Markov Decision Process

Sunmook Choi felixchoi@korea.ac.kr

August 6, 2023

1 Markov Decision Processes

Definition 1 (Markov Decision Process, MDP)**.** *MDP is a tuple* $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$ *where*

- $\mathcal{S}$ *is the state space,*
- $\mathcal{A}$ *is the action space,*
- P *is the transition probability,*
  - *the transition probability from the state* $s \in \mathcal{S}$ *to the state* $s' \in \mathcal{S}$ *given an action* $a \in \mathcal{A}$*.*
  - $P^a_{ss'} := P(S_{t+1} = s' | S_t = s, A_t = a)$
  - $P(S_{t+1} = s' | S_t = s) = P(S_{t+1} = s' | S_t = s, S_{t-1} = s_{t-1}, \ldots, S_1 = s_1, S_0 = s_0)$
  - *We assume stationary Markov Process, that is,* $P^a_{ss'}$ *does not depend on time step* t*.*
- $\mathcal{R}$ *is the reward function, and*
- $\gamma \in [0, 1]$ *is the discount factor.*

Example 1 (Grid World)**.** Let's consider an example, the Grid World, illustrated in Fig. 1.

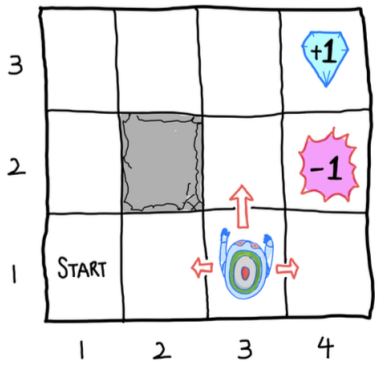

Figure 1: An example of Grid world Environment

We call a robot agent which is moving in the grid world. The thing it interacts with, comprising everything outside the agent, is called environment. There are total 11 places that an agent can stay

in, $(1,1),(2,1),\ldots,(4,3)$, except for $(2,2)$. Such places are called *states* of the grid world. The agent starts the game from the state $(1,1)$, and the game ends when it reaches one of the final states, $(4,2)$ or $(4,3)$. The agent can move either North, South, East, or West. The agent should take an *action* in order to move; the actions are North, South, East, and West. However, the grid world is *slippery* so that the next state is determined stochastically. The transition occurs corresponding to the transition probability $P^a_{ss'}$, which has the Markov Property, and an example is as follows:

$$P(s_{t+1} = (3,2)|s_t = (3,1), a_t = North) = 0.8$$
$$P(s_{t+1} = (2,1)|s_t = (3,1), a_t = North) = 0.1$$
$$P(s_{t+1} = (4,1)|s_t = (3,1), a_t = North) = 0.1.$$

Here, if the state transition is impossible due to the wall of the grid world, the agent stays in the current state. For each step, the agent gets reward from the environment. The agent gets small negative reward c (*e.g.*, $c = -0.1$) at each step until it reaches the final state. This encourages the agent to reach the final state as soon as possible. The agent gets $+1$ reward when it reaches the state $(4,3)$, and gets -1 reward when it reaches $(4,2)$. The description above defines the state space, the action space, the transition probability, and the reward function. With a hyperparameter γ, the grid world is a Markov Decision Process.

The general interaction between an agent and an environment can be illustrated as the figure below.

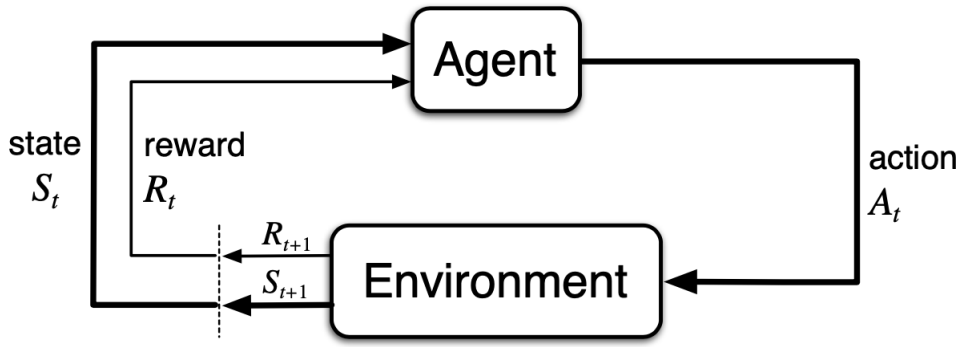

Figure 2: The agent-environment interaction in a Markov Decision Process.

