Lecture 5: Markov Decision Processes

Akshay Krishnamurthy
akshay@cs.umass.edu

February 21, 2022

1 Recap and Introduction

So far, we have been discussing simple exploration problems in various "bandit" formulations. In all of these problems the agent chooses actions which influence the immediate reward but, in the bandit protocols, these actions have no further influence on future interactions. This means that our actions have no long-term consequences, or equivalently, credit assignment is relatively straightforward. However, many sequential decision making scenarios cannot be effectively modeled as bandit problems because actions do have long-term consequences. Developing algorithms for these settings requires introducing new interaction protocols/models.

While we have seen relatively sophisticated algorithms for bandit problems, such as LinUCB and SquareCB, to incorporate credit assignment, we'll have to take a step back and simplify considerably. In particular, today we'll think only about the credit assignment capability, in the absence of both exploration and generalization.

2 Markov Decision Processes

A Markov decision process formalizes a decision making problem with *state* that evolves as a consequence of the agents actions. The schematic is displayed in Figure 1

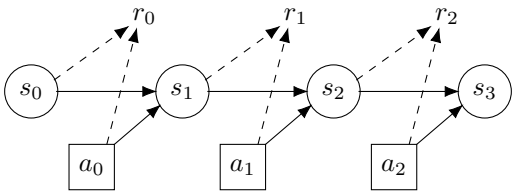

Figure 1: A schematic of a Markov decision process

Here the basic objects are:

- A state space $\mathcal{S}$, which could be finite or infinite. For now let us think of this as finite and relatively small.
- An action space $\mathcal{A}$, which could also be finite or infinite. Again let's assume this is small for now.
- A reward function $R : \mathcal{S} \times \mathcal{A} \to \Delta([0, 1])$ that associates a distribution over rewards to each state-action pair. We'll say that $r \sim R(s, a)$ and also that $r(s, a) := \mathbb{E}[r \mid s, a]$ which is a slight abuse of notation.
- A transition operator $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ that associates a distribution over next states to each state-action pair. As with the reward, we'll say that $s' \sim P(s, a)$ to denote a sample drawn from this operator.
- An initial state distribution $\mu \in \Delta(\mathcal{S})$ that describes how the initial state s_0 will be chosen.

A key property is that the dynamics are *Markovian*, meaning that the distribution of the next state s' and reward r depend only on the most recent state s and action a. With these basic objects there are many formulations.

3 Finite horizon, episodic setting

Let us start with the simpler finite horizon setting. In this setting, there is also a *horizon* $H \in \mathbb{N}$ that describes how long an episode lasts. With horizon H an episode produces a *trajectory* $\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \ldots, s_{H-1}, a_{H-1}, r_{H-1})$ in the following manner: $s_0 \sim \mu$, $s_h \sim P(s_{h-1}, a_{h-1})$ and $r_h \sim R(s_h, a_h)$ for each h, and all actions $a_{0:H-1}$ are chosen by the agent. The Markov property simply means that $(s_h, r_h) \perp s_{h-2} \mid (s_{h-1}, a_{h-1})$ so the entire past is summarized in the current state. Note however that we can choose actions in a non-Markovian manner.

3.1 Policies, Value functions, and Objective

To set up the objective, we must introduce the concept of a *policy*. In general, a policy is a decision making strategy that makes a (possibly randomized) decision based on the current trajectory, that is $\pi : \mathcal{H} \to \Delta(\mathcal{A})$ where $\mathcal{H}$ is the set of all partial trajectories. A non-stationary *Markov* policy instead compresses the history to the current state and time-step only, that is $\pi : \mathcal{S} \times [H] \to \Delta(\mathcal{A})$. For such policies, we use the notation $\pi_h(s) \in \Delta(\mathcal{A})$ to denote the action distribution on state s and time step h. If the policy is deterministic, then the range is just $\mathcal{A}$.

