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 [2]: 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 [3]: env.render()
```

```
In [4]: env.close()
```

```
In [5]: policy=np.random.randint(0,4,env.state_count)
```

```
In [6]: policy
```

```
Out[6]: array([2, 3, 3, 3, 3, 2, 3, 1, 3, 0, 3, 2, 3, 3, 1, 2, 1, 0, 2, 2, 0, 3,
               0, 0, 3, 3, 1, 2, 0, 3, 2, 1, 1, 3, 3, 1, 2, 3, 2, 1, 1, 2, 1, 1,
               1, 3, 1, 1, 0, 0, 0, 0, 3])
```

0.0.2 Monte-Carlo Policy Evaluation

Offline Policy evaluation

```
In [7]: 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 [8]: episodes=get_episodes(policy)
```

```
In [9]: len(episodes)
```

```
Out[9]: 201
```

```
In [10]: 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 [11]: experience=[get_returns(get_episodes(policy)) for i in range(1000)]
```

```
In [12]: G_total=np.zeros(env.state_count)
         N_total=np.zeros(env.state_count)
```

```
In [13]: len(experience)
```

```
Out[13]: 1000
```

0.0.3 For every Visit

```
In [14]: 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 [15]: V
```

```
Out[15]: array([-65.79310333, -59.65157961, -55.20022204, -52.22472432,
                -49.4485016 , -48.5127885 , -42.7798297 , -39.16084703,
                -65.64843289, -58.55140148, -55.51843199, -52.50676132,
                -51.623454  , -46.21130525, -50.33909952, -54.40860588,
                -47.336966  , -37.79447325, -40.46779492, -51.30226166,
                -53.06919078, -51.92335275, -55.45524803, -51.74580787,
                -47.90666584, -32.04353368, -34.38883569, -60.52133558,
                -53.97824806, -54.29078446, -57.76069737, -45.05271693,
                -51.18924734, -31.44082065, -25.06050503, -46.11206997,
                -46.65094927, -46.02980948, -39.83036137, -45.81979561,
                -48.06673296, -36.7109775 ,          nan, -44.16514498,
                         nan, -37.34386773, -37.97042905, -41.77592628,
                         nan,          nan,          nan,          nan,
                         nan])
```

```
In [16]: N_total
```

```
Out[16]: array([4.5447e+04, 3.2765e+04, 5.2855e+04, 2.4634e+04, 2.1511e+04,
                1.6830e+03, 1.1520e+03, 1.2620e+03, 3.6520e+03, 3.1330e+03,
                8.7040e+03, 2.5810e+03, 2.0100e+02, 1.8000e+02, 3.0000e+01,
                1.8000e+01, 1.8800e+02, 9.0000e+01, 8.3000e+01, 8.0000e+00,
                1.7000e+01, 3.9000e+01, 3.2000e+01, 4.3000e+01, 2.1600e+02,
                1.6000e+01, 2.1000e+01, 2.0000e+00, 4.0000e+00, 1.4000e+01,
                2.0000e+00, 1.2000e+01, 3.1000e+01, 4.0000e+00, 1.7000e+01,
                1.0000e+00, 1.0000e+00, 9.0000e+00, 1.3000e+01, 1.3600e+02,
                1.6400e+02, 1.0000e+00, 0.0000e+00, 2.0000e+00, 0.0000e+00,
                1.0000e+00, 1.0000e+00, 1.1000e+01, 0.0000e+00, 0.0000e+00,
                0.0000e+00, 0.0000e+00, 0.0000e+00])
```

0.0.4 For FIrst visit

```
In [17]: 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 [18]: 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 [19]: V
```

```
Out[19]: array([-66.24389506, -60.37416803, -55.68114355, -53.00740476,
                -49.80580146, -50.21830037, -42.99921988, -39.43298195,
                -68.04463881, -62.21967081, -57.85334074, -55.48681329,
                -51.90185405, -47.35722016, -52.28711331, -57.42148104,
                -47.88203673, -38.24177017, -40.85165623, -48.81681641,
                -54.05687658, -53.32326   , -57.57886578, -54.00125489,
                -48.56340729, -34.4025413 , -34.86948567, -65.60033552,
                -53.97824806, -56.04383889, -57.76069737, -46.50097869,
                -52.12151021, -36.89362996, -24.85608703, -46.11206997,
                -46.65094927, -45.11408301, -41.68994653, -46.60327574,
                -48.40255458, -36.7109775 ,          nan, -44.63267913,
                         nan, -37.34386773, -37.97042905, -42.04609298,
                         nan,          nan,          nan,          nan,
                         nan])
```

Online Evauation

