3])/sum(sig2)

0.92343226776306209

>>> sum(sig2[:4])/sum(sig2)

0.96676726685929748

>>> sum(sig2[:4])/sum(sig2)

0.96676726685929748

>>>

>>> eye(4)

array([[ 1., 0., 0., 0.],

[ 0., 1., 0., 0.],

[ 0., 0., 1., 0.],

[ 0., 0., 0., 1.]])

>>> eye(4)\*s4[:4]

array([[ 15.77075346, 0. , 0. , 0. ],

[ 0. , 11.40670395, 0. , 0. ],

[ 0. , 0. , 11.03044558, 0. ],

[ 0. , 0. , 0. , 4.84639758]])

>>> sig4 = eye(4)\*s4[:4]

>>> sig4.I

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

AttributeError: 'numpy.ndarray' object has no attribute 'I'

>>> sig4 = mat(eye(4)\*s4[:4])

>>> sig4.I

matrix([[ 0.06340851, 0. , 0. , 0. ],

[ 0. , 0.08766774, 0. , 0. ],

[ 0. , 0. , 0.09065817, 0. ],

[ 0. , 0. , 0. , 0.20633883]])

>>> sig4.I \* sig4

matrix([[ 1., 0., 0., 0.],

[ 0., 1., 0., 0.],

[ 0., 0., 1., 0.],

[ 0., 0., 0., 1.]])

>>> xformedItems = mymat2.T \* u4[:,:4] \* sig2.I

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

AttributeError: 'numpy.ndarray' object has no attribute 'I'

>>> xformedItems = mymat2.T \* u4[:,:4] \* sig4.I

>>>

>>> xformedItems

matrix([[-0.45137416, 0.03084799, -0.00290108, 0.01189185],

[-0.36239706, 0.02584428, -0.00189127, 0.01348796],

[-0.46879252, 0.03296133, -0.00281253, 0.01656192],

[-0.01007685, -0.34024331, -0.22728592, 0.14546051],

[-0.01567036, -0.38750193, 0.61197998, -0.17137451],

[-0.01664563, -0.52000097, -0.3608907 , -0.14984063],

[-0.00474684, -0.18887149, -0.00924222, 0.94228361],

[-0.46712774, 0.00389831, 0.03349951, -0.02080674],

[-0.47223188, 0.02853952, -0.00504059, 0.00160266],

[-0.01591788, -0.39205093, 0.55707516, 0.04356321],

[-0.0552444 , -0.52034959, -0.36330956, -0.19023805]])

>>> svd.recommend(mymat2, 1, N=4)

the 0 and 3 similarity is: 0.000000

the 0 and 5 similarity is: 0.000000

the 0 and 10 similarity is: 1.000000

the 1 and 3 similarity is: 0.000000

the 1 and 5 similarity is: 0.000000

the 1 and 10 similarity is: 1.000000

the 2 and 3 similarity is: 0.000000

the 2 and 5 similarity is: 0.000000

the 2 and 10 similarity is: 1.000000

the 4 and 3 similarity is: 0.000000

the 4 and 5 similarity is: 0.000000

the 4 and 10 similarity is: 0.000000

the 6 and 3 similarity is: 1.000000

the 6 and 5 similarity is: 1.000000

the 6 and 10 similarity is: 1.000000

the 7 and 3 similarity is: 0.000000

the 7 and 5 similarity is: 0.000000

the 7 and 10 similarity is: 1.000000

the 8 and 3 similarity is: 0.000000

the 8 and 5 similarity is: 0.000000

the 8 and 10 similarity is: 1.000000

the 9 and 3 similarity is: 1.000000

the 9 and 5 similarity is: 1.000000

the 9 and 10 similarity is: 1.000000

[(6, 3.3333333333333335), (9, 3.3333333333333335), (0, 3.0), (1, 3.0)]

>>>

>>> svd.recommend(mymat2, 1, N=8)

the 0 and 3 similarity is: 0.000000

the 0 and 5 similarity is: 0.000000

the 0 and 10 similarity is: 1.000000

the 1 and 3 similarity is: 0.000000

the 1 and 5 similarity is: 0.000000

the 1 and 10 similarity is: 1.000000

the 2 and 3 similarity is: 0.000000

the 2 and 5 similarity is: 0.000000

the 2 and 10 similarity is: 1.000000

the 4 and 3 similarity is: 0.000000

the 4 and 5 similarity is: 0.000000

the 4 and 10 similarity is: 0.000000

the 6 and 3 similarity is: 1.000000

the 6 and 5 similarity is: 1.000000

the 6 and 10 similarity is: 1.000000

the 7 and 3 similarity is: 0.000000

the 7 and 5 similarity is: 0.000000

the 7 and 10 similarity is: 1.000000

the 8 and 3 similarity is: 0.000000

the 8 and 5 similarity is: 0.000000

the 8 and 10 similarity is: 1.000000

the 9 and 3 similarity is: 1.000000

the 9 and 5 similarity is: 1.000000

the 9 and 10 similarity is: 1.000000

[(6, 3.3333333333333335), (9, 3.3333333333333335), (0, 3.0), (1, 3.0), (2, 3.0), (7, 3.0), (8, 3.0), (4, 0)]

>>> svd.recommend(mymat2, 1, N=8, svd.svdEst)

