Chapter 2

State Values and Bellman Equation

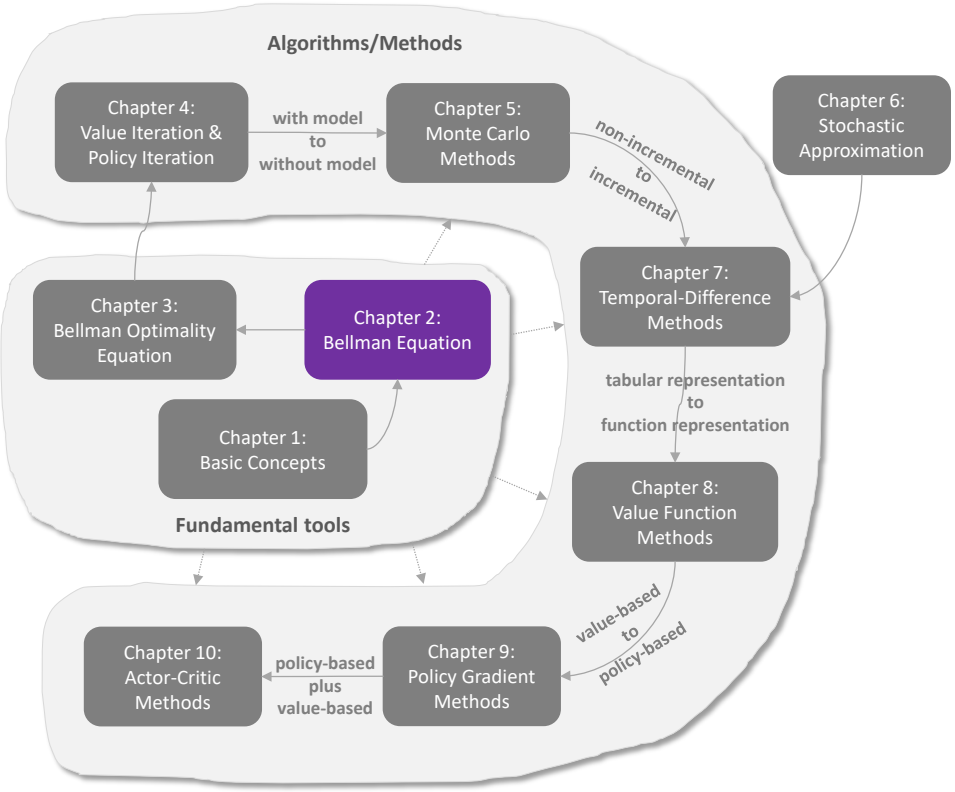


Figure 2.1: Where we are in this book.

This chapter introduces *a core concept* and *an important tool*. The core concept is the *state value*, which is defined as the average reward that an agent can obtain if it follows a given policy. The greater the state value is, the better the corresponding policy is. State values can be used as a metric to evaluate whether a policy is good or not. While state values are important, how can we analyze them? The answer is the *Bellman equation*, which is an important tool for analyzing state values. In a nutshell, the Bellman equation describes the relationships between the values of all states. By solving the Bellman equation, we can obtain the state values. This process is called *policy evaluation*, which is a fundamental concept in reinforcement learning. Finally, this

chapter introduces another important concept called the *action value*.

2.1 Motivating example 1: Why are returns important?

The previous chapter introduced the concept of returns. In fact, returns play a fundamental role in reinforcement learning since they can evaluate whether a policy is good or not. This is demonstrated by the following examples.

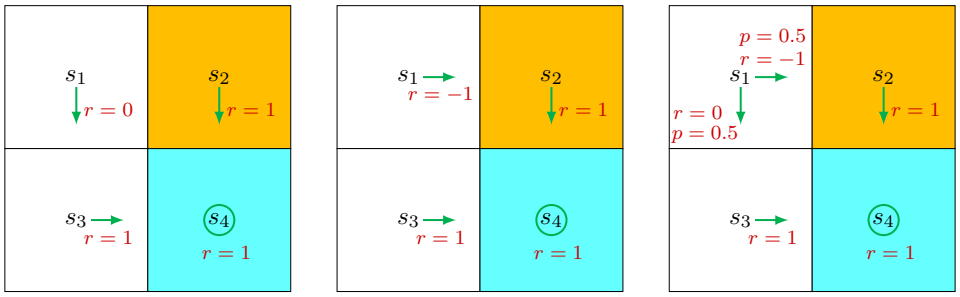


Figure 2.2: Examples for demonstrating the importance of returns. The three examples have different policies for s_1.

Consider the three policies shown in Figure 2.2. It can be seen that the three policies are different at s_1. Which is the best and which is the worst? Intuitively, the leftmost policy is the best because the agent starting from s_1 can avoid the forbidden area. The middle policy is intuitively worse because the agent starting from s_1 moves to the forbidden area. The rightmost policy is in between the others because it has a probability of 0.5 to go to the forbidden area.

While the above analysis is based on intuition, a question that immediately follows is whether we can use mathematics to describe such intuition. The answer is yes and relies on the return concept. In particular, suppose that the agent starts from s_1.

- ◇ Following the first policy, the trajectory is $s_1 \to s_3 \to s_4 \to s_4 \cdots$. The corresponding discounted return is

$$\begin{aligned} \text{return}_1 &= 0 + \gamma 1 + \gamma^2 1 + \dots \\ &= \gamma(1 + \gamma + \gamma^2 + \dots) \\ &= \frac{\gamma}{1-\gamma}, \end{aligned}$$

  where $\gamma \in (0, 1)$ is the discount rate.

