1 Markov Decision Processes

In reinforcement learning, the interactions between the agent and the environment are often described by a Markov Decision Process (MDP) \cite{1}, specified by:

- **State space** \( \mathcal{S} \). In this course we only consider finite state spaces.
- **Action space** \( \mathcal{A} \). In this course we only consider finite action spaces.
- **Transition function** \( P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S}) \), where \( \Delta(\mathcal{S}) \) is the space of probability distributions over \( \mathcal{S} \) (i.e., the probability simplex). \( P(s'|s,a) \) is the probability of transitioning into state \( s' \) upon taking action \( a \) in state \( s \).
- **Reward function** \( R : \mathcal{S} \times \mathcal{A} \to [0,R_{\text{max}}] \), where \( R_{\text{max}} > 0 \) is a constant. \( R(s,a) \) is the immediate reward associated with taking action \( a \) in state \( s \).
- **Discount factor** \( \gamma \in [0,1) \), which defines a horizon for the problem.

1.1 Interaction protocol

In a given MDP \( M = (\mathcal{S}, \mathcal{A}, P, R, \gamma) \), the agent interacts with the environment according to the following protocol: the agent starts at some state \( s_1 \); at each time step \( t = 1, 2, \ldots \), the agent takes an action \( a_t \in \mathcal{A} \), obtains the immediate reward \( r_t = R(s_t, a_t) \), and observes the next state \( s_{t+1} \) sampled from \( P(s_t, a_t) \), or \( s_{t+1} \sim P(s_t, a_t) \). The interaction record

\[
\tau = (s_1, a_1, r_1, s_2, \ldots, s_{H+1})
\]

is called a trajectory of length \( H \).

In some situations, it is necessary to specify how the initial state \( s_1 \) is generated. We consider \( s_1 \) sampled from an initial distribution \( \mu \in \Delta(\mathcal{S}) \). When \( \mu \) is of importance to the discussion, we include it as part of the MDP definition, and write \( M = (\mathcal{S}, \mathcal{A}, P, R, \gamma, \mu) \).

1.2 Policy and value

A (deterministic and stationary) policy \( \pi : \mathcal{S} \to \mathcal{A} \) specifies a decision-making strategy in which the agent chooses actions adaptively based on the current state, i.e., \( a_t = \pi(s_t) \). More generally, the agent
may also choose actions according to a stochastic policy \( \pi : S \rightarrow \Delta(A) \), and with a slight abuse of notation we write \( a_t \sim \pi(s_t) \). A deterministic policy is its special case when \( \pi(s) \) is a point mass for all \( s \in S \).

The goal of the agent is to choose a policy \( \pi \) to maximize the expected discounted sum of rewards, or value:

\[
E \left[ \sum_{t=1}^{\infty} \gamma^{t-1} r_t \mid \pi, s_1 \right].
\] (1)

The expectation is with respect to the randomness of the trajectory, that is, the randomness in state transitions and the stochasticity of \( \pi \). Notice that, since \( r_t \) is nonnegative and upper bounded by \( R_{\text{max}} \), we have

\[
0 \leq \sum_{t=1}^{\infty} \gamma^{t-1} r_t \leq \sum_{t=1}^{\infty} \gamma^{t-1} R_{\text{max}} = \frac{R_{\text{max}}}{1 - \gamma}.
\] (2)

Hence, the discounted sum of rewards (or the discounted return) along any actual trajectory is always bounded in range \([0, R_{\text{max}}/(1 - \gamma)]\), and so is its expectation of any form. This fact will be important when we later analyze the error propagation of planning and learning algorithms.

Note that for a fixed policy, its value may differ for different choice of \( s_1 \), and we define the value function \( V_{\pi}^M : S \rightarrow \mathbb{R} \) as

\[
V_{\pi}^M(s) = E \left[ \sum_{t=1}^{\infty} \gamma^{t-1} r_t \mid \pi, s_1 = s \right],
\]

which is the value obtained by following policy \( \pi \) starting at state \( s \). Similarly we define the action-value (or Q-value) function \( Q_{\pi}^M : S \times A \rightarrow \mathbb{R} \) as

\[
Q_{\pi}^M(s, a) = E \left[ \sum_{t=1}^{\infty} \gamma^{t-1} r_t \mid \pi, s_1 = s, a_1 = a \right].
\]

Henceforth, the dependence of any notation on \( M \) will be made implicit whenever it is clear from context.

