Chapter 2

State Value and Bellman Equation

This chapter will mainly introduce a core concept and an important tool to analyze the concept. The core concept is state value as well as action value. The important tool is the Bellman equation, which characterizes the relationship among the values of different states and can be used to calculate state values. The contents of this chapter are very fundamental in reinforcement learning (RL).

2.1 Motivating examples: Why return is important?

In the last chapter, we introduced the concept of return. In fact, return plays a fundamental role in RL. We next use examples to demonstrate.

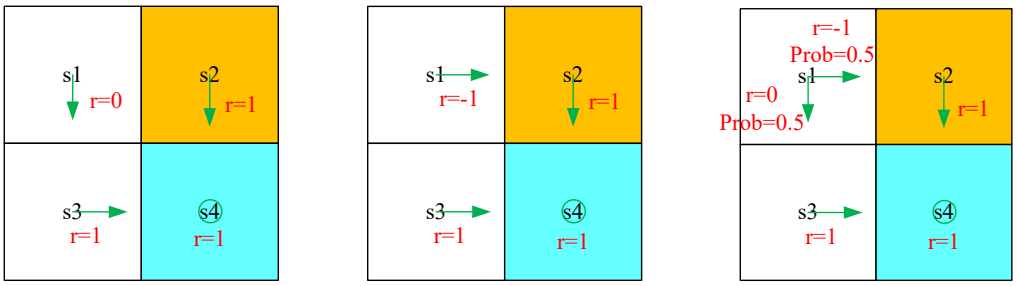


Figure 2.1: Examples to demonstrate the importance of return. The three examples have the same problem setup but different policies.

Consider the three policies as shown in Figure 2.1. The problem setup of the three examples is exactly the same, but the policies are different. Intuitively, the leftmost policy is the best because the agent starting from s_1 can avoid the forbidden area to reach the target. The middle policy is intuitively worse because the agent starting from s_1 moves to the forbidden area. The rightmost policy is in between because it has a probability of 0.5 to go rightwards to the forbidden area.

While the above analysis is based on intuition, a question that immediately follows is: can we use mathematics to describe such intuition? The answer relies on the notion of return.

Suppose the starting state is s_1. Following the first policy, the resulting trajectory is $s_1 \to s_3 \to s_4 \to s_4$. 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 $s_1 \to s_2 \to s_4 \to s_4$. 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, there are two possible trajectories. One is $s_1 \to s_3 \to s_4 \to s_4$ and the other is the trajectory is $s_1 \to s_2 \to s_4 \to s_4$. The probability of taking either of them is 0.5. Then, the average of the discounted returns 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 can easily see that

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

for any value of γ. The above inequality suggests that the first policy is the best and the second policy is the worst, which is exactly the same as our intuition.

The above example demonstrates that return can be used to evaluate different policies: a policy is better if the return obtained following that policy is greater. Of course, this is a naive idea. Such an idea will be formalized when we define state value functions. Finally, it is notable that return_3 does not comply with the definition of return strictly because it is more like an expected value. It will become clear that return_3 is actually a state value.

2.2 Motivating example: How to calculate return?

While we have demonstrated the importance of return, a question that immediately follows is how to calculate the return of a policy starting from different states. There are

two ways to do that.

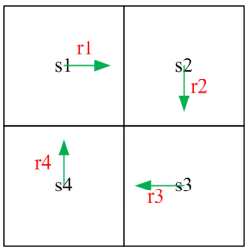


Figure 2.2: An example to demonstrate how to calculate return.

The first is simply by definition: return is defined as the discounted summation of rewards along a trajectory. Consider the example in Figure 2.2. Let v_i denote the return obtained starting from s_i for $i = 1, \dots, 4$. Then, the returns starting from the four states in Figure 2.2 can be respectively calculated as

$$\begin{aligned} v_1 &= r_1 + \gamma r_2 + \gamma^2 r_3 + \dots, \\ v_2 &= r_2 + \gamma r_3 + \gamma^2 r_4 + \dots, \\ v_3 &= r_3 + \gamma r_4 + \gamma^2 r_1 + \dots, \\ v_4 &= r_4 + \gamma r_1 + \gamma^2 r_2 + \dots. \end{aligned} \tag{2.1}$$

The second way is by the idea of *bootstrapping*. By observing the expressions of the returns in (2.1), we can rewrite it to

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

The above equations indicate that the calculation of v_1 relies on the value of v_2. Similarly, v_2 relies on v_3, v_3 relies on v_4, and finally v_4 relies on v_1. This reflects the idea of bootstrapping, which is to obtain something from itself.