- ◇ Following the second policy, the trajectory is $s_1 \to s_2 \to s_4 \to s_4 \cdots$. The discounted

return is

$$\begin{aligned}\text{return}_2 &= -1 + \gamma 1 + \gamma^2 1 + \dots \\ &= -1 + \gamma(1 + \gamma + \gamma^2 + \dots) \\ &= -1 + \frac{\gamma}{1-\gamma}.\end{aligned}$$

◇ Following the third policy, two trajectories can possibly be obtained. One is $s_1 \to s_3 \to s_4 \to s_4 \cdots$, and the other is $s_1 \to s_2 \to s_4 \to s_4 \cdots$. The probability of either of the two trajectories is 0.5. Then, the average return that can be obtained starting from s_1 is

$$\begin{aligned}\text{return}_3 &= 0.5\left(-1 + \frac{\gamma}{1-\gamma}\right) + 0.5\left(\frac{\gamma}{1-\gamma}\right) \\ &= -0.5 + \frac{\gamma}{1-\gamma}.\end{aligned}$$

By comparing the returns of the three policies, we notice that

$$\text{return}_1 > \text{return}_3 > \text{return}_2 \tag{2.1}$$

for any value of γ. Inequality (2.1) suggests that the first policy is the best because its return is the greatest, and the second policy is the worst because its return is the smallest. This mathematical conclusion is consistent with the aforementioned intuition: the first policy is the best since it can avoid entering the forbidden area, and the second policy is the worst because it leads to the forbidden area.

The above examples demonstrate that returns can be used to evaluate policies: a policy is better if the return obtained by following that policy is greater. Finally, it is notable that return_3 does not strictly comply with the definition of returns because it is more like an expected value. It will become clear later that return_3 is actually a state value.

2.2 Motivating example 2: How to calculate returns?

While we have demonstrated the importance of returns, a question that immediately follows is how to calculate the returns when following a given policy.

There are two ways to calculate returns.

◇ The first is simply by definition: a return equals the discounted sum of all the rewards collected along a trajectory. Consider the example in Figure 2.3. Let v_i denote the return obtained by starting from s_i for $i = 1, 2, 3, 4$. Then, the returns obtained when

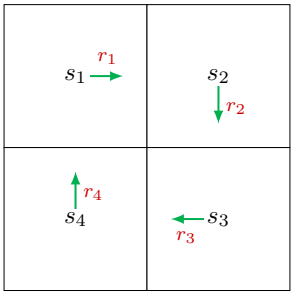


Figure 2.3: An example for demonstrating how to calculate returns. There are no target or forbidden cells in this example.

starting from the four states in Figure 2.3 can be calculated as

$$\begin{aligned} v_1 &= r_1 + \gamma r_2 + \gamma^2 r_3 + \dots, \\ v_2 &= r_2 + \gamma r_3 + \gamma^2 r_4 + \dots, \\ v_3 &= r_3 + \gamma r_4 + \gamma^2 r_1 + \dots, \\ v_4 &= r_4 + \gamma r_1 + \gamma^2 r_2 + \dots. \end{aligned} \tag{2.2}$$

◇ The second way, which is more important, is based on the idea of *bootstrapping*. By observing the expressions of the returns in (2.2), we can rewrite them as

$$\begin{aligned} v_1 &= r_1 + \gamma(r_2 + \gamma r_3 + \dots) = r_1 + \gamma v_2, \\ v_2 &= r_2 + \gamma(r_3 + \gamma r_4 + \dots) = r_2 + \gamma v_3, \\ v_3 &= r_3 + \gamma(r_4 + \gamma r_1 + \dots) = r_3 + \gamma v_4, \\ v_4 &= r_4 + \gamma(r_1 + \gamma r_2 + \dots) = r_4 + \gamma v_1. \end{aligned} \tag{2.3}$$

The above equations indicate an interesting phenomenon that the values of the returns rely on each other. More specifically, v_1 relies on v_2, v_2 relies on v_3, v_3 relies on v_4, and v_4 relies on v_1. This reflects the idea of bootstrapping, which is to obtain the values of some quantities from themselves.

At first glance, bootstrapping is an endless loop because the calculation of an unknown value relies on another unknown value. In fact, bootstrapping is easier to understand if we view it from a mathematical perspective. In particular, the equations in (2.3) can be reformed into a linear matrix-vector equation:

$$\underbrace{\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix}}_{v} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix} + \begin{bmatrix} \gamma v_2 \\ \gamma v_3 \\ \gamma v_4 \\ \gamma v_1 \end{bmatrix} = \underbrace{\begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix}}_{r} + \gamma \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}}_{P} \underbrace{\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix}}_{v},$$

which can be written compactly as

$$v = r + \gamma P v.$$

Thus, the value of v can be calculated easily as $v = (I - \gamma P)^{-1} r$, where I is the identity matrix with appropriate dimensions. One may ask whether $I - \gamma P$ is always invertible. The answer is yes and explained in Section 2.7.1.

In fact, (2.3) is the Bellman equation for this simple example. Although it is simple, (2.3) demonstrates the core idea of the Bellman equation: the return obtained by starting from one state depends on those obtained when starting from other states. The idea of bootstrapping and the Bellman equation for general scenarios will be formalized in the following sections.

2.3 State values

We mentioned that returns can be used to evaluate policies. However, they are inapplicable to stochastic systems because starting from one state may lead to different returns. Motivated by this problem, we introduce the concept of state value in this section.

First, we need to introduce some necessary notations. Consider a sequence of time steps $t = 0, 1, 2, \ldots$. At time t, the agent is in state S_t, and the action taken following a policy π is A_t. The next state is S_{t+1}, and the immediate reward obtained is R_{t+1}. This process can be expressed concisely as

$$S_t \xrightarrow{A_t} S_{t+1}, R_{t+1}.$$

Note that $S_t, S_{t+1}, A_t, R_{t+1}$ are all *random variables*. Moreover, $S_t, S_{t+1} \in \mathcal{S}$, $A_t \in \mathcal{A}(S_t)$, and $R_{t+1} \in \mathcal{R}(S_t, A_t)$.

Starting from t, we can obtain a state-action-reward trajectory:

$$S_t \xrightarrow{A_t} S_{t+1}, R_{t+1} \xrightarrow{A_{t+1}} S_{t+2}, R_{t+2} \xrightarrow{A_{t+2}} S_{t+3}, R_{t+3} \ldots.$$

By definition, the discounted return along the trajectory is

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots,$$

where $\gamma \in (0, 1)$ is the discount rate. Note that G_t is a random variable since $R_{t+1}, R_{t+2}, \ldots$ are all random variables.

Since G_t is a random variable, we can calculate its expected value (also called the expectation or mean):

$$v_\pi(s) \doteq \mathbb{E}[G_t | S_t = s].$$

Here, $v_\pi(s)$ is called the *state-value function* or simply the *state value* of s. Some important remarks are given below.