Objective. Every policy π has a *value*, which is the total reward we expect to accumulate if we deploy policy π. This is defined as

$$J(\pi) := \mathbb{E}\left[\sum_{h=0}^{H-1} r_h \mid s_0 \sim \mu, a_{0:H-1} \sim \pi\right]$$

The objective in an MDP is to find a policy π that maximizes $J(\pi)$.

Value functions. For a Markov policy π, we can consider a more-refined notion: how much reward would we get if we started in state s and time h and executed policy π until the end of the episode? This defines the *value function* or the *state-value function*

$$V_h^\pi(s) := \mathbb{E}\left[\sum_{h'=h}^{H-1} r_{h'} \mid s_h = s, a_{h:H-1} \sim \pi\right].$$

The *state-action value function* or just *action-value function* is similar. It describes how much reward we would get if we started in state s at time h, took action a first, then followed π for the subsequent steps. This is defined as

$$Q_h^\pi(s, a) := \mathbb{E}\left[\sum_{h'=h}^{H-1} r_{h'} \mid s_h = s, a_h = a, a_{h+1:H-1} \sim \pi\right].$$

It is worth making a couple of observations. First, we can write $V_h^\pi(s) = \mathbb{E}_{a_h \sim \pi_h(s)}[Q_h^\pi(s, a_h)]$ to relate the two types of value functions. A related and more fundamental property is that these functions satisfy a recursive relationship, known as Bellman equations (or Bellman equations for policy evaluation):

$$\begin{aligned} V_h^\pi(s) &= \mathbb{E}\left[r_h + V_{h+1}^\pi(s_{h+1}) \mid s_h = s, a_h \sim \pi\right] \\ Q_h^\pi(s, a) &= \mathbb{E}\left[r_h + Q_{h+1}^\pi(s_{h+1}, a_{h+1}) \mid s_h = s, a_h = a, a_{h+1} \sim \pi\right] \end{aligned} \tag{1}$$

Intuitively, the reward we expect to get by following π from (s, h) is the immediate reward r_h plus the reward we expect to get by following π from the next state and the next time step.

Next, if π is Markov, then $J(\pi) = \mathbb{E}_{s_0 \sim \mu}[V_0^\pi(s_0)]$ which relates the MDP objective to the value functions. Finally, for non-Markov policies, we cannot really define these functions since the policy's actions may depend on information from the past that is not available. However, we will next see that the optimal policy is Markovian, which justifies our effort in setting up these definitions.

3.2 Optimality and Bellman equations

A fundamental result in the theory of Markov Decision processes is that the optimal policy is Markovian and the optimal value functions satisfy an elegant recursive definition. This is captured in the following theorem.

Theorem 1. *Define the value function* $(V_0^\star, \dots, V_H^\star)$ *recursively as*

$$V_H^\star \equiv 0, \qquad \forall (s,h) : V_h^\star(s) = \max_a r(s,a) + \mathbb{E}_{s' \sim P(s,a)} \left[V_{h+1}^\star(s') \right]. \tag{2}$$

Then $\sup_\pi J(\pi) = \mathbb{E}_{s_0 \sim \mu}[V_0^\star(s_0)]$ *where the supremum is taken over all, possibly non-Markovian, policies. Additionally, define the policy* $\pi^\star := (\pi_0^\star, \dots, \pi_{H-1}^\star)$ *as*

$$\forall (s,h) : \pi_h^\star(s) = \operatorname*{argmax}_a r(s,a) + \mathbb{E}_{s' \sim P(s,a)} \left[V_{h+1}^\star(s') \right]$$

Then $\pi^\star$ *achieves value* $V^\star$ *for all* (s,h) *pairs and hence* $\pi^\star$ *is an optimal policy.*

The two important aspects of this theorem are: (a) we need not consider non-Markovian policies since the optimal policy is Markov, (b) there is a simple recursive formula for computing the optimal value function and the optimal policy can be concisely described in terms of this value function.

