Tutorial 5 - Options Intro

Please complete this tutorial to get an overview of options and an implementation of SMDP Q-Learning and Intra-Option Q-Learning.

References:

Recent Advances in Hierarchical Reinforcement Learning is a strong recommendation for topics in HRL that was covered in class. Watch Prof. Ravi's lectures on moodle or nptel for further understanding the core concepts. Contact the TAs for further resources if needed.

```
'''
A bunch of imports, you don't have to worry about these
'''

import numpy as np
import random
import gym
#from gym.wrappers import Monitor
import glob
import io
import matplotlib.pyplot as plt
from IPython.display import HTML
```

```
'''
The environment used here is extremely similar to the openai gym ones.
At first glance it might look slightly different.
The usual commands we use for our experiments are added to this cell to aid you
work using this environment.
'''

#Setting up the environment
from gym.envs.toy_text.cliffwalking import CliffWalkingEnv
env = CliffWalkingEnv()

env.reset()

#Current State
print(env.s)

# 4x12 grid = 48 states
print ("Number of states:", env.nS)

# Primitive Actions
action = ["up", "right", "down", "left"]
#correspond to [0,1,2,3] that's actually passed to the environment

# either go left, up, down or right
print ("Number of actions that an agent can take:", env.nA)

# Example Transitions
rnd_action = random.randint(0, 3)
print ("Action taken:", action[rnd_action])
next_state, reward, is_terminal, t_prob,_ = env.step(rnd_action)
print ("Transition probability:", t_prob)
print ("Next state:", next_state)
print ("Reward recieved:", reward)
print ("Terminal state:", is_terminal)
#env.render()
```

```
36
Number of states: 48
Number of actions that an agent can take: 4
Action taken: down
Transition probability: False
Next state: 36
Reward recieved: -1
Terminal state: False
```

Options

We custom define very simple options here. They might not be the logical options for this settings deliberately chosen to visualise the Q Table better.

```
# We are defining two more options here
# Option 1 ["Away"] - > Away from Cliff (ie keep going up)
# Option 2 ["Close"] - > Close to Cliff (ie keep going down)
```

```
def Away(env,state):

    optdone = False
    optact = 0

    if (int(state/12) == 0):
        optdone = True

    return [optact,optdone]

def Close(env,state):

    optdone = False
    optact = 2

    if (int(state/12) == 2) or (int(state/12)==3):
        optdone = True

    return [optact,optdone]


'''
Now the new action space will contain
Primitive Actions: ["up", "right", "down", "left"]
Options: ["Away","Close"]
Total Actions :["up", "right", "down", "left", "Away", "Close"]
Corresponding to [0,1,2,3,4,5]
'''
```

```
'\nNow the new action space will contain\nPrimitive Actions: ["up", "right", "down", "left"]\nOptions:
["Away","Close"]\nTotal Actions :["up", "right", "down", "left", "Away", "Close"]\nCorresponding to
[0,1,2,3,4,5]\n'
```

Task 1

Complete the code cell below

```
#Q-Table: (States x Actions) === (env.ns(48) x total actions(6))
q_values_SMDP = np.zeros((48,6))

#Update_Frequency Data structure? Check TODO 4
freq_1=np.zeros((48,6))
# TODO: epsilon-greedy action selection function
actions=[0,1,2,3,4,5]
seed = 18
rg = np.random.RandomState(seed)
def egreedy_policy(q_values,state,epsilon=0.1):
  if rg.rand() < epsilon:
        return rg.choice(actions)
  else:
    return np.argmax(q_values[state])
```