- ◇ $v_\pi(s)$ depends on s. This is because its definition is a conditional expectation with the condition that the agent starts from $S_t = s$.
- ◇ $v_\pi(s)$ depends on π. This is because the trajectories are generated by following the policy π. For a different policy, the state value may be different.
- ◇ $v_\pi(s)$ does not depend on t. If the agent moves in the state space, t represents the current time step. The value of $v_\pi(s)$ is determined once the policy is given.

The relationship between state values and returns is further clarified as follows. When both the policy and the system model are deterministic, starting from a state always leads to the same trajectory. In this case, the return obtained starting from a state is equal to the value of that state. By contrast, when either the policy or the system model is stochastic, starting from the same state may generate different trajectories. In this case, the returns of different trajectories are different, and the state value is the mean of these returns.

Although returns can be used to evaluate policies as shown in Section 2.1, it is more formal to use state values to evaluate policies: policies that generate greater state values are better. Therefore, state values constitute a core concept in reinforcement learning. While state values are important, a question that immediately follows is how to calculate them. This question is answered in the next section.

2.4 Bellman equation

We now introduce the Bellman equation, a mathematical tool for analyzing state values. In a nutshell, the Bellman equation is a set of linear equations that describe the relationships between the values of all the states.

We next derive the Bellman equation. First, note that G_t can be rewritten as

$$\begin{aligned} G_t &= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \\ &= R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + \dots) \\ &= R_{t+1} + \gamma G_{t+1}, \end{aligned}$$

where $G_{t+1} = R_{t+2} + \gamma R_{t+3} + \dots$. This equation establishes the relationship between G_t and G_{t+1}. Then, the state value can be written as

$$\begin{aligned} v_\pi(s) &= \mathbb{E}[G_t | S_t = s] \\ &= \mathbb{E}[R_{t+1} + \gamma G_{t+1} | S_t = s] \\ &= \mathbb{E}[R_{t+1} | S_t = s] + \gamma \mathbb{E}[G_{t+1} | S_t = s]. \end{aligned} \tag{2.4}$$

The two terms in (2.4) are analyzed below.

◇ The first term, $\mathbb{E}[R_{t+1}|S_t = s]$, is the expectation of the immediate rewards. By using the law of total expectation (Appendix A), it can be calculated as

$$\begin{aligned}\mathbb{E}[R_{t+1}|S_t = s] &= \sum_{a\in\mathcal{A}} \pi(a|s)\mathbb{E}[R_{t+1}|S_t = s, A_t = a] \\ &= \sum_{a\in\mathcal{A}} \pi(a|s) \sum_{r\in\mathcal{R}} p(r|s,a)r. \end{aligned} \tag{2.5}$$

Here, $\mathcal{A}$ and $\mathcal{R}$ are the sets of possible actions and rewards, respectively. It should be noted that $\mathcal{A}$ may be different for different states. In this case, $\mathcal{A}$ should be written as $\mathcal{A}(s)$. Similarly, $\mathcal{R}$ may also depend on (s, a). We drop the dependence on s or (s, a) for the sake of simplicity in this book. Nevertheless, the conclusions are still valid in the presence of dependence.

◇ The second term, $\mathbb{E}[G_{t+1}|S_t = s]$, is the expectation of the future rewards. It can be calculated as

$$\begin{aligned}\mathbb{E}[G_{t+1}|S_t = s] &= \sum_{s'\in\mathcal{S}} \mathbb{E}[G_{t+1}|S_t = s, S_{t+1} = s']p(s'|s) \\ &= \sum_{s'\in\mathcal{S}} \mathbb{E}[G_{t+1}|S_{t+1} = s']p(s'|s) \qquad \text{(due to the Markov property)} \\ &= \sum_{s'\in\mathcal{S}} v_\pi(s')p(s'|s) \\ &= \sum_{s'\in\mathcal{S}} v_\pi(s') \sum_{a\in\mathcal{A}} p(s'|s,a)\pi(a|s). \end{aligned} \tag{2.6}$$

The above derivation uses the fact that $\mathbb{E}[G_{t+1}|S_t = s, S_{t+1} = s'] = \mathbb{E}[G_{t+1}|S_{t+1} = s']$, which is due to the Markov property that the future rewards depend merely on the present state rather than the previous ones.

Substituting (2.5)-(2.6) into (2.4) yields

$$\begin{aligned} v_\pi(s) &= \mathbb{E}[R_{t+1}|S_t = s] + \gamma\mathbb{E}[G_{t+1}|S_t = s], \\ &= \underbrace{\sum_{a\in\mathcal{A}} \pi(a|s) \sum_{r\in\mathcal{R}} p(r|s,a)r}_{\text{mean of immediate rewards}} + \underbrace{\gamma \sum_{a\in\mathcal{A}} \pi(a|s) \sum_{s'\in\mathcal{S}} p(s'|s,a)v_\pi(s')}_{\text{mean of future rewards}} \\ &= \sum_{a\in\mathcal{A}} \pi(a|s) \left[\sum_{r\in\mathcal{R}} p(r|s,a)r + \gamma \sum_{s'\in\mathcal{S}} p(s'|s,a)v_\pi(s')\right], \quad \text{for all } s \in \mathcal{S}. \end{aligned} \tag{2.7}$$

This equation is the *Bellman equation*, which characterizes the relationships of state values. It is a fundamental tool for designing and analyzing reinforcement learning algorithms.

The Bellman equation seems complex at first glance. In fact, it has a clear structure. Some remarks are given below.

- ◇ $v_\pi(s)$ and $v_\pi(s')$ are unknown state values to be calculated. It may be confusing to beginners how to calculate the unknown $v_\pi(s)$ given that it relies on another unknown $v_\pi(s')$. It must be noted that the Bellman equation refers to a set of linear equations for all states rather than a single equation. If we put these equations together, it becomes clear how to calculate all the state values. Details will be given in Section 2.7.
- ◇ $\pi(a|s)$ is a given policy. Since state values can be used to evaluate a policy, solving the state values from the Bellman equation is a *policy evaluation* process, which is an important process in many reinforcement learning algorithms, as we will see later in the book.
- ◇ $p(r|s,a)$ and $p(s'|s,a)$ represent the system model. We will first show how to calculate the state values *with* this model in Section 2.7, and then show how to do that *without* the model by using model-free algorithms later in this book.

