**CSC 696H: Topics in reinforcement learning theory** **Fall 2024**

# Lecture 9: Episodic MDPs and Policy Evaluation

*Lecturer: Dr. Chicheng Zhang* *Scribe: Chi-Heng Yang*

# 1 Episodic Markov Decision Processes (MDPs)

An episodic MDP starts with an initial state $s$ drawn from a distribution $\mu$. The agent proceeds in the following manner for each step $h = 1, \dots, H$:

1. Observe a state $S_h$.
2. Take action $A_h$.
3. Get reward $r_h = R(S_h, A_h)$, where $R$ is the reward function.
4. Store transition $S_{h+1} \sim P_h(\cdot \mid s_h, a_h)$, where $P$ is the state transition function.

## 1.1 Performance Measures

The performance measure is the expected return:

$$\mathbb{E}\left[\sum_{h=1}^{H} r_h\right]$$

This is a random variable due to randomness in state transitions and potential randomization in actions.

## 1.2 Policy Types

- **Markovian Policy ($\Pi^M$)**: Each $\pi_h(a \mid s)$ is a conditional probability of taking action $a$ given state $s$. It only depends on the current state.
- **History-dependent Policy ($\Pi$)**: Each $\pi_h(a \mid s_1, a_1, \dots, s_{h-1})$ depends on the entire history up to step $h-1$.

# 2 Planning (Optimal Control)

The goal is to find a policy $\pi$ that maximizes the expected return:

$$J(\pi) = \mathbb{E}\left[\sum_{h=1}^{H} r_h \middle| \pi\right]$$

Given inputs $((R_h)_{h=1}^H, (P_h)_{h=1}^H, \mu)$, it turns out that it suffices to restrict our search to Markovian policies $\pi^M$. In fact, we have:

$$\max_{\pi \in \Pi} J(\pi) = \max_{\pi \in \Pi^M} J(\pi)$$

## 2.1 Policy Evaluation

A natural question is how to compute $J(\pi)$ given $\pi \in \Pi^M$. This process is also known as *policy evaluation.*

**Definition 1** (Value Function of a Policy)**.** *Given a policy $\pi = (\pi_1, \ldots, \pi_H) \in \Pi^M$, we define its value function as follows:*

*For step $h = 1, \ldots, H$, the value function $V_h^\pi(s)$ is given by:*

$$V_h^\pi(s) = \mathbb{E}\left[\sum_{t=h}^{H} r_t \mid s_h = s, \pi\right]$$

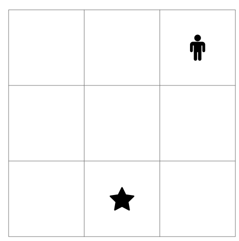
We have the following result for the value function:

$$\mathbb{E}_{s_1 \sim \mu}[V_1^\pi(s_1)] = J(\pi)$$

Conventionally, this function will be used for $h = 1, \ldots, H$. After step $H$, we no longer collect rewards. That is, $V_{H+1}^\pi(s) = 0$

**Example 1.** *In a grid world, where the agent get reward of 1 if at ★ and 0 otherwise, the policy $\pi_h(s) =$ stay $\forall s$ have the value function:*

$$V_1^\pi(s) = \begin{cases} 0 & \text{if } s \neq \bigstar \\ H & \text{if } s = \bigstar \end{cases}$$

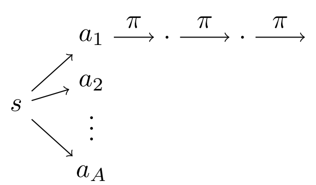


**Definition 2.** *Action Value Function The action value function for a step $h$ is defined as:*

$$Q_h^\pi(s,a) = \mathbb{E}\left[\sum_{t=h}^{H} r_t \middle| s_h = s, a_h = a, \pi\right]$$

## 2.2 Representing $V_h^\pi$ Using $Q_h^\pi$

Consider the action selection at step $h$ under the state $s$:




If we select $a_1$, then the expected return $= Q_h^\pi(s, a_1)$. Let $\mathcal{A} = \{a_1, \ldots, a_A\}$ be the action space. The value function can be represented using the action value function as:

$$V_h^\pi(s) = \sum_{a \in \mathcal{A}} \pi_h(a \mid s) Q_h^\pi(s,a) \tag{1}$$

This representation can also be written as the inner product of the policy $\pi_h$ and the action value function $Q_h^\pi$:

$$\langle \begin{pmatrix} \pi(a_1|s) \\ \vdots \\ \pi(a_A|s) \end{pmatrix}, \begin{pmatrix} Q_h^\pi(s,a_1) \\ \vdots \\ Q_h^\pi(s,a_A) \end{pmatrix} \rangle = \langle \pi(\cdot|s), Q_h^\pi(s,\cdot) \rangle$$

