D_case

Tae Hyeon Kwon, undergrad(ITM)

2021 2 5

1. Problem

-In korea, Thanksgiving day, many people go to the their hometown. So, that day many cars in highway and many people take a rest in service area. There are 5 area between Seoul and Busan. And almost all vehicles stop at service area least 1. The more you use a rest area, the higher your satisfaction, but if you use a busy area, your satisfaction may decrease.

2. state

S = {Seoul,A,B,C,D,E,Busan}

(A,B,C,D,E) is name of service area

3. transition matrix

Probability of visiting service area

4. reward

-Seoul, Busan and A,E is -1.0

-B,D is -1.5

-C is -2.0 (because area C is middle of Seoul to Busan. So, many cars want to rest there.)

states and transition matrix, reward

```
import numpy as np
import pandas as pd


states = ['0','A','B','C','D','E','1']
P_transition = pd.DataFrame(np.matrix([[0,0.05,0.15,0.4,0.25,0.05,0.1],
                                       [0.05,0,0.05,0.15,0.4,0.15,0.2],
                                       [0.15,0.05,0,0.05,0.2,0.25,0.3],
                                       [0.35,0.15,0.05,0,0.05,0.15,0.35],
                                       [0.3,0.25,0.2,0.05,0,0.05,0.15],
                                       [0.2,0.15,0.4,0.15,0.05,0,0.05],
                                      [0.1,0.05,0.25,0.4,0.15,0.05,0]]),index=states,columns=states)


print(P_transition)
```

```
##       0     A     B     C     D     E     1
## 0  0.00  0.05  0.15  0.40  0.25  0.05  0.10
## A  0.05  0.00  0.05  0.15  0.40  0.15  0.20
## B  0.15  0.05  0.00  0.05  0.20  0.25  0.30
## C  0.35  0.15  0.05  0.00  0.05  0.15  0.35
## D  0.30  0.25  0.20  0.05  0.00  0.05  0.15
## E  0.20  0.15  0.40  0.15  0.05  0.00  0.05
## 1  0.10  0.05  0.25  0.40  0.15  0.05  0.00
```

```
R_s_a = pd.DataFrame(np.matrix([-1,-1,-1.5,-2.0,-1.5,-1,-1]).reshape(len(states),1),index=states,columns


print(R_s_a.T)
```

```
##           0    A    B    C    D    E    1
## reward -1.0 -1.0 -1.5 -2.0 -1.5 -1.0 -1.0
```

```
pi_stop = pd.DataFrame(np.c_[np.repeat(0,len(states))],index=states,columns=['reward'])

print(pi_stop.T)
```

```
##         0  A  B  C  D  E  1
## reward  0  0  0  0  0  0  0
```

gamma

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

v_old = np.array(np.zeros(7,)).reshape(7,1)
v_new = R_s_a + np.dot(gamma*P_transition, v_old)

results = v_old.T
results = np.vstack((results,v_new.T))
while np.max(np.abs(v_new-v_old)).item() > epsilon:
    v_old = v_new
    v_new = R_s_a + np.dot(gamma*P_transition, v_old)
    results = np.vstack((results,v_new.T))

results = pd.DataFrame(results, columns=states)
print(v_new.T)
```

```
##                 0          A          B  ...          D          E          1
## reward -15.482393 -15.071241 -15.382861  ... -15.382861 -15.071241 -15.482393
##
## [1 rows x 7 columns]
```

```
print(results.head())
```

```
##           0         A         B         C         D         E         1
## 0  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
## 1 -1.000000 -1.000000 -1.500000 -2.000000 -1.500000 -1.000000 -1.000000
## 2 -2.440000 -2.237500 -2.535000 -3.035000 -2.535000 -2.237500 -2.440000
## 3 -3.426175 -3.287463 -3.685200 -4.369475 -3.685200 -3.287463 -3.426175
## 4 -4.503910 -4.297082 -4.635178 -5.377773 -4.635178 -4.297082 -4.503910
```

```
print(results.tail())
```

```
##              0          A          B        C          D          E          1
## 211 -15.482393 -15.071241 -15.382861 -17.2076 -15.382861 -15.071241 -15.482393
## 212 -15.482393 -15.071241 -15.382861 -17.2076 -15.382861 -15.071241 -15.482393
## 213 -15.482393 -15.071241 -15.382861 -17.2076 -15.382861 -15.071241 -15.482393
## 214 -15.482393 -15.071241 -15.382861 -17.2076 -15.382861 -15.071241 -15.482393
## 215 -15.482393 -15.071241 -15.382861 -17.2076 -15.382861 -15.071241 -15.482393
```

MC

```
pi = pi_stop
np.random.seed(1234)

history = list()
MC_N = 10**4

for MC_i in range(MC_N):
    s_now = '0'
    history_i = list(s_now)

    while s_now != '1':

        a_now = 'reward'
        P = P_transition

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

    history.append(history_i)

history_stop = history

func = np.vectorize(lambda x: ','.join(x))