In addition to the expression in (2.7), readers may also encounter other expressions of the Bellman equation in the literature. We next introduce two equivalent expressions.

First, it follows from the law of total probability that

$$p(s'|s,a) = \sum_{r\in\mathcal{R}} p(s',r|s,a),$$
$$p(r|s,a) = \sum_{s'\in\mathcal{S}} p(s',r|s,a).$$

Then, equation (2.7) can be rewritten as

$$v_\pi(s) = \sum_{a\in\mathcal{A}} \pi(a|s) \sum_{s'\in\mathcal{S}} \sum_{r\in\mathcal{R}} p(s',r|s,a) \left[r + \gamma v_\pi(s')\right].$$

This is the expression used in [3].

Second, the reward r may depend solely on the next state s' in some problems. As a result, we can write the reward as $r(s')$ and hence $p(r(s')|s,a) = p(s'|s,a)$, substituting which into (2.7) gives

$$v_\pi(s) = \sum_{a\in\mathcal{A}} \pi(a|s) \sum_{s'\in\mathcal{S}} p(s'|s,a) \left[r(s') + \gamma v_\pi(s')\right].$$

2.5 Examples for illustrating the Bellman equation

We next use two examples to demonstrate how to write out the Bellman equation and calculate the state values step by step. Readers are advised to carefully go through the examples to gain a better understanding of the Bellman equation.

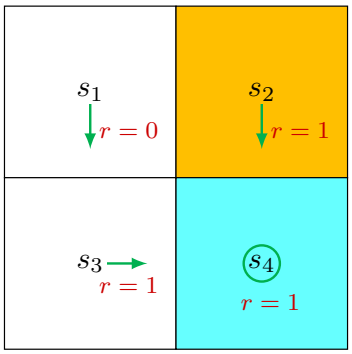


Figure 2.4: An example for demonstrating the Bellman equation. The policy in this example is deterministic.

◇ Consider the first example shown in Figure 2.4, where the policy is deterministic. We next write out the Bellman equation and then solve the state values from it.

First, consider state s_1. Under the policy, the probabilities of taking the actions are $\pi(a = a_3|s_1) = 1$ and $\pi(a \neq a_3|s_1) = 0$. The state transition probabilities are $p(s' = s_3|s_1, a_3) = 1$ and $p(s' \neq s_3|s_1, a_3) = 0$. The reward probabilities are $p(r = 0|s_1, a_3) = 1$ and $p(r \neq 0|s_1, a_3) = 0$. Substituting these values into (2.7) gives

$$v_\pi(s_1) = 0 + \gamma v_\pi(s_3).$$

Interestingly, although the expression of the Bellman equation in (2.7) seems complex, the expression for this specific state is very simple.

Similarly, it can be obtained that

$$\begin{aligned} v_\pi(s_2) &= 1 + \gamma v_\pi(s_4), \\ v_\pi(s_3) &= 1 + \gamma v_\pi(s_4), \\ v_\pi(s_4) &= 1 + \gamma v_\pi(s_4). \end{aligned}$$

We can solve the state values from these equations. Since the equations are simple, we can manually solve them. More complicated equations can be solved by the algorithms presented in Section 2.7. Here, the state values can be solved as

$$\begin{aligned} v_\pi(s_4) &= \frac{1}{1-\gamma}, \\ v_\pi(s_3) &= \frac{1}{1-\gamma}, \\ v_\pi(s_2) &= \frac{1}{1-\gamma}, \\ v_\pi(s_1) &= \frac{\gamma}{1-\gamma}. \end{aligned}$$

Furthermore, if we set $\gamma = 0.9$, then

$$
\begin{aligned}
v_\pi(s_4) &= \frac{1}{1-0.9} = 10, \\
v_\pi(s_3) &= \frac{1}{1-0.9} = 10, \\
v_\pi(s_2) &= \frac{1}{1-0.9} = 10, \\
v_\pi(s_1) &= \frac{0.9}{1-0.9} = 9.
\end{aligned}
$$

infinite time horizon

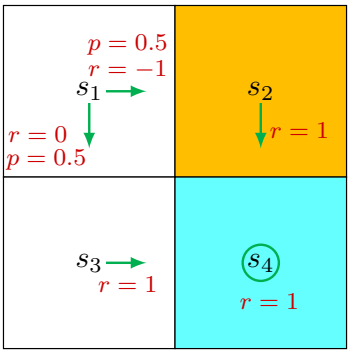


Figure 2.5: An example for demonstrating the Bellman equation. The policy in this example is stochastic.

◇ Consider the second example shown in Figure 2.5, where the policy is stochastic. We next write out the Bellman equation and then solve the state values from it.

In state s_1, the probabilities of going right and down equal 0.5. Mathematically, we have $\pi(a = a_2|s_1) = 0.5$ and $\pi(a = a_3|s_1) = 0.5$. The state transition probability is deterministic since $p(s' = s_3|s_1, a_3) = 1$ and $p(s' = s_2|s_1, a_2) = 1$. The reward probability is also deterministic since $p(r = 0|s_1, a_3) = 1$ and $p(r = -1|s_1, a_2) = 1$. Substituting these values into (2.7) gives

$$
v_\pi(s_1) = 0.5[0 + \gamma v_\pi(s_3)] + 0.5[-1 + \gamma v_\pi(s_2)].
$$

Similarly, it can be obtained that

$$
\begin{aligned}
v_\pi(s_2) &= 1 + \gamma v_\pi(s_4), \\
v_\pi(s_3) &= 1 + \gamma v_\pi(s_4), \\
v_\pi(s_4) &= 1 + \gamma v_\pi(s_4).
\end{aligned}
$$

The state values can be solved from the above equations. Since the equations are

simple, we can solve the state values manually and obtain

$$\begin{aligned}
v_\pi(s_4) &= \frac{1}{1-\gamma}, \\
v_\pi(s_3) &= \frac{1}{1-\gamma}, \\
v_\pi(s_2) &= \frac{1}{1-\gamma}, \\
v_\pi(s_1) &= 0.5[0 + \gamma v_\pi(s_3)] + 0.5[-1 + \gamma v_\pi(s_2)], \\
&= -0.5 + \frac{\gamma}{1-\gamma}.
\end{aligned}$$

Furthermore, if we set $\gamma = 0.9$, then

$$\begin{aligned}
v_\pi(s_4) &= 10, \\
v_\pi(s_3) &= 10, \\
v_\pi(s_2) &= 10, \\
v_\pi(s_1) &= -0.5 + 9 = 8.5.
\end{aligned}$$

If we compare the state values of the two policies in the above examples, it can be seen that

$$v_{\pi_1}(s_i) \geq v_{\pi_2}(s_i), \quad i = 1, 2, 3, 4,$$

which indicates that the policy in Figure 2.4 is better because it has greater state values. This mathematical conclusion is consistent with the intuition that the first policy is better because it can avoid entering the forbidden area when the agent starts from s_1. As a result, the above two examples demonstrate that state values can be used to evaluate policies.

