**Problem 1 - Modelling: Frog on a Rock**

**A cartoon of a frog jumping into water

AI-generated content may be incorrect.**

Can you define the uniform random policy on this MDP? What about the optimal policy?

Answer:

We define three states, rock0, rock1, and land. We will assume that the episode terminates once we have reached land, as that is Hop Along’s goal. For the same reason, a reasonable reward function could give a reward of 10 for reaching land (with all other rewards 0). Though other values could be used.

The state land will be an absorbing state as we are not interested in further controlling Hop Along after that. We can reach land from all our potential starting states, as long as the Reinforcement Algorithm we use does not get caught in a loop between rock0 and rock1. If we can guarantee that, then we can set our discount factor to 1.

[Can you see how such a loop might happen?]

We could set a smaller discount factor (e.g. 0.9) as this will help prioritise policies that reach land sooner. Another way of achieving the latter would be to add a small negative reward to, either all transitions to rock0 and rock1, or all actions that are not “Jump to land”, as the negative rewards would accumulate the longer the task takes to complete.

As land is an absorbing state, we have to define one action from it (which we have named “stay”) that 100% transitions the state back to land, and with a reward of 0.

For defining the policies, the uniformly random policy is defined for each state such that the probability of each possible action from that state is equal. In this case, π(a1|rock0) = 1, π(aland|rock1) = 0.5 and π(a0|rock1) = 0.5. We may also consider the possibility of a ’no-op’ action where Hop Along stays at the current location. This would of course change the possible actions and the corresponding probabilities.

The optimal policy, however, depends on the reward we receive for taking each action (and there may be multiple optimum policies depending on the reward model), so we cannot define (or learn) this until the reward model has been defined. In this case, assuming we receive a reward of 10 for reaching land and 0 otherwise, and our discount factor is 0.9, then we do have a unique, deterministic optimal policy: π(a1|rock0) = 1, π(aland|rock1) = 1. However, note that if the discount factor is 1 (there is no penalty for falling in the water), the optimal policy is not unique; in fact there are an infinite number of stochastic optimal policies with different distributions for π(a|rock1) (the random policy included) since they will all have a return of 10 eventually!

**Problem 2 - Modelling: Frog on a Rock 2**

**A cartoon of frogs jumping into a small fish

AI-generated content may be incorrect.**

**Part a**

Formulate Hop Along’s attempt to catch as many flies as possible as a MultiArmed Bandit (MAB) problem (see Sutton and Barto [2018], Ch. 2).

Answer:

We have four actions/arms to choose from at each time-step: {South, West, North, East}. In each time-step, we select one of the four actions and receive a reward (in number of flies) from the environment. Our goal is to maximise the expected total number of flies across time.

**Part b**

Can you define two simple exploration strategies that Hop Along could employ to try to maximise the total number of flies he catches? What is an advantage and disadvantage of each potential strategy?

Answer:

Hop Along could employ -greedy or UCB (Upper Confidence Bound) strategies. UCB is likely to give a lower regret (defined after T rounds as the difference between the total number of flies caught, and the expected total number of flies that would have been caught, if the optimal action had been taken from the beginning), if we can assume a well-behaved reward distribution (e.g. Gaussian), but epsilon-greedy action selection will be simpler and does not require an assumption on the reward distribution. There are of course other possible strategies with their own assumptions and corresponding advantages and disadvantages.

**Part c**

An implied assumption in Part a is that the frog will never stop jumping. If there was a limited amount φ of jumps the frog could do (let’s say φ = 100 jumps), would it still make sense to model this problem as a MAB Problem? Why? If not, how would you go about modelling this scenario?

Answer:

The problem with a limited number of jumps is that we can no longer assume that we are taking actions to infinity. In other words, we can only take a limited number of actions. [Of course, there might be values of φ for which we can still practically assume taking actions to infinity. Can you tell why?]

(You can also imagine a variation of this problem where the budget is on the available number of flies. Then we are still limited in the number of jumps, though we might not know exactly how many that is before running the experiment. [Why don’t we know exactly how many actions we can take in this version?])

It would make sense to instead model this limited number of jumps scenario as a Reinforcement Learning (RL) problem that includes a state variable for the number of jumps left/taken. As is, we would have to look for model-free RL algorithms to solve that problem. [What part of the environment model would we not have access to?]

Ultimately however, even though some theoretical guarantees do not hold when we have to take a finite set of actions, we can still model our problem as a MAB. There is no explicit assumption of an infinite horizon (i.e. end time).

Note, that there are alternative MAB formulations in the literature which could be used here instead, such as “Budget-Limited Multi-Armed Bandits” TranThanh et al. [2010]; This is an entirely optional and not necessarily useful reading, other than to point out that there are variants to the problem in the literature. However, it has been asked in class if there are cases where we treat the exploration as a different phase from exploitation, and this paper provides one such scenario, if you are curious.

You could also more generally look into MAB solutions that explicitly address the finite horizon (limited number of actions if you consider that we take one action per time-step). Try “Finite-Horizon Multi-Armed Bandit” (with and without quotations) in your favourite search engine (the search results will not be as clean as you might want them, but this is often the case with such searches).

**Part d**

Let’s go back to assuming an infinite number of jumps. Another implied assumption in Part a has been that we can control Hop Along’s actions. If we couldn’t though (perhaps because we are not Hop Along ourselves, but a researcher interested in how many flies it eats), would it still make sense to model this as a MAB problem? Why? If not, how would you go about modelling this scenario?

Answer:

The apparent answer is “no”. A Multi-Armed Bandit (MAB) problem assumes access to a space of actions1 at each time-step. If there is no action to take or, in other words, no decision to make, then we cannot formulate a MAB problem.

It might be best to simply model this as a statistical inference problem, where we are trying to learn a distribution over Hop Along’s behaviour in relation to its jumps so far.

One can of course still imagine formulating the problem as a MAB from the frog’s perspective and then comparing the frog’s actual behaviour to the solutions we come up with (perhaps we are interested in seeing whether frog behaviour approximates epsilon-greedy behaviour).

Finally, to answer the question: What if the much-troubled frog chasing flies was an oil-rig company sampling oil deposits, and the four cardinal directions were four different extraction sites?, we would need to discuss the cost of the test digs as well as the general question of using of non-renewable energies or the dangers of oil spills for marine life, which all suggests a wider study of the literature and of the problem than possible within this course. Nevertheless, the methods that are taught in this course can be useful also in more complex situation when we cooperate with the right experts and stakeholders.