Task 2

Below is an incomplete code cell with the flow of SMDP Q-Learning. Complete the cell and train the agent using SMDP Q-Learning algorithm. Keep the **final Q-table** and **Update Frequency** table handy (You'll need it in TODO 4)

```
#### SMDP Q-Learning

# Add parameters you might need here

q_values_SMDP = np.zeros((48,6))

#Update_Frequency Data structure? Check TODO 4
freq_1=np.zeros((48,6))
gamma = 0.9
alpha=0.4

# Iterate over 1000 episodes
for _ in range(1000):
    print("Epi",_)
    state = env.reset()
    done = False

    # While episode is not over
```

```
    while not done:

        # Choose action
        action = egreedy_policy(q_values_SMDP, state, epsilon=0.1)

        # Checking if primitive action
        if action < 4:
            # Perform regular Q-Learning update for state-action pair
            next_state, reward, done,_ ,info= env.step(action)
            q_values_SMDP[state][action]=(1-alpha)*(q_values_SMDP[state][action])+(alpha)*(reward+gamma*(np.max(q_values_SMDP[next_state]
            freq_1[state,action]+=1
            state=next_state

        # Checking if action chosen is an option
        reward_bar = 0
        if action == 4: # action => Away option
            initial_state=np.copy(state)
            optdone = False
            cnt=0

            while (optdone == False):

                # Think about what this function might do?
                optact,optdone = Away(env,state)
                next_state, reward, done,_,info = env.step(optact)

                #TO check if the next state is the termination state
                optact,optdone = Away(env,next_state)

                # Is this formulation right? What is this term? No. This is not the discounted return formulation.Here, the first reward
                #This is a formulation used to calculate discount return when rewards are in reverse order a trajectory.
                #reward_bar=gamma*reward_bar+reward
                reward_bar = reward_bar + np.power(gamma,cnt)*reward
                cnt+=1

                # Complete SMDP Q-Learning Update
                # Remember SMDP Updates. When & What do you update? After the ermination state is reached we will update

                state = next_state

            q_values_SMDP[initial_state, action] = (1-alpha)* (q_values_SMDP[initial_state, action])+alpha*(reward_bar + (np.power(gamma,
            freq_1[state,action]+=1

        if action == 5: # action => Close option
            initial_state=np.copy(state)
            optdone = False
            cnt=0

            while (optdone == False):

                # Think about what this function might do?
                optact,optdone = Away(env,state)
                next_state, reward, done,_,info = env.step(optact)
                #TO check if the next state is the termination state
                optact,optdone = Away(env,next_state)

                 # Is this formulation right? What is this term? No. This is not the discounted return formulation.Here, the first reward
                #This is a formulation used to calculate discount return when rewards are in reverse order a trajectory.
                #reward_bar=gamma*reward_bar+reward
                reward_bar = reward_bar + np.power(gamma,cnt)*reward
                cnt+=1

                # Complete SMDP Q-Learning Update
                # Remember SMDP Updates. When & What do you update?

                state = next_state

            q_values_SMDP[initial_state, action] = (1-alpha)* (q_values_SMDP[initial_state, action])+alpha*(reward_bar + (np.power(gamma,
            freq_1[state,action]+=1
```

```
Epi 492
Epi 493
Epi 494
Epi 495
Epi 496
Epi 497
Epi 498
Epi 499
Epi 500
Epi 501
Epi 502
Epi 503
Epi 504
Epi 505
Epi 506
Epi 507
Epi 508
Epi 509
Epi 510
Epi 511
Epi 512
Epi 513
Epi 514
Epi 515
Epi 516
Epi 517
Epi 518
Epi 519
Epi 520
Epi 521
Epi 522
Epi 523
Epi 524
Epi 525
Epi 526
Epi 527
Epi 528
Epi 529
Epi 530
Epi 531
Epi 532
Epi 533
Epi 534
Epi 535
Epi 536
```

```
np.power(3,4)
```

```
81
```

