Setup 1: Binary Rewards, 6 Food Items

May 17, 2020

[12]:
```
import numpy
import matplotlib
import matplotlib.pyplot as plt
```

[2]:
```
#Experiments to run: Exp 1 -- random initialization of agent and goal positions␣
 ↪for different environments and compute fitness
#In static environment -- having more rewards may not necessarily be better
#Exp 2 -- same experiment but now the different goals will disappear after␣
 ↪certain time-steps
%matplotlib inline

import matplotlib
import time
import random
import numpy as np
import pandas as pd
import seaborn as sns

import os
import sys
module_path = os.path.abspath(os.path.join('..'))
if module_path not in sys.path:
    sys.path.append(module_path)

from pyrlap.domains.gridworld import GridWorld
from pyrlap.algorithms.qlearning import Qlearning
from pyrlap.domains.gridworld.gridworldvis import visualize_trajectory
import warnings
warnings.filterwarnings('ignore')
```

[3]:
```
# define the gridworld here

def create_array():
    y_loc = (random.randint(1, 3), 2) #rewards only in 2nd half of grid world,␣
 ↪in different columns
    x_loc = (random.randint(1, 3), 5)
    z_loc = (random.randint(1, 3), 8)
```

```
    q_loc = (random.randint(6, 8), 2) #rewards only in 2nd half of grid world,␣
↪in different columns
    w_loc = (random.randint(6, 8), 5)
    e_loc = (random.randint(6, 8), 8)
    locs = [y_loc, x_loc, z_loc, q_loc, w_loc, e_loc]

    gw_array = ['.........', '.........', '.........', '.........', '.......
↪..',
                '.........', '.........', '.........', '.........', '.......
↪..',
                ]
    #n*n grid

    s = gw_array[x_loc[0]]
    new = list(s)
    new[x_loc[1]] = 'x'
    s = ''.join(new)
    gw_array[x_loc[0]] = s

    s = gw_array[y_loc[0]]
    new = list(s)
    new[y_loc[1]] = 'y'
    s = ''.join(new)
    gw_array[y_loc[0]] = s

    s = gw_array[z_loc[0]]
    new = list(s)
    new[z_loc[1]] = 'z'
    s = ''.join(new)
    gw_array[z_loc[0]] = s


    s = gw_array[q_loc[0]]
    new = list(s)
    new[q_loc[1]] = 'q'
    s = ''.join(new)
    gw_array[q_loc[0]] = s

    s = gw_array[w_loc[0]]
    new = list(s)
    new[w_loc[1]] = 'w'
    s = ''.join(new)
    gw_array[w_loc[0]] = s

    s = gw_array[e_loc[0]]
    new = list(s)
    new[e_loc[1]] = 'e'
```

```
    s = ''.join(new)
    gw_array[e_loc[0]] = s

    return gw_array, (locs)

def create_gridworld(rewards={'.':0, 'x':0, 'y':0, 'z':0, 'q':0, 'w':0, 'e':0}):
    gw_array, locs = create_array() #get the array and the location of goals to␣
 ↪compute fitness
    in_state = (random.randint(4, 6), random.randint(4, 6)) #initial location at␣
 ↪bottom left corner
    gw = GridWorld(gridworld_array=gw_array,
    init_state=in_state, #random initialization of the initial state
    feature_rewards=rewards) #the reward function
    return gw, locs

def learn(gw,m):
    np.random.seed(1234)
    all_run_data = []
    for i in range(1):
        params = {'learning_rate': 1,
              'eligibility_trace_decay': .8,
              'initial_qvalue': 10}
        qlearn = Qlearning(gw,
                       softmax_temp=.2,
                       discount_rate=.99,
                       **params)
