Reinforcement Learning
Markov Decision Process

Thomas Bonald
Institut Polytechnique de Paris
March 2024

Reinforcement learning refers to a set of problems where an agent takes sequential decisions and receives feedback through rewards. The actions might modify the state of the environment. This can be represented by a Markov Decision Process.

1 Markov Decision Process

Consider an agent taking sequential decisions at time \( t = 0, 1, 2, \ldots \). There are a finite set of states and a finite set of actions. At time \( t \), the agent is in state \( s_t \) and takes action \( a_t \). The agent then receives reward \( r_t \) and the environment moves to state \( s_{t+1} \). The system, known as a Markov Decision Process, is thus defined by two conditional distributions, for the reward and for the new state.

A Markov Decision Process (MDP) is defined by:

- the reward distribution, \( p(r|s,a) \),
- the state transition distribution, \( p(s'|s,a) \),

for each state \( s \) and action \( a \).

Some states might be terminal, meaning that the process stops. This is the case of most games (e.g., chess). We denote by \( S \) the set of non-terminal states. Let \( A \) be the set of actions. Some actions might be forbidden in some states. We denote by \( A(s) \subset A \) the set of all available actions in state \( s \in S \).

Assuming discrete rewards, we have:

\[
\forall s \in S, \forall a \in A(s), \, \sum_r p(r|s,a) = 1.
\]

Similarly, we have the the state transitions:

\[
\forall s \in S, \forall a \in A(s), \, \sum_{s'} p(s'|s,a) = 1.
\]

In some environments, the reward depends only on new state \( s' \), i.e., the reward is \( r = f(s') \) for some deterministic function \( f \). This is the case of most games for instance, where the reward (+1 for a win, −1 for a defeat and 0 otherwise) is a simple (known) function of the state \( s' \). Observe that this is a particular case of the above framework, with:

\[
p(r|s,a) = \sum_{s',r=f(s')} p(s'|s,a).
\]
2 Policy

The policy defines the behavior of the agent in each non-terminal state. Specifically, it is a probability distribution over the actions, conditionally to the state.

We say that the agent applies policy $\pi$ if the probability to take action $a$ in state $s$ is $\pi(a|s)$.

So a policy is stochastic in general, and we have:

$$\forall s \in S, \sum_{a \in A(s)} \pi(a|s) = 1.$$ 

Given a policy, the sequence of states $s_0, s_1, s_2, \ldots$ defines a Markov chain with state transition distribution:

$$\forall s \in S, \quad p(s'|s) = \sum_{a \in A(s)} \pi(a|s)p(s'|s, a).$$

When the policy is deterministic, we use the simple notation $\pi(s)$ for the action selected in state $s$.

For a deterministic policy $\pi$, we denote by $a = \pi(s)$ the action taken in state $s$.

The objective of reinforcement learning is to find an optimal policy, in a sense to be defined later. In particular, we might consider a sequence of policies $\pi_0, \pi_1, \pi_2, \ldots$, corresponding to different versions of the learning agent, converging to the optimal policy. Each such policy will define a Markov chain for the sequence of states. We will also consider a single policy that evolves over time, while the agent interacts with the environment. In this case, the probability distribution $\pi$ is not stationary and the resulting sequence of states $s_0, s_1, s_2, \ldots$ is no longer a Markov chain.

3 Value function

The agent will collect a sequence of rewards $r_0, r_1, r_2, \ldots$, possibly finite. The objective is to maximize the gain, defined as the discounted total reward.

The agent aims at maximizing the gain:

$$G = r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots,$$

where $\gamma \in [0, 1]$ is the discount factor.

In the absence of terminal states, we take $\gamma < 1$.

The value function of a policy characterizes its expected gain in each state.

The value of state $s$ under policy $\pi$ is the expected gain when starting from $s$, that is:

$$V_{\pi}(s) = E(G|s_0 = s)$$

with the convention that $V_{\pi}(s) = 0$ for all $s \notin S$ (terminal states).
4 Bellman’s equation

The gain $G$ is a random variable whose probability distribution is not explicit. To compute the value function of a policy $\pi$, we can use Bellman’s equation, exploiting the Markov property of the system. The proof is provided in the Appendix.

The value function $V_\pi$ of policy $\pi$ is a solution to **Bellman’s equation**:

$$\forall s \in S, \quad V(s) = E(r_0 + \gamma V(s_1)|s_0 = s)$$

This defines a linear system with $n = |S|$ variables, written in developed form as:

$$\forall s \in S, \quad V(s) = \sum_{a \in A(s)} \pi(a|s) \sum_r r p(r|s,a) + \gamma \sum_{a \in A(s)} \pi(a|s) \sum_{s' \in S} V(s') p(s'|s,a).$$

Solving this system exactly involves the inversion of a square matrix of size $n$, for a computational cost in $O(n^3)$. In practice, we can find a very good approximation by fixed-point iteration, for a computational cost in $O(kn^2)$ where $k$ is the number of iterations. The convergence is geometric at rate $\gamma$, as shown in the Appendix.

If $\gamma < 1$, the value function $V_\pi$ of policy $\pi$ is the unique solution to Bellman’s equation and follows from the fixed-point iteration:

$$\forall s \in S, \quad V(s) \leftarrow E(r_0 + \gamma V(s_1)|s_0 = s)$$

Appendix

A Proof of Bellman’s equation

By definition,

$$G = r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots,$$

$$= r_0 + \gamma (r_1 + \gamma r_2 + \ldots),$$

$$= r_0 + \gamma G_1,$$

where $G_1$ is the gain starting from state $s_1$. We deduce:

$$V_\pi(s) = E(G|s_0 = s) = E(r_0|s_0 = s) + \gamma E(G_1|s_0 = s).$$

By conditional expectation,

$$E(G_1|s_0 = s) = E[E(G_1|s_0 = s, s_1)|s_0 = s].$$

Now it follows from the Markov property that:

$$E(G_1|s_0 = s, s_1) = E(G_1|s_1) = V_\pi(s_1).$$

We conclude that:

$$V_\pi(s) = E(r_0 + \gamma V_\pi(s_1)|s_0 = s).$$
B Proof of the fixed-point iteration

Let $F$ be the operator defined by:

$$F(V) = E(r_0 + \gamma V(s_1)|s_0 = s),$$

for any function $V : S \rightarrow \mathbb{R}$.

Considering the sup norm, we get:

$$||F(V) - F(U)||_{\infty} = \gamma \sup_{s \in S} |E(V(s_1) - U(s_1)|s_0 = s)|,$$

$$= \gamma \sup_{s \in S} \sum_{s'} p(s_1 = s'|s_0 = s)(V(s') - U(s'))|,$$

$$\leq \gamma \sup_{s' \in S} |V(s') - U(s')|,$$

$$= \gamma ||V - U||_{\infty}.$$

Thus the operator is contracting for the sup norm, and the convergence is a consequence of Banach fixed-point theorem.