37 Markov Decision Processes

Bandit environments are a sensible model for many simple problems, but they do not model more complex environments where actions have long-term consequences. A brewing company needs to plan ahead when ordering ingredients and the decisions made today affect their position to brew the right amount of beer in the future. A student learning mathematics benefits not only from the immediate reward of learning an interesting topic, but also from their improved job prospects.

A **Markov decision process** is a simple way to incorporate long-term planning into the bandit framework. Like in bandits, the learner chooses actions and receives rewards. But they also observe a **state** and the rewards for different actions depend on the state. Furthermore, the actions chosen affect which state will be observed next.

37.1 Problem setup

A Markov decision process (MDP) is defined by a tuple $M = (\mathcal{S}, \mathcal{A}, P, r)$. The first two items $\mathcal{S}$ and $\mathcal{A}$ are sets called the **state space** and **action space** respectively and $\mathrm{S} = |\mathcal{S}|$ and $\mathrm{A} = |\mathcal{A}|$ are their sizes, which may be infinite. An MDP is finite if $\mathrm{S}, \mathrm{A} < \infty$. The quantity $P = (P_a : s \in \mathcal{S}, a \in \mathcal{A})$ is called the **transition function** with $P_a : \mathcal{S} \times \mathcal{S} \to [0,1]$ so that $P_a(s, s')$ is the probability that the learner transitions from state s to s' when taking action a. The last element in the tuple is the reward $r = (r_a : a \in \mathcal{A})$, which is a collection of **reward functions** with $r_a : \mathcal{S} \to [0,1]$. When the learner takes action a in state s it receives a deterministic reward of $r_a(s)$. The transition and reward functions are often represented by vectors or matrices. When the state space is finite we may assume without loss of generality that $\mathcal{S} = [\mathrm{S}]$. We write $P_a(s) \in [0,1]^{\mathrm{S}}$ as the probability vector with s'th coordinate given by $P_a(s, s')$. In the same way we let $P_a \in [0,1]^{\mathrm{S} \times \mathrm{S}}$ be the right stochastic matrix with $(P_a)_{s,s'} = P_a(s, s')$. Finally, we view r_a as a vector in $[0,1]^{\mathrm{S}}$ in the natural way.

While we use the same action-set in different states $s, s' \in \mathcal{S}$, this does not mean that $P_a(s)$ or $r_a(s)$ has any relationship to $P_a(s')$ or $r_a(s')$. By learning about P_a at s the learner does not gain information about P_a at $s' \neq s$. In this

sense the notation is a bit misleading and perhaps it would be better to use an entirely different set of actions for each state. This could be done with no changes to any of the results we present. And while we are at it, of course one could also allow the number of actions to vary over the state space. The only justification for assuming that the same set of actions is available in all states is that it simplifies the presentation.

The interaction protocol is very similar to bandits. Before the game starts, the initial state S_1 is sampled from a distribution $\mu \in \mathcal{P}(\mathcal{S})$. In each round t the learner observes the state $S_t \in \mathcal{S}$, chooses an action $A_t \in \mathcal{A}$ and receives reward $r_{A_t}(S_t)$. The environment then samples S_{t+1} from the probability vector $P_{A_t}(S_t)$ and then the next round begins (Fig. 37.1).

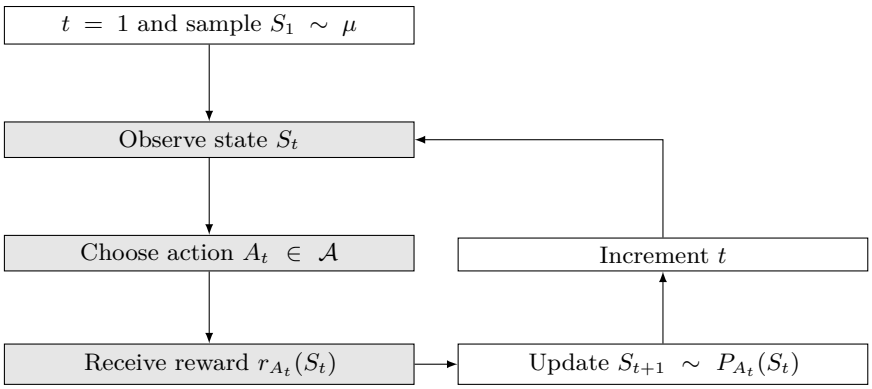


Figure 37.1 Interaction protocol for Markov decision processes

Histories and policies

Before considering the learning problem, we start by explaining how to act in a known MDP. Because there is no learning going on we call our protagonist the 'agent' rather than 'learner'. In a stochastic bandit the optimal policy given knowledge of the bandit is to choose the action with the largest expected reward in every round, which maximizes the expected cumulative reward. In a Markov decision process even the definition of optimality is much less clear.

The **history** $H_t = (S_1, A_1, \ldots, S_{t-1}, A_{t-1}, S_t)$ in round t contains the information available before the action for the round is to be chosen. Note that state S_t is included in H_t. The actions are also included because the agent may randomize. For simplicity the rewards are omitted because the all-knowing agent can just recompute them if needed from the state-action pairs.

A **policy** is a (possibly randomized) map from the set of possible histories to actions. Simple policies include **memoryless policies**, which choose actions based on only the current state, possibly in a randomized manner. The set

of such policies is denoted by Π_{M} and its elements are identified with maps $\pi : \mathcal{S} \times \mathcal{A} \times [0,1]$ with $\sum_{a \in \mathcal{A}} \pi(s,a) = 1$ for any $s \in \mathcal{S}$ so that $\pi(s,a)$ is interpreted as the probability that policy π takes action a in state s.

A memoryless policy that does not randomize is called a **memoryless deterministic policy**. To reduce clutter such policies are written as $\mathcal{S} \to \mathcal{A}$ maps and the set of all such policies is denoted by Π_{DM}. A policy is called a **Markov policy** if the actions are randomized and depend only on the round index and the previous state. These policies are represented by fixed sequences of memoryless policies. Under a Markov policy the sequence of states $(S_1, S_2, \dots)$ evolve as a Markov chain (see Section 3.2). If the Markov policy is memoryless, this chain is homogeneous.

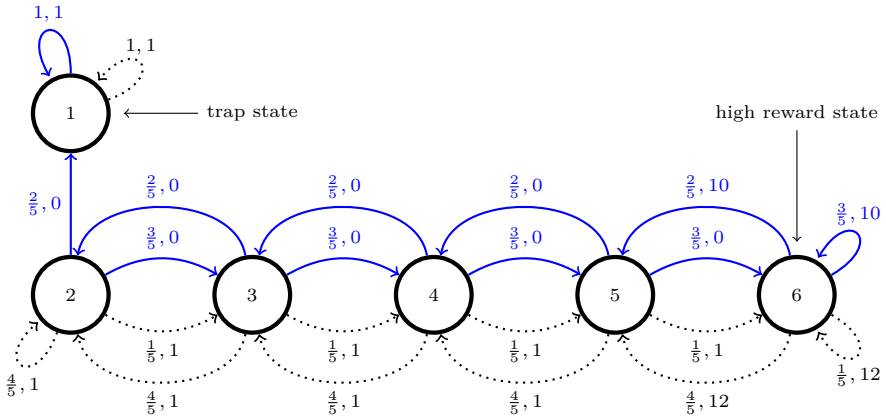


Figure 37.2 A Markov decision process with six states and two actions represented by solid and dashed arrows, respectively. The numbers next to each arrow represent the probability of transition and reward for the action respectively. For example, taking the solid action in state three results in a reward of zero and the probability of moving to state four is 3/5 and the probability of moving to state three is 2/5. For human interpretability only, the actions are given consistent meaning across the states (blue/solid actions 'increment' the state index, black/dashed actions decrement it). In reality there is no sense of similarity between states or actions build into the MDP formalism.

Probability spaces

It will be convenient to allow infinitely long interactions between the learner and environment. In line with Fig. 37.1, when the agent or learner follows a policy π in MDP $M = (\mathcal{S}, \mathcal{A}, P, r)$ such a never ending interaction gives rise to a random process $(S_1, A_1, S_2, A_2, \dots)$ so that for any $s, s' \in \mathcal{S}$, $a \in \mathcal{A}$ and $t \geq 1$,

(a) $\mathbb{P}(S_1 = s) = \mu(s)$;
(b) $\mathbb{P}(S_{t+1} = s' \mid H_t, A_t) = P_{A_t}(S_t, s')$;
(c) $\mathbb{P}(A_t = a \mid H_t) = \pi(H_t, a)$,

where $\mu \in \mathcal{P}(\mathcal{S})$ is the initial state distribution and $\pi(H_t, a)$ stands for the probability of the agent selecting action a in the tth round of interaction when the history is $H_t = (S_1, A_1, \dots, S_{t-1}, A_{t-1}, S_t)$. Meticulous readers may wonder whether there exists a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ holding the infinite sequence of random variables $(S_1, A_1, S_2, A_2, \dots)$ that make (a)–(c) hold true regardless the choice of M, π and μ. Our readers should find it pleasing that the Ionescu Tulcea theorem (Theorem 3.3) furnishes us with a positive answer (Exercise 37.1). Item (b) above is known as the **Markov property**. Of course the measure $\mathbb{P}$ depends on both the policy and Markov decision process and the initial distribution. For most of the chapter these quantities will be fixed and the dependence is omitted from the notation. In the few places where disambiguation is necessary we provide additional notation. In addition to this, to minimize clutter, we allow ourselves to write $\mathbb{P}(\cdot \mid S_1 = s)$, which just means the probability distribution that results from the interconnection of π and M, while replacing μ with an alternative initial state distribution that is a Dirac at s.

Traps and the diameter of a Markov decision process

A significant complication in MDPs is the potential for traps. A trap is a subset of the state space that there is no escape from. For example, the MDP in Fig. 37.2 has a trap state. If being in the trap has a suboptimal yield in terms of the reward, the learner should avoid the trap, but since the learner can only discover that an action leads to a trap by trying that action and since, by definition, there are no second chances (the environment-agent interaction is continuous and is uninterrupted, with no option to somehow reset the environment), the problem of learning while competing with a fully informed agent is hopeless (Exercise 37.27).

To avoid this complication we restrict our attention to MDPs with no traps. A Markov decision process is called **strongly connected** or **communicating** if for any pair of states $s, s' \in \mathcal{S}$ there exists a policy such that when starting from s there is a positive probability of reaching s' some time in the future while follow the policy. One can also define a real-valued measure of the connectedness of an MDP called the **diameter**. MDPs with smaller diameter are usually easier to learn because a policy can recover from mistakes more quickly.

DEFINITION 37.1 Define stopping time $\tau_s = \min\{ t \geq 1 : S_t = s\}$. The **diameter** of M is

$$D(M) = \max_{s \neq s'} \min_{\pi : \mathcal{S} \to \mathcal{A}} \mathbb{E}^{\pi} \left[\tau_{s'} \mid S_1 = s \right] ,$$

where the expectation is taken with respect to the probability measure $\mathbb{P}^{\pi}$ that results from the interconnection of π and M.

A number of observations are in order about this definition. First, the order of the maximum and minimum means that for any pair of states a different policy may be used. Second, travel times are always minimized by deterministic memoryless policies so the restriction to these policies in the minimum is

inessential (Exercise 37.3). Finally, the definition only considers distinct states. We also note that when the number of states is finite it holds that $D(M) < \infty$ if and only if M is strongly connected (Exercise 37.4). The diameter of an MDP with S states and A actions cannot be smaller than $\log_{\mathrm{A}}(\mathrm{S}) - 3$ (Exercise 37.5).

For the remainder of this chapter, unless otherwise specified, all MDPs are assumed to be strongly connected.

37.2 Optimal policies and the Bellman optimality equation

We now define the notion of an optimal policy and outline the proof that there exists a deterministic memoryless optimal policy. Along the way we define what is called the Bellman optimality equation. Methods that solve this equation are the basis for finding optimal policies in an efficient manner and also play a significant role in learning algorithms.

Throughout we fix a strongly connected Markov decision process M. The **gain** of a policy π is the long-term average reward expected from using that policy when starting in state s.

$$\rho_s^\pi = \lim_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} \mathbb{E}^\pi[r_{A_t}(S_t) \mid S_1 = s] \,,$$

where $\mathbb{E}^\pi$ denotes the expectation on the interaction sequence when policy π interacts with MDP M. In general the limit need not exist, so we also introduce

$$\bar{\rho}_s^\pi = \limsup_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} \mathbb{E}^\pi[r_{A_t}(S_t) \mid S_1 = s] \,,$$

which exists for any policy. Of course, whenever ρ_s^π exists we have $\rho_s^\pi = \bar{\rho}_s^\pi$. The **optimal gain** is a real value

$$\rho^* = \max_{s\in\mathcal{S}} \sup_\pi \bar{\rho}_s^\pi \,,$$

where the supremum is taken over all policies. A π policy is an **optimal policy** if $\rho^\pi = \rho^* \mathbf{1}$. For strongly connected MDPs an optimal policy is guaranteed to exist. This is far from trivial, however, and we will spend the next little while outlining the proof. When the MDP is not strongly connected a new notion of optimality is required since in general there is no guarantee that the optimal gain is the same regardless of the starting state.

