Lecture 1: MDP Basics and Planning

Qi Zhang (qzhang9@wpi.edu)

Last Updated: August 2025

1 Markov Decision Processes: The Finite-Horizon Setting

In reinforcement learning, the interaction between the agent and its environment is often formulated as a Markov Decision Process (MDP). This note focuses on the *finite-horizon* setting where an MDP is specified by tuple $M = (\mathcal{S}, \mathcal{A}, P, R, H)$:

- State space $\mathcal{S}$. This note only considers finite state spaces.
- Action space $\mathcal{A}$. This note only considers finite action spaces.
- Horizon H. This is a constant positive integer. Let $h \in [H] := \{1, 2, \ldots, H\}$ index the discrete timesteps.
- Transition function $P : \mathcal{S} \times \mathcal{A} \times [H] \to \Delta(\mathcal{S})$, where is the space of probability distributions over $\mathcal{S}$. $P_h(s'|s, a)$ is the probability of transiting to state s' after taking action a in state s.
- Reward function $R : \mathcal{S} \times \mathcal{A} \times [H] \to [0, 1]$. $R_h(s, a)$ is the immediate reward associated with taking action a in state s at timestep h.

1.1 Interaction protocol

Starting in some state $s_1 \in \mathcal{S}$, at each timestep $h \in [H]$, the agent takes an action $a_h \in \mathcal{A}$, obtains the immediate reward $r_h := R_h(s_h, a_h)$, and observes the next state $s_{h+1} \in \mathcal{S}$ sampled from $P_h(s_h, a_h)$, or $s_{h+1} \sim P_h(s_h, a_h)$. The interaction record

$$\tau = (s_1, a_1, r_1, s_2, \ldots, s_H, a_H, r_H, s_{H+1})$$

is called an *episode*, which is a trajectory of length H.

The process above can be iterated to form multiple episodes. We often consider the case where the initial state s_1 is fixed. More generally, s_1 is generated by sampling from a distribution $d_1 \in \Delta(\mathcal{S})$. When d_1 is of importance to the discussion, we include it as part of the MDP tuple, writing $M = (\mathcal{S}, \mathcal{A}, P, R, H, d_1)$.

2 Policy and Value

A *policy* specifies how the actions are chosen. A policy denoted as $\pi : \mathcal{S} \times [H] \to \mathcal{A}$ chooses actions deterministically based on the current state and the timestep, i.e., $a_h = \pi_h(s_h)$. More generally, a policy denoted as $\pi : \mathcal{S} \times [H] \to \Delta(\mathcal{A})$ chooses actions stochastically denoted as $a_h \sim \pi_h(s_h)$, writing the probability as $\pi_h(a_h|s_h)$. It is important to base the policy on the timestep h, because the optimal decision-making strategy may well depend on h in this finite-horizon setting. We therefore often write $\pi = \{\pi_h\}_{h=1}^{H}$.

Given MDP $M = (\mathcal{S}, \mathcal{A}, P, R, H, d_1)$ and policy $\pi = \{\pi_h\}_{h=1}^{H}$ in place, one can explicitly write out the probability of generating any trajectory $\tau = (s_1, a_1, r_1, s_2, \ldots, s_H, a_H, r_H, s_{H+1})$:

$$\Pr^{M,\pi}(\tau) = d_1(s_1)\pi_1(a_1|s_1)P_1(s_2|s_1, a_1) \cdots \pi_H(a_H|s_H)P_H(s_{H+1}|s_H, a_H)$$

where rewards will be deterministically generated as $r_h = R_h(s_h, a_h)$.

The goal of the agent is to find a policy π that maximizes the expected sum of rewards received in the future, or *values*. In MDP M, following policy π from state $s \in \mathcal{S}$ at timestep $h \in [H]$, the expected cumulative reward to the end is denoted as $V_h^{M,\pi}(s)$, i.e.,

$$V_h^{M,\pi}(s) := \mathbb{E}_{M,\pi}\left[\sum_{h'=h}^{H} R_{h'}\left(s_{h'}, a_{h'}\right) \mid s_h = s\right],$$

which is called the state-value. Similarly, the action-value (or Q-value) is defined as

$$Q_h^{M,\pi}(s,a) := \mathbb{E}_{M,\pi}\left[\sum_{h'=h}^{H} R_{h'}\left(s_{h'}, a_{h'}\right) \mid s_h = s, a_h = a\right]$$

i.e., the expected cumulative reward by following policy π in MDP M from taking action a in state $s \in \mathcal{S}$ at timestep $h \in [H]$.