        run_data = qlearn.train(episodes=1,
                            max_steps=m,
                            run_id=i,
                            return_run_data=True)
        for r in run_data:
            r.update(params)
        all_run_data.extend(run_data)

    return qlearn, (all_run_data)

def plot_learnt(qlearn, gw): #function to plot the learnt trajectory
    traj = qlearn.run(softmax_temp=0.0, randchoose=0.0, max_steps=1000)
    gwp = gw.plot()
    gwp.plot_trajectory(traj=[(s, a) for s, a, ns, r in traj])

def plot_history(gw, all_run_data):
    run_df = pd.DataFrame(all_run_data) #convert to pandas
    state = run_df["s"] #get states visited
    state_array = state.values #convert to matrix

    action = run_df["a"] #get actions taken
```

```
    action_array = action.values #convert to matrix

    gwp = gw.plot() #plot it
    gwp.plot_trajectory(traj=[(s, a) for s in state_array for a in action_array])

def compute_fitness(all_run_data, locs, gw): #function computes fitness over the␣
 ↪history (you can also compute fitness over the learnt policy)
    fitness = 0

    run_df = pd.DataFrame(all_run_data) #convert to pandas
    state = run_df["s"] #get states visited
    state_array = state.values #convert to matrix

    for i in range(len(state_array)):
        a = state_array[i]
        if((9-a[1],a[0]) in locs): #loop through the traj, if traj position is␣
 ↪not same as food position then fitness decreases, else increases by one
            fitness = fitness+1
        else:
            fitness = fitness-0.1
    return fitness
```

[8]:
```
## REWARDS = Binary, all combinations, THOUSAND steps, CENTER initial state
# Initial state = center
# Episodes = 50
# Trials = 10
# Steps = 1000
# Environments = 10

fitness_vals_thousand = np.zeros(64)
x_vals_thousand = np.zeros(64)
y_vals_thousand = np.zeros(64)
z_vals_thousand = np.zeros(64)
q_vals_thousand = np.zeros(64)
w_vals_thousand = np.zeros(64)
e_vals_thousand = np.zeros(64)

count = 0


for x in range(2):
    for y in range(2):
        for z in range(2):
                for q in range(2):
                    for w in range(2):
                        for e in range(2):
```

```
                                rewards={'.':0, 'x':x*10, 'y':y*10, 'z':z*10,␣
↪'q':q*10, 'w':w*10, 'e':e*10}
                                fit = []
                                for env in range(100): #Repeatedly sample lots␣
↪of environments
                                    gw, locs = create_gridworld(rewards) #create␣
↪gridworld here
                                    qlearn, all_run_data = learn(gw, 10000)   ␣
↪#Q-learn here
                                    fitness= compute_fitness(all_run_data, locs,␣
↪gw) #compute fitness here for the enviroment
                                    fit.append(fitness)
                                    if (env == 1):
                                        print("Rewards are set as:")
                                        print('x =', x, 'y =', y, 'z =', z, 'q␣
↪=', q, 'w =', w, 'e =', e)
                                        plot_learnt(qlearn, gw)


                                fitness_vals_thousand[count] = sum(fit)/