File "<stdin>", line 1

SyntaxError: non-keyword arg after keyword arg

>>> svd.recommend(mymat2, 1, N=8, svd.cosSim, svd.svdEst)

File "<stdin>", line 1

SyntaxError: non-keyword arg after keyword arg

>>> svd.recommend(mymat2, 1, N=8, simMeas = svd.cosSim, estMethod = svd.svdEst)

the 0 and 3 similarity is: 0.490950

the 0 and 5 similarity is: 0.484274

the 0 and 10 similarity is: 0.512755

the 1 and 3 similarity is: 0.491294

the 1 and 5 similarity is: 0.481516

the 1 and 10 similarity is: 0.509709

the 2 and 3 similarity is: 0.491573

the 2 and 5 similarity is: 0.482346

the 2 and 10 similarity is: 0.510584

the 4 and 3 similarity is: 0.450495

the 4 and 5 similarity is: 0.506795

the 4 and 10 similarity is: 0.512896

the 6 and 3 similarity is: 0.743699

the 6 and 5 similarity is: 0.468366

the 6 and 10 similarity is: 0.439465

the 7 and 3 similarity is: 0.482175

the 7 and 5 similarity is: 0.494716

the 7 and 10 similarity is: 0.524970

the 8 and 3 similarity is: 0.491307

the 8 and 5 similarity is: 0.491228

the 8 and 10 similarity is: 0.520290

the 9 and 3 similarity is: 0.522379

the 9 and 5 similarity is: 0.496130

the 9 and 10 similarity is: 0.493617

[(4, 3.3447149384692283), (7, 3.3294020724526971), (9, 3.3281008763900695), (8, 3.3268696652631058), (0, 3.3254572893474776), (2, 3.3249209939900819), (1, 3.3247959071304263), (6, 3.2835952404299045)]

>>> svd.recommend(mymat2, 1, N=8, simMeas = svd.pearsSim, estMethod = svd.svdEst)

the 0 and 3 similarity is: 0.341942

the 0 and 5 similarity is: 0.124132

the 0 and 10 similarity is: 0.116698

the 1 and 3 similarity is: 0.345560

the 1 and 5 similarity is: 0.126456

the 1 and 10 similarity is: 0.118892

the 2 and 3 similarity is: 0.345149

the 2 and 5 similarity is: 0.126190

the 2 and 10 similarity is: 0.118640

the 4 and 3 similarity is: 0.450126

the 4 and 5 similarity is: 0.528504

the 4 and 10 similarity is: 0.544647

the 6 and 3 similarity is: 0.923822

the 6 and 5 similarity is: 0.724840

the 6 and 10 similarity is: 0.710896

the 7 and 3 similarity is: 0.319482

the 7 and 5 similarity is: 0.118324

the 7 and 10 similarity is: 0.113370

the 8 and 3 similarity is: 0.334910

the 8 and 5 similarity is: 0.119673

the 8 and 10 similarity is: 0.112497

the 9 and 3 similarity is: 0.566918

the 9 and 5 similarity is: 0.590049

the 9 and 10 similarity is: 0.602380

[(4, 3.3469521867021736), (9, 3.3353796573274703), (6, 3.3071930278130375), (7, 3.2146760877396727), (1, 3.2140035252044821), (2, 3.2138893399077335), (0, 3.2130026962675946), (8, 3.2110329803036919)]

>>> svd.recommend(mymat2, 1, N=8, simMeas = svd.ecludSim, estMethod = svd.svdEst)

the 0 and 3 similarity is: 0.612384

the 0 and 5 similarity is: 0.554262

the 0 and 10 similarity is: 0.557208

the 1 and 3 similarity is: 0.636421

the 1 and 5 similarity is: 0.569152

the 1 and 10 similarity is: 0.570929

the 2 and 3 similarity is: 0.607708

the 2 and 5 similarity is: 0.550613

the 2 and 10 similarity is: 0.553660

the 4 and 3 similarity is: 0.526776

the 4 and 5 similarity is: 0.504518

the 4 and 10 similarity is: 0.503709

the 6 and 3 similarity is: 0.543511

the 6 and 5 similarity is: 0.455741

the 6 and 10 similarity is: 0.447819

the 7 and 3 similarity is: 0.605924

the 7 and 5 similarity is: 0.553717

the 7 and 10 similarity is: 0.557372

the 8 and 3 similarity is: 0.606861

the 8 and 5 similarity is: 0.552125

the 8 and 10 similarity is: 0.555494

the 9 and 3 similarity is: 0.557828

the 9 and 5 similarity is: 0.513662

the 9 and 10 similarity is: 0.510451

[(4, 3.3286756747000443), (9, 3.3247038080937834), (7, 3.3224884985810177), (8, 3.3220363533819586), (2, 3.3216232888873254), (0, 3.3215250530500051), (1, 3.3203779611397621), (6, 3.3149402667167074)]

>>> print 0

0

>>> svd.imgCompress()

\*\*\*\*original matrix\*\*\*\*\*\*

0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0

0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0

0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0

0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0

0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0

0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0

0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0

0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0

0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0

0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0

0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0

0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0

0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0

0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0

0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0

0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0

0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0

0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0

\*\*\*\*reconstructed matrix using 3 singular values\*\*\*\*\*\*

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0

0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0

0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0

0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0

0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0

0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0

0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0

0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0

0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0

0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0

0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0