Before continuing we need some new notation. For memoryless policy π define

$$P_\pi(s,s') = \sum_{a\in\mathcal{A}} \pi(s,a) P_a(s,s') \qquad \text{and} \qquad r_\pi(s) = \sum_{a\in\mathcal{A}} \pi(s,a) r_a(s) \,. \tag{37.1}$$

We view P_π as an $\mathrm{S}\times\mathrm{S}$ **transition matrix** and r_π as a vector in $\mathbb{R}^{\mathrm{S}}$. With this

notation P_π is the transition matrix of the homogeneous Markov chain $S_1, S_2, \ldots$ when $A_t \sim \pi(S_t, \cdot)$. The gain of memoryless policy π satisfies

$$\rho^\pi = \lim_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} P_\pi^{t-1} r_\pi = P_\pi^* r_\pi \,, \tag{37.2}$$

where $P_\pi^* = \lim_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} P_\pi^{t-1}$ is called the **stationary transition matrix**, the existence of which you will prove in Exercise 37.8. For each $k \in \mathbb{N}$ define

$$v_\pi^{(k)} = \sum_{t=1}^{k} P_\pi^{t-1}(r_\pi - \rho^\pi) \,.$$

For $s \in \mathcal{S}$, $v_\pi^{(k)}(s)$ gives the total expected excess reward collected by π when the process starts at state s and lasts for k time steps. The **(differential) value function** of a policy is a function $v_\pi : \mathcal{S} \to \mathbb{R}$ defined as the **Cesàro sum** of the sequence $(P_\pi^t(r_\pi - \rho^\pi))_{t\geq 0}$,

$$v_\pi = \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} v_\pi^{(k)} = ((I - P_\pi + P_\pi^*)^{-1} - P_\pi^*) r_\pi \,. \tag{37.3}$$

Note, the second equality above is nontrivial (Exercise 37.8). The definition implies that $v_\pi(s) - v_\pi(s')$ is the 'average' long-term advantage of starting in state s relative to starting to state s' when following policy π. These quantities are only defined for memoryless policies where they are also guaranteed to exist (Exercise 37.8). Observe the definition of P_π^* implies that $P_\pi^* P_\pi = P_\pi^*$, which in turn implies that $P_\pi^* v_\pi = 0$. Combining this with Eq. (37.2) and Eq. (37.3) shows that for any memoryless policy π,

$$\rho^\pi + v_\pi = r_\pi + P_\pi v_\pi \,. \tag{37.4}$$

A **value function** is a function $v : \mathcal{S} \to \mathbb{R}$ and its **span** is given by

$$\operatorname{span}(v) = \max_{s\in\mathcal{S}} v(s) - \min_{s\in\mathcal{S}} v(s) \,.$$

As with other quantities, value functions are associated with vectors in $\mathbb{R}^{\mathcal{S}}$. A **greedy policy** with respect to value function v is a deterministic memoryless policy π_v given by

$$\pi_v(s) = \operatorname{argmax}_{a\in\mathcal{A}} r_a(s) + \langle P_a(s), v \rangle \,.$$

There may be many policies that are greedy with respect to some value function v due to ties in the maximum. Usually the ties do not matter, but for consistency and for the sake of simplifying matters, we assume that ties are broken in a systematic fashion. In particular, this makes π_v well-defined for any value function.

One way to find the optimal policy is as the greedy policy with respect to a value function that satisfies the **Bellman optimality equation**, which is

$$\rho + v(s) = \max_{a\in\mathcal{A}} (r_a(s) + \langle P_a(s), v \rangle) \qquad \text{for all } s \in \mathcal{S} \,. \tag{37.5}$$

This is a system of S nonlinear equations with unknowns $\rho \in \mathbb{R}$ and $v \in \mathbb{R}^{\mathrm{S}}$. The reader will notice that if $v : \mathcal{S} \to \mathbb{R}$ is a solution to Eq. (37.5), then so is $v + c\mathbf{1}$ for any constant $c \in \mathbb{R}$ and hence the Bellman optimality equation lacks unique solutions. However, it is *not* true that the optimal value function is unique up to translation, even when M is strongly connected (Exercise 37.12). The v-part of a solution pair (ρ, v) of Eq. (37.5) is called an **optimal (differential) value function**.

THEOREM 37.1 *The following hold:*

(a) There exists a pair (ρ, v) that satisfies the Bellman optimality equation.
(b) If (ρ, v) satisfies the Bellman optimality equation, then $\rho = \rho^*$ and π_v is optimal.
(c) There exists a deterministic memoryless optimal policy.

Proof sketch The proof of Part (a) is too long to include here, but we guide you through it in Exercise 37.11. For Part (b) let (ρ, v) satisfy the Bellman equation and $\pi^* = \pi_v$ be the greedy policy with respect to v. Then, by Eq. (37.2),

$$\rho^{\pi^*} = \lim_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} P_{\pi^*}^{t-1} r_{\pi^*} = \lim_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} P_{\pi^*}^{t-1} \left(\rho\mathbf{1} + v - P_{\pi^*} v\right) = \rho\mathbf{1}\,.$$

Next let π be an arbitrary Markov policy. We show that $\bar{\rho}^{\pi} \le \rho\mathbf{1}$. The result is then completed using the result of Exercise 37.2, where you will prove that for any policy π there exists a Markov policy with the same expected rewards. Denote by π_t the memoryless policy used at time $t = 1, 2, \ldots$ when following the Markov policy π and for $t \ge 1$ let $P_{\pi}^{(t)} = P_{\pi_1} \ldots P_{\pi_t}$, while for $t = 0$ let $P_{\pi}^{(0)} = I$. Thus, $P_{\pi}^{(t)}(s, s')$ is the probability of ending up in state s' while following π from state s for t time steps. It follows that $\bar{\rho}^{\pi} = \limsup_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} P_{\pi}^{(t-1)} r_{\pi_t}$. Fix $t \ge 1$. Using the fact that π^* is the greedy policy with respect to v gives

$$\begin{aligned}
P_{\pi}^{(t-1)} r_{\pi_t} &= P_{\pi}^{(t-1)} (r_{\pi_t} + P_{\pi_t} v - P_{\pi_t} v) \\
&\le P_{\pi}^{(t-1)} (r_{\pi^*} + P_{\pi^*} v - P_{\pi_t} v) \\
&= P_{\pi}^{(t-1)} (\rho\mathbf{1} + v - P_{\pi_t} v) \\
&= \rho\mathbf{1} + P_{\pi}^{(t-1)} v - P_{\pi}^{(t)} v\,.
\end{aligned}$$

Taking the average of both sides over $t \in [n]$ and then taking the limit shows that $\bar{\rho}^{\pi} \le \rho\mathbf{1}$, finishing the proof. Part (c) follows immediately from the first two parts. □

The theorem shows that there exist solutions to the Bellman optimality equation and that the greedy policy with respect to the resulting value function is an optimal policy. We need one more result about solutions to the Bellman optimality equation, the proof of which you will provide in Exercise 37.13.

LEMMA 37.1 *Suppose that (ρ, v) satisfies the Bellman optimality equation. Then* $\operatorname{span}(v) \le D(M)$.

The operator $T : \mathbb{R}^{\mathrm{S}} \to \mathbb{R}^{\mathrm{S}}$ defined by $(Tv)(s) = \max_{a \in \mathcal{A}} r_a(s) + \langle P_a(s), v \rangle$ is called the **Bellman operator**. The Bellman operator allows one to write the Bellman optimality equation in the short form $\rho \mathbf{1} + v = Tv$. Furthermore, if we let $v_n^*(s)$ denote the maximum achievable expected cumulative reward over n rounds when starting in state s, it is not hard to check that $v_n^* = T^n \mathbf{0}$. Clearly $v_n^* = T v_{n-1}^*$ for any $n \geq 1$. Let $v_0 \in \mathbb{R}^{\mathrm{S}}$ and $v_{k+1} = T v_k$. The algorithm that computes the sequence $(v_k)_k$ is called **value iteration**. Under certain conditions the greedy policy with respect to v_k converges to an optimal policy as k tends to infinity. For more on this see the notes.

We have not said how to solve the Bellman optimality equation. When the computational cost is important this becomes surprisingly subtle. Readers who are more interested in the learning aspect of the problem can skip the details, which are provided in the next section.

37.3 Finding an optimal policy (†)

There are many ways to find an optimal policy, including value iteration, policy iteration and enumeration. These ideas are briefly discussed in the notes. Here we describe an approach based on linear programming. As in the previous section we fix a strongly connected finite Markov decision process. Consider the following (constrained) linear optimization problem:

$$\begin{aligned} &\underset{\rho \in \mathbb{R}, v \in \mathbb{R}^{\mathrm{S}}}{\text{minimize}} \quad \rho \\ &\text{subject to} \quad \rho + v(s) \geq r_a(s) + \langle P_a(s), v \rangle \text{ for all } s, a\,. \end{aligned} \tag{37.6}$$

Recall that a constrained optimization problem is said to be **feasible** if the constraint set (the set of values that satisfy the constraints) is non-empty.

THEOREM 37.2 *The optimization problem in Eq. (37.6) is feasible and if (ρ, v) is a solution, then $\rho = \rho^*$ is the optimal gain.*

Proof Let v^* be such that (ρ^*, v^*) satisfies the Bellman optimality equation, which means that

$$\rho^* + v^*(s) = \max_{a \in \mathcal{A}} r_a(s) + \langle P_a(s), v^* \rangle \qquad \text{for all state/action pairs } (s, a)\,.$$

By Theorem 37.1, such a v^* exists. Hence the pair (ρ^*, v^*) satisfies the constraints in Eq. (37.6) and witnesses feasibility. Next let (ρ, v) be a solution of Eq. (37.6). Since (ρ^*, v^*) satisfy the constraints, $\rho \leq \rho^*$ is immediate. It remains to prove that $\rho \geq \rho^*$. Let $\pi = \pi_v$ be the greedy policy with respect to v and π^* be greedy with respect to v^*. By Theorem 37.1, $\rho^* \mathbf{1} = \rho^{\pi^*}$. Furthermore,

$$P_{\pi^*}^t r_{\pi^*} \leq P_{\pi^*}^t (r_\pi + P_\pi v - P_{\pi^*} v) \leq P_{\pi^*}^t (\rho \mathbf{1} + v - P_{\pi^*} v) = \rho \mathbf{1} + P_{\pi^*}^t v - P_{\pi^*}^{t+1} v\,.$$

Summing over t shows that $\rho^* \mathbf{1} = \lim_{n\to\infty} \frac{1}{n} \sum_{t=0}^{n-1} P_{\pi^*}^t r_{\pi^*} \le \rho \mathbf{1}$, thus completing the proof. □

We have not claimed that solutions to this linear program satisfy the Bellman optimality equation or that the greedy policy is optimal. Both can fail to be true (Exercise 37.30).There are several ways to fix this deficiency. Perhaps the simplest is to solve the linear program in Eq. (37.6) to find ρ^* and then solve another linear program that fixes the gain while minimizing the value function. Let $\tilde{s} \in \mathcal{S}$ and consider the following linear program:

$$\begin{aligned} \underset{v \in \mathbb{R}^{\mathrm{S}}}{\text{minimize}} \quad & \langle v, \mathbf{1} \rangle \\ \text{subject to} \quad & \rho^* + v(s) \ge r_a(s) + \langle P_a(s), v \rangle \text{ for all } s, a \\ & v(\tilde{s}) = 0 \,. \end{aligned} \tag{37.7}$$

The second constraint is crucial in order for the minimum to exist, since otherwise the value function can be arbitrarily small. The next theorem shows that provided $\tilde{s}$ is chosen appropriately, then the solution of Eq. (37.7) satisfies the Bellman optimality equation.

THEOREM 37.3 *Let v be a solution of Eq. (37.7) and assume there exists an optimal policy π^* such that $P_{\pi^*}^*(s, \tilde{s}) > 0$ for all $s \in \mathcal{S}$. Then (ρ^*, v) satisfies the Bellman optimality equation.*

Proof Let $\varepsilon = v + \rho^* \mathbf{1} - Tv$, which, by the first constraint, satisfies $\varepsilon \ge 0$. Let π^* be an optimal policy satisfying the requirements of the theorem statement and π be the greedy policy with respect to v. Then

$$P_{\pi^*}^t r_{\pi^*} \le P_{\pi^*}^t (r_\pi + P_\pi v - P_{\pi^*} v) = P_{\pi^*}^t (\rho^* \mathbf{1} + v - \varepsilon - P_{\pi^*} v) \,.$$

Hence $\rho^* \mathbf{1} = \rho^{\pi^*} \mathbf{1} \le \rho^* \mathbf{1} - P_{\pi^*}^* \varepsilon$, which means that $P_{\pi^*}^* \varepsilon \le 0$. Since $\varepsilon \ge 0$ and $P_{\pi^*}^*$ is stochastic, hence, $P_{\pi^*}^* f \ge 0$ whenever $f \ge 0$, we get $P_{\pi^*}^* \varepsilon = 0$. Using again that $\varepsilon \ge 0$ we see that $\varepsilon(s) = 0$ for all states s in any recurrence class of π^*. By our assumption on $P_{\pi^*}^*$ we conclude that $\varepsilon(\tilde{s}) = 0$. It follows that $\tilde{v} = v - \varepsilon$ also satisfies the constraints in Eq. (37.7). Since v is a solution to the optimization problem, $\langle \tilde{v}, \mathbf{1} \rangle \ge \langle v, \mathbf{1} \rangle$, implying that $\langle \varepsilon, \mathbf{1} \rangle \le 0$. Since $\varepsilon \ge 0$, we conclude that $\varepsilon = 0$. □

To complete the procedure we need to find a state $\tilde{s}$ that is recurrent under some optimal policy. There is a relatively simple procedure for doing this using the solution to Eq. (37.6), but its analysis depends on the basic theory of duality from linear programming, which is beyond the scope of this text. More details are in Note 11 at the end of the chapter. Instead we observe that one can simply solve Eq. (37.7) for all choices of $\tilde{s}$ and take the first solution that satisfies the Bellman optimality equation.

