Dynamic Programming for Markov Decision Process I

Notation:

- uppercase (e.g. R_t): random variable
- lowercase (e.g. s_t): instance of random variable
- bold straight (e.g. $\mathbf{v}$): determineistic vector

Markov Decision Process

A Markov decision process (MDP) consists of

- $\mathcal{S}$: set of states.
- $\mathcal{A}$: set of actions.
- $p(\cdot \mid s, a)$: state transition probability. i.e. the probability distribution of the new state given the current state s and current action a.
- $\gamma \in [0, 1)$: discount factor.
- $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$: reward function. bounded.

In the following, we consider stationary MDP. i.e. Both the station transition probability and reward function are time-independent.

Rewards

Given the state s_t at time t, the agent takes action a_t, which leads to

- the new state s_{t+1}, sampled from $p(\cdot \mid s_t, a_t)$
- and the reward r_t, determined by the reward function $r_t = r(s_t, a_t)$.

Continue taking actions $a_{t+1}, a_{t+2}, \ldots$, we get a state-action-reward trajectory

$$s_t \xrightarrow[r_t]{a_t} s_{t+1} \xrightarrow[r_{t+1}]{a_{t+1}} s_{t+2} \xrightarrow[r_{t+2}]{a_{t+2}} s_{t+3} \cdots$$

$\to$ Intuitive goal: Maximize the sum of all r_t.

In the stochastic setting, all states $S_t, S_{t+1}, \ldots$ are random. Since the agent takes action based on current state, the action sequence is also random. Similary, the reward sequence is also random. $\implies$ stochastic state-action-reward trajectory:

$$S_t \xrightarrow[R_t]{A_t} S_{t+1} \xrightarrow[R_{t+1}]{A_{t+1}} S_{t+2} \xrightarrow[R_{t+2}]{A_{t+2}} S_{t+3} \cdots$$

$\to$ Intuitive goal: Maximize $\mathbb{E}[\text{sum of all } R_t]$.

The total (discounted) reward G_t is defined as the sum of (discounted) rewards starting from S_t

$$G_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \ldots \tag{1}$$

$$= \sum_{k=0}^{\infty} \gamma^k R_{t+k} \tag{2}$$

Remarks:

- G_t is a random quantity as the state transition is stochastic. Taking the same action sequence from s_t again will yield a different total reward.
- The discount factor γ serves two purposes:
  1. it ensures that the infinite sum is defined. Recall from math: If $\sum_{n=1}^{\infty} a_n$ converges absolutely and $\{b_n\}$ is bounded, then $\sum_{n=1}^{\infty} a_n b_n$ converges absolutely. Here, we have a geometric series which converges absolutely and a bounded sequence of rewards.
  2. it puts more weight on short-term rewards over long-term rewards. e.g. Practical meaning in finance, 100 dollar today is worth more than 100 dollar next year.

Recursive structure: The total reward G_t comprises immediate reward R_t plus a discounted future reward G_{t+1}.

$$G_t = R_t + \gamma G_{t+1} \tag{3}$$

Policy

In MDP, the agent performs actions determined by current state, according to a ***policy***

$$\pi : \mathcal{S} \to \mathcal{A}, s \mapsto a = \pi(s) \tag{4}$$

Remark: The policy defined here is **time-invariant** and **deterministic**, i.e. For a certain state s, the agent takes the **same** action **whenever** he arrives at s.

Following a policy π from a fixed initial state s, G_t can be written as

$$\begin{aligned} G_t &= r(s,a) + \gamma r\left(S_{t+1}, A_{t+1}\right) + \gamma^2 r(S_{t+2}, A_{t+2}) + \dots \\ &= r(s,\pi(s)) + \gamma r\left(S_{t+1}, \pi(S_{t+1})\right) + \gamma^2 r(S_{t+2}, \pi(S_{t+2})) + \dots \end{aligned}$$

Hence, all stochasticity of G_t comes from stochasticity in $\{S_k\}_{k\geq t+1}$.

State Value Function

> For a certain policy π, the corresponding ***state value function*** is mapping $v_\pi : \mathcal{S} \to \mathbb{R}, s \mapsto v_\pi(s)$. $v_\pi(s)$ is called ***state value***, defined as the expected total reward by executing policy π starting from state s.
>
> $$v_\pi(s) = \mathbb{E}\left[G_t \mid S_t = s\right], \quad \forall s \in \mathcal{S} \tag{5}$$

Remarks:

- The expectation is take over $\{S_k\}_{k\geq t+1}$. The random variable G_t depends implicity on policy π as it represents the total reward by executing π.
- State value $v_\pi(s)$ is defined for **ALL** possible states in space $\mathcal{S}$.
- For a fixed policy π, $v_\pi(s)$ quantifies the goodness of state s.
- For a fixed initial state s, $v_\pi(s)$ quantifies the goodness of policy π.
- For a stationanry MDP, $v_\pi(s)$ is independent of t. i.e. The state value of s remains the same regardless of when the agents arrives at s. Hence, we can assume without loss of generality that the agent arrived at s at $t = 0$. The state value then becomes

$$v_\pi(s) = \mathbb{E}\left[G_0 \mid S_0 = s\right]$$

Q-function (State-Action Value)

The ***Q-function*** (or ***state action value***, or simply ***action value***) for a certain policy π as follows

> $$q_\pi(s,a) = \mathbb{E}\left[G_t \mid S_t = s, A_t = a\right], \quad \forall s \in \mathcal{S}, \forall a \in \mathcal{A} \tag{6}$$

Relation between Q-function and state value for the same π:

- Interpretation: $q_\pi(s,a)$ represents the total reward of taking action a at initial state s and then following a policy π. vs. $v_\pi(s)$ represents the total reward of following π from s onward.

- For fixed π and s, if $q_\pi(s, a_1) > q_\pi(s, a_2)$, we say that a_1 is the better action to take over a_2 at state s.
- Compute $v_\pi(s)$ from $q_\pi(s, a)$: simply let $a = \pi(s)$, i.e.

$$v_\pi(s) = q_\pi(s, a)\Big|_{a=\pi(s)} \tag{7}$$

- Compute $q_\pi(s, a)$ from $v_\pi(s)$: use Bellman equation (will be proved later)

$$q_\pi(s, a) = r(s, a) + \gamma \mathbb{E}_{s' \sim p(\cdot|s,a)}[v_\pi(s')] \tag{8}$$

$$= r(s, a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s, a) v_\pi(s') \qquad \text{if } \mathcal{S} \text{ is finite} \tag{9}$$

Depending on the context, the term ***value function*** may refer to state value function or Q-function.

Bellman Equations

Bellman Equations for State Value

> Bellman Equations: The state values have the recursive structure
>
> $$v_\pi(s) = r(s, \pi(s)) + \gamma \mathbb{E}_{s' \sim p(\cdot|s,\pi(s))}[v_\pi(s')], \quad \forall s \in \mathcal{S} \tag{10}$$

Remarks:

- The expected total reward is the sum of immediate reward plus the (discounted) expected future reward.
  - $r(s, \pi(s))$: immediate reward at current state s by executing the policy π
  - $v_\pi(s')$: future reward from the next state s' by executing the policy π. $\mathbb{E}_{s' \sim p(\cdot|s,\pi(s))}[v_\pi(s')]$ is the average future reward over all possible s'.
- The current action $\pi(s)$ influences the immediate reward and the probability distribution of the next state s'.
- Bellman equations hold for all $s \in \mathcal{S}$. If $\mathcal{S}$ is finite, there are $|\mathcal{S}|$ Bellman equations.

Proof: Without loss of generality, assume $t = 0$. Using $G_0 = R_0 + \gamma G_1$, we get

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_{S_1, S_2, \dots}[G_0 \mid S_0 = s] \\ &= \mathbb{E}_{S_1, S_2, \dots}[R_0 + \gamma G_1 \mid S_0 = s] \\ &= \mathbb{E}_{S_1, S_2, \dots}[R_0 \mid S_0 = s] + \gamma \mathbb{E}_{S_1, S_2, \dots}[G_1 \mid S_0 = s] \\ &= r(s, \pi(s)) + \gamma \mathbb{E}_{S_1, S_2, \dots}[G_1 \mid S_0 = s] \end{aligned}$$

Using the law of total expectation and the markov properties (c.f. Appendix), we get

$$v_\pi(s) = r(s, \pi(s)) + \gamma \mathbb{E}_{S_1 \sim p(\cdot|s,\pi(s))}\left[\underbrace{\mathbb{E}_{S_2, S_3, \dots}[G_1 \mid S_1 = s_1]}_{v_\pi(s_1)}\right]$$

The underbraced term is the expected total reward by executing policy π starting from state s_1 which is by definition exactly the state value of $s_1 s_1$. Hence, we conclude. ■

For finite state space, the expected future reward in Bellman equation can be expressed in a sum. The Bellman equations become

$$v_\pi(s) = r(s, \pi(s)) + \gamma \sum_{s'} p(s' \mid s, \pi(s)) \cdot v_\pi(s'), \quad \forall s \in \mathcal{S} \tag{11}$$

Example: We model the state of a stock market as $\mathcal{S} = \{\text{bull}, \text{bear}, \text{flat}\}$. The agent uses some investment policy π (not detailed here) which yields the following state transition probability (Note that each row sums to 1).

