F3 Python Code

Kang, Eui Hyeon

2021-02-22

차 례

Preparation

```
# model
states=np.arange(0,70+10,10).astype('str')
states
```

```
## array(['0', '10', '20', '30', '40', '50', '60', '70'], dtype='<U11')
```

```
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)

R_s_a=pd.DataFrame(np.c_[[-1,-1,-1,-1,0,-1,-1,0],[-1.5,-1.5,-1.5,-1.5,-0.5,-1.5,-1.5,0]], index=states, colum

q_s_a_init=pd.DataFrame(np.c_[np.repeat(0.0,len(states)), np.repeat(0.0,len(states))], index=states, columns=


# policy

pi_speed=pd.DataFrame(np.c_[np.repeat(0,len(states)), np.repeat(1, len(states))], index=states, columns=['n',

pi_50=pd.DataFrame(np.c_[np.repeat(0.5, len(states)), np.repeat(0.5, len(states))], index=states, columns=['r



# simul_path()
```

```
def simul_path(pi, P_normal, P_speed, R_s_a):
    s_now='0'
    history_i=list(s_now)

    while s_now!='70':
        if np.random.uniform(0,1) < pi.loc[s_now]['n']:
            a_now='n'
            P=P_speed
        else:
            a_now='s'
            P=P_speed

        r_now=R_s_a.loc[s_now][a_now]
        s_next=states[np.argmin(P.loc[s_now].cumsum()<np.random.uniform(0,1))].item()
        history_i.extend([a_now,r_now,s_next])
        s_now=s_next

    return history_i



# simul_step()

def simul_step(pi, s_now, P_normal, P_speed, R_s_a):
    if np.random.uniform(0,1) < pi.loc[s_now]['n']:
        a_now='n'
        P=P_normal
    else:
        a_now='s'
        P=P_speed

    r_now=R_s_a.loc[s_now][a_now]
    s_next=states[np.argmin(P.loc[s_now].cumsum() < np.random.uniform(0,1))].item()

    if np.random.uniform(0,1) < pi.loc[s_next]['n']:
        a_next='n'
    else:
        a_next='s'

    sarsa=[s_now, a_now, r_now, s_next, a_next]
    return sarsa



## pol_eval_MC()

def pol_eval_MC(sample_path, q_s_a, alpha):
    for j in range(0, len(sample_path)-1, 3):
        s=sample_path[j]
        a=sample_path[j+1]
        G=sum([sample_path[g] for g in range(j+2, len(sample_path)-1, 3)])

        q_s_a.loc[s][a]=q_s_a.loc[s][a]+alpha*(G-q_s_a.loc[s][a])

    return q_s_a



## pol_eval_TD()

def pol_eval_TD(sample_step, q_s_a, alpha):
    s=sample_step[0]
    a=sample_step[1]
    r=sample_step[2]
    s_next=sample_step[3]
    a_next=sample_step[4]

    q_s_a.loc[s][a]=q_s_a.loc[s][a]+alpha*(r+q_s_a.loc[s_next][a_next]-q_s_a.loc[s][a])

    return q_s_a



# pol_imp()

def pol_imp(pi, q_s_a, epsilon): # epsilon=exploration_rate
    # VERSION 1
    for i in range(0, pi.shape[0]):
        if np.random.uniform(0,1)>epsilon: #exploitation
            pi.iloc[i]=0
            pi.iloc[i][q_s_a.iloc[i].idxmax()]=1
        else:
            pi.iloc[i]=1/q_s_a.shape[1]
    return pi
```

6p

```
import time
from copy import deepcopy

num_ep=10**3

beg_time=time.time()
q_s_a=deepcopy(q_s_a_init)
pi=deepcopy(pi_50)

for epi_i in range(1,num_ep+1):
    sample_path_i=simul_path(pi,P_normal,P_speed,R_s_a)
    q_s_a=pol_eval_MC(sample_path_i, q_s_a, alpha=1/epi_i)
    pi=pol_imp(pi,q_s_a,epsilon=1/epi_i)

end_time=time.time()

print(end_time-beg_time)
```

```
## 9.53550410270691
```

```
pi.T
```

```
##      0   10   20   30   40   50   60   70
## n  0.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0
## s  1.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
```

```
q_s_a.T
```

```
##           0        10        20        30        40        50        60   70
## n -5.289187 -1.028212 -2.556330 -1.563940 -1.275982 -0.852303 -1.126229  0.0
## s -4.258753 -1.030933 -3.077051 -1.564477 -1.969025 -0.852731 -1.173440  0.0
```

