E2 python ver

Lee SungHo

2021-01-22

⚜ ⚜

Page 4 policy eval

```
import numpy as np
import pandas as pd

gamma = 1
states = np.arange(0,80,10).astype('str')
P_normal = pd.DataFrame(np.matrix([[0,1,0,0,0,0,0,0],
                    [0,0,1,0,0,0,0,0],
                    [0,0,0,1,0,0,0,0],
                    [0,0,0,0,1,0,0,0],
                    [0,0,0,0,0,1,0,0],
                    [0,0,0,0,0,0,1,0],
                    [0,0,0,0,0,0,0,1],
                    [0,0,0,0,0,0,0,1]]), index=states,columns=states)

P_speed=pd.DataFrame(np.matrix([[.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]]), index=states, columns=states)

def transition(given_pi, states, P_normal, P_speed):
    P_out=pd.DataFrame(np.zeros((len(states),len(states))),index=states, columns=states)

    for s in states:
        action_dist=given_pi.loc[s]
        P=action_dist['normal']*P_normal+action_dist['speed']*P_speed
        P_out.loc[s]=P.loc[s]

    return P_out

R_s_a=pd.DataFrame(np.matrix([-1,-1,-1,-1,0.0,-1,-1,0,-1.5,-1.5,-1.5,-1.5,-0.5,-1.5,-1.5,0]).reshape(len(stat

def reward_fn(given_pi):

    R_pi=np.array((given_pi*R_s_a).sum(axis=1)).reshape(-1,1)
```

```
    return R_pi

def policy_eval(given_pi):
    R=reward_fn(given_pi)
    P=transition(given_pi, states=states, P_normal=P_normal, P_speed=P_speed)

    gamma=1.0
    epsilon=10**(-8)

    v_old=np.repeat(0,8).reshape(8,1)
    v_new=R+np.dot(gamma*P, v_old)

    while np.max(np.abs(v_new-v_old))>epsilon:
        v_old=v_new
        v_new=R+np.dot(gamma*P,v_old)

    return v_new

pi_speed=pd.DataFrame(np.c_[np.repeat(0,len(states)), np.repeat(1,len(states))],index=states, columns=['norma
a = policy_eval(pi_speed).T
a = a.flatten()

pi_speed_dict = dict(zip([0,10,20,30,40,50,60,70],a))
print(pi_speed_dict)
```

```
## {0: -5.805929051549786, 10: -5.208781102937935, 20: -4.139262388253073, 30: -3.4757646663132333, 40: -
2.353760309257031, 50: -1.7353760309119723, 60: -1.6735376030808617, 70: 0.0}
```

```
pi_50=pd.DataFrame(np.c_[np.repeat(0.5,len(states)), np.repeat(0.5,len(states))],index=states, columns=['norm
b = policy_eval(pi_50).T
b = b.flatten()


pi_50_dict = dict(zip([0,10,20,30,40,50,60,70],b))
print(pi_50_dict)
```

```
## {0: -5.969237864510072, 10: -5.133592223843624, 20: -4.119955245757107, 30: -3.3892282405722756, 40: -
2.041470032106737, 50: -2.0277676939516147, 60: -1.351388384697219, 70: 0.0}
```

p.12 Implementation

```
V_old=policy_eval(pi_speed)
pi_old=pi_speed

q_s_a=R_s_a+np.hstack((np.dot(gamma*P_normal,V_old),np.dot(gamma*P_speed,V_old)))
print(q_s_a)
```

```
##       normal     speed
## 0  -6.208781 -5.805929
## 10 -5.139262 -5.208781
## 20 -4.475765 -4.139262
## 30 -3.353760 -3.475765
## 40 -1.735376 -2.353760
## 50 -2.673538 -1.735376
## 60 -1.000000 -1.673538
## 70  0.000000  0.000000
```

p.12 Implementation

```
idxmax = q_s_a.idxmax(axis=1).tolist()
count = 0
pi_new = pd.DataFrame(np.zeros(16).reshape(8,2),index = q_s_a.index,columns = q_s_a.columns)
for i in q_s_a.index.tolist():

    pi_new.loc[i][idxmax[count]] = 1
    count +=1
pi_new
```

```
##     normal  speed
## 0      0.0    1.0
## 10     1.0    0.0
## 20     0.0    1.0
## 30     1.0    0.0
## 40     1.0    0.0
```

```
## 50     0.0    1.0
## 60     1.0    0.0
## 70     1.0    0.0
```

```
def policy_improve(V_old, pi_old=pi_old, R_s_a=R_s_a, gamma=gamma, P_normal=P_normal, P_speed=P_speed):
    q_s_a=R_s_a+np.c_[np.dot(gamma*P_normal,V_old), np.dot(gamma*P_speed, V_old)]

    pi_new_vec=q_s_a.idxmax(axis=1)
    pi_new=pd.DataFrame(np.zeros(pi_old.shape), index=pi_old.index, columns=pi_old.columns)

    for i in range(len(pi_new_vec)):
        pi_new.iloc[i][pi_new_vec[i]]=1

    return pi_new
pi_old=pi_speed
V_old=policy_eval(pi_old)
pi_new=policy_improve(V_old, pi_old=pi_old, R_s_a=R_s_a, gamma=gamma, P_normal=P_normal, P_speed=P_speed)
pi_old
```

```
##     normal  speed
## 0        0      1
## 10       0      1
## 20       0      1
## 30       0      1
## 40       0      1
## 50       0      1
## 60       0      1
## 70       0      1
```

```
pi_new
```

```
##     normal  speed
## 0      0.0    1.0
## 10     1.0    0.0
## 20     0.0    1.0
## 30     1.0    0.0
## 40     1.0    0.0
## 50     0.0    1.0
```

```
## 60     1.0    0.0
## 70     1.0    0.0
```

