C2_황재훈

Jaehun Hwang

1/6/2021

P12.Method1 - eigen-decomposition

$xP = x \Rightarrow P^t x^t = x^t \Rightarrow 1 \cdot x^t$

```
import numpy as np

P = np.array([.7, .5, .3, .5]).reshape((2,2), order="F")
values, vectors = np.linalg.eig(P.transpose())

print(values)
print(vectors)

x_1 = vectors[:,0] # get first column
v = x_1 / sum(x_1)

print(v)
```

```
## [ 1.   0.2]
## [[ 0.85749293 -0.70710678]
##  [ 0.51449576  0.70710678]]
## [ 0.625  0.375]
```

P15.Method2 - system of linear equation

```
import numpy as np

P = np.array([.7, .5, .3, .5]).reshape((2,2), order="F")
n = P.shape[0]
I = np.eye(n)

A = np.c_[P-I, np.array([1] * n)]
b = np.array([0] * n + [1]).reshape((1, n+1))
v = np.linalg.solve(A.dot(A.transpose()), A.dot(b.transpose()))

print(v)
```

```
## [[ 0.625]
##  [ 0.375]]
```

P17.cont.

```
import numpy as np

P = np.array([.7, .5, .3, .5]).reshape((2,2), order="F")
```

```
for i, power in enumerate([1,2,3,4,20]):
    # print("<", i+1, "th w/", power, ">")
    print(np.linalg.matrix_power(P, power))
```

```
## [[ 0.7  0.3]
##  [ 0.5  0.5]]
## [[ 0.64  0.36]
##  [ 0.6   0.4 ]]
## [[ 0.628  0.372]
##  [ 0.62   0.38 ]]
## [[ 0.6256  0.3744]
##  [ 0.624   0.376 ]]
## [[ 0.625  0.375]
##  [ 0.625  0.375]]
```

P19.Limiting probabilities

```
import numpy as np

P = np.array([0, 1, 1, 0]).reshape((2,2), order="F")

for i, power in enumerate([2,3,4,5]):
    # print("<", i+1, "th w/", power, ">")
    print(np.linalg.matrix_power(P, power))
```

```
## [[1 0]
##  [0 1]]
## [[0 1]
##  [1 0]]
## [[1 0]
##  [0 1]]
## [[0 1]
##  [1 0]]
```

End of file