E1_손민상

Son Min Sang

2021-01-20

차 례

page 21

```
import numpy as np
import pandas as pd

# 0: For a fixed \pi,
# 0: For all states s,
#      define R^{\pi}(s) using Eq.(1)
R = np.hstack((np.repeat(-1.5, 4), -0.5, np.repeat(-1.5, 2), 0)).reshape(8, 1)
# 0: For all states s,s'
#      define P^{\pi}_{ss'} using Eq.(2)
states = np.arange(0, 80, step=10)
P = np.array ([[.1, 0, .9, 0, 0, 0, 0, 0],
               [.1, 0, 0, .9, 0, 0, 0, 0],
               [0, .1, 0, 0, .9, 0, 0, 0],
               [0, 0, .1, 0, 0, .9, 0, 0],
               [0, 0, 0, .1, 0, 0, .9, 0],
               [0, 0, 0, 0, .1, 0, 0, .9],
               [0, 0, 0, 0, 0, .1, 0, .9],
               [0, 0, 0, 0, 0,  0, 0, 1]])

P=pd.DataFrame(P,columns=states,index=states)
```

```
print(P)
```

```
##      0    10   20   30   40   50   60   70
## 0   0.1  0.0  0.9  0.0  0.0  0.0  0.0  0.0
## 10  0.1  0.0  0.0  0.9  0.0  0.0  0.0  0.0
## 20  0.0  0.1  0.0  0.0  0.9  0.0  0.0  0.0
## 30  0.0  0.0  0.1  0.0  0.0  0.9  0.0  0.0
## 40  0.0  0.0  0.0  0.1  0.0  0.0  0.9  0.0
## 50  0.0  0.0  0.0  0.0  0.1  0.0  0.0  0.9
## 60  0.0  0.0  0.0  0.0  0.0  0.1  0.0  0.9
## 70  0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0
```

```
print(R)
```

```
## [[-1.5]
##  [-1.5]
##  [-1.5]
##  [-1.5]
```

```
##  [-0.5]
##  [-1.5]
##  [-1.5]
##  [ 0. ]]
```

```
# 1: Let epsilon <- 10^{-8}
gamma = 1.0
epsilon = 10**(-8)
# 2: Let v_0 <- zero vector
v_old = np.array(np.repeat(0, 8)).reshape(8,1)
# 3: Let v_1 <- R + \gamma*P*v_0
v_new = R+gamma*np.dot(P, v_old)
# 4: i <- 1
# 5: While ||v_i-v_{i-1}|| > epsilon
# 6: v_{i+1} <- R + \gamma*P*v_{i}
# 7: i <- i+1
# 8: Return v_{i+1}
while np.max(np.abs(v_new-v_old)) > epsilon:
    v_old = v_new
    v_new = R+gamma*np.dot(P, v_old)
print(v_new.T)
```

```
## [[-5.80592905 -5.2087811  -4.13926239 -3.47576467 -2.35376031 -1.73537603
##   -1.6735376   0.        ]]
```

```
# R, P are already assigned
gamma = 1.0
epsilon = 10**(-8)
v_old = np.array(np.repeat(0, 8)).reshape(8,1)
v_new = R+gamma*np.dot(P, v_old)
results = np.array(v_old) # to save
results= np.append(results,v_new,axis=0) # to save
while np.max(np.abs(v_new-v_old))>epsilon:
    v_old=v_new
    v_new=R+gamma*np.dot(P,v_old)
    results= np.append(results,v_new) # to save
```

```
import pandas as pd
results = pd.DataFrame(np.array(results).reshape(len(results)//8,8),columns = states)
```

```
results.head()
```

```
##       0       10      20     30     40      50     60   70
## 0  0.000  0.0000  0.0000  0.000  0.000  0.0000  0.000  0.0
## 1 -1.500 -1.5000 -1.5000 -1.500 -0.500 -1.5000 -1.500  0.0
## 2 -3.000 -3.0000 -2.1000 -3.000 -2.000 -1.5500 -1.650  0.0
## 3 -3.690 -4.5000 -3.6000 -3.105 -2.285 -1.7000 -1.655  0.0
## 4 -5.109 -4.6635 -4.0065 -3.390 -2.300 -1.7285 -1.670  0.0
```

