lnotes2_MDP

Tae Hyeon Kwon, undergrad(ITM)

2021-01-29

```
import numpy as np
import pandas as pd

states = ['S1','S2','S3','S4','S5','S6','S7']

P_forward = pd.DataFrame(np.matrix([[1,0,0,0,0,0,0],
                       [1,0,0,0,0,0,0],
                       [0,1,0,0,0,0,0],
                       [0,0,1,0,0,0,0],
                       [0,0,0,1,0,0,0],
                       [0,0,0,0,1,0,0],
                       [0,0,0,0,0,1,0]]), index = states, columns= states)

P_backward = pd.DataFrame(np.matrix([[0,1,0,0,0,0,0],
                       [0,0,1,0,0,0,0],
                       [0,0,0,1,0,0,0],
                       [0,0,0,0,1,0,0],
                       [0,0,0,0,0,1,0],
                       [0,0,0,0,0,0,1],
                        [0,0,0,0,0,0,1]]),index= states, columns=states)


print(P_forward)
```

```
##     S1  S2  S3  S4  S5  S6  S7
## S1   1   0   0   0   0   0   0
## S2   1   0   0   0   0   0   0
## S3   0   1   0   0   0   0   0
## S4   0   0   1   0   0   0   0
## S5   0   0   0   1   0   0   0
## S6   0   0   0   0   1   0   0
## S7   0   0   0   0   0   1   0
```

```
print(P_backward)
```

```
##     S1  S2  S3  S4  S5  S6  S7
## S1   0   1   0   0   0   0   0
## S2   0   0   1   0   0   0   0
## S3   0   0   0   1   0   0   0
## S4   0   0   0   0   1   0   0
## S5   0   0   0   0   0   1   0
## S6   0   0   0   0   0   0   1
## S7   0   0   0   0   0   0   1
```

```
pi_forward = pd.DataFrame(np.c_[np.repeat(1,len(states)), np.repeat(0,len(states))],index = states, colu

print(pi_forward)
```

```
##     forward  backward
## S1        1         0
## S2        1         0
## S3        1         0
## S4        1         0
## S5        1         0
## S6        1         0
## S7        1         0
```

```
pi_backward = pd.DataFrame(np.c_[np.repeat(0,len(states)), np.repeat(1,len(states))],index = states, col

print(pi_backward)
```

```
##     forward  backward
## S1        0         1
## S2        0         1
## S3        0         1
## S4        0         1
## S5        0         1
## S6        0         1
## S7        0         1
```

```
pi_50 = pd.DataFrame(np.c_[np.repeat(0.5,len(states)), np.repeat(0.5,len(states))],index = states, colum

print(pi_50)
```

```
##     forward  backward
## S1      0.5       0.5
## S2      0.5       0.5
## S3      0.5       0.5
## S4      0.5       0.5
## S5      0.5       0.5
## S6      0.5       0.5
## S7      0.5       0.5
```

```
def transition(given_pi,states,P_forward,P_backward):
    P_out = pd.DataFrame(np.zeros((len(states),len(states))),index= states,columns=states)

    for s in range(len(states)):
        action_dist = given_pi.iloc[s]
        P = action_dist[’forward’]*P_forward+action_dist[’backward’]*P_backward
        P_out.iloc[s] = P.iloc[s]

    return P_out

print(transition(pi_forward,states=states,P_forward=P_forward,P_backward=P_backward))
```

```
##      S1   S2   S3   S4   S5   S6   S7
## S1  1.0  0.0  0.0  0.0  0.0  0.0  0.0
## S2  1.0  0.0  0.0  0.0  0.0  0.0  0.0
## S3  0.0  1.0  0.0  0.0  0.0  0.0  0.0
## S4  0.0  0.0  1.0  0.0  0.0  0.0  0.0
## S5  0.0  0.0  0.0  1.0  0.0  0.0  0.0
## S6  0.0  0.0  0.0  0.0  1.0  0.0  0.0
## S7  0.0  0.0  0.0  0.0  0.0  1.0  0.0
```

```
R_s_a = pd.DataFrame(np.matrix([1,0,0,0,0,0,10,1,0,0,0,0,0,10]).reshape(len(states),2,order=’F’),index=

print(R_s_a)
```

```
##     forward  backward
## S1        1         1
## S2        0         0
## S3        0         0
## S4        0         0
## S5        0         0
## S6        0         0
## S7       10        10
```

```
def reward_fn(given_pi):
    R_s_a = pd.DataFrame(np.matrix([1,0,0,0,0,0,10,1,0,0,0,0,0,10]).reshape(len(states),2,order=’F’),ind
    R_pi = np.asarray((given_pi*R_s_a).sum(axis=1)).reshape(-1,1)

    return R_pi

print(reward_fn(pi_forward))
```

```
## [[ 1]
##  [ 0]
##  [ 0]
##  [ 0]
##  [ 0]
##  [ 0]
##  [10]]
```

```
def policy_eval(given_pi):
    R = reward_fn(given_pi)
    P = transition(given_pi, states=states, P_forward = P_forward, P_backward= P_backward)

    gamma = 0.9
    epsilon = 10**(-8)

    v_old = np.repeat(0,7).reshape(7,1)
    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

print(policy_eval(pi_forward))
```

```
## [[ 9.99999991]
##  [ 8.99999991]
##  [ 8.09999991]
##  [ 7.28999991]
##  [ 6.56099991]
##  [ 5.90489991]
##  [15.31440991]]
```

```
gamma = 0.9
V_old = policy_eval(pi_forward)
pi_old = pi_forward
q_s_a = R_s_a+np.c_[np.dot(gamma*P_forward,V_old),np.dot(gamma*P_backward,V_old)]

print(q_s_a)
```

```
##      forward   backward
## S1  10.00000   9.100000
## S2   9.00000   7.290000
## S3   8.10000   6.561000
## S4   7.29000   5.904900
## S5   6.56100   5.314410
## S6   5.90490  13.782969
## S7  15.31441  23.782969
```

```
pi_new_vec = q_s_a.idxmax(axis=1)
print(pi_new_vec)
```

```
## S1     forward
## S2     forward
## S3     forward
## S4     forward
## S5     forward
## S6    backward
## S7    backward
## dtype: object
```

```
pi_new = pd.DataFrame(np.zeros(shape=(pi_old.shape)),columns=['foward','backward'])

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


print(pi_new)
```

```
##    foward  backward
## 0     0.0       0.0
## 1     0.0       0.0
## 2     0.0       0.0
## 3     0.0       0.0
## 4     0.0       0.0
## 5     0.0       1.0
## 6     0.0       1.0
```

```
def policy_improve(V_old, pi_old, R_s_a = R_s_a, gamma=gamma, P_forward = P_forward, P_backward=P_backwa

    q_s_a = R_s_a + np.c_[np.dot(gamma*P_forward,V_old),np.dot(gamma*P_backward,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_forward
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_forward=P_forward, P_backward=

print(pi_old)
```

```
##     forward  backward
## S1        1         0
## S2        1         0
## S3        1         0
## S4        1         0
## S5        1         0
## S6        1         0
## S7        1         0
```

```
print(pi_new)
```

```
##     forward  backward
## S1      1.0       0.0
## S2      1.0       0.0
## S3      1.0       0.0
## S4      1.0       0.0
## S5      1.0       0.0
## S6      0.0       1.0
## S7      0.0       1.0
```

```
pi_old=pi_forward
cnt = 0

while True:
    print(cnt, '-th iteration')
    print(pi_old)

    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_forward=P_forward, P_backw
    if pi_new.equals(pi_old)==True:
        break

    pi_old = pi_new
    cnt+=1
```

```
## 0 -th iteration
##     forward  backward
## S1        1         0
## S2        1         0
## S3        1         0
## S4        1         0
## S5        1         0
## S6        1         0
## S7        1         0
## 1 -th iteration
##     forward  backward
## S1      1.0       0.0
## S2      1.0       0.0
## S3      1.0       0.0
## S4      1.0       0.0
## S5      1.0       0.0
## S6      0.0       1.0
## S7      0.0       1.0
## 2 -th iteration
##     forward  backward
## S1      1.0       0.0
## S2      1.0       0.0
## S3      1.0       0.0
## S4      1.0       0.0
## S5      0.0       1.0
## S6      0.0       1.0
## S7      0.0       1.0
## 3 -th iteration
##     forward  backward
## S1      1.0       0.0
## S2      1.0       0.0
## S3      1.0       0.0
## S4      0.0       1.0
## S5      0.0       1.0
## S6      0.0       1.0
## S7      0.0       1.0
## 4 -th iteration
##     forward  backward
## S1      1.0       0.0
## S2      1.0       0.0
## S3      0.0       1.0
## S4      0.0       1.0
## S5      0.0       1.0
## S6      0.0       1.0
## S7      0.0       1.0
## 5 -th iteration
##     forward  backward
## S1      1.0       0.0
## S2      0.0       1.0
## S3      0.0       1.0
## S4      0.0       1.0
## S5      0.0       1.0
## S6      0.0       1.0
## S7      0.0       1.0
## 6 -th iteration
##     forward  backward
## S1      0.0       1.0
## S2      0.0       1.0
## S3      0.0       1.0
## S4      0.0       1.0
## S5      0.0       1.0
## S6      0.0       1.0
## S7      0.0       1.0
```