$p(s' \mid s, \pi(s))$	$s' =$ bull	$s' =$ bear	$s' =$ flat
$s =$ bull	0.8	0.1	0.1
$s =$ bear	0.1	0.7	0.2
$s =$ flat	0	0.1	0.9

Suppose that immediate rewards under this policy are

$$r(\text{bull}, \pi(\text{bull})) = 8, \quad r(\text{bear}, \pi(\text{bear})) = -9, \quad r(\text{flat}, \pi(\text{flat})) = 2,$$

Then, the three Bellman equations are

$$\begin{aligned} v_\pi(\text{bull}) &= 8 + \gamma \left[0.8 v_\pi(\text{bull}) + 0.1 v_\pi(\text{bear}) + 0.1 v_\pi(\text{flat})\right] \\ v_\pi(\text{bear}) &= -9 + \gamma \left[0.1 v_\pi(\text{bull}) + 0.7 v_\pi(\text{bear}) + 0.2 v_\pi(\text{flat})\right] \\ v_\pi(\text{flat}) &= 2 + \gamma \left[0 \cdot v_\pi(\text{bull}) + 0.1 v_\pi(\text{bear}) + 0.9 v_\pi(\text{flat})\right] \end{aligned}$$

Reformulation in vector form:

$$\begin{bmatrix} v_\pi(\text{bull}) \\ v_\pi(\text{bear}) \\ v_\pi(\text{flat}) \end{bmatrix} = \begin{bmatrix} 8 \\ -9 \\ 2 \end{bmatrix} + \gamma \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.7 & 0.2 \\ 0 & 0.1 & 0.9 \end{bmatrix} \begin{bmatrix} v_\pi(\text{bull}) \\ v_\pi(\text{bear}) \\ v_\pi(\text{flat}) \end{bmatrix}$$

In general, let $\mathcal{S} = \{\varsigma_1, \ldots, \varsigma_n\}$. (To avoid confusion, we do not use $\{s_1, \ldots, s_n\}$ to denote $\mathcal{S}$ because the indices of s represent time.) Then, we can write n Bellman equations into the vector form

$$\underbrace{\begin{bmatrix} v_\pi(\varsigma_1) \\ v_\pi(\varsigma_2) \\ \vdots \\ v_\pi(\varsigma_n) \end{bmatrix}}_{\mathbf{v}_\pi} = \underbrace{\begin{bmatrix} r(\varsigma_1, \pi(\varsigma_1)) \\ r(\varsigma_2, \pi(\varsigma_2)) \\ \vdots \\ r(\varsigma_n, \pi(\varsigma_n)) \end{bmatrix}}_{\mathbf{r}_\pi} + \gamma \underbrace{\begin{bmatrix} p(\varsigma_1 \mid \varsigma_1, \pi(\varsigma_1)) & \ldots & p(\varsigma_n \mid \varsigma_1, \pi(\varsigma_1)) \\ p(\varsigma_1 \mid \varsigma_2, \pi(\varsigma_1)) & \ldots & p(\varsigma_n \mid \varsigma_2, \pi(\varsigma_1)) \\ \vdots & \ldots & \vdots \\ p(\varsigma_1 \mid \varsigma_n, \pi(\varsigma_1)) & \ldots & p(\varsigma_n \mid \varsigma_n, \pi(\varsigma_1)) \end{bmatrix}}_{\mathbf{P}_\pi} \cdot \underbrace{\begin{bmatrix} v_\pi(\varsigma_1) \\ v_\pi(\varsigma_2) \\ \vdots \\ v_\pi(\varsigma_n) \end{bmatrix}}_{\mathbf{v}_\pi}$$

Bellman equation (vector form)

$$\mathbf{v}_\pi = \mathbf{r}_\pi + \gamma \mathbf{P}_\pi \mathbf{v}_\pi \tag{12}$$

Remarks:

- $\mathbf{v}_\pi$ comprises state values for all $s \in \mathcal{S}$ under policy π
- $\mathbf{r}_\pi$ comprises immediate rewards arriving at $s \in \mathcal{S}$ under policy π
- $\mathbf{P}_\pi$ comprises all state transition probabilities under policy π
- All state values, immediate rewards and state transition probabilities depend on policy π.

Bellman Equation for Q-function

Similarly, Q-function also has recursive structure

$$q_\pi(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim p(\cdot|s,a)}[v_\pi(s')] \tag{13}$$
$$= r(s,a) + \gamma \mathbb{E}_{s' \sim p(\cdot|s,a)}[q_\pi(s', \pi(s'))] \tag{14}$$

Policy Evaluation

Given a policy π, computing its value function $v_\pi(\cdot)$ is called ***policy evaluation***. Effectively, we would like to evaluate how good π is for each state s. This is equivelent to solving Bellman equations. We will see later:

- If $\mathcal{S}$ is a finite set, solving Bellman equations boils down to solving a system of linear equations.
- If $\mathcal{S}$ is a infinite set, there is generally no closed-form solution for $v_\pi(s)$ expect for a few special cases (not covered here).

Computing State Values for Finite State Space

Analytical Solution

If $\mathcal{S}$ is fininte, policy evaluation boils down to solving $\mathbf{v}_\pi = \mathbf{r}_\pi + \gamma \mathbf{P}_\pi \mathbf{v}_\pi$ for $\mathbf{v}_\pi$. It is easy to verify that the analytical solution to the Bellman equations is

$$\mathbf{v}_\pi = (\mathbf{I} - \gamma \mathbf{P}_\pi)^{-1} \mathbf{r}_\pi \tag{15}$$

where $\mathbf{I}$ is the $|\mathcal{S}| \times |\mathcal{S}|$ identity matrix.

Remarks:

- The analytical solution is useful for theoretical study. Only practical when $|\mathcal{S}|$ is small.
- Drawback of analytical solution:
  - Requires matrix inversion. High computational complexity (nearly $\mathcal{O}(|\mathcal{S}|^3)$) when $|\mathcal{S}|$ is large.
  - No generalization to infinite state space as we can not pack pack all $v(s), s \in \mathcal{S}$ into a vector

Bellman Update

Algorithm to compute state values:

BELLMAN UPDATE (vector form)
Initialize $\mathbf{v}_0$ arbitrarily.
For $n = 0, 1, \ldots$, run until $\mathbf{v}_n$ converges

$$\mathbf{v}_{n+1} = \mathbf{r}_\pi + \gamma \mathbf{P}_\pi \mathbf{v}_n \tag{16}$$

The above algorithm can be reformulated element-wise as follows

BELLMAN UPDATE (element-wise form)
Init $v_0(s)$ for all $s \in \mathcal{S}$
For $n = 0, 1, \ldots$, run until $v_n(s)$ converges
    For each $s \in \mathcal{S}$, do

$$v_{n+1}(s) = r(s, \pi(s)) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s, \pi(s)) \cdot v_n(s')$$

Remarks:

- During iteration, $\mathbf{v}_n$ itself does not neccessarily satisfy Bellman equation for any policy.
- The sequence $\mathbf{v}_0, \mathbf{v}_1, \ldots$ obtained from Bellman update converges to $\mathbf{v}_\pi$, i.e.

$$\lim_{n \to \infty} \mathbf{v}_n = \mathbf{v}_\pi = (\mathbf{I} - \gamma \mathbf{P}_\pi)^{-1} \mathbf{r}_\pi \tag{17}$$

Proof of convergence: Define the affine function

$$f_\pi : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{S}|}, \mathbf{v} \mapsto \mathbf{r}_\pi + \gamma \mathbf{P}_\pi \mathbf{v}$$

We show that $f_\pi(\cdot)$ is a contractive mapping under infinity norm.

$$\begin{aligned} \|f_\pi(\mathbf{u}) - f_\pi(\mathbf{v})\|_\infty &= \|(\mathbf{r}_\pi + \gamma \mathbf{P}_\pi \mathbf{u}) - (\mathbf{r}_\pi + \gamma \mathbf{P}_\pi \mathbf{v})\|_\infty \\ &= \gamma \|\mathbf{P}_\pi (\mathbf{u} - \mathbf{v})\|_\infty \\ &\leq \gamma \|(\mathbf{u} - \mathbf{v})\|_\infty, \end{aligned}$$

The last step follows from the fact that $\|\mathbf{P}_\pi \mathbf{x}\|_\infty \leq \|\mathbf{x}\|_\infty, \forall x \in \mathbb{R}^{|\mathcal{S}|}$, i.e. multiplication with row stochastic matrix does not increase infnity norm. (c.f. Appendix).