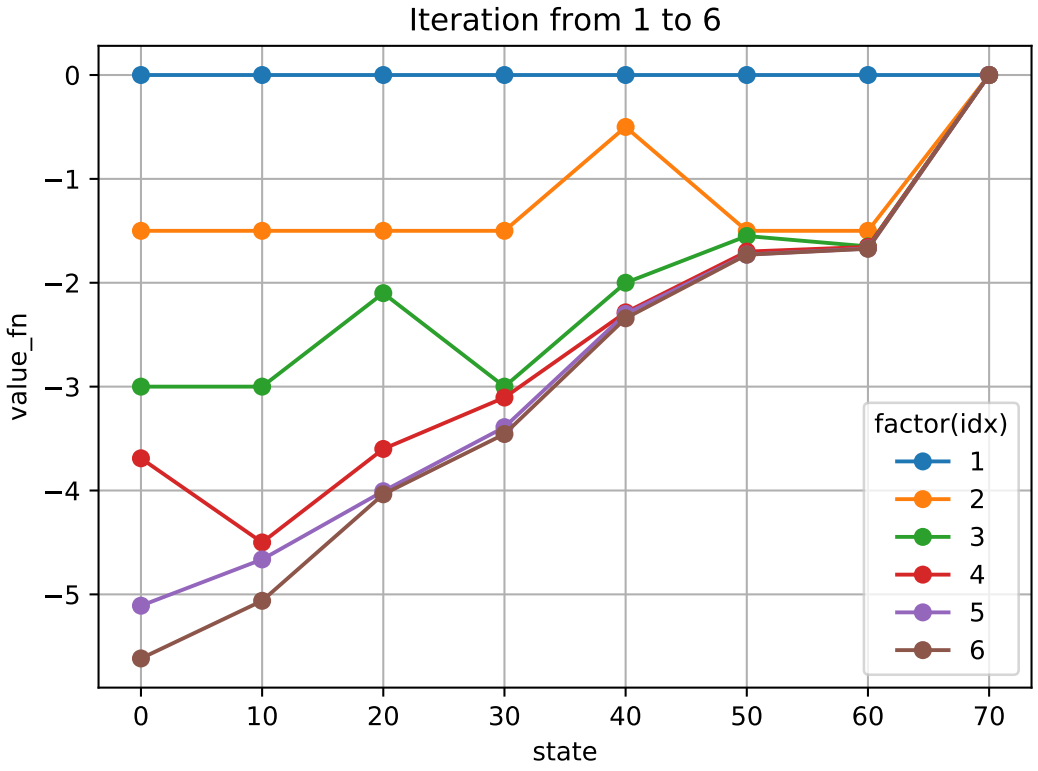
```
results.tail()
```

```
##           0         10        20        30       40        50        60   70
## 18 -5.805929 -5.208781 -4.139262 -3.475765 -2.35376 -1.735376 -1.673538  0.0
## 19 -5.805929 -5.208781 -4.139262 -3.475765 -2.35376 -1.735376 -1.673538  0.0
## 20 -5.805929 -5.208781 -4.139262 -3.475765 -2.35376 -1.735376 -1.673538  0.0
## 21 -5.805929 -5.208781 -4.139262 -3.475765 -2.35376 -1.735376 -1.673538  0.0
## 22 -5.805929 -5.208781 -4.139262 -3.475765 -2.35376 -1.735376 -1.673538  0.0
```

page 24

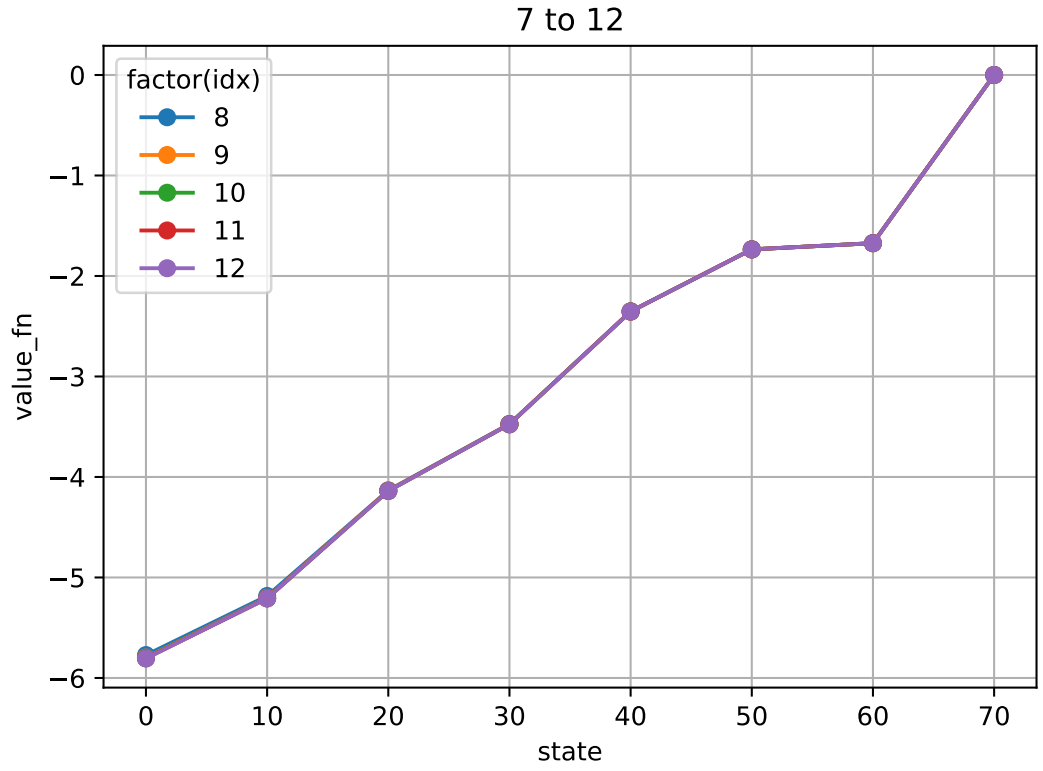
```
import matplotlib.pyplot as plt
for i in range(0,6):
    plt.plot(states, results.iloc[i], marker='o', label=str(i+1))

plt.grid(True)
plt.legend(title='factor(idx)')
plt.xlabel('state')
plt.ylabel('value_fn')
plt.title('Iteration from 1 to 6')
plt.show()
```



```
import matplotlib.pyplot as plt
for i in range(7,12):
    plt.plot(states, results.iloc[i], marker='o', label=str(i+1))

plt.grid(True)
plt.legend(title='factor(idx)')
plt.xlabel('state')
plt.ylabel('value_fn')
plt.title('7 to 12')
plt.show()
```



```
import matplotlib.pyplot as plt
for i in range(13,18):
    plt.plot(states, results.iloc[i], marker='o', label=str(i+1))
plt.grid(True)
plt.legend(title='factor(idx)')
plt.xlabel('state')
plt.ylabel('value_fn')
plt.title('Iteration from 13 to 18')
plt.show()
```