MDP Preliminaries

Nan Jiang

Oct. 5, 2019; updated: Sept. 27, 2024

1 Markov Decision Processes

In reinforcement learning, the interactions between the agent and the environment are often described by a Markov Decision Process (MDP) [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 $R : \mathcal{S} \times \mathcal{A} \to [0, R_{\max}]$, where $R_{\max} > 0$ 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 $d_0 \in \Delta(\mathcal{S})$. When d_0 is of importance to the discussion, we include it as part of the MDP definition, and write $M = (\mathcal{S}, \mathcal{A}, P, R, \gamma, d_0)$.

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 $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, 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 $s \in \mathcal{S}$.

The goal of the agent is to choose a policy π to maximize the expected discounted sum of rewards, or *value*:

$$\mathbb{E}\Big[\sum_{t=1}^{\infty} \gamma^{t-1} r_t \mid \pi, s_1\Big]. \tag{1}$$

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

$$0 \le \sum_{t=1}^{\infty} \gamma^{t-1} r_t \le \sum_{t=1}^{\infty} \gamma^{t-1} R_{\max} = \frac{R_{\max}}{1-\gamma}. \tag{2}$$

Hence, the discounted sum of rewards (or the discounted **return**) along any actual trajectory is always bounded in range $[0, \frac{R_{\max}}{1-\gamma}]$, 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 $V_M^\pi : \mathcal{S} \to \mathbb{R}$ as

$$V_M^\pi(s) = \mathbb{E}\Big[\sum_{t=1}^{\infty} \gamma^{t-1} r_t \mid \pi, s_1 = s\Big],$$

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

$$Q_M^\pi(s,a) = \mathbb{E}\Big[\sum_{t=1}^{\infty} \gamma^{t-1} r_t \mid \pi, s_1 = s, a_1 = a\Big].$$

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^π and Q^π can be computed using the following *Bellman equations for policy evaluation*: $\forall s \in \mathcal{S}, a \in \mathcal{A}$,

$$\begin{aligned} V^\pi(s) &= Q^\pi(s, \pi(s)). \\ Q^\pi(s,a) &= R(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a)}\big[V^\pi(s')\big]. \end{aligned} \tag{3}$$

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

Since $\mathcal{S}$ is assumed to be finite, upon fixing an arbitrary order of states (and actions), we can treat V^π and any distribution over $\mathcal{S}$ as vectors in $\mathbb{R}^{|\mathcal{S}|}$, and R and Q^π as vectors in $\mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$. This is particularly helpful as we can rewrite Equation 3 in an matrix-vector form and derive an analytical solution for V^π using linear algebra as below.

Define P^π as the transition matrix for policy π with dimension $|\mathcal{S}| \times |\mathcal{S}|$, whose (s, s')-th entry is

$$[P^\pi]_{s,s'} = \mathbb{E}_{a\sim\pi(s)}[P(s'|s,a)].$$

In fact, this matrix describes a Markov chain induced by MDP M and policy π. Its s-th row is the distribution over next-states upon taking actions according to π at state s, which we also write as $[P(s,\pi)]^\top$.

Similarly define R^π as the reward vector for policy π with dimension $|\mathcal{S}| \times 1$, whose s-th entry is

$$[R^\pi]_s = \mathbb{E}_{a\sim\pi(s)}[R(s,a)].$$

Then from Equation 3 we have

$$\begin{aligned}[V^\pi]_s = Q^\pi(s,\pi(s)) &= [R^\pi]_s + \gamma \mathbb{E}_{a\sim\pi(s)}\mathbb{E}_{s'\sim P(s,a)}\left[V^\pi(s')\right] \\ &= [R^\pi]_s + \gamma \mathbb{E}_{s'\sim P(s,\pi)}\left[V^\pi(s')\right] \\ &= [R^\pi]_s + \gamma \langle P(s,\pi), V^\pi\rangle,\end{aligned}$$

where $\langle \cdot\,, \cdot \rangle$ is dot product. Since this equation holds for every $s \in \mathcal{S}$, we have

$$V^\pi = R^\pi + \gamma P^\pi V^\pi \quad \Rightarrow \quad (\mathbf{I}_{|\mathcal{S}|} - \gamma P^\pi)V^\pi = R^\pi,$$

where $\mathbf{I}_{|\mathcal{S}|}$ is the identity matrix. Now we notice that matrix $(\mathbf{I}_{|\mathcal{S}|} - \gamma P^\pi)$ is always invertible. In fact, for any non-zero vector $x \in \mathbb{R}^{|\mathcal{S}|}$,