37.3.1 Efficient computation

The linear programs in Eq. (37.6) and Eq. (37.7) can be solved efficiently under assumptions that will be satisfied in subsequent applications. To explain we need a little theory from linear programming. The general form of a linear program is an optimization problem of the form

$$\begin{aligned} &\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \langle c, x \rangle \\ &\text{subject to} \quad Ax \geq b\,, \end{aligned}$$

where $c \in \mathbb{R}^n$ and $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$ are parameters of the problem. This general problem can be solved in time that depends polynomially on n and m. When m is very large or infinite these algorithms may become impractical, but nevertheless one can often still solve the optimization problem in time polynomial in n only, provided that the constraints satisfy certain structural properties. Let $\mathcal{K} \subset \mathbb{R}^n$ be convex and consider

$$\begin{aligned} &\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \langle c, x \rangle \\ &\text{subject to} \quad x \in \mathcal{K}\,. \end{aligned} \tag{37.8}$$

Algorithms for this problem generally have a slightly different flavor because $\mathcal{K}$ may have no corners. Suppose the following holds:

(a) There exists a known $R > 0$ such that $\mathcal{K} \subset \{x \in \mathbb{R}^n : \|x\| \leq R\}$.

(b) There exist a separation oracle, which we recall from Chapter 27, is a computational procedure to evaluate some function ϕ on $\mathbb{R}^n$ with $\phi(x) = \textsc{True}$ for $x \in \mathcal{K}$ and otherwise $\phi(x) = u$ with $\langle y, u \rangle > \langle x, u \rangle$ for all $y \in \mathcal{K}$ (see Fig. 27.1).

(c) There exists a $\delta > 0$ and $x_0 \in \mathbb{R}^d$ such that $\{x \in \mathbb{R}^n : \|x - x_0\| \leq \delta\} \subset \mathcal{K}$.

Under these circumstances the ellipsoid method accepts as input the size of the bounding sphere R, the separation oracle and an accuracy parameter $\varepsilon > 0$ and returns a point x in time polynomial in n and $\log(R/(\delta\varepsilon))$ such that $x \in \mathcal{K}$ and $\langle c, x \rangle \leq \langle c, x^* \rangle + \varepsilon$ where x^* is the minimizer of Eq. (37.8). The reader can find references to this method at the end of the chapter.

The linear programs in Eq. (37.6) and Eq. (37.7) do not have bounded feasible regions because if v is feasible, then $v + c\mathbf{1}$ is also feasible for any $c \in \mathbb{R}$. For strongly connected MDPs with diameter D, however, Lemma 37.1 allows us to add the constraint that $\|v\|_\infty \leq D$. If the rewards are bounded in $[0, 1]$, then we may also add the constraint that $0 \leq \rho \leq 1$ and then set $R = \sqrt{1 + D^2 \mathrm{S}}$. When the diameter is unknown one may use a doubling procedure. In order to guarantee the feasible region contains a small ball we add some slack to the constraints. Let

$\varepsilon > 0$ and consider the following linear program.

$$\begin{aligned} \underset{\rho\in\mathbb{R}, v\in\mathbb{R}^S}{\text{minimize}} \quad & \rho \\ \text{subject to} \quad & \varepsilon + \rho + v(s) \geq r_a(s) + \langle P_a(s), v\rangle \text{ for all } s, a\,. \\ & v(s) \geq -D \text{ for all } s \\ & v(s) \leq D \text{ for all } s \\ & \rho \leq 1 \text{ for all } s \\ & \rho \geq 0 \text{ for all } s\,. \end{aligned} \tag{37.9}$$

Note that any x in the feasible region of Eq. (37.9) there exists a y that is feasible for Eq. (37.6) with $\|x - y\|_\infty \leq \varepsilon$. Furthermore, the solution to the above linear program is at most ε away from the solution to Eq. (37.6). What we have bought by adding this slack is that now the linear program in Eq. (37.9) satisfies the conditions (a) and (c) above. The final step is to give a condition when a separation oracle exists for the convex set determined by the constraints in the above program. Define convex set

$$\mathcal{K} = \{(\rho, v) \in \mathbb{R}^{d+1} : \varepsilon + v(s) \geq r_a(s) + \langle P_a(s), v\rangle \text{ for all } s, a\}\,. \tag{37.10}$$

Assuming that

$$\text{argmax}_{a\in\mathcal{A}}(r_a(s) + \langle P_a(s), v\rangle) \tag{37.11}$$

can be solved efficiently, then Algorithm 23 provides a separation oracle for $\mathcal{K}$. For the specialized case considered later Eq. (37.11) is trivial to compute efficiently. The actual constraints in Eq. (37.9) consists of $\mathcal{K}$ intersected with a small number of half-spaces. In Exercise 37.31 you will show how to efficiently extend a separation oracle for arbitrary convex set $\mathcal{K}$ to $\bigcap_{i=1}^n H_k \cap \mathcal{K}$ where $(H_k)_{k=1}^n$ are half-spaces. You will show in Exercise 37.14 that approximately solving Eq. (37.7) works in the same way as the above, as well as the correctness of Algorithm 23.

In Theorem 37.1 we assumed an exact solution of the Bellman optimality equation, which may not be possible in practice. Fortunately, approximate solutions to the Bellman optimality equation with approximately greedy policies yield approximately optimal policies. Details are deferred to Exercise 37.15.

37.4 Learning in Markov decision processes

The problem of finding an optimal policy in an unknown Markov decision process is no longer just an optimization problem and the regret is introduced to measure the price of the uncertainty. For simplicity we assume that only the transition matrix is unknown while the reward function is given. This assumption is not

function SEPARATIONORACLE(ρ, v)
  For each $s \in \mathcal{S}$ find $a_s^* \in \operatorname{argmax}_a(r_a(s) + \langle P_a(s), v\rangle)$
  if $\varepsilon + \rho + v(s) \geq r_{a_s^*}(s) + \langle P_{a_s^*}(s), v\rangle$ for all $s \in \mathcal{S}$ **then**
    return TRUE
  else
    Find state s with $\varepsilon + \rho + v(s) < r_{a_s^*}(s) + \langle P_{a_s^*}(s), v\rangle$
    Return $(1, e_s - P_{a_s^*}(s))$
  end if
end function

Algorithm 23: Separation oracle for Eq. (37.6).

especially restrictive as the case where the rewards are also unknown is easily covered using either a reduction or a simple generalization as we explain in the notes. The regret of a policy π is the deficit of rewards suffered relative to the expected average reward of an optimal policy:

$$\hat{R}_n = n\rho^* - \sum_{t=1}^{n} r_{A_t}(S_t)\,.$$

The reader will notice we are comparing the nonrandom $n\rho^*$ to the random sum of rewards received by the learner, which was also true in the study of stochastic bandits. The difference is that ρ^* is an asymptotic quantity while for stochastic bandits the analogous quantity was $n\mu^*$. The definition stills makes sense, however, because for MDPs with finite diameter D the optimal expected value over n rounds is at least $n\rho^* - D$ so the difference is negligible (Exercise 37.16). The main result of this chapter is the following. For the theorem statement let $\mathrm{S}, \mathrm{A}, n$ be fixed positive integers and $\delta \in (0,1)$ a fixed error probability.

THEOREM 37.4 *There exists an efficiently computable policy π that when interacting with any MDP $M = (\mathcal{S}, \mathcal{A}, P, r)$ with S states, A actions, rewards in $[0,1]$ and any initial state distribution satisfies with probability at least $1-\delta$,*

$$\hat{R}_n < CD(M)\mathrm{S}\sqrt{\mathrm{A}n\log(n\mathrm{SA}/\delta)}\,,$$

where C is a universal constant.

In Exercise 37.17 we ask you to use the assumption that the rewards are bounded to find a choice of $\delta \in (0,1)$ such that

$$\mathbb{E}[\hat{R}_n] \leq 1 + CD(M)\mathrm{S}\sqrt{2\mathrm{A}n\log(n)}\,. \tag{37.12}$$

This result is complemented by the following lower bound.

THEOREM 37.5 *Let $\mathrm{S} \geq 3$, $\mathrm{A} \geq 2$, $D \geq 6 + 2\log_{\mathrm{A}}\mathrm{S}$ and $n \geq D\mathrm{SA}$. Then for any policy π there exists a Markov decision process with S states, A actions and diameter at most D such that*

$$\mathbb{E}[\hat{R}_n] \geq C\sqrt{D\mathrm{SA}n}\,,$$

where $C > 0$ is again a universal constant.

The upper and lower bounds are separated by a factor of at least $\sqrt{DS}$, which is a considerable gap. Recent work has made progress towards closing this gap as we explain in the notes.

37.5 Upper confidence bounds for reinforcement learning

Reinforcement learning is the subfield of machine learning devoted to designing and studying algorithms that learn to maximize long-term reward in sequential context, like the one we study. The algorithm that establishes Theorem 37.4 is called UCRL2 because it is the second version of the 'upper-confidence bounds for reinforcement learning' algorithm. Its pseudocode is shown in Algorithm 24.

At the start of each phase, UCRL2 computes an optimal policy for the statistically plausible MDP with the largest optimal gain. The details of this computation are left to the next section. This policy is then implemented until the number of visits to some state/action pair doubles when a new phase starts and the process begins again. The use of phases is important, not just for computational efficiency. Recalculating the optimistic policy in each round may lead to a dithering behavior in which the algorithm frequently changes its plan and suffers linear regret (Exercise 37.18).

To complete the specification of the algorithm, we must define confidence sets on the unknown quantity, which in this case is the transition matrix. The confidence sets are centered at the empirical transition probabilities defined by

$$\hat{P}_{t,a}(s,s') = \frac{\sum_{u=1}^{t} \mathbb{I}\{S_u = s, A_u = a, S_{u+1} = s'\}}{1 \vee T_t(s,a)} ,$$

where $T_t(s,a) = \sum_{u=1}^{t} \mathbb{I}\{S_u = s, A_u = a\}$ is the number of times action a was taken in state s. As before we let $\hat{P}_{t,a}(s)$ be the vector whose s'th entry is $\hat{P}_{t,a}(s,s')$. Given a state/action pair s, a define

$$\mathcal{C}_t(s,a) = \left\{ P \in \mathcal{P}(\mathcal{S}) : \|P - \hat{P}_{t-1,a}(s)\|_1 \leq \sqrt{\frac{\mathrm{S}L_{t-1}(s,a)}{1 \vee T_{t-1}(s,a)}} \right\} , \tag{37.13}$$

where for $T_t(s,a) > 0$ we set

$$L_t(s,a) = 2 \log \left(\frac{4\mathrm{SA}T_t(s,a)(1 + T_t(s,a))}{\delta} \right)$$

and for $T_t(s,a) = 0$ we set $L_t(s,a) = 1$. Note that in this case $\mathcal{C}_{t+1}(s,a) = \mathcal{P}(\mathcal{S})$. Then define confidence set on the space of transition kernels by

$$\mathcal{C}_t = \{P = (P_a(s))_{s,a} : P_a(s) \in \mathcal{C}_t(s,a) \text{ for all } s, a \in \mathcal{S} \times \mathcal{A}\} , \tag{37.14}$$

Clearly $T_t(s,a)$ cannot be larger than the total number of rounds n so

$$L_t(s,a) \leq L = 2\log\left(\frac{4\mathrm{SA}n(n+1)}{\delta}\right). \tag{37.15}$$

The algorithm operates in phases $k = 1, 2, 3, \ldots$ with the first phase starting in round $\tau_1 = 1$ and the $(k+1)$th phase starting in round τ_{k+1} defined inductively by

$$\tau_{k+1} = 1 + \min\left\{t : T_t(S_t, A_t) \geq 2T_{\tau_k - 1}(S_t, A_t)\right\},$$

which means that the next phase starts once the number of visits to some state at least doubles.

Input $\mathcal{S}$, $\mathcal{A}$, r, $\delta \in (0,1)$
$t = 0$
for $k = 1, 2, \ldots$ **do**
  $\tau_k = t + 1$
  Find π_k as the greedy policy with respect v_k satisfying Eq. (37.16)
  do
    $t \leftarrow t + 1$, observe S_t and take action $A_t = \pi_k(S_t)$
  while $T_t(S_t, A_t) < 2T_{\tau_k - 1}(S_t, A_t)$
end for

Algorithm 24: UCRL2

37.5.1 The extended Markov decision process

The confidence set $\mathcal{C}_t$ defines a set of plausible transition probability functions at the start of round t. Since the reward function is known already this corresponds to a set of plausible MDPs. The algorithm plays according to the optimal policy in the plausible MDP with the largest gain. There is some subtlety because the optimal policy is not unique and what is really needed is to find a policy that is greedy with respect to a value function satisfying the Bellman optimality equation in the plausible MDP with the largest gain. Precisely, at the start of the kth phase the algorithm must find a value function v_k, gain ρ_k and MDP $M_k = (\mathcal{S}, \mathcal{A}, P_k, r)$ with $P_k \in \mathcal{C}_{\tau_k}$ such that

$$\begin{aligned} \rho_k + v_k(s) &= \max_{a \in \mathcal{A}} r_a(s) + \langle P_{k,a}(s), v_k \rangle \text{ for all } s \in \mathcal{S} \text{ and } a \in \mathcal{A}, \\ \rho_k &= \max_{s \in \mathcal{S}} \max_{\pi \in \Pi_{\mathrm{DM}}} \max_{P \in \mathcal{C}_{\tau_k}} \rho_s^\pi(P), \end{aligned} \tag{37.16}$$

where $\rho_s^\pi(P)$ is the gain of deterministic memoryless policy π starting in state s in the MDP with transition probability function P. The algorithm then plays according to π_k defined as the greedy policy with respect to v_k. There is quite a lot hidden in these equations. The gain is only guaranteed to be constant when M_k has a finite diameter, but this may not hold for all plausible MDPs. As it

happens, however, solutions to Eq. (37.16) are guaranteed to exist and can be found efficiently. To see why this is true we introduce the **extended Markov decision process** $\tilde{M}_k$, which has state-space $\mathcal{S}$ and state-dependent action-space $\tilde{\mathcal{A}}_s$ given by

$$\tilde{\mathcal{A}}_s = \{(a, P) : a \in \mathcal{A},\ P \in \mathcal{C}_{\tau_k}(s, a)\}\ .$$

The reward function of the extended MDP is $\tilde{r}_{(a,P)}(s) = r_a(s)$ and the transitions are $\tilde{P}_{a,P}(s) = P_a(s)$. The action-space in the extended MDP allows the agent to choose both $a \in \mathcal{A}$ and a plausible transition vector $P_a(s) \in \mathcal{C}_{\tau_k}(s, a)$. By the definition of the confidence sets, for any pair of states s, s' and action $a \in \mathcal{A}$ there always exists a transition vector $P_a(s) \in \mathcal{C}_{\tau_k}(s, a)$ such that $P_a(s, s') > 0$, which means that $\tilde{M}_k$ is strongly connected. Hence solving the Bellman optimality equation for $\tilde{M}_k$ yields a value function v_k and constant gain $\rho_k \in \mathbb{R}$ that satisfy Eq. (37.16). A minor detail is that the extended action-sets are infinite while the analysis in previous sections only demonstrated existence of solutions to the Bellman optimality equation for finite MDPs. We leave it to the reader to convince themselves that $\mathcal{C}_t(s, a)$ is convex and has finitely many extremal points. Restricting the confidence sets to these points makes the extended MDP finite without changing the optimal policy.