An analogous results holds for the action-value functions if we define them as

$$Q_h^\star(s,a) = r(s,a) + \mathbb{E}_{s' \sim P(s,a)} \left[\max_{a'} Q_{h+1}^\star(s', a') \right]. \tag{3}$$

Then we can simply define the optimal policy as $\pi_h^\star : s \mapsto \operatorname{argmax}_a Q_h^\star(s,a)$. Both (2) and (3) are referred to as Bellman optimality equations. Note that these equations show us how to address credit assignment in some sense. Indeed it could be that $r(s,a) \ll Q_h^\star(s,a)$, since taking action a in state s leads to some very favorable conditions later in the episode. So defining $\pi_h^\star$ to maximize $Q_h^\star$ leads to "non-myopic" behavior which is essential for decision making with a long horizon.

Proof. The proof is based on the dynamic programming principle or an induction argument. The key fact is that $\pi^\star$ is optimal *pointwise*, meaning from every state-time pair. This means that we can ignore how we arrive at a distribution over states (i.e., the past) when optimizing for the future.

As the base case, consider time step H. There are no further actions and no further rewards, so all policies accumulate 0 reward from here on. This means that $V_H^\star = 0$ correctly describes the optimal reward achievable at the H^{th} time step.

For the induction, assume that $V_{h+1}^\star(s)$ satisfies the following two properties:

- The Markovian policy $(\pi_{h+1}^\star, \dots, \pi_{H-1}^\star)$ achieves value $V_{h+1}^\star$ for each $s \in \mathcal{S}$ at time $h+1$.
- For all possibly non-Markovian policies $\tilde{\pi}$ we have $\mathbb{E}\left[\sum_{h'=h+1}^{H-1} r_{h'} \mid \tilde{\pi}\right] \le \mathbb{E}\left[V_{h+1}^\star(s_{h+1}) \mid \tilde{\pi}\right]$.

(Note that these two properties hold for $V_H^\star$ trivially.) The second property allows us to study non-Markovian policies, since we can upper bound their future value via our function $V_{h+1}^\star$.

We establish these two properties for the value function $V_h^\star$ defined via (2). With $\pi_h^\star(s) = \operatorname{argmax}_a r(s,a) + \mathbb{E}_{s' \sim P(s,a)} \left[V_{h+1}^\star(s') \right]$, the first property holds simply by the definition and the first inductive hypothesis.

For the second property consider some possibly non-Markovian policy $\tilde{\pi}$. We have

$$\begin{aligned}
\mathbb{E}\left[\sum_{h'=h}^{H-1} r_{h'} \mid \tilde{\pi}\right] &= \mathbb{E}\left[r_h + \sum_{h'=h+1}^{H-1} r_{h'} \mid \tilde{\pi}\right] \\
&\le \mathbb{E}\left[r_h + V_{h+1}^\star(s_{h+1}) \mid \tilde{\pi}\right] && \text{(Inductive hypothesis)} \\
&= \mathbb{E}\left[\mathbb{E}\left[r(s,a) + V_{h+1}^\star(s_{h+1}) \mid s_h = s, a_h = a\right] \mid \tilde{\pi}\right] && \text{(Iterated expectation)} \\
&= \mathbb{E}\left[\mathbb{E}\left[r(s,a) + \mathbb{E}_{s' \sim P(s,a)}\left[V_{h+1}^\star(s')\right] \mid s_h = s, a_h = a\right] \mid \tilde{\pi}\right] && \text{(Markov property of } s_{h+1}\text{)} \\
&\le \mathbb{E}\left[\mathbb{E}\left[\max_{a \in \mathcal{A}} r(s,a) + \mathbb{E}_{s' \sim P(s,a)}\left[V_{h+1}^\star(s')\right] \mid s_h = s\right] \mid \tilde{\pi}\right] \\
&= \mathbb{E}\left[V_h^\star(s_h) \mid \tilde{\pi}\right] && \text{(Definition of } V_h^\star\text{)}
\end{aligned}$$

