Model_free_evaluation

August 16, 2020

0.0.1 Policy Evaluation: Model Free

```
In [1]: from gridworld import GridWorld
        import numpy as np
        import time
```

```
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
```

```
In [5]: world=\
            """
            wwwwwwwwww
            wa       w
            w    wwwww
            wwww     w
            w        w
            w        w
            w        w
            w g  wwwww
            w        w
            wwwwwwwwww
            """
        env=GridWorld(world)
        env._max_epi_step=200
```

```
In [6]: env.render()
```

```
In [7]: env.close()
```

```
In [8]: policy=np.random.randint(0,4,env.state_count)
```

```
In [9]: policy
```

```
Out[9]: array([2, 1, 3, 2, 0, 1, 3, 1, 0, 2, 0, 3, 0, 1, 3, 1, 2, 0, 2, 0, 2, 3,
               2, 3, 0, 1, 1, 0, 2, 3, 0, 3, 0, 3, 2, 2, 2, 3, 1, 1, 2, 3, 1, 3,
               3, 3, 3, 3, 3, 0, 2, 1, 0])
```

0.0.2 Monte-Carlo Policy Evaluation

Offline Policy evaluation

```
In [12]: def get_episodes(policy):
             episodes=[]
             curr_state=env.reset()
             done=False

             while not done:
                 action=policy[curr_state]
                 next_state,reward,done,info=env.step(action)
                 episodes.append((curr_state,reward))
                 curr_state=next_state

             return episodes


In [13]: episodes=get_episodes(policy)

In [15]: len(episodes)

Out[15]: 201

In [20]: def get_returns(episode):
             Gt=[]
             for i,(s,r) in enumerate(episode):
                 sum_1=r
                 future=episode[i+1:]   ###From one to last
                 for j,(s_f,r_f) in enumerate(future):
                     sum_1=sum_1+(0.99*(j+1)*r_f)
                 Gt.append((s,sum_1))   ##Saving sum of all possible rewards for a particular
             return Gt

In [21]: experience=[get_returns(get_episodes(policy)) for i in range(1000)]

In [23]: G_total=np.zeros(env.state_count)
         N_total=np.zeros(env.state_count)

In [28]: len(experience)

Out[28]: 1000
```

0.0.3 For every Visit

```
In [25]: for ep in experience:
             for s,G in ep:
                 G_total[s]+=G
                 N_total[s]+=1

         V=G_total/N_total
```

```
/home/abhijit/.local/lib/python3.6/site-packages/ipykernel_launcher.py:6: RuntimeWarning: inval
```

```
In [26]: V
```

```
Out[26]: array([-10093.65304279,  -8002.93730769,  -6480.48408897,  -6715.04354192,
                 -4500.09422642,  -3678.22080437,  -2947.77532829,  -2466.18768935,
                 -7812.69103167,  -7757.76471611,  -7115.28665119,  -6928.90760976,
                 -4934.86728033,  -3752.86424311,  -3751.21856302,  -2979.65359087,
                 -2887.94702703,  -1563.79535714,  -1623.89741935,  -3018.07146625,
                 -3108.0975378 ,  -3651.25296548,  -3600.8762256 ,  -2870.2488164 ,
                 -2658.14205438,  -1329.58496418,  -1634.12033981,  -2661.2135443 ,
                 -2620.64788462,  -3359.36875   ,  -3216.74982143,  -3432.03592857,
                 -3516.21960352,  -1300.15926302,  -1408.63492754,  -1701.60314286,
                 -1852.003     ,  -2832.829     ,  -1484.50857143,  -3314.71403846,
                 -3235.38823529,  -1034.48585185,            nan,   -951.884     ,
                 -1725.3325    ,  -2461.975     ,            nan,     38.6       ,
                           nan,            nan,            nan,            nan,
                           nan])
```

```
In [27]: N_total
```

```
Out[27]: array([2.7925e+04, 6.3180e+03, 2.6706e+04, 5.8330e+03, 1.3250e+03,
                1.2084e+04, 8.7270e+03, 7.7900e+03, 3.3819e+04, 3.4520e+04,
                4.7390e+03, 4.5100e+03, 7.1700e+02, 4.4260e+03, 2.0390e+03,
                1.1390e+03, 1.1100e+02, 9.2400e+02, 8.9900e+02, 1.2890e+03,
                1.3890e+03, 4.2590e+03, 4.6100e+02, 1.0730e+03, 3.3100e+02,
                1.3960e+03, 2.0600e+02, 4.7400e+02, 4.6800e+02, 3.3600e+02,
                5.6000e+01, 1.4000e+02, 2.2700e+02, 1.5740e+03, 3.4500e+02,
                1.0500e+02, 4.0000e+01, 3.0000e+01, 7.7000e+01, 1.5600e+02,
                1.7000e+01, 1.3500e+02, 0.0000e+00, 1.0000e+01, 4.0000e+00,
                6.0000e+00, 0.0000e+00, 1.0000e+00, 0.0000e+00, 0.0000e+00,
                0.0000e+00, 0.0000e+00, 0.0000e+00])
```