37.5.2 Computing the optimistic policy (†)

Here we explain how to efficiently solve the Bellman optimality equation for the extended MDP. The results in Section 37.3 show that the Bellman optimality equation for $\tilde{M}_k$ can be solved efficiently provided that for any value function $v \in \mathbb{R}^{\mathrm{S}}$ computing

$$\operatorname{argmax}_{a \in \mathcal{A}} \left(r_a(s) + \max_{P \in \mathcal{C}_{\tau_k}(s,a)} \langle P, v \rangle \right) \tag{37.17}$$

can be carried out in an efficient manner. The inner optimization is another linear program with S variables and $O(\mathrm{S})$ constraints and can be solved in polynomial time. This procedure is repeated for each $a \in \mathcal{A}$ to compute the outcome of (37.17). In fact the inner optimization can be solved more straightforwardly by sorting the entries of v and then allocating P coordinate-by-coordinate to be as large as allowed by the constraints in decreasing order of v. The total computation cost of solving Eq. (37.17) in this way is $O(\mathrm{S}(\mathrm{A} + \log \mathrm{S}))$. Combining this with Algorithm 23 gives the required separation oracle.

The next problem is to find an R such that the set of feasible solutions to the linear programs in Eq. (37.6) and Eq. (37.7) are contained in the set $\{x : \|x\| \le R\}$. As discussed beforehand, a suitable value is $R = \sqrt{1 + D^2 \mathrm{S}}$ where D is an upper bound on the diameter of the MDP. It turns out that $D = \sqrt{n}$ works because for each pair of states s, s' there exists an action a and $P \in \mathcal{C}_{\tau_k}(s, a)$ such that $P(s, s') \ge 1 \wedge (1/\sqrt{n})$ so $D(\tilde{M}_k) \le \sqrt{n}$. Combining this with the tools developed in Section 37.3 shows that the Bellman optimality equation for $\tilde{M}_k$ may be

solved using linear programming in polynomial time. Note that the additional constraints require a minor adaptation of the separation oracle, which we leave for the reader.

37.6 Proof of upper bound

The proof is developed in three steps. First we decompose the regret into phases and define a failure event where the confidence intervals fail. In the second step we bound the regret in each phase and in the third step we sum over the phases. Recall that $M = (\mathcal{S}, \mathcal{A}, P, r)$ is the true Markov decision process with diameter $D = D(M)$. The initial state distribution is $\mu \in \mathcal{P}(\mathcal{S})$, which is arbitrary.

Step 1: Failure events and decomposition
Let K be the (random) number of phases and for $k \in [K]$ let $E_k = \{\tau_k, \tau_k + 1, \ldots, \tau_{k+1} - 1\}$ be the set of rounds in the kth phase where τ_{K+1} is defined to be $n + 1$. Let $T_{(k)}(s, a)$ be the number of times state/action pair s, a is visited in the kth phase:

$$T_{(k)}(s, a) = \sum_{t \in E_k} \mathbb{I}\{S_t = s, A_t = a\} \; .$$

Define F as the failure event that $P \notin \mathcal{C}_{\tau_k}$ for some $k \in [K]$. The first lemma shows that F has lower probability:

LEMMA 37.2 $\mathbb{P}(F) \leq \delta/2$.

The proof is based on a concentration inequality derived for categorical distributions and is left for Exercise 37.20. When F does not hold, the true transition kernel is in $\mathcal{C}_{\tau_k}$ for all k, which means that $\rho^* \leq \rho_k$ and

$$\hat{R}_n = \sum_{t=1}^{n} (\rho^* - r_{A_t}(S_t)) \leq \sum_{k=1}^{K} \underbrace{\sum_{t \in E_k} (\rho_k - r_{A_t}(S_t))}_{\tilde{R}_k} \; .$$

In the next step we bound $\tilde{R}_k$ under the assumption that F does not hold.

Step 2: Bounding the regret in each phase
Assume that F does not occur and fix $k \in [K]$. Recall that v_k is a value function satisfying the Bellman optimality equation in the optimistic MDP M_k and ρ_k is its gain. Hence

$$\rho_k = r_{\pi_k}(s) - v_k(s) + \langle P_{k,\pi_k}(s), v_k \rangle \quad \text{for all } s \in \mathcal{S} \, . \tag{37.18}$$

As noted earlier, solutions to the Bellman optimality equation remain solutions when translated so we may assume without loss of generality that v_k is such that

$\|v_k\|_\infty \le \operatorname{span}(v_k)/2$, which means that

$$\|v_k\|_\infty \le \frac{1}{2}\operatorname{span}(v_k) \le \frac{D}{2}\,, \tag{37.19}$$

where the second inequality follows from Lemma 37.1 and the fact that when F does not hold the diameter of the extended MDP $\tilde{M}_k$ is at most D and v_k also satisfies the Bellman-optimality equation in this MDP. By the definition of the policy we have $A_t = \pi_k(S_t)$ for $t \in E_k$, which implies that

$$\rho_k = r_{A_t}(S_t) - v_k(S_t) + \langle P_{k,A_t}(S_t), v_k\rangle \quad \text{for all } t \in E_k\,.$$

Rearranging and substituting yields

$$\begin{aligned}
\tilde{R}_k &= \sum_{t\in E_k} \left(-v_k(S_t) + \langle P_{k,A_t}(S_t), v_k\rangle\right) \\
&= \sum_{t\in E_k} \left(-v_k(S_t) + \langle P_{A_t}(S_t), v_k\rangle\right) + \sum_{t\in E_k} \langle P_{k,A_t}(S_t) - P_{A_t}(S_t), v_k\rangle \\
&\le \underbrace{\sum_{t\in E_k} \left(-v_k(S_t) + \langle P_{A_t}(S_t), v_k\rangle\right)}_{\text{(A)}} + \underbrace{\frac{D}{2}\sum_{t\in E_k} \|P_{k,A_t}(S_t) - P_{A_t}(S_t)\|_1}_{\text{(B)}}\,,
\end{aligned} \tag{37.20}$$

where the inequality follows from Hölder's inequality and Eq. (37.19). Let $\mathbb{E}_t[\cdot]$ denote the conditional expectation with respect to $\mathbb{P}$ conditioned on $\sigma(S_1, A_1, \ldots, S_{t-1}, A_{t-1}, S_t)$. To bound (A) we reorder the terms and use the fact that $\operatorname{span}(v_k) \le D$ on the event F^c. We get

$$\begin{aligned}
\text{(A)} &= \sum_{t\in E_k} \left(v_k(S_{t+1}) - v_k(S_t) + \langle P_{A_t}(S_t), v_k\rangle - v_k(S_{t+1})\right) \\
&= v_k(S_{\tau_{k+1}}) - v_k(S_{\tau_k}) + \sum_{t\in E_k} \left(\langle P_{A_t}(S_t), v_k\rangle - v_k(S_{t+1})\right) \\
&\le D + \sum_{t\in E_k} \left(\mathbb{E}_t[v_k(S_{t+1})] - v_k(S_{t+1})\right)\,,
\end{aligned}$$

where the second equality used that $\max E_k = \tau_{k+1} - 1$ and $\min E_k = \tau_k$. We leave this here for now and move on to term (B) in Eq. (37.20). The definition of the confidence intervals and the assumption that F does not occur shows that

$$\text{(B)} \le \frac{D\sqrt{LS}}{2} \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \frac{T_{(k)}(s,a)}{\sqrt{1 \vee T_{\tau_k - 1}(s,a)}}\,.$$

Combining the bounds (A) and (B) yields

$$\tilde{R}_k \le D + \sum_{t\in E_k} \left(\mathbb{E}_t[v_k(S_{t+1})] - v_k(S_{t+1})\right) + \frac{D\sqrt{LS}}{2} \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \frac{T_{(k)}(s,a)}{\sqrt{1 \vee T_{\tau_k - 1}(s,a)}}\,.$$

Step 3: Bounding the number of phases and summing
Let K_t be the phase in round t so that $t \in E_{K_t}$. By the work in the previous two steps, if F does not occur then

$$\hat{R}_n \le \sum_{k=1}^{K} \tilde{R}_k \le KD + \sum_{t=1}^{n} (\mathbb{E}_t[v_{K_t}(S_{t+1})] - v_{K_t}(S_{t+1})) + \frac{D\sqrt{LS}}{2} \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \sum_{k=1}^{K} \frac{T_{(k)}(s,a)}{\sqrt{1 \vee T_{\tau_k - 1}(s,a)}} .$$

The first sum is bounded using a version of Hoeffding–Azuma (Exercise 20.6):

$$\mathbb{P}\left(F^c \text{ and } \sum_{t=1}^{n} (\mathbb{E}_t[v_{K_t}(S_{t+1})] - v_{K_t}(S_{t+1})) \ge D\sqrt{\frac{n\log(2/\delta)}{2}}\right) \le \frac{\delta}{2} .$$

For the second term we note that $T_{(k)}(s,a)/\sqrt{1 \vee T_{\tau_k-1}(s,a)}$ cannot be large too often. A continuous approximation often provides intuition for the correct form. Recalling the thousands of integrals you did at school, for any differentiable $f : [0,\infty) \to \mathbb{R}$ we have

$$\int_0^K \frac{f'(k)}{\sqrt{f(k)}} dk = 2\sqrt{f(K)} - 2\sqrt{f(0)} . \tag{37.21}$$

Here we are thinking of $f(k)$ as the continuous approximation of $T_{\tau_k-1}(s,a)$ and its derivative as $T_{(k)}(s,a)$. In Exercise 37.21 we ask you to make this argument rigorous by showing that

$$\sum_{k=1}^{K} \frac{T_{(k)}(s,a)}{\sqrt{1 \vee T_{\tau_k-1}(s,a)}} \le \left(\sqrt{2}+1\right)\sqrt{T_n(s,a)} .$$

Then by Cauchy-Schwartz and the fact that $\sum_{s,a\in\mathcal{S}\times\mathcal{A}} T_n(s,a) = n$,

$$\sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} \sqrt{T_n(s,a)} \le \sqrt{\mathrm{SA}n} .$$

It remains to bound the number of phases. A new phase starts when the visit count for some state/action pair doubles. Hence K cannot be more than the number of times the counters double in total for each of the states. It is easy to see that $1 + \log_2 T_n(s,a)$ gives an upper bound on how many times the counter for this pair may double (the constant 1 is there to account for the counter changing from zero to one). Thus $K \le K' = \sum_{s,a} 1 + \log_2 T_n(s,a)$. Noting that $0 \le T_n(s,a)$ and $\sum_{s,a} T_n(s,a) = n$ and relaxing $T_n(s,a)$ to take real values we find that the value of K' is the largest when $T_n(s,a) = n/(\mathrm{SA})$, which shows that

$$K \le \mathrm{SA}\left(1 + \log_2\left(\frac{n}{\mathrm{SA}}\right)\right) .$$

Putting everything together gives the desired result.

37.7 Proof of lower bound

The lower bound is proven by crafting a difficult MDP that models a bandit with approximately SA arms. This a cumbersome endeavour, but intuitively straightforward and the explanations that follow should be made clear in Fig. 37.3. Given S and A, the first step is to construct a tree of minimum depth with at most A children for each node using exactly $\mathrm{S}-2$ states. The root of the tree is denoted by $s_\circ$ and transitions within the tree are deterministic, so in any given node the learner can simply select which child to transition to. Let L be the number of leaves and label these states $s_1, \dots, s_L$. The last two states are s_g and s_b ('good' and 'bad' respectively). For each $i \in [L]$ the learner can take any action $a \in \mathcal{A}$ and transitions to either the good state or the bad state according to

$$P_a(s_i, s_g) = \frac{1}{2} + \varepsilon(a,i) \qquad \text{and} \qquad P_a(s_i, s_b) = \frac{1}{2} - \varepsilon(a,i)\,.$$

The function ε will be chosen so that $\varepsilon(a,i) = 0$ for all (a,i) pairs except one. For this special state/action pair we let $\varepsilon(a,i) = \Delta$ for appropriately tuned $\Delta > 0$. The good state and the bad state have the same transitions for all actions:

$$\begin{aligned} P_a(s_g, s_g) &= 1-\delta\,, & P_a(s_g, s_\circ) &= \delta\,, \\ P_a(s_b, s_b) &= 1-\delta\,, & P_a(s_b, s_\circ) &= \delta\,. \end{aligned}$$

Choosing $\delta = 4/D$, which under the assumptions of the theorem is guaranteed to be in $(0,1]$, ensures that the diameter of the described MDP is at most D, regardless the value of Δ. The reward function is $r_a(s) = 1$ if $s = s_g$ and $r_a(s) = 0$ otherwise.