By contraction mapping theorem (c.f. Appendix), we conclude that

1. $f_\pi(\cdot)$ has a unique fixed point. Since $\mathbf{v}_\pi = f_\pi(\mathbf{v}_\pi)$ by Bellman equation, $\mathbf{v}_\pi$ is the unique fixed point.
2. $\forall \mathbf{v} \in \mathbb{R}^{|\mathcal{S}|}$, the sequence of vectors $f_\pi^n(\mathbf{v})$ converges to $\mathbf{v}_\pi$ in infnity norm

Since all p-norms in $\mathbb{R}^{|\mathcal{S}|}$ are equivalent, convergence in infnity norm implies convergence in any p-norm. ■

Bellman Operator for Infinite State Space

Generalization of Bellman update to any $\mathcal{S}$ (possibly an infinite set). Instead of considering state values $v_\pi(s), \forall s \in \mathcal{S}$ as a vector in $\mathbb{R}^{|\mathcal{S}|}$, we consider the state value function $v_\pi(\cdot)$ as a "point" in the following function space.

Let $\mathcal{V}$ be the set of all **bounded** value functions.

$$\mathcal{V} = \{v : \mathcal{S} \to \mathbb{R} \;\Big|\; \|v\|_\infty < \infty\}$$

where the ***sup norm*** is defined as

$$\|v\|_\infty \triangleq \max_{s \in \mathcal{S}} |v(s)|$$

The ***sup norm metric*** $d : \mathcal{V} \times \mathcal{V} \to \mathbb{R}, (u, v) \mapsto d(u, v)$ is defined as

$$d(u, v) = \|u - v\|_\infty = \max_{s \in \mathcal{S}} |u(s) - v(s)|$$

One can verify that $d(\cdot, \cdot)$ satisfies the metric axioms and that $(\mathcal{V}, d)$ is a **complete** metric space.

For a certain policy π, we define the corresponding Bellman operator $\mathcal{B}_\pi$ which maps a state value function $v(\cdot)$ to a new value function $\mathcal{B}_\pi v(\cdot)$.

$$\mathcal{B}_\pi : \mathcal{V} \to \mathcal{V}, v \mapsto \mathcal{B}_\pi v$$

where the new value function is defined as

$$\mathcal{B}_\pi v(s) = r(s, \pi(s)) + \gamma \mathbb{E}_{s' \sim p(\cdot|s,\pi(s))}[v(s')]$$

Remarks:

- $\mathcal{B}_\pi v$ is well defined even though v itself might not satisfy Bellman equation for any policy. Nevertheless, if v happens to be the value function $v_{\tilde{\pi}}$ for some policy $\tilde{\pi}$. Then,

$$\mathcal{B}_\pi v_{\tilde{\pi}}(s) = q_{\tilde{\pi}}(s, \pi(s))$$

- We will see later that solving Bellman equation is equivalent to search for fixed point of $\mathcal{B}_\pi$ in function space $\mathcal{V}$.

Properties of Bellman operator:

1. $\mathcal{B}_\pi$ is monotonic, i.e.

$$u(s) \leq v(s), \forall s \in \mathcal{S} \implies \mathcal{B}_\pi u(s) \leq \mathcal{B}_\pi v(s), \forall s \in \mathcal{S}$$

2. $\mathcal{B}_\pi$ is a contractive mapping. i.e.

$$\forall u, v \in \mathcal{V},\ \|\mathcal{B}_\pi u - \mathcal{B}_\pi v\|_\infty \leq \gamma \|u - v\|_\infty$$

3. $v_\pi(\cdot)$ is the unique fixed point of $\mathcal{B}_\pi$, i.e.

$$\mathcal{B}_\pi v_\pi(s) = v_\pi(s), \forall s \in \mathcal{S}$$

Proof 1:
By the monotonicity of expectation
$\mathbb{E}_{s' \sim p(\cdot|s,\pi(s))}[u(s')] \leq \mathbb{E}_{s' \sim p(\cdot|s,\pi(s))}[v(s')]$, we conclude. ■

Proof 2:
Consider $|\mathcal{B}_\pi u(s) - \mathcal{B}_\pi v(s)|$ for all $s \in \mathcal{S}$.

$$\begin{aligned}
|\mathcal{B}_\pi u(s) - \mathcal{B}_\pi v(s)| &= \left| r(s, \pi(s)) + \gamma \mathbb{E}_{s' \sim p(\cdot|s,\pi(s))}[u(s')] - r(s, \pi(s)) - \gamma \mathbb{E}_{s' \sim p(\cdot|s,\pi(s))}[v(s')] \right| \\
&= |\gamma| \cdot \left| \mathbb{E}_{s' \sim p(\cdot|s,\pi(s))}[u(s') - v(s')] \right| \\
&\leq \gamma \cdot \mathbb{E}_{s' \sim p(\cdot|s,\pi(s))} \left[|u(s') - v(s')| \right] \\
&\leq \gamma \cdot \mathbb{E}_{s' \sim p(\cdot|s,\pi(s))} \left[\|u - v\|_\infty \right] \\
&\leq \gamma \cdot \|u - v\|_\infty
\end{aligned}$$

Hence, we conclude

$$\|\mathcal{B}_\pi u - \mathcal{B}_\pi v\|_\infty = \max_{s \in \mathcal{S}} |\mathcal{B}_\pi u(s) - \mathcal{B}_\pi v(s)| \leq \gamma \cdot \|u - v\|_\infty$$ ■

Proof 3:
By Bellman equation, we know that v_π is a fixed point of $\mathcal{B}_\pi$. The uniqueness follows from contraction mapping theorem and completeness of $\mathcal{V}$. ■

Hence, starting from any $v \in \mathcal{V}$ (which does not neccessarily need to satisfy Bellman equation for any policy). Repeatedly applying $\mathcal{B}_\pi$ on v leads to convergence to v_π. Formally,

$$\forall v \in \mathcal{V} : \lim_{n\to\infty} \|B_\pi^n v - v_\pi\|_\infty = 0$$

Using the fact that convergence in sup norm implies point-wise convergence(c.f. Appendix), we get

$$\forall v \in \mathcal{V} : \lim_{n\to\infty} B_\pi^n v(s) = v_\pi(s), \forall s \in \mathcal{S}$$

Let $v_n \triangleq \mathcal{B}_\pi^n v$. Then, the function sequence can be computed recursively

$$\begin{aligned} v_n(s) &= \mathcal{B}_\pi v_{n-1}(s) \\ &= r(s,\pi(s)) + \gamma \mathbb{E}_{s'\sim p(\cdot|s,\pi(s))}[v_{n-1}(s')] \end{aligned}$$

Remarks:

- For finite state space $\mathcal{S}$, the above equation reduces to Bellman updates shown in previous section

$$v_n(s) = r(s,\pi(s)) + \gamma \sum_{s'\in\mathcal{S}} p(s' \mid s,\pi(s)) \cdot v_{n-1}(s')$$

- For continuous state space $\mathcal{S}$, the Bellman operator requires computing an integral in $\mathcal{S}$, which is generally intractable to compute exactly. The primary purpose of introducing Bellman operator is for theoretical convergence analysis rather than direct algorithm design.

Bellman Optimality Equations

A policy π outperforms (or improves) another policy $\tilde{\pi}$ iff the state values $v_\pi(s)$ is nonless than $v_{\tilde{\pi}}(s)$ for **ALL** states $s \in \mathcal{S}$. i.e.

$$\forall s \in \mathcal{S},\ v_\pi(s) \geq v_{\tilde{\pi}}(s)$$

A policy π^* is optimal if it outperforms any other policies, i.e.

$$\forall \pi, \forall s \in \mathcal{S},\ v_{\pi^*}(s) \geq v_\pi(s)$$

The optimal state value $v^*(s)$ is the state value under π^*

$$v^*(s) \triangleq v_{\pi^*}(s) = \max_\pi v_\pi(s) \tag{18}$$

We haven't proved the existence of π^*. For now, let's assume its existence and discover what conditions have to be met for π^* and $v^*(\cdot)$. This will lead us to Bellman optimality equations, from which we will derive an algorithm to compute π^* (and thus prove its existence).