```
In [20]: V=np.zeros(env.state_count)
         N_total=np.zeros(env.state_count)
```

0.0.5 For Every Visit

```
In [21]: for ep in experience:
             for s,G in ep:
                 V[s]=V[s]+(1/(N_total[s]+1))*(G-V[s])
                 N_total[s]+=1
```

```
In [22]: V
```

```
Out[22]: array([-65.79310333, -59.65157961, -55.20022204, -52.22472432,
                -49.4485016 , -48.5127885 , -42.7798297 , -39.16084703,
                -65.64843289, -58.55140148, -55.51843199, -52.50676132,
                -51.623454  , -46.21130525, -50.33909952, -54.40860588,
                -47.336966  , -37.79447325, -40.46779492, -51.30226166,
                -53.06919078, -51.92335275, -55.45524803, -51.74580787,
                -47.90666584, -32.04353368, -34.38883569, -60.52133558,
                -53.97824806, -54.29078446, -57.76069737, -45.05271693,
                -51.18924734, -31.44082065, -25.06050503, -46.11206997,
                -46.65094927, -46.02980948, -39.83036137, -45.81979561,
                -48.06673296, -36.7109775 ,   0.        , -44.16514498,
                  0.        , -37.34386773, -37.97042905, -41.77592628,
                  0.        ,   0.        ,   0.        ,   0.        ,
                  0.        ])
```

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

0.0.6 For First Visit

```
In [25]: for ep in experience:
             seen=[]
             for s,G in ep:
                 if s not in seen:
                     V[s]=V[s]+(1/(N_total[s]+1))*(G-V[s])
                     N_total[s]+=1
                     seen.append(s)
```

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

```
Out[26]: array([-86.73102671, -84.07347681, -81.12569448, -72.82297526,
              -60.20617668, -55.80269043, -45.57817384, -47.60998854,
              -78.09164686, -74.25389891, -78.2785123 , -64.3378077 ,
              -52.28777411, -57.179348  , -58.13115469, -68.2678316 ,
              -59.26795867, -51.66067771, -51.47181934, -42.18896241,
              -55.25620934, -56.53481195, -65.1295067 , -64.7772795 ,
              -64.32520196, -46.98391528, -38.23403554, -75.7583354 ,
              -53.97824806, -59.54994773, -57.76069737, -49.3975022 ,
              -55.73402882, -47.79924859, -23.69771839, -46.11206997,
              -46.65094927, -43.9367204 , -44.71177243, -59.92243798,
              -55.2868978 , -36.7109775 ,   0.        , -45.56774744,
                0.        , -37.34386773, -37.97042905, -45.01792671,
                0.        ,   0.        ,   0.        ,   0.        ,
                0.        ])
```

0.0.7 TD Learning

```
In [27]: 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 [28]: generator=get_states(policy)
```

```
In [29]: generator
```

```
Out[29]: <generator object get_states at 0x7f1c73970ba0>
```

```
In [30]: alpha=0.001
         gamma=0.99
         v=np.zeros(env.state_count)
```

```
In [31]: for step in range(100000):
             s,r,s_prime=next(generator)
             v[s]=v[s]+alpha*((r+gamma*v[s_prime])-v[s])
```

```
In [32]: v
```

```
Out[32]: array([-1.33785740e+01, -1.22713570e+01, -1.36420089e+01, -1.01302412e+01,
                -8.81320313e+00, -2.92625865e+00, -6.76350099e-01, -6.97991255e-01,
                -7.73348913e+00, -6.06481679e+00, -1.09631577e+01, -4.89132828e+00,
                -2.18400550e-01, -1.07452047e-01, -1.29735602e-02, -5.01962838e-03,
                -3.41432513e-02,  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                -9.04970198e-03, -1.95004960e-02, -1.09828725e-02, -9.09697556e-03,
                -4.17501783e-02,  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                 0.00000000e+00, -2.01506385e-03, -1.99999294e-03, -2.02173719e-03,
                -8.09245801e-03,  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                 0.00000000e+00,  0.00000000e+00,  0.00000000e+00, -3.00123880e-02,
                -7.08813469e-02,  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
                 0.00000000e+00])
```

```
In [ ]:
```

```
In [ ]:
```