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 (2.2) can be reformed to a linear matrix-vector equation:

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

which can be written in short as

$$\mathbf{v} = \mathbf{r} + \gamma \mathbf{P}\mathbf{v}.$$

Thus, the value of $\mathbf{v}$ can be calculated easily as $\mathbf{v} = (\mathbf{I} - \gamma \mathbf{P})^{-1}\mathbf{r}$.

In fact, (2.2) is the Bellman equation for this simple example. Though simple, (2.2) demonstrates the core idea of the Bellman equation: that is, the return obtained starting from one state depends on those 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 value

Although return can be used to evaluate if a policy is good or not, it does not apply to stochastic systems because starting from a state may lead to different trajectories and hence different returns. Motivated by this problem, we define the *mean* of all possible returns starting from a state as the *state value*.

The mathematical definition of state value is derived as follows. First of all, we need to introduce some necessary notations. Consider a sequence of time steps $t = 0, 1, 2, \ldots$. At time t, the agent is at 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*. In particular, $S_t, S_{t+1} \in \mathcal{S}$, $A_t \in \mathcal{A}(S_t)$, and $R_{t+1} \in \mathcal{R}(S_t, A_t)$. It is worth mentioning that the reward obtained after the agent takes action A_t can be also denoted R_t instead of R_{t+1}. Mathematically, it does not make any difference.

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

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

where $\gamma \in (0, 1)$ is a 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 expectation (or called expected value 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 *state value* of s. Some important remarks are given below.

– $v_\pi(s)$ depends on s because its definition is a conditional expectation with the condition that the state starts from $S_t = s$. As its name suggests, it represents the "value" of a state, which is the expected value of the rewards that can be obtained starting from s.

– $v_\pi(s)$ depends on π because the trajectories are generated by following the policy π. For a different policy, the state value may be different.

The relationship between state value and return is further clarified as follows. When everything (system model and policy) is deterministic, the value of a state is equal to the return obtained starting from that state. In the presence of randomness, the returns of different trajectories would be different. In this case, the state value is the mean of these returns. While Section 2.1 demonstrates that return can be used to evaluate policies, a more general way is to use state value to evaluate policies: policies generating greater state values are better. More details about optimal policies will be given in the next chapter.

2.4 Bellman equation

We now introduce the Bellman equation, a mathematical tool to analyze state values. In one word, the Bellman equation is a set of linear equations describing the relationship among the values of all the states.

We next derive the Bellman equation. First of all, 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

$$\begin{aligned} v_\pi(s) &= \mathbb{E}[G_t|S_t = s] \\ &= \mathbb{E}[R_{t+1} + \gamma G_{t+1}|S_t = s] \\ &= \mathbb{E}[R_{t+1}|S_t = s] + \gamma \mathbb{E}[G_{t+1}|S_t = s]. \end{aligned} \tag{2.3}$$

The first term, $\mathbb{E}[R_{t+1}|S_t = s]$, in (2.3) is the expectation of the immediate reward that can be obtained starting from s. By using the law of total expectation, it can be calculated as

$$\begin{aligned} \mathbb{E}[R_{t+1}|S_t = s] &= \sum_a \pi(a|s)\mathbb{E}[R_{t+1}|S_t = s, A_t = a] \\ &= \sum_a \pi(a|s) \sum_r p(r|s,a)r. \end{aligned} \tag{2.4}$$

The second term, $\mathbb{E}[G_{t+1}|S_t = s]$, in (2.3) is the expectation of the future reward. It can be calculated as

$$\begin{aligned}
\mathbb{E}[G_{t+1}|S_t = s] &= \sum_{s'} \mathbb{E}[G_{t+1}|S_t = s, S_{t+1} = s']p(s'|s) \\
&= \sum_{s'} \mathbb{E}[G_{t+1}|S_{t+1} = s']p(s'|s) \qquad \text{(due to conditional independence)} \\
&= \sum_{s'} v_\pi(s')p(s'|s) \\
&= \sum_{s'} v_\pi(s') \sum_a p(s'|s,a)\pi(a|s). \qquad (2.5)
\end{aligned}$$