### 1.3 Bellman equations for policy evaluation

Based on the principles of dynamic programming, \( V^\pi \) and \( Q^\pi \) can be computed using the following Bellman equations for policy evaluation: \( \forall s \in S, a \in A \),

\[
V^\pi(s) = Q^\pi(s, \pi(s)).
\]

\[
Q^\pi(s, a) = R(s, a) + \gamma E_{s' \sim P(s, a)} \left[ V^\pi(s') \right].
\] (3)

In \( Q^\pi(s, \pi(s)) \) we treat \( \pi \) as a deterministic policy for brevity, and for stochastic policies this shorthand should be interpreted as \( E_{a \sim \pi(s)}[Q^\pi(s, a)] \).

Since \( S \) is assumed to be finite, upon fixing an arbitrary order of states (and actions), we can treat \( V^\pi \) and any distribution over \( S \) as vectors in \( \mathbb{R}^{|S|} \), and \( R \) and \( Q^\pi \) as vectors in \( \mathbb{R}^{|S \times A|} \). This is particularly helpful as we can rewrite Equation 3 in an matrix-vector form and derive an analytical solution for \( V^\pi \) using linear algebra as below.
Define $P^\pi$ as the transition matrix for policy $\pi$ with dimension $|S| \times |S|$, whose $(s, s')$-th entry is
\[
[P^\pi]_{s,s'} = \mathbb{E}_{a \sim \pi(s)}[P(s', a)],
\]
In fact, this matrix describes a Markov chain induced by MDP $M$ and policy $\pi$. Its $s$-th row is the distribution over next-states upon taking actions according to $\pi$ at state $s$, which we also write as $[P(s, \pi)]^\top$.

Similarly define $R^\pi$ as the reward vector for policy $\pi$ with dimension $|S| \times 1$, whose $s$-th entry is
\[
[R^\pi]_s = \mathbb{E}_{a \sim \pi(s)}[R(s, a)].
\]

Then from Equation 3 we have
\[
[V^\pi]_s = Q^\pi(s, \pi(s)) = [R^\pi]_s + \gamma \mathbb{E}_{a \sim \pi(s)} \mathbb{E}_{s' \sim P(s, a)} [V^\pi(s')]
= [R^\pi]_s + \gamma \mathbb{E}_{s' \sim P(s, \pi)} [V^\pi(s')]
= [R^\pi]_s + \gamma \langle P(s, \pi), V^\pi \rangle,
\]
where $\langle \cdot, \cdot \rangle$ is dot product. Since this equation holds for every $s \in S$, we have
\[
V^\pi = R^\pi + \gamma P^\pi V^\pi \quad \Rightarrow \quad (I_{|S|} - \gamma P^\pi)^{-1} R^\pi = V^\pi,
\]
where $I_{|S|}$ is the identity matrix. Now we notice that matrix $(I_{|S|} - \gamma P^\pi)^{-1}$ is always invertible. In fact, for any non-zero vector $x \in \mathbb{R}^{|S|}$,
\[
\| (I_{|S|} - \gamma P^\pi) x \|_\infty = \| x - \gamma P^\pi x \|_\infty \\
\geq \| x \|_\infty - \gamma \| P^\pi x \|_\infty \\
\geq \| x \|_\infty - \gamma \| x \|_\infty \\
= (1 - \gamma) \| x \|_\infty > 0
\]
(Each element of $P^\pi x$ is a convex average of $x$)

So we can conclude that
\[
V^\pi = (I_{|S|} - \gamma P^\pi)^{-1} R^\pi.
\]

State occupancy
When the reward function only depends on the current state, i.e., $R(s, a) = R(s)$, $R^\pi$ is independent of $\pi$, and Equation 4 exhibits an interesting structure: implies that the value of a policy is linear in rewards, and the rows of the matrix $(I_{|S|} - \gamma P^\pi)^{-1}$ give the linear coefficients that depend on the initial state. Such coefficients, often represented as a vector, are called discounted state occupancy (or state occupancy for short). It can be interpreted as the expected number of times that each state is visited along a trajectory, where later visits are discounted more heavily.

