Reinforcement Learning Homework 2

Deadline 29th October 2024
Students Sahiti Chebolu, Surabhi S Nath, Xin Sui

Q1 State-Action Value Function and Policy Iteration

(a)

$$q_\pi(s, a) = R_s^a + v_\pi(s') = -1 + v_\pi(s') \tag{1}$$

generally holds true for this environment because :

1. rewards are deterministically -1.
2. task is undiscounted.
3. state transitions are deterministic, i.e. there exists only one possible next state s' regardless of the action taken.

Therefore:

$$q_\pi(11, \text{down}) = -1 + v_\pi(\text{terminal state}) = -1 \tag{2}$$

$$q_\pi(7, \text{down}) = -1 + v_\pi(11) = -1 + (-14) = -15 \tag{3}$$

$$q_\pi(9, \text{left}) = -1 + v_\pi(8) = -1 + (-20) = -21 \tag{4}$$

(b)

$$v_*(s) = \max_a q_*(s, a) \tag{5}$$

(c)

$$q_*(s, a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_*(s') \tag{6}$$

(d)

$$\pi_*(a \mid s) = [\![a = \arg\max_a q_*(s, a)]\!] \tag{7}$$

(e)

$$q_\pi(s, a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \sum_{a' \in A} \pi(a' \mid s') q_\pi(s', a') \tag{8}$$

Q2 Value Iteration

2.2 a

It requires 5 iterations for start state (4, 0) to have non-zero value. This is because, moving to through the grid has reward = 0. Only reaching the terminal state has positive reward. Hence, it takes some iterations for this positive value to propagate backwards to the start state (4, 0). However, it takes more iterations for (2, 0) and (3, 0) to get non-zero value since they are more distant in the grid form the terminal state.

2.2 b

At noise = 0, for larger discount rates, the agent dares to cross the bridge because it doesn't discount the delayed reward = 10 at the end of the bridge. However at smaller discounts, the immediate negative reward of falling off the bridge looms large and the optimal policy is not to cross. The threshold discount rate at which it changes from the this short-sighted to long-sighted behavior is $\gamma = 0.565$.

As noise increases, the agent rapidly gets more cautious. Even for $\gamma = 1$, the agent doesn't cross the bridge beyond noise = 0.03.

2.2 c

(a)

Prefers close exit, risking the cliff

At noise = 0, a discount factor of upto and including ≈ 0.316 produces this behaviour.

If the noise is increased upto 0.06, this behaviour is achieved upto discount factor ≈ 0.348.

A noise larger than 0.06 and a discount factor larger than 0.348 cannot attain this behaviour. Also, in the noise $>$ 0 case, the agent may hit the cliff.

(b)

Prefers close exit, avoiding the cliff

When noise is greater than 0.06 and the discount factor is less than ≈ 0.343, this behaviour is achieved.

However, as noise becomes larger than 0.2, behaviour is hard to judge.

(c)

Prefers far exit, risking the cliff

At noise = 0, a discount factor of $>\approx 0.316$ produces this behaviour.

If the noise is increased upto 0.06, this behaviour is achieved at discount factor $>\approx 0.348$. A noise greater than 0.06 and a discount factor less than 0.316 cannot attain this behaviour. Also, in the noise $>$ 0 case, the agent may hit the cliff.

If the noise is increased further upto 0.12, this behaviour is achieved at discount factor $\geq\approx 0.5$ and $<\approx 0.9$.

A noise greater than 0.12 and a discount factor less than 0.316 cannot attain this behaviour. Also, in the noise $>$ 0 case, the agent may hit the cliff.

(d)

Prefers far exit, avoiding the cliff

When the noise is a bit higher than 0.12, and the discount factor is higher than 0.36, this behaviour is achieved. If noise is increased further, the discount factor also needs to be increased to achieve this behaviour.

However, as noise becomes larger than 0.2, behaviour is hard to judge.

