第2章 状态价值与贝尔曼方程

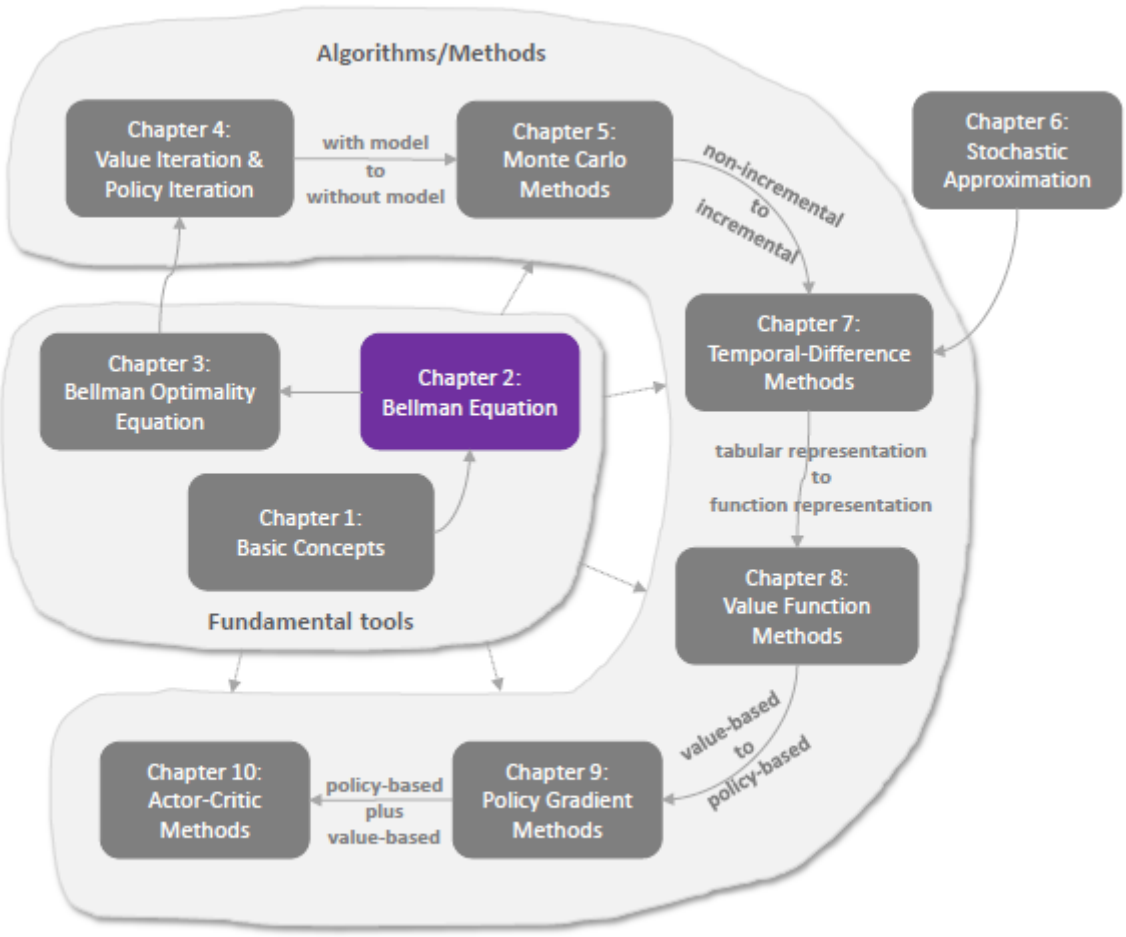


Figure 2.1: Where we are in this book.

图 2.1：我们在本书中的位置。

本章介绍了一个*核心概念*和一个*重要工具*。核心概念是***状态价值***（state value），其定义为智能体如果遵循给定的策略所能获得的平均奖励。状态价值越大，对应的策略就越好。状态价值可以作为一个度量指标，用于评估一个策略的优劣。虽然状态价值很重要，但我们该如何分析它们呢？答案就是***贝尔曼方程***（Bellman equation），它是分析状态价值的重要工具。简而言之，贝尔曼方程描述了所有状态的价值之间的关系。通过求解贝尔曼方程，我们可以得到状态价值。这个过程被称为***策略评估（policy evaluation）***，它是强化学习中的一个基本概念。最后，本章介绍了另一个重要的概念，称为*动作价值*（action value）。

2.1 激励性示例 1：为什么回报很重要？

上一章介绍了回报的概念。事实上，回报在强化学习中起着基础性的作用，因为它们可以用来评估一个策略是好是坏。下面的例子证明了这一点。

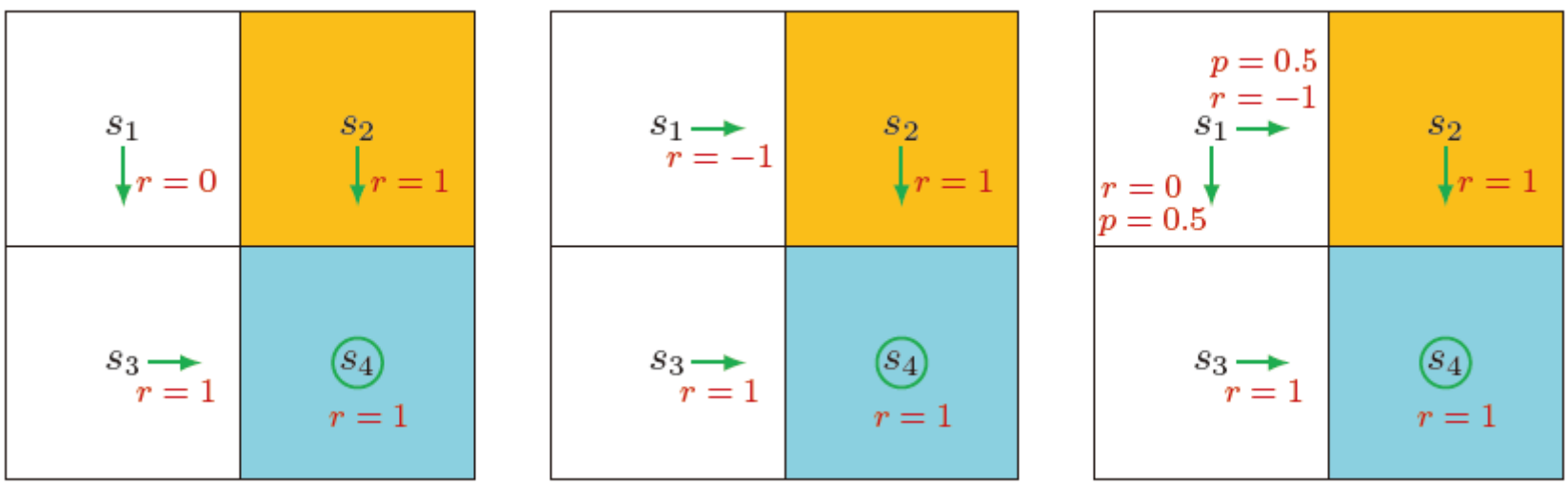


图 2.2：展示回报重要性的示例。这三个示例对 s_1 采取了不同的策略。

考虑图 2.2 中所示的三个策略。可以看出，这三个策略在 s_1 处是不同的。哪个最好，哪个最差？直观上，最左边的策略是最好的，因为从 s_1 出发的智能体可以避开禁止区域。中间的策略直观上更差，因为从 s_1 出发的智能体移动到了禁止区域。最右边的策略介