7p

```
num_ep=10**4

beg_time=time.time()
q_s_a=deepcopy(q_s_a_init)
pi=deepcopy(pi_50)

for epi_i in range(1,num_ep+1):
    sample_path_i=simul_path(pi,P_normal,P_speed,R_s_a)
    q_s_a=pol_eval_MC(sample_path_i, q_s_a, alpha=1/epi_i)
    pi=pol_imp(pi,q_s_a,epsilon=1/epi_i)

end_time=time.time()

print(end_time-beg_time)
```

```
## 96.02949500083923
```

```
pi.T
```

```
##      0   10   20   30   40   50   60   70
## n  1.0  0.0  1.0  0.0  0.0  1.0  1.0  1.0
## s  0.0  1.0  0.0  1.0  1.0  0.0  0.0  0.0
```

```
q_s_a.T
```

```
##           0         10         20         30         40         50         60   70
## n -4.108983 -1.597157 -3.001617 -1.769957 -1.873124 -1.013091 -1.131579  0.0
## s -4.286798 -1.596853 -3.195652 -1.769765 -1.788312 -1.013258 -1.522222  0.0
```

8p

```
num_ep=10**5

beg_time=time.time()
q_s_a=deepcopy(q_s_a_init)
pi=deepcopy(pi_50)
exploration_rate=1

for epi_i in range(1,num_ep+1):
    sample_path_i=simul_path(pi,P_normal,P_speed,R_s_a)
    q_s_a=pol_eval_MC(sample_path_i, q_s_a, alpha=1/epi_i)
    pi=pol_imp(pi,q_s_a,epsilon=exploration_rate)
    exploration_rate=exploration_rate*0.9995 # exponential decay

end_time=time.time()

print(end_time-beg_time)
```

```
## 959.0042493343353
```

```
pi.T
```

```
##      0   10   20   30   40   50   60   70
## n  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0
## s  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
```

```
q_s_a.T
```

```
##           0        10        20        30        40        50        60   70
## n -3.566691 -1.793025 -2.454543 -1.736483 -1.231051 -0.977252 -1.113674  0.0
## s -4.744891 -4.513975 -3.304857 -2.983478 -1.921747 -1.555915 -1.572006  0.0
```

Policy iteration 2 - TD control

```
# 11p

num_ep=10**3

beg_time=time.time()
q_s_a=deepcopy(q_s_a_init)
pi=deepcopy(pi_50)
for epi_i in range(1,num_ep+1):
    s_now='0'
    while s_now!='70':
        sample_step=simul_step(pi, s_now, P_normal, P_speed, R_s_a)
        q_s_a=pol_eval_TD(sample_step, q_s_a, alpha=1/epi_i)
        pi=pol_imp(pi, q_s_a, epsilon=1/epi_i)
        s_now=sample_step[3]

end_time=time.time()

print(end_time-beg_time)
```

```
## 30.204890966415405
```

```
q_s_a.T
```

```
##           0        10        20        30        40        50        60   70
## n -3.423858 -2.899619 -2.577529 -2.134927 -1.389212 -1.545944 -0.951793  0.0
## s -3.423666 -2.900234 -2.577116 -2.135522 -1.390092 -1.545532 -1.505833  0.0
```

```
pi.T
```

```
##      0   10   20   30   40   50   60   70
## n  0.0  1.0  0.0  1.0  1.0  0.0  1.0  1.0
## s  1.0  0.0  1.0  0.0  0.0  1.0  0.0  0.0
```