The connection to finite-armed bandits is straightforward. Each time the learner arrives in state $s_\circ$ it selects which leaf to visit and then chooses an action from that leaf. This corresponds to choosing one of $K = L\mathrm{A} = \Omega(\mathrm{SA})$ meta actions. The optimal policy is to select the meta action with the largest probability of transitioning to the good state. The choice of δ means the learner expects to stay in the good/bad state for approximately D rounds, which also makes the diameter of this MDP about D. All up this means the learner expects to make about n/D decisions and the rewards are roughly in $[0, D]$ so we should expect the regret to be $\Omega(D\sqrt{n/DK}) = \Omega(\sqrt{nD\mathrm{SA}})$.

One could almost claim victory here and not bother with the proof. As usual, however, there are some technical difficulties, which in this case arise because the number of visits to the decision state $s_\circ$ is a random quantity. For this reason we give the proof, leaving as exercises the parts that are both obvious and annoying.

Proof of Theorem 37.5 The proof follows the path suggested in Exercise 15.1. We break things up into two steps. Throughout we fix an arbitrary policy π.

Step 1: Notation and facts about the MDP
Let d be the depth of the tree in the MDP construction and L the number of leaves and $K = L\mathrm{A}$ ($d = 3$ and $L = 3$ in Fig. 37.3). This leaves K state-action

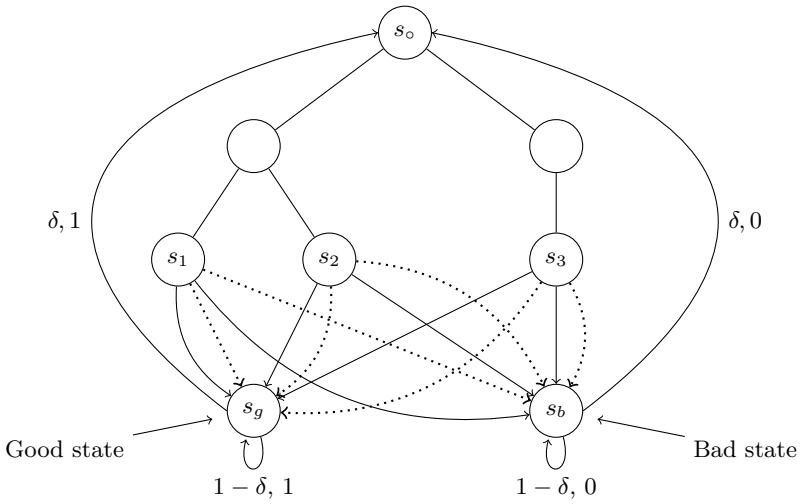


Figure 37.3 Lower bound construction for A = 2 and S = 8. The resulting MDP is roughly equivalent to a bandit with six actions.

pairs that potentially lead to either s_g or s_b. Let M_0 be the MDP with $\varepsilon(s,a) = 0$ for all relevant state-action pairs s, a and M_k be the MDP with $\varepsilon(s,a) = \Delta$ for the kth state/action pair with the state on the fringe of the tree, ordered in some arbitrary way. Define stopping time τ by

$$\tau = n \wedge \min \left\{ t : \sum_{s=1}^{t} \mathbb{I}\{S_t = s_\circ\} \geq \frac{n}{D} - 1 \right\},$$

which is the first round when the number of visits to state $s_\circ$ is at least $n/D - 1$, or n if $s_\circ$ is visited fewer times than n/D. Next let T_k be the number of visits to state-action pair $k \in [K]$ until stopping time τ and $T_\sigma = \sum_{k=1}^{K} T_k$. For $0 \leq k \leq K$, let $\mathbb{P}_k$ be the law of $T_1, \ldots, T_K$ induced by the interaction of π and M_k and $\mathbb{E}_k[\cdot]$ the expectation with respect to $\mathbb{P}_k$. None of the following claims are surprising, but they are all tiresome to prove to some extent. The claims are listed in increasing order of difficulty and left to the reader in Exercise 37.24.

CLAIM 37.1 *For all $k \in [K]$ the diameter is bounded by $D(M_k) \leq D$.*

CLAIM 37.2 *There exist universal constants $0 < c_1 < c_2 < \infty$ such that*

$$D\mathbb{E}_0[T_\sigma]/n \in [c_1, c_2].$$

CLAIM 37.3 *Let R_{nk} be the expected regret of policy π in MDP M_k over n rounds. There exists a universal constant $c_3 > 0$ such that*

$$R_{nk} \geq c_3 \Delta D \, \mathbb{E}_k[T_\sigma - T_k].$$

Step 2: Bounding the regret
Notice that M_0 and M_k only differ when state-action pair k is visited. In Exercise 37.29 you are invited to use this fact and the chain rule for relative entropy given in Exercise 14.9 to prove that

$$\mathrm{D}(\mathbb{P}_0, \mathbb{P}_k) = \mathbb{E}_0[T_k] d(1/2, 1/2 + \Delta)\,, \tag{37.22}$$

where $d(p, q)$ is the relative entropy between Bernoulli distributions with biases p and q respectively. Since $\Delta \leq 1/4$ it follows from the entropy inequalities in Eq. (14.12) that

$$\mathrm{D}(\mathbb{P}_0, \mathbb{P}_k) \leq 4\Delta^2 \mathbb{E}_0[T_k]\,, \tag{37.23}$$

Using the fact that $0 \leq T_\sigma - T_k \leq T_\sigma \leq n/D$, Exercise 14.1, Pinsker's inequality (Eq. (14.9)) and (37.23),

$$\mathbb{E}_k\left[T_\sigma - T_k\right] \geq \mathbb{E}_0\left[T_\sigma - T_k\right] - \frac{n}{D}\sqrt{\frac{\mathrm{D}(\mathbb{P}_0, \mathbb{P}_k)}{2}} \geq \mathbb{E}_0\left[T_\sigma - T_k\right] - \frac{n\Delta}{D}\sqrt{2\mathbb{E}_0[T_k]}\,.$$

Summing over k and applying Cauchy-Schwartz yields

$$\begin{aligned}
\sum_{k=1}^{K} \mathbb{E}_k\left[T_\sigma - T_k\right] &\geq \sum_{k=1}^{K} \mathbb{E}_0\left[T_\sigma - T_k\right] - \frac{n\Delta}{D}\sum_{k=1}^{K}\sqrt{2\mathbb{E}_0[T_k]} \\
&\geq (K-1)\mathbb{E}_0\left[T_\sigma\right] - \frac{n\Delta}{D}\sqrt{2K\mathbb{E}_0[T_\sigma]} \\
&\geq \frac{c_1 n(K-1)}{D} - \frac{n\Delta}{D}\sqrt{\frac{2c_2 nK}{D}} \\
&\geq \frac{c_1 n(K-1)}{2D}\,,
\end{aligned} \tag{37.24}$$

where the last inequality follows by choosing

$$\Delta = \frac{c_1(K-1)}{2}\sqrt{\frac{D}{2c_2 nK}}\,.$$

By Eq. (37.24) there exists a $k \in [K]$ such that

$$\mathbb{E}_k\left[T_\sigma - T_k\right] \geq \frac{c_1 n(K-1)}{2DK}\,.$$

Then for last step apply Claim 37.3 to show that

$$R_{nk} \geq c_3 D\Delta \mathbb{E}_k[T_\sigma - T_k] \geq \frac{c_1^2 c_3 n(K-1)^2}{4K}\sqrt{\frac{D}{2c_2 nK}}\,.$$

Naive bounding and simplification concludes the result. □

37.8 Notes

1 MDPs in applications can have millions (or "Billions and Billions") of states, which should make the reader worried that the bound in Theorem 37.4 could

be extremely large. The takeaway should be that learning in large MDPs without additional assumptions is hard, as attested by the lower bound in Theorem 37.5.

2 The key to choosing the state space is that the state must be observable and sufficiently informative that the Markov property is satisfied. Blowing up the size of the state space may help to increase the fidelity of the approximation (the entire history always works), but will almost always slow down learning.

3 We simplified the definition of MDPs by making the rewards a deterministic function of the current state and the action chosen. A more general definition allows the rewards to evolve in a random fashion, jointly with the next state. In this definition, the mean reward functions are dropped and the transition kernel P_a is replaced with an $\mathcal{S} \to \mathcal{S} \times \mathbb{R}$ stochastic kernel, call it, $\tilde{P}_a$. Thus, for every $s \in \mathcal{S}$, $\tilde{P}_a(s)$ is a probability measure over $\mathcal{S} \times \mathbb{R}$. The meaning of this is that when action a is chosen in state s, a random transition, $(S, R) \sim \tilde{P}_a(s)$ happens to state S, while reward R is received. Note that the mean reward along this transition is $r_a(s) = \int x \tilde{P}_a(s, ds', dx)$.

4 A state $s \in \mathcal{S}$ is **absorbing** if $P_a(s, s) = 1$ for all $a \in \mathcal{A}$. An MDP is **episodic** if there exists an absorbing state that is reached almost surely by any policy. The average reward criterion is meaningless in episodic MDPs because all policies are optimal. In this case the usual objective is to maximize the expected reward until the absorbing state is reached without limits or normalization, sometimes with discounting. An MDP is **finite-horizon** if it is episodic and the absorbing state is always reached after some fixed number of rounds. The learning community studies these in the same way as bandits, where in each 'round' the learner interacts with the MDP from some starting state until the absorbing state is reached. The simplification of the setting eases the analysis and preserves most of the intuition from the general setting.

5 A **partially observable MDP** is a generalization where the learner does not observe the underlying state. Instead they receive an observation that is a (possibly random) function of the state. Given a fixed (known) initial state distribution, any POMDP can be mapped into an MDP at the price of enlarging the state space. A simple way to achieve this is to let the new state be the space of all histories. Alternatively you can use any sufficient statistic for the hidden state as the state. A natural choice is the posterior distribution over the hidden state given the interaction history, which is called the **belief space**. While the value function over the belief space has some nice structure, in general even computing the optimal policy is hard [Papadimitriou and Tsitsiklis, 1987].

6 We called the all-knowing entity that interacts with the MDP an **agent**. In operations research the term is **decision maker** and in control theory it is **controller**. In control theory the environment would be called the **controlled system** or the **plant** (for power-plant, not a biological plant). Acting in an MDP is studied in control theory under **stochastic optimal control**, while in operations research the area is called **multistage decision making**

under uncertainty or **multistage stochastic programming**. In the control community the infinite horizon setting with the average cost criterion is perhaps the most common, while in operations research the episodic setting is typical.

7 The definition of the optimal gain that is appropriate for MDPs that are not strongly connected is a vector $\rho^* \in \mathbb{R}^{\mathrm{S}}$ given by $\rho^*_s = \sup_\pi \bar{\rho}^\pi_s$. A policy is optimal if it achieves the supremum in this definition and such a policy always exists as long as the MDP is finite. In strongly connected MDPs the two definitions coincide. For infinite MDPs everything becomes more delicate and a large portion of the literature on MDPs is devoted to this case.

8 In applications where the asymptotic nature of gain optimality is unacceptable there are criteria that make finer distinctions between the policies. A memoryless policy π^* is **bias optimal** if it is gain optimal and $v_{\pi^*} \geq v_\pi$ for all memoryless policies π. Even more sensitive criteria exist. Some keywords to search for are **Blackwell optimality** and ***n*-discount** optimality.

9 The **Cesàro sum** of a real-valued sequence $(a_n)_n$ is the asymptotic average of its partial sums. Let $s_n = a_0 + \cdots + a_{n-1}$ be the nth partial sum. The Cesàro sum of this sequence is $A = \lim_{n\to\infty} \frac{1}{n}(s_1 + \cdots + s_n)$ when this limit exists. The idea is that Cesàro summation smoothes out periodicity, which means that for certain sequences the Cesáro sum exists while s_n does not converge. For example, the alternating sequence $(+1, -1, +1, -1, \dots)$ is Cesàro summable and its Cesàro sum is easily seen to be $1/2$, while it is not summable in the normal sense. If a sequence is summable, then its sum and its Cesàro sum coincide. The differential value of a policy is defined as a Cesàro sum so that it is well-defined even if the underlying Markov chain has periodic states.