于两者之间，因为它有 0.5 的概率进入禁止区域。

虽然上述分析基于直觉，但紧随其后的一个问题是，我们能否用数学来描述这种直觉。答案是肯定的，并且依赖于回报的概念。特别地，假设智能体从 s_1 出发。

◇ 遵循第一个策略，轨迹是 $s_1 \to s_3 \to s_4 \to s_4 \cdots$。相应的折扣回报是

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

其中 $\gamma \in (0, 1)$ 是折扣率。

◇ 遵循第二个策略，轨迹是 $s_1 \to s_2 \to s_4 \to s_4 \cdots$。折扣后的回报是

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

- 遵循第三个策略，可能得到两条轨迹。一条是 $s_1 \to s_3 \to s_4 \to s_4 \cdots$，另一条是 $s_1 \to s_2 \to s_4 \to s_4 \cdots$。这两条轨迹出现的概率都是 0.5。那么，从 s_1 出发能获得的平均回报是

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

通过比较这三个策略的回报，我们注意到

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

对于任意 γ 值都成立。不等式 (2.1) 表明第一个策略是最好的，因为它的回报最大；第二个策略是最差的，因为它的回报最小。这一数学结论与上述直觉是一致的：第一个策略最好，因为它可以避免进入禁止区域；第二个策略最差，因为它会导致进入禁止区域。

上述例子表明，回报可以用来评估策略：如果遵循某个策略获得的回报更大，那么该策略就更好。最后，值得注意的是，return_3 并不严格符合回报的定义，因为它更像是一个期望值。后面我们会清楚地看到，return_3 实际上是一个状态价值。

2.2 激励性示例 2：如何计算回报？

虽然我们已经证明了回报的重要性，但紧接着的一个问题是，当遵循一个给定的策略时，如何计算回报。

计算回报有两种方法。

- 第一种很简单，就是根据定义：回报等于沿轨迹收集的所有奖励的折扣和。考虑图 2.3 中的例子。令 v_i 表示从 s_i 出发获得的回报，其中 $i = 1, 2, 3, 4$。那么，当从图 2.3 中的四个状态出发时，回报可以计算为

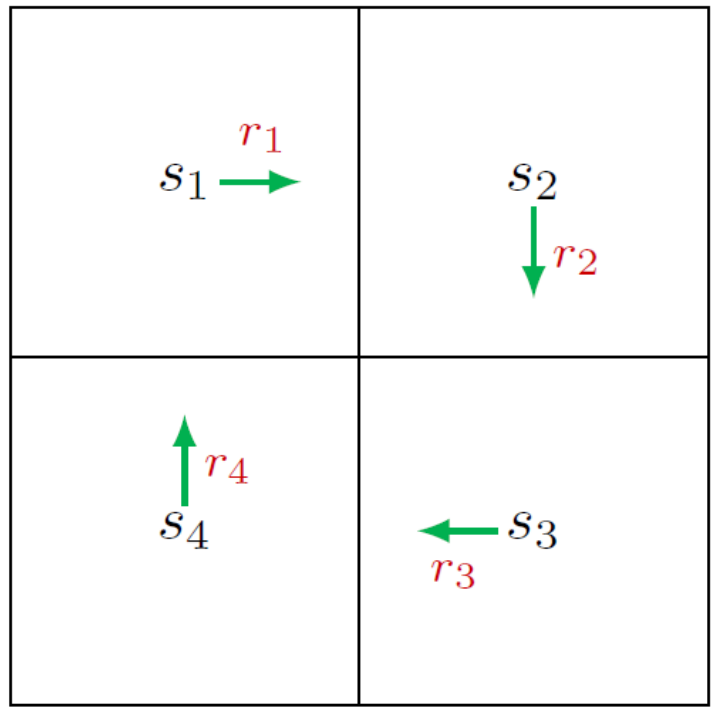


图 2.3：演示如何计算Return的示例。本例中没有目标单元格或禁止单元格。

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

- 第二种方法更重要，它基于**bootstrapping**的思想。通过观察 (2.2) 中回报的表达式，我们可以将其重写为

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

上述方程揭示了一个有趣的现象，即Return的值相互依赖。具体来说，v_1 依赖于 v_2 ，v_2 依赖于 v_3 ，v_3 依赖于 v_4 ，v_4 依赖于 v_1 。这反映了**bootstrapping**的思想，即从自身获取某些量的值。

乍一看，**bootstrapping**似乎是一个死循环，因为未知值的计算依赖于另一个未知值。事实上，如果我们从数学角度来看，**bootstrapping**更容易理解。特别地，方程组 (2.3) 可以改写为线性矩阵-向量方程：

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

它可以紧凑地写为

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

因此，v 的值可以很容易地计算为 $v = (I - \gamma P)^{-1} r$ ，其中 I 是具有适当维度的单位矩阵。人们可能会问 $I - \gamma P$ 是否总是可逆的。答案是肯定的，并在 2.7.1 节中进行了解释。

事实上，(2.3) 是这个简单示例的**贝尔曼方程（Bellman equation）**。虽然简单，但 (2.3) 展示了贝尔曼方程的核心思想：从一个状态出发获得的Return依赖于从其他状态出发获得的回报。**bootstrapping**的思想和通用场景下的贝尔曼方程将在接下来的章节中进行形式化。

2.3 状态价值

我们提到过Return可以用来评估策略。然而，它们不适用于随机系统，因为从一个状态出发可能会导致不同的回报。受此问题的启发，我们在本节中引入**状态价值（state value）**的概念。