Task 3

Using the same options and the SMDP code, implement Intra Option Q-Learning (In the code cell below). You *might not* always have to search through options to find the options with similar policies, think about it. Keep the **final Q-table** and **Update Frequency** table handy (You'll need it in TODO 4)

```
#### Intra-Option Q-Learning

# Add parameters you might need here

q_values_intra = np.zeros((48,6))

#Update_Frequency Data structure? Check TODO 4
freq_2=np.zeros((48,6))
gamma = 0.9
alpha=0.4

# Iterate over 1000 episodes
for _ in range(1000):
    print("Epi",_)
    state = env.reset()
    done = False

    # While episode is not over
    while not done:

        # Choose action
        action = egreedy_policy(q_values_intra, state, epsilon=0.1)

        # Checking if primitive action
        if action < 4:
            # Perform regular Q-Learning update for state-action pair
```

```
            next_state, reward, done,_ ,info= env.step(action)
            q_values_intra[state][action]=(1-alpha)*(q_values_intra[state][action])+(alpha)*(reward+gamma*(np.max(q_values_intra[next_sta
            freq_2[state,action]+=1
            state=next_state

        # Checking if action chosen is an option

        if action == 4 or action==5: # action => Away,CLose option (Coupled into a single if statement)
            initial_state=np.copy(state)
            optdone = False
            cnt=0

            while (optdone == False):

                # Think about what this function might do?
                optact,optdone = Away(env,state)
                next_state, reward, done,_,info = env.step(optact)
                q_values_intra[state, optact] += alpha*(reward + gamma*np.max(q_values_intra[next_state]) - q_values_intra[state, optact]
                freq_2[state,action]+=1

                optact,optdone = Away(env,next_state)

                beta=int(optdone)

                U_s_a=(1-beta)*(q_values_intra[next_state,action]) + beta*(np.max(q_values_intra[next_state]))
                q_values_intra[state,action]=(1-alpha)*(q_values_intra[state,action])+ alpha*(reward+gamma*(U_s_a))
                freq_2[state,action] += 1




                state = next_state
```

```
Epi 0
Epi 1
Epi 2
Epi 3
Epi 4
Epi 5
Epi 6
Epi 7
Epi 8
Epi 9
Epi 10
Epi 11
Epi 12
Epi 13
Epi 14
Epi 15
Epi 16
Epi 17
Epi 18
Epi 19
Epi 20
Epi 21
Epi 22
Epi 23
Epi 24
Epi 25
Epi 26
Epi 27
Epi 28
Epi 29
Epi 30
Epi 31
Epi 32
Epi 33
Epi 34
Epi 35
Epi 36
Epi 37
Epi 38
Epi 39
Epi 40
Epi 41
Epi 42
Epi 43
Epi 44
Epi 45
Epi 46
Epi 47
Epi 48
Epi 49
```

```
Epi 50
Epi 51
Epi 52
Epi 53
Epi 54
Epi 55
Epi 56
```