10 For $\gamma \in (0, 1)$ the γ-discounted average of sequence $(a_n)_n$ is $A_\gamma = (1 - \gamma)\sum_{n=0}^\infty \gamma^n a_n$. An elementary argument shows that if A_γ is well-defined, then $A_\gamma = (1-\gamma)^2 \sum_{n=1}^\infty \gamma^{n-1} s_n$. Suppose the Cesáro sum $A = \lim_{n\to\infty} \frac{1}{n}\sum_{t=1}^n s_t$ exists, then using the fact that $1 = (1-\gamma)^2 \sum_{n=1}^\infty \gamma^{n-1} n$ we have $A_\gamma - A = (1-\gamma)^2 \sum_{n=1}^\infty \gamma^{n-1}(s_n - nA)$. It is not hard to see that $|\sum_{n=1}^\infty \gamma^{n-1}(s_n - nA)| = O(1/(1-\gamma))$ and thus $A_\gamma - A = O(1-\gamma)$ as $\gamma \to 1$, which means that $\lim_{\gamma\to 1} A_\gamma = A$. The value $\lim_{\gamma\to 1} A_\gamma$ is called the **Abel sum** of $(a_n)_n$. Put simply, the Abel sum of a sequence is equal to its Cesàro sum when the latter exists. Abel summation is stronger in the sense that there are sequences that are Abel summable but not Cesáro summable. The approach of approximating Cesàro sums through γ-discounted averages and taking the limit as $\gamma \to 1$ is called the **vanishing discount approach** and is one of the standard ways to prove the (average reward) Bellman equation has a solution (see Exercises 37.10 and 37.11). As an aside, the systematic study of how to define the 'sum' of a divergent series is relatively modern endeavour. An enjoyable historical account is given in the first chapter of the book on the topic by Hardy [1973].

11 Given a solution (ρ, v) to Eq. (37.6) we mentioned a procedure for finding a state $\tilde{s} \in \mathcal{S}$ that is recurrent under some optimal policy. This works as follows. Let $C_0 = \{(s, a) : \rho + v(s) = r_a(s) + \langle P_a(s), v\rangle\}$ and $I_0 = \{s : (s, a) \in C_0 \text{ for some } a \in \mathcal{A}\}$. Then define C_{k+1} and I_{k+1} inductively by the

following algorithm. First find an $(s,a) \in C_k$ such that $P_a(s,s') > 0$ for some $s' \notin I_k$. If no such pair exists then halt. Otherwise let $C_{k+1} = C_k \setminus \{(s,a)\}$ and $I_{k+1} = \{s : (s,a) \in C_{k+1} \text{ for some } a \in \mathcal{A}\}$. Now use the complementary slackness conditions of the dual program to Eq. (37.6) to prove that the algorithm halts with some non-empty I_k and that these states are recurrent under some optimal policy. For more details have a look at Exercise 4.15 of the second volume of the book by Bertsekas [2012].

12 As it has already been noted at the beginning of the book, 'operator' is just a fancy word for 'function', reserved usually to those cases, when the domain is large, an example of which is when the domain is a set of functions itself (the expectation operator is of course of this form). The Bellman operator is a function from the space of value functions to itself. Operators are usually denoted by capital letters and brackets are omitted in their application so that Tv is shorthand for $T(v)$. It is not a requirement of the definition, but operators are usually defined on spaces of functions and preserve certain structures of the space.

13 We mentioned enumeration, value iteration and policy iteration as other methods for computing optimal policies. Enumeration just means enumerating all deterministic memoryless policies and selecting the one with the highest gain. This is obviously too expensive. **Policy iteration** is an iterative process that starts with a policy π_0. In each round the algorithm computes π_{k+1} from π_k by computing v_{π_k} and then choosing π_{k+1} to be the greedy policy with respect to v_{π_k}. In general this method may not converge to an optimal policy, but by slightly modifying the update process one can prove convergence. For more details see Chapter 4 of Volume 2 of the book by Bertsekas [2012]. **Value iteration** works by choosing an arbitrary value function v_0 and then inductively defining $v_{k+1} = Tv_k$ where $(Tv)(s) = \max_{a\in\mathcal{A}} r_a(s) + \langle P_a(s), v\rangle$ is the Bellman operator. Under certain technical conditions one can prove that the greedy policy with respect to v_k converges to an optimal policy. Note that $v_{k+1} = \Omega(k)$, which can be a problem numerically. A simple idea is to let $v_{k+1} = Tv_k - \delta_k$ where $\delta_k = \max_{s\in\mathcal{S}} v_k(s)$. Since the greedy policy is the same for v and $v + c\mathbf{1}$ this does not change the mathematics, but improves the numerical situation. The aforementioned book by Bertsekas is again a good source for more details. Unfortunately none of these algorithms have known polynomial time guarantees on the computation complexity of finding an optimal policy without stronger assumptions than we would like. In practice, however, both value and policy iteration work quite well, while the ellipsoid method for solving linear programs should be avoided at all costs.

14 One can modify the concept of regret to allow for MDPs that have traps, allowing for finite MDPs with infinite diameter. In any finite MDP there exists finitely many disjoint classes of states (what these classes are depends only on the MDP structure) so that each class is a trap in the sense that no policy can escape from it once entered. Now, rule out all those policies that have linear regret in strongly connected MDPs as a reasonable learner should achieve

sublinear regret in such MDPs. What remains are policies that will necessarily get trapped in any MDP that is not not strongly connected. For such MDPs, the regret is redefined by 'restarting the clock' at the time when the policy gets trapped. For details, see Exercise 37.28, where you are also asked to show a policy that achieves sublinear regret in any finite MDP.

15 The assumption that the reward function is known can be relaxed without difficulty. It is left as an exercise to figure out how to modify algorithm and analysis to the case when r is unknown and reward observed in round t is bounded in $[0, 1]$ and has conditional mean $r_{A_t}(S_t)$. See Exercise 37.23.

16 Although it has not been done yet in this setting, the path to removing the spurious $\sqrt{\mathrm{S}}$ from the bound is to avoid the application of Cauchy-Schwartz in Eq. (37.20). Instead one should define confidence intervals directly on $\langle \hat{P}_k - P, v_k \rangle$, where the dependence on the state and action has been omitted. Of course this requires one to modify the algorithm. At first sight it seems that one could apply Hoeffding's bound directly to the inner product, but there is a subtle problem that has spoiled a number of attempts: v_k and $\hat{P}_k$ are not independent. This non-independence is unfortunately quite pernicious and appears from many angles. We advise extreme caution (some references for guidance are given at in the bibliographic remarks).

37.9 Bibliographical remarks

Richard Bellman

The study of sequential decision making has a long history and we recommend the introduction of the book by Puterman [2009] as a good starting point. One of the main architects in modern times is Richard Bellman, who wrote an influential book [Bellman, 1954]. Bellman had an interesting life, working at Los Alamos near the end of the war and later at RAND. Besides 'dynamic programming' he also coined the term 'curse of dimensionality' which, although it is not his fault, curses us still today. His autobiography is so entertaining that reading it slowed the writing of this chapter: 'The Eye of the Hurricane' [Bellman, 1984]. As a curiosity, Bellman knew about bandit problems after accidentally encountering a paper by Thompson [1935]. For the tidbit see page 260 of the aforementioned biography.

Markov decision processes are studied by multiple research communities, including control, operations research and artificial intelligence. The two-volume book by Bertsekas [2012] provides a thorough and formal introduction to the basics. The perspective is quite interdisciplinary, but with a slight (good) bias towards the control literature. The perspective of an operations researcher is most precisely conveyed in the comprehensive book by Puterman [2009]. A very

readable shorter introductory book is by Ross [1983]. Arapostathis et al. [1993] surveyed existing analytical results (existence, uniqueness of optimal policies, validity of the Bellman optimality equation) for average-reward MDPs with an emphasis on continuous state and action space models. The online lecture notes of Kallenberg [2016] are a recent comprehensive alternate account for the theory of discrete MDPs. There are many texts on linear/convex optimization and the ellipsoid method. The introductory book on linear optimization by Bertsimas and Tsitsiklis [1997] is a pleasant read while the ellipsoid method is explained in detail by Grötschel et al. [2012].

The problem considered in this chapter is part of a broader field called reinforcement learning (RL), which has recently seen a surge of interest. The books by Sutton and Barto [1998] and Bertsekas and Tsitsiklis [1996] describe the foundations. The first book provides an intuitive introduction aimed at computer scientists, while the second book focuses on the theoretical results of the fundamental algorithms. A book by one of the present authors focuses on cataloging the range of learning problems encountered in reinforcement learning and summarizing the basic ideas and algorithms [Szepesvári, 2010].

The UCRL algorithm and the upper and lower regret analysis is due to Auer et al. [2009, 2010]. Our proofs differ in minor ways. A more significant difference is that these works used value iteration for finding the optimistic policy and hence cannot provide polynomial time computation guarantees. In practice this may be preferable to linear programming anyway.

The number of rigorous results for bounding the regret of various algorithms is limited. One idea is to replace the optimistic approach with Thompson sampling, which was first adapted to reinforcement learning by Strens [2000] under the name PSRL (posterior sampling reinforcement learning). Agrawal and Jia [2017] recently made an attempt to improve the dependence of the regret on the state-space. The proof is not quite correct, however, and at the time of writing the holes of not yet been patched. Azar et al. [2017] also improve upon the UCRL2 bound, but for finite-horizon episodic problems where they derive an optimistic algorithm with regret $\tilde{O}(\sqrt{HSAn})$, which after adapting UCRL to the episodic setting improves on its regret by a factor of $\sqrt{SH}$. The main innovation is to use Freedman's Bernstein-style inequality for computing bonuses directly while computing action values using backwards induction from the end of the episode rather than keeping confidence estimates for the transition probabilities. An issue with both of these improvements is that lower-order terms in the bounds mean they only hold for large n. It remains to be seen if these terms arise from the analysis or if the algorithms need modification.

Tewari and Bartlett [2008] use an optimistic version of linear programming to obtain finite-time logarithmic bounds with suboptimal instance dependent constants. Note this paper mistakenly drops some constants from the confidence intervals, which after fixing would make the constants even worse. Similar results are also available for UCRL2 [Auer and Ortner, 2007]. Burnetas and Katehakis [1997a] prove asymptotic guarantees with optimal constants, but with

the crucial assumption that the support of the next-state distributions $P_a(s)$ are known. Lai and Graves [1997] also consider asymptotic optimality. However, they consider general state spaces where the set of transition probabilities is smoothly parameterized with a known parameterization, but under the weakened goal of competing with the best of finitely many memoryless policies given to the learner as black-boxes.

Finite-time regret for large state and action space MDPs under additional structural assumptions are also considered by Abbasi-Yadkori and Szepesvári [2011], Abbasi-Yadkori [2012], Ortner and Ryabko [2012]. Abbasi-Yadkori and Szepesvári [2011] and Abbasi-Yadkori [2012] give algorithms with $O(\sqrt{n})$ regret for linearly parameterized MDP problems with quadratic cost (linear quadratic regulation, or LQR), while Ortner and Ryabko [2012] gives $O(n^{(2d+1)/(2d+2)})$ regret bounds under a Lipschitz assumption, where d is the dimensionality of the state space. The algorithms in these works are not guaranteed to be computationally efficient because they rely on optimistic policies. In theory, this could be addressed by Thompson sampling, which is that is considered by Abeille and Lazaric [2017b] who obtain partial results for the LQR setting. Thompson sampling has also been studied in the Bayesian framework by Osband et al. [2013], Abbasi-Yadkori and Szepesvári [2015], Osband and Van Roy [2017], Theocharous et al. [2017], of which Abbasi-Yadkori and Szepesvári [2015] and Theocharous et al. [2017] consider general parametrizations, while the other papers are concerned with finite state/action MDPs. Learning in MDPs has also been studied in the Probability Approximately Correct (PAC) framework introduced by Kearns and Singh [2002] where the objective is to design policies for which the number of badly suboptimal actions is small with high probability. The focus of these papers is on the discounted reward setting rather than average reward. The algorithms are again built on the optimism principle. Algorithms that are known to be PAC-MDP include R-max Brafman and Tennenholtz [2003], Kakade [2003], MBIE Strehl and Littman [2005, 2008], Delayed Q-learning Strehl et al. [2006], the optimistic-initialization-based algorithm of Szita and Lőrincz [2009], MorMax by Szita and Szepesvári [2010], and an adaptation of UCRL by Lattimore and Hutter [2012], which they call UCRLγ. The latter work presents optimal results (matching upper and lower bounds) for the case when the transition structure is sparse, while the optimal dependence on the number of state/action pairs is achieved by Delayed Q-learning and Mormax [Strehl et al., 2006, Szita and Szepesvári, 2010], though the Mormax bound is better in its dependency on the discount factor. The idea to incorporate the uncertainty in the transitions into the action-space to solve the optimistic optimization problem appeared in the analysis of MBIE [Strehl and Littman, 2008]. A hybrid between stochastic and adversarial settings is when the reward sequence is chosen by an adversary, while transitions are stochastic. This problem has been introduced by Even-Dar et al. [2004]. State-of-the-art results for the bandit case are due to Neu et al. [2014], where the reader can also find further pointers to the literature. The

case when both the rewards and the transition probability distributions are also adversarially chosen in various cases by [Abbasi-Yadkori et al., 2013].

37.10 Exercises

37.1 Let $M = (\mathcal{S}, \mathcal{A}, P)$ be a finite **controlled Markov environment**, which is a finite Markov decision process without the reward function. Let π be an arbitrary policy for this environment, i.e., $\pi : \cup_{t=0}^{\infty}(\mathcal{S} \times \mathcal{A})^t \times \mathcal{S} \times \mathcal{A} \to [0, 1]$ such that for any $t \geq 0$, $h_t \in (\mathcal{S} \times \mathcal{A})^t \times \mathcal{S}$, $\sum_{a \in \mathcal{A}} \pi(h_t, a) = 1$, and fix a distribution $\mu \in \mathcal{P}(\mathcal{S})$ in an arbitrary manner. Show that there exists a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and an infinite sequence $(S_1, A_1, S_2, A_2, \dots)$ of random elements on it such that for $t \in \mathbb{N}$, S_t is $\mathcal{S}$-valued, A_t is $\mathcal{A}$-valued, and for any $s, s' \in \mathcal{S}$, $a \in \mathcal{A}$ and $t \in \mathbb{N}$,

(a) $\mathbb{P}(S_1 = s) = \mu(s)$;
(b) $\mathbb{P}(S_{t+1} = s' \mid H_t, A_t) = P_{A_t}(S_t, s')$;
(c) $\mathbb{P}(A_t = a \mid H_t) = \pi(H_t, a)$,

where $H_t = (S_1, A_1, \dots, S_{t-1}, A_{t-1}, S_t)$.