2.6 Matrix-vector form of the Bellman equation

The Bellman equation in (2.7) is in an *elementwise form*. Since it is valid for every state, we can combine all these equations and write them concisely in a *matrix-vector form*, which will be frequently used to analyze the Bellman equation.

To derive the matrix-vector form, we first rewrite the Bellman equation in (2.7) as

$$v_\pi(s) = r_\pi(s) + \gamma \sum_{s' \in \mathcal{S}} p_\pi(s'|s) v_\pi(s'), \tag{2.8}$$

where

$$r_\pi(s) \doteq \sum_{a\in\mathcal{A}} \pi(a|s) \sum_{r\in\mathcal{R}} p(r|s,a)r,$$
$$p_\pi(s'|s) \doteq \sum_{a\in\mathcal{A}} \pi(a|s) p(s'|s,a).$$

Here, $r_\pi(s)$ denotes the mean of the immediate rewards, and $p_\pi(s'|s)$ is the probability of transitioning from s to s' under policy π.

Suppose that the states are indexed as s_i with $i = 1, \ldots, n$, where $n = |\mathcal{S}|$. For state s_i, (2.8) can be written as

$$v_\pi(s_i) = r_\pi(s_i) + \gamma \sum_{s_j\in\mathcal{S}} p_\pi(s_j|s_i) v_\pi(s_j). \tag{2.9}$$

Let $v_\pi = [v_\pi(s_1), \ldots, v_\pi(s_n)]^T \in \mathbb{R}^n$, $r_\pi = [r_\pi(s_1), \ldots, r_\pi(s_n)]^T \in \mathbb{R}^n$, and $P_\pi \in \mathbb{R}^{n\times n}$ with $[P_\pi]_{ij} = p_\pi(s_j|s_i)$. Then, (2.9) can be written in the following matrix-vector form:

$$v_\pi = r_\pi + \gamma P_\pi v_\pi, \tag{2.10}$$

where v_π is the unknown to be solved, and r_π, P_π are known.

The matrix P_π has some interesting properties. First, it is a nonnegative matrix, meaning that all its elements are equal to or greater than zero. This property is denoted as $P_\pi \geq 0$, where 0 denotes a zero matrix with appropriate dimensions. In this book, $\geq$ or $\leq$ represents an elementwise comparison operation. Second, P_π is a stochastic matrix, meaning that the sum of the values in every row is equal to one. This property is denoted as $P_\pi \mathbf{1} = \mathbf{1}$, where $\mathbf{1} = [1, \ldots, 1]^T$ has appropriate dimensions.

Consider the example shown in Figure 2.6. The matrix-vector form of the Bellman equation is

$$\underbrace{\begin{bmatrix} v_\pi(s_1) \\ v_\pi(s_2) \\ v_\pi(s_3) \\ v_\pi(s_4) \end{bmatrix}}_{v_\pi} = \underbrace{\begin{bmatrix} r_\pi(s_1) \\ r_\pi(s_2) \\ r_\pi(s_3) \\ r_\pi(s_4) \end{bmatrix}}_{r_\pi} + \gamma \underbrace{\begin{bmatrix} p_\pi(s_1|s_1) & p_\pi(s_2|s_1) & p_\pi(s_3|s_1) & p_\pi(s_4|s_1) \\ p_\pi(s_1|s_2) & p_\pi(s_2|s_2) & p_\pi(s_3|s_2) & p_\pi(s_4|s_2) \\ p_\pi(s_1|s_3) & p_\pi(s_2|s_3) & p_\pi(s_3|s_3) & p_\pi(s_4|s_3) \\ p_\pi(s_1|s_4) & p_\pi(s_2|s_4) & p_\pi(s_3|s_4) & p_\pi(s_4|s_4) \end{bmatrix}}_{P_\pi} \underbrace{\begin{bmatrix} v_\pi(s_1) \\ v_\pi(s_2) \\ v_\pi(s_3) \\ v_\pi(s_4) \end{bmatrix}}_{v_\pi}.$$

Substituting the specific values into the above equation gives

$$\begin{bmatrix} v_\pi(s_1) \\ v_\pi(s_2) \\ v_\pi(s_3) \\ v_\pi(s_4) \end{bmatrix} = \begin{bmatrix} 0.5(0) + 0.5(-1) \\ 1 \\ 1 \\ 1 \end{bmatrix} + \gamma \begin{bmatrix} 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_\pi(s_1) \\ v_\pi(s_2) \\ v_\pi(s_3) \\ v_\pi(s_4) \end{bmatrix}.$$

It can be seen that P_π satisfies $P_\pi \mathbf{1} = \mathbf{1}$.

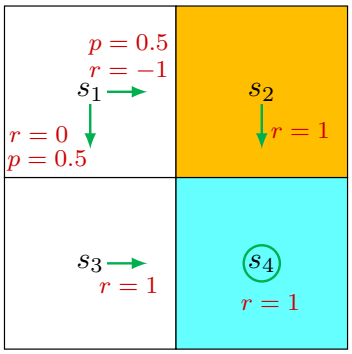


Figure 2.6: An example for demonstrating the matrix-vector form of the Bellman equation.

2.7 Solving state values from the Bellman equation

Calculating the state values of a given policy is a fundamental problem in reinforcement learning. This problem is often referred to as *policy evaluation*. In this section, we present two methods for calculating state values from the Bellman equation.