0.0.4 For FIrst visit

```
In [29]: for ep in experience:
             seen=[]
             for s,G in ep:
                 if s not in seen:

                     G_total[s]+=G
                     N_total[s]+=1
                     seen.append(s)

         V=G_total/N_total
```

```
/home/abhijit/.local/lib/python3.6/site-packages/ipykernel_launcher.py:10: RuntimeWarning: inva
  # Remove the CWD from sys.path while we load stuff.
```

```
In [30]: V=G_total/N_total
```

```
/home/abhijit/.local/lib/python3.6/site-packages/ipykernel_launcher.py:1: RuntimeWarning: inval
  """Entry point for launching an IPython kernel.
```

```
In [31]: V
```

```
Out[31]: array([-10420.82006154,  -9115.6442268 ,  -6666.44763728,  -7416.81957384,
                -4952.35985128,  -3751.87960058,  -3025.54980149,  -2518.12888889,
                -8093.26483468,  -8034.84459275,  -8031.27569182,  -7789.73119534,
                -5311.46207207,  -3883.37347213,  -3847.57633077,  -3082.80748201,
                -3026.12833333,  -1630.725   ,  -1702.72477477,  -3162.26259386,
                -3264.86270253,  -3778.18672723,  -3821.15568807,  -2981.13730025,
                -2654.07428571,  -1380.0204008 ,  -1712.4262    ,  -2820.64478992,
                -2768.42818336,  -3635.71832653,  -3402.82623762,  -3655.51028302,
                -3525.57354582,  -1345.38044723,  -1516.74695556,  -1856.02537572,
                -1853.68875   ,  -2990.872     ,  -1508.968     ,  -3308.13482759,
                -3617.47      ,  -1064.03409756,            nan,  -1060.01888889,
                -1725.3325    ,  -2165.68      ,            nan,     38.6       ,
                          nan,            nan,            nan,            nan,
                          nan])
```

Online Evauation

```
In [40]: V_total=np.zeros(env.state_count)
         N_total=np.zeros(env.state_count)
```

0.0.5 For Every Visit

```
In [38]: for ep in experience:
             for s,G in ep:
                 V_total[s]=V_total[s]+(1/(N_total[s]+1))*(G-V_total[s])
                 N_total[s]+=1
```

```
In [39]: V_total
```

```
Out[39]: array([-10093.65304279,  -8002.93730769,  -6480.48408897,  -6715.04354192,
                -4500.09422642,  -3678.22080437,  -2947.77532829,  -2466.18768935,
                -7812.69103167,  -7757.76471611,  -7115.28665119,  -6928.90760976,
                -4934.86728033,  -3752.86424311,  -3751.21856302,  -2979.65359087,
                -2887.94702703,  -1563.79535714,  -1623.89741935,  -3018.07146625,
```

```
                -3108.0975378 ,  -3651.25296548,  -3600.8762256 ,  -2870.2488164 ,
                -2658.14205438,  -1329.58496418,  -1634.12033981,  -2661.2135443 ,
                -2620.64788462,  -3359.36875   ,  -3216.74982143,  -3432.03592857,
                -3516.21960352,  -1300.15926302,  -1408.63492754,  -1701.60314286,
                -1852.003     ,  -2832.829     ,  -1484.50857143,  -3314.71403846,
                -3235.38823529,  -1034.48585185,      0.        ,   -951.884     ,
                -1725.3325    ,  -2461.975     ,      0.        ,     38.6       ,
                    0.        ,      0.        ,      0.        ,      0.        ,
                    0.        ])
```