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

```
## [[54.14409991]
##  [59.04899991]
##  [65.60999991]
##  [72.89999991]
##  [80.99999991]
##  [89.99999991]
##  [99.99999991]]
```

```
pi_old=pi_50
cnt = 0

while True:
    print(cnt, '-th iteration')
    print(pi_old)

    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_forward=P_forward, P_backw
    if pi_new.equals(pi_old)==True:
        break

    pi_old = pi_new
    cnt+=1
```

```
## 0 -th iteration
##     forward  backward
## S1      0.5       0.5
## S2      0.5       0.5
## S3      0.5       0.5
## S4      0.5       0.5
## S5      0.5       0.5
## S6      0.5       0.5
## S7      0.5       0.5
## 1 -th iteration
##     forward  backward
## S1      1.0       0.0
## S2      0.0       1.0
## S3      0.0       1.0
## S4      0.0       1.0
## S5      0.0       1.0
## S6      0.0       1.0
## S7      0.0       1.0
## 2 -th iteration
##     forward  backward
## S1      0.0       1.0
## S2      0.0       1.0
## S3      0.0       1.0
## S4      0.0       1.0
## S5      0.0       1.0
## S6      0.0       1.0
## S7      0.0       1.0
```

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

```
## [[54.14409991]
##  [59.04899991]
##  [65.60999991]
##  [72.89999991]
##  [80.99999991]
##  [89.99999991]
##  [99.99999991]]
```