首先，我们需要引入一些必要的符号。考虑一系列时间步 $t = 0, 1, 2, \ldots$。在时刻 t，智能体处于状态 S_t，根据策略 π 采取的动作是 A_t。下一个状态是 S_{t+1}，获得的即时奖励是 R_{t+1}。这个过程可以简洁地表示为

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

注意 $S_t, S_{t+1}, A_t, R_{t+1}$ 都是随机变量。此外，$S_t, S_{t+1} \in \mathcal{S}$，$A_t \in \mathcal{A}(S_t)$，并且 $R_{t+1} \in \mathcal{R}(S_t, A_t)$。

从 t 开始，我们可以得到一个状态-动作-奖励轨迹：

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

根据定义，沿该轨迹的折扣回报是

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

其中 $\gamma \in (0, 1)$ 是折扣率。注意 G_t 是一个随机变量，因为 $R_{t+1}, R_{t+2}, \ldots$ 都是随机变量。

既然 G_t 是一个随机变量，我们可以计算它的期望值（也称为期望或均值）：

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

在这里，$v_\pi(s)$ 被称为*状态价值函数*（state-value function）或简称为 s 的*状态价值*（state value）。以下是一些重要的说明。

- $v_\pi(s)$ 依赖于 s。这是因为它的定义是一个条件期望，条件是智能体从 $S_t = s$ 出发。
- $v_\pi(s)$ 依赖于 π。这是因为轨迹是遵循策略 π 生成的。对于不同的策略，状态价值可能是不同的。
- $v_\pi(s)$ 不依赖于 t。如果智能体在状态空间中移动，t 代表当前的时间步。一旦给定了策略，$v_\pi(s)$ 的值就是确定的。

状态价值与回报之间的关系进一步阐明如下。

当策略和系统模型都是确定性的时，从一个状态出发总是会导致相同的轨迹。在这种情况下，**从一个状态出发获得的回报等于该状态的价值**。

相比之下，当策略或系统模型是随机的时，**从同一个状态出发可能会产生不同的轨迹。在这种情况下，不同轨迹的回报是不同的，而状态价值是这些回报的均值**。

虽然如 2.1 节所示，回报可以用来评估策略，但使用状态价值来评估策略更加正式：产生更大状态价值的策略更好。因此，状态价值构成了强化学习中的一个核心概念。虽然状态价值很重要，但紧接着的一个问题是如何计算它们。这个问题将在下一节中回答。

2.4 贝尔曼方程

我们现在介绍贝尔曼方程（Bellman equation），这是一个用于分析状态价值的数学工具。简而言之，贝尔曼方程是一组描述所有状态价值之间关系的线性方程。

状态价值是return的均值

我们接下来推导贝尔曼方程。首先，注意 G_t (轨迹的折扣回报)可以重写为

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

|注释：在时刻 t，智能体处于状态 S_t，根据策略 π 采取的动作是 A_t。下一个状态是 S_{t+1}，获得的即时奖励是 R_{t+1}。|

其中 $G_{t+1} = R_{t+2} + \gamma R_{t+3} + \ldots$。这个方程建立了 G_t 和 G_{t+1} 之间的关系。然后，状态价值可以写为

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

下面对 (2.4) 中的两项进行分析。

- ◇ 第一项 $\mathbb{E}[R_{t+1} | S_t = s]$ 是即时奖励的期望。利用全期望公式（见附录 A），它可以计算如下：

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

- 这里，$\mathcal{A}$ 和 $\mathcal{R}$ 分别是可能的动作集合和奖励集合。值得注意的是，对于不同的状态，$\mathcal{A}$ 可能是不同的。在这种情况下，$\mathcal{A}$ 应写作 $\mathcal{A}(s)$。同样地，$\mathcal{R}$ 也可能依赖于 (s,a)。为了本书的简洁性，我们省略了对 s 或 (s,a) 的依赖标记。然而，即使存在这种依赖关系，结论仍然成立。
- ◇ 第二项 $\mathbb{E}[G_{t+1}|S_t=s]$ 是未来奖励的期望。它可以计算如下：

$$\begin{aligned}\mathbb{E}[G_{t+1}|S_t=s] &= \sum_{s'\in\mathcal{S}}\mathbb{E}[G_{t+1}|S_t=s,S_{t+1}=s']p(s'|s) \\ &= \sum_{s'\in\mathcal{S}}\mathbb{E}[G_{t+1}|S_{t+1}=s']p(s'|s) \quad \text{(由于马尔可夫性质)} \\ &= \sum_{s'\in\mathcal{S}}v_\pi(s')p(s'|s) \\ &= \sum_{s'\in\mathcal{S}}v_\pi(s')\sum_{a\in\mathcal{A}}p(s'|s,a)\pi(a|s).\end{aligned} \tag{2.6}$$

- 上述推导使用了 $\mathbb{E}[G_{t+1}|S_t=s,S_{t+1}=s']=\mathbb{E}[G_{t+1}|S_{t+1}=s']$ 这一事实，这是由于马尔可夫性质，即未来奖励仅依赖于当前状态，而不依赖于之前的状态。

将 (2.5)-(2.6) 代入 (2.4) 可得

$$\begin{aligned}v_\pi(s) &= \mathbb{E}[R_{t+1}|S_t=s]+\gamma\mathbb{E}[G_{t+1}|S_t=s], \\ &= \sum_{a\in\mathcal{A}}\pi(a|s)\sum_{r\in\mathcal{R}}p(r|s,a)r+\gamma\sum_{a\in\mathcal{A}}\pi(a|s)\sum_{s'\in\mathcal{S}}p(s'|s,a)v_\pi(s') \\ &= \underbrace{\sum_{a\in\mathcal{A}}\pi(a|s)\sum_{r\in\mathcal{R}}p(r|s,a)r}_{\text{即时奖励的均值}}+\gamma\underbrace{\sum_{a\in\mathcal{A}}\pi(a|s)\sum_{s'\in\mathcal{S}}p(s'|s,a)v_\pi(s')}_{\text{未来奖励的均值}} \\ &= \sum_{a\in\mathcal{A}}\pi(a|s)\left[\sum_{r\in\mathcal{R}}p(r|s,a)r+\gamma\sum_{s'\in\mathcal{S}}p(s'|s,a)v_\pi(s')\right], \quad \text{对于所有 } s\in\mathcal{S}.\end{aligned} \tag{2.7}$$

