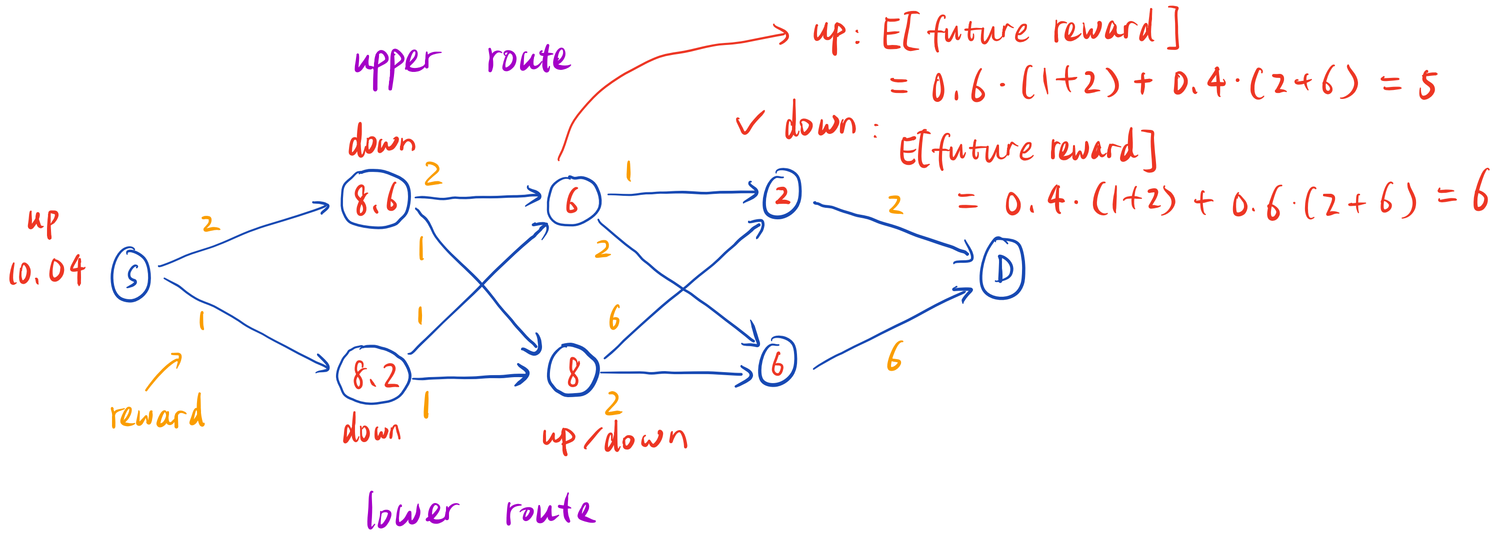
An example of an MDP

You are playing Simple Mario Party and given the following map:



Choose action = "up": w.p. 0.6 → go up

w.p. 0.4 → go down

Choose action = "down": w.p. 0.4 → go up

w.p. 0.6 → go down

- state: where you are
- action: up / down

(state and action) decide the distr. for next state

- reward (per step): func. of state and action

  total reward: sum of rewards over the time horizon
- policy: which action to take in each state at each time step.
- goal: design a policy to maximize total expected reward

We next introduce the general form of an MDP, where we also generalize the time horizon from a finite one to an infinite one.

Markov decision processes (MDPs)

An MDP is a discrete-time process specified by:

- State space $\mathcal{S}$. Let $s_t \in \mathcal{S}$ be the state at time t.
- Action space $\mathcal{A}$. $a_t \in \mathcal{A}$: action at time t.

For simplicity, we focus on the setting where $\mathcal{S}$ and $\mathcal{A}$ are finite.

- Transition probabilities $P: \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, $\Delta(\mathcal{S})$: prob. distr. over $\mathcal{S}$.

  $P(s' | s, a)$: prob. of transitioning to state s' when taking action a in state s.
- Reward function: $r: \mathcal{S} \times \mathcal{A} \to \mathbb{R}$.

  $r(s, a)$: immediate reward when taking action a in state s.

  $r(s, a)$ could be random, in which case $r(s, a)$ denotes its mean.
- Policy: In general, a policy can choose a_t based on the full history $\mathcal{H}_t = (s_0, a_0, r_0, s_1, a_1, r_1, \cdots, s_{t-1}, a_{t-1}, r_{t-1}, s_t)$.