\[1\text{When rewards depend on actions, we can define discounted state-action occupancy in a similar way and recover the fact that value is linear in reward.} \]
1.4 Bellman optimality equations

There always exists a stationary and deterministic policy that simultaneously maximizes $V^\pi(s)$ for all $s \in S$ and maximizes $Q^\pi(s,a)$ for all $s \in S, a \in A$ [1], and we denote this optimal policy as $\pi^*_M$ (or $\pi^*$). We use $V^*$ as a shorthand for $V^\pi^*$, and $Q^*$ similarly.

$V^*$ and $Q^*$ satisfy the following set of Bellman optimality equations [2]:

\begin{align}
    V^*(s) &= \max_{a \in A} Q^*(s,a), \\
    Q^*(s,a) &= R(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a)}[V^*(s')].
\end{align}

Once we have $Q^*$, we can obtain $\pi^*$ by choosing actions greedily (with arbitrary tie-breaking mechanisms):

$$\pi^*(s) = \arg \max_{a \in A} Q^*(s,a), \ \forall s \in S.$$ 

We use shorthand $\pi_Q$ to denote the procedure of turning a Q-value function into its greedy policy, and the above equation can be written as

$$\pi^* = \pi_Q.$$

To facilitate future discussions, define the Bellman optimality operator $T_M : \mathbb{R}^{|S \times A|} \rightarrow \mathbb{R}^{|S \times A|}$ (or simply $T$) as follows: when applied to some vector $f \in \mathbb{R}^{|S \times A|},$

\begin{align}
    (Tf)(s,a) &= R(s,a) + \gamma \langle P(s,a), V_f \rangle,
\end{align}

where $V_f(\cdot) := \max_{a \in A} f(\cdot,a)$. This allows us to rewrite Equation [5] in the following concise form, which implies that $Q^*$ is the fixed point of the operator $T$:

$$Q^* = TQ^*.$$ 

1.5 Notes on the MDP setup

Before moving on, we make notes on our setup of MDP and discuss alternative setups considered in the literature.

Finite horizon and episodic setting

Our definition of value (Equation [1]) corresponds to the infinite-horizon discounted setting of MDPs. Popular alternative choices include the finite-horizon undiscounted setting (actual return of a trajectory is $\sum_{t=1}^{H} r_t$ with some finite horizon $H < \infty$) and the infinite-horizon average reward setting (return is $\lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} r_t$). The latter case often requires additional conditions on the transition dynamics (such as ergodicity) so that values can be well-defined [3], and will not be discussed in this course.

The finite-horizon undiscounted (or simply finite-horizon) setting can be emulated using the discounted setting by augmenting the state space. Suppose we have an MDP $M$ with finite horizon $H$. Define a new MDP $M' = (\tilde{S}, \tilde{A}, \tilde{P}, \tilde{R}, \gamma)$ such that $\tilde{S} = S \times \{H\} \cup \{s_{\text{absorbing}}\}$ ($\{H\} = \{1, \ldots, H\}$). Essentially we make $H$ copies of the state space and organize them in levels, with an additional absorbing
state \( s_{\text{absorbing}} \) where all actions transition to itself and yield 0 reward. There is only non-zero transition probability from states at level \( h \) to states at level \( h + 1 \) with \( \tilde{P}((s', h + 1) \mid (s, h), a) = P(s' \mid s, a) \), and states at the last level \((s, H)\) transition to \( s_{\text{absorbing}} \) deterministically. Finally we let \( \tilde{R}((s, h), a) = R(s, a) \) and \( \gamma = 1 \). (In general \( \gamma = 1 \) may lead to infinite value, but here the agent always loops in the absorbing state after \( H \) steps and gets finite total rewards.) The optimal policy for finite-horizon MDPs is generally non-stationary, that is, it depends on both \( s \) and the time step \( h \).

The MDP described in the construction above can be viewed as an example of episodic tasks: the environment deterministically transitions into an absorbing state after a fixed number of time steps. The absorbing state often corresponds to the notion of termination, and many problems are naturally modeled using an episodic formulation, including board games (a game terminates once the winner is determined) and dialog systems (a session terminates when the conversation is concluded).