12p

```
num_ep=10**4

beg_time=time.time()
q_s_a=deepcopy(q_s_a_init)
pi=deepcopy(pi_50)
for epi_i in range(1,num_ep+1):
    s_now='0'
    while s_now!='70':
        sample_step=simul_step(pi, s_now, P_normal, P_speed, R_s_a)
        q_s_a=pol_eval_TD(sample_step, q_s_a, alpha=1/epi_i)
        pi=pol_imp(pi, q_s_a, epsilon=1/epi_i)
        s_now=sample_step[3]

end_time=time.time()

print(end_time-beg_time)
```

```
## 298.99387288093567
```

```
q_s_a.T
```

```
##           0        10        20        30        40        50        60   70
## n -4.012386 -3.385288 -2.926170 -2.373648 -1.548923 -1.634846 -1.000000  0.0
## s -4.012352 -3.385322 -2.926169 -2.373683 -1.548996 -1.634177 -1.016484  0.0
```

```
pi.T
```

```
##      0   10   20   30   40   50   60   70
## n  0.0  1.0  0.0  1.0  1.0  0.0  1.0  1.0
## s  1.0  0.0  1.0  0.0  0.0  1.0  0.0  0.0
```

13p

```
num_ep=10**5

beg_time=time.time()
q_s_a=deepcopy(q_s_a_init)
pi=deepcopy(pi_50)
exploration_rate=1
for epi_i in range(1,num_ep+1):
    s_now='0'
    while s_now!='70':
        sample_step=simul_step(pi, s_now, P_normal, P_speed, R_s_a)
        q_s_a=pol_eval_TD(sample_step, q_s_a, alpha=max(1/epi_i, 0.01))
        pi=pol_imp(pi, q_s_a, epsilon=1/exploration_rate)
        s_now=sample_step[3]
        exploration_rate=exploration_rate*0.9995

end_time=time.time()

print(end_time-beg_time)
```

```
## 1362.6688046455383
```

```
q_s_a.T
```

```
##            0         10         20         30         40         50         60   70
## n -6.075284 -5.067282 -4.359235 -3.012834 -2.017691 -2.334969 -1.000000  0.0
## s -5.712250 -5.113906 -3.737914 -3.695547 -2.008677 -1.702930 -1.723847  0.0
```

```
pi.T
```

```
##      0   10   20   30   40   50   60   70
## n  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5
## s  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5
```

Exercise (MC, num_ep=10**3, exploration_rate=1, exponential decay=0.995)

```
num_ep=10**3

beg_time=time.time()
q_s_a=deepcopy(q_s_a_init)
pi=deepcopy(pi_50)
exploration_rate=1

for epi_i in range(1,num_ep+1):
    sample_path_i=simul_path(pi,P_normal,P_speed,R_s_a)
    q_s_a=pol_eval_MC(sample_path_i, q_s_a, alpha=1/epi_i)
    pi=pol_imp(pi,q_s_a,epsilon=exploration_rate)
    exploration_rate=exploration_rate*0.995 # exponential decay

end_time=time.time()

print(end_time-beg_time)
```

```
## 9.386426448822021
```

```
pi.T
```

```
##      0   10   20   30   40   50   60   70
## n  1.0  1.0  1.0  0.0  1.0  1.0  1.0  1.0
## s  0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0
```

```
q_s_a.T
```

```
##           0        10        20        30        40        50        60   70
## n -3.824220 -1.405553 -2.732911 -2.654827 -1.362451 -0.835658 -1.083534  0.0
## s -4.406265 -1.444427 -3.216727 -1.686573 -1.906007 -1.677361 -1.497578  0.0
```

Exercise (TD, num_ep=10**3, exploration_rate=1, exponential decay=0.99)

```
num_ep=10**3

beg_time=time.time()
q_s_a=deepcopy(q_s_a_init)
pi=deepcopy(pi_50)
exploration_rate=1
for epi_i in range(1,num_ep+1):
    s_now='0'
    while s_now!='70':
        sample_step=simul_step(pi, s_now, P_normal, P_speed, R_s_a)
        q_s_a=pol_eval_TD(sample_step, q_s_a, alpha=max(1/epi_i, 0.01))
        pi=pol_imp(pi, q_s_a, epsilon=1/exploration_rate)
        s_now=sample_step[3]
        exploration_rate=exploration_rate*0.99

end_time=time.time()

print(end_time-beg_time)
```

```
## 13.578648328781128
```

```
q_s_a.T
```

```
##           0         10         20         30         40         50         60   70
## n -4.784667 -4.134550 -3.793974 -2.694717 -1.885541 -2.294475 -1.000000  0.0
## s -5.001486 -4.197086 -3.466907 -3.427844 -1.948695 -1.692082 -1.746623  0.0
```

```
pi.T
```

```
##      0   10   20   30   40   50   60   70
## n  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5
## s  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5
```