  We focus on stationary policies $\pi: \mathcal{S} \to \Delta(\mathcal{A})$, which chooses actions based on only the current state, i.e., $a_t \sim \pi(\cdot | s_t)$.

  We sometimes simply write $a_t = \pi(s_t)$.
- Goal: Find a policy π to solve

$$\underset{\pi}{\text{maximize}} \; V^\pi(s) \triangleq \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \,\Big|\, s_0 = s, \pi\right]$$

  $\gamma \in (0,1)$: discount factor.

  Explanation: (1) γ: prob. that the problem continues after each time step.

  (2) Reward now is more important than that in the future.

  $V^\pi: \mathcal{S} \to \mathbb{R}$: value function of policy π.

Remark. In general, the optimization is over all policies, which can be non-stationary and non-Markovian. But it can be shown that optimality can be achieved by a stationary policy.

Bellman Equation (Dynamic Programming Equation)

Note that for a fixed policy π,

$$V^\pi(s) = E\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \,\middle|\, s_0 = s, \pi\right]$$

$$= r(s, \pi(s))$$

$$+ \sum_{s'} \underbrace{E\left[\sum_{t=1}^{\infty} \gamma^t r(s_t, a_t) \,\middle|\, s_1 = s', \pi\right]}_{\gamma V^\pi(s')} \cdot P(s' \mid s, \pi)$$

$$= r(s, \pi(s)) + \gamma E\left[V^\pi(s_1) \mid s_0 = s, a_0 \sim \pi(\cdot \mid s)\right],$$

which gives an equation for V^π.

Let $V^*(s) = \sup_\pi V^\pi(s)$ be the optimal value function. Then V^* satisfies a similiar equation, referred to as the Bellman equation.

Theorem. The optimal value function V^* satisfies

$$V^*(s) = \max_a \left(r(s, a) + \gamma E\left[V^*(s_1) \mid s_0 = s, a_0 = a \right] \right), \quad \forall s. \tag{1}$$

Moreover, let policy π^* be specified as

$$\pi^*(s) \in \operatorname*{argmax}_a \left(r(s, a) + \gamma E\left[V^*(s_1) \mid s_0 = s, a_0 = a \right] \right). \tag{2}$$

Then π^* is an optimal policy.

Remark π^* is a stationary, deterministic policy.

How can we make use of the Bellman equation to get an optimal policy? Naturally, we want to solve the Bellman equation to get V^*, and then use equation (2) to get an optimal policy. To be able to do so, we need to answer the following questions:

(i) If we find a solution to the Bellman equation, is it guaranteed to be V^*?

(ii) How do we find a solution to the Bellman equation?

To answer both questions, it is convenient to write the Bellman equation using the so-called Bellman operator.

Bellman Operator

We index the state space as $\mathcal{S} = \{1, 2, \cdots, d\}$. Then a value function V can be written as a vector: $V = (V(1), V(2), \cdots, V(d)) \in \mathbb{R}^d$.

Recall the Bellman equation:

$$V(s) = \max_a \left(r(s,a) + \gamma E[V(s_1) \mid s_0 = s, a_0 = a] \right), \forall s \in \mathcal{S}.$$

We can rewrite the right-hand-side of the Bellman equation by defining the Bellman operator $T: \mathbb{R}^d \to \mathbb{R}^d$, which takes a value function as input and outputs another value function. Specifically, for any $V \in \mathbb{R}^d$, $TV \in \mathbb{R}^d$ is defined as

$$TV(s) = \max_a r(s,a) + \gamma E[V(s_1) \mid s_0 = s, a_0 = a], \forall s \in \mathcal{S}.$$

Then the Bellman equation can be written as: $V = TV$.

Now let's return to the two questions:

(i) If we find a solution to the Bellman equation $V = TV$, is it guaranteed to be V^*?

(ii) How do we find a solution to $V = TV$?

Solving $V = TV$ is to find a fixed point of the operator T. If T is a contraction mapping, then these two questions can be answered by the Banach fixed-point theorem.

<u>Contraction mapping</u>: Let (X, d) be a complete metric space. Then a mapping $T: X \to X$ is said to be a contraction mapping if there exists $r \in [0,1)$ such that $d(T(x), T(y)) \le \underbrace{r}_{\text{contraction coefficient}} \cdot d(x,y)$, $\forall x, y \in X$.

We say T has a fixed point x^* if $T(x^*) = x^*$.

<u>Banach fixed-point theorem (contraction mapping theorem)</u>

Let T be a contraction mapping on a complete metric space (X, d) with a contraction coefficient r. Then

(1) T has a unique fixed point x^*.