The above derivation uses the fact that $\mathbb{E}[G_{t+1}|S_t = s, S_{t+1} = s'] = \mathbb{E}[G_{t+1}|S_{t+1} = s']$. This is due to the conditional independence property thanks to the memoryless Markov property that the future behavior totally depends on the present state. Substituting (2.4)-(2.5) into (2.3) yields

$$\begin{aligned}
v_\pi(s) &= \mathbb{E}[R_{t+1}|S_t = s] + \gamma\mathbb{E}[G_{t+1}|S_t = s], \\
&= \underbrace{\sum_a \pi(a|s) \sum_r p(r|s,a)r}_{\text{mean of immediate rewards}} + \underbrace{\gamma \sum_a \pi(a|s) \sum_{s'} p(s'|s,a)v_\pi(s')}_{\text{mean of future rewards}}, \\
&= \sum_a \pi(a|s) \left[\sum_r p(r|s,a)r + \gamma \sum_{s'} p(s'|s,a)v_\pi(s') \right], \quad \text{for all } s \in \mathcal{S}. \qquad (2.6)
\end{aligned}$$

This equation is called the *Bellman equation*, which characterizes the relationship of state values. It is a fundamental tool for designing and analyzing RL algorithms.

At first glance, the Bellman equation is quite complex. In fact, it has a clear structure.

- $v_\pi(s)$ and $v_\pi(s')$ are 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 is a set of linear equations rather than a single equation. If we put these equations together, it would be clear how to calculate all the state values. Details will be given in Section 2.5.
- $\pi(a|s)$ is a given policy. Since state values can be used to evaluate a policy, calculating the state values from the Bellman equation is called *policy evaluation*, which is an important step for many RL algorithms as we will see later in the book.
- $p(r|s,a)$ and $p(s'|s,a)$ represent the dynamic model. We will first show how to calculate the state values given the model and later show how to do that without the model.

2.4.1 Illustrative examples

We next use examples to demonstrate how to manually write out the Bellman equation and calculate the state values step by step.

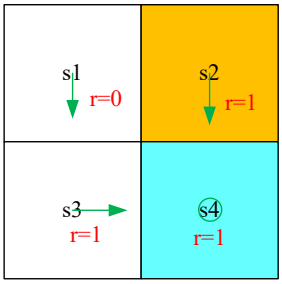


Figure 2.3: An example to demonstrate the Bellman equation. The policy in this example is deterministic.

Consider the first example shown in Figure 2.3 where the policy is deterministic. First, consider state s_1. Under the policy, the probability to take actions is $\pi(a = a_3|s_1) = 1$ and $\pi(a \neq a_3|s_1) = 0$. The state transition probability is $p(s' = s_3|s_1, a_3) = 1$ and $p(s' \neq s_3|s_1, a_3) = 0$. The reward probability is $p(r = -1|s_1, a_3) = 1$ and $p(r \neq -1|s_1, a_3) = 0$. Substituting them into the Bellman equation gives

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

Similarly, it can be obtained that

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

Solving the above equations one by one from the last to the first gives

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

If $\gamma = 0.9$, then

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

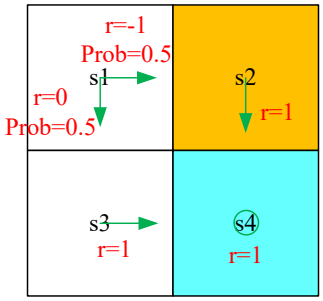


Figure 2.4: An example to demonstrate the Bellman equation. The policy in this example is stochastic.

Consider the second example shown in Figure 2.4 where the policy is stochastic. We next write out the Bellman equation and then compare this policy with the one in Figure 2.3.

At state s_1, the probabilities to go right and down are equal to 0.5. Mathematically, the probability to take actions is $\pi(a = a_2|s_1) = 0.5$ 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 $p(r = -1|s_1, a_2) = 1$. Therefore, we have

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

Similarly, it can be obtained that

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

Solving the above equations one by one from the last to the first gives

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

Substituting $\gamma = 0.9$ yields

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

If we compare the two policies in the above two examples, the first policy is better since $v_{\pi_1}(s_1) = 9 > v_{\pi_2}(s_1) = 8.5$. This is also consistent with the intuition that the policy in the second example is not good since the agent may move to the forbidden cell from s_1.

