1 Reinforcement Learning: General Framework

A general reinforcement learning problem, with full observability, can be defined as follows:

Definition 1.1 *(Reinforcement Learning Problem) A RL problem is defined by a tuple* $(\mathbb{G}, \mathbb{S}, \mathbb{A}, \mathcal{P}, \Pi, \mathcal{R}, \gamma, \mathbb{T}, \mu)$*, where* $\mathbb{G} = \{g_0, g_1\}$ *is the set with the environment* g_0 *and the agent* g_1*,* $\mathbb{S}$ *is the set of states,* $\mathbb{A}$ *is the set of actions,* $\mathcal{P} : \mathbb{S} \times \mathbb{A} \to \Delta(\mathbb{S})$ *is the environment state transition function, where* $\Delta(\mathbb{S})$ *is the space of probability distributions over* $\mathbb{S}$*,* Π *is the agent's policy space,* $\mathcal{R} : \mathbb{S} \times \mathbb{A} \times \mathbb{S} \to \Delta(\mathbb{R})$ *is the reward function,* $\gamma \in [0, 1]$ *is the discount factor,* $\mathbb{T}$ *is the time set and* $\mu \in \Delta(\mathbb{S})$ *is the distribution of the initial state* $s_0 \in \mathbb{S}$*.*

While the literarure describes RL around the Markov Decision Process (MDP) [1], Definition 1.1 takes a different approach by incorporating MDPs into a broader RL problem definition. An MDP models decision-making problems where the states transitions satisfy the Markov property and are partially controlled by an agent. Formally, an MDP is defined as a tuple $(\mathbb{S}, \mathbb{A}, \mathcal{P}, \mathcal{R}, \gamma, \mathbb{T}, \mu)$, where $\mathbb{S}$ is the set of states, $\mathbb{A}$ is the set of actions, $\mathcal{P} : \mathbb{S} \times \mathbb{A} \to \Delta(\mathbb{S})$ is the state transition function, $\mathcal{R} : \mathbb{S} \times \mathbb{A} \times \mathbb{S} \to \Delta(\mathbb{R})$ is the reward function, $\gamma \in [0, 1]$ is the discount factor and $\mu \in \Delta(\mathbb{S})$ is the distribution of the initial state $s_0 \in \mathbb{S}$.

In RL there are two primary entities: the agent and the environment. The environment represents the external system with which the agent interacts. These interactions occur within a temporal context that can be either continuous or discrete and may extend over a finite or infinite time horizon. For the purposes of this discussion, we will focus on scenarios within a discrete-time framework.

The environment is characterized by a state space $\mathbb{S}$, whose dynamics are govern by a transition probability function $\mathcal{P}$. In a discrete-time setting, at each time step $t \in \mathbb{T}$, the environment is in a state $s_t \in \mathbb{S}$, with the initial state being $s_0 \sim \mu$. Given the current state s_t, the agent performs an action a_t, prompting the environment to transition to a new state $s_{t+1} \sim \mathcal{P}(s_t, a_t)$. Concurrently, the agent receives a reward $r_{t+1} \sim \mathcal{R}(s_t, a_t, s_{t+1})$. This iterative process continues indefinitely or until a termination condition is met, thus defining a trajectory $\tau_t = \{s_0, a_0, s_1, r_1, a_1, \ldots, s_t, r_t, a_t, s_{t+1}, r_{t+1}\}$, at each time step $t \in \mathbb{T}$.

Let $\mathcal{T}_t$ be the set of all trajectories of lenght t:

$$\mathcal{T}_t = \{\tau_t : \tau_t = (s_0, a_0, r_1, s_1, a_1, r_2, s_2, \ldots, s_t, a_t, r_{t+1}, s_{t+1})\}$$

The trajectory space $\mathcal{T}$ is defined as the union of all $\mathcal{T}_t$, for $t \in \mathbb{T}$:

$$\mathcal{T} = \bigcup_{t \in \mathbb{T}} \mathcal{T}_t$$

To operate within the environment, the agent selects a policy $\pi \in \Pi$, a function that maps the current state to a probability distribution over the action space $\mathbb{A}$, $\pi : \mathbb{S} \to \Delta(\mathbb{A})$. Since the environment is a MDP, the agent's decision depends only on the current state

s_t, and thus his policy takes only the current state as input. A reinforcement learning algorithm, such as Q-Learning, can be conceptualized as a function $L : \mathcal{T} \to \Pi$ that maps a realized trajectory to a policy. At each discrete time step $t \in \mathbb{T}$, given a trajectory τ_t, the agent updates his policy π_t using $L(\tau_t)$. Upon observing the current state s_t, the agent then samples an action a_t from the probability distribution defined by $\pi_t(s_t)$.

2 Partially Observable Reinforcement Learning

In an environment with partial observability, the agent doesn't have direct access to the complete state of the environment. Instead, it receives observations that may provide incomplete or noisy information about the true state. A first-price auction is a good example of a partially observable environment, where bidders don't know other bidders' private valuations or, in some cases, the total number of participants. Such scenarios are formally modeled using Partially Observable Reinforcement Learning.

Definition 2.1 *(Partially Observable Reinforcement Learning Problem) A partially observable reinforcement learning problem is defined by a tuple* $(\mathbb{G}, \mathbb{S}, \mathbb{A}, \mathbb{O}, \mathcal{P}, \mathcal{O}, \Pi, \mathcal{R}, \gamma, \mathbb{T}, \mu)$*, where* $\mathbb{G} = \{g_0, g_1\}$ *is the set containing the environment* g_0 *and the agent* g_1*,* $\mathbb{S}$ *is the set of states,* $\mathbb{A}$ *is the set of actions,* $\mathbb{O}$ *is the set of observations,* $\mathcal{P} : \mathbb{S} \times \mathbb{A} \to \Delta(\mathbb{S})$ *is the environment state transition function, where* $\Delta(\mathbb{S})$ *is the space of probability distributions over* $\mathbb{S}$*,* $\mathcal{O} : \mathbb{S} \times \mathbb{S} \to \Delta(\mathbb{O})$ *is the observation function, where* $\Delta(\mathbb{O})$ *is the space of probability distributions over* $\mathbb{O}$*,* Π *is the agent's policy space, where policies map histories of observations and actions to distributions over actions,* $\mathcal{R} : \mathbb{S} \times \mathbb{A} \times \mathbb{S} \to \Delta(\mathbb{R})$ *is the reward function,* $\gamma \in [0, 1]$ *is the discount factor,* $\mathbb{T}$ *is the time set,* $\mu \in \Delta(\mathbb{S})$ *is the distribution of the initial state* $s_0 \in \mathbb{S}$*.*

A partially observable reinforcement learning problem is structured around two fundamental entities: the environment and the agent, collectively denoted as the set $\mathbb{G}$. Within this framework, the environment exists in various states, represented by the set $\mathbb{S}$, while the agent can perform actions from the set $\mathbb{A}$. The crucial characteristic that distinguishes this from standard reinforcement learning is that the agent cannot directly observe the true state of the environment. Instead, it receives observations from the set $\mathbb{O}$, which may provide incomplete or noisy information about the actual state.

References

[1] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.