Based on environments, the rewards can also be randomly given according to the current state or the chosen action. In this sense, we generalize the transition probabilities into *dynamics*, which we denote by $p(s', r|s, a)$.

The agent and environment interact at each of a sequence of discrete time steps, $t = 0, 1, 2, \ldots$ At each time step t, the agent receives its state $S_t \in \mathcal{S}$ and takes action $A_t \in \mathcal{A}$. In part as a consequence of its action, the agent receives a reward $R_{t+1} \in \mathcal{R}$, and finds itself in a new state S_{t+1}. The MDP and agent together give rise to a stochastic process or trajectory as follows:

$$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, \ldots$$

Based on the game, the episode can be either finite or infinite. If the state space of an MDP has a final state, then the episode terminates at the final state. In such cases, we usually write the trajectory as follows:

$$S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T, S_T$$

We also call the whole sequence an episode of the game/environment if it terminates at some time step. Notice that the states, actions, and rewards are written as random variables due to the dynamics and policy (we discuss about policy right in the next section).

2 Goal of the agent in an environment

The goal of an agent is to maximize the cumulative sum of rewards during the game. In this sense, given an MDP, the agent aims to **find a rule that chooses an action at each state.** We call such a rule the policy, which can be defined mathematically as a function π from the state space $\mathcal{S}$ to the action space $\mathcal{A}$.

The policy $\pi\colon \mathcal{S} \to \mathcal{A}$ can be either stochastic or deterministic. For example, in the grid world, a policy at the state (1,1) can be defined as follows:

$$\pi(a|s) = P(A_t = a|S_t = s) = \begin{cases} 0.6 & (a = North) \\ 0.4 & (a = East) \\ 0 & otherwise \end{cases} \tag{2.1}$$

The stochastic policy is indeed a conditional probability function. One should be able to discriminate two probabilities, the transition probability and the policy. Given a state $S_t = s$, the agent first chooses an action a according to the policy $\pi(\cdot|s)$.

The agent aims to find an optimal policy that maximizes the cumulative sum of rewards in an episode. This idea can be stated informally as the reward hypothesis.

Remark 1 (Reward Hypothesis)**.** *That all of what we mean by goals and purposes can be well thought of as the maximization of the expected value of the cumulative sum of rewards.*

We must stay tuned to the idea that the agent is not trying to maximize immediate reward but the cumulative sum of rewards in the long run. We define the cumulative sum of rewards formally as the return or the discounted return.

Definition 2 (Return)**.** *At each time step t, the return G_t is defined to be*

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \cdots . \tag{2.2}$$

More generally, with the discount factor $\gamma \in [0, 1]$ given in an MDP, the discounted return is defined to be

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{2.3}$$

Note that the return is also a random variable. The discount rate determines the present value of future rewards: a reward received k time steps in the future is worth only γ^{k-1} times what it would be worth if it were received immediately. We usually use the discounted return for the following reasons.

1. It is mathematically convenient. If the rewards are bounded, then the discounted return converges.
2. The future is uncertain. That is, the agents do not know how many steps they should go to earn a specific future reward.
3. For agents, immediate rewards may earn more interest than delayed rewards.

Now we formally state the goal of the agents. The agent finds an *optimal policy* that maximizes the expected return $\mathbb{E}_{\tau\sim\pi}[G_0(\tau)] = \int G_0(\tau)\,\pi(\tau)\,d\tau$. Here, τ is a random trajectory, and because of the Markov property, the probability $\pi(\tau)$ of a terminating trajectory τ is specified as follows:

$$\pi(\tau) = p(s_0) \prod_{t=0}^{T-1} \pi(a_t|s_t)\; p(s_{t+1}|s_t, a_t) \tag{2.4}$$

To summary, the agents seeks an optimal policy π^* which is equal to

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\tau\sim\pi}[G_0(\tau)]. \tag{2.5}$$

3 Value Functions

Instead of maximizing the expected discounted return directly, we consider a method that assigns some scalar value for each state or each state-action pair. The value is determined by the expected return that an agent can receive in the environment starting in the state (and taking an action). The final goal of this method is also to obtain an optimal policy which maximizes the values. With keeping this concept in mind, we introduce the value functions.

