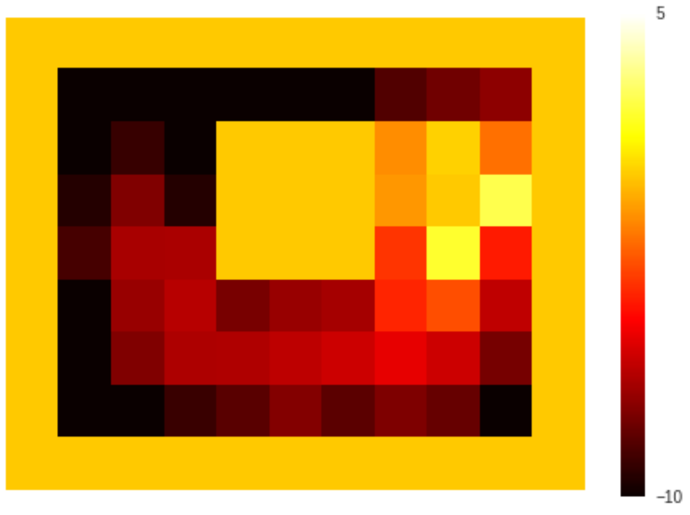
Reinforcement Learning Coursework 2

WU Zhi   17040772   UCABWUA@UCL.AC.UK
Department of Computer Science
University College London

Part 1: Implement agents

Agent 1 TD learning

Plot



Question

[5 pts] Would the greedy after one such iteration of policy evaluation and policy improvement be optimal on this problem? Explain (in one or two sentences) why or why not.

NO， in one iteration, we can only update current state value via the next state, but there are 4 possible state for updating current state. The one iteration one-step TD learning can not converge to optimal.

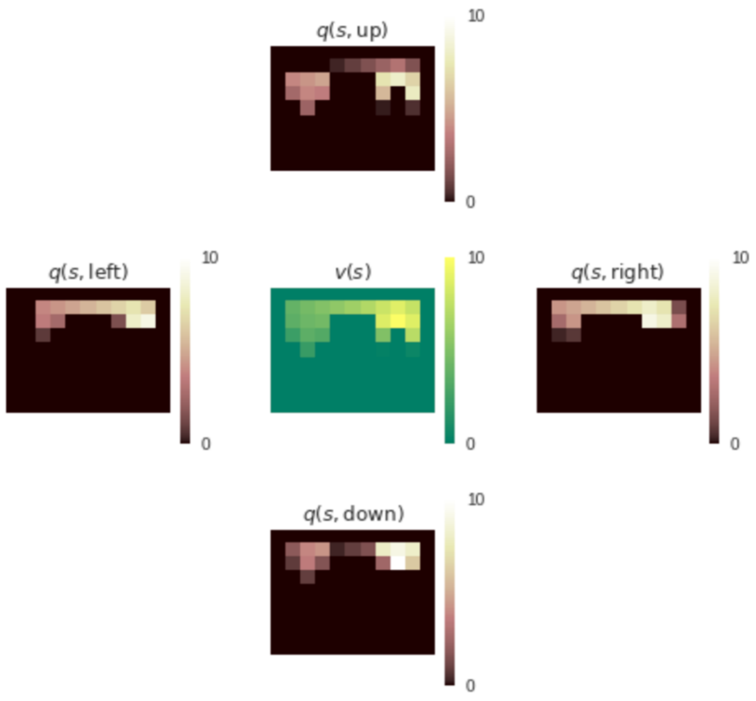
[5 pts] If we repeat the process over and over again, and repeatedly evaluate the greedy policy and then perform another improvement step, would then the policy eventually become optimal? Explain (in one or two sentences) why of why not.

Yes. It uses batch updating, under which one-step TD converges deterministically to a single answer independent of the step-size parameter, α as long as α is chosen to be sufficiently small. The batch TD(0) always finds the estimate that would be exactly correct for the maximun-likeihood model of the Markov process so that we can compute the estimate of the value function that would be exactly correct if the model were exactly correct.