When the MDP M is clear from the context, we often drop superscript/subscript M and write V_h^π, Q_h^π, $\Pr^\pi$, $\mathbb{E}_\pi$, etc.

The goal of the agent is to find a policy π that maximizes $V_1^\pi\left(s_1\right)$ if the initial state s_1 is fixed or $\mathbb{E}_{s_1 \sim d_1}[V_1^\pi\left(s_1\right)]$ if the initial state s_1 is sampled from distribution d_1.

3 Planning in MDPs

Planning refers to the problem of computing a value-maximizing policy given the full MDP specification $M = (\mathcal{S}, \mathcal{A}, P, R, H)$. A related problem is *policy evaluation*, which aims to obtain the values of a given policy.

3.1 Bellman equation for policy evaluation

Policy evaluation is the problem of finding the values (V_h^π and/or Q_h^π) of a given policy π. In the context of planning, we also assume the full knowledge of the MDP specification M. By definition, the values can be computed by expanding the expectation, e.g.,

$$V_1^\pi(s) = \sum_{\tau : s_1 = s}\left[\Pr^\pi(\tau | s_1 = s) \cdot \textstyle\sum_{h=1}^{H} r_h\right]$$

where $\tau : s_1 = s$ enumerates all trajectories with the first state $s_1 = s$ and $\Pr^\pi(\tau|s_1 = s) = \pi_1(a_1|s_1 = s)P_1(s_2|s_1 = s, a_1) \cdots \pi_H(a_H|s_H)P_H(s_{H+1}|s_H, a_H)$ is the probability of getting such a trajectory τ by following π starting in $s_1 = s$. This approach is highly inefficient because it enumerates the trajectories. In particular, there are in total $|\mathcal{S} \times \mathcal{A}|^H$ trajectories of length H, which is exponentially large in horizon H.

A more efficient way to do policy evaluation is based on the principles of dynamic programming. By definition, the values can be computed recursively via the following *Bellman equations*: letting $V_{H+1}^\pi(s) \equiv 0$, $\forall s \in \mathcal{S}, a \in \mathcal{A}, h \in [H]$,

$$\begin{aligned} V_h^\pi(s) &= \mathbb{E}_{a \sim \pi_h(s)}[Q_h^\pi(s,a)], \\ Q_h^\pi(s,a) &= R_h(s,a) + \mathbb{E}_{s' \sim P_h(s,a)}[V_{h+1}^\pi(s')] \end{aligned} \tag{1}$$

where the recursion proceeds backward in time as $h = H, H-1, \dots, 1$, so the computation is linear in horizon H.

It is often convenient to rewrite Eq. (1) in a matrix-vector form. Since $\mathcal{S}$ and $\mathcal{A}$ are assumed to be finite, upon fixing an arbitrary order of states and actions, we can treat the relevant quantities as matrices or vectors of proper shapes. Specifically, Eq. (1) can be rewritten as

$$\begin{aligned} V_h^\pi &= (\pi_h \odot Q_h^\pi)\,\mathbf{1} \quad \text{with } V_h^\pi \in \mathbb{R}^{|\mathcal{S}|}, \pi_h, Q_h^\pi \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|},\ \text{all-one vector } \mathbf{1} \in \mathbb{R}^{|\mathcal{A}|}, \\ Q_h^\pi &= R_h + P_h V_h^\pi \quad \text{with } Q_h^\pi, R_h \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}, P_h \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|\times|\mathcal{S}|}, V_{h+1}^\pi \in \mathbb{R}^{|\mathcal{S}|}\,. \end{aligned}$$

Note that we treat Q_h^π above as a matrix in the first equation and a vector in the second equation.

3.2 Bellman optimality equation

With a slight modification of replacing the $\mathbb{E}_{a\sim\pi_h(s)}$ in Bellman equations with the greedy action $\max_{a\in\mathcal{A}}$, we obtain the *Bellman optimality equations*: letting $V_{H+1}^*(s) \equiv 0$, $\forall s \in \mathcal{S}, a \in \mathcal{A}, h \in [H]$,

$$\begin{aligned} V_h^*(s) &:= \max_{a\in\mathcal{A}} Q_h^*(s,a), \\ Q_h^*(s,a) &:= R_h(s,a) + \mathbb{E}_{s'\sim P_h(s,a)}[V_{h+1}^*(s')] \end{aligned} \tag{2}$$

which recursively define quantities V_h^* and Q_h^*. Consider the policy that is acting greedy with respect to Q_h^*, i.e.,

$$\pi_h^*(s) := \arg\max_{a\in\mathcal{A}} Q_h^*(s,a) \quad \forall s \in \mathcal{S} \tag{3}$$

We have Proposition 1 showing that π^* is an optimal policy with its values equal to V^* and Q^*.