2.7.1 Closed-form solution

Since $v_\pi = r_\pi + \gamma P_\pi v_\pi$ is a simple linear equation, its *closed-form solution* can be easily obtained as

$$v_\pi = (I - \gamma P_\pi)^{-1} r_\pi.$$

Some properties of $(I - \gamma P_\pi)^{-1}$ are given below.

- $I - \gamma P_\pi$ is invertible. The proof is as follows. According to the Gershgorin circle theorem [4], every eigenvalue of $I - \gamma P_\pi$ lies within at least one of the Gershgorin circles. The ith Gershgorin circle has a center at $[I - \gamma P_\pi]_{ii} = 1 - \gamma p_\pi(s_i|s_i)$ and a radius equal to $\sum_{j \neq i} [I - \gamma P_\pi]_{ij} = -\sum_{j \neq i} \gamma p_\pi(s_j|s_i)$. Since $\gamma < 1$, we know that the radius is less than the magnitude of the center: $\sum_{j \neq i} \gamma p_\pi(s_j|s_i) < 1 - \gamma p_\pi(s_i|s_i)$. Therefore, all Gershgorin circles do not encircle the origin, and hence no eigenvalue of $I - \gamma P_\pi$ is zero.
- $(I - \gamma P_\pi)^{-1} \geq I$, meaning that every element of $(I - \gamma P_\pi)^{-1}$ is nonnegative and, more specifically, no less than that of the identity matrix. This is because P_π has nonnegative entries, and hence $(I - \gamma P_\pi)^{-1} = I + \gamma P_\pi + \gamma^2 P_\pi^2 + \cdots \geq I \geq 0$.
- For any vector $r \geq 0$, it holds that $(I - \gamma P_\pi)^{-1} r \geq r \geq 0$. This property follows from the second property because $[(I - \gamma P_\pi)^{-1} - I] r \geq 0$. As a consequence, if $r_1 \geq r_2$, we have $(I - \gamma P_\pi)^{-1} r_1 \geq (I - \gamma P_\pi)^{-1} r_2$.

2.7.2 Iterative solution

Although the closed-form solution is useful for theoretical analysis purposes, it is not applicable in practice because it involves a matrix inversion operation, which still needs to be calculated by other numerical algorithms. In fact, we can directly solve the Bellman equation using the following iterative algorithm:

$$v_{k+1} = r_\pi + \gamma P_\pi v_k, \quad k = 0, 1, 2, \ldots \tag{2.11}$$

This algorithm generates a sequence of values $\{v_0, v_1, v_2, \ldots\}$, where $v_0 \in \mathbb{R}^n$ is an initial guess of v_π. It holds that

$$v_k \to v_\pi = (I - \gamma P_\pi)^{-1} r_\pi, \quad \text{as } k \to \infty. \tag{2.12}$$

Interested readers may see the proof in Box 2.1.

Box 2.1: Convergence proof of (2.12)

Define the error as $\delta_k = v_k - v_\pi$. We only need to show that $\delta_k \to 0$. Substituting $v_{k+1} = \delta_{k+1} + v_\pi$ and $v_k = \delta_k + v_\pi$ into $v_{k+1} = r_\pi + \gamma P_\pi v_k$ gives

$$\delta_{k+1} + v_\pi = r_\pi + \gamma P_\pi(\delta_k + v_\pi),$$

which can be rewritten as

$$\begin{aligned} \delta_{k+1} &= -v_\pi + r_\pi + \gamma P_\pi \delta_k + \gamma P_\pi v_\pi, \\ &= \gamma P_\pi \delta_k - v_\pi + (r_\pi + \gamma P_\pi v_\pi), \\ &= \gamma P_\pi \delta_k. \end{aligned}$$

As a result,

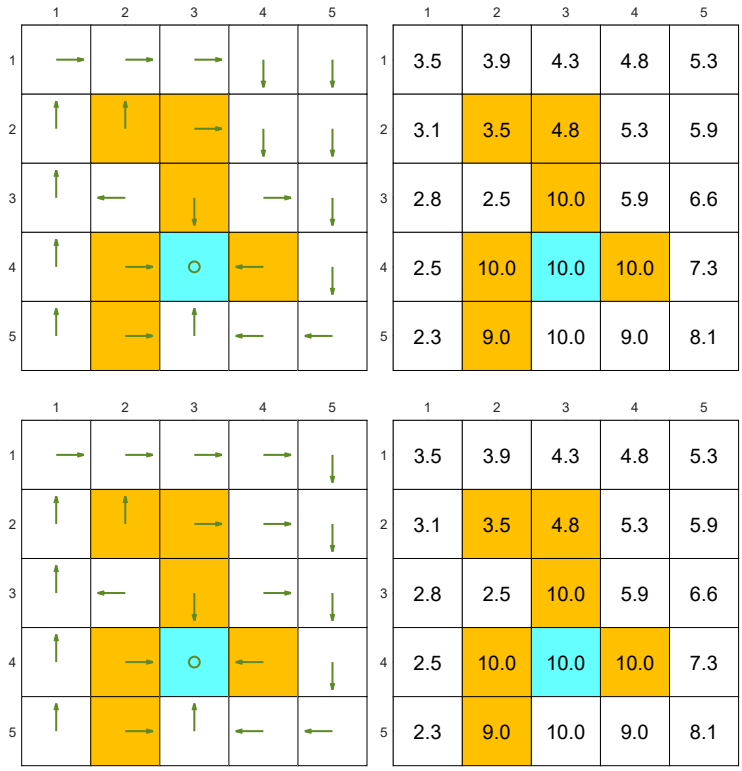
$$\delta_{k+1} = \gamma P_\pi \delta_k = \gamma^2 P_\pi^2 \delta_{k-1} = \cdots = \gamma^{k+1} P_\pi^{k+1} \delta_0.$$

Since every entry of P_π is nonnegative and no greater than one, we have that $0 \leq P_\pi^k \leq 1$ for any k. That is, every entry of P_π^k is no greater than 1. On the other hand, since $\gamma < 1$, we know that $\gamma^k \to 0$, and hence $\delta_{k+1} = \gamma^{k+1} P_\pi^{k+1} \delta_0 \to 0$ as $k \to \infty$.