BOE for State Values

Recall the Bellman equation for $v_\pi(s)$ holds for any policy. In particular, Bellman equations also hold for π^*:

$$v^*(s) = r(s,\pi^*(s)) + \gamma \mathbb{E}_{s'\sim p(\cdot|s,\pi^*(s))}[v^*(s')], \quad \forall s \in \mathcal{S}$$

Yet, we are unable to solve the optimal state value since the optimal current action $\pi^*(s)$ is unknown. However, we can reformulate $v^*(s)$ without explicit reference to $\pi^*(s)$.

Trick: Note that $\pi^*(s)$ is the best action to take on current state s. $\implies$ Taking $\pi^*(s)$ now followed by executing π^* yields a state value no less than taking any other $a \in \mathcal{A}$ followed by exectuting π^*, as illustrated below.

- Optimal: Executing the optimal policy from now onward.

$$s \xrightarrow[r(s,\pi^*(s))]{\pi^*(s)} S_1 \xrightarrow[R_1]{\pi^*(S_1)} S_2 \xrightarrow[R_2]{\pi^*(S_2)} S_3 \cdots$$

- (Sub)optimal: Taking any other $a \in \mathcal{A} \setminus \{\pi^*(s)\}$ now, followed by executing π^* onward.

$$s \xrightarrow[r(s,a)]{a} S_1 \xrightarrow[R_1]{\pi^*(S_1)} S_2 \xrightarrow[R_2]{\pi^*(S_2)} S_3 \cdots$$

Formally, this means

$$\forall s \in \mathcal{S}, \forall a \in \mathcal{A}: \; v^*(s) \geq r(s,a) + \gamma \mathbb{E}_{s' \sim p(\cdot|s,a)}[v^*(s')]$$

where the equality holds iff $a = \pi^*(s)$. Namely, an optimal action at state s should maximize the sum of the immediate reward and the (discounted) expected future reward

$$\pi^*(s) = \arg\max_{a \in \mathcal{A}} \left\{ r(s,a) + \gamma \mathbb{E}_{s' \sim p(\cdot|s,a)}[v^*(s')] \right\}, \quad \forall s \in \mathcal{S} \tag{19}$$

Remarks:

- This equation holds for all $s \in \mathcal{S}$.
- There may be multiple actions that maximize the right-hand side. In such cases, we let $\pi^*(s)$ be any one of those optimizers. Hence, the optimal policy is **not unique** in general.
- If we have already solved optimal value function $v^*(\cdot)$, then plugging it into this equation yields an optimal policy.

The optimal state value $v^*(s)$ thus satisfies the ***Bellman optimality equations (BOE)***:

$$v^*(s) = \max_{a \in \mathcal{A}} \left\{ r(s,a) + \gamma \mathbb{E}_{s' \sim p(\cdot|s,a)}[v^*(s')] \right\}, \quad \forall s \in \mathcal{S} \tag{20}$$

Remarks:

- Previously, computing $v^*(s)$ requires knowledge of $r(s, \pi^*(s))$ and $p(\cdot \mid s, \pi^*(s))$, but since $\pi^*(s)$ is unknown, solving $v^*(s)$ is challenging.
- Now with the BOE, we bypass the need to know $\pi^*(s)$ explicitly. Instead, we evaluate $r(s,a)$ and $p(\cdot \mid s,a)$ (which are provided by MDP) for all $a \in \mathcal{A}$. Then, $v^*(s)$ can be solved by solving the optimization problem (detailed later).
- Just like Bellman equation, BOE holds for all $s \in \mathcal{S}$.

For finite state space, the BOE becomes

$$v^*(s) = \max_{a\in\mathcal{A}} \left\{ r(s,a) + \gamma \sum_{s'} p(s' \mid s,a) \cdot v^*(s') \right\}, \quad \forall s \in \mathcal{S} \tag{21}$$

Again, let $\mathcal{S} = \{\varsigma_1, \dots, \varsigma_n\}$. Then, we can write n BOEs into vector form

$$\underbrace{\begin{bmatrix} v^*(\varsigma_1) \\ v^*(\varsigma_2) \\ \vdots \\ v^*(\varsigma_n) \end{bmatrix}}_{\mathbf{v}^*} = \max_{a\in\mathcal{A}} \left\{ \underbrace{\begin{bmatrix} r(\varsigma_1,a) \\ r(\varsigma_2,a) \\ \vdots \\ r(\varsigma_n,a) \end{bmatrix}}_{\mathbf{r}_a} + \gamma \underbrace{\begin{bmatrix} p(\varsigma_1 \mid \varsigma_1,a) & \dots & p(\varsigma_n \mid \varsigma_1,a) \\ p(\varsigma_1 \mid \varsigma_2,a) & \dots & p(\varsigma_n \mid \varsigma_2,a) \\ \vdots & \dots & \vdots \\ p(\varsigma_1 \mid \varsigma_n,a) & \dots & p(\varsigma_n \mid \varsigma_n,a) \end{bmatrix}}_{\mathbf{P}_a} \cdot \underbrace{\begin{bmatrix} v^*(\varsigma_1) \\ v^*(\varsigma_2) \\ \vdots \\ v^*(\varsigma_n) \end{bmatrix}}_{\mathbf{v}^*} \right\}$$

where $\max$ is taken element-wise.

$$\max_a \begin{bmatrix} f_1(a) \\ \vdots \\ f_n(a) \end{bmatrix} = \begin{bmatrix} \max_a f_1(a) \\ \vdots \\ \max_a f_n(a) \end{bmatrix} = \begin{bmatrix} f_1(\hat{a}_1) \\ \vdots \\ f_n(\hat{a}_n) \end{bmatrix}$$

Hence, we get BOE in vector form

$$\mathbf{v}^* = \max_{a\in\mathcal{A}} \{ \mathbf{r}_a + \gamma \mathbf{P}_a \mathbf{v}^* \} \tag{22}$$

BOE for Q-function

Similary, the optimal Q-function is defined as

$$q^*(s,a) = q_{\pi^*}(s,a)$$

Relation between optimal Q-function and optimal state value:

- Compute $v^*(s)$ from $q^*(s,a)$: simply let $a = \pi^*(s)$, i.e.

$$v^*(s) = q^*(s,a)\Big|_{a=\pi^*(s)} \tag{23}$$

- Compute $q^*(s,a)$ from $v^*(s)$: use recurisve structure

$$\begin{aligned} q^*(s,a) &= r(s,a) + \gamma \mathbb{E}_{s'\sim p(\cdot|s,a)}[v^*(s')] && (24) \\ &= r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s' \mid s,a) v^*(s') \qquad \text{if } \mathcal{S} \text{ is finite} && (25) \end{aligned}$$

The Bellman optimality criterion can also be formulated in terms of Q-function:

$$\begin{aligned} v^*(s) &= \max_{a\in\mathcal{A}} q^*(s,a) && (26) \\ \pi^*(s) &= \arg\max_{a\in\mathcal{A}} q^*(s,a) && (27) \end{aligned}$$

Dynamic Programming

Given the parameters of an MDP:

- state transition probabilities $p(s' \mid s, a)$ for all $s, s' \in \mathcal{S}, a \in \mathcal{A}$
- state-action-rewards $r(s, a)$ for all $s \in \mathcal{S}, a \in \mathcal{A}$

How to compute the optimal policy?
$\rightarrow$ Dynamic programming

Value Iteration

The recurrent structure in BOE motivates us to define the Bellman optimality operator, which is at the core of value iteration. We will show that iteration over Bellman optimality operator leads to the optimal value function. (Just like iterating over Bellman operator for a certain policy leads to the value function for that policy)

Bellman Optimality Operator

Recall the metric space $(\mathcal{V}, d)$ of bounded value functions equipped with sup norm metirc.

Analogous to Bellman operator for a certain policy, we define the ***Bellman optimality operator*** $\mathcal{B}_*$ as

$$\mathcal{B}_* : \mathcal{V} \to \mathcal{V}, v(\cdot) \mapsto \mathcal{B}_* v(\cdot)$$

where

$$\mathcal{B}_* v(s) = \max_{a \in \mathcal{A}} \left\{ r(s, a) + \gamma \mathbb{E}_{s' \sim p(\cdot \mid s, a)} [v(s')] \right\}$$

Remarks:

- If v happens to be the value function v_π for some policy π. Then,

$$\mathcal{B}_* v_\pi(s) = \max_{a \in \mathcal{A}} q_\pi(s, a)$$

- We will see later that solving BOE amounts to finding a fixed point of $\mathcal{B}_*$ in function space $\mathcal{V}$. However, the primary purpose of introducing $\mathcal{B}_*$ is for theoretical convergence analysis rather than direct algorithm design.