print(pd.Series(func(history_stop[:20])))
```

```
## 0     0,reward,-1.0,B,reward,-1.5,E,reward,-1.0,B,re...
## 1     0,reward,-1.0,D,reward,-1.5,0,reward,-1.0,C,re...
## 2                                       0,reward,-1.0,1
## 3     0,reward,-1.0,E,reward,-1.0,B,reward,-1.5,E,re...
## 4     0,reward,-1.0,C,reward,-2.0,D,reward,-1.5,A,re...
## 5     0,reward,-1.0,C,reward,-2.0,E,reward,-1.0,0,re...
## 6     0,reward,-1.0,D,reward,-1.5,A,reward,-1.0,E,re...
## 7                         0,reward,-1.0,C,reward,-2.0,1
## 8                         0,reward,-1.0,B,reward,-1.5,1
## 9           0,reward,-1.0,D,reward,-1.5,0,reward,-1.0,1
## 10                        0,reward,-1.0,C,reward,-2.0,1
## 11    0,reward,-1.0,B,reward,-1.5,A,reward,-1.0,0,re...
## 12                        0,reward,-1.0,D,reward,-1.5,1
## 13                                      0,reward,-1.0,1
## 14    0,reward,-1.0,D,reward,-1.5,0,reward,-1.0,D,re...
## 15    0,reward,-1.0,B,reward,-1.5,0,reward,-1.0,D,re...
## 16    0,reward,-1.0,C,reward,-2.0,B,reward,-1.5,0,re...
## 17    0,reward,-1.0,D,reward,-1.5,B,reward,-1.5,0,re...
## 18    0,reward,-1.0,C,reward,-2.0,B,reward,-1.5,E,re...
## 19    0,reward,-1.0,B,reward,-1.5,0,reward,-1.0,C,re...
## dtype: object
##
```

```
## D:\miniconda3\envs\r-reticulate\lib\site-packages\numpy\core\_asarray.py:83: VisibleDeprecationWarni
##   return array(a, dtype, copy=False, order=order)
```

TD (Seoul -> Busan)

```
pol_eval = pd.DataFrame(np.zeros((len(states),2)),index=states,columns=['count','est'])

for episode_i in range(len(history_stop)):
    history_i = history_stop[episode_i]

    for j in range(0,len(history_i),3):
        pol_eval.loc[history_i[j]]['count']+=1
        current_cnt = pol_eval.loc[history_i[j]]['count']

        if j+3 < len(history_i):
            TD_tgt = float(history_i[j+2])+pol_eval.loc[history_i[j+3]]['est']
        else:
            TD_tgt = 0

        alpha = 1/current_cnt

        pol_eval.loc[history_i[j]]['est']+=alpha*(TD_tgt-pol_eval.loc[history_i[j]]['est'])

print(np.round(pol_eval.T,2))
```

```
##               0        A        B         C       D        E        1
## count  19043.00  6202.00  8025.00  10292.00  9600.0  5851.00  10000.0
## est       -6.45    -5.97    -5.78     -6.51    -6.5    -6.55      0.0
```

MC (Busan -> Seoul)

```
pi = pi_stop
np.random.seed(1234)

history = list()
MC_N = 10**4

for MC_i in range(MC_N):
    s_now = '1'
    history_i = list(s_now)

    while s_now != '0':

        a_now = 'reward'
        P = P_transition

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

    history.append(history_i)

history_stop = history

func = np.vectorize(lambda x: ','.join(x))

print(pd.Series(func(history_stop[:20])))
```

```
## 0     1,reward,-1.0,B,reward,-1.5,E,reward,-1.0,B,re...
## 1     1,reward,-1.0,B,reward,-1.5,1,reward,-1.0,E,re...
## 2     1,reward,-1.0,C,reward,-2.0,1,reward,-1.0,B,re...
## 3     1,reward,-1.0,B,reward,-1.5,1,reward,-1.0,C,re...
## 4                         1,reward,-1.0,B,reward,-1.5,0
## 5     1,reward,-1.0,C,reward,-2.0,D,reward,-1.5,A,re...
## 6                         1,reward,-1.0,C,reward,-2.0,0
## 7                         1,reward,-1.0,C,reward,-2.0,0
## 8           1,reward,-1.0,C,reward,-2.0,D,reward,-1.5,0
## 9     1,reward,-1.0,C,reward,-2.0,1,reward,-1.0,C,re...
## 10    1,reward,-1.0,C,reward,-2.0,A,reward,-1.0,C,re...
## 11    1,reward,-1.0,B,reward,-1.5,1,reward,-1.0,C,re...
## 12    1,reward,-1.0,B,reward,-1.5,D,reward,-1.5,B,re...
## 13    1,reward,-1.0,C,reward,-2.0,1,reward,-1.0,C,re...
## 14    1,reward,-1.0,C,reward,-2.0,1,reward,-1.0,C,re...
## 15                        1,reward,-1.0,C,reward,-2.0,0
## 16    1,reward,-1.0,E,reward,-1.0,A,reward,-1.0,C,re...
## 17    1,reward,-1.0,B,reward,-1.5,A,reward,-1.0,1,re...
## 18    1,reward,-1.0,B,reward,-1.5,D,reward,-1.5,B,re...
## 19                                    1,reward,-1.0,0
## dtype: object
##
```

```
## D:\miniconda3\envs\r-reticulate\lib\site-packages\numpy\core\_asarray.py:83: VisibleDeprecationWarni
##   return array(a, dtype, copy=False, order=order)
```

TD (Busan -> Seoul)

```
pol_eval = pd.DataFrame(np.zeros((len(states),2)),index=states,columns=['count','est'])

for episode_i in range(len(history_stop)):
    history_i = history_stop[episode_i]

    for j in range(0,len(history_i),3):
        pol_eval.loc[history_i[j]]['count']+=1
        current_cnt = pol_eval.loc[history_i[j]]['count']

        if j+3 < len(history_i):
            TD_tgt = float(history_i[j+2])+pol_eval.loc[history_i[j+3]]['est']
        else:
            TD_tgt = 0

        alpha = 1/current_cnt

        pol_eval.loc[history_i[j]]['est']+=alpha*(TD_tgt-pol_eval.loc[history_i[j]]['est'])

print(np.round(pol_eval.T,2))
```

```
##              0        A       B        C        D        E         1
## count  10000.0  5418.00  8771.0  9504.00  7329.00  5700.00  17423.00
## est        0.0    -6.34    -6.3    -5.81    -5.68    -5.74     -6.13
```