Use Theorem 3.3.

37.2 Let $M = (\mathcal{S}, \mathcal{A}, P)$ be a finite controlled Markov environment, π be an arbitrary policy and $\mu \in \mathcal{P}(\mathcal{S})$ an arbitrary initial state distribution. Denote by $\mathbb{P}_\mu^\pi$ the probability distribution that results from the interconnection of π and M while the initial state distribution is μ.

(a) Show there exists a Markov policy π' such that

$$\mathbb{P}_\mu^\pi(S_t = s, A_t = a) = \mathbb{P}_\mu^{\pi'}(S_t = a, A_t = a)\,.$$

for *all* $t \geq 1$ and $s, a \in \mathcal{S} \times \mathcal{A}$.
(b) Conclude that for any policy π there exists Markov policies π', π'' such that for any $s \in \mathcal{S}$, $\bar{\rho}_s^\pi = \bar{\rho}_s^{\pi'}$.

Define π' in an inductive manner by first considering $t = 1$, then $t = 2$ and so-on. Puterman [Theorem 5.5.1 2009] proves this result and credits Strauch [1966].

37.3 Let P be some transition structure over some finite state space $\mathcal{S}$ and some finite action space $\mathcal{A}$. Show that the expected travel time between two states s, s' of $\mathcal{S}$ is minimized by a deterministic policy.

Let $\tau^*(s, s')$ be the best expected travel time between some arbitrary pairs of states; for $s = s'$ we define the best travel time to be zero. Show that this satisfies the fixed point equation

$$\tau^*(s, s') = \begin{cases} 0, & \text{if } s = s'; \\ 1 + \min_a \sum_{s''} P_a(s, s'')\, \tau^*(s'', s'), & \text{otherwise}. \end{cases}$$

37.4 Let M be an MDP. Prove that $D(M) < \infty$ is equivalent to M being strongly connected.

37.5 Let $M = (\mathcal{S}, \mathcal{A}, P, r)$ be any MDP. Show that $D(M) \geq \log_{\mathrm{A}}(\mathrm{S}) - 3$.

Denote by $d^*(s, s')$ the minimum expected time it takes to reach state s' when starting from state s. The definition of d^* can be extended to arbitrary initial distributions μ_0 over states and sets $U \subset \mathcal{S}$ of target states: $d^*(\mu_0, U) = \sum_s \mu_0(s) \sum_{s' \in U} d^*(s, s')$. Prove by induction on the size of U that

$$d^*(\mu_0, U) \geq \min \left\{ \sum_{k \geq 0} k n_k \,|\, 0 \leq n_k \leq \mathrm{A}^k, k \geq 0, \sum_{k \geq 0} n_k = |U| \right\} \tag{37.25}$$

and then conclude that the proposition holds by choosing $U = \mathcal{S}$ [Auer et al., 2010, Cor. 15].

37.6 Let e_i be the ith element of the standard Euclidean basis and π be a memoryless policy. Show that $e_i^\top P_\pi^t e_j$ is the probability of arriving in state j from state i in t rounds using policy π.

37.7 Let M be a finite MDP and π a memoryless policy. Prove that for any $i \in \mathcal{S}$ the expected cumulative reward collected by policy π in M is $e_i^\top \sum_{t=1}^n P_\pi^t r_\pi$.

37.8 Let P be any $\mathrm{S} \times \mathrm{S}$ right stochastic matrix. Show that the following hold:

(a) $A_n = \frac{1}{n} \sum_{t=0}^{n-1} P^t$ is right stochastic.
(b) $A_n + \frac{1}{n}(P^n - I) = A_n P = P A_n$.
(c) $P^* = \lim_{n \to \infty} \frac{1}{n} \sum_{t=0}^{n-1} P^t$ exists.
(d) $P^* P = P P^* = P^* P^* = P^*$.
(e) The matrix $H = (I - P + P^*)^{-1}$ is well-defined.
(f) Let $D = H - P^*$. Then $D = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^n \sum_{k=0}^{i-1} (P^k - P^*)$.
(g) Let $r \in \mathbb{R}^{\mathrm{S}}$ and $\rho = P^* r$. Then $v = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^n \sum_{k=0}^{i-1} P^k (r - \rho)$ is well-defined and satisfies (37.3). (**Hint:** Show that $v = Dr$.)
(h) With the notation of the previous part, $v + \rho = r + Pv$.

Note that the first four parts of this exercise are the same as in Chapter 36. For Parts (c) and (d) you will likely find it useful that the space of right stochastic matrices is compact. Then show that all cluster points of (A_n) are the same.

The previous exercise implies that the gain and the differential value function of *any* memoryless policy in *any* MDP is well-defined. The matrix H is called the **fundamental matrix** and D is called the **deviation matrix**.

37.9 Let $\gamma \in (0,1)$ and define the operator $T_\gamma : \mathbb{R}^{\mathrm{S}} \to \mathbb{R}^{\mathrm{S}}$ by

$$(T_\gamma v)(s) = \max_{a \in \mathcal{A}} r_a(s) + \gamma \langle P_a(s), v \rangle \,.$$

(a) Prove that T_γ is a contraction with respect to the supremum norm:

$$\|T_\gamma v - T_\gamma w\|_\infty \le \gamma \|v - w\|_\infty \text{ for any } v, w \in \mathbb{R}^{\mathrm{S}} \,.$$

(b) Prove that there exists a $v \in \mathbb{R}^{\mathrm{S}}$ such that $T_\gamma v = v$.
(c) Let π be the greedy policy with respect to v. Show $v = r_\pi + \gamma P_\pi v$.
(d) Prove that $v = (I - \gamma P_\pi)^{-1} r$.
(e) Define the γ-discounted value function v_γ^π of a policy π as the function that for any given state $s \in \mathcal{S}$ gives the total expected discounted reward of the policy when it is started from state s. Let $v_\gamma^* \in \mathbb{R}^{\mathrm{S}}$ be defined by $v_\gamma^*(s) = \max_\pi v_\gamma^\pi(s)$, $s \in \mathcal{S}$. We call π γ-discount optimal if $v_\gamma^* = v_\gamma^\pi$. Show that if π is greedy with respect to v from Part (b) then π is a γ-optimal policy.

For (b) you should use the contraction mapping theorem (or Banach fixed point theorem), which says that if $(\mathcal{X}, d)$ is a complete metric space and $T : \mathcal{X} \to \mathcal{X}$ satisfies $d(T(x), T(y)) \le \gamma d(x, y)$ for $\gamma \in [0, 1)$, then there exists an $x \in \mathcal{X}$ such that $T(x) = x$. For (e) use (d) and Exercise 37.2 combined to show it suffices to check that $v_\gamma^\pi \le v$ for any Markov policy π. Verify this by using the fact that T_γ is monotone ($f \le g$ implies that $T_\gamma f \le T_\gamma g$) and showing that $v_{\gamma,n}^\pi \le T_\gamma^n \mathbf{0}$ holds for any n where $v_{\gamma,n}^\pi(s)$ is the the total expected discounted reward of the policy when it is started from state s and is followed for n steps.

37.10 Recall that $H = (I - P + P^*)^{-1}$, $D = H - P^*$ and let $P_\gamma^* = (1-\gamma)(I - \gamma P)^{-1}$. Show that

(a) $\lim_{\gamma \to 1-} P_\gamma^* = P^*$.
(b) $\lim_{\gamma \to 1-} \frac{P_\gamma^* - P^*}{1-\gamma} = D$.

For Item (a) start by manipulating the expressions $P_\gamma^* P$ and $(P_\gamma^*)^{-1} P^*$. For Item (b) consider $H^{-1}(P_\gamma^* - P^*)$.

Another way to prove Part (a) is by resorting the relation between Abel summability and Cesàro summability, mentioned beforehand in the notes. However, since the sequence $(P^t)_{t\geq 0}$ is quite special, there is a simpler direct proof. Note that the results of this exercise can be thought of providing information about the smooth continuation of the function $\gamma \mapsto P^*_\gamma$ defined over $[0,1)$ at $\gamma = 1$. In particular, Part (a) gives the value this function takes at $\gamma = 1$, while Part (b) gives the value the (left) derivative of this function takes at $\gamma = 1$. These can be seen as the first two terms of the Taylor-series expansion of this function. These of course also give the behavior of the unnormalized map $\gamma \mapsto P^*_\gamma/(1-\gamma)$, which has a singularity at $\gamma = 1$. For maps with singularities, the Taylor-series expansion is called the **Laurent series**, and thus in the literature the results of this exercise would often be connected to Laurent series.

37.11 In this exercise you will prove Part (a) of Theorem 37.1.

(a) Prove there exists a deterministic stationary policy π and monotone increasing sequence of discount rates (γ_n) with $\gamma_n < 1$ and $\lim_{n\to\infty} \gamma_n = 1$ such that π is a greedy policy with respect to the fixed point v_n of T_{γ_n} for all n.

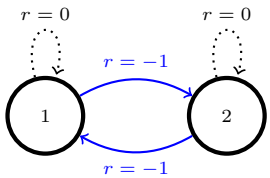
(b) For the remainder of the exercise, fix a policy π whose existence is guaranteed by Part (a). Show that $\rho^\pi = \rho \mathbf{1}$ is constant.

(c) Let $v = v_\pi$ be the value function and $\rho = \rho_\pi$ be the gain of policy π. Show that (ρ, v) satisfies the Bellman optimality equation.

For (a) use the fact that for finite MDPs there are only finitely many memoryless deterministic policies. For (b) and (c) use Exercise 37.10.

37.12 Consider the deterministic Markov decision process shown below with two states and two actions. The first action STAY keeps the state the same and the second action GO moves the learner to the other state while incurring a reward of negative one. Show that in this example solutions (ρ, v) to the Bellman optimality equations (Eq. (37.5)) are exactly the elements of the set

$$\{(\rho, v) \in \mathbb{R} \times \mathbb{R}^2 \,:\, \rho = 0,\, v(1) - 1 \leq v(2) \leq v(1) + 1\}\,.$$



37.13 Let M be a strongly connected MDP and (ρ, v) be a solution to the Bellman optimality equation. Show that $\mathrm{span}(v) \leq (\rho^* - \min_{s,a} r_a(s)) D(M)$.

Note that by Theorem 37.1, $\rho = \rho^*$. Fix some states $s_1 \neq s_2$ and a memoryless policy π. Show that

$$v(s_2) - v(s_1) \leq (\rho^* - \min_{s,a} r_a(s))\mathbb{E}^\pi[\tau_{s_2} \mid S_1 = s_1]\,.$$

Note for the sake of curiosity that the above display continues to hold for weakly communicating MDPs.

The proof of Theorem 4 in the paper by Bartlett and Tewari [2009] is incorrect, as is the sketch of the same result by Auer et al. [2010]. The problem is that the statement needs to hold for any solution v of the Bellman optimality equation. Both proofs use an argument that hinges on the fact that in an aperiodic strongly connected MDP, v is in the set $\{c\mathbf{1} + \lim_{n\to\infty} T^n\mathbf{0} - n\rho^* \,:\, c \in \mathbb{R}\}$. However, Exercise 37.12 shows that there are some MDPs with the required properties where this does not hold.

37.14 Solve the following problems:

(a) Prove that Algorithm 23 provides a separation oracle for convex set $\mathcal{K}$ defined in Eq. (37.10).
(b) Assuming that Algorithm 23 can be implemented efficiently, explain how to find an approximate solution to Eq. (37.7).

37.15 Consider a strongly connected MDP and suppose that ρ and v approximately satisfy the Bellman optimality equation in the sense that there exists an $\varepsilon > 0$ such that

$$\left|\rho + v(s) - \max_{a\in\mathcal{A}} r_a(s) + \langle P_a(s), v\rangle\right| \leq \varepsilon \qquad \text{for all state/action pairs } s, a\,.$$

Show that the following hold:

(a) Show that $\rho \geq \rho^* - \varepsilon$. Let $\tilde{\pi}$ be the greedy policy with respect to v.
(b) Assume that $\tilde{\pi}$ is ε'-greedy with respect to v in the sense that $r_{\tilde{\pi}(s)}(s) + \langle P_{\tilde{\pi}(s)}(s), v\rangle \geq \max_{a\in\mathcal{A}} r_a(s) + \langle P_a(s), v\rangle - \varepsilon'$ holds for all $s \in \mathcal{S}$. Show that $\tilde{\pi}$ is $2\varepsilon + \varepsilon'$ optimal: $\rho^{\tilde{\pi}} \geq \rho^* - (2\varepsilon + \varepsilon')$.

37.16 Let M be a strongly connected MDP with rewards in $[0, 1]$, diameter $D < \infty$ and optimal gain ρ^*. Let $v_n^*(s)$ be the maximum total expected reward in n steps when the process starts in state s. Prove that $v_n^*(s) \leq n\rho^* + D$.

37.17 Prove that (37.12) follows from Theorem 37.4.

37.18 The purpose of this exercise is to show that without phases UCRL2 may suffer linear regret. For convenience we consider the modified version of UCRL2 in Exercise 37.23 that does not know the reward. Now suppose we further modify

this algorithm to re-solve the optimistic MDP in every round ($\tau_k = k$ for all k). We make use of the following deterministic Markov decision process with two actions $\mathcal{A} = \{\text{STAY}, \text{GO}\}$ represented by dashed and solid arrows respectively.