p.16 Policy iteration

```
# step0
pi_old=pi_speed
pi_old
# step1
```

```
##     normal  speed
## 0        0      1
## 10       0      1
## 20       0      1
## 30       0      1
## 40       0      1
## 50       0      1
## 60       0      1
## 70       0      1
```

```
V_old=policy_eval(pi_old)
pi_new=policy_improve(V_old, pi_old=pi_old, R_s_a=R_s_a, gamma=gamma, P_normal=P_normal, P_speed=P_speed)
pi_old=pi_new
pi_old
# step2
```

```
##     normal  speed
## 0      0.0    1.0
## 10     1.0    0.0
## 20     0.0    1.0
## 30     1.0    0.0
## 40     1.0    0.0
## 50     0.0    1.0
## 60     1.0    0.0
## 70     1.0    0.0
```

```
V_old=policy_eval(pi_old)
pi_new=policy_improve(V_old, pi_old=pi_old, R_s_a=R_s_a, gamma=gamma, P_normal=P_normal, P_speed=P_speed)
pi_old=pi_new
```

```
pi_old
# step3
```

```
##     normal  speed
## 0      0.0    1.0
## 10     0.0    1.0
## 20     0.0    1.0
## 30     1.0    0.0
## 40     1.0    0.0
## 50     0.0    1.0
## 60     1.0    0.0
## 70     1.0    0.0
```

```
V_old=policy_eval(pi_old)
pi_new=policy_improve(V_old, pi_old=pi_old, R_s_a=R_s_a, gamma=gamma, P_normal=P_normal, P_speed=P_speed)
pi_old=pi_new
pi_old
```

```
##     normal  speed
## 0      0.0    1.0
## 10     0.0    1.0
## 20     0.0    1.0
## 30     1.0    0.0
## 40     1.0    0.0
## 50     0.0    1.0
## 60     1.0    0.0
## 70     1.0    0.0
```

P.18 Policy iteration process (from π^{speed})

```
## 0 -th iteration
##         0  10  20  30  40  50  60  70
## normal  0   0   0   0   0   0   0   0
## speed   1   1   1   1   1   1   1   1
## 1 -th iteration
##           0   10   20   30   40   50   60   70
## normal  0.0  1.0  0.0  1.0  1.0  0.0  1.0  1.0
## speed   1.0  0.0  1.0  0.0  0.0  1.0  0.0  0.0
## 2 -th iteration
##           0   10   20   30   40   50   60   70
## normal  0.0  0.0  0.0  1.0  1.0  0.0  1.0  1.0
## speed   1.0  1.0  1.0  0.0  0.0  1.0  0.0  0.0

## [[-5.1077441 ]
##  [-4.41077441]
##  [-3.44107744]
##  [-2.66666667]
##  [-1.66666667]
##  [-1.66666667]
##  [-1.        ]
##  [ 0.        ]]
```

P.19 Policy iteration process (from π^{50})

```
pi_old=pi_50
cnt=0
while True:
    print(cnt,'-th iteration')
    print(pi_old.T)

    V_old=policy_eval(pi_old)
    pi_new=policy_improve(V_old, pi_old=pi_old, R_s_a=R_s_a, gamma=gamma, P_normal=P_normal, P_speed=P_speed)

    if pi_new.equals(pi_old)==True:
        break

    pi_old=pi_new
    cnt+=1
```

```
## 0 -th iteration
##           0   10   20   30   40   50   60   70
## normal  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5
## speed   0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5
## 1 -th iteration
##           0   10   20   30   40   50   60   70
## normal  0.0  1.0  0.0  1.0  1.0  0.0  1.0  1.0
## speed   1.0  0.0  1.0  0.0  0.0  1.0  0.0  0.0
## 2 -th iteration
##           0   10   20   30   40   50   60   70
## normal  0.0  0.0  0.0  1.0  1.0  0.0  1.0  1.0
## speed   1.0  1.0  1.0  0.0  0.0  1.0  0.0  0.0
```

```
print(policy_eval(pi_new))
```

```
## [[-5.1077441 ]
##  [-4.41077441]
##  [-3.44107744]
##  [-2.66666667]
##  [-1.66666667]
```

```
##  [-1.66666667]
##  [-1.        ]
##  [ 0.        ]]
```