lec07

强化学习及其在解决马尔可夫决策过程（MDP）问题中的应用，包括价值函数、贝尔曼方程及其优化

Value Function

状态值函数（State-value function）

状态值函数 $V^{\pi}(s)$ 表示在给定策略 π 下，从状态 s 开始的期望回报

$$V^{\pi}(s) = \mathbb{E}_{\pi}\{R_t \mid s_t = s\} = \mathbb{E}_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s\right\}$$

其中：

- $\mathbb{E}_{\pi}$ 表示在策略 π 下的期望
- γ 是折扣因子
- r 是奖励
- $R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$
- γ 的取值是 $(0, 1]$ 区间内的实数，这意味着 r_{t+1} 的回报权重最大，而远期的回报权重较小

动作值函数（Action-value function）

动作值函数 $Q^{\pi}(s, a)$ 表示在给定策略 π 下，在状态 s 执行动作 a 的期望回报

$$Q^{\pi}(s, a) = \mathbb{E}\{R_t \mid s_t = s, a_t = a\} = \mathbb{E}_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a\right\}$$

区别

尽管形式上相似，但两者的概念和应用场景不同：

- 状态值函数：
  - 表示从某个状态开始，遵循策略 π 所得到的期望回报。它只考虑状态，不涉及具体的动作
  - 用于评估某个状态的价值，通常用于策略评估和策略改进过程中
- 动作值函数：
  - 表示在某个状态执行特定动作，并随后遵循策略 π 所得到的期望回报。它不仅考虑状态，还考虑了具体的动作
  - 用于评估在某个状态执行特定动作的价值，通常用于选择最优动作（例如，在Q-learning中）

Bellman Equation

累计回报 R_t 可以表示为即刻奖励和未来回报的折扣和

$$
\begin{aligned}
R_t &= \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \\
&= r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots \\
&= r_{t+1} + \gamma (r_{t+2} + \gamma r_{t+3} + \gamma^2 r_{t+4} + \dots) \\
&= r_{t+1} + \gamma R_{t+1}
\end{aligned}
$$

因此，状态值函数的贝尔曼方程为

$$
\begin{aligned}
V^{\pi}(s) &= \mathbb{E}_{\pi}\{R_t \mid s_t = s\} \\
&= \mathbb{E}_{\pi}\left\{r_{t+1} + \gamma V^{\pi}(s_{t+1}) \mid s_t = s\right\}
\end{aligned}
$$

这是一组关于每个状态的方程（事实上是线性的），值函数 V^{π} 是其唯一解
这意味着状态 s 的状态值等于在该状态下采取某个动作后获得的即时奖励和从下一个状态开始的值函数的折扣的期望值

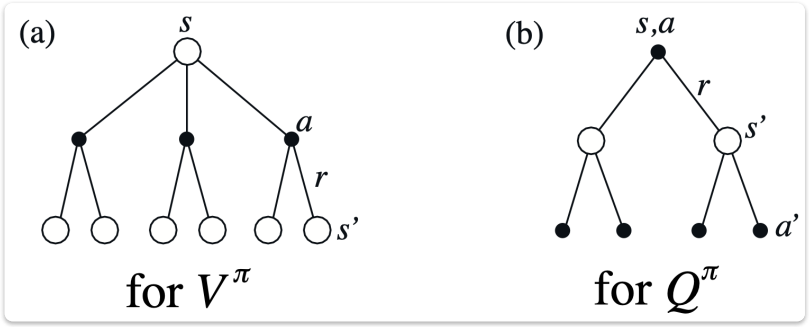
无期望操作符的形式：

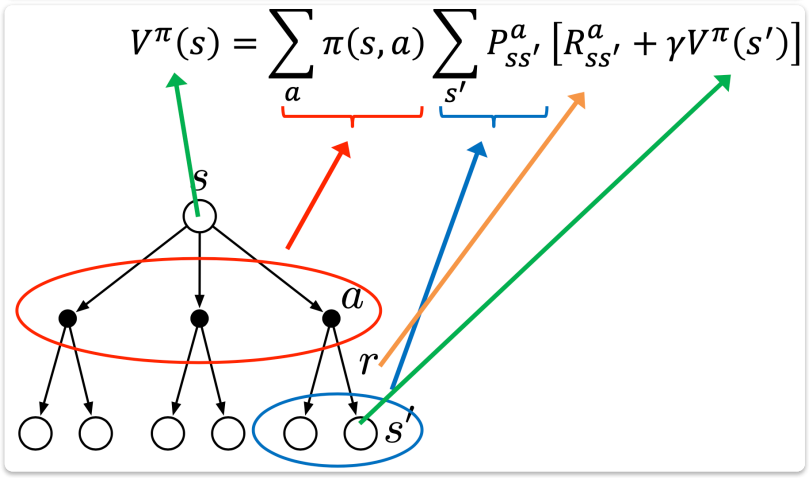
$$
\begin{aligned}
V^{\pi}(s) &= \mathbb{E}_{\pi}\{R_t \mid s_t = s\} \\
&= \mathbb{E}_{\pi}\{r_{t+1} + \gamma R_{t+1}\} \\
&= \sum_{a} \pi(s,a) \sum_{s'} P^{a}_{ss'} \left[R^{a}_{ss'} + \gamma V^{\pi}(s')\right]
\end{aligned}
$$