Concluding the induction, we have value functions $(V_0^\star, \ldots, V_H^\star)$ and a policy $(\pi_0^\star, \ldots, \pi_{H-1}^\star)$ that achieves the value function at every (s, h) pair. Additionally, for any $\tilde{\pi}$:

$$J(\tilde{\pi}) = \mathbb{E}\left[\sum_{h=0}^{H-1} r_h \mid \tilde{\pi}\right] \leq \mathbb{E}\left[V_0^\star(s_0) \mid \tilde{\pi}\right] = J(\pi^\star),$$

since $s_0 \sim \mu$ does not depend on the choice of policy. This proves the theorem. □

3.3 Planning algorithms for the episodic setting

Theorem 1 directly motivates one strategy for computing the optimal value function and policy. This is known as *value iteration*. The algorithm is to simply apply (2) or (3) from time step H down to time step 0. Then we can directly obtain the optimal policy from the value functions we compute.

Another algorithm is called *policy iteration*. Starting with any non-stationary Markov policy $\pi^{(0)}$, in the t^{th} iteration we update via

1. Compute $Q^{\pi^{(t-1)}}$ via (1).
2. Update $\pi^{(t)}$ to be the greedy policy with respect to $Q^{\pi^{(t-1)}}$ that is $\pi_h^{(t)}(s) := \operatorname{argmax}_a Q_h^{\pi^{(t-1)}}(s, a)$.

It can be easily seen that this algorithm converges in H iterations. Indeed, observe that even though $\pi^{(0)}$ is arbitrary, we have that $\pi_{H-1}^{(1)} = \pi_{H-1}^\star$, since Q_{H-1}^π actually does not depend on π and is equal to $Q_{H-1}^\star$.

4 Infinite horizon discounted setting

Let us shift gears and consider a different formulation of reinforcement learning in an MDP. This is referred to as the discounted setting, or infinite horizon discounted setting. Here, instead of a horizon H, we have a discount factor $\gamma \in (0, 1)$ that captures how much we prefer immediate rewards over future rewards. Here trajectories are infinitely long $\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \ldots)$ but follow the same probabilistic model as before. For any policy π we can define the objective and value function as

$$J(\pi) := \mathbb{E}\left[\sum_{h=0}^{\infty} \gamma^h r_h \mid \pi\right], \qquad V^\pi(s) = \mathbb{E}\left[\sum_{h=0}^{\infty} \gamma^h r_h \mid s_0 = s, \pi\right] \qquad Q^\pi(s, a) = \mathbb{E}\left[\sum_{h=0}^{\infty} \gamma^h r_h \mid s_0 = s, a_0 = a, \pi\right]$$

(Note that we are now defining value functions for all policies, including non-Markovian ones.)

Roughly speaking all of the results for the episodic setting carry over here, with appropriate modifications. For example the optimal policy is Markov and, in fact, stationary (meaning it does not depend on the time step) and the optimal value function is the unique solution to the following fixed point equation

$$V^\star(s) = \max_a r(s, a) + \gamma \mathbb{E}_{s' \sim P(s,a)}\left[V^\star(s')\right] \tag{4}$$

This should be viewed as the infinite horizon version of (2), but note that we are not time-indexing $V^\star$ and it appears on both sides of the equation. That said, an analogous result to Theorem 1 holds in this setting, although it is a bit more subtle.

4.1 Planning algorithms

Both value iteration and policy iteration can be modified for the discounted setting. To describe these algorithms, and for use in subsequent lectures, it is helpful to define the *Bellman operator* $\mathcal{T}$ as

$$\mathcal{T} f : (s, a) \mapsto r(s, a) + \gamma \mathbb{E}_{s' \sim P(s,a)}\left[\max_a f(s', a')\right].$$

Note that this operator takes Q-functions as input and produces a Q-function as output. Then the Bellman optimality equation (the analog of (3)) is simply that $Q^\star = \mathcal{T} Q^\star$.