Properties of Bellman optimality operator:

1. $\mathcal{B}_*$ is monotonic, i.e.

$$u(s) \leq v(s), \forall s \in \mathcal{S} \implies \mathcal{B}_* u(s) \leq \mathcal{B}_* v(s), \forall s \in \mathcal{S}$$

2. $\mathcal{B}_*$ is a contractive mapping. i.e.

$$\forall u, v \in \mathcal{V},\ \|\mathcal{B}_* u - \mathcal{B}_* v\|_\infty \leq \gamma \|u - v\|_\infty$$

3. $v_*(\cdot)$ is the unique fixed point of $\mathcal{B}_*$, i.e.

$$\mathcal{B}_* v_*(s) = v_*(s), \forall s \in \mathcal{S}$$

Proof 1 and 3: Same as the proof for $\mathcal{B}_\pi$. ■

Proof 2: We show that $\forall s \in \mathcal{S}$: $|B_* u(s) - \mathcal{B}_* v(s)| \leq \gamma \|u - v\|_\infty$ as follows

$$
\begin{aligned}
|B_* u(s) - \mathcal{B}_* v(s)| &= \left| \max_{a \in \mathcal{A}} \left\{ r(s,a) + \gamma \mathbb{E}_{s' \sim p(\cdot|s,a)}[u(s')] \right\} - \max_{a \in \mathcal{A}} \left\{ r(s,a) + \gamma \mathbb{E}_{s' \sim p(\cdot|s,a)}[v(s')] \right\} \right| \\
&\leq \max_{a \in \mathcal{A}} \left| \left\{ r(s,a) + \gamma \mathbb{E}_{s' \sim p(\cdot|s,a)}[u(s')] \right\} - \left\{ r(s,a) + \gamma \mathbb{E}_{s' \sim p(\cdot|s,a)}[v(s')] \right\} \right| \\
&= \gamma \max_{a \in \mathcal{A}} \left| \mathbb{E}_{s' \sim p(\cdot|s,a)}[u(s') - v(s')] \right| \\
&\leq \gamma \max_{a \in \mathcal{A}} \mathbb{E}_{s' \sim p(\cdot|s,a)} \left[|u(s') - v(s')| \right] \\
&\leq \gamma \max_{a \in \mathcal{A}} \mathbb{E}_{s' \sim p(\cdot|s,a)} \left[\|u - v\|_\infty \right] \\
&= \gamma \|u - v\|_\infty
\end{aligned}
$$

where the 2nd step follows from the fact (c.f. Appendix) that

$$
\left| \max_{x \in \mathcal{X}} f(x) - \max_{x \in \mathcal{X}} g(x) \right| \leq \max_{x \in \mathcal{X}} |f(x) - g(x)| \qquad \blacksquare
$$

Once again, starting from any $v \in \mathcal{V}$ (which does not neccessarily need to satisfy Bellman equation for any policy), repeatedly applying $\mathcal{B}_*$ leads convergence to the optimal value function v_*.

$$
\forall v \in \mathcal{V} : \lim_{n \to \infty} \|B_*^n v - v_*\|_\infty \implies \lim_{n \to \infty} B_*^n v(s) = v_*(s)
$$

Let $v_n \triangleq \mathcal{B}_*^n v$. Then, v_n can be iteratively computed as follows (called ***value iteration***)

$$
\begin{aligned}
v_n(s) &= \mathcal{B}_* v_{n-1}(s) && (28) \\
&= \max_{a \in \mathcal{A}} \left\{ r(s,a) + \gamma \mathbb{E}_{s' \sim p(\cdot|s,a)}[v_{n-1}(s')] \right\} && (29) \\
&= \max_{a \in \mathcal{A}} \left\{ r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s,a)[v_{n-1}(s')] \right\} \qquad \text{if } \mathcal{S} \text{ is finite} && (30)
\end{aligned}
$$

Having computed the optimal value function v^*, the optimal policy is obtained from

$$
\pi^*(s) = \arg\max_{a \in \mathcal{A}} \left\{ r(s,a) + \gamma \mathbb{E}_{s' \sim p(\cdot|s,a)}[v^*(s')] \right\}
$$

The Algorithm

Now, we present value iteration algorithm for finite state space.

> **VALUE ITERATION (vector form)**
> Init $\mathbf{v}_0$ arbitrarily
> For $n = 0, 1, \ldots,$ do
> **Policy update**: $\boldsymbol{\pi}_{n+1} = \arg\max_{a \in \mathcal{A}} \{ \mathbf{r}_a + \gamma \mathbf{P}_a \mathbf{v}_n \}$
> **Value update**: $\mathbf{v}_{n+1} = \max_{a \in \mathcal{A}} \{ \mathbf{r}_a + \gamma \mathbf{P}_a \mathbf{v}_n \}$
> where $\arg\max_{a \in \mathcal{A}}$ and $\max_{a \in \mathcal{A}}$ are taken row by row.
> until $\|\mathbf{v}_n - \mathbf{v}_{n-1}\|_\infty < \epsilon$
> Return $\mathbf{v}_n$ and $\boldsymbol{\pi}_n$.

Equivalent reformulation in element-wise form

VALUE ITERATION (element-wise form)
Init $v_0(s)$ for all $s \in \mathcal{S}$ arbitrarily
For $n = 0, 1, \ldots$, do
  For each $s \in \mathcal{S}$, do
    For each $a \in \mathcal{A}$, do
      compute: $q_n(s,a) = r(s,a) + \gamma \sum_{s'} p(s' \mid s,a) \cdot v_n(s')$
    Policy update: $\pi_{n+1}(s) = \arg\max_{a \in \mathcal{A}} q_n(s,a)$
    Value update: $v_{n+1}(s) = \max_{a \in \mathcal{A}} q_n(s,a)$
until $\|v_n - v_{n-1}\|_\infty < \epsilon$
Return $v_n(s)$ and $\pi_n(s)$ for all $s \in \mathcal{S}$

Remarks:

- In vector form, $\boldsymbol{\pi}_n$ consists of $\pi(s)$ for all $s \in \mathcal{S}$.
- The sequence $v_n(\cdot)$ converges to $v^*(\cdot)$ which satisfies BOE. However, $v_n(\cdot)$ itself do **not** generally satisfy Bellman equation for **any** policy. $v_n(s)$ should be interpreted as the estimate of $v^*(s)$ at n-th iteration rather than the state values under some policy. In particular,

$$
\begin{aligned}
&v_n(s) \neq v_{\pi_n}(s) \\
&v_n(s) \neq r(s,\pi_n(s)) + \gamma \sum_{s'} p(s' \mid s, \pi_n(s)) v_n(s) = \mathcal{B}_{\pi_n} v_n(s)
\end{aligned}
$$

- Likewise, $q_n(s,a)$ represents the estimate of $q^*(s,a)$ at n-th iteration rather than the real Q-function for any policy. In particular, $q_n(s,a) \neq q_{\pi_n}(s,a)$

Policy Iteration

Policy iteration is another algorithm to compute optimal policy. It starts with abitrary policy and iteratively improves it. The algorithm is backed by policy improvement theorem.

Policy Improvement Theorem

Let π and π' be two policies s.t.

$$
\forall s \in \mathcal{S} : q_\pi(s, \pi'(s)) \geq q_\pi(s, \pi(s))
$$

Then, π' is an improvement of π, i.e.

$$
\forall s \in \mathcal{S}, v_{\pi'}(s) \geq v_\pi(s)
$$

Proof: Using the definition of Q-functions and Bellman operator, we get

$$
\begin{aligned}
q_\pi(s, \pi'(s)) &\geq q_\pi(s, \pi(s)) \\
r(s,\pi'(s)) + \mathbb{E}_{s' \sim p(\cdot \mid s, \pi'(s))}[v_\pi(s')] &\geq v_\pi(s) \\
\mathcal{B}_{\pi'} v_\pi(s) &\geq v_\pi(s)
\end{aligned}
$$

By induction, we have

$$
\mathcal{B}_{\pi'}^n v_\pi(s) \geq v_\pi(s) \tag{$\star$}
$$

Recall that $\mathcal{B}_{\pi'}^n$ converges to $v_{\pi'}$ for any $v \in \mathcal{V}$. Taking the limit in $(\star)$, we get

$$\lim_{n\to\infty} \mathcal{B}_{\pi'}^n v_\pi(s) = v_{\pi'}(s) \geq v_\pi(s) \qquad \blacksquare$$

Greedy Policy

Given a poliy π, how to construct a better policy π'? A natural approach would be for each s, pick a, such that $q_\pi(s,\cdot)$ is maximized. The resulting called greedy policy.