这个方程就是*贝尔曼方程*（Bellman equation），它刻画了状态价值之间的关系。它是设计和分析强化学习算法的基本工具。

贝尔曼方程乍一看似乎很复杂。事实上，它有一个清晰的结构。下面给出一些说明。

- $v_\pi(s)$ 和 $v_\pi(s')$ 是待计算的未知状态价值。对于初学者来说，在 $v_\pi(s)$ 依赖于另一个未知量 $v_\pi(s')$ 的情况下如何计算它可能会让人感到困惑。必须注意的是，**贝尔曼方程是指针对所有状态的一组线性方程（这句话很重要），**而不是单个方程。如果我们把这些方程放在一起，如何计算所有状态价值就变得清晰了。详细信息将在 2.7 节中给出。
- $\pi(a|s)$ 是一个给定的策略。由于状态价值可以用来评估策略，通过贝尔曼方程求解状态价值是一个*策略评估过程*（policy evaluation process），这是许多强化学习算法中的一个重要过程，我们将在本书后面看到。
- $p(r|s,a)$ 和 $p(s'|s,a)$ 代表**系统模型(system model)**。我们将在 2.7 节首先展示如何利用该模型计算状态价值，然后在本书后面展示如何使用无模型算法在*没有*该模型的情况下计算状态价值。
  - 注：这里可以的模型是可以有这个转移概率，这个是明确的

除了 (2.7) 中的表达式外，读者在文献中可能还会遇到贝尔曼方程的其他表达式。接下来我们介绍两种等价的表达式。

首先，根据全概率公式可得

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

那么，（带入全概率公式）方程 (2.7) 可以重写为

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

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

这是文献 [3] 中使用的表达式。

其次，在某些问题中，奖励 r 可能仅依赖于下一个状态 s'。因此，我们可以将奖励写为 $r(s')$，从而 $p(r(s')|s,a)=p(s'|s,a)$，将其代入 (2.7) 可得

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

2.5 说明贝尔曼方程的示例

接下来我们使用两个例子来演示如何写出贝尔曼方程并逐步计算状态价值。建议读者仔细阅读这些示例，以更好地理解贝尔曼方程。

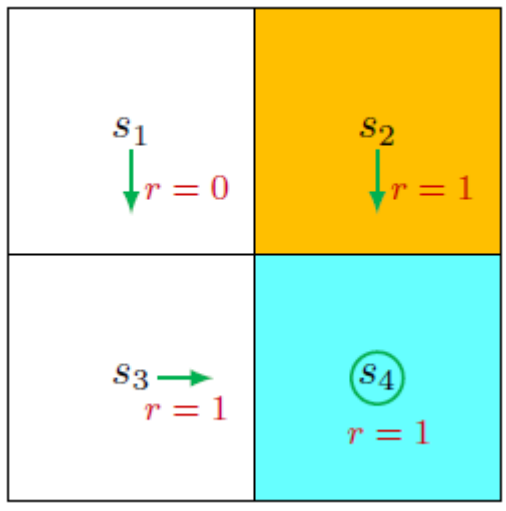


图 2.4：演示贝尔曼方程的示例。本例中的策略是确定性的。

- 考虑图 2.4 中展示的第一个例子，其中的策略是确定性的。我们接下来写出贝尔曼方程，然后从中求解状态价值。

首先，考虑状态 s_1。在该策略下，采取动作的概率为 $\pi(a=a_3|s_1)=1$ 且 $\pi(a\neq a_3|s_1)=0$。状态转移概率为 $p(s'=s_3|s_1,a_3)=1$ 且 $p(s'\neq s_3|s_1,a_3)=0$。奖励概率为 $p(r=0|s_1,a_3)=1$ 且 $p(r\neq 0|s_1,a_3)=0$。将这些值代入 (2.7) 可得 $v_\pi(s_1)=0+\gamma v_\pi(s_3)$.

有趣的是，虽然 (2.7) 中贝尔曼方程的表达式看起来很复杂，但对于这个特定状态的表达式却非常简单。

同样地，可以得到

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

我们可以从这些方程中求解状态价值。由于方程很简单，我们可以手动求解。更复杂的方程可以通过 2.7 节中介绍的算法来求解。在这里，状态价值可以求解为

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

此外，如果我们设 $\gamma=0.9$，那么

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

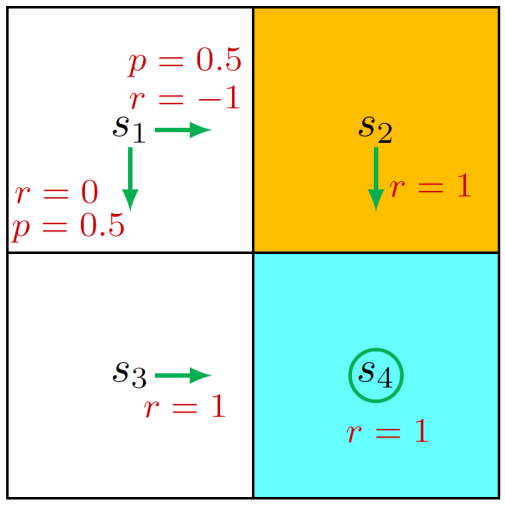


图 2.5：演示贝尔曼方程的示例。本例中的策略是随机的。

- 考虑图 2.5 中展示的第二个例子，其中的策略是随机的。我们接下来写出贝尔曼方程，然后从中求解状态价值。

在状态 s_1 ，向右走和向下走的概率都等于 0.5。数学上，我们要 $\pi(a = a_2|s_1) = 0.5$ 和 $\pi(a = a_3|s_1) = 0.5$。状态转移概率是确定性的，因为 $p(s' = s_3|s_1, a_3) = 1$ 且 $p(s' = s_2|s_1, a_2) = 1$。奖励概率也是确定性的，因为 $p(r = 0|s_1, a_3) = 1$ 且 $p(r = -1|s_1, a_2) = 1$。将这些值代入 (2.7) 可得

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

同理，可以得到

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

状态价值可以从上述方程中求解。由于这些方程是很简单，我们可以手动求解状态价值并得到

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

此外，如果我们设 $\gamma = 0.9$ ，那么

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

如果我们比较上述示例中两个策略的状态价值，可以看出

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

这表明图 2.4 中的策略更好，因为它具有更大的状态价值。这一数学结论与直觉一致，即第一个策略更好，因为当智能体从 s_1 出发时，它可以避免进入禁止区域。总之，上述两个示例表明状态价值可以用来评估策略。

2.6 贝尔曼方程的矩阵-向量形式

(2.7) 中的贝尔曼方程是*元素形式*（elementwise form）。由于它对每个状态都成立，我们可以将所有这些方程组合起来，简洁地写成*矩阵-向量形式*（matrix-vector form），这将经常用于分析贝尔曼方程。