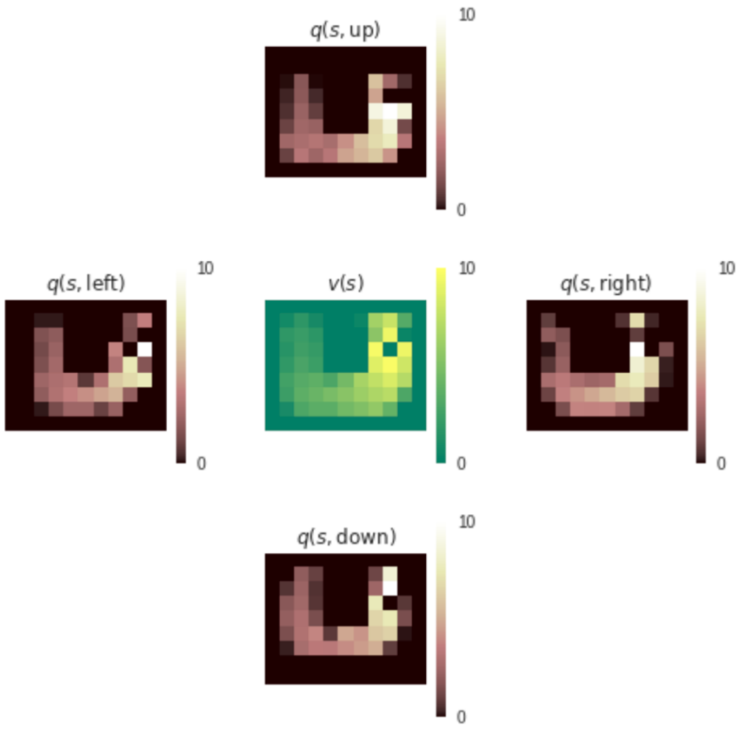
Part 2: Analyse Results

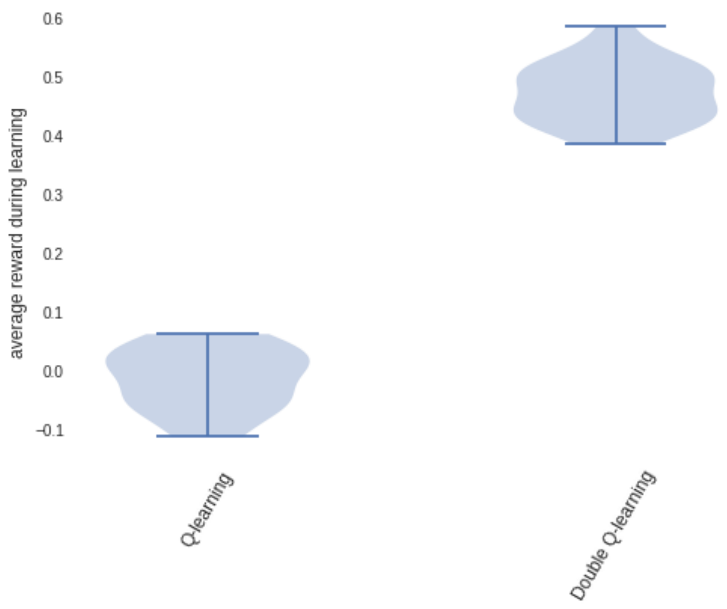
Plot

```
Q-learning
```

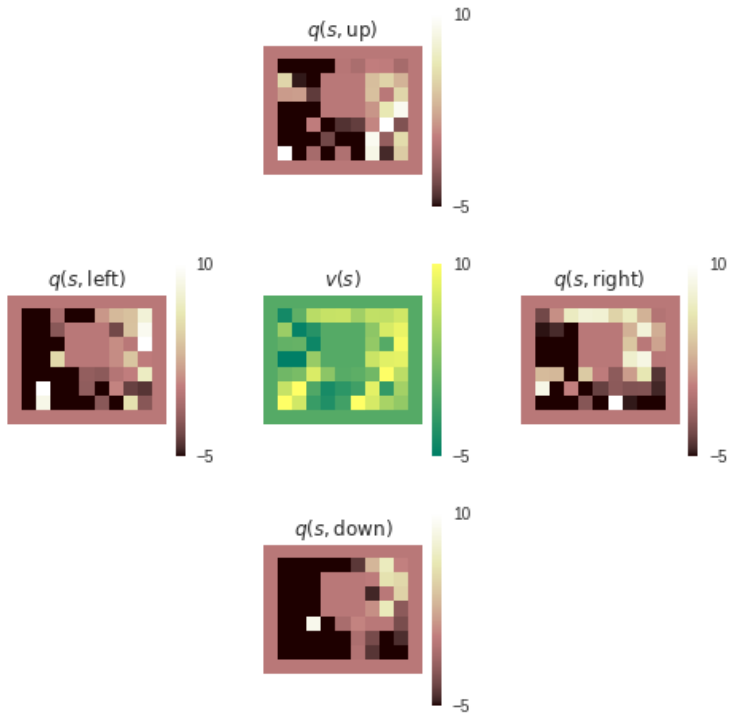


```
SARSA
```

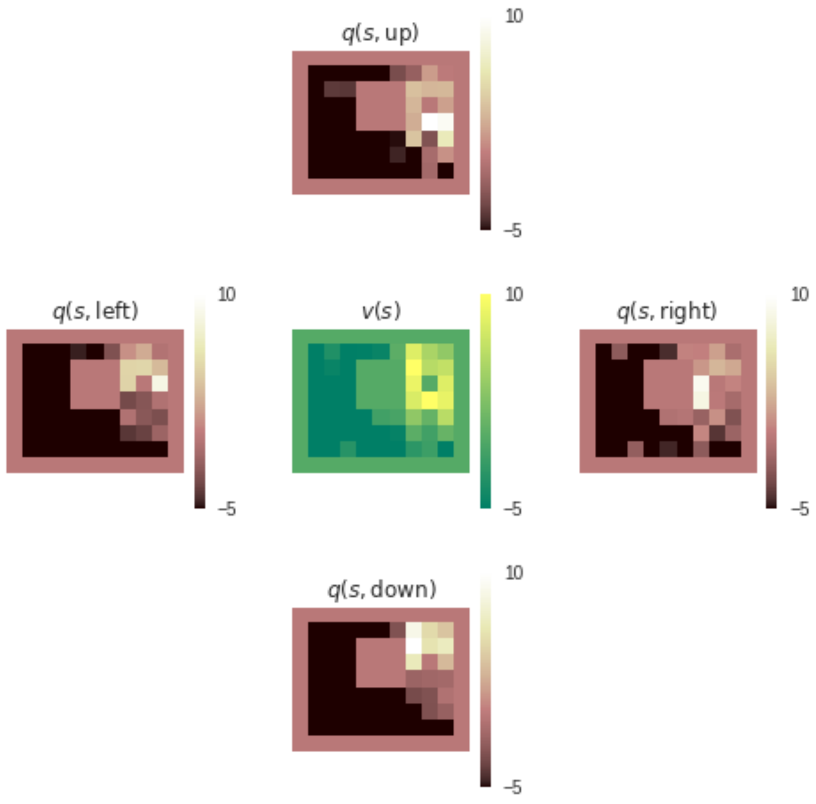




average reward during learning
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
Q-learning
Double Q-learning



$q(s, \text{up})$
$q(s, \text{left})$
$v(s)$
$q(s, \text{right})$
$q(s, \text{down})$
10
−5



$q(s,\mathrm{up})$
$q(s,\mathrm{left})$
$v(s)$
$q(s,\mathrm{right})$
$q(s,\mathrm{down})$
10
−5

Question

[10 pts] *How* do the policies found by Q-learning and Sarsa differ? (Explain qualitatively how the behaviour differs in one or two sentences.)

Q-learning is off-policy learning. Because it have $max_a Q(s_{t+1}, a)$ in action value update fucntion, and it always choose the most shortest path to goal, no matter how the rewards in the process. In the figure above, Q-learning choose shorter but moere safe path compared with Q-learning.

Sarsa is on-policy learning. Sarsa is quite conservative, it will choose to stay far away from low rewards. In the figure above, Sarsa choose farther but more dangerous path compared with Q-learning.

[10 pts] *Why* do the policies differ in this way?

The biggest difference between Q-learning and SARSA is that Q-learning is off-policy, whereas SARSA is on-policy.

The updated equations for Q-learning and SARSA are below:

$$Q - learning : Q(s_t, a_t) = Q(s_t, a_t) + a[r + \gamma max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$$

$$Sarsa : Q(s_t, a_t) = Q(s_t, a_t) + a[r + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$$