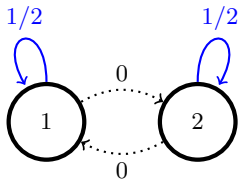


Figure 37.4 Transitions and rewards are deterministic. Numbers indicate the rewards.

(a) Find all memoryless optimal policies for the MDP in Fig. 37.4.
(b) Prove that the version of UCRL2 given in Exercise 37.23 modified to re-solve the optimistic MDP in every round suffers linear regret on this MDP.

Since UCRL2 and the environment are both deterministic you can examine the behavior of the algorithm on the MDP. You should aim to prove that eventually the algorithm will alternate between actions STAY and GO.

37.19 Let $\tilde{M}$ be the extended MPD defined in Section 37.5.2. Prove that $P \in \mathcal{C}$ implies that $\tilde{M}$ is strongly connected.

37.20 Prove Lemma 37.2.

Use the result of Exercise 5.19 and apply a union bound over all state/action pairs and the number of samples. Use the Markov property to argue that the independence assumption in Exercise 5.19 is not problematic.

LEMMA 37.3 *Let (a_k) and (A_k) be nonnegative numbers so that for any $k \geq 0$, $a_{k+1} \leq A_k = 1 \vee (a_1 + \cdots + a_k)$. Then for any $m \geq 1$,*

$$\sum_{k=1}^{m} \frac{a_k}{A_{k-1}} \leq \left(\sqrt{2}+1\right)\sqrt{A_m}\,.$$

37.21 Prove Lemma 37.3.

Fix $(a_k)_k$ and $(A_k)_k$. Consider some $m \geq 1$. The statement is trivial if $\sum_{k=1}^{m-1} a_k \leq 1$. If this does not hold, use induction based on $m = n, n+1, \ldots$ where n is the first integer such that $\sum_{k=1}^{n-1} a_k > 1$.

37.22 There are many variations of the end-of-phase definition in UCRL2 using which UCRL2 enjoys essentially the same regret.

(a) Which step of the proof of Theorem 37.4 would "break" (become linear in n) if we removed the phases from UCRL2 (more precisely, if UCRL2 would recompute the optimistic policy at the beginning of each round)? Does this explain
(b) Imagine that we modified UCRL2 to start a new phase after every $\sqrt{n}$ steps. Will this variant of UCRL2 enjoy a similar regret bound to the one stated in Theorem 37.4?
(c) Now imagine that UCRL2 is modified to start a new phase in rounds $\tau_{k+1} = \tau_k + f(k)$ with some function $f : \mathbb{N} \to \mathbb{N}$. Propose some choices of f under which the regret bound for the new variant stays essentially the same as it was before.
(d) Discuss the pros and cons of the different choices of the various phase definitions.

37.23 In this exercise you will modify the algorithm to handle the situation where r is unknown and rewards are stochastic. More precisely, assume there exists a function $r_a(s) \in [0,1]$ for all $a \in \mathcal{A}$ and $s \in \mathcal{S}$. Then in each round the learner observes S_t, chooses an action A_t and receives a reward $X_t \in [0,1]$ with

$$\mathbb{E}[X_t \mid A_t, S_t] = r_{A_t}(S_t)\,.$$

In order to accomodate the unknown reward function we modify UCRL2 in the following way. First define the empirical reward at the start of the kth phase by

$$\hat{r}_{k,a}(s) = \sum_{u=1}^{\tau_k - 1} \frac{\mathbb{I}\left\{S_u = s, A_u = a\right\} X_t}{1 \vee T_{\tau_k - 1}(s,a)}\,.$$

Then let $\tilde{r}_{t,a}(s)$ be an upper confidence bound given by

$$\tilde{r}_{k,a}(s) = \hat{r}_{t,a}(s) + \sqrt{\frac{L}{2(1 \vee T_{\tau_k - 1}(s,a))}}\,,$$

where L is as in the proof of Theorem 37.4. The modified algorithm operates exactly like Algorithm 24, but replaces the unknown $r_a(s)$ with $\tilde{r}_{k,a}(s)$ when solving the extended MDP. Prove that with probability at least $1 - 3\delta/2$ the modified policy in the modified setting has regret at most

$$\hat{R}_n \leq CD(M)\mathrm{S}\sqrt{n\mathrm{A}\log\left(\frac{n\mathrm{SA}}{\delta}\right)}\,,$$

where $C > 0$ is a universal constant.

37.24 In this exercise you will prove the claims to complete the proof of the lower bound.

(a) Prove Claim 37.1.

(b) Prove Claim 37.2.
(c) Prove Claim 37.3.

37.25 Consider the MDP $M = (\mathcal{S}, \mathcal{A}, P, r)$ where $P_a(s) = p$ for some fixed categorical distribution p for any $(s, a) \in \mathcal{S} \times \mathcal{A}$, where $\min_{s \in \mathcal{S}} p(s) > 0$. Assume that the rewards for action a in state s are sampled from a distribution supported on $[0, 1]$ (cf. note Item 3). An MDP like this defines nothing but a contextual bandit.

(a) Derive the optimal policy and the average optimal reward.
(b) Show an optimal value function that solves the Bellman optimality equation.
(c) Prove that the diameter of this MDP is $D = \max_s 1/p(s)$.
(d) Consider the algorithm that puts one instance of an appropriate version of UCB into every state (the same idea was explored in the context of adversarial bandits in Section 18.1). Prove that the expected regret of your algorithm will be at most $O(\sqrt{\mathrm{SA}n})$.
(e) Does the scaling behavior of the upper bound in Theorem 37.4 match the actual scaling behavior of the expected regret of UCRL2? Why or why not?
(f) Design and run an experiment to confirm your claim.

37.26 This is a thinking and coding exercise to illustrate the difficulty of learning in Markov decision processes. The RiverSwim environment is originally due to Strehl and Littman [2008]. The environment has two actions $\mathcal{A} = \{\text{LEFT}, \text{RIGHT}\}$ and $\mathcal{S} = [\mathrm{S}]$ with $\mathrm{S} \geq 2$. In all states $s > 1$, action LEFT deterministically leads to state $s - 1$ and provides no reward. In state 1, action LEFT leaves the state unchanged and yields a reward of 0.05. The action RIGHT tends to make the agent move right, but not deterministically (the learner is swimming against a current). With probability 0.3 the state is incremented, with probability 0.6 the state is left unchanged, while with probability of 0.1 the state is decremented. This actions incurs reward zero in all states except in state S where it receives a reward of 1. The situation when $\mathrm{S} = 5$ is illustrated in Fig. 37.5.

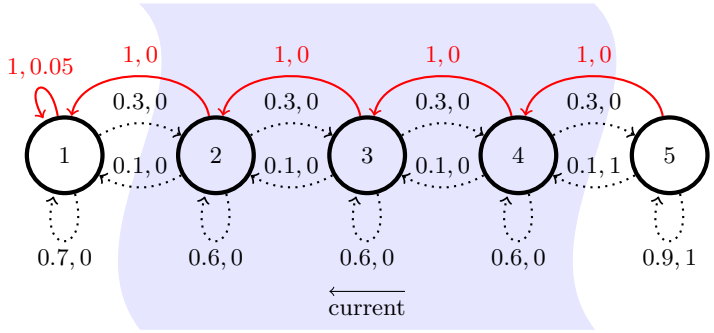


Figure 37.5 The RiverSwim MDP when $\mathrm{S} = 5$. Solid arrows correspond to action LEFT and dashed ones to action RIGHT. The right-hand bank is slippery, so the learner sometimes falls back into the river.

(a) Show that the optimal policy always takes action RIGHT and calculate the optimal average reward ρ^* as a function of S.
(b) Implement the MDP and test the optimal policy when started from state 1. Plot the total reward as a function of time and compare it with the plot of $t \mapsto t\rho^*$. Run multiple simulations to produce error bars. How fast do you think the total reward concentrates around $t\rho^*$? Experiment with different values of S.
(c) The ε-greedy strategy can also be implemented in MDPs as follows: Based on the data previously collected estimate the transition probabilities and rewards using empirical means. Find the optimal policy π^* of the resulting MDP and if the current state is s, use the action $\pi^*(s)$ with probability $1-\varepsilon$ and choose one of the two actions uniformly at random with the remaining probability. To ensure the empirical MDP has a well-defined optimal policy, mix the empirical estimate of the next state distributions $P_a(s)$ with the uniform distribution with a small mixture coefficient. Implement this strategy and plot the trajectories it exhibits for various MDP sizes. Explain what you see.
(d) Implement UCRL2 and produce the same plots. Can you explain what you see?
(e) Run simulations in RiverSwim instances of various sizes to compare the regret of UCRL2 and ε-greedy. What do you conclude?

37.27 Fix state-space $\mathcal{S}$, action-space $\mathcal{A}$ and reward function r. Let π be a policy with sublinear regret in all strongly connected MDPs $(\mathcal{S}, \mathcal{A}, r, P)$. Now suppose that $(\mathcal{S}, \mathcal{A}, r, P)$ is an MDP that is not strongly connected such that for all $s \in \mathcal{S}$ there exists state s' such is reachable from s under some policy and where $\rho^*_{s'} < \max_u \rho^*_u$. Finally, assume that $\rho^*_{S_1} = \max_u \rho^*_u$ almost surely. Prove that π has linear regret on this MDP.

37.28 This exercise develops the ideas mentioned in Note 14. First, we need some definitions: Fix $\mathcal{S}$ and $\mathcal{A}$ and define Π_0 as the set of policies (learner strategies) for MDPs with state space $\mathcal{S}$ and action space $\mathcal{A}$ that achieve sublinear regret in any strongly connected MDP with state space $\mathcal{S}$ and action space $\mathcal{A}$. Now consider an arbitrary finite MDP $M = (\mathcal{S}, \mathcal{A}, P, r)$ that has $\mathcal{S}$ as state space and $\mathcal{A}$ as action space. A state $s \in \mathcal{S}$ is reachable from state $s' \in \mathcal{S}$ if there is a policy that, when started in s' reaches state s with positive probability after *one* or more steps. A set of states $C \subset \mathcal{S}$ is **strongly-connected component** (SCC) if every state $s \in U$ is reachable from every other state $s' \in C$ (allowing for the possibility that $s = s'$). Call C **maximal** if we cannot add more states to C and still maintain the SCC property. A maximal SCC is called a **maximal end-component** (MEC). Show the following:

(a) Two MECs C_1 and C_2 are either equal, or disjoint.
(b) Let $C_1, \dots, C_k$ be all the distinct MECs of an MDP. The MDP structure defines a connectivity over $C_1, \dots, C_k$ as follows: For $i \neq j$, we say that C_i is

connected to C_j if from some state in C_i it is possible to reach some state of C_j with positive probability under some policy. Show that this connectivity structure defines a directed graph, which must be acyclic.

(c) Let $C_1, \dots, C_m$ with $m \le k$ be the sinks (the nodes with no out-edges) of this graph. Show that if M is strongly connected then $m = 1$ and $C_1 = \mathcal{S}$.

(d) Show that for any $i \in [m]$ and for any policy $\pi \in \Pi_0$ it holds that π will reach C_i in finite time with positive probability if the initial state distribution assigns positive mass to the non-trap states $\mathcal{S} \setminus \cup_{i \in [m]} C_i$.

(e) Show that for $i \le m$, for any $s \in C_i$ and any action $a \in \mathcal{A}$, $P_a(s, s') = 0$ for any $s' \in \mathcal{S} \setminus C_i$, i.e., C_i is **closed**.

(f) Show that the restriction of M to C_i defined as

$$M_i = (C_i, \mathcal{A}, (P_a(s))_{s \in C_i, a \in \mathcal{A}}, (r_a(s))_{s \in C_i, a \in \mathcal{A}})$$

is an MDP.

(g) Show that M_i is strongly connected.

(h) Let τ be the time when the learner enters one of $C_1, \dots, C_m$ and let $I \in [m]$ be the index of the class that is entered at time τ. That is, $S_\tau \in C_I$. Show that if M is strongly connected then $\tau = 1$ with probability one.

(i) We redefine the regret as follows: $R'_n = \mathbb{E}\left[\sum_{t=\tau}^{\tau+n-1} r_{A_t}(S_t) - n\rho^*(M_I)\right]$. Show that if M is strongly connected then $R_n = R'_n$.

(j) Show that there exist a learner such that (37.12) continues to hold in the sense that $R'_n \le 1 + C\mathbb{E}\left[D(M_I)|C_I|\sqrt{2\mathrm{A}n\log(n)}\right]$.

The logic of the regret definition in Part (i) is that by Part (d), reasonable policies cannot control which trap they fall into in an MDP that has more than one traps. As such, policies should not be penalized for what trap they fall into. However, once a policy falls into some "trap", we expect it to start to behave near optimally. What this definition is still lacking is that it is insensitive to how fast a policy gets trapped.

37.29 Prove the claim in Eq. (37.22).

Make use of the result in Exercise 14.9.

37.30 Given an example of a Markov decision process and a solution (ρ, v) to the linear program in Eq. (37.6) such that v does not satisfy the Bellman optimality equation and the greedy policy with respect to v is not optimal.

37.31 Let $\mathcal{K} \subset \mathbb{R}^d$ be a convex set and ϕ be a separation oracle for $\mathcal{K}$. Suppose that $a_1, \dots, a_n$ is a collection of vectors with $a_k \in \mathbb{R}^d$ and $b_1, \dots, b_k$ be a collection of scalars. Let $H_k = \{x \in \mathbb{R}^d : \langle a_k, x \rangle \ge b_k\}$. Devise an efficient separation oracle for $\bigcap_{k=1}^n \mathcal{K} \cap H_k$.