**Stochastic rewards**

Our setup assumes that reward \( r_t \) only depends on \( s_t \) and \( a_t \) deterministically. In general, \( r_t \) may also depend on \( s_{t+1} \) and contain additional noise that is independent from state transitions as well as reward noise in other time steps. As special cases, in inverse RL literature [4, 5], reward only depends on state, and in contextual bandit literature [6], reward depends on the state (or context in bandit terminologies) and action but has additional independent noise.

All these setups are equivalent to having a state-action reward with regard to the policy values: define \( \tilde{R}(s, a) = \mathbb{E}[r_t \mid s_t = s, a_t = a] \) where \( s_{t+1} \) and the independent noise are marginalized out. The value functions \( V^\pi \) and \( Q^\pi \) for any \( \pi \) remains the same when we substitute in this equivalent reward function. That said, reward randomness may introduce additional noise in the sample trajectories and affect learning efficiency.

**Negative rewards**

Our setup assumes that \( r_t \in [0, R_{\text{max}}] \). This is without loss of generality in the infinite-horizon discounted setting: for any constant \( c > 0 \), a reward function \( R \in \mathbb{R}^{\mathcal{S} \times \mathcal{A}} \) is equivalent to \( R + c \mathbf{1}_{\mathcal{S} \times \mathcal{A}} \), as adding \( c \) units of reward to each state-action pair simply adds a constant “background” value of \( c/(1 - \gamma) \) to the value of all policies for all initial states. Therefore, when the rewards may be negative but still have bounded range, e.g., \( R(s, a) \in [-a, b] \) with \( a, b > 0 \), we can add a constant offset \( c = a \) to the reward function and define \( R_{\text{max}} = a + b \), so that after adding the offset the reward lies in \([0, R_{\text{max}}]\).

### 2 Planning in MDPs

*Planning* refers to the problem of computing \( \pi^*_M \) given the MDP specification \( M = (\mathcal{S}, \mathcal{A}, P, R, \gamma) \). This section reviews classical planning algorithms that compute \( Q^* \).
2.1 Policy Iteration

The policy iteration algorithm starts from an arbitrary policy $\pi_0$, and repeat the following iterative procedure: for $k = 1, 2, \ldots$

$$\pi_k = \pi Q^{\pi_{k-1}}.$$  

Essentially, in each iteration we compute the Q-value function of $\pi_{k-1}$ (e.g., using the analytical form given in Equation 4), and then compute the greedy policy for the next iteration. The first step is often called policy evaluation, and the second step is often called policy improvement.

The policy value is guaranteed to improve monotonically over all states until $\pi^*$ is found [1].

Theorem 1 (Policy improvement theorem). In policy iteration, $V^{\pi_k}(s) \geq V^{\pi_{k-1}}(s)$ holds for all $k \geq 1$ and $s \in \mathcal{S}, a \in \mathcal{A}$, and the improvement is strictly positive in at least one state until $\pi^*$ is found.

Therefore, the termination criterion for policy iteration is $Q^{\pi_k} = Q^{\pi_{k-1}}$. Since we are only searching over stationary and deterministic policies, and a new policy that is different from all previous ones is found every iteration, the algorithm is guaranteed to terminate in $|\mathcal{A}| |\mathcal{S}|$ iterations.

To prove the policy improvement theorem, we introduce an important concept called advantage.

Definition 1 (Advantage). The advantage of action $a$ at state $s$ over policy $\pi$ is defined as $A^{\pi}(s, a) := Q^{\pi}(s, a) - V^{\pi}(s)$. The advantage of policy $\pi'$ over policy $\pi$ is defined as $A^{\pi}(s, \pi') := A^{\pi}(s, \pi'(s))$.

Since policy iteration always takes the greedy policy of the current policy’s Q-value function, by definition the advantage of the new policy over the old one is non-negative. The next result shows that the value difference between two policies can be expressed using the advantage function. The policy improvement theorem immediately follows, since $V^{\pi_k}(s) - V^{\pi_{k-1}}(s)$ can be decomposed into the sum of nonnegative terms.