In state s', Q-learning only calculate which a' to take in s' to get bigger Q, and do not really take this action a'. The selection of action a is based on the current Q network and the strategy (e-greedy), and the calculation of the target Q value is calculated based on the action a' with the largest Q value. Therefore, it is the off-policy learning

In state s', Sarsa know which a' it want to take and really take this action. The selection of action a follows the e-greedy policy, and the calculation of the target Q value is also calculated according to the action a' obtained by the (e-greedy) policy, thus it is on-policy learning.

[10 pts] Which greedy policy is better, in terms of actual value?

If the goal is to train an optimal agent in simulation, or in a low-cost and fast-iterating environment, then Q-learning is a good choice, due to the first point (learning optimal policy directly). If your agent learns online, and you care about rewards gained whilst learning, then SARSA may be a better choice.

In the grid environment of this coursework, Q-learning can choose the shorter path, and Sarsa can gain more rewards in the learning process.

[10 pts] Explain why Double Q-learning has a higher average reward. Use at most four sentences, and discuss at least a) the dynamics of the algorithm, b) how this affects behaviour, and c) why the resulting behaviour yields higher rewards for Double Q-learning than for Q-learning.

Because Q-learning use max operator in action value update euqation, and it will cause maximiztion bias problem, and double Q-learning can solve this problem. In the double Q-learning algorithm there are two action-value functions, and we update one of the two value functions by assigning each experience randomly. The goal is to decorrelate the selection of the best action from the evaluation of this action to avoid maximiztion bias problem, so double Q-learning have higher average reward.