Me?todo de potencias

May 5, 2014

In [2]:
```
import numpy as np
from numpy import linalg
```

In [3]:
```
M = np.array([2., 1., 1. , 1.]).reshape(2, 2)
M
```

Out[3]:
```
array([[ 2.,  1.],
       [ 1.,  1.]])
```

In [4]:
```
v = np.array([1, 1.])
```

In [5]:
```
v
```

Out[5]:
```
array([ 1.,  1.])
```

In [6]:
```
np.dot(M, v)
```

Out[6]:
```
array([ 3.,  2.])
```

In [7]:
```
M.shape
```

Out[7]:
```
(2, 2)
```

In [8]:
```
type(_)
```

Out[8]:
```
tuple
```

In [9]:
```
def metodo_de_potencias(M):

    L = M.shape[0]

    v = np.ones(L)
    v_vieja = np.array(np.zeros(L))

    eps = 1e-15

    while linalg.norm(v - v_vieja) > eps:
        v_vieja = v
        v = np.dot(M, v)
        v /= linalg.norm(v)

        #print v

    return v
```

In [10]:
```
metodo_de_potencias(M)
```

```
Out[10]: array([ 0.85065081,  0.52573111])

In [12]: M = np.diag([1., 2, 3])
         M

Out[12]: array([[ 1.,  0.,  0.],
                [ 0.,  2.,  0.],
                [ 0.,  0.,  3.]])

In [13]: metodo_de_potencias(M)

Out[13]: array([  8.35246383e-41,   1.61560147e-15,   1.00000000e+00])

In [14]: from numpy import random

In [17]: M = random.rand(100, 100)
         M = M + M.T

In [19]: v = metodo_de_potencias(M)

In [20]: np.dot(M, v) / v

Out[20]: array([ 100.85139224,  100.85139224,  100.85139224,  100.85139224,
                 100.85139224,  100.85139224,  100.85139224,  100.85139224,
                 100.85139224,  100.85139224,  100.85139224,  100.85139224,
                 100.85139224,  100.85139224,  100.85139224,  100.85139224,
                 100.85139224,  100.85139224,  100.85139224,  100.85139224,
                 100.85139224,  100.85139224,  100.85139224,  100.85139224,
                 100.85139224,  100.85139224,  100.85139224,  100.85139224,
                 100.85139224,  100.85139224,  100.85139224,  100.85139224,
                 100.85139224,  100.85139224,  100.85139224,  100.85139224,
                 100.85139224,  100.85139224,  100.85139224,  100.85139224,
                 100.85139224,  100.85139224,  100.85139224,  100.85139224,
                 100.85139224,  100.85139224,  100.85139224,  100.85139224,
                 100.85139224,  100.85139224,  100.85139224,  100.85139224,
                 100.85139224,  100.85139224,  100.85139224,  100.85139224,
                 100.85139224,  100.85139224,  100.85139224,  100.85139224,
                 100.85139224,  100.85139224,  100.85139224,  100.85139224,
                 100.85139224,  100.85139224,  100.85139224,  100.85139224,
                 100.85139224,  100.85139224,  100.85139224,  100.85139224,
                 100.85139224,  100.85139224,  100.85139224,  100.85139224,
                 100.85139224,  100.85139224,  100.85139224,  100.85139224,
                 100.85139224,  100.85139224,  100.85139224,  100.85139224,
                 100.85139224,  100.85139224,  100.85139224,  100.85139224,
                 100.85139224,  100.85139224,  100.85139224,  100.85139224,
                 100.85139224,  100.85139224,  100.85139224,  100.85139224,
                 100.85139224,  100.85139224,  100.85139224,  100.85139224])

In []:

In []:

In []:
```