Implementing MAXQ and Qlearning methods in R and applying them to the taxi problem [*]

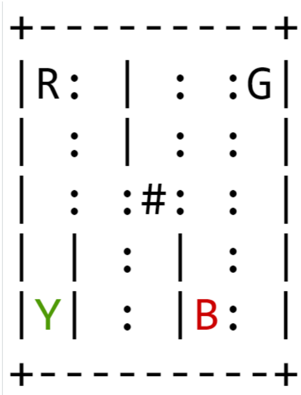
Mihai Groza
Concordia University
fill this

Khaled Fouda
Concordia University
khaledsfouda@gmail.com

December 2, 2020

Abstract

last thing to do.



```
+---------+
|R: | : :G|
| : | : : |
| : :#: : |
| | : | : |
|Y| : |B: |
+---------+
```

Figure 1: *generated by render(s)*

I. Introduction

later.

II. Environment

The taxi's - shown as # - goal is to first pick up the passenger from one of the four places (R,Y,G,B) where the current location of the passenger is shown in green. After picking up the customer, the next and final goal is to drive him to his destination which is one of the four places (R,Y,G,B) and the destination is shown in red.

The state is defined as a set (taxi_row, taxi_column, passenger_location, destination_location) where:
taxi_row and taxi_column take values from 1 to 5 as we have 5x5 grid,
passenger_location and destination_location take values [1,2,3,4] for [R,Y,G,B], furthermore, if the passenger is in the taxi then they takes a location 5.
That being said, we have $5 * 5 * 5 * 4 = 500$ different states.

The set of actions available is (North, South, East, Wast) with ids 1 to 6.

If the taxi successfully dropped off the passenger then the episode is over and they receive a reward of +20, if they dropped the passenger at a wrong location or before picking them up first then they receive a reward of -10 and continue the episode to drop the passenger at the right location (ie. no change in the state). Similarly if they attempted to pick the passenger at the wrong place. Otherwise, they receive a reward of -1 for each step taken. Note that hitting the wall

[*] Reinforcement learning undergraduate class project

results in a reward of -1 and no change in the state.

I have implemented the following functions for the environment:

- render(state) : returns nothing and prints out the environment at the current state.
- encode(s) : maps each state to an integer between 1 and 500.
- decode(i): The inverse of encode(s). it takes an integer between 1 and 500 and returns the corresponding state.
- loc.indx(i): maps the four pick/drop locations (1,2,3,4) to a (row,column) set.
- hitting.wallQ(r,c,a): returns True if taking the action (a) at the location (r,c) would results in hitting the wall.
- step(s,a) returns (s′,r) at state s, it takes the action a, observes the reward r and the next state s′

III. MAXQ method

IV. Qlearning method

V. Results

VI. Similar projects

VII. Final thoughts

References

[Figueredo and Wolf, 2009] Figueredo, A. J. and Wolf, P. S. A. (2009). Assortative pairing and life history strategy - a cross-cultural study. *Human Nature*, 20:317–330.