其中：

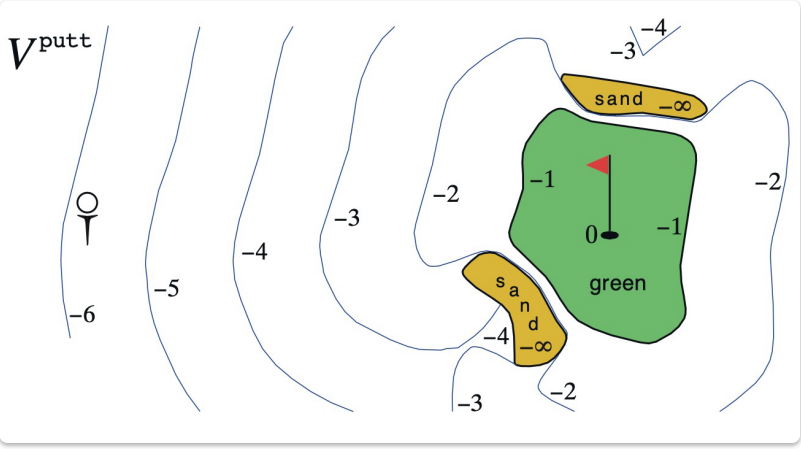
- $\pi(s,a)$ 是在状态 s 下选择动作 a 的概率
- $P^{a}_{ss'}$ 是在动作 a 下从状态 s 转移到状态 s' 的概率
- $R^{a}_{ss'}$ 是在动作 a 下从状态 s 转移到状态 s' 时的即时奖励
- $R^{a}_{ss'} + \gamma V^{\pi}(s')$：即时奖励加上从下一个状态 s' 开始的折扣回报的和

公式的直观解释：
这些方程表达了一个状态的值与其后继状态之间的递归关系，并对所有可能性进行平均，并按其发生的概率对每个可能性进行加权





高尔夫球示例



1. 状态（State）：
   - 球的位置表示当前状态
2. 奖励（Reward）：
   - 每击球一次得 -1 分，直到球进洞为止
3. 状态的价值（Value of a state）：
   - 状态的价值是指从该状态开始期望得到的累计奖励。在这个例子中，价值是球从当前位置到进洞所需的击球次数的负值
4. 动作（Actions）：
   - 推杆（putt）：使用推杆
   - 开球（driver）：使用开球杆
   - 任何在果岭上的推杆动作都会成功
5. 图示：
   - 图中显示了一个高尔夫球场的一部分，其中包括球洞（用旗子标记的绿色区域），沙坑（标记为sand的黄色区域）和其他地形
   - 每个区域的数字表示从该位置开始的价值。例如，果岭上球洞旁边的位置值为0，表示球已经进洞；距离球洞越远的位置，数值越低（负值越大），表示需要更多次击球才能进洞