A policy π' is called ***greedy*** w.r.t. q_π if it is constructed as

$$\forall s \in \mathcal{S},\ \pi'(s) = \arg\max_a q_\pi(s,a) \tag{31}$$

$$= \arg\max_a \Big\{ r(s,a) + \gamma \mathbb{E}_{s'\sim p(\cdot|s,a)}[v_\pi(s')] \Big\} \tag{32}$$

> **Fact**: If π' is greedy w.r.t. q_π, then it improves the original policy π , i.e.
>
> $$\forall s \in \mathcal{S},\ \pi'(s) = \arg\max_a q_\pi(s,a) \implies v_{\pi'} \geq v_\pi \tag{33}$$

Proof: By construction of π', we have

$$\forall s \in \mathcal{S}, \forall a \in \mathcal{A},\ q_\pi(s,\pi'(s)) \geq q_\pi(s,a)$$

In particular,

$$\forall s \in \mathcal{S},\ q_\pi(s,\pi'(s)) \geq q_\pi(s,\pi(s))$$

By policy improvement theorem, we conclude that $v_{\pi'} \geq v_\pi$. $\blacksquare$

Suppose we improved π by constructing a greedy policy π'. We can repeat this process to improve π' and get π''. Keep doing so is called ***policy iteration***. The question is whether policy iteration will lead to an optimal policy. The answer is yes due to the following fact.

> Let π_0 be any policy. Construct a sequence of greedy policy
>
> $$\pi_{n+1}(s) = \arg\max_{a\in\mathcal{A}} q_{\pi_n}(s,a), \quad \forall s \in \mathcal{S}$$
>
> Then, the corresponding sequence of value functions $(v_{\pi_n})_{n\in\mathbb{N}}$ converges to the optimal value function v^*.

Proof: By construction, π_{n+1} is greedy w.r.t. q_{π_n} for all $n \in \mathbb{N}$. By property of greedy policy, we have $v_{\pi_{n+1}} \geq v_{\pi_n}$, i.e. The sequence of functions v_{π_n} monotonically increases.

Moreover, all v_{π_n} are bounded by v^*. By monotonic convergence theorem (c.f. Appendix), v_{π_n} converges to some $\bar{v} \in \mathcal{V}$ with $\bar{v} \leq v^*$. To show that $\bar{v} = v^*$, it remains to show that $v^* \leq \bar{v}$.

Note that we can express $q_{\pi_n}(s,\pi_{n+1}(s))$ in two differenet ways:

$$\begin{aligned} q_{\pi_n}(s,\pi_{n+1}(s)) &= \max_a \Big\{ r(s,a) + \gamma \mathbb{E}_{s'\sim p(\cdot|s,a)}[v_{\pi_n}(s')] \Big\} = \mathcal{B}_* v_{\pi_n}(s) \\ &= r(s,\pi_{n+1}(s)) + \gamma \mathbb{E}_{s'\sim p(\cdot|s,\pi_{n+1}(s))}[v_{\pi_n}(s')] = \mathcal{B}_{\pi_{n+1}} v_{\pi_n}(s) \end{aligned}$$

Namely, $\mathcal{B}_* v_{\pi_n}(s) = q_{\pi_n}(s,\pi_{n+1}(s)) = \mathcal{B}_{\pi_{n+1}} v_{\pi_n}(s), \forall s \in \mathcal{S}$.

By monotonicity and fixed point property of $\mathcal{B}_{\pi_{n+1}}$,

$$\boxed{\mathcal{B}_* v_{\pi_n}} = \mathcal{B}_{\pi_{n+1}} v_{\pi_n} \leq \mathcal{B}_{\pi_{n+1}} v_{\pi_{n+1}} = \boxed{v_{\pi_{n+1}}}$$

By induction, we get

$$\mathcal{B}_*^{n+1} v_{\pi_0} \leq v_{\pi_{n+1}}$$

Taking the limit on both sides, we conclude.

$$\boxed{v^*} = \lim_{n\to\infty} \mathcal{B}_*^{n+1} v_{\pi_0} \leq \lim_{n\to\infty} v_{\pi_{n+1}} = \boxed{\bar{v}}$$ ■

The Algorithm

Now, we present policy iteration in pseudocode for finite state space. The greedy policy construction requires computing q_{π_n}, which depends on v_{π_n}. Thus, we must first perform policy evaluation to compute v_{π_n} before improving π_n.

POLICY ITERATION(vector form)
Init π_0 arbitrarily
For $n = 0, 1, 2, \ldots$, do
  Policy evaluation: Compute $\mathbf{v}_{\pi_n}$ by solving $\mathbf{v}_{\pi_n} = \mathbf{r}_{\pi_n} + \gamma \mathbf{P}_{\pi_n} \mathbf{v}_{\pi_n}$
  Policy improvement: Compute greedy policy $\boldsymbol{\pi}_{n+1} = \arg\max_{a\in\mathcal{A}} \{\mathbf{r}_a + \gamma \mathbf{P}_a \mathbf{v}_{\pi_n}\}$
until $\|\mathbf{v}_{\pi_n} - \mathbf{v}_{\pi_{n-1}}\|_\infty < \epsilon$
Return $\mathbf{v}_{\pi_n}$ and $\boldsymbol{\pi}_n$

Equivalent element-wise formutaiton of policy iteration:

POLICY ITERATION(element-wise form)
Init $\pi_0(s)$ for all $s \in \mathcal{S}$ arbitrarily
For $n = 0, 1, 2, \ldots$, do
  Policy evaluation: Compute $v_{\pi_n}(s)$ for all $s \in \mathcal{S}$ given π_n.
  For each $s \in \mathcal{S}$, do
    For each $a \in \mathcal{A}$, do
      Compute Q-function: $q_{\pi_n}(s,a) = r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s' \mid s,a)\, v_{\pi_n}(s')$
    Policy improvement: compute greedy policy $\pi_{n+1}(s) = \arg\max_{a\in\mathcal{A}} q_{\pi_n}(s,a)$
until $\|v_{\pi_n} - v_{\pi_{n-1}}\|_\infty < \epsilon$
Return $v_{\pi_n}(s)$ and $\pi_n(s)$ for all $s \in \mathcal{S}$

Remarks:

- In policy evaluation, either analytical solution (for small state space) or Bellman update (for large state space) is used. The analytical solution provides exact state values of $v_{\pi_n}(s)$ while Bellman updates gives approximation of $v_{\pi_n}(s)$. We will see later that the approximation error in Bellman update does not prevent the algorithm from converging.

Comparison of Value Iteration and Policy Iteration

Suppose the value iteration starts with $v_0 = v_{\pi_0}$ where π_0 is the initial guess of policy iteration. Assume that policy evaluation always gives the exact solution.

	Policy Iteration	**Value Iteration**
1	$\texttt{init}: \pi_0(\cdot)$	$-$
2	$\texttt{PE}: v_{\pi_0} = \mathcal{B}_{\pi_0} v_{\pi_0}$	$\texttt{init}: v_0 = v_{\pi_0}$
3	$\texttt{PI}: \pi_1(s) = \arg\max_a q_{\pi_0}(s,a), \forall s \in \mathcal{S}.$	$\texttt{PU}: \pi_1(s) = \arg\max_a q_0(s,a), \forall s \in \mathcal{S}.$
$\boxed{4}$	$\texttt{PE}: v_{\pi_1} = \mathcal{B}_{\pi_1} v_{\pi_1}$	$\texttt{VU}: v_1 = \mathcal{B}_* v_0$
5	$\texttt{PI}: \pi_2(s) = \arg\max_a q_{\pi_1}(s,a), \forall s \in \mathcal{S}.$	$\texttt{PU}: \pi_2(s) = \arg\max_a q_1(s,a), \forall s \in \mathcal{S}.$
$\vdots$	$\vdots$	$\vdots$

Since $v_0 = v_{\pi_0}$, policy iteration and value iteration coincide from steps 1 to step 3. The step 4 is the key difference between the two algorithms and is anaylsed as follows.