2.4.2 Alternative expressions of the Bellman equation

In addition to the expression in (2.6), the reader may also encounter other expressions of the Bellman equation in the literature. We next give another two alternative expressions.

First, it follows from the law of total probability that

$$
\begin{aligned}
p(s'|s,a) &= \sum_r p(s', r|s, a), \\
p(r|s,a) &= \sum_{s'} p(s', r|s, a).
\end{aligned}
$$

Then, equation (2.6) can be rewritten as

$$
v(s) = \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a) \left[r + \gamma v(s') \right], \tag{2.7}
$$

where $\sum_{s',r} = \sum_{s'} \sum_r$. This is the same as the one used in [3].

Second, in some problems, the next state s' and the reward r is one-to-one matched. As a result, we have $p(r|s,a) = p(s'|s,a)$, substituting which into (2.6) gives

$$
v(s) = \sum_a \pi(a|s) \sum_{s'} p(s'|s, a) \left[r + \gamma v(s') \right]. \tag{2.8}
$$

The assumption of $p(r|s,a) = p(s'|s,a)$ is valid for some simple problems. For example, in the grid-world examples, the next state and the immediate reward are one-to-one matched. However, it may not be true in general. Hence, the expression in (2.9) is not as general as the one in (2.6).

2.5 Solving state values from the Bellman equation

Calculating the state values of a given policy is a fundamental problem in RL. This problem is often referred to as *policy evaluation*, an important step in many RL algorithms. In this section, we show how to obtain state values by solving the Bellman equation.

The Bellman equation in (2.6) is an *elementwise form*. It indicates that the value of a state depends on the values of some other states. Since such an equation is valid for every state, there are $|\mathcal{S}|$ equations like this. If we put all these equations together, we obtain a set of linear equations, which can be expressed concisely in a *matrix-vector form*. Then, it will be much more clear to see how to solve the state values. The matrix-vector form is elegant and will be used frequently to analyze the Bellman equation. Next, we first present the matrix-vector form of the Bellman equation and then introduce two ways to solve the equation.

2.5.1 Matrix-vector form of the Bellman equation

To derive the matrix-vector form, we first rewrite the Bellman equation in (2.6) as

$$v_\pi(s) = r_\pi(s) + \gamma \sum_{s'} p_\pi(s'|s) v_\pi(s'), \tag{2.9}$$

where

$$r_\pi(s) \triangleq \sum_a \pi(a|s) \sum_r p(r|s,a) r, \qquad p_\pi(s'|s) \triangleq \sum_a \pi(a|s) \sum_{s'} p(s'|s,a).$$

Here, $r_\pi(s)$ denotes the mean of the immediate rewards that can be obtained starting from s under policy π, and $p_\pi(s'|s)$ is the probability jumping from s to s' under policy π.

In order to write in a matrix form, suppose that the states are indexed as s_i ($i = 1, \ldots, n$). For state s_i, the Bellman equation is

$$v_\pi(s_i) = r_\pi(s_i) + \gamma \sum_{s_j} p_\pi(s_j|s_i) v_\pi(s_j).$$

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 $P_\pi \in \mathbb{R}^{n \times n}$ where $[P_\pi]_{ij} = p_\pi(s_j|s_i)$. Then we have the matrix-vector form as

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

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

To illustrate, consider the policy in Figure 2.5. 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 equation gives

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

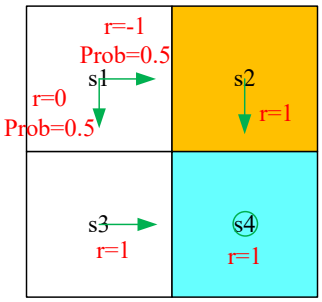


Figure 2.5: An example to demonstrate the matrix-vector form of the Bellman equation.

2.5.2 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

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

The matrix $(I - \gamma P_\pi)^{-1}$ has some interesting properties.

– $(I - \gamma P_\pi)^{-1} \geq I$, indicating that every element of $(I - \gamma P_\pi)^{-1}$ is nonnegative and more specifically no less than that of the identity matrix. This fact is because P_π has nonnegative entries and hence $(I - \gamma P_\pi)^{-1} = I + \gamma P_\pi + \gamma^2 P_\pi^2 + \cdots \geq I \geq 0$. In this book, $\geq$ or $\leq$ is elementwise.