Definition 3 (Value Function)**.** *Given an MDP and a policy π of an agent, the state-value function $v_\pi(s)$ of a state $s \in \mathcal{S}$ is defined to be the expected return when starting in s and following π thereafter. Formally,*

$$v_\pi(s) = \mathbb{E}_\pi\big[G_t|S_t = s\big]. \tag{3.1}$$

The action-value function $q_\pi(s, a)$ of taking action $a \in \mathcal{A}$ in a state $s \in \mathcal{S}$ is defined to be the expected return starting from s, taking the action a, and thereafter following π. Formally,

$$q_\pi(s, a) = \mathbb{E}_\pi\big[G_t|S_t = s, A_t = a\big] \tag{3.2}$$

Note that, by the law of total probability, we have

$$v_\pi(s) = \mathbb{E}_\pi\big[G_t|S_t = s\big] = \sum_{a\in\mathbb{A}} \pi(a|s)\,\mathbb{E}_\pi\big[G_t|S_t = s, A_t = a\big] = \sum_{a\in\mathcal{A}} \pi(a|s)\,q_\pi(s, a). \tag{3.3}$$

A fundamental property of value functions used throughout reinforcement learning and dynamic programming is that they satisfy recursive relationships. Such recursive formulae are called Bellman expectation equations.

Theorem 1 (Bellman Expectation Equations)**.** *The value functions can be decomposed into the immediate reward R_{t+1} and the discounted successor value function, $\gamma v_\pi(S_{t+1})$ or $\gamma q_\pi(S_{t+1}, A_{t+1})$.*

$$v_\pi(s) = \mathbb{E}_\pi\big[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s\big] \tag{3.4}$$

$$q_\pi(s,a) = \mathbb{E}_\pi\big[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) | S_t = s, A_t = a\big] \tag{3.5}$$

Proof. □

4 Optimal Policies and Optimal Value Functions

For finite MDPs, we can precisely define an optimal policy in the following way. Value functions define a partial ordering over policies. If we say that $\pi \geq \pi'$ if and only if $v_\pi(s) \geq v_{\pi'}(s)$ for all $s \in \mathcal{S}$, then the relation $\geq$ on the set of policies becomes a partial order.

We define the optimal value function of MDP.

Definition 4 (Optimal Value Function)**.** *The optimal state-value function v_* and the optimal action-value function $q_*(s,a)$ is defined as*

$$v_*(s) = \max_\pi v_\pi(s), \quad \forall s \in \mathcal{S}, \tag{4.1}$$

$$q_\pi(s,a) = \max_\pi q_\pi(s,a), \quad \forall s \in \mathcal{S},\ \forall a \in \mathcal{A}. \tag{4.2}$$

According to the definition, we should keep in mind that $\arg\max_\pi v_\pi(s)$ and $\arg\max_\pi v_\pi(s')$ are not the same in general if $s \neq s'$. With this caution in mind, consider the following theorem.

Theorem 2 (Optimal Policies and Optimal Value Functions)**.** *Any finite MDP satisfies the following.*

1. *There exists an optimal policy $\pi_* \geq \pi$ for all π.*
2. *All optimal policies achieve the optimal state-value function $v_{\pi_*}(s) = v_*(s)$ for all $s \in \mathcal{S}$.*
3. *All optimal policies achieve the optimal action-value function $q_{\pi_*}(s,a) = q_*(s,a)$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$.*

Proof. Note that the second and the third result is immediate from the first result. □

By the theorem above, v_* is the value function of an optimal policy π_*. Therefore, we say that an MDP is solved if we know the optimal value function v_*.

An optimal policy π_* can be obtained by maximizing $q_*(s, a)$.

$$\pi_*(a|s) = \begin{cases} 1 & \text{if } a = \arg\max_a q_*(s, a) \\ 0 & \text{otherwise.} \end{cases} \tag{4.3}$$

The proof that the policy is indeed an optimal policy is given below.

Proof. □

By the optimal policy π_*, we then have

$$v_*(s) = \sum_{a \in \mathcal{A}} \pi_*(a|s)\ q_*(s, a) = \max_{a \in \mathcal{A}}\ q_*(s, a). \tag{4.4}$$

With the optimal policy given above, the Bellman expectation equation can be restated as the Bellman optimality equation.

Theorem 3 (Bellman Optimality Equation)**.**

$$v_*(s) = \max_a \sum_{s', r} p(s', r|s, a)[r + \gamma v_*(s')] \tag{4.5}$$

$$q_*(s, a) = \sum_{s', r} p(s', r|s, a)[r + \gamma \max_{a'} q_*(s', a')] \tag{4.6}$$

$$= \sum_{s', r} p(s', r|s, a)[r + \gamma v_*(s')] \tag{4.7}$$

Proof. □

Under known MDP (model-based), if we find either an optimal value function, we can find an optimal policy.