### 2.3 Representing $Q_h^\pi$ Using $V_{h+1}^\pi$

Consider the state transition after selecting action $a$ at state $s$ at step $h$:

$$\begin{tikzcd}
 & & s_1 \arrow[r, "\pi"] & \cdot \arrow[r, "\pi"] & \cdot \arrow[r, "\pi"] & \cdot \\
 & & s_2 \\
s \arrow[r] & a \arrow[uur] \arrow[ur] \arrow[dr] & \vdots \\
 & & s_S
\end{tikzcd}$$

The expected return after transitioning to state $s_1$ is $V_{h+1}^\pi(s_1)$. Considering all possible state transitions after taking the action $a$, we sum together the product of transition probabilities $P_h(s'|s,a)$ and the expected return $V_{h+1}^\pi(s')$ at all possible staes $s' \in S$ to get the overall expected return:

$$\sum_{s' \in \mathcal{S}} P_h(s' \mid s,a) V_{h+1}^\pi(s')$$

However, taking action $a$ at state $s$ also leads to an immediate reward $R_h(s,a)$. Therefore, The action value function $Q_h^\pi(s,a)$ can then be represented as:

$$Q_h^\pi(s,a) = \sum_{s' \in \mathcal{S}} P_h(s' \mid s,a) V_{h+1}^\pi(s') + R_h(s,a) \tag{2}$$

Similarly, this can be represented in inner products:

$$\langle \begin{pmatrix} P_h(s_1|s,a) \\ \vdots \\ P_h(s_S|s,a) \end{pmatrix}, \begin{pmatrix} V_{h+1}^\pi(s_1) \\ \vdots \\ V_{h+1}^\pi(s_S) \end{pmatrix} \rangle + R_h(s,a) = \langle P_h(\cdot|s,a), V_{h+1}^\pi(s|\cdot) \rangle + R_h(s,a)$$

Therefore, we can compute $V_1^\pi$ and $J(\pi)$ by these two equations (also known as the **Bellman Consistency Equation**). In particular, we know that $V_{H+1}^\pi \equiv 0$, then we can perform the following process:

- Compute $Q_H^\pi$ by $V_{H+1}^\pi$ using equation 2
- Compute $V_H^\pi$ by $Q_H^\pi$ using equation 1
- Compute $Q_{H-1}^\pi$ by $V_H^\pi$ using equation 2
- ...
- Compute $V_1^\pi$ by $Q_1^\pi$ using equation 1

**Definition 3** (Bellman Backup Operator)**.** *Given an MDP $M$, for step $h$, define the Bellman backup operator $\mathcal{T}_h^\pi$.*

- *Input:* $f : S \times A \to \mathbb{R}$
- *Output:* $(\mathcal{T}_h^\pi f) : S \times A \to \mathbb{R}$

*The Bellman backup operator is given by:*

$$(\mathcal{T}_h^\pi f)(s,a) = R_h(s,a) + \sum_{s' \in \mathcal{S}} \sum_{a' \in A} P_h(s'|s,a) \pi_{h+1}(a'|s') f(s',a')$$

Applying it to the action value function, we have $\mathcal{T}_h^\pi Q_{h+1}^\pi = Q_h^\pi$. This is another form of the Bellman Consistency Equation.

# 3 Finding the Optimal Policy

How to find optimal policy $\pi \in \Pi^M$ that maximize $J(\pi)$?

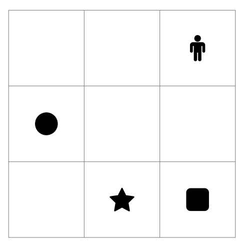
**Definition 4** (Optimal Value Function of a Policy)**.** *For MDP $M$, we define its optimal value function: For step $h = 1, \ldots, H$, the optimal value function $V_h^*(s)$ is given by:*

$$V_h^*(s) = \max_{\pi \in \Pi^M} \mathbb{E}\left[\sum_{t=h}^{H} r_t \mid s_h = s, \pi\right]$$

This objectively meausres how advantageous state $s$ is. Similar the the optimal policy definition, we use the convention that $V_{H+1}^*(s) = 0$.

**Example 2.** *Consider the same grid world, where the agent get reward of 1 if at $\bigstar$ and 0 otherwise. The optimal value function at the following state would be:*

- *At state $s = \bigstar$: $V_1^*(s) = H$, which can be achieved by staying for all steps $1, \ldots, H$*
- *At state $s = \blacksquare$: $V_1^*(s) = H - 1$, which can be achieved by going left at the first step (no rewards) and then stay until $H$*
- *At state $s = \bullet$: $V_1^*(s) = H - 2$, which can be acheived by going right and down at the first two steps (no rewards) and then stay until $H$*



**Definition 5.** *Optimal Action Value Function The optimal action value function for a step $h$ is defined as:*

$$Q_h^*(s, a) = \max_{\pi \in \Pi^M} \mathbb{E}\left[\sum_{t=h}^{H} r_t \middle| s_h = s, a_h = a, \pi\right]$$

Given these, the policy $\pi_h^* = (\pi_h^*)_{h+1}^H$: $\pi_h^*(s) = \arg\max_{a \in \mathcal{A}} Q_h(s, a)$ is the optimal policy.