- In policy iteration, π_1 is greedy w.r.t. q_{π_0}. By property of greedy policy, we have $v_{\pi_1} \geq v_{\pi_0}$. Hence,

$$v_{\pi_1}(s) = \mathcal{B}_{\pi_1} v_{\pi_1}(s) \geq \mathcal{B}_{\pi_1} v_{\pi_0}(s) = q_{\pi_0}(s, \pi_1(s))$$

- In value iteration, $v_0 = v_{\pi_0}$, $q_0 = q_{\pi_0}$. Step 3 and 4 yield

$$v_1(s) = \mathcal{B}_* v_0(s) = \max_a q_0(s,a) = \max_a q_{\pi_0}(s,a) = q_{\pi_0}(s, \pi_1(s))$$

Hence, $v_{\pi_1} \geq v_1$.

> **Claim**: Starting with $v_0 = v_{\pi_0}$, policy iteration converges faster than value iteration, i.e.
>
> $$v_{\pi_n} \geq v_n, \forall n \geq 1$$

Proof: The base case is already shown. Now suppose the inequality holds for any n. We will show $v_{\pi_{n+1}} \geq v_{n+1}$.
By induction hypothesis, we have $\forall s \in \mathbb{S}, \forall a \in \mathcal{A}$:

$$q_{\pi_n}(s,a) = r(s,a) + \mathbb{E}[v_{\pi_n}(s')] \geq r(s,a) + \mathbb{E}[v_n(s')] = q_n(s,a)$$

Maximizing q_{π_n} and q_n over a yields

$$\begin{aligned} \max_a q_{\pi_n}(s,a) &\geq \max_a q_n(s,a) \\ q_{\pi_n}(s, \pi_{n+1}(s)) &\geq v_{n+1}(s) \end{aligned}$$

Using the property of greedy policy, we conclude the inequality for $n+1$:

$$v_{\pi_{n+1}}(s) = \mathcal{B}_{\pi_{n+1}} v_{\pi_{n+1}}(s) \geq \mathcal{B}_{\pi_{n+1}} v_{\pi_n}(s) = q_{\pi_n}(s, \pi_{n+1}(s)) \geq v_{n+1}(s)$$ ■

In the previous dicussion, we assumed that policy evaluation yields the exact solution. In practice, policy evalution is performed by iterative Bellman updates which gives approximate solution. In particular, consider policy evaluation with a single-step Bellman update:

$$\tilde{v}_{\pi_n} = \mathcal{B}_{\pi_n} \tilde{v}_{\pi_{n-1}}, \quad \forall n \geq 1$$

This leads to the single-step truncated policy iteration as follows

	Policy Iteration (single-step truncated)	**Value Iteration**
1	`init` : $\pi_0(\cdot)$	$-$
2	`PE` : $\tilde{v}_{\pi_0} = \mathcal{B}_{\pi_0}\tilde{v} \quad \text{for some } \tilde{v} \in \mathcal{V}$	`init` : $v_0 = \tilde{v}_{\pi_0}$
3	`PI` : $\pi_1(s) = \arg\max_a \tilde{q}_{\pi_0}(s,a), \forall s \in \mathcal{S}.$	`PU` : $\pi_1(s) = \arg\max_a q_0(s,a), \forall s \in \mathcal{S}.$
$\boxed{4}$	`PE` : $\tilde{v}_{\pi_1} = \mathcal{B}_{\pi_1}\tilde{v}_{\pi_0}$	`VU` : $v_1 = \mathcal{B}_* v_0$
5	`PI` : $\pi_2(s) = \arg\max_a \tilde{q}_{\pi_1}(s,a), \forall s \in \mathcal{S}.$	`PU` : $\pi_2(s) = \arg\max_a q_1(s,a), \forall s \in \mathcal{S}.$
$\vdots$	$\vdots$	$\vdots$

Remarks:

- Here, $\tilde{v}_{\pi_n}$ no longer satisfies the Bellman equation for π_n. Rather, it is an approximation of v_{π_n}. Likewise, $\tilde{q}_{\pi_n}$ is an approximation of the true Q-function q_{π_n}, defined as

$$\tilde{q}_{\pi_n}(s,a) = r(s,a) + \gamma \mathbb{E}[\tilde{v}_{\pi_n}(s')]$$

- In policy evaluation, we take $\tilde{v}_{\pi_{n-1}}$ as the initial guess and apply Bellman operator $\mathcal{B}_{\pi_n}$ only once.

> **Claim**: Starting with $v_0 = \tilde{v}_{\pi_0}$, single-step truncated policy iteration coincides with value iteration, i.e.
>
> $$\tilde{v}_{\pi_n} = v_n, \forall n \geq 0$$

Proof: We show this claim by induction. The base case for $n = 0$ is true by assumption. Now, suppose $\tilde{v}_{\pi_n} = v_n$, we will show that $\tilde{v}_{\pi_{n+1}} = v_{n+1}$.

By induction hypothesis, both algorithms share the same Q-function as

$$\tilde{q}_{\pi_n}(s,a) = r(s,a) + \gamma \mathbb{E}[\tilde{v}_{\pi_n}(s')] = r(s,a) + \gamma \mathbb{E}[\tilde{v}_n(s')] = q_n(s,a)$$

Thus, both algorithms select the same policy π_{n+1}, since $\arg\max_a \tilde{q}_{\pi_n}(s,a) = \arg\max_a q_n(s,a)$.

- In policy iteration:

$$\tilde{v}_{\pi_{n+1}}(s) = \mathcal{B}_{\pi_{n+1}}\tilde{v}_{\pi_n}(s) = \tilde{q}_{\pi_n}(s, \pi_{n+1}(s)) = q_n(s, \pi_{n+1}(s))$$

- In value iteration:

$$v_{n+1}(s) = \mathcal{B}_* v_n(s) = \max_a q_n(s,a) = q_n(s, \pi_{n+1}(s))$$

Theorefore, we conclude that $\tilde{v}_{\pi_{n+1}} = v_{n+1}$. ■

Truncated Policy Iteration

In policy evaluation, what if we perform an m-step Bellman updates instead of a single-step Bellman update? This yields ***truncated policy iteration***.

	Truncated Policy Iteration	**Value Iteration**
1	$\texttt{init}: \pi_0(\cdot)$	$-$
2	$\texttt{PE}: \tilde{v}_{\pi_0} = \mathcal{B}_{\pi_0}^m \tilde{v} \quad \text{for some } \tilde{v} \in \mathcal{V}$	$\texttt{init}: v_0 = \tilde{v}_{\pi_0}$
3	$\texttt{PI}: \pi_1(s) = \arg\max_a \tilde{q}_{\pi_0}(s,a), \forall s \in \mathcal{S}.$	$\texttt{PU}: \pi_1(s) = \arg\max_a q_0(s,a), \forall s \in \mathcal{S}.$
$\boxed{4}$	$\texttt{PE}: \tilde{v}_{\pi_1} = \mathcal{B}_{\pi_1}^m \tilde{v}_{\pi_0}$	$\texttt{VU}: v_1 = \mathcal{B}_* v_0$
5	$\texttt{PI}: \pi_2(s) = \arg\max_a \tilde{q}_{\pi_1}(s,a), \forall s \in \mathcal{S}.$	$\texttt{PU}: \pi_2(s) = \arg\max_a q_1(s,a), \forall s \in \mathcal{S}.$
$\vdots$	$\vdots$	$\vdots$

Remarks:

- If $m = 1$, truncated policy iteration becomes value update.
- As m increases, $\tilde{v}_{\pi_n}$ provides a better approximation of v_{π_n}. Truncated policy iteration behaves more similarly to the standard policy iteration.
- If $m \to \infty$, truncated policy iteration becomes standard policy iteration.
- In practice, policy iteration is typically implemented as truncated policy iteration because policy evaluation must be performed in a finite number of steps.

> **Summary**: Starting from the same initial condition,
>
> 1. Policy iteration generally converges faster than value iteration.
> 2. Policy iteration becomes equivalent to value iteration when policy evaluation is performed using a single-step Bellman update at each iteration.
> 3. Both policy iteration and value iteration can be seen as special cases of truncated policy iteration.

Appendix

Contractive Mapping

Let $(\mathcal{X}, d)$ be a metric space. We say that $f: \mathcal{X} \to \mathcal{X}$ is a ***contractive*** mapping if

$$\exists \gamma \in [0,1), \text{ s.t. } \forall x, y \in \mathcal{X}, d(f(x), f(y)) \leq \gamma d(x,y)$$

> **Contraction Mapping Theorem**
> If $(\mathcal{X}, d)$ is **complete** and $f: \mathcal{X} \to \mathcal{X}$ is contractive, then
>
> 1. f has unique fixed point. i.e. $\exists!\, x^* \in \mathcal{X}$ s.t. $f(x^*) = x^*$
> 2. Construction of the fixed point: $\forall x \in \mathcal{X}, \lim_{n \to \infty} f^n(x) = x^*$

Proof: c.f. Separete notes.