$$\begin{aligned}&\left\|(\mathbf{I}_{|\mathcal{S}|} - \gamma P^\pi)x\right\|_\infty = \|x - \gamma P^\pi x\|_\infty \\ &\geq \|x\|_\infty - \gamma\|P^\pi x\|_\infty && \text{(triangular inequality for norms)} \\ &\geq \|x\|_\infty - \gamma\|x\|_\infty && \text{(each element of } P^\pi x \text{ is a convex average of } x\text{)} \\ &= (1-\gamma)\|x\|_\infty > 0 && (\gamma < 1,\, x \neq \mathbf{0})\end{aligned}$$

So we can conclude that

$$V^\pi = (\mathbf{I}_{|\mathcal{S}|} - \gamma P^\pi)^{-1} R^\pi. \tag{4}$$

State occupancy When the reward function only depends on the current state, i.e., $R(s,a) = R(s)$, R^π is independent of π, and Equation 4 exhibits an interesting structure: implies that the value of a policy is *linear* in rewards, and the rows of the matrix $(\mathbf{I}_{|\mathcal{S}|} - \gamma P^\pi)^{-1}$ give the linear coefficients that depend on the initial state. Such coefficients, often represented as a vector, are called (unnormalized) *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. The entries in the vector sum up to $1/(1-\gamma)$, so we often multiply the vector with $(1-\gamma)$ to normalize it and obtain the normalized version, $d^{\pi,s}$.

An alternative way to define state occupancy is $d^{\pi,s} = (1-\gamma)\sum_{t=1}^\infty \gamma^{t-1} d_t^{\pi,s}$, where $d_t^{\pi,s}(s') = \mathbb{P}[s_t = s'|s_1 = s, \pi]$, i.e., the distribution of state at time step t when we start in s and follow policy π. We often write d^π when the trajectory starts with drawing a random initial state from some distribution d_0, and d_0 is often omitted in the notation of d^π.

Finally, one can also define state-action occupancy in a similar manner. We will often abuse the notation and use d^π to refer to either the state or the state-action occupancy when it is clear from the context which one we are referring to.

1.4 Bellman optimality equations

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

$V^\star$ and $Q^\star$ satisfy the following set of *Bellman optimality equations* [2]: $\forall s \in \mathcal{S}, a \in \mathcal{A}$,

$$\begin{aligned} V^\star(s) &= \max_{a \in \mathcal{A}} Q^\star(s,a). \\ Q^\star(s,a) &= R(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a)}\big[V^\star(s')\big]. \end{aligned} \tag{5}$$

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

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

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

$$\pi^\star = \pi_{Q^\star}.$$

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

$$(\mathcal{T} f)(s,a) := R(s,a) + \gamma \langle P(s,a), V_f \rangle, \tag{6}$$

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

$$Q^\star = \mathcal{T} Q^\star.$$

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 $\tilde{M} = (\tilde{\mathcal{S}}, \mathcal{A}, \tilde{P}, \tilde{R}, \gamma)$ such that $\tilde{\mathcal{S}} = \mathcal{S} \times [H] \bigcup \{s_{\text{absorbing}}\}$ ($[H] = \{1, \ldots, H\}$). 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 $\tilde{P}((s', h+1) \mid (s,h), a) = P(s'|s,a)$, 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 $R(s,a) = \mathbb{E}[r_t | s_t = s, a_t = a]$ where s_{t+1} and the independent noise are marginalized out. The value functions V^π and Q^π for any π 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 $r_t \in [0, R_{\max}]$. 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 $R_{\max} = a + b$, so that after adding the offset the reward lies in $[0, R_{\max}]$.

2 Planning in MDPs

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

2.1 Policy Iteration

The policy iteration algorithm starts from an arbitrary policy π_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 π_{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 $\pi^\star$ 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 $\pi^\star$ 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 π is defined as $A^\pi(s,a) := Q^\pi(s,a) - V^\pi(s)$. The advantage of policy π' over policy π 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 ("Performance Difference Lemma" [7]). *For any π, π', and any state $s \in \mathcal{S}$,*

$$V^{\pi'}(s) - V^\pi(s) = \frac{1}{1-\gamma}\mathbb{E}_{s' \sim d^{\pi',s}}[A^\pi(s',\pi')].$$

where $d^{\pi',s}$ is the normalized discounted occupancy induced by policy π' 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, π_i is the non-stationary policy that follows π' for the first i time-steps and switches to π for the remainder of the trajectory. Now we can rewrite the LHS of the statement as:

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

For each term on the RHS, observe that π_i and π_{i+1} share the same "roll-in" policy π' for the first i steps, which defines a roll-in distribution $\mathbb{P}[s_{i+1}|s_1 = s, \pi']$. They also share the same "roll-out" policy π 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 π_i and π_{i+1}. Putting together, we have

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

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

Proof of Theorem 1. Invoke the performance difference lemma on $\pi' = \pi_{k+1}$ and $\pi = \pi_k$, and that $V^{\pi_{k+1}} \geq V^{\pi_k}$ follows immediately by observing that $A^{\pi_k}(s, \pi_{k+1})$ is non-negative in every state. The strict improvement part follows by realizing that $A^{\pi_k}(s, \pi_{k+1})$ must be strictly positive in some state s_0, otherwise $\pi_k = \pi^\star$. In this case, $V^{\pi_{k+1}}(s_0) > V^{\pi_k}(s_0)$, because decomposing their difference using the performance difference lemma reveals that the difference contains $A^{\pi_k}(s_0, \pi_{k+1})$ (since the point mass on s_0 is "contained" in d^{π_{k+1}, s_0}), which is strictly positive. □

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 $|\mathcal{A}|^{|\mathcal{S}|}$, 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., 8], which is not well known.

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

Proof. We will use two facts: (a) $\mathcal{T}^{\pi_{k+1}} Q^{\pi_k} \geq \mathcal{T}^\pi Q^{\pi_k} \ \forall \pi$, (b) $\mathcal{T}^{\pi_{k+1}} Q^{\pi_k} \leq Q^{\pi_{k+1}}$. Here "$\leq$" and "$\geq$" are element-wise, and we will verify (a) and (b) at the end of this proof. Given (a) and (b), we have

$$Q^\star - Q^{\pi_{k+1}} = (Q^\star - \mathcal{T}^{\pi_{k+1}} Q^{\pi_k}) + (\mathcal{T}^{\pi_{k+1}} Q^{\pi_k} - Q^{\pi_{k+1}}) \leq \mathcal{T}^{\pi^\star} Q^\star - \mathcal{T}^{\pi^\star} Q^{\pi_k}.$$

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

$$\|Q^\star - Q^{\pi_{k+1}}\|_\infty \leq \|\mathcal{T}^{\pi^\star} Q^\star - \mathcal{T}^{\pi^\star} Q^{\pi_k}\|_\infty \qquad (Q^\star - Q^{\pi_{k+1}} \text{ is non-negative})$$
$$\leq \gamma \|Q^\star - Q^{\pi_k}\|_\infty. \qquad (\mathcal{T}^\pi \text{ is a } \gamma\text{-contraction for any } \pi)$$

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

$$(\mathcal{T}^{\pi_{k+1}} Q^{\pi_k})(s,a) = \mathbb{E}[\sum_{h=1}^{\infty} \gamma^{h-1} r_h | s_1 = s, a_1 = a, a_2 \sim \pi_{k+1}, a_{3:\infty} \sim \pi_k], \tag{7}$$

$$(\mathcal{T}^{\pi} Q^{\pi_k})(s,a) = \mathbb{E}[\sum_{h=1}^{\infty} \gamma^{h-1} r_h | s_1 = s, a_1 = a, a_2 \sim \pi, a_{3:\infty} \sim \pi_k], \tag{8}$$

$$Q^{\pi_{k+1}}(s,a) = \mathbb{E}[\sum_{h=1}^{\infty} \gamma^{h-1} r_h | s_1 = s, a_1 = a, a_2 \sim \pi_{k+1}, a_{3:\infty} \sim \pi_{k+1}], \tag{9}$$

where $a_{3:\infty}$ denote all the actions from the 3rd time step onwards, and $a_h \sim \pi$ is a shorthand for $a_h = \pi(s_h)$. Since π_{k+1} greedily optimizes Q^{π_k}, (7) $\geq$ (8) and (a) follows. (b) follows due to the policy improvement theorem, i.e., (9) $\geq$ (7) because π_{k+1} outperforms π_k in all states. □

Alternative Proof of Theorem 1 Below is an alternative (and more classical) proof that uses the *monotone* property of the Bellman update operators and some of the properties we established in the proof of Theorem 3.