Proposition 2 (Advantage decomposition of policy values). For any $\pi, \pi'$, and any state $s \in \mathcal{S}$,

$$V^{\pi'}(s) - V^{\pi}(s) = \frac{1}{1 - \gamma} E_{s' \sim \eta^{\pi'}_s} [A^{\pi}(s', \pi')],$$

where $\eta^{\pi'}_s$ is the normalized discounted occupancy induced by policy $\pi'$ from starting state $s$.

Proof. Consider a sequence of (potentially non-stationary) policies $\{\pi_i\}_{i \geq 0}$, where $\pi_0 = \pi$, $\pi_\infty = \pi'$. For any intermediate $i$, $\pi_i$ is the non-stationary policy that follows $\pi'$ for the first $i$ time-steps and switches to $\pi$ for the remainder of the trajectory. Now we can rewrite the LHS of the statement as:

$$V^{\pi'}(s) - V^{\pi}(s) = \sum_{i=0}^{\infty} (V^{\pi_{i+1}}(s) - V^{\pi_i}(s)).$$

For each term on the RHS, observe that $\pi_i$ and $\pi_{i+1}$ share the same “roll-in” policy $\pi'$ for the first $i$ steps, which defines a roll-in distribution $P[s_{i+1}|s_1 = s, \pi']$. They also share the same “roll-out” policy $\pi$ starting from the $(i+2)$-th time step, so conditioned on $s_{i+1} = s, a_{i+1} = a$, the total expected reward picked up in the remainder of the trajectory is $\gamma^i Q^{\pi}(s, a)$ for both $\pi$ and $\pi_{i+1}$. Putting together, we
have

\[ V^{\pi'}(s) - V^\pi(s) = \sum_{i=0}^{\infty} \gamma^i \sum_{s' \in S} \mathbb{P}[s_{i+1} = s'|s_1 = s, \pi'] (Q^\pi(s', \pi'(s')) - Q^\pi(s', \pi(s'))) \]

\[ = \sum_{i=0}^{\infty} \gamma^i \sum_{s' \in S} \mathbb{P}[s_{i+1} = s'|s_1 = s, \pi'] A^\pi(s', \pi'). \]

The result follows by noticing that \( \sum_{i=0}^{\infty} \gamma^i \mathbb{P}[s_{i+1} = s'|s_1 = s, \pi'] = \frac{1}{1-\gamma} \eta^\pi(s'). \)

Policy iteration usually converges very fast in practice (for tabular problems), however the theoretical property is not completely clear; we know that the number of iterations is upper bounded by \(|A||S|\), and for certain variants of the algorithm such exponential computational complexity can occur in the worst case. However, what we usually want is an approximate solution, and we can show that policy iteration also enjoys exponential convergence [see e.g.], which is not well known.

**Theorem 3** (Policy iteration enjoys exponential convergence). \( \|Q^* - Q^\pi_{k+1}\|_\infty \leq \gamma \|Q^* - Q^\pi_k\|_\infty. \)

**Proof.** We will use two facts: (a) \( \sum_{i=0}^{\infty} \gamma^i \mathbb{P}[s_{i+1} = s'|s_1 = s, \pi'] \mathbb{E}(|T^\pi|) \leq \gamma \|Q^* - Q^\pi_k\|_\infty \), (b) \( \sum_{i=0}^{\infty} \gamma^i \mathbb{P}[s_{i+1} = s'|s_1 = s, \pi'] A^\pi(s', \pi') \mathbb{E}(|A^\pi|) \leq \gamma \|Q^* - Q^\pi_k\|_\infty \).

The first step just adds and subtracts the same quantity. The second step applies (a) and (b) to the two parentheses, respectively. Now

\[ \frac{\|Q^* - Q^\pi_{k+1}\|_\infty}{\|Q^* - Q^\pi_k\|_\infty} \leq \frac{\|T^\pi Q^* - T^\pi Q^\pi_k\|_\infty}{\|Q^* - Q^\pi_k\|_\infty} \leq \gamma \|Q^* - Q^\pi_k\|_\infty. \]