(2) The iterative algorithm $x_{k+1} = T(x_k)$, starting from any initial point $x_0 \in X$, has the property $d(x_{k+1}, x^*) \le r \cdot d(x_k, x^*)$.

As a result, $x_k \to x^*$ geometrically fast, with the following equivalent descriptions of the convergence speed:

(i) $d(x_k, x^*) \le r^k d(x_0, x^*)$

(ii) $d(x_k, x^*) \le \frac{r^k}{1-r} d(x_1, x_0)$

$$\left(d(x_1, x_0) \ge d(x_0, x^*) - d(x_1, x^*) \ge (1-r) d(x_0, x^*) \ge \frac{1-r}{r^k} d(x_k, x^*) \right)$$

Is the Bellman operator a contraction mapping then?

Theorem. The Bellman operator T is a contraction mapping on $\mathbb{R}^d$ under $\|\cdot\|_\infty$ with the discount factor γ as a contraction coefficient,

i.e., $\forall V_1, V_2 \in \mathbb{R}^d$, $\|TV_1 - TV_2\|_\infty \le \gamma \|V_1 - V_2\|_\infty$.

$(\|x\|_\infty = \max\{|x_1|, |x_2|, \cdots, |x_d|\}, \forall x \in \mathbb{R}^d)$

Proof. Let $s \in \mathcal{S}$. Then

$TV_1(s) - TV_2(s)$ → suppose this max is achieved at a^*

$$= \max_a \left(r(s,a) + \gamma \mathbb{E}[V_1(s_1) \mid s_0 = s, a_0 = a] \right) - \max_{a'} \left(r(s,a') + \gamma \mathbb{E}[V_2(s_1) \mid s_0 = s, a_0 = a'] \right)$$

$$\le r(s,a^*) + \gamma \mathbb{E}[V_1(s_1) \mid s_0 = s, a_0 = a^*] - \left(r(s,a^*) + \gamma \mathbb{E}[V_2(s_1) \mid s_0 = s, a_0 = a^*] \right)$$

$$= \gamma \mathbb{E}[\underbrace{V_1(s_1) - V_2(s_1)}_{\le \|V_1 - V_2\|_\infty} \mid s_0 = s, a_0 = a^*]$$

$$\le \gamma \|V_1 - V_2\|_\infty$$

Similarly, $TV_2(x) - TV_1(x) \le \gamma \|V_1 - V_2\|_\infty$

Therefore, $\|TV_1 - TV_2\|_\infty \le \gamma \|V_1 - V_2\|_\infty$. □

Implications.

1. The Bellman equation $V = TV$ has a unique solution.
   Therefore, the solution must be the optimal value function V^*.
2. The iterative algorithm $V_{k+1} = TV_k$ guarantees that $V_k \to V^*$ as $k \to \infty$.
   This gives rise to the value iteration algorithm below.

Computational Techniques

Value iteration (VI) Starting at some V, we iteratively apply T: $V \leftarrow TV$.

Algorithm
1. Initialize with a guess V_0, set $k=0$.
2. $V_{k+1} = T V_k$
3. $k \leftarrow k+1$
4. Repeat 2-3 until "convergence".
5. Let V_K be the output value function. Output policy π_K defined by

$$\pi_K(s) \in \arg\max_a \left(r(s,a) + \gamma \mathbb{E}[V_K(s_1) \mid s_0 = s, a_0 = a] \right).$$

From the contraction mapping theorem, we have convergence.

In practice, we need to use some stopping criterion.

If we stop after K steps, how good is V_K and how good is π_K?

- Bound on $\|V_K - V^*\|_\infty$. By the contraction mapping theorem,

$$\|V_K - V^*\|_\infty \le \frac{\gamma^K}{1-\gamma} \|V_1 - V_0\|.$$

This bound is more useful than the bound $\|V_K - V^*\|_\infty \le \gamma^K \|V_0 - V^*\|_\infty$ because $\|V_1 - V_0\|_\infty$ is computable while $\|V_0 - V^*\|_\infty$ is unknown.

How good is π_K? Note that V_K is not necessarily the value function of π_K, but they are close. Recall that we use V^{π_K} to denote the value function of π_K.