Task 4

Compare the two Q-Tables and Update Frequencies and provide comments.

```
# Use this cell for Task 4 Code
import pandas as pd
print("Q Table for intra option Q Learning")
print(pd.DataFrame(q_values_intra,columns=["up", "right", "down", "left", "Away", "Close"]))
print("-----------------------------------------------------------------------")
print("-----------------------------------------------------------------------")
print("-----------------------------------------------------------------------")
print("Q Table for SMDP Q Learning")
print(pd.DataFrame(q_values_SMDP,columns=["up", "right", "down", "left", "Away", "Close"]))
```

```
46  0.000000    0.000000    0.000000  0.000000  0.000000  0.000000
47  0.000000    0.000000    0.000000  0.000000  0.000000  0.000000
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
Q Table for SMDP Q Learning
          up       right        down      left      Away     Close
0  -7.836496   -7.712318   -7.712317 -7.819231 -7.855008 -7.927853
1  -7.585884   -7.458134   -7.458134 -7.489827 -7.608444 -7.587029
2  -7.278951   -7.175705   -7.175705 -7.597302 -7.387095 -7.386828
3  -7.032321   -6.861894   -6.861894 -7.127332 -7.105683 -7.106193
4  -6.698412   -6.513216   -6.513216 -7.023656 -6.617134 -6.694010
5  -6.332924   -6.125795   -6.125795 -6.181741 -6.397344 -6.429473
6  -6.004221   -5.695328   -5.695328 -6.339973 -6.066937 -5.980369
7  -5.573726   -5.217031   -5.217031 -5.404798 -5.650860 -5.672036
8  -5.043737   -4.685590   -4.685590 -5.596694 -5.036382 -5.006659
9  -4.508025   -4.095100   -4.095100 -5.152847 -4.486582 -4.590217
10 -4.032050   -3.439000   -3.439000 -4.576491 -3.517882 -3.580210
11 -2.786104   -3.200795   -2.710000 -3.363008 -3.036705 -3.354385
12 -7.778745   -7.458132   -7.458132 -7.568982 -7.622461 -7.529285
13 -7.326800   -7.175705   -7.175705 -7.262449 -7.343472 -7.200412
14 -7.220709   -6.861894   -6.861894 -7.306747 -7.355202 -6.873555
15 -7.063932   -6.513216   -6.513216 -7.017617 -6.728482 -7.098983
16 -6.508723   -6.125795   -6.125795 -6.263106 -6.563491 -6.634372
17 -6.352826   -5.695328   -5.695328 -6.051694 -6.272494 -6.389445
18 -5.892751   -5.217031   -5.217031 -5.902850 -5.975798 -5.596308
19 -5.535034   -4.685590   -4.685590 -5.373940 -5.623400 -5.400005
20 -5.101861   -4.095100   -4.095100 -5.064953 -4.872687 -4.823958
21 -4.537471   -3.439000   -3.439000 -4.549108 -3.654941 -4.335927
22 -3.868257   -2.710000   -2.710000 -3.707590 -3.441766 -3.658205
23 -2.564537   -2.630393   -1.900000 -2.958278 -3.126545 -3.351441
24 -7.710680   -7.175705   -7.712192 -7.458132 -8.145982 -8.146933
25 -7.457966   -6.861894 -106.712203 -7.457365 -7.940889 -7.941033
26 -7.175425   -6.513216 -106.710862 -7.175520 -7.712306 -7.712260
27 -6.859558   -6.125795 -106.712125 -6.860879 -7.458124 -7.452014
28 -6.511757   -5.695328 -106.705496 -6.513206 -7.175362 -7.175067
29 -6.125505   -5.217031 -106.709847 -6.125712 -6.861868 -6.861834
30 -5.693583   -4.685590 -106.038811 -5.695247 -6.513116 -6.512295
31 -5.212560   -4.095100 -106.463035 -5.217021 -6.121060 -6.124338
32 -4.682033   -3.439000 -106.686667 -4.685477 -5.693384 -5.695190
33 -4.094823   -2.710000 -106.046539 -4.094899 -5.156223 -5.216383
34 -3.438944   -1.900000 -106.678082 -3.436181 -4.683594 -4.685108
35 -2.709985   -1.899999   -1.000000 -2.709990 -4.083223 -4.095063
36 -7.458134 -106.710431   -7.712289 -7.712320 -8.331766 -8.330178
37  0.000000    0.000000    0.000000  0.000000  0.000000  0.000000
38  0.000000    0.000000    0.000000  0.000000  0.000000  0.000000
39  0.000000    0.000000    0.000000  0.000000  0.000000  0.000000
40  0.000000    0.000000    0.000000  0.000000  0.000000  0.000000
41  0.000000    0.000000    0.000000  0.000000  0.000000  0.000000
42  0.000000    0.000000    0.000000  0.000000  0.000000  0.000000
43  0.000000    0.000000    0.000000  0.000000  0.000000  0.000000
44  0.000000    0.000000    0.000000  0.000000  0.000000  0.000000
45  0.000000    0.000000    0.000000  0.000000  0.000000  0.000000
46  0.000000    0.000000    0.000000  0.000000  0.000000  0.000000
47  0.000000    0.000000    0.000000  0.000000  0.000000  0.000000
/usr/local/lib/python3.9/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transfor
  and should_run_async(code)
```

```
print("Frequency Table for intra option Q Learning")
print(pd.DataFrame(freq_1,columns=["up", "right", "down", "left", "Away", "Close"]))