为了推导矩阵-向量形式，我们首先将 (2.7) 中的贝尔曼方程重写为

- 2.7方程

- $$\begin{aligned} v_\pi(s) &= \mathbb{E}[R_{t+1}|S_t=s] + \gamma\mathbb{E}[G_{t+1}|S_t=s], \\ &= \underbrace{\sum_{a\in\mathcal{A}}\pi(a|s)\sum_{r\in\mathcal{R}}p(r|s,a)r}_{\text{即时奖励的均值}} + \gamma\underbrace{\sum_{a\in\mathcal{A}}\pi(a|s)\sum_{s'\in\mathcal{S}}p(s'|s,a)v_\pi(s')}_{\text{未来奖励的均值}} \\ &= \sum_{a\in\mathcal{A}}\pi(a|s)\left[\sum_{r\in\mathcal{R}}p(r|s,a)r + \gamma\sum_{s'\in\mathcal{S}}p(s'|s,a)v_\pi(s')\right], \quad \text{对于所有 } s\in\mathcal{S}. \end{aligned} \tag{2.7}$$

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

其中

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

这里，$r_\pi(s)$ 表示即时奖励的均值，$p_\pi(s'|s)$ 是在策略 π 下从 s 转移到 s' 的概率。

假设状态索引为 s_i，其中 $i=1,\dots,n$，且 $n=|S|$。对于状态 s_i，(2.8) 可以写为

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

令 $v_\pi = [v_\pi(s_1),\dots,v_\pi(s_n)]^T\in\mathbb{R}^n$，$r_\pi = [r_\pi(s_1),\dots,r_\pi(s_n)]^T\in\mathbb{R}^n$，以及 $P_\pi\in\mathbb{R}^{n\times n}$ 且 $[P_\pi]_{ij} = p_\pi(s_j|s_i)$。那么，(2.9) 可以写成如下的矩阵-向量形式：

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

其中 v_π 是待求解的未知量，而 r_π 和 P_π 是已知量。

矩阵 P_π 有一些有趣的性质。首先，它是一个非负矩阵（nonnegative matrix），意味着其所有元素均大于或等于零。该性质记为 $P_\pi\geq 0$，其中 0 表示具有适当维度的零矩阵。在本书中，$\geq$ 或 $\leq$ 代表逐元素比较运算。其次，P_π 是一个随机矩阵（stochastic matrix），意味着每一行中的数值之和等于 1。该性质记为 $P_\pi\mathbf{1}=\mathbf{1}$，其中 $\mathbf{1}=[1,\dots,1]^T$ 具有适当的维度。

考虑图 2.6 中所示的例子。贝尔曼方程的矩阵-向量形式为

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

将具体数值代入上述方程可得

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

可以看出 P_π 满足 $P_\pi\mathbf{1}=\mathbf{1}$。

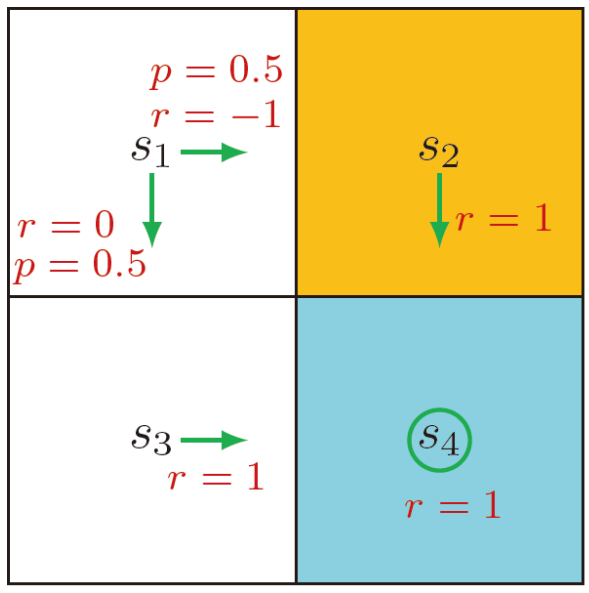


图 2.6：演示贝尔曼方程矩阵-向量形式的示例。

2.7 从贝尔曼方程求解状态价值

计算给定策略的状态价值是强化学习中的一个基本问题。这个问题通常被称为*策略评估*（policy evaluation）。在本节中，我们介绍两种从贝尔曼方程计算状态价值的方法。

2.7.1 闭式解

由于 $v_\pi = r_\pi + \gamma P_\pi v_\pi$ 是一个简单的线性方程，其*闭式解*（closed-form solution）可以很容易地得到

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

下面给出 $(I - \gamma P_\pi)^{-1}$ 的一些性质。

- $I - \gamma P_\pi$ 是可逆的。证明如下：根据盖尔什戈林圆定理（Gershgorin circle theorem）[4]，$I - \gamma P_\pi$ 的每个特征值至少位于一个盖尔什戈林圆内。第 i 个盖尔什戈林圆的中心在 $[I - \gamma P_\pi]_{ii} = 1 - \gamma p_\pi(s_i|s_i)$，半径等于 $\sum_{j \neq i} |[I - \gamma P_\pi]_{ij}| = \sum_{j \neq i} \gamma p_\pi(s_j|s_i)$。由于 $\gamma < 1$，我们知道半径小于中心的幅度：$\sum_{j \neq i} \gamma p_\pi(s_j|s_i) < 1 - \gamma p_\pi(s_i|s_i)$。因此，所有的盖尔什戈林圆都不包含原点，故 $I - \gamma P_\pi$ 没有零特征值(**虽然看不懂，但是证明出没有零特征值是有用的**)。
- $(I - \gamma P_\pi)^{-1} \geq I$，这意味着 $(I - \gamma P_\pi)^{-1}$ 的每个元素都是非负的，更具体地说，不小于单位矩阵的对应元素。这是因为 P_π 具有非负元素，因此 $(I - \gamma P_\pi)^{-1} = I + \gamma P_\pi + \gamma^2 P_\pi^2 + \cdots \geq I \geq 0$。
- 对于任意向量 $r \geq 0$，成立 $(I - \gamma P_\pi)^{-1} r \geq r \geq 0$。该性质由第二个性质推导而来，因为 $[(I - \gamma P_\pi)^{-1} - I] r \geq 0$。**由此可得，如果** $r_1 \geq r_2$，我们有 $(I - \gamma P_\pi)^{-1} r_1 \geq (I - \gamma P_\pi)^{-1} r_2$。