Finally we verify (a) and (b) by noting that

\[ (T^\pi Q^\pi_k)(s, a) = \mathbb{E} \left[ \sum_{h=1}^{\infty} \gamma^{h-1} r_h | s_1 = s, a_1 = a, a_2 \sim \pi_{k+1}, a_{3:\infty} \sim \pi_k \right], \]

\[ (T^\pi Q^\pi_k)(s, a) = \mathbb{E} \left[ \sum_{h=1}^{\infty} \gamma^{h-1} r_h | s_1 = s, a_1 = a, a_2 \sim \pi, a_{3:\infty} \sim \pi_k \right], \]

\[ Q^\pi_{k+1}(s, a) = \mathbb{E} \left[ \sum_{h=1}^{\infty} \gamma^{h-1} r_h | s_1 = s, a_1 = a, a_2 \sim \pi_{k+1}, a_{3:\infty} \sim \pi_{k+1} \right]. \]

2.2 Value Iteration

Value Iteration computes a series of Q-value functions to directly approximate \( Q^* \), without going back and forth between value functions and policies as in Policy Iteration. Let \( Q^{*0} \) be the initial
value function, often initialized to $0_{|S \times A|}$. The algorithm computes $Q^{*,h}$ for $h = 1, 2, \ldots, H$ in the following manner:

$$Q^{*,h} = \mathcal{T} Q^{*,h-1}.$$  \hfill (10)

Recall that $\mathcal{T}$ is the Bellman optimality operator defined in Equation 6.

We provide two different interpretations to understand the behavior of the algorithm. Both interpretations will lead to the same bound on $\|Q^{*,H} - Q^*\|_\infty$ as a function of $H$. If $H$ is large enough, we can guarantee that $Q^{*,H}$ is sufficiently close to $Q^*$, and the following result bounds the suboptimality (or loss) of acting greedily with respect to an approximate Q-value function, $f$:

**Lemma 4 ([9]).** $\|V^* - V^{\pi_f}\|_\infty \leq 2\|f - Q^*\|_\infty / (1 - \gamma)$.

**Proof.** For any $s \in S$,

$$V^*(s) - V^{\pi_f}(s) = Q^*(s, \pi^*(s)) - Q^*(s, \pi_f(s)) + Q^*(s, \pi_f(s)) - Q^{\pi_f}(s, \pi_f(s)) \leq Q^*(s, \pi^*(s)) - f(s, \pi^*(s)) + f(s, \pi_f(s)) - Q^*(s, \pi_f(s)) \leq 2\|f - Q^*\|_\infty + \gamma\|V^* - V^{\pi_f}\|_\infty.$$

Bounding $\|Q^{*,H} - Q^*\|_\infty$: the fixed point interpretation

Value Iteration can be viewed as solving for the fixed point of $\mathcal{T}$, i.e., $Q^* = \mathcal{T}Q^*$. The convergence of such iterative methods is typically analyzed by examining the contraction of the operator. In fact, the Bellman optimality operator is a $\gamma$-contraction under $\ell_\infty$ norm [1]: for any $f, f' \in \mathbb{R}^{|S \times A|}$

$$\|\mathcal{T}f - \mathcal{T}f'\|_\infty \leq \gamma \|f - f'\|_\infty.$$

(11)

To verify, we expand the definition of $\mathcal{T}$ for each entry of $(\mathcal{T}f - \mathcal{T}f')$:

$$\|\mathcal{T}f - \mathcal{T}f'\|_{s,a} = |R(s,a) + \gamma E(P(s,a)V_f) - R(s,a) - \gamma E(P(s,a)V_{f'})| \leq \gamma \|V_f - V_{f'}\|_\infty \leq \gamma \|f - f'\|_\infty.$$

The last step uses the fact that $\forall s \in S, |V_f(s) - V_{f'}(s)| = \max_{a \in A} |f(s,a) - f'(s,a)|$. The easiest way to see this is to assume $V_f(s) > V_{f'}(s)$ (the other direction is symmetric), and let $a_0$ be the greedy action for $f$ at $s$. Then

$$|V_f(s) - V_{f'}(s)| = f(s, a_0) - \max_{a \in A} f'(s, a) \leq f(s, a_0) - f'(s, a_0) \leq \max_{a \in A} |f(s, a) - f'(s, a)|.$$