print("-----------------------------------------------------------------------")
print("-----------------------------------------------------------------------")
```

```
print("------------------------------------------------------------------------")
print("Frequency Table for SMDP Q Learning")
print(pd.DataFrame(freq_2,columns=["up", "right", "down", "left", "Away", "Close"]))
```

```
Frequency Table for intra option Q Learning
        up   right    down  left   Away  Close
0     39.0    91.0   133.0  38.0  121.0  134.0
1     35.0   103.0   112.0  26.0   86.0   87.0
2     33.0   116.0   111.0  27.0   92.0   83.0
3     32.0   122.0   108.0  24.0   84.0   73.0
4     28.0   117.0    99.0  24.0   71.0   68.0
5     25.0   128.0    97.0  18.0   71.0   72.0
6     24.0   121.0    92.0  19.0   65.0   57.0
7     21.0   114.0    86.0  15.0   56.0   55.0
8     18.0   108.0    80.0  17.0   47.0   51.0
9     16.0    86.0    76.0  16.0   33.0   45.0
10    15.0    63.0    76.0  14.0   34.0   37.0
11     8.0    10.0   110.0   9.0   28.0   44.0
12    24.0    78.0    59.0  35.0    0.0    0.0
13    21.0   102.0    70.0  25.0    0.0    0.0
14    21.0   114.0    74.0  25.0    0.0    0.0
15    21.0   111.0    75.0  24.0    0.0    0.0
16    17.0   112.0    75.0  18.0    0.0    0.0
17    17.0   115.0    75.0  17.0    0.0    0.0
18    15.0   112.0    75.0  16.0    0.0    0.0
19    14.0   105.0    73.0  14.0    0.0    0.0
20    13.0    95.0    74.0  14.0    0.0    0.0
21    11.0    87.0    76.0  13.0    0.0    0.0
22     9.0    82.0    80.0   9.0    0.0    0.0
23     5.0    10.0   189.0   7.0    0.0    0.0
24    47.0  1095.0    38.0  55.0    0.0    0.0
25    44.0  1034.0    27.0  33.0    0.0    0.0
26    37.0   979.0    22.0  34.0    0.0    0.0
27    29.0   951.0    26.0  30.0    0.0    0.0
28    29.0   924.0    19.0  37.0    0.0    0.0
29    32.0   887.0    21.0  32.0    0.0    0.0
30    25.0   883.0    10.0  30.0    0.0    0.0
31    21.0   881.0    12.0  32.0    0.0    0.0
32    19.0   876.0    17.0  26.0    0.0    0.0
33    23.0   883.0    10.0  24.0    0.0    0.0
34    24.0   897.0    16.0  17.0    0.0    0.0
35    25.0    29.0  1000.0  26.0    0.0    0.0
36  1161.0    22.0    53.0  61.0    0.0    0.0
37     0.0     0.0     0.0   0.0    0.0    0.0
38     0.0     0.0     0.0   0.0    0.0    0.0
39     0.0     0.0     0.0   0.0    0.0    0.0
40     0.0     0.0     0.0   0.0    0.0    0.0
41     0.0     0.0     0.0   0.0    0.0    0.0
42     0.0     0.0     0.0   0.0    0.0    0.0
43     0.0     0.0     0.0   0.0    0.0    0.0
44     0.0     0.0     0.0   0.0    0.0    0.0
45     0.0     0.0     0.0   0.0    0.0    0.0
46     0.0     0.0     0.0   0.0    0.0    0.0
47     0.0     0.0     0.0   0.0    0.0    0.0
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
Frequency Table for SMDP Q Learning
        up   right    down  left   Away  Close
0      6.0    85.0    88.0  40.0   78.0   76.0
1      3.0    89.0    78.0  28.0   70.0   70.0
```

```
from pandas.core.api import DataFrame
print("SMDP")
print("Total No of Updates:",np.sum(freq_1))
print(pd.DataFrame(np.sum(freq_1,axis=0).reshape(1,6),columns=["up", "right", "down", "left", "Away", "Close"]))
print("____________________________________________________________________________")
print()
print("Inta Option Q Learning")
print("Total No of Updates:",np.sum(freq_2))
print(pd.DataFrame(np.sum(freq_2,axis=0).reshape(1,6),columns=["up", "right", "down", "left", "Away", "Close"]))
print("____________________________________________________________________________")
```

```
SMDP
Total No of Updates: 20582.0
       up    right    down   left   Away  Close
0  1998.0  12643.0  3446.0  901.0  788.0  806.0
____________________________________________________________________________

Inta Option Q Learning
Total No of Updates: 22904.0
       up    right    down   left    Away   Close
0  1451.0  12648.0  3223.0  876.0  2424.0  2282.0
____________________________________________________________________________
/usr/local/lib/python3.9/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_c
  and should_run_async(code)
```

```
print(pd.DataFrame(((q_values_SMDP-q_values_intra)))) #Difference between Action-Value
```

```
           0             1             2         3         4         5
0   0.093210 -2.748877e-05 -2.433917e-05  0.055936 -0.022251 -0.091216