2.7.2 迭代解法

虽然闭式解在理论分析中很有用，但在实践中并不适用，因为它涉及矩阵求逆运算，而这仍然需要通过其他数值算法来计算。事实上，我们可以使用以下迭代算法直接求解贝尔曼方程：

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

该算法生成一个值序列 $\{v_0, v_1, v_2, \ldots\}$，其中 $v_0 \in \mathbb{R}^n$ 是对 v_π 的初始猜测。结论如下：

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

感兴趣的读者可以在方框 2.1 中查看证明。

> **方框 2.1：(2.12) 的收敛性证明**
>
> 定义误差为 $\delta_k = v_k - v_\pi$。我们只需要证明 $\delta_k \to 0$。将 $v_{k+1} = \delta_{k+1} + v_\pi$ 和 $v_k = \delta_k + v_\pi$ 代入 $v_{k+1} = r_\pi + \gamma P_\pi v_k$ 可得
>
> $$\delta_{k+1} + v_\pi = r_\pi + \gamma P_\pi (\delta_k + v_\pi),$$
>
> 它可以重写为

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

结果是，

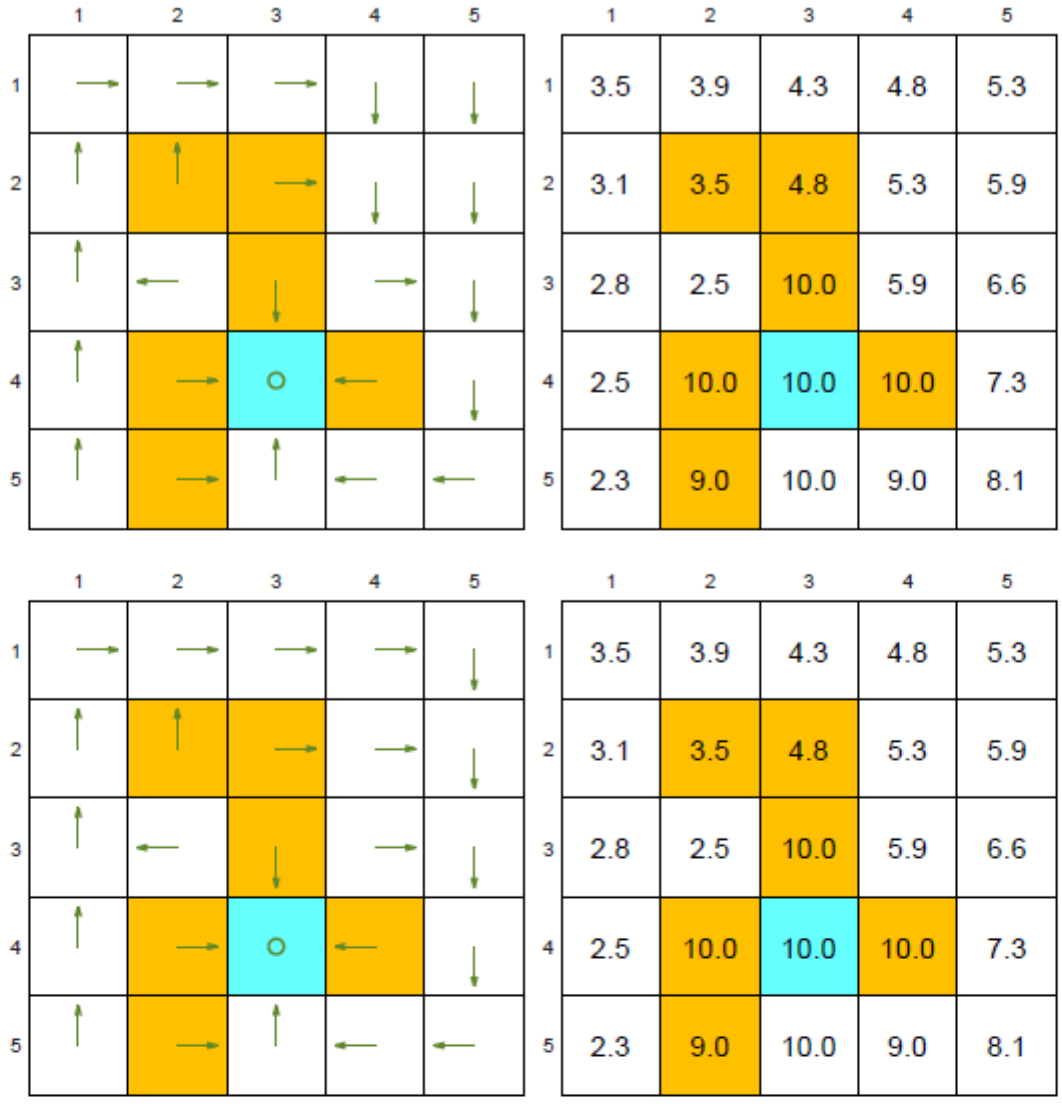
$$
\delta_{k+1} = \gamma P_\pi \delta_k = \gamma^2 P_\pi^2 \delta_{k-1} = \cdots = \gamma^{k+1} P_\pi^{k+1} \delta_0 .
$$

由于 P_π 的每个元素都是非负的且不大于 1，因此对于任意 k ，我们有 $0 \leq P_\pi^k \leq 1$ 。也就是说， P_π^k 的每个元素都不大于 1。另一方面，由于 $\gamma < 1$ ，我们知道 $\gamma^k \to 0$ ，因此当 $k \to \infty$ 时， $\delta_{k+1} = \gamma^{k+1} P_\pi^{k+1} \delta_0 \to 0$, 因此，**这个是迭代方程是收敛的**。