↪float(len(fit))
                                print("Average Fitness",␣
↪fitness_vals_thousand[count])
                                x_vals_thousand[count] = x
                                y_vals_thousand[count] = y
                                z_vals_thousand[count] = z
                                q_vals_thousand[count] = q
                                w_vals_thousand[count] = w
                                e_vals_thousand[count] = e
                                count += 1
```

```
Rewards are set as:
x = 0 y = 0 z = 0 q = 0 w = 0 e = 0
Average Fitness -239.66899999996826
Rewards are set as:
x = 0 y = 0 z = 0 q = 0 w = 0 e = 1
Average Fitness 3288.1960000003123
Rewards are set as:
x = 0 y = 0 z = 0 q = 0 w = 1 e = 0
Average Fitness 3352.32600000033
Rewards are set as:
x = 0 y = 0 z = 0 q = 0 w = 1 e = 1
Average Fitness 3398.372000000334
Rewards are set as:
x = 0 y = 0 z = 0 q = 1 w = 0 e = 0
Average Fitness 3413.1670000003323
```

```
Rewards are set as:
x = 0 y = 0 z = 0 q = 1 w = 0 e = 1
Average Fitness 3409.4490000003343
Rewards are set as:
x = 0 y = 0 z = 0 q = 1 w = 1 e = 0
Average Fitness 3400.055000000336
Rewards are set as:
x = 0 y = 0 z = 0 q = 1 w = 1 e = 1
Average Fitness 3428.4350000003365
Rewards are set as:
x = 0 y = 0 z = 1 q = 0 w = 0 e = 0
Average Fitness 3345.913000000319
Rewards are set as:
x = 0 y = 0 z = 1 q = 0 w = 0 e = 1
Average Fitness 3400.924000000327
Rewards are set as:
x = 0 y = 0 z = 1 q = 0 w = 1 e = 0
Average Fitness 3398.3280000003324
Rewards are set as:
x = 0 y = 0 z = 1 q = 0 w = 1 e = 1
Average Fitness 3398.911000000334
Rewards are set as:
x = 0 y = 0 z = 1 q = 1 w = 0 e = 0
Average Fitness 3420.9110000003316
Rewards are set as:
x = 0 y = 0 z = 1 q = 1 w = 0 e = 1
Average Fitness 3413.3540000003313
Rewards are set as:
x = 0 y = 0 z = 1 q = 1 w = 1 e = 0
Average Fitness 3436.784000000338
Rewards are set as:
x = 0 y = 0 z = 1 q = 1 w = 1 e = 1
Average Fitness 3389.990000000334
Rewards are set as:
x = 0 y = 1 z = 0 q = 0 w = 0 e = 0
Average Fitness 3380.4420000003283
Rewards are set as:
x = 0 y = 1 z = 0 q = 0 w = 0 e = 1
Average Fitness 3406.4020000003325
Rewards are set as:
x = 0 y = 1 z = 0 q = 0 w = 1 e = 0
Average Fitness 3369.8270000003313
Rewards are set as:
x = 0 y = 1 z = 0 q = 0 w = 1 e = 1
Average Fitness 3396.8980000003316
Rewards are set as:
x = 0 y = 1 z = 0 q = 1 w = 0 e = 0
Average Fitness 3422.5500000003335
```

```
Rewards are set as:
x = 0 y = 1 z = 0 q = 1 w = 0 e = 1
Average Fitness 3425.278000000335
Rewards are set as:
x = 0 y = 1 z = 0 q = 1 w = 1 e = 0
Average Fitness 3411.4290000003357
Rewards are set as:
x = 0 y = 1 z = 0 q = 1 w = 1 e = 1
Average Fitness 3427.071000000335
Rewards are set as:
x = 0 y = 1 z = 1 q = 0 w = 0 e = 0
Average Fitness 3380.1670000003282
Rewards are set as:
x = 0 y = 1 z = 1 q = 0 w = 0 e = 1
Average Fitness 3411.9020000003325
Rewards are set as:
x = 0 y = 1 z = 1 q = 0 w = 1 e = 0
Average Fitness 3389.1430000003306
Rewards are set as:
x = 0 y = 1 z = 1 q = 0 w = 1 e = 1
Average Fitness 3383.665000000331
Rewards are set as:
x = 0 y = 1 z = 1 q = 1 w = 0 e = 0
Average Fitness 3429.898000000334
Rewards are set as:
x = 0 y = 1 z = 1 q = 1 w = 0 e = 1
Average Fitness 3425.8390000003355
Rewards are set as:
x = 0 y = 1 z = 1 q = 1 w = 1 e = 0
Average Fitness 3413.882000000334
Rewards are set as:
x = 0 y = 1 z = 1 q = 1 w = 1 e = 1
Average Fitness 3408.7340000003333
Rewards are set as:
x = 1 y = 0 z = 0 q = 0 w = 0 e = 0
Average Fitness 3346.859000000325
Rewards are set as:
x = 1 y = 0 z = 0 q = 0 w = 0 e = 1
Average Fitness 3394.31300000033
Rewards are set as:
x = 1 y = 0 z = 0 q = 0 w = 1 e = 0
Average Fitness 3384.600000000334
Rewards are set as:
x = 1 y = 0 z = 0 q = 0 w = 1 e = 1
Average Fitness 3387.9000000003325
Rewards are set as:
x = 1 y = 0 z = 0 q = 1 w = 0 e = 0
Average Fitness 3393.422000000333
```

```
Rewards are set as:
x = 1 y = 0 z = 0 q = 1 w = 0 e = 1
Average Fitness 3419.0520000003335
Rewards are set as:
x = 1 y = 0 z = 0 q = 1 w = 1 e = 0
Average Fitness 3409.801000000337
Rewards are set as:
x = 1 y = 0 z = 0 q = 1 w = 1 e = 1
Average Fitness 3406.468000000337
Rewards are set as:
x = 1 y = 0 z = 1 q = 0 w = 0 e = 0
Average Fitness 3381.740000000328
Rewards are set as:
x = 1 y = 0 z = 1 q = 0 w = 0 e = 1
Average Fitness 3374.5240000003264
Rewards are set as:
x = 1 y = 0 z = 1 q = 0 w = 1 e = 0
Average Fitness 3412.1330000003345
Rewards are set as:
x = 1 y = 0 z = 1 q = 0 w = 1 e = 1
Average Fitness 3412.2650000003355
Rewards are set as:
x = 1 y = 0 z = 1 q = 1 w = 0 e = 0
Average Fitness 3407.865000000333
Rewards are set as:
x = 1 y = 0 z = 1 q = 1 w = 0 e = 1
Average Fitness 3414.773000000332
Rewards are set as:
x = 1 y = 0 z = 1 q = 1 w = 1 e = 0
Average Fitness 3417.6110000003364
Rewards are set as:
x = 1 y = 0 z = 1 q = 1 w = 1 e = 1
Average Fitness 3403.3550000003365
Rewards are set as:
x = 1 y = 1 z = 0 q = 0 w = 0 e = 0
Average Fitness 3388.6590000003284
Rewards are set as:
x = 1 y = 1 z = 0 q = 0 w = 0 e = 1
Average Fitness 3406.43500000033
Rewards are set as:
x = 1 y = 1 z = 0 q = 0 w = 1 e = 0
Average Fitness 3390.5620000003332
Rewards are set as:
x = 1 y = 1 z = 0 q = 0 w = 1 e = 1
Average Fitness 3384.4790000003322
Rewards are set as:
x = 1 y = 1 z = 0 q = 1 w = 0 e = 0
Average Fitness 3404.3890000003325
```

```
Rewards are set as:
x = 1 y = 1 z = 0 q = 1 w = 0 e = 1
Average Fitness 3422.605000000336
Rewards are set as:
x = 1 y = 1 z = 0 q = 1 w = 1 e = 0
Average Fitness 3413.717000000337
Rewards are set as:
x = 1 y = 1 z = 0 q = 1 w = 1 e = 1
Average Fitness 3387.0090000003333
Rewards are set as:
x = 1 y = 1 z = 1 q = 0 w = 0 e = 0
Average Fitness 3406.99600000033
Rewards are set as:
x = 1 y = 1 z = 1 q = 0 w = 0 e = 1
Average Fitness 3401.507000000334
Rewards are set as:
x = 1 y = 1 z = 1 q = 0 w = 1 e = 0
Average Fitness 3393.4660000003364
Rewards are set as:
x = 1 y = 1 z = 1 q = 0 w = 1 e = 1
Average Fitness 3391.7170000003307
Rewards are set as:
x = 1 y = 1 z = 1 q = 1 w = 0 e = 0
Average Fitness 3422.6270000003324
Rewards are set as:
x = 1 y = 1 z = 1 q = 1 w = 0 e = 1
Average Fitness 3419.7450000003364
Rewards are set as:
x = 1 y = 1 z = 1 q = 1 w = 1 e = 0
Average Fitness 3413.211000000337
Rewards are set as:
x = 1 y = 1 z = 1 q = 1 w = 1 e = 1
Average Fitness 3405.9400000003334
```