1   0.070885 -9.815021e-06 -6.601484e-06  0.169251 -0.120688 -0.016572
2   0.158585 -6.691029e-07 -5.489984e-07 -0.189041  0.015506 -0.100062
3   0.130935 -8.954081e-08 -7.532411e-08  0.171493 -0.128855 -0.031624
4   0.145608 -2.128413e-08 -1.302457e-08  0.051151  0.149937  0.000054
5   0.152399 -4.101712e-09 -4.676363e-09  0.044102  0.010053 -0.076760
6   0.112887  1.691181e-10 -6.527525e-10 -0.174363 -0.272935 -0.094953
7   0.050462  4.203553e-10  3.196332e-10 -0.171905 -0.255052 -0.277048
8   0.139877  1.358975e-10  1.137064e-10 -0.263011 -0.011032 -0.073630
9   0.157395  5.467093e-11  5.403589e-11 -0.934774  0.075314 -0.290715
10  0.052572  4.449774e-12  4.345857e-12 -0.902473  0.405235  0.411667
11  0.626272 -2.079254e-01  0.000000e+00  0.313944  0.196754 -0.307148
12  0.162290 -3.550539e-06 -3.555339e-06 -0.064852  0.318347  0.411694
13  0.385423 -1.228797e-07 -8.606550e-08  0.054708  0.363453  0.510692
14  0.237422 -9.716089e-09 -1.052688e-08 -0.193664  0.102886  0.584261
15  0.111771  2.584803e-10  5.696474e-10 -0.133299  0.446400  0.076025
16  0.353171  3.229061e-11  4.584511e-11  0.276138  0.298373  0.227190
17  0.160389  1.661693e-11  8.100187e-12  0.421183  0.240699  0.123734
18  0.233044  5.891287e-12  3.721468e-12  0.087062  0.149542  0.529447
19  0.160294  2.066791e-12  1.383782e-12  0.210791  0.071901  0.295274
20  0.115170  3.570477e-13  2.993161e-13 -0.042170  0.344339  0.392961
21  0.148119  1.509903e-14  6.217249e-15 -1.099056  1.030648  0.348409
22  0.226843  0.000000e+00  0.000000e+00 -0.019143  0.653332  0.436861
23  0.874463 -1.139215e-01  0.000000e+00  0.293779  0.312429  0.087532
24  0.001631  0.000000e+00  1.226003e-04  0.000001 -0.000351 -0.000480
25  0.000167  0.000000e+00 -1.378797e-03  0.000557 -0.031053 -0.008064
26  0.000279  0.000000e+00 -8.782257e-02  0.000099 -0.000331 -0.003673
27  0.002336  0.000000e+00 -6.815042e-03  0.000824 -0.006982 -0.002676
28  0.001459  0.000000e+00 -4.847677e-03 -0.000114  0.000013 -0.004554
29  0.000290  0.000000e+00  2.086284e-03 -0.001746 -0.000440 -0.000799
30  0.001745  0.000000e+00  5.858062e-01  0.000070 -0.008947  0.000168
31  0.004471  0.000000e+00  1.032495e-02 -0.001412  0.003348  0.000476
32  0.003557  0.000000e+00  2.324795e-02 -0.000483  0.001787 -0.003680
33  0.000277  0.000000e+00  4.252569e-01  0.000093  0.060749 -0.025658
34  0.000056  0.000000e+00 -5.270400e-02  0.002801  0.001909 -0.000714
35  0.000015 -1.341490e-04  0.000000e+00 -0.000935  0.011202 -0.000647
36  0.000000  1.871738e-03  3.081038e-05 -0.000012 -0.019119 -0.003180
37  0.000000  0.000000e+00  0.000000e+00  0.000000  0.000000  0.000000
38  0.000000  0.000000e+00  0.000000e+00  0.000000  0.000000  0.000000
39  0.000000  0.000000e+00  0.000000e+00  0.000000  0.000000  0.000000
40  0.000000  0.000000e+00  0.000000e+00  0.000000  0.000000  0.000000
41  0.000000  0.000000e+00  0.000000e+00  0.000000  0.000000  0.000000
42  0.000000  0.000000e+00  0.000000e+00  0.000000  0.000000  0.000000
43  0.000000  0.000000e+00  0.000000e+00  0.000000  0.000000  0.000000
44  0.000000  0.000000e+00  0.000000e+00  0.000000  0.000000  0.000000
45  0.000000  0.000000e+00  0.000000e+00  0.000000  0.000000  0.000000
46  0.000000  0.000000e+00  0.000000e+00  0.000000  0.000000  0.000000
47  0.000000  0.000000e+00  0.000000e+00  0.000000  0.000000  0.000000
```

Use this text cell for your comments - Task 4
Ans:Since the options are Markov here, we have performed the intra option Q-learning update. We see the total updates performed in intra options Q-Learning is 22904 and in SMDP Q-Learning is 22508.

Thus intra option q learning has more updates for the Away and Close options which allows learning useful information before an option terminates and can be used for multiple options simultaneously as it is off policy.

Since the difference of Q values from both SMDP and intra options are small(negigible in many state action values), we could not observe considerable difference of Intra Option method over SMDPs. However, if we consider problems with bottlenecks such as the hallways discussed in the class, Intra option q learning would provide better results than SMDP as they help explore structure inside the options, whereas SMDPs have to wait till termination state is reached to perform an update

0s completed at 11:01 PM

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.