– If a vector $r \geq 0$, then $(I - \gamma P_\pi)^{-1} r \geq r \geq 0$. The proof is as follows. $(I - \gamma P_\pi)^{-1} r = r + \gamma P_\pi r + \gamma^2 P_\pi^2 r + \cdots \geq r$.

– If $r_1 \geq r_2$, then $(I - \gamma P_\pi)^{-1} r_1 \geq (I - \gamma P_\pi)^{-1} r_2$. This property directly follows from the second one.

2.5.3 Iterative solution

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

$$v_{k+1} = r_\pi + \gamma P_\pi v_k.$$

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

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

Interested readers may see the proof below.

Proof. Define the error as $\delta_k = v_k - v_\pi$. We only need to show $\delta_k \to 0$. 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,

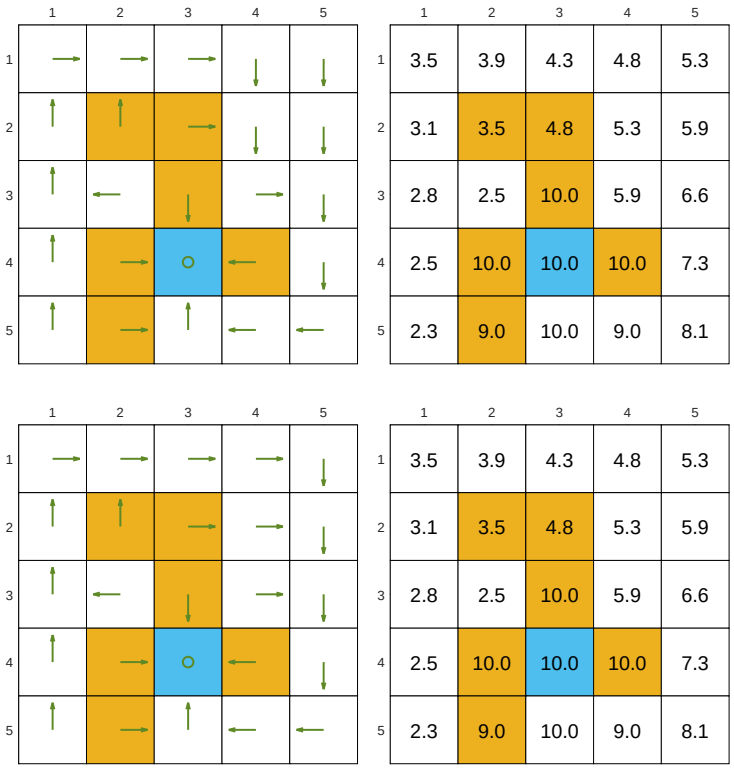
$$\delta_{k+1} = \gamma P_\pi \delta_k = \gamma^2 P_\pi^2 \delta_{k-1} = \cdots = \gamma^{k+1} P_\pi^{k+1} \delta_0.$$

Since P_π is a nonnegative stochastic matrix satisfying $P_\pi \mathbf{1} = \mathbf{1}$ where $\mathbf{1} = [1, \dots, 1]^T$, we have $0 \le P_\pi^k \le 1$ 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 $\gamma^k \to 0$ and hence $\delta_{k+1} = \gamma^{k+1} P_\pi^{k+1} \delta_0 \to 0$ as $k \to \infty$. □

