Introduction to Artificial Intelligence (236501) | Assignment #3

July 6, 2023

Part A – MDP

(1)

(a)

$$\pi : S \to A$$
$$U^{\pi}(s) = E_{\pi}\left[\sum_{t=0}^{\infty} \gamma^{t} R\left(S_t, \pi\left(S_t\right)\right) | S_0 = s\right]$$

(b)

$$U(s) = \max_{a \in A}\left[R(s,a) + \gamma \sum_{s' \in S} P\left(s'|s,a\right) U\left(s'\right)\right] \equiv \max_{a \in A}\left[\sum_{s' \in S} P\left(s'|s,a\right) R(s,a) + \gamma \sum_{s' \in S} P\left(s'|s,a\right) U\left(s'\right)\right]$$
$$\equiv \max_{a \in A} \sum_{s' \in S} P\left(s'|s,a\right)\left[R(s,a) + \gamma U\left(s'\right)\right]$$

(c)

Algorithm 1 Action-Reward Value Iteration

Local Variables: U, U', δ
Init: $U' \leftarrow 0$
repeat:

$U \leftarrow U', \delta \leftarrow 0$
for each $state\ s$ **in** S **do:**

$U'(s) \leftarrow \max_{a \in A(s)} \sum_{s' \in S} P\left(s'|s,a\right)\left[R(s,a) + \gamma U\left(s'\right)\right]$
$\delta = \max\left\{\delta, |U'(s) - U(s)|\right\}$

until $\delta < \frac{\epsilon(1-\gamma)}{\gamma}$
return U

(d)

Algorithm 2 Action-Reward Policy Iteration

Local Variables: U, π
Init: $\pi \leftarrow Random$
repeat:

$U \leftarrow Policy - Evaluation\left(\pi, U, mdp\right)$
$unchanged? \leftarrow true$
for each $state\ s$ **in** S **do:**

if $\max_{a \in A(s)} \sum_{s' \in S} P\left(s'|s,a\right)\left[R\left(s,a\right)+\gamma U\left(s'\right)\right] > \sum_{s' \in S} P\left(s'|s,\pi\left[s\right]\right)\left[R\left(s,\pi\left[s\right]\right)+\gamma U\left(s'\right)\right]$ **then do:**

$\pi\left[s\right] = \arg\max_{a \in A(s)} \sum_{s' \in S} P\left(s'|s,a\right)\left[R\left(s,a\right)+\gamma U\left(s'\right)\right]$
$unchanged? \leftarrow false$

until $unchanged?$
return π

By using $\gamma = 1$, there are no diminishing returns on the values we get.
In the Value-Iteration algorithm, the stopping condition will be 0, meaning that we stop only if the all state utilities have converged to the optimal utility.
In the Policy-Iteration algorithm we only care about changes in the policy, therefore the case where $\gamma = 1$ is similar to the rest.
There are conditions that have to be met in order for the algorithms to find the optimal policy:

- $|S| < \infty$. If not, the values might diverge to ∞ and the algorithms won't make sense.
- $|R| < \infty$ (reward function is bounded).
- no positive cycles. If there are, the utilites will diverge to ∞.

(2)

(a)

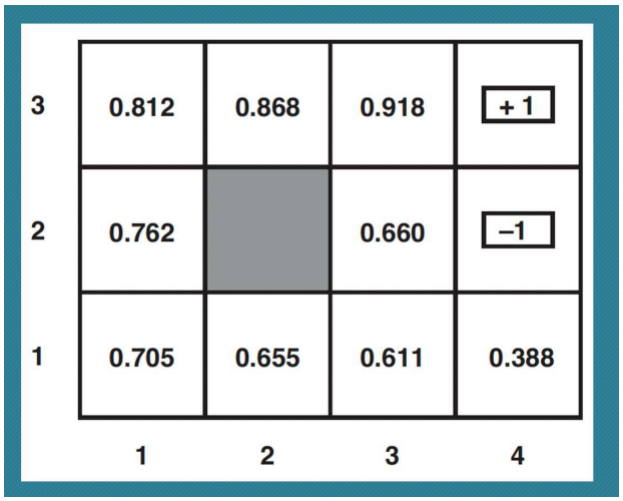
Incorrect:
Take $\gamma = 0.9$, $r_9 = 0.9$, $r_6 = 1000$.
If we try to go up, with probability:

- 0.1 we reach the terminal state, granting us $0.9 + 0.9 \cdot 1$
- 0.1 we reach $(3,2)$, granting us at least $0.9 + 0.9 \cdot 1000 > 900$
- 0.8 we end up in the same place

Therefore, the optimal utility of $(3,3)$ is at least $0.1 \cdot 900 = 90 > 1$, because we found an action that has an expectency of more than 900, while $r_9 < 1$.

(b)

Incorrect.
We saw in the tutorial that the utilities for the states, in the case where the reward for all non-terminal states is 0, is:



(c)

Incorrect. Explanation:
v_8 can be smaller than v_1 because it's directly below the -1. If $r_1 = r_2 = ... = r_9$ are arbitrarily close to 0 (yet still negative), they can have as little effect on the utilities as we want.
Therefore the utilities can be as close as we want to the graph in the previous question, in which we see that $v_1 < v_8$, for example.

(d)

Incorrect.
Taking, v_4 and v_6 to $-\infty$ (not really $-\infty$ but extremely small) will make us never want to go up.

(e)

With $\gamma = 0$, the utility of a state is the reward of it.
Therefore, the policy doesn't matter, and every policy is optimal.
A policy is determined by the action on each state. There are 4 actions possible from each non-terminal state so there are $|S \backslash S_G|^4$ optimal policies.

(f)

Because there's a discount $\gamma < 1$, we would never want to reach $(1, 3)$ or $(2, 4)$ early.
The optimal utilities for these states are the same so the best we can do is delay when we reach either of them, and by doing so reducing their cost (beacuse of γ).
Therefore, an optimal policy is either DOWN or RIGHT.

(g)

If the optimal policy is to go LEFT:

$$v_1 = r_1 + 0.9 \cdot \gamma \cdot v_1 + 0.1 \cdot \gamma \cdot v_2$$
$$r_1 = (1 - 0.9 \cdot \gamma)\, v_1 - 0.1 \cdot \gamma \cdot v_2$$

if the optimal policy is to go UP:

$$v_1 = r_1 + 0.8 \cdot \gamma \cdot v_2 + 0.1 \cdot \gamma \cdot v_1 + 0.1 \cdot \gamma \cdot v_3$$
$$r_1 = (1 - 0.1 \cdot \gamma)\, v_1 - \gamma\,(0.8 \cdot v_2 + 0.1 \cdot v_3)$$

Because $v_2 > v_3$, the optimal policy can't be to go DOWN or RIGHT.
Plotting in $\gamma = 0, 1$ in the above equations we get the possible extreme points of r_1:

$$r_1 = v_1$$
$$r_1 = 0.1\,(v_1 - v_2)$$
$$r_1 = 0.9 \cdot v_1 - 0.8 \cdot v_2 - 0.1 \cdot v_3 > 0.1\,(v_1 - v_2)$$

The upper bound is v_1 and the lower bound is $\min 0.1\,(v_1 - v_2)$.

Part B – Intro to Learning

Part A – dry part

(a)

(b)

(c)

(d)

(e)

(f)

Splitting the Fun