- Bound on $\|V^{\pi_K} - V^*\|_\infty$.

$$\|V^{\pi_K} - V^*\|_\infty \le \|V^{\pi_K} - V_K\|_\infty + \underbrace{\|V_K - V^*\|_\infty}_{\text{just bounded}}.$$

Note that
$$\begin{aligned} V^{\pi_K}(s) &= r(s, \pi_K(s)) + \gamma \mathbb{E}[V^{\pi_K}(s_1) \mid s_0 = s, a_0 = \pi_K(s)] \\ &= \underbrace{r(s, \pi_K(s)) + \gamma \mathbb{E}[V_K(s_1) \mid s_0 = s, a_0 = \pi_K(s)]}_{V_{K+1}(x) \text{ by definition of } \pi_K} \end{aligned}$$

$$+ \gamma \mathbb{E}\left[V^{\pi_k}(s_1) - V_k(s_1) \mid s_0 = s, a_0 = \pi_k(s) \right]$$

$$= V_{k+1}(s) + \gamma \mathbb{E}\left[V^{\pi_k}(s_1) - V_k(s_1) \mid s_0 = s, a_0 = \pi_k(s) \right].$$

Thus $\| V^{\pi_k} - V_{k+1} \|_\infty \le \gamma \| V^{\pi_k} - V_k \|_\infty$.

We also know that $\| V^{\pi_k} - V_{k+1} \|_\infty \ge \| V^{\pi_k} - V_k \|_\infty - \| V_{k+1} - V_k \|_\infty$.

So $\| V^{\pi_k} - V_k \|_\infty \le \frac{1}{1-\gamma} \| V_{k+1} - V_k \|_\infty$

$$\le \frac{\gamma^k}{1-\gamma} \| V_1 - V_0 \|_\infty .$$

Putting them together, we have

$$\| V^{\pi_k} - V^* \|_\infty \le \| V^{\pi_k} - V_k \|_\infty + \| V_k - V^* \|_\infty$$

$$\le \frac{2\gamma^k}{1-\gamma} \| V_1 - V_0 \|_\infty .$$

The value iteration algorithm centers around the value function: it first makes sure that the value function obtained is close enough to the optimal value function, and then outputs a policy. Next we introduce another algorithm that promotes a more policy-centered view.

Policy iteration (PI). The structure of PI is as follows. We start from an arbitrary policy, and repeat the following iterative procedure:

1. Policy evaluation: calculate the value function of the policy.
2. Policy improvement: update the policy to improve it.

To make these two steps more concrete, we first define the operator associated with a policy for convenience. When we fix a policy π, we know that its value function V^π satisfies

$$V^\pi(s) = r(s, \pi(s)) + \gamma \mathbb{E}[V^\pi(s_1) \mid s_0 = s, a_0 = \pi(s)], \ \forall s \in \mathcal{S}.$$

Similar to the Bellman operator, the operator T^π associated with policy π is defined based on the right-hand-side of the equation. Specifically, for any $V \in \mathbb{R}^d$, $T^\pi V \in \mathbb{R}^d$ is defined as

$$T^\pi V(s) = r(s, \pi(s)) + \gamma \mathbb{E}[V^\pi(s_1) \mid s_0 = s, a_0 = \pi(s)], \ \forall s \in \mathcal{S}.$$

Then the equation for policy π can be written as: $V^\pi = T^\pi V^\pi$.

Note that T^π is a linear operator.

Claim T^π is a contraction mapping on $\mathbb{R}^d$ under $\|\cdot\|_\infty$ with the discount factor γ as a contraction coefficient, i.e., $\forall V_1, V_2 \in \mathbb{R}^d$,

$$\|T^\pi V_1 - T^\pi V_2\|_\infty \le \gamma \|V_1 - V_2\|_\infty.$$

Implication. 1. The equation $V = T^\pi V$ has a unique solution, which is the value function of π, V^π.

2. In the policy evaluation step, we can use the iterative algorithm $V_{k+1} = T^\pi V_k$ to calculate V^π.

We can also show that both the Bellman operator T and the operator T^π are monotonic, i.e., $V_1 \le V_2 \Rightarrow TV_1 \le TV_2$, $T^\pi V_1 \le T^\pi V_2$.

The policy improvement step.

We can improve a policy using the right-hand-side of the Bellman equation.