Value iteration. Motivated by the fixed point relationship for $Q^\star$, value iteration simply iterates the operation $f^{(t+1)} \leftarrow \mathcal{T} f^{(t)}$ starting from an arbitrary initial Q-function $f^{(0)}$. The key to the convergence of this algorithm is a certain *contraction property* for the bellman operator

Lemma 2 (Contraction)**.** *For any two Q-functions f, f', we have $\|\mathcal{T} f - \mathcal{T} f'\|_\infty \leq \gamma \|f - f'\|_\infty$.*

Proof. Considering any (s, a) pair we have

$$\begin{aligned}
|(\mathcal{T} f)_{s,a} - (\mathcal{T} f')_{s,a}| &= |\gamma \left(\mathbb{E}_{s' \sim P(s,a)} \max_{a'} f(s', a') - \max_{a''} f'(s', a'') \right)| \\
&\leq \gamma \mathbb{E}_{s' \sim P(s,a)} \max_{a'} |f(s', a') - f'(s', a')| \\
&\leq \gamma \|f - f'\|_\infty
\end{aligned}$$

The first inequality follows since, if $\max_{a'} f(s', a') \geq \max_{a''} f(s', a'')$ then:

$$f(s', a') - \max_{a''} f'(s', a'') \leq f(s', a') - f'(s', a') \leq \max_a |f(s', a) - f'(s', a)|,$$

with a similar calculation for the other case. □

This will immediately give us a geometric convergence to $Q^\star$ by iterating the bellman operator, but we need to translate error in $Q^\star$ to policy performance. This is given in the next lemma.

The notation here is a bit confusing. We will use f to denote any function of the type $\mathcal{S} \times \mathcal{A} \to \mathbb{R}$, which as the same type as the Q-functions. For a policy π, Q^π is the true action value function for π in the MDP. The confusing part is that if $\pi_f : s \mapsto \operatorname{argmax}_a f(s, a)$ is the greedy policy with respect to function f, then Q^{π_f} will in general be distinct from f. This is why we are using f to denote general functions with this type.

Lemma 3 (Policy error)**.** *For any function f we have $J(\pi^\star) \leq J(\pi_f) + \frac{2}{1-\gamma} \|f - Q^\star\|_\infty$.*

Proof. Consider state s and let $a = \pi_f(s) = \operatorname{argmax}_{a'} f(s, a')$. Then

$$\begin{aligned}
V^\star(s) - V^{\pi_f}(s) &= Q^\star(s, \pi^\star(s)) - Q^\star(s, a) + Q^\star(s, a) - Q^{\pi_f}(s, a) \\
&\leq Q^\star(s, \pi^\star(s)) - f(s, \pi^\star(s)) + f(s, a) - Q^\star(s, a) + Q^\star(s, a) - Q^{\pi_f}(s, a) \\
&\leq 2\|Q^\star - f\|_\infty + \gamma \mathbb{E}_{s' \sim P(s,a)} \left[V^\star(s') - V^{\pi_f}(s') \right] \\
&\leq 2\|Q^\star - f\|_\infty + \gamma \|V^\star - V^{\pi_f}\|_\infty
\end{aligned}$$

Re-arranging this inequality actually proves a stronger statement, namely that $V^\star$ and V^{π_f} are close, which implies that $J(\pi^\star)$ and $J(\pi_f)$ are close. □

Taking these two together, we immediately have an iteration complexity bound for value iteration.

Theorem 4. *Set $f^{(0)} = 0$, run value iteration for T iterations and define $\hat{\pi} = \pi_{f^{(T)}}$. Then*

$$J(\pi^\star) - J(\hat{\pi}) \leq \frac{2\gamma^T \|Q^\star\|_\infty}{1 - \gamma}.$$