Proposition 1. For V^* and Q^* defined in (2) and policy π^* defined in (3), we have

$$V_h^*(s) = V_h^{\pi^*}(s) = \max_\pi V_h^\pi(s), \quad Q_h^*(s,a) = Q_h^{\pi^*}(s,a) = \max_\pi Q_h^\pi(s,a) \qquad \forall (s,a,h)$$

where the maximization is taken over all policies $\pi = \{\pi_h\}_{h=1}^H$.

Proof. We prove it by induction over $h = H+1, H, \dots, 1$.

Base case. For $h = H+1$, we have $V_{H+1}^*(s) \equiv 0 = V_{H+1}^{\pi^*}(s) = \max_\pi V_{H+1}^\pi(s)$ because $V_{H+1}^\pi(s) \equiv 0$ for any π.

Assume the statement in the proposition holds for $h+1$, i.e., $\forall s \in \mathcal{S}, a \in \mathcal{A}$

$$V_{h+1}^*(s) = V_{h+1}^{\pi^*_{h+1:H}}(s) = \max_{\pi_{h+1:H}} V_{h+1}^{\pi_{h+1:H}}(s), \quad Q_{h+1}^*(s,a) = Q_{h+1}^{\pi^*_{h+1:H}}(s,a) = \max_{\pi_{h+1:H}} Q_{h+1}^\pi(s,a).$$

Note here we explicitly write $V_{h+1}^\pi \equiv V_{h+1}^{\pi_{h+1:H}}$ (and for Q as well) to emphasize the fact that the value function for timestep $h+1$ does not depend on policy before that timestep. The induction proceeds to $h \in [H]$ in the following three steps.

1) Q_h^* is optimal. We have, $\forall(s,a)$,

$$\begin{align*}
&\max_{\pi}\ Q_h^{\pi}(s,a) \\
=&\max_{\pi}\ R_h(s,a)+\mathbb{E}_{s'\sim P_h(s,a)}[V_{h+1}^{\pi}(s')] \tag{1a} \\
=&R_h(s,a)+\max_{\pi}\mathbb{E}_{s'\sim P_h(s,a)}[V_{h+1}^{\pi}(s')] \\
=&R_h(s,a)+\mathbb{E}_{s'\sim P_h(s,a)}\left[V_{h+1}^{\pi^*_{h+1:H}}(s')\right] \tag{1b} \\
=&R_h(s,a)+\mathbb{E}_{s'\sim P_h(s,a)}[V_{h+1}^{*}(s')] \tag{Induction hypothesis} \\
=&Q_h^*(s,a) \tag{1c}
\end{align*}$$

2) V_h^* is optimal. We have, $\forall s$,

$$\begin{align*}
\max_{\pi}\ V_h^{\pi}(s) =&\max_{\pi}\ \sum_a \pi_h(a|s)Q_h^{\pi}(s,a) \\
=&\max_{\pi_h,\pi_{h+1:H}}\ \sum_a \pi_h(a|s)Q_h^{\pi_{h+1:H}}(s,a) \\
=&\max_{\pi_h}\ \sum_a \pi_h(a|s)Q_h^{*}(s,a) \tag{1d} \\
=&\max_{a}\ Q_h^*(s,a) \\
=&V_h^*(s) \tag{1e}
\end{align*}$$

3) $\pi^*_{h:H}$ is optimal. We have, $\forall s$,

$$\begin{align*}
V_h^{\pi^*_{h:H}}(s) =&Q_h^{\pi^*_{h+1:H}}(s,\bar{a}) \tag{where $\bar{a}:=\pi_h^*(s)=\arg\max_{a\in\mathcal{A}}Q_h^*(s,a)$} \\
=&R_h(s,\bar{a})+\mathbb{E}_{s'\sim P_h(s,\bar{a})}\left[V_{h+1}^{\pi^*_{h+1:H}}(s')\right] \\
=&R_h(s,\bar{a})+\mathbb{E}_{s'\sim P_h(s,\bar{a})}\left[V_{h+1}^{*}(s')\right] \tag{1f} \\
=&Q_h^*(s,\bar{a}) \\
=&\max_a Q_h^*(s,a)=V_h^*(s)
\end{align*}$$

The proof completes because steps (1,2,3) finish the induction on h. □

References