To update the policy π_k at the kth iteration, we define π_{k+1} as

$$\pi_{k+1}(s) \in \arg\max_a \left(r(x,a) + \gamma E\left[V^{\pi_k}(s_1) \mid s_0 = s, a_0 = a \right] \right), \forall s.$$

Using the notation of operators, this implies that

$$T^{\pi_{k+1}} V^{\pi_k} = T V^{\pi_k}.$$

Putting the two steps together, the PI algorithm is given by:

1. Start with a policy π_0. Set $k=0$.
2. Compute the value function V^{π_k} of policy π_k using the equation $V = T^{\pi_k} V$.
3. Update the policy:
$$\pi_{k+1}(s) \in \arg\max_a \left(r(x,a) + \gamma E\left[V^{\pi_k}(s_1) \mid s_0 = s, a_0 = a \right] \right), \forall s.$$
4. $k \leftarrow k+1$
5. Repeat 2-4 until "convergence".

<u>Theorem</u> Under policy iteration, we have

(1) $V^{\pi_{k+1}} \geq V^{\pi_k}$, i.e., the policy improves at each step, and

(2) If $V^{\pi_{k+1}} = V^{\pi_k}$, then π_k is an optimal policy.

<u>Proof</u> (1) By step 3, $T^{\pi_{k+1}} V^{\pi_k} = T V^{\pi_k} \geq T^{\pi_k} V^{\pi_k} = V^{\pi_k}$.

By the monotinicity of $T^{\pi_{k+1}}$, we have

$$T^{\pi_{k+1}}(T^{\pi_{k+1}} V^{\pi_k}) \geq T^{\pi_{k+1}}(V^{\pi_k}) \leq V^{\pi_k}$$

Keep applying $T^{\pi_{k+1}}$ N times, we get

$$(T^{\pi_{k+1}})^N V^{\pi_k} \geq V^{\pi_k}.$$

By the contraction property of $T^{\pi_{k+1}}$, taking $N \to \infty$ gives

$$V^{\pi_{k+1}} \geq V^{\pi_k}$$

$V^{\pi_{k+1}}$

(2) If $V^{\pi_{k+1}} = V^{\pi_k}$, then $T^{\pi_{k+1}} V^{\pi_k} = T V^{\pi_k} \Rightarrow T^{\pi_{k+1}} V^{\pi_{k+1}} = T V^{\pi_{k+1}}$ $\Rightarrow V^{\pi_{k+1}} = T V^{\pi_{k+1}}$. So $V^{\pi_{k+1}}$ satisfies the Bellman equation, which means that π_{k+1} and π_k are optimal policies.

Implications of the theorem. The theorem says that at each step, you either get an improved policy or you have found the optimal policy.

- So in principle, PI converges in a finite number of steps when the state space and action space are finite.
- However, in each step, one needs to compute V^{π_k}. This can be done using the iterative algorithm $V_{i+1} = T^{\pi_k} V_i$. This inner loop can take a long time to produce an accurate value for V^{π_k}.

Q-function

Recall the Bellman equation:

$$V^*(s) = \max_a \left(r(s,a) + \gamma \sum_{s'} V^*(s') \cdot P(s' \mid s,a) \right)$$

Suppose V^* is known, we still cannot solve this maximization problem to get the optimal policy without knowing the model $P(s' \mid s,a)$. However, if we obtain the following function

$$Q(s,a) \triangleq r(s,a) + \gamma \sum_{s'} V^*(s') \cdot P(s' \mid s,a),$$

then we can solve $\max_a Q(s,a)$ to get the optimal policy. The function $Q: \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is called the (optimal) Q-function.

Meaning of $Q(s,a)$: the total discounted reward when we take action a in the current step and follow the optimal policy in all the future time steps.

How can we get the Q function? A starting point is the equation below derived from the Bellman equation. Note that $V^*(s) = \max_a Q(s,a)$.

Then

$$\begin{aligned} Q(s,a) &= r(s,a) + \gamma \sum_{s'} V^*(s') \cdot P(s' \mid s,a) \\ &= r(s,a) + \gamma \sum_{s'} P(s' \mid s,a) \cdot \max_{a'} Q(s',a'). \end{aligned}$$

Directly evaluating the right-hand-side still requires the knowledge of $P(s' \mid s,a)$, but there are many ways to learn the Q-function when the model is unknown.

We can also define the Q-function for a fixed policy π as follows:

$$Q^\pi(s,a) = r(s,a) + \gamma E[V^\pi(s_1) \mid s_0 = s, a_0 = a].$$

This is the total discounted reward when we take action a in the current time step and follow the policy π in the future. Then

$$V^\pi(s) = E[Q^\pi(s,a) \mid a \sim \pi(\cdot|s)].$$

In many RL approaches, we need to evaluate the Q-function for a given policy π.