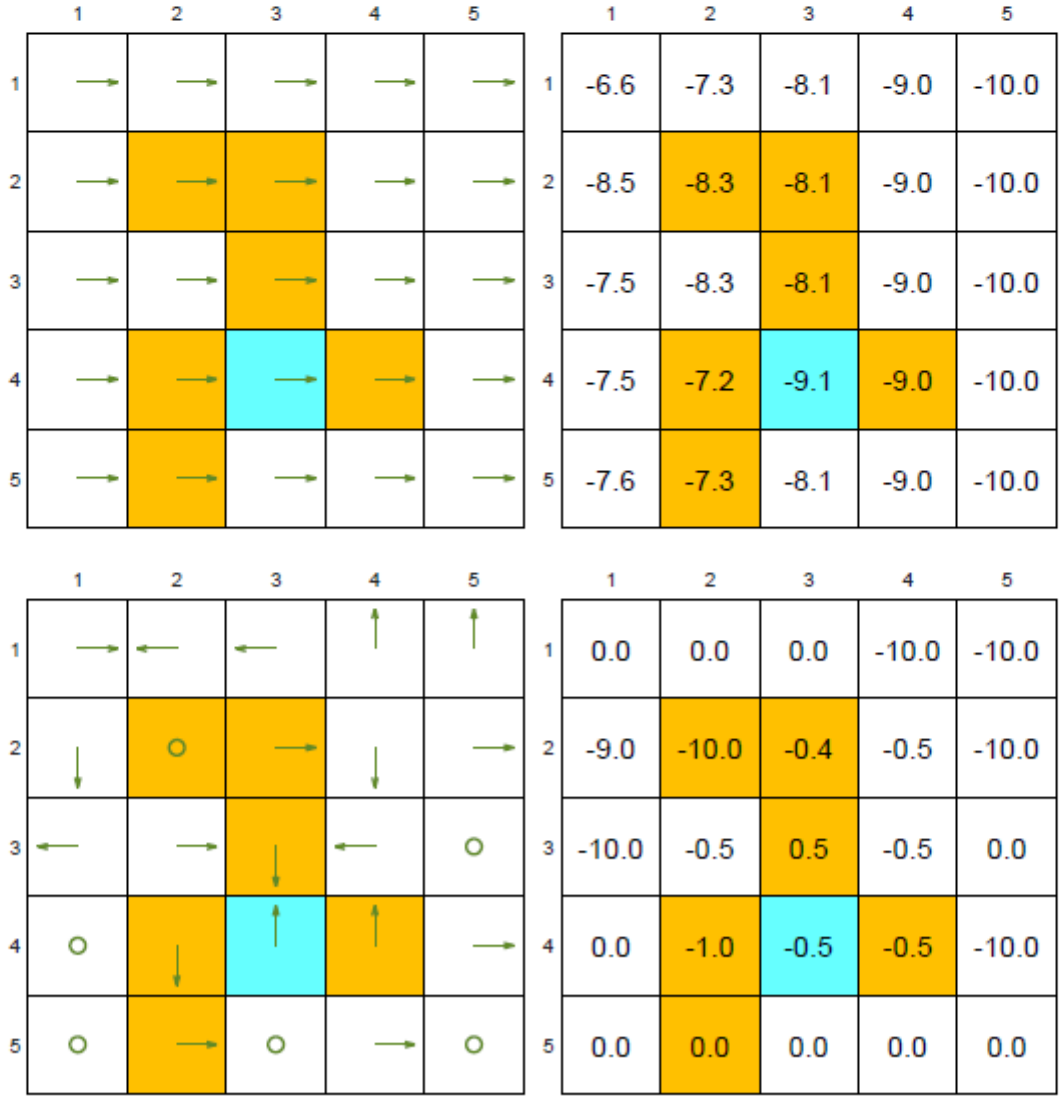
2.7.3 说明性示例

接下来，我们将应用 (2.11) 中的算法来求解一些示例的状态价值。

示例显示在图 2.7 中。橙色单元格代表禁止区域。蓝色单元格代表目标区域。奖励设置为 $r_{\text{boundary}} = r_{\text{forbidden}} = -1$ 和 $r_{\text{target}} = 1$ 且折扣率为 $\gamma = 0.9$ 。



(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.

图 2.7(a) 展示了两个“好”策略及其通过 (2.11) 获得的相应状态价值。**这两个策略具有相同的状态价值，**但在第四列顶部的两个状态上有所不同。因此，我们知道**不同的策略可能具有相同的状态价值**。

图 2.7(b) 展示了两个“坏”策略及其相应的状态价值。这两个策略很糟糕，因为许多状态的动作直观上是不合理的。这种直觉得到了所获得的状态价值的支持。可以看出，这两个策略的状态价值为负，**且远小于图 2.7(a) 中好策略的状态价值**。

2.8 从状态价值到动作价值

到目前为止，我们在本章中一直在讨论状态价值，现在我们转向**动作价值（action value）**，它表示在某个状态下采取某个动作的“价值”。虽然动作价值的概念很重要，但之所以在本章的最后一节介绍它，是因为它严重依赖于状态价值的概念。在学习动作价值之前，先很好地理解状态价值是很重要的。

状态-动作对 (s,a) 的动作价值定义为

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

可以看出，**动作价值定义为在状态下采取动作后能获得的期望回报**。必须注意的是，$q_\pi(s,a)$ 依赖于状态-动作对 (s,a) 而不仅仅是动作本身。**将其称为状态-动作价值（state-action value）可能更严谨**，但为了简单起见，通常将其称为动作价值。

动作价值和状态价值之间有什么关系？

- 首先，根据条件期望的性质可得

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

由此可得

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

结果是，状态价值是与该状态相关联的动作价值的期望。

- 其次，由于状态价值由下式给出

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

将其与 (2.13) 进行比较可得

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

可以看出，动作价值由两项组成。第一项是**即时奖励的均值**，第二项是**未来奖励的均值**。

(2.13) 和 (2.14) 都描述了状态价值和动作价值之间的关系。它们是一枚硬币的两面：(2.13) 展示了如何从动作价值获得状态价值，而 (2.14) **展示了如何从状态价值获得动作价值。**

2.8.1 说明性示例

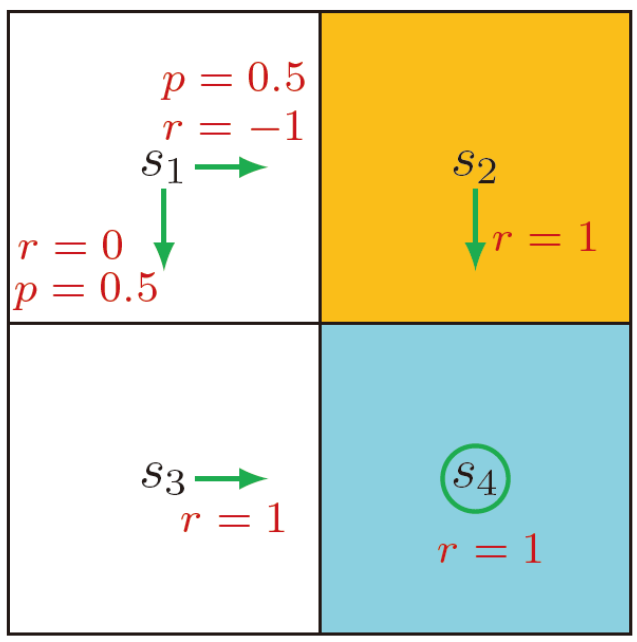


图 2.8：演示计算动作价值过程的示例。

接下来我们给出一个例子来说明计算动作价值的过程，并讨论初学者可能会犯的一个常见错误。

考虑图 2.8 中所示的随机策略。接下来我们仅检查 s_1 的动作。其他状态可以类似地检查。(s_1, a_2) 的动作价值是

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

其中 s_2 是下一个状态。类似地，可以得到

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

初学者可能会犯的一个常见错误是关于给定策略未选择的动作的价值。例如，图 2.8 中的策略只能选择 a_2 或 a_3，而不能选择 a_1, a_4, a_5。有人可能会争辩说，既然策略不选择 a_1, a_4, a_5，我们就不需要计算它们的动作价值，或者我们可以简单地设 $q_\pi(s_1, a_1) = q_\pi(s_1, a_4) = q_\pi(s_1, a_5) = 0$。这是错误的。