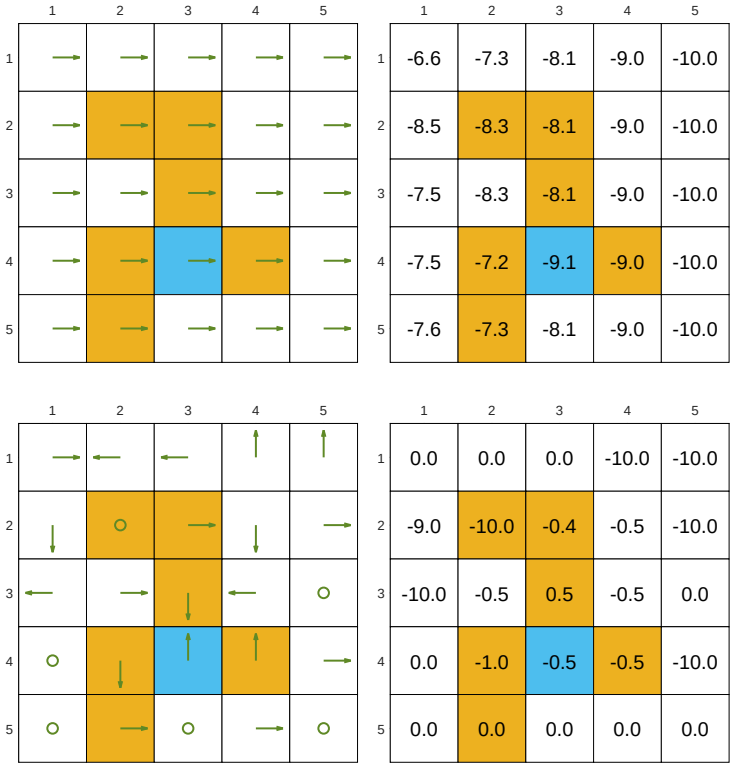
2.5.4 Illustrative examples

We next show some grid-world examples to demonstrate the algorithms introduced above. The orange cells represent forbidden areas. The blue cell represents the target area. The reward setting is $r_{\text{boundary}} = r_{\text{forbidden}} = -1$ and $r_{\text{target}} = 1$. Here, the discount rate is $\gamma = 0.9$.

Figure 2.6(a) shows two "good" policies and their corresponding state values. The two policies have exactly the same state values, although they are different in terms of the



(a) Two "good" policies and their state values. The state values of the two policies are exactly the same although the two policies are different in terms of 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 2.6: Examples of policies and the corresponding state values.

top two states in the fourth column. For the two specific states, the agent either moving rightwards or moving downwards makes no difference in terms of the state values.

Figure 2.6(b) shows two "bad" policies and their corresponding state values. The two policies are bad because the actions of many states are not reasonable intuitively. Such intuition is supported by the state values. As can be seen, the state values of these two policies are much lower than those of the good policies in Figure 2.6(a).

2.6 From state value to action value

While we have introduced the concept of state value, we now turn to *action value*, which indicates the "value" of taking an action. Action value is a very important concept. As we will see later in the book, we often care about action values more than state values because action values can be used to generate optimal policies.

The definition of action value is

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

for any state $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$. It is clear that the action value is the average return that can be obtained after taking an action. Note that $q_\pi(s,a)$ depends on a state-action pair (s,a) instead of an action alone.

What is the relationship between action value and state value? First, it follows from the properties of conditional expectation that

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

Hence,

$$v_\pi(s) = \sum_a \pi(a|s) q_\pi(s,a). \tag{2.11}$$

As a result, the value of a state is the expectation of the action values associated to that state. Second, since the state value is given by $v_\pi(s) = \sum_a \pi(a|s)\Big[\sum_r p(r|s,a)r + \gamma \sum_{s'} p(s'|s,a)v_\pi(s')\Big]$ as in the Bellman equation, comparing it with (2.11) leads to

$$q_\pi(s,a) = \sum_r p(r|s,a)r + \gamma \sum_{s'} p(s'|s,a)v_\pi(s'). \tag{2.12}$$

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.

It is interesting that both (2.11) and (2.12) describe the relationship between state value and action value. They are the two sides of the same coin: (2.11) shows how to

obtain state values from action values, whereas (2.12) shows the reverse that is how to obtain action values from state values.

2.6.1 Illustrative examples

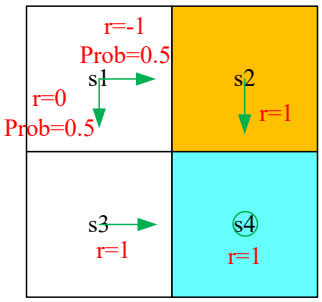


Figure 2.7: An example to demonstrate the calculation of action values.