Optimal Value Function

对于有限 MDP，策略可以部分排序：

$$\pi \geq \pi' \text{ iff } V^{\pi}(s) \geq V^{\pi'}(s),\ \forall s \in S$$

即，总有一个或多个策略是优于或等于所有其他策略的，这些就是最优策略，记作 π^*

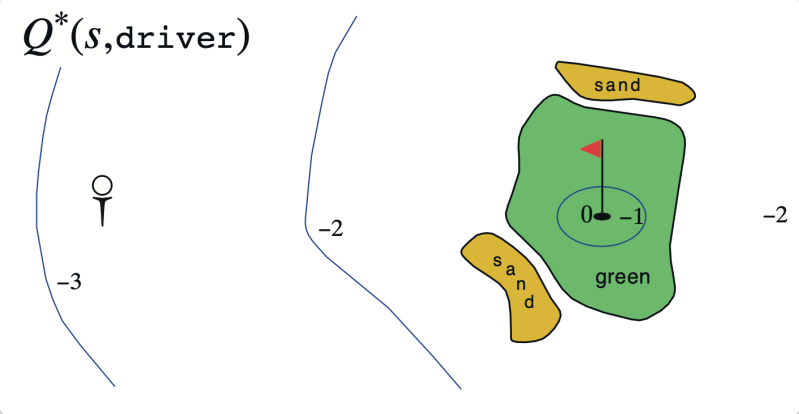
最优策略共享相同的最优状态值函数：

$$V^*(s) = \max_{\pi} V^{\pi}(s),\ \forall s \in S$$

最优策略也共享相同的最优动作值函数：

$$Q^*(s,a) = \max_{\pi} Q^{\pi}(s,a),\ \forall s \in S, a \in A$$

高尔夫的最优价值函数



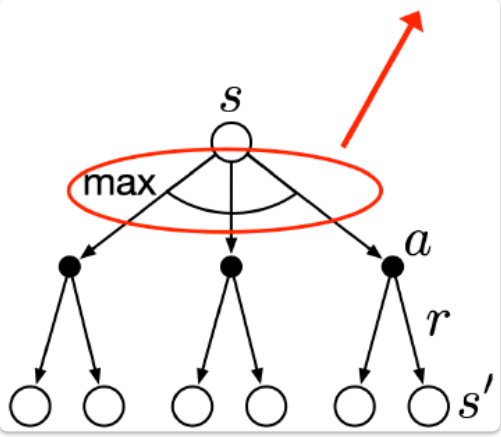
我们可以

- 使用开球杆（driver）击球：可以将球击得更远，但准确性较差
- 使用推杆（putter）击球：准确性高，但击球距离较短

$Q^*(s, \text{driver})$ 表示在状态 s 下，首先使用开球杆（driver）击球，然后使用最优策略采取后续动作的期望回报

Bellman Optimality Equation

$$
\begin{aligned}
V^*(s) &= \max_\pi V^\pi(s),\ \forall s \in S \\
&= \max_a \sum_{s'} P^a_{ss'} \left[R^a_{ss'} + \gamma V^*(s') \right]
\end{aligned}
$$

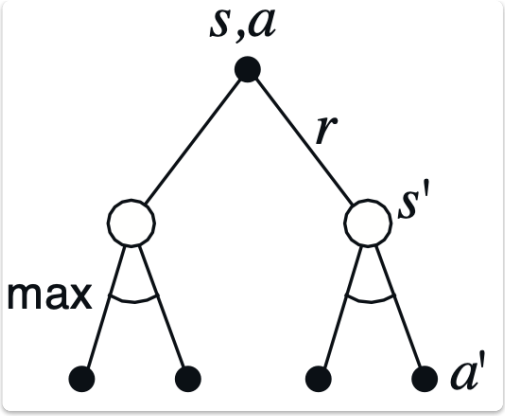


（箭头指的是 max 部分）

而对于 action-value Q^*：

$$
\begin{aligned}
Q^*(s,a) &= \mathbb{E}\left\{ r_{t+1} + \gamma \max_{a'} Q^*(s_{t+1}, a') \mid s_t = s, a_t = a \right\} \\
&= \sum_{s'} P^a_{ss'} \left[R^a_{ss'} + \gamma \max_{a'} Q^*(s', a') \right] \\
&= \max_\pi Q^\pi(s,a),\ \forall s \in S, a \in A
\end{aligned}
$$

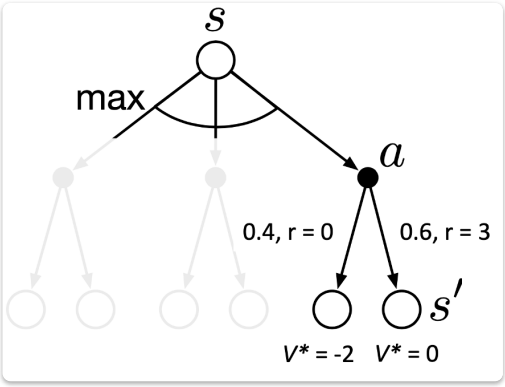
其中，Q^* 是这一系统非线性方程的唯一解



在状态 s 下采取动作 a 后，最优动作值函数等于

1. 即时奖励 r_{t+1} 加上从下一个状态 s_{t+1} 开始的最优动作值函数的折扣期望值
2. 所有可能后续状态 s' 的转移概率 $P^a_{ss'}$ 加权的即时奖励和折扣的最优后续动作值函数之和
3. 对于任何状态 s 和动作 a，最优动作值函数是所有策略中动作值函数的最大值

计算



$$V^*(s) = 0.6 \times (3 + 0) + 0.4 \times (0 - 2\gamma) = 1.8 - 0.8\gamma$$

Optimal State-Value Functions 的有效性

- 任何相对于 V^* 采取贪婪策略的策略都是最优策略：如果一个策略在每个状态下都选择使 V^* 最大的动作，那么这个策略就是最优策略
- 给定 V^*，一步搜索就能产生长期的最优动作：在已知最优状态值函数 V^* 的情况下，只需要进行一步前瞻搜索就可以找到最优的行动策略，而不需要进行多步复杂的计算

Action-Value Functions 的有效性

给定 Q^* 后，智能体甚至不需要进行一步前瞻搜索

$$a^* = \underset{a \in A}{\operatorname{argmax}}\, Q^*(s, a)$$