Proof of Theorem 1. For any $Q \le Q'$, we have $\mathcal{T}Q \le \mathcal{T}Q'$. So,

$$Q^{\pi_k} = \mathcal{T}^{\pi_k} Q^{\pi_k} \le \mathcal{T} Q^{\pi_k} = \mathcal{T}^{\pi_{k+1}} Q^{\pi_k}.$$

Applying this multiple times, we get

$$Q^{\pi_k} \le \mathcal{T}^{\pi_{k+1}} Q^{\pi_k} \le \mathcal{T}^{\pi_{k+1}}(\mathcal{T}^{\pi_{k+1}} Q^{\pi_k}) \le \ldots \le (\mathcal{T}^{\pi_{k+1}})^{\infty} Q^{\pi_k} = Q^{\pi_{k+1}}. \qquad \square$$

2.2 Value Iteration

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

$$Q^{\star,h} = \mathcal{T} Q^{\star,h-1}. \tag{10}$$

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 $\|Q^{\star,H} - Q^\star\|_\infty$ as a function of H. If H is large enough, we can guarantee that $Q^{\star,H}$ is sufficiently close to $Q^\star$, and the following result bounds the suboptimality (or loss) of acting greedily with respect to an approximate Q-value function, f:

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

Proof. For any $s \in \mathcal{S}$,

$$\begin{aligned} V^\star(s) - V^{\pi_f}(s) &= Q^\star(s, \pi^\star(s)) - Q^\star(s, \pi_f(s)) + Q^\star(s, \pi_f(s)) - Q^{\pi_f}(s, \pi_f(s)) \\ &\le Q^\star(s, \pi^\star(s)) - f(s, \pi^\star(s)) + f(s, \pi_f(s)) - Q^\star(s, \pi_f(s)) \\ &\quad + \gamma \mathbb{E}_{s' \sim P(s, \pi_f(s))}[V^\star(s') - V^{\pi_f}(s')] \\ &\le 2\|f - Q^\star\|_\infty + \gamma \|V^\star - V^{\pi_f}\|_\infty. \end{aligned} \qquad \square$$

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

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

$$\|\mathcal{T}f - \mathcal{T}f'\|_\infty \le \gamma \|f - f'\|_\infty. \tag{11}$$

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

$$\begin{aligned} \left|[\mathcal{T}f - \mathcal{T}f']_{s,a}\right| &= \left|R(s,a) + \gamma\langle P(s,a), V_f\rangle - R(s,a) - \gamma\langle P(s,a), V_{f'}\rangle\right| \\ &\le \gamma\left|\langle P(s,a), V_f - V_{f'}\rangle\right| \le \gamma \|V_f - V_{f'}\|_\infty \le \gamma \|f - f'\|_\infty. \end{aligned}$$

The last step uses the fact that $\forall s \in \mathcal{S}$, $|V_f(s) - V_{f'}(s)| \le \max_{a\in\mathcal{A}} |f(s,a) - f'(s,a)|$. 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

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

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

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

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

$$\|Q^{\star,H} - Q^\star\|_\infty \le \gamma^H R_{\max}/(1-\gamma). \tag{12}$$

To guarantee that we compute a value function ϵ-close to $Q^\star$, it is sufficient to set

$$H \ge \frac{\log \frac{R_{\max}}{\epsilon(1-\gamma)}}{1-\gamma}. \tag{13}$$

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

$$\gamma^H \frac{R_{\max}}{1-\gamma} = (1-(1-\gamma))^{\frac{1}{1-\gamma}\cdot H(1-\gamma)} \frac{R_{\max}}{1-\gamma} \le \left(\frac{1}{e}\right)^{\log \frac{R_{\max}}{\epsilon(1-\gamma)}} \frac{R_{\max}}{1-\gamma} = \epsilon.$$

Here we used the fact that $(1-1/x)^x \le 1/e$ 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.[1] 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 $\|Q^{\star,H} - Q^\star\|_\infty$: the finite-horizon interpretation

Equation 12 can be derived using an alternative argument, which views Value Iteration as optimizing value for a finite horizon. $V^{\star,H}(s)$ 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 π, define its H-step truncated value

$$V^{\pi,H}(s) = \mathbb{E}\Big[\sum_{t=1}^H \gamma^{t-1} r_t \mid \pi, s_1 = s\Big]. \tag{14}$$

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

$$V^{\pi^\star,H}(s) \le V^{\star,H}(s).$$

Note that the LHS and RHS are not to be confused: $\pi^\star$ 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).