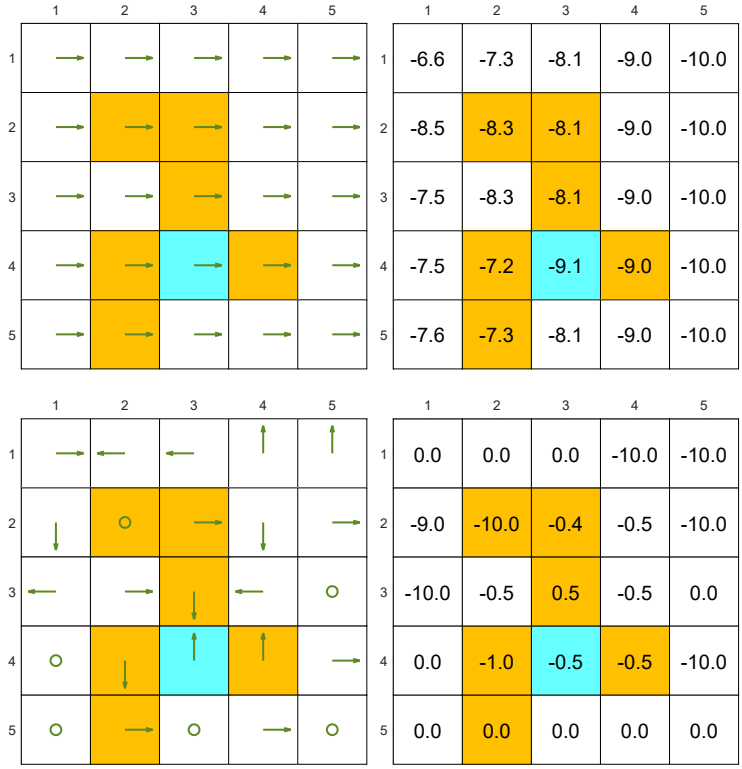
2.7.3 Illustrative examples

We next apply the algorithm in (2.11) to solve the state values of some examples.

The examples are shown in Figure 2.7. The orange cells represent forbidden areas. The blue cell represents the target area. The reward settings are $r_{\text{boundary}} = r_{\text{forbidden}} = -1$



(a) Two "good" policies and their state values. The state values of the two policies are the same, but the two policies are different at the top two states in the fourth column.



(b) Two "bad" policies and their state values. The state values are smaller than those of the "good" policies.

Figure 2.7: Examples of policies and their corresponding state values.

and $r_{\text{target}} = 1$. Here, the discount rate is $\gamma = 0.9$.

Figure 2.7(a) shows two "good" policies and their corresponding state values obtained by (2.11). The two policies have the same state values but differ at the top two states in the fourth column. Therefore, we know that different policies may have the same state values.

Figure 2.7(b) shows two "bad" policies and their corresponding state values. These two policies are bad because the actions of many states are intuitively unreasonable. Such intuition is supported by the obtained state values. As can be seen, the state values of these two policies are negative and much smaller than those of the good policies in Figure 2.7(a).

2.8 From state value to action value

While we have been discussing state values thus far in this chapter, we now turn to the *action value*, which indicates the "value" of taking an action at a state. While the concept of action value is important, the reason why it is introduced in the last section of this chapter is that it heavily relies on the concept of state values. It is important to understand state values well first before studying action values.

The action value of a state-action pair (s, a) is defined as

$$q_\pi(s, a) \doteq \mathbb{E}[G_t|S_t = s, A_t = a].$$

As can be seen, the action value is defined as the expected return that can be obtained after taking an action at a state. It must be noted that $q_\pi(s, a)$ depends on a state-action pair (s, a) rather than an action alone. It may be more rigorous to call this value a state-action value, but it is conventionally called an action value for simplicity.

What is the relationship between action values and state values?

◇ First, it follows from the properties of conditional expectation that

$$\underbrace{\mathbb{E}[G_t|S_t = s]}_{v_\pi(s)} = \sum_{a\in\mathcal{A}} \underbrace{\mathbb{E}[G_t|S_t = s, A_t = a]}_{q_\pi(s,a)} \pi(a|s).$$

It then follows that

$$v_\pi(s) = \sum_{a\in\mathcal{A}} \pi(a|s) q_\pi(s, a). \tag{2.13}$$

As a result, a state value is the expectation of the action values associated with that state.

◇ Second, since the state value is given by

$$v_\pi(s) = \sum_{a\in\mathcal{A}} \pi(a|s) \Big[\sum_{r\in\mathcal{R}} p(r|s, a)r + \gamma \sum_{s'\in\mathcal{S}} p(s'|s, a)v_\pi(s') \Big],$$

comparing it with (2.13) leads to

$$q_\pi(s,a) = \sum_{r\in\mathcal{R}} p(r|s,a)r + \gamma \sum_{s'\in\mathcal{S}} p(s'|s,a)v_\pi(s'). \tag{2.14}$$

It can be seen that the action value consists of two terms. The first term is the mean of the immediate rewards, and the second term is the mean of the future rewards.

Both (2.13) and (2.14) describe the relationship between state values and action values. They are the two sides of the same coin: (2.13) shows how to obtain state values from action values, whereas (2.14) shows how to obtain action values from state values.

2.8.1 Illustrative examples

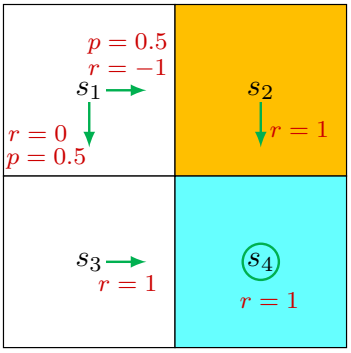


Figure 2.8: An example for demonstrating the process of calculating action values.

We next present an example to illustrate the process of calculating action values and discuss a common mistake that beginners may make.

Consider the stochastic policy shown in Figure 2.8. We next only examine the actions of s_1. The other states can be examined similarly. The action value of (s_1, a_2) is

$$q_\pi(s_1, a_2) = -1 + \gamma v_\pi(s_2),$$

where s_2 is the next state. Similarly, it can be obtained that

$$q_\pi(s_1, a_3) = 0 + \gamma v_\pi(s_3).$$

A common mistake that beginners may make is about the values of the actions that the given policy does not select. For example, the policy in Figure 2.8 can only select a_2 or a_3 and cannot select a_1, a_4, a_5. One may argue that since the policy does not select a_1, a_4, a_5, we do not need to calculate their action values, or we can simply set $q_\pi(s_1, a_1) = q_\pi(s_1, a_4) = q_\pi(s_1, a_5) = 0$. This is wrong.