(e)

Avoids both exits and the cliff

When noise is 0 and discount factor is set to either 0 or 1, the value for all non-terminal states are the same (0 for discount 0, or 10 for discount 1) and the optimal policy for all states is "up". As a result, the agent starts in state (3, 0) and goes all the way to (0, 0) and gets stcuck there. Therefore, here it avoids both exits and the cliff.

2.2 d

The optimal value of start state (4, 0) after 100 value iteration rounds is 0.2822. The average returns after 10 episodes is variable, we got 0.293. The difference is due to noise in the actions, which can change the trajectories in each episode and hence the average returns.

After 10000 episodes, we got 0.2817, which is very close to the optimal value. It also doesn't vary much on re-runs.

2.3 Create your own gridworld

We created a new grid called "HorizonGrid", which is set up as follows:

1. starting state is the middle cell of the leftmost column (2,0).
2. there's a medium-sized punishment (-8) in the agent's path to a larger reward (10).
3. there's a medium-sized reward (8) in the agent's path to a larger punishment (-10).

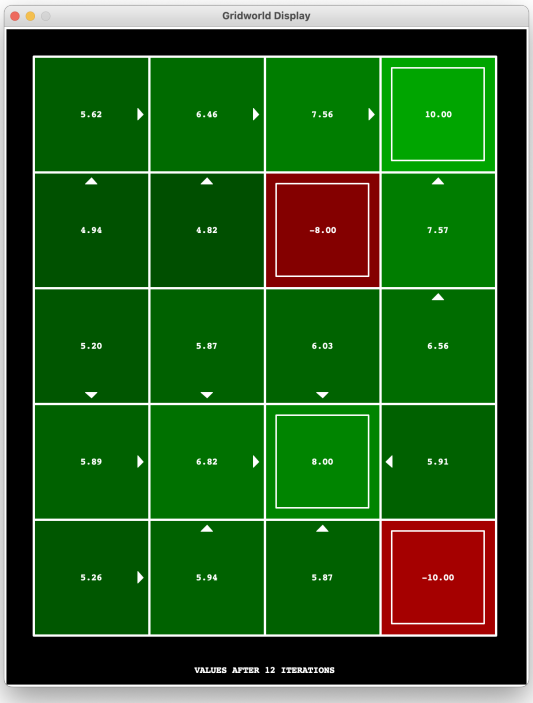
Such a grid enables interesting demostrations (see Figure 1 below) of how an agent's "horizon" is affected by different noise level (e.g. default 0.2 vs. 0), discounting factor (e.g. default 0.9 vs. 0.99), and living reward (e.g. default 0 vs. 1 per step).

In the default setting, the value iteration agent's optimal policy from the start state is to move towards the medium reward.

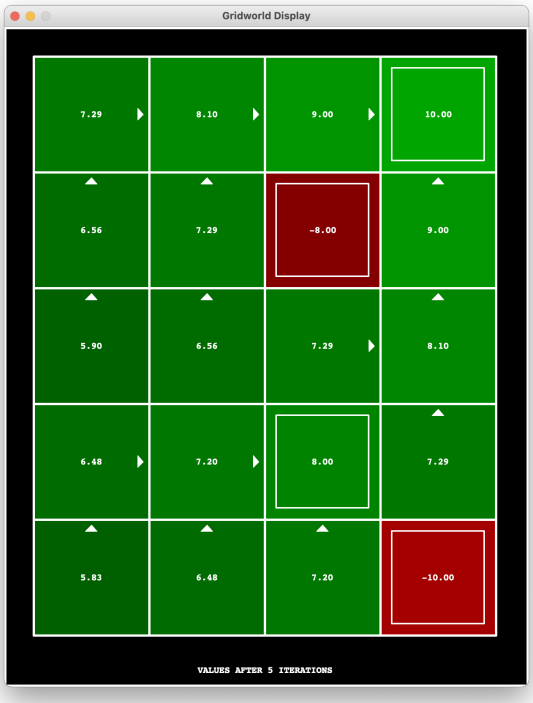
Noise level Contrasting Figure 1b with Figure 1a: after eliminating noise, the value iteration agent's optimal policy from the start state is changed to moving towards the large reward instead. Its value function at the starting state also increases slightly. Both changes make sense because now there's no long the possibility of falling into the medium punishment trap due to noise.