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

Consider the example in Figure 2.7. The policy is stochastic. For the sake of simplicity, we only consider the action values of s_1. In particular, taking action a_2 at state s_1 would get a total reward of

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

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 computing the action values of a_1, a_4, a_5. One may say that, since the policy would not take the actions of a_1, a_4, a_5, we have $q_\pi(s_1, a_1) = q_\pi(s_1, a_4) = q_\pi(s_1, a_5) = 0$, or we do not need to calculate them. In fact, the action value at a specific state does not depend on the policy at that specific state. In particular, since the next state is still s_1 after taking a_1, a_4, or a_5 at s_1, we have

$$\begin{aligned} q_\pi(s_1, a_1) &= -1 + \gamma v_\pi(s_1), \\ q_\pi(s_1, a_4) &= -1 + \gamma v_\pi(s_1), \\ q_\pi(s_1, a_5) &= 0 + \gamma v_\pi(s_1). \end{aligned}$$

The reason that we care about the values of those actions that may not be taken by the current policy is that these actions may be good and missed by the current policy. Therefore, we have to explore the values of these actions.

Finally, after computing the action values, we can also calculate the state value fol-

lowing (2.12):

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

2.6.2 Bellman equation in terms of action values

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

Substituting (2.11) into (2.12) gives

$$q_\pi(s, a) = \sum_r p(r|s, a)r + \gamma \sum_{s'} p(s'|s, a) \sum_{a' \in \mathcal{A}(s')} \pi(a'|s')q_\pi(s', a'), \tag{2.13}$$

which is an equation of action values. Suppose each state has the same number of actions. The matrix-vector form of (2.13) is

$$q_\pi = \tilde{r} + \gamma P \Pi q_\pi, \tag{2.14}$$

where $q_\pi \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ is the action value vector indexed by state-action pairs. In particular, the (s, a)th element is $[q_\pi]_{(s,a)} = q_\pi(s, a)$. Here, $\tilde{r} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ is the immediate reward vector indexed by state-action pairs. In particular,

$$[\tilde{r}]_{(s,a)} = \sum_r p(r|s, a)r.$$

Here, $P \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}|}$ is the probability transition matrix, whose row is indexed by state-action pairs and column indexed by states. In particular,

$$[P]_{(s,a),s'} = p(s'|s, a)$$

and $\Pi \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}||\mathcal{A}|}$ describes the policy π. In particular,

$$\Pi_{s',(s',a')} = \pi(a'|s')$$

and the other entries of Π are zero. Π is a block diagonal matrix with each block as a $1 \times |\mathcal{A}|$ vector.

Compared to the Bellman equation in terms of state values, the one in terms of action values has some unique features. For example, $\tilde{r}$ and P are independent of the policy and merely determined by the state model. The policy is totally contained in Π. It can also be verified that (2.14) is also a contraction mapping and hence has a unique solution, which can be solved iteratively. More details can be found in [4].

2.7 Summary

State value is the most important concept introduced in this chapter. Mathematically, it is the expectation or mean value of the returns that the agent can obtain 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. Such a phenomenon might be the most confusing part for beginners. It is related to an important concept called bootstrapping, which means calculating something from itself. Although bootstrapping may be confusing intuitively, it is crystal 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 relationship among the values of all states. If we put all the equations together into a matrix-vector form, it will be clear that solving the state values is simply solving a linear equation.

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

Finally, another important concept, action value, describes the value of taking one action at a state. As we will see later in this book, action value plays a more direct role than state value when we attempt to find optimal policies.

The Bellman equation is not restricted to the RL domain. 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 the discrete Markov decision process. A complete treatment of this topic can be found in [2].

2.8 Q&A

– Q: What is the relationship between state value and return?

  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 value?

  A: State value can be used to evaluate the "goodness" of a policy. In fact, the optimal policies are defined based on state values. This point will be clearer in the next chapter when we introduce the Bellman optimality equation.

– Q: Why do we care about the Bellman equation?

  A: The Bellman equation describes the relationship among the values of all the states. It is the tool to analyze the state values.

– Q: Why solving the Bellman equation is called policy evaluation?

A: Solving the Bellman equation is to solve the state values. Since state values can be used to evaluate the "goodness" of a policy, solving the Bellman equation can be interpreted as evaluating the policy.

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

A: The Bellman equation actually refers to a set of linear equations established for all the states. In order 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: Why do we care about action values?

A: That is simply because the ultimate goal of reinforcement learning is to find out which actions are more valuable. This point will be clearer in the following chapters.

– Q: What is the relationship between state value and action value?

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

– Q: Why do we care about the values of those actions that the given policy would not take?

A: Although some actions would not be taken by the given policy, it does not mean these actions are not good. On the contrary, it is possible that the given policy is not good and misses the best action. Therefore, we must keep exploring to check the values of these actions.