- (这一点很重要)首先，**即使一个动作不会被策略选中，它仍然具有动作价值**。在这个例子中，虽然策略 π 在 s_1 处不采取 a_1，但我们仍然可以计算它的动作价值，方法是观察在采取该动作后我们会得到什么。具体来说，在采取 a_1 后，智能体被弹回 s_1（因此，即时奖励为 -1），然后根据策略 π 继续在状态空间中移动（因此，未来奖励为 $\gamma v_\pi(s_1)$）。结果是，(s_1, a_1) 的动作价值为

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

同样地，对于给定的策略也不可能选择的 a_4 和 a_5，我们有

$$q_\pi(s_1, a_4) = -1 + \gamma v_\pi(s_1),$$
$$q_\pi(s_1, a_5) = 0 + \gamma v_\pi(s_1).$$

- 第二，我们为什么要关心给定策略不会选择的动作？**虽然某些动作不可能被给定的策略选中，但这并不意味着这些动作不好。有可能是给定的策略不好，所以它没有选择最好的动作。**强化学习的目的是找到最优策略。为此，我们必须保持探索所有动作以确定每个状态的更好动作。

最后，在计算动作价值之后，我们也可以根据 (2.13) 计算状态价值：

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

2.8.2 动作价值形式的贝尔曼方程

我们之前介绍的贝尔曼方程是基于状态价值定义的。事实上，它也可以用动作价值来表示。

特别地，将 (2.13) 代入 (2.14) 可得

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

这是一个动作价值的方程。上述方程对每个状态-动作对都成立。如果我们将所有这些方程放在一起，它们的矩阵-向量形式为

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

其中 q_π 是以状态-动作对为索引的动作价值向量：其第 (s, a) 个元素是 $[q_\pi]_{(s,a)} = q_\pi(s, a)$。$\tilde{r}$ 是以状态-动作对为索引的即时奖励向量：$[\tilde{r}]_{(s,a)} = \sum_{r \in \mathcal{R}} p(r|s, a) r$。矩阵 P 是概率转移矩阵，其行由状态-动作对索引，列由状态索引：$[P]_{(s,a),s'} = p(s'|s, a)$。此外，$\Pi$ 是一个块对角矩阵，其中每个块是一个 $1 \times |\mathcal{A}|$ 的向量：$\Pi_{s',(s',a')} = \pi(a'|s')$，且 Π 的其他元素为零。

与基于状态价值定义的贝尔曼方程相比，**基于动作价值定义的方程具有一些独特的特征**。例如，$\tilde{r}$ 和 P 独立于策略，仅由系统模型决定。策略被嵌入在 Π 中。可以验证 (2.15) 也是一个压缩映射（contraction mapping），并且具有唯一解，可以通过迭代求解。更多细节可以在文献 [5] 中找到。

2.9 总结

本章介绍的最重要的概念是状态价值（state value）。在数学上，状态价值是智能体从某个状态出发可以获得的期望回报。不同状态的价值是相互关联的。也就是说，状态 s 的价值依赖于其他一些状态的价值，而这些状态的价值可能进一步依赖于状态 s 本身的价值。这种现象对于初学者来说可能是本章最令人困惑的部分。它与一个称为自举（bootstrapping）的重要概念有关，该概念涉

及从自身计算某物。虽然自举在直觉上可能令人困惑，但如果我们检查贝尔曼方程的矩阵-向量形式，它就很清楚了。特别是，贝尔曼方程是一组描述所有状态价值之间关系的线性方程。

由于状态价值可用于评估策略的好坏，通过贝尔曼方程求解策略的状态价值的过程称为*策略评估*（policy evaluation）。正如我们在本书后面将看到的，策略评估是许多强化学习算法中的重要步骤。

另一个重要的概念，动作价值（action value），被引入来描述在状态下采取某一动作的价值。正如我们在本书后面将看到的，当我们试图寻找最优策略时，动作价值比状态价值发挥更直接的作用。最后，贝尔曼方程不仅限于强化学习领域。相反，它广泛存在于控制理论和运筹学等许多领域。在不同的领域中，贝尔曼方程可能有不同的表达形式。在本书中，贝尔曼方程是在离散马尔可夫决策过程下研究的。关于该主题的更多信息可以在文献 [2] 中找到。

2.10 问与答

- 问：状态价值与回报之间有什么关系？

  答：状态的价值是智能体从该状态出发所能获得的回报的均值。

- 问：我们为什么关心状态价值？

  答：状态价值可以用来评估策略。事实上，最优策略是基于状态价值定义的。这一点将在下一章变得更加清晰。

- 问：我们为什么关心贝尔曼方程？

  答：贝尔曼方程描述了所有状态价值之间的关系。它是分析状态价值的工具。

- 问：为什么求解贝尔曼方程的过程被称为策略评估？

  答：求解贝尔曼方程可以得到状态价值。由于状态价值可以用来评估一个策略，因此求解贝尔曼方程可以被解释为评估相应的策略。

- 问：为什么我们需要研究贝尔曼方程的矩阵-向量形式？

  答：贝尔曼方程是指为所有状态建立的一组线性方程。为了求解状态价值，我们需要将所有这些线性方程放在一起。矩阵-向量形式是这些线性方程的简洁表达。

- 问：状态价值和动作价值之间有什么关系？

  答：一方面，状态价值是该状态下动作价值的均值。另一方面，动作价值依赖于智能体采取该动作后可能转移到的下一个状态的价值。

- 问：我们为什么关心给定策略无法选择的动作的价值？

  答：虽然给定策略无法选择某些动作，但这并不意味着这些动作不好。相反，有可能是给定的策略不好，错过了最佳动作。为了找到更好的策略，我们必须保持探索不同的动作，即使其中一些动作可能不会被给定的策略选中。