Autonomous Systems LAB Session 4: MDPs

Arnau Colom, Emma Fraxanet

I. EXERCISE D.1.BERKELEY PACMAN EXERCISES

A. Q2

In this section we have to set the noise to zero because otherwise there is a probability to move downwards/upwards and fall into the high-negative reward spaces.

B. Q3

1. **Prefer the close exit (+1), risking the cliff (-10). Discount: 0.3, Noise: 0, Living Reward: 0.** Setting the noise to zero leads to more risky policies, such as the ones moving along the edge of the cliff. In order to go towards the close exit we have to set a smaller discount, which will prefer earlier rewards.
2. **Prefer the close exit (+1), but avoiding the cliff (-10). Discount: 0.3, Noise: 0.3, Living Reward: 0.** Setting a noise parameter higher than zero will avoid risking the cliff. In order to go towards the close exit we have to set a smaller discount, which will prefer earlier rewards.
3. **Prefer the distant exit (+10), risking the cliff (-10). Discount: 0.9, Noise: 0, Living Reward: 0.** Setting the noise to zero leads to more risky policies, such as the ones moving along the edge of the cliff. In order to go towards the further exit we have to set a bigger discount, which will allow later rewards.
4. **Prefer the distant exit (+10), avoiding the cliff (-10). Discount: 0.3, Noise: 0.3, Living Reward: 0.** Setting a noise parameter higher than zero will avoid risking the cliff. In order to go towards the further exit we have to set a bigger discount, which will allow later rewards.
5. **Avoid both exits and the cliff (so an episode should never terminate). Discount: 1, Noise: 0, Living Reward: 1.** In this case we want the agent to not be in a rush, so we set the discount parameter to 1. The noise is set to 0 to avoid the cliffs (to not have any probability to fall) and the living reward is set to 1 in order to avoid the episode to terminate (the biggest reward situation is keep moving around and avoiding both the cliff and the exits).

C. Q6

Is there an epsilon and a learning rate for which it is highly likely (greater than 99%) that the optimal policy will be learned after 50 iterations?

There is not. This is because with 50 iterations there is not enough exploration (even if epsilon is set to higher values) in order to reach the high-reward exit.

II. EXERCISE D.2. PUSH YOUR LUCK

A. Description of the model

For this first section of the exercise we suppose that we have a fair dice with six faces. The values that the player can return on each roll are $d \in \mathbb{N} : d \in [1, 6]$.

The state will be described using the following variables {*empty, once, twice, Terminal, non-Terminal*}. For instance the sequence $t = \langle 6, 2, 1, 6, STOP \rangle$, would be represented with the state s = (once, once, empty, empty, empty, twice, Terminal). Since the numbers that we can obtain (the faces in the dice) is six and each number can be in three different situations, considering also the Terminal variable, the number of states that we can have is $2 \cdot 3^6 = 729$.

The player can take three actions that are: Stop the episode, re-roll another dice and start again. The set of actions can be described as Actions = {*Stop*, *Re-roll start* }. To simplify we will refer to the Stop action as $\boldsymbol{o}$, the Start action as $\boldsymbol{st}$ and the re-roll action as $\boldsymbol{f}$.

The transaction probabilities $P(s' \mid a, s)$ will depend on the action we take on each state. The transition probability from a sequence with the Terminal state will always need the action Start such that $P(s' \mid st, s) = 1$, with the new state $s' = s_{init}$ = (empty, empty, empty, empty, empty, empty, non-Terminal) and the other probabilities to zero. For the re-roll action, since we know that we are playing with a fair dice the transition probability will be described by, in the case the state s does not contain any *twice*: $P(s' \mid f, s) = \frac{1}{6}$, where the new state s' contains the information of the previous sequence s plus the change in the variable i that will account for the new sequence. To take into account the possibility where a variable is set to *twice*, any action Stop or Re-Roll applied to that state will lead to $s' = s_{init}$ = (empty, empty, empty, empty, empty, empty, non-Terminal), given by $P(s_{init} \mid f, s_{tw}) = 1$ or $P(s_{init} \mid o, s_{tw}) = 1$. This will allow to get a reward zero and initialize the episode again. If we decide to stop the play, that is we apply the Stop action, in the case where theres not any *Twice* state in the sequence the probability will be: $P = (s' \mid o, s) = 1$ if the new state s' contains the previous sequence s changing the variable non-Terminal to Terminal, and it will be 0 otherwise.

In this way the player can play infinite episodes and the state will be initialized after each stop or each time it has a number twice.

We can define our reward based on the different states of our sequence and the next state. If we have decided to collect the accumulated reward of the episode (Stop) the reward will be defined as the multiplication of all the numbers of our sequence. This can be defined by only considering reward for states that contain the variable Terminal. For example, if we stopped before getting any twice and got the State: (once, once, empty, empty, empty, once, Terminal), the reward will be 12, since it belongs to the sequence of numbers: $t = \langle 1, 2, 6 \rangle$. Therefore if we consider our state as a vector, with a 0 where it is empty, a 1 for once and a 2 for twice, the reward will be $r(a, s) = \prod_{i=1,..,6} i\hat{v}_i$, where $\hat{v}_i$ are the components of the vector. Note that when a state contained a *Twice*, we sent it to an initialized vector, where everything is empty and we have a non-terminal state. Therefore, this states will never be have any reward, as all the other intermediate states.

The initial state s_0 is the initialized state s_{init}.

B. Dice having a number N of sides.

By changing the number of sides of our dice, we need to change the size of our states, some of the transition probabilities, the reward and the initial state.

Regarding the states of our problem, they will be represented in the same way, but their number will increase exponentially $3^N \cdot 2$. The transition probabilities will be defined almost the same way as before. The only change will be that the probabilities of re-rolling a dice in the case where there is no *Twice* in the state will be $P(s' \mid f, s) = \frac{1}{N}$.

For the reward function we simply have to consider all slots in the state vector: $r(a, s) = \prod_{i=1,..,N} i\hat{v}_i$ in the case the state s has the Terminal state.

C. Variation of the game

In this case the problem would consist in a single episode and therefore wouldn't be an infinite-reward MDP. In this case we don't see how the discounted reward would be beneficial for the modelling.

III. BONUS EXERCISE

Discuss the relation between the terms "Noise" and "Living Reward" and the standard MDP formulation components. What are the transition probabilities P_a in Pacman? What is the reward function, and how is it related to the "living reward"?

The transition probabilities in this case will be defined by the noise. For example, for a 0.2 noise there is a 20% chance that the agent won't move to the direction indicated by the action. In the other cases, it will move towards the defined direction.

The living reward option simply adds a reward to states that are not terminal states. This will mean that moving from one cell to the other will produce as much reward as the living reward parameter given. In the case of setting a positive living reward with no discount parameter, like in Q3 exercise 5, moving infinitely around the board without entering any terminal state might be the most rewarding option, since it will keep accumulating reward with each move. If there is a discount parameter the option of choosing the high reward terminal state could still be the optimal one. This parameter is defaulted to zero for most of the exercises.