## 3.1 Representing $V_h^*$ Using $Q_h^*$

Consider the action selection at step $h$ under the state $s$, how do we select the opimal action?

$$\begin{tikzcd}
 & a_1 \arrow[r, "\pi"] & \cdot \arrow[r, "\pi"] & \cdot \arrow[r, "\pi"] & \cdot \\
s \arrow[ur] \arrow[r] \arrow[ddr] & a_2 & & & \\
 & \vdots & & & \\
 & a_A & & &
\end{tikzcd}$$

Suppose that we act optimally for all steps after selecting the action $a_1$, then the expected return would be $Q_h^*(s, a_1)$. The action that should be taken at step $h$ is the one that optimize expected future return. That is,

$$a^* = \arg\max_{a \in \mathcal{A}} Q_h^\pi(s, a)$$

Therefore, the optimal value at state $s$ is the $Q_h^*$ with the optimal action taken. That is,

$$V_h^*(s) = \max_{a \in \mathcal{A}} Q_h^*(s, a)$$

## 3.2 Representing $Q_h^*$ Using $V_{h+1}^*$

Consider the state transition after selecting action $a$ at state $s$ at step $h$:

$$
\begin{tikzcd}
 & & s_1 \arrow[r, "\pi"] & \cdot \arrow[r, "\pi"] & \cdot \arrow[r, "\pi"] & \cdot \\
 & & s_2 & & & \\
s \arrow[r] & a \arrow[uur] \arrow[ur] \arrow[dr] & \vdots & & & \\
 & & s_S & & &
\end{tikzcd}
$$

Suppose that we act optimally for all steps after transitioning to state $s_1$ at step $h+1$, then the expected value would be $V_{h+1}^*(s_1)$. Therefore, the optimal action value function $Q_h^*$ can be represented by considering the transition function $P_h$ and the immediate reward $R_h(s,a)$:

$$Q_h^*(s,a) = \sum_{s' \in \mathcal{S}} P_h(s' \mid s,a) V_{h+1}^*(s') + R_h(s,a)$$

## 3.3 Fact

We have (can be shown by induction from $h = H$)

1. Policy $\pi$ step $h$, $V_h^* \geq V_h^{\pi^*}$ (by definition), and $Q_h^* \geq Q_h^{\pi}$.
2. Policy $\pi^*$: $V_h^* = V_h^{\pi^*}$, $Q_h^* = Q_h^{\pi^*}$

## 3.4 Bellman Backup Equation (Revisited)

The Bellman backup equation for step $h$ is given by:

$$(\mathcal{T}_h^* f)(s,a) = R_h(s,a) + \sum_{s' \in \mathcal{S}} P_h(s' \mid s,a) \max_{a' \in A} f(s',a')$$

Thus:

$$\mathcal{T}_h^* Q_{h+1} = Q_h^*$$

**Remark 1.** *The process of iteratively applying $Q_h^* = \mathcal{T}_h^* Q_{h+1}^*$, $h = H, \ldots, 1$, is called* value iteration.

In an infinite horizon setting, we introduce a discount factor $\gamma < 1$, and the expected return becomes:

$$(\mathcal{T}^* f)(s,a) = R_h(s,a) + \gamma \sum_{s' \in \mathcal{S}} P_h(s' \mid s,a) \max_{a'} f(s',a')$$

# 4 Online Reinforcement Learning

## 4.1 Interaction Protocol

The agent knows $(R_h)_{h=1}^H$ but does not know $(P_h)_{h=1}^H$.

For episodes $t = 1, 2, \ldots T$:

- See initial state $s_1^t \sim \mu$.
- For steps $h = 1, \ldots, H$:
  - Observe $S_h^t$.
  - Take action $A_h^t$.
  - Get reward $r_h^t = R_h(s_h^t, a_h^t)$.
  - Transition to $S_{h+1}^t \sim P_h(\cdot \mid s_h^t, a_h^t)$.

### 4.2 Regret Minimization

The goal is to minimize the regret:

$$\text{Reg}(T) := TJ(\pi^*) - \mathbb{E}\left[\sum_{t=1}^{T} J(\pi^t)\right]$$

where $TJ(\pi^*)$ is the optimal return for all $T$ episodes and $\pi^t$ is the policy used at episode $t$. The regret measures the difference between the optimal return for all $T$ episodes and the cumulative return obtained by the agent's policies over $T$ episodes. When the initial state $s_1$ is deterministic, we can write the regret as:

$$\text{Reg}(T) = V_1^*(s_1) - V_1^{\pi^t}(s_1)$$

To solve this, we apply the optimism principle, which defines the bonus to motivate the model to explore.

- Model optimism: Use the transition probabilities to represent the world and estimate them based on the collected trajectory data. We then maintain a confidence set of the transition probabilities. Then apply the optimism principle so that it gives us the highest possible reward under the best possible transition.
- Value optimism: Don't maintain a model estimate of the world. Intead, just construc optimisitc upper bounds on the optimal $Q$ and $V$ functions and take the action greedily with resepect to these upper bounds.