> **Montonic Contraction**
> Suppose that $(\mathcal{X}, d)$ is a complete metric space with a **partial oder** "$\leq$". Let $f: \mathcal{X} \to \mathcal{X}$ be a **monotonic increasing** contraction mapping w.r.t. "$\leq$". Then,

$$\forall x : x \leq x^* \iff x \leq f(x)$$

Proof: The claim is trivially true if $x = x^*$. Hence, we assume $x \neq x^*$ in the following.

$\Longleftarrow$:
Using the monotonicity of f, we get the induction

$$\boxed{x} \leq f(x) \leq f^2(x) \leq \cdots \leq \boxed{f^n(x)}, \quad \forall x \in \mathbb{N}$$

Taking the limit, we conclude $x \leq \lim_{n\to\infty} f^n(x) = x^*$. ■

$\Longrightarrow$:
For the sake of contradiction, suppose $\exists x : x \leq x^*$ but $x > f(x)$. By induction,

$$x > f^n(x), \quad \forall x \in \mathbb{N}$$

Taking the limit, we get $x \geq \lim_{n\to\infty} f^n(x) = x^*$. Since $x \neq x^*$ by assumption, we get $x > x^*$ which contradicts with $x \leq x^*$. ■

Dealing with Max and Abs

Let $f, g : \mathcal{X} \to \mathbb{R}$ be any two real-valued functions, where $\mathcal{X}$ can be any set. Then,

$$\left|\max_{x\in\mathcal{X}} f(x) - \max_{x\in\mathcal{X}} g(x)\right| \leq \max_{x\in\mathcal{X}} |f(x) - g(x)|$$

Proof: Let $M_f = \max_{x\in\mathcal{X}} f(x)$ and $M_g = \max_{x\in\mathcal{X}} g(x)$. Without loss of generality, assume that $M_f \geq M_g$. Then, $\forall x \in \mathcal{X}$:

$$\boxed{|M_f - M_g|} = M_f - M_g \leq M_f - g(x) = \boxed{|M_f - g(x)|}$$

Let $\hat{x}$ be the maximizer of f, i.e. $M_f = f(\hat{x})$. We conclude

$$|M_f - M_g| \leq |M_f - g(\hat{x})| = |f(\hat{x}) - g(\hat{x})| \leq \max_{x\in\mathcal{X}} |f(x) - g(x)|$$ ■

Cascade of Expectations

Suppose $U - X - Y$ forms a markov chain, then

$$\begin{aligned}\mathbb{E}_{XY}[g(x,y) \mid u] &\triangleq \mathbb{E}_{XY\sim p(x,y|u)}[g(x,y)] \\ &= \mathbb{E}_{X\sim p(x|u)}\left[\mathbb{E}_{Y\sim p(y|x)}[g(x,y)]\right]\end{aligned}$$

Row Stochastic Matrix

A square matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is called a ***row stochastic matrix*** iff each row of $\mathbf{A}$ is a probability vector. i.e.

- all its entires are nonnegative: $a_{ij} \geq 0, \forall i, j \in \{1, \ldots, n\}$
- and each row sums to 1. $\sum_{j=1}^{n} a_{ij} = 1, \forall i \in \{1, \ldots, n\}$

Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be a state transition matrix. Then,

1. 1 is always an eigenvalue of $\mathbf{A}$.
2. Multiplication with $\mathbf{A}$ does not increase the infinity norm. i.e.

$$\|\mathbf{A}\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_\infty, \forall \mathbf{x} \in \mathbb{R}^n \tag{34}$$

3. The eigenvalues of $\mathbf{A}$ are at most 1. i.e.

$$|\lambda| \leq 1, \forall \lambda \in \text{spec}(A) \tag{35}$$

Proof 1: Let $\mathbf{u} = [1, \dots, 1]^\top \in \mathbb{R}^n$ be all-one vector. It is easy to verify that $\mathbf{A}\mathbf{u} = \mathbf{u}$. Hence, $\mathbf{u}$ is an eigenvector of $\mathbf{A}$ with eigen value 1. ■

Proof 2: Recall the infinity norm is defined by

$$\|\mathbf{x}\|_\infty \triangleq \max_{i=1,\dots,n} |x_i|$$

Let $\mathbf{y} = \mathbf{A}\mathbf{x}$. Consider the abs of i-th element of $\mathbf{y}$:

$$|y_i| = \left| \sum_{j=1}^n a_{ij} x_j \right| \leq \sum_{j=1}^n \left| a_{ij} x_j \right| = \sum_{j=1}^n a_{ij} |x_j| \leq \sum_{j=1}^n a_{ij} \|\mathbf{x}\|_\infty = \|\mathbf{x}\|_\infty$$

$\implies$ All elements of $\mathbf{y}$ are upper-bounded by $\|\mathbf{x}\|_\infty$ in abs. Hence,

$$\|\mathbf{y}\|_\infty = \max_{i=1,\dots,n} |y_i| \leq \|\mathbf{x}\|_\infty \qquad \blacksquare$$

Proof 3: Let λ be any eigenvalue of $\mathbf{A}$ and $\mathbf{v}$ be the corresponding eigenvector. Using the fact that $\|\mathbf{A}\mathbf{v}\|_\infty \leq \|\mathbf{v}\|_\infty$, we conclude

$$\|\mathbf{A}\mathbf{v}\|_\infty = \|\lambda \mathbf{v}\|_\infty = |\lambda| \cdot \|\mathbf{v}\|_\infty \leq |\mathbf{v}\|_\infty$$

Eigenvector $\mathbf{v}$ is nonzero $\implies |\lambda| \leq 1$. ■

Convergence of Sequence of Functions

Let $\mathcal{X}$ be any set and $(f_n)_{n \in \mathbb{N}} : \mathcal{X} \to \mathbb{R}$ be a sequence of functions. Then, we say that

$(f_n)_{n \in \mathbb{N}}$ converges to f **point-wise** iff

$$\forall x \in \mathcal{X}, \lim_{n \to \infty} f_n(x) = f(x) \tag{36}$$

$$\Updownarrow$$

$$\forall x \in \mathcal{X}, \forall \epsilon > 0, \exists N \in \mathbb{N}, \text{ s.t. } \forall n \geq N, |f_n(x) - f(x)| < \epsilon \tag{37}$$

$(f_n)_{n \in \mathbb{N}}$ converges to f **uniformly** iff

$$\forall \epsilon > 0, \exists N \in \mathbb{N}, \text{ s.t. } \forall n \geq N, \forall x \in \mathcal{X}, |f_n(x) - f(x)| < \epsilon \tag{38}$$

$(f_n)_{n \in \mathbb{N}}$ converges to f **in sup norm** iff

$$\lim_{n \to \infty} \|f_n(x) - f(x)\|_\infty = 0 \tag{39}$$

$$\Updownarrow$$

$$\forall \epsilon > 0, \exists N \in \mathbb{N}, \text{ s.t. } \forall n \geq N, \sup_{x \in \mathcal{X}} |f_n(x) - f(x)| < \epsilon \tag{40}$$

Relation between different types of convergence:

$$\text{uniform convg.} \iff \text{convg. in sup norm} \implies \text{point-wise convg.}$$

$(f_n)_{n\in\mathbb{N}}$ is **monotonically increasing** iff for each $x \in \mathcal{X}$, the sequence $(f_n(x))_{n\in\mathbb{N}}$ is monotonically increasing, i.e.

$$\forall x \in \mathcal{X}, \forall n \in \mathbb{N}, f_n(x) \leq f_{n+1}(x)$$

$(f_n)_{n\in\mathbb{N}}$ is **point-wise bounded** iff for each $x \in \mathcal{X}$, the sequence $(f_n(x))_{n\in\mathbb{N}}$ is bounded by some $M_x \in \mathbb{R}$, i.e.

$$\forall x \in \mathcal{X}, \exists M_x \in \mathbb{R} \text{ s.t. } \forall n \in \mathbb{N}, |f_n(x)| \leq M_x$$

$(f_n)_{n\in\mathbb{N}}$ is **uniformly bounded** iff

$$\exists M \in \mathbb{R} \text{ s.t. } \forall x \in \mathcal{X}, \forall n \in \mathbb{N}, |f_n(x)| \leq M$$

Monotonic Convergence Theorem

Let $(f_n)_{n\in\mathbb{N}}$ be point-wise bounded and monotonically increasing, then $(f_n)_{n\in\mathbb{N}}$ converges to some f.

Remark: The limit function is not necessarily equal to the pointwise bound! i.e. Let $|f_n(x)|$ be bounded by M_x. Then, $f(x) \neq M_x$ in general.