Using the contraction property of $\mathcal{T}$, we can show that as $h$ increases, $Q^*$ and $Q^{*,h}$ becomes exponentially closer under $\ell_\infty$ norm:

$$\|Q^{*,h} - Q^*\|_\infty = \|\mathcal{T}Q^{*,h-1} - \mathcal{T}Q^*\|_\infty \leq \gamma \|Q^{*,h-1} - Q^*\|_\infty.$$

Since $Q^*$ has bounded range (recall Equation 2), for $Q^{*,0} = 0_{|S \times A|}$ (or any function in the same range) we have $\|Q^{*,0} - Q^*\|_\infty \leq R_{\text{max}}/(1 - \gamma)$. After $H$ iterations, the distance shrinks to

$$\|Q^{*,H} - Q^*\|_\infty \leq \gamma^H R_{\text{max}}/(1 - \gamma).$$

(12)
To guarantee that we compute a value function \( \epsilon \)-close to \( Q^\star \), it is sufficient to set

\[
H \geq \frac{\log \left( \frac{R_{\max}}{(1 - \gamma) \epsilon} \right)}{1 - \gamma}.
\]  

(13)

The base of \( \log \) is \( e \) in this course unless specified otherwise. To verify,

\[
\gamma^H \frac{R_{\max}}{1 - \gamma} = (1 - (1 - \gamma))^{\frac{1}{1 - \gamma}} \leq \left( \frac{1}{e} \right)^{\log \left( \frac{R_{\max}}{(1 - \gamma) \epsilon} \right)} \frac{R_{\max}}{1 - \gamma} = \epsilon.
\]

Here we used the fact that \((1 - 1/x)^x \leq 1/e\) for \( x > 1 \).

Equation (13) is often referred to as the effective horizon. The bound is often simplified as \( H = O(\frac{1}{1 - \gamma}) \), and used as a rule of thumb to translate between the finite-horizon undiscounted and the infinite-horizon discounted settings. From now on we will often use the term “horizon” generically, which should be interpreted as \( O(\frac{1}{1 - \gamma}) \) in the discounted setting.

**Bounding \( \|Q^* H - Q^*\|_\infty \): the finite-horizon interpretation**

Equation (12) can be derived using an alternative argument, which views Value Iteration as optimizing value for a finite horizon. \( V^{* H}(s) \) is essentially the optimal value for the expected value of the finite-horizon return: \( \sum_{t=1}^{H} \gamma^{t-1} r_t \). For any stationary policy \( \pi \), define its \( H \)-step truncated value

\[
V^{\pi, H}(s) = \mathbb{E} \left[ \sum_{t=1}^{H} \gamma^{t-1} r_t \mid \pi, s_1 = s \right].
\]  

(14)

Due to the optimality of \( V^{* H} \), we can conclude that for any \( s \in S \) and \( \pi : S \to A \), \( V^{\pi, H}(s) \leq V^{* H}(s) \). In particular,

\[
V^{\pi^*, H}(s) \leq V^{* H}(s).
\]

Note that the LHS and RHS are not to be confused: \( \pi^* \) is the stationary policy that is optimal for infinite horizon, and to achieve the finite-horizon optimal value on the RHS we may need a non-stationary policy (recall the discussion in Section 1.5).

The LHS can be lower bounded as \( V^{\pi^*, H}(s) \geq V^*(s) - \gamma^H R_{\max}/(1 - \gamma) \), because \( V^{\pi^*, H} \) does not include the nonnegative rewards from time step \( H + 1 \) on. (In fact the same bound applies to all policies.) The RHS can be upper bounded as \( V^{* H}(s) \leq V^*(s) \): \( V^* \) should dominate any stationary and non-stationary policies, including the one that first achieves \( V^{* H} \) within \( H \) steps and picks up some non-negative rewards afterwards with any behavior. Combining the lower and the upper bounds, we have \( \forall s \in S \),

\[
V^*(s) - \gamma^H \frac{R_{\max}}{1 - \gamma} \leq V^{* H}(s) \leq V^*(s),
\]

which immediately leads to Equation (12).

**References**


---

2The logarithmic dependence on \( 1/(1 - \gamma) \) is ignored as it is due to the magnitude of the value function.


