C2_Exercises

Kwon do yun

2021-01-06

Method 1 - eigen-decomposition (p. 12)

```
import numpy as np

P = np.matrix([[0.7,0.3],
              [0.5,0.5]])

values , vectors = np.linalg.eig(P.T)

print(values)
```

```
## [1.  0.2]
```

```
print(vectors)
```

```
## [[ 0.85749293 -0.70710678]
##  [ 0.51449576  0.70710678]]
```

```
x_1 = vectors[:,0]
v = x_1/sum(x_1)
print("The stationary distribution")
```

```
## The stationary distribution
```

```
print(v)
```

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

p. 15

```
import numpy as np
P = np.matrix([[0.7,0.3],
              [0.5,0.5]])
n = np.size(P,0)
I = np.eye(n)
A = np.c_[P-I,np.repeat(1,n)]
b = np.append(np.repeat(0,n),np.array([1]))

print(A)
```

```
## [[-0.3  0.3  1. ]
##  [ 0.5 -0.5  1. ]]
```

```
print(b)
```

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

```
v = np.dot(np.linalg.inv(np.dot(A,A.T)),np.dot(A,b.T).T)

print(v)
```

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

Motivation (p. 17)

```
P = np.matrix([[0.7,0.3],
              [0.5,0.5]])
```

```
np.dot(P,P) #matrix multiplication
```

```
## matrix([[0.64, 0.36],
##         [0.6 , 0.4 ]])
```

```
P**3
```

```
## matrix([[0.628, 0.372],
##         [0.62 , 0.38 ]])
```

```
P**4
```

```
## matrix([[0.6256, 0.3744],
##         [0.624 , 0.376 ]])
```

```
P**20
```

```
## matrix([[0.625, 0.375],
##         [0.625, 0.375]])
```

p. 19

```
P = np.matrix([[0,1],[1,0]])
```

```
P**2
```

```
## matrix([[1, 0],
##         [0, 1]])
```

```
P**3
```

```
## matrix([[0, 1],
##         [1, 0]])
```

```
"Man can learn nothing unless he proceeds from the known to the unknown. - Claude Bernard"
```