◇ First, even if an action would not be selected by a policy, it still has an action value. In this example, although policy π does not take a_1 at s_1, we can still calculate its

action value by observing what we would obtain after taking this action. Specifically, after taking a_1, the agent is bounced back to s_1 (hence, the immediate reward is -1) and then continues moving in the state space starting from s_1 by following π (hence, the future reward is $\gamma v_\pi(s_1)$). As a result, the action value of (s_1, a_1) is

$$q_\pi(s_1, a_1) = -1 + \gamma v_\pi(s_1).$$

Similarly, for a_4 and a_5, which cannot be possibly selected by the given policy either, we have

$$\begin{aligned} q_\pi(s_1, a_4) &= -1 + \gamma v_\pi(s_1), \\ q_\pi(s_1, a_5) &= 0 + \gamma v_\pi(s_1). \end{aligned}$$

◇ Second, why do we care about the actions that the given policy would not select? Although some actions cannot be possibly selected by a given policy, this does not mean that these actions are not good. It is possible that the given policy is not good, so it cannot select the best action. The purpose of reinforcement learning is to find optimal policies. To that end, we must keep exploring all actions to determine better actions for each state.

Finally, after computing the action values, we can also calculate the state value according to (2.14):

$$\begin{aligned} v_\pi(s_1) &= 0.5 q_\pi(s_1, a_2) + 0.5 q_\pi(s_1, a_3), \\ &= 0.5[0 + \gamma v_\pi(s_3)] + 0.5[-1 + \gamma v_\pi(s_2)]. \end{aligned}$$

2.8.2 The Bellman equation in terms of action values

The Bellman equation that we previously introduced was defined based on state values. In fact, it can also be expressed in terms of action values.

In particular, substituting (2.13) into (2.14) yields

$$q_\pi(s, a) = \sum_{r \in \mathcal{R}} p(r|s, a) r + \gamma \sum_{s' \in \mathcal{S}} p(s'|s, a) \sum_{a' \in \mathcal{A}(s')} \pi(a'|s') q_\pi(s', a'),$$

which is an equation of action values. The above equation is valid for every state-action pair. If we put all these equations together, their matrix-vector form is

$$q_\pi = \tilde{r} + \gamma P \Pi q_\pi, \tag{2.15}$$

where q_π is the action value vector indexed by the state-action pairs: its (s, a)th element is $[q_\pi]_{(s,a)} = q_\pi(s, a)$. $\tilde{r}$ is the immediate reward vector indexed by the state-action pairs: $[\tilde{r}]_{(s,a)} = \sum_{r \in \mathcal{R}} p(r|s, a) r$. The matrix P is the probability transition matrix, whose

row is indexed by the state-action pairs and whose column is indexed by the states: $[P]_{(s,a),s'} = p(s'|s,a)$. Moreover, Π is a block diagonal matrix in which each block is a $1 \times |\mathcal{A}|$ vector: $\Pi_{s',(s',a')} = \pi(a'|s')$ and the other entries of Π are zero.

Compared to the Bellman equation defined in terms of state values, the equation defined in terms of action values has some unique features. For example, $\tilde{r}$ and P are independent of the policy and are merely determined by the system model. The policy is embedded in Π. It can be verified that (2.15) is also a contraction mapping and has a unique solution that can be iteratively solved. More details can be found in [5].

2.9 Summary

The most important concept introduced in this chapter is the state value. Mathematically, a state value is the expected return that the agent can obtain by starting from a state. The values of different states are related to each other. That is, the value of state s relies on the values of some other states, which may further rely on the value of state s itself. This phenomenon might be the most confusing part of this chapter for beginners. It is related to an important concept called bootstrapping, which involves calculating something from itself. Although bootstrapping may be intuitively confusing, it is clear if we examine the matrix-vector form of the Bellman equation. In particular, the Bellman equation is a set of linear equations that describe the relationships between the values of all states.

Since state values can be used to evaluate whether a policy is good or not, the process of solving the state values of a policy from the Bellman equation is called policy evaluation. As we will see later in this book, policy evaluation is an important step in many reinforcement learning algorithms.

Another important concept, action value, was introduced to describe the value of taking one action at a state. As we will see later in this book, action values play a more direct role than state values when we attempt to find optimal policies. Finally, the Bellman equation is not restricted to the reinforcement learning field. Instead, it widely exists in many fields such as control theories and operation research. In different fields, the Bellman equation may have different expressions. In this book, the Bellman equation is studied under discrete Markov decision processes. More information about this topic can be found in [2].

2.10 Q&A

◇ Q: What is the relationship between state values and returns?

A: The value of a state is the mean of the returns that can be obtained if the agent starts from that state.

- ◇ Q: Why do we care about state values?

  A: State values can be used to evaluate policies. In fact, optimal policies are defined based on state values. This point will become clearer in the next chapter.

- ◇ Q: Why do we care about the Bellman equation?

  A: The Bellman equation describes the relationships among the values of all states. It is the tool for analyzing state values.

- ◇ Q: Why is the process of solving the Bellman equation called policy evaluation?

  A: Solving the Bellman equation yields state values. Since state values can be used to evaluate a policy, solving the Bellman equation can be interpreted as evaluating the corresponding policy.

- ◇ Q: Why do we need to study the matrix-vector form of the Bellman equation?

  A: The Bellman equation refers to a set of linear equations established for all the states. To solve state values, we must put all the linear equations together. The matrix-vector form is a concise expression of these linear equations.

- ◇ Q: What is the relationship between state values and action values?

  A: On the one hand, a state value is the mean of the action values for that state. On the other hand, an action value relies on the values of the next states that the agent may transition to after taking the action.

- ◇ Q: Why do we care about the values of the actions that a given policy cannot select?

  A: Although a given policy cannot select some actions, this does not mean that these actions are not good. On the contrary, it is possible that the given policy is not good and misses the best action. To find better policies, we must keep exploring different actions even though some of them may not be selected by the given policy.