Discounting factor Contrasting Figure 1c with Figure 1a: after increasing agent's discounting factor to very close to 1 (i.e. almost undisctount), the value iteration agent's optimal policy from the start state is also changed to moving towards the further but larger reward. Its value function at the starting state also increases. Both changes make sense because now there's very little discounting of future rewards, which raises the agent's effective horizon.

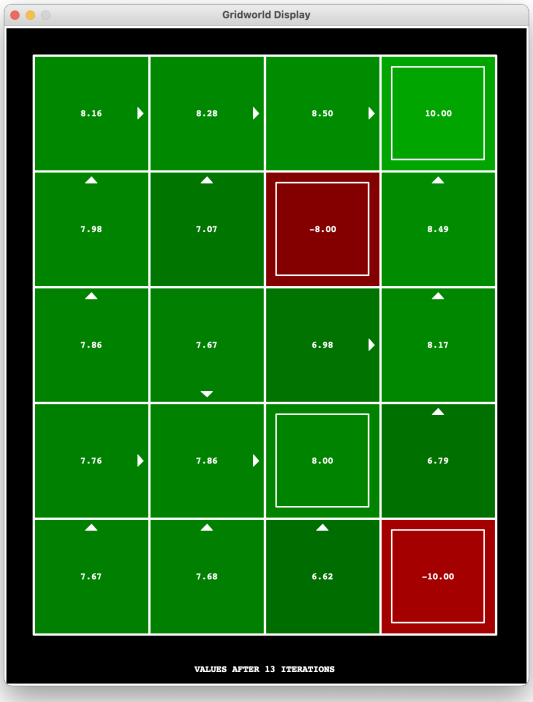
Living reward Contrasting Figure 1d with Figure 1a: after giving the agent a living reward of 1 per time step the value iteration agent's optimal policies are to move away from any rewarding (and punishing) states except for when they're right next to the large reward. Its value function at the starting state also increases a lot. Both changes make sense because now the agent is incentivised to stay away from terminal states in order to collect more living rewards, except for when they're right next to the large reward, in which case it's more worthwhile to collect a certern, large reward than risking falling into the -8 punishing state due to the 0.2 noise level.



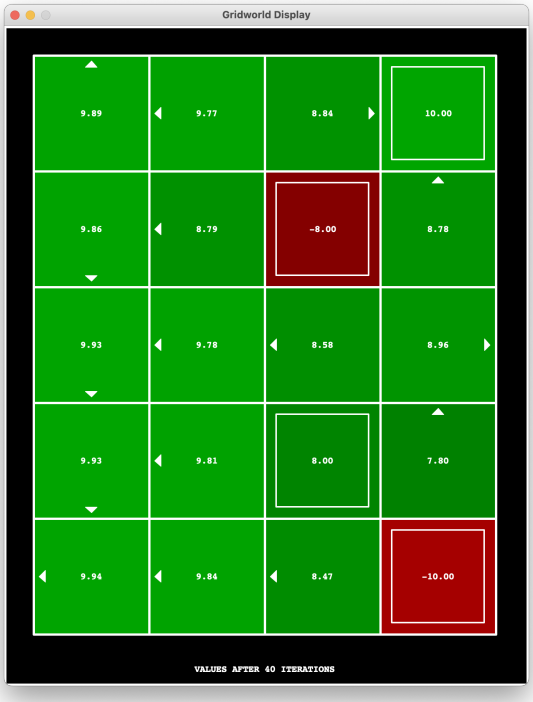
(a) Default (noise=0.2, discount=0.9, livingReward=0)



(b) noise = 0 (everything else default)



(c) discount = 0.99 (everything else default)



(d) livingReward = 1 (everything else default)

Figure 1: **Value functions and optimal policies in the HorizonGrid** (contrasting **(a)** with **(b)(c)(d)**). Starting state is the middle cell of the leftmost column, (2,0). Number of iterations in each condition is enough for the convergence of the value functions (threshold = 0.001).