0.0.6 For First Visit

```
In [41]: for ep in experience:
             seen=[]
             for s,G in ep:
                 if s not in seen:
                     V_total[s]=V_total[s]+(1/(N_total[s]+1))*(G-V_total[s])
                     N_total[s]+=1
                     seen.append(s)
```

```
In [42]: V_total
```

```
Out[42]: array([-19556.95906   , -16461.60275862, -12172.37060976, -11842.1812973 ,
                 -6635.65185393,  -6416.82838323,  -5366.02506897,  -4285.03694323,
                -17658.51225806, -17609.21630631, -12438.25240609, -11759.37779141,
                 -6268.9743617 ,  -6130.97599222,  -4976.73431034,  -4131.84571429,
                 -3326.87588235,  -2249.1613    ,  -2411.3827    ,  -4218.29875   ,
                 -4404.89837696,  -5906.57606299,  -4347.31544041,  -4006.85577586,
                 -2625.42680851,  -2077.1280198 ,  -1884.03265957,  -3445.19363636,
                 -3340.00818182,  -4238.66285714,  -3634.388     ,  -4090.04375   ,
                 -3614.04625   ,  -2036.43038835,  -1871.97219048,  -2094.47147059,
                 -1855.7959375 ,  -3180.5236    ,  -1653.84307692,  -3251.115     ,
                 -4158.7525    ,  -1121.02      ,      0.        ,  -1195.1875    ,
                 -1725.3325    ,  -1810.126     ,      0.        ,     38.6       ,
                     0.        ,      0.        ,      0.        ,      0.        ,
                     0.        ])
```

0.0.7 TD Learning

```
In [55]: def get_states(policy):
             curr_state=env.reset()
             while True:
                 action=policy[curr_state]
                 next_state,reward,done, info=env.step(action)
                 yield(curr_state,reward,next_state)
                 if done:
```

```
                curr_state=env.reset()
            else:
                curr_state=next_state
```

In [56]:
```
generator=get_states(policy)
```

In [57]:
```
generator
```

Out[57]:
```
<generator object get_states at 0x7fc139d7cdb0>
```

In [52]:
```
alpha=0.001
gamma=0.99
v=np.zeros(env.state_count)
```

In [53]:
```
for step in range(100000):
    s,r,s_prime=next(generator)
    v[s]=v[s]+alpha*((r+gamma*v[s_prime])-v[s])
```

In [54]:
```
v
```

Out[54]:
```
array([-1.11561982e+01, -8.95664619e+00, -8.19675733e+00, -5.94938312e+00,
       -1.78903331e+00, -5.16975781e+00, -4.88849630e+00, -4.67592196e+00,
       -1.20198327e+01, -1.15783029e+01, -3.54735003e+00, -3.89243334e+00,
       -5.69364502e-01, -1.88358832e+00, -1.00849532e+00, -6.36564067e-01,
       -1.02250312e-01, -3.82527245e-01, -3.67586368e-01, -5.69740568e-01,
       -6.27359074e-01, -1.80729658e+00, -3.84005541e-01, -5.92196860e-01,
       -2.47681048e-01, -6.95390399e-01, -9.75145818e-02, -2.55218395e-01,
       -2.54244127e-01, -2.47949787e-01, -3.42163179e-02, -1.02393031e-01,
       -1.68981330e-01, -7.43675133e-01, -1.07732827e-01, -3.79925815e-02,
       -1.41306465e-02, -2.07355639e-02, -7.67481614e-02, -2.17438642e-01,
       -2.56255342e-02, -2.29252496e-02,  0.00000000e+00, -3.04624739e-03,
        0.00000000e+00, -4.02869618e-03,  0.00000000e+00,  0.00000000e+00,
        0.00000000e+00,  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
        0.00000000e+00])
```

In [ ]:
```
```