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

The LHS can be lower bounded as $V^{\pi^\star,H}(s) \ge V^\star(s) - \gamma^H R_{\max}/(1-\gamma)$, because $V^{\pi^\star,H}$ 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 $V^{\star,H}(s) \le V^\star(s)$: $V^\star$ should dominate any stationary and non-stationary policies, including the one that first achieves $V^{\star,H}$ within H steps and picks up some non-negative rewards afterwards with any behavior. Combining the lower and the upper bounds, we have $\forall s \in \mathcal{S}$,

$$V^\star(s) - \frac{\gamma^H R_{\max}}{1-\gamma} \le V^{\star,H}(s) \le V^\star(s), \tag{15}$$

which immediately leads to Equation 12.

On the use of stationary policies
The above analysis shows that the suboptimality of the policy obtained by VI is $2\gamma^H R_{\max}/(1-\gamma)^2$. It turns out by slighting tweaking the algorithm we can drop a factor of the effective horizon in the bound (a related idea has been discussed in [10]): We run VI as usual, and instead of outputting the greedy policy of the final Q-function, we output a non-stationary policy that is the concatenation of $\pi_{Q^{\star,H}}, \pi_{Q^{\star,H-1}}, \ldots, \pi_{Q^{\star,1}}$, followed by arbitrary policies. Recall from above that the value function of such a non-stationary policy is at least $V^{\star,H}$, and Eq.(15) shows that its suboptimality gap is bounded by $\gamma^H R_{\max}/(1-\gamma)$.

2.3 Linear Programming

The last basic algorithm we will introduce is to cast the planning problem as a linear program (LP), and one can solve the latter with a plethora of solvers out there.

Primal Form The primal LP is:

$$\min_{V \in \mathbb{R}^{\mathcal{S}}} d_0^\top V$$
$$\text{s.t. } V \ge \mathcal{T}V.$$

Here $d_0 \in \mathbb{R}^{\mathcal{S}}$ is the vector that represents the initial state distribution. The decision variable is V, which represents a value function. The optimal solution, as we will show below, is $V = V^\star$ when d_0 is fully supported on $\mathcal{S}$ (i.e., $d_0(s) > 0\ \forall s$).

It is worth pointing out that the constraints $V \ge \mathcal{T}V$ are *not* linear: for each $s \in \mathcal{S}$, the constraint

$$V(s) \ge \max_{a \in \mathcal{A}} \left(R(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a)}[V(s')] \right),$$

is nonlinear due to the $\max_a$ operator. However, they can be converted into equivalent linear constraints, at the cost of blowing up the number of constraints from $|\mathcal{S}|$ to $|\mathcal{S} \times \mathcal{A}|$; the above constraint for s is equivalent to the following $|\mathcal{A}|$ constraints:

$$V(s) \ge R(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a)}[V(s')],\ \forall a \in \mathcal{A}.$$

Therefore, the primal LP of MDP has $|\mathcal{S}|$ decision variables and $|\mathcal{S} \times \mathcal{A}|$ constraints.

It then remains to show that $V = V^\star$ is the optimal solution. A counterintuitive part of the primal form is that we *minimize* the value function V instead of maximizing it. In fact, this is precisely what we want given the constraints: every feasible V is pointwise *above* $V^\star$, which is why minimizing $d_0^\top V$ for a fully-supported V yields $V = V^\star$.

Proposition 5. *$V \geq \mathcal{T}V \Rightarrow V \geq V^\star$, and $V^\star$ is feasible.*

Proof. We will use the monotonicity of $\mathcal{T}$, that is, for any $V \geq V'$, we have $\mathcal{T}V \geq \mathcal{T}V'$. Therefore, for any V that satisfies $V \geq \mathcal{T}V$, we let $V' := \mathcal{T}V$ and invoke the monotone property:

$$\mathcal{T}V \geq \mathcal{T}(\mathcal{T}V) := \mathcal{T}^2 V.$$

So $V \geq \mathcal{T}V \geq \mathcal{T}^2 V$. By induction we have $V \geq \mathcal{T}^n V$ for any positive integer n. As $n \to \infty$, we have $V \geq \mathcal{T}^\infty V = V^\star$. The feasibility of $V^\star$ follows from $V^\star = \mathcal{T}V^\star$. □

Dual Form Taking the dual of the primal LP yields:

$$\begin{aligned} &\max_{d \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}, d \geq \mathbf{0}} d^\top R \\ &\text{s.t. } [\textstyle\sum_{a\in\mathcal{A}} d(s,a)]_{s\in\mathcal{S}} = \gamma P^\top d + (1-\gamma) d_0. \end{aligned}$$

