June 9, 2022

3 Exercises

3.1 Question

Devise three example tasks of your own that fit into the MDP framework, identifying for each its states, actions, and rewards. Make the three examples as dierent from each other as possible. The framework is abstract and flexible and can be applied in many dierent ways. Stretch its limits in some way in at least one of your examples

Answer

Stock Trading

An agent managing the given amount of cache in a stock. The agent buys and sells in order to maximize the cache it is given.

States: Cache amount, stock amount, buy/sell prices, market indicators, global indices (e.g. risk index)

Actions: Buy, Sell, Do nothing

Rewards: Cache amount deviation from the initial amount.

Landing an UAV

An agent that can land an UAV in any weather conditions. The agent locates the landing point, considers the distance and the altitude, and controls engine thrust and control planes to the land the aircraft.

States: Distance to the landing point, altitude, azimuth, velocity, wind info, engine thrust etc.

Actions: Increase/Decrease elevation, change heading, change engine thrust, control landing gears.

Rewards: Positive reward upon landing, possibly increasing with proportionally to proximity to desired landing location. A very low reward upon crash.

Playing ATARI Games

An agent that can play atari games by looking at screen pixel images.

States: Screen pixels.

Actions: Game controls, e.g. left, right, shoot

Rewards: Positive reward upon winning. Negative reward upon loosing.

3.2 Question

Is the MDP framework adequate to usefully represent all goal-directed learning tasks? Can you think of any clear exceptions?

Answer

Agent needs to learn the reward at the end of the episode. If the environment does not provide a reward or does provide a vector of rewards the agent cannot learn.

3.3 Question

Consider the problem of driving. You could define the actions in terms of the accelerator, steering wheel, and brake, that is, where your body meets the machine. Or you could define them farther out—say, where the rubber meets the road, considering your actions to be tire torques. Or you could define them farther in—say, where your brain meets your body, the actions being muscle twitches to control your limbs. Or you could go to a really high level and say that your actions are your choices of where to drive. What is the right level, the right place to draw the line between agent and environment? On what basis is one location of the line to be preferred over another? Is there any fundamental reason for preferring one location over another, or is it a free choice?

Answer

It depends on the problem we are trying to solve.

Car driving is designed to be a function of acceleration, steering wheel and the brake. This interface is already present in any car. Natural approach would be to exploit that interface.

Now let's consider using a lower level control and use brain signals. In order to drive the brain process and produces many signals. Much of the

processed signals are even may not always be related to the driving. Produced signals are do not directly control the car but the body. More work is needed to translate the signals. The brain level control introduces a lot of extra work.

Now let's move to a higher level and use choices of locations as actions. In this case it will be necessary to relate all the locations, road conditions e.g. pedestrians, lights, and car state e.g. turning, acceleration which produces an infinite MDP. It is very hard to learn in this setting.

3.4 Question

Give a table analogous to that in Example 3.3, but for p(s', r —s, a). It should have columns for s, a, s 0 , r, and p(s', r— s, a), and a row for every 4-tuple for which p(s 0 , r —s, a) ¿ 0

Answer

Rewards are deterministic. p(s', r —s, a) column will be similar to p(s'—s, a) column. Answer will be similar to the table in example 3.3 excluding rows with 0 probability.

3.5 Question

The equations in Section 3.1 are for the continuing case and need to be modified (very slightly) to apply to episodic tasks. Show that you know the modifications needed by giving the modified version of (3.3).

Answer

$$\sum_{s' \in S^+} \sum_{r \in R} p(s', r|s, a) = 1 \text{ for all } s \in S, a \in A(s), S^+\text{being all states, } S \text{ being non-terminal states.} \tag{1}$$

3.6 Question

Suppose you treated pole-balancing as an episodic task but also used discounting, with all rewards zero except for 1 upon failure. What then would the return be at each time? How does this return differ from that in the discounted, continuing formulation of this task?

Answer

Similar to the continuing case, the return will be related to $-^K$.

The difference is that in continuing case the return will keep accumulating.

3.7 Question

Imagine that you are designing a robot to run a maze. You decide to give it a reward of +1 for escaping from the maze and a reward of zero at all other times. The task seems to break down naturally into episodes—the successive runs through the maze—so you decide to treat it as an episodic task, where the goal is to maximize expected total reward (3.7). After running the learning agent for a while, you find that it is showing no improvement in escaping from the maze. What is going wrong? Have you effectively communicated to the agent what you want it to achieve?

Answer

The robot should be incentivized to exit the maze as soon as possible.

3.8 Question

Suppose = $\gamma = 0.50$ and the following sequence of rewards is received R1 = -1, R2 = 2, R3 = 6, R4 = 3, and R5 = 2, with T = 5. What are G0, G1,. . . , G5 ? Hint: Work backwards.

Answer

$$G_t = R_{t+1} + \gamma G_{t-1} \tag{2}$$