The LHS of the constraint is a vector in $\mathbb{R}^{\mathcal{S}}$. The dual LP has a very interesting interpretation: the decision variable $d \in \mathbb{R}$ can be thought of as the discounted state-action occupancy of a stationary (and possibly stochastic) policy starting from d_0 as the initial state distribution, so its dot product with the reward function $R \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ naturally corresponds to the expected return of the policy under d_0 and it makes sense to maximize such a return. We can see this from the dual constraint, which are sometimes called the *Bellman flow equations*. To make sense of the constraint, it is instructive to verify that $d = d^\pi$, the state-action occupancy of some policy π, satisfies the constraint. In this part we will use $\tilde{d}^\pi$ to refer to the state occupancy to distinguish from the state-action occupancy.

The LHS of the constraint marginalizes out the actions, so we obtain the state occupancy induced by π: $[\sum_{a\in\mathcal{A}} d^\pi(s,a)]_{s\in\mathcal{S}} = \tilde{d}^\pi$. The first term on the RHS is $\gamma P^\top d$. Since $P \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|\times|\mathcal{S}|}$ is the transition matrix of the MDP, for any state-action distribution $q \in \Delta(\mathcal{S}\times\mathcal{A})$, $P^\top q$ represents a state distribution that can be described by the following generative process:

$$s' \sim P^\top q \Leftrightarrow (s,a) \sim q, s' \sim P(\cdot|s,a).$$

So essentially $P^\top q$ "pushes" q through the dynamics of P for one time step. A key realization here is that for $q = d_t^\pi$, we have $P^\top d_t^\pi = \tilde{d}_{t+1}^\pi$, i.e., pushing the t-th step state-action distribution of π through the dynamics yields the $(t+1)$-th step state distribution of the same policy. With this, we have

$$\begin{aligned} \gamma P^\top d &= \gamma(1-\gamma) P^\top \textstyle\sum_{t=1}^\infty \gamma^{t-1} d_t^\pi = \gamma(1-\gamma)\sum_{t=1}^\infty \gamma^{t-1} P^\top d_t^\pi \\ &= (1-\gamma)\sum_{t=1}^\infty \gamma^t \tilde{d}_{t+1}^\pi = (1-\gamma)\sum_{t=2}^\infty \gamma^{t-1}\tilde{d}_t^\pi. \end{aligned}$$

So we almost have the discounted state occupancy of π, $\tilde{d}^\pi = (1-\gamma)\sum_{t=1}^\infty \gamma^{t-1}\tilde{d}_t^\pi$, except that the $t=1$ term in the summation is absent. However, that missing term is precisely the second term on the RHS of the constraint: $(1-\gamma)\tilde{d}_1^\pi = (1-\gamma)d_0$.

As a final remark, the constraint in the dual LP describes the space of state-action occupancy of *all* stationary (and possibly stochastic) policies. In fact, for any feasible d, we can recover the policy that induces such an occupancy by

$$\pi(a|s) = \frac{d(s,a)}{\sum_{a' \in \mathcal{A}} d(s,a')},$$

which is how one would back out $\pi^\star$ from the solution of the dual LP.

References

[1] ML Puterman. Markov decision processes. *Jhon Wiley & Sons, New Jersey*, 1994.

[2] Richard Bellman. Dynamic programming and Lagrange multipliers. *Proceedings of the National Academy of Sciences*, 42(10):767–769, 1956.

[3] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, March 1998.

[4] Andrew Y Ng and Stuart J Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the 17th International Conference on Machine Learning*, pages 663–670, 2000.

[5] Pieter Abbeel and Andrew Y Ng. Apprenticeship Learning via Inverse Reinforcement Learning. In *Proceedings of the 21st International Conference on Machine learning*, page 1. ACM, 2004.

[6] John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in Neural Information Processing Systems*, pages 817–824, 2008.

[7] Sham Kakade and John Langford. Approximately Optimal Approximate Reinforcement Learning. In *Proceedings of the 19th International Conference on Machine Learning*, volume 2, pages 267–274, 2002.

[8] Rémi Munos. Error bounds for approximate policy iteration. In *ICML*, volume 3, pages 560–567, 2003.

[9] Satinder Singh and Richard Yee. An upper bound on the loss from approximate optimal-value functions. *Machine Learning*, 16(3):227–233, 1994.

[10] Bruno Scherrer and Boris Lesner. On the use of non-stationary policies for stationary infinite-horizon markov decision processes. In *Advances in Neural Information Processing Systems*, pages 1826–1834, 2012.