# 马尔可夫决策过程

本章开始介绍马尔可夫决策过程的基本概念，包括马尔可夫性质、回报、状态转移矩阵等内容。马尔可夫决策过程是强化学习的核心问题模型，即想用强化学习来解决问题，首先需要将问题建模为马尔可夫决策过程，并明确状态空间、动作空间、状态转移概率和奖励函数等要素。此外，还介绍了策略、状态价值和动作价值等重要概念，这些概念在后续的强化学习算法中会频繁用到，务必牢记。

## 马尔可夫决策过程

在强化学习中，马尔可夫决策过程（Markov Decision Process, MDP）是用来描述智能体与环境交互的数学模型。如图 1 所示，智能体（Agent）与环境（Environment）在一系列离散的时步（ time step ）中交互，在每个时步 $t$，智能体接收环境的状态 $s_t$，并根据该状态选择一个动作 $a_t$。执行该动作后，智能体会收到一个奖励 $r_t$，同时环境会转移到下一个状态 $s_{t+1}$。

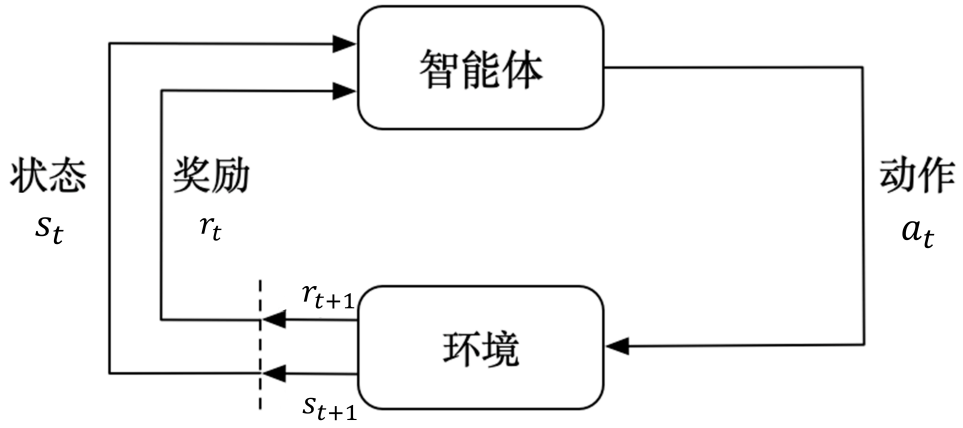


图 1: 智能体与环境的交互过程

这个过程不断重复，形成一条**轨迹**，如式 (1) 所示。

$$s_0, a_0, r_0, s_1, a_1, r_1, \cdots, s_t, a_t, r_t, \cdots \tag{1}$$

完成一条完整的轨迹，即从初始状态到终止状态（例如玩游戏玩到最终的胜负结算阶段），也称为一个**回合**（ episode ），通常在有限的时步 $T$ 后结束，即 $t = 0, 1, 2, \cdots, T$，$T$ 是回合的最大步数。

**如果要用强化学习来解决问题，首先需要将问题建模为马尔可夫决策过程，即明确状态空间、动作空间、状态转移概率和奖励函数等要素。**通常我们用一个五元组来定义马尔可夫决策过程，如式 (2) 所示。

$$MDP = (S, A, P, R, \gamma) \tag{2}$$

其中 $S$ 是状态空间，表示所有可能的环境状态的集合，$A$ 是动作空间，表示智能体可以选择的所有可能动作的集合，$P$ 是状态转移概率矩阵，描述了在给定当前状态和动作的情况下，环境转移到下一个状态的概率分布，$R$ 是奖励函数，定义了在特定状态下执行某个动作所获得的即时奖励，$\gamma$ 是折扣因子，用于权衡当前奖励和未来奖励的重要性，其取值范围在 0 到 1 之间。其中状态转移矩阵和折扣因子将在下文详细展开说明。

## 马尔可夫性质

马尔可夫决策过程的核心假设是**马尔可夫性质（Markov Property）**，即系统未来状态的概率分布只依赖于当前的状态和动作，而与过去的状态和动作无关，如式 (3) 所示。

$$P(s_{t+1}|s_t, a_t, s_{t-1}, a_{t-1}, \cdots, s_0, a_0) = P(s_{t+1}|s_t, a_t) \tag{3}$$

然而，在真实世界中严格满足马尔可夫性质的情况并不多见，但大多情况下，我们依然可以通过适当的状态表示来近似满足马尔可夫性质。

例如在自动驾驶中，将当前车辆的位置、速度和周围环境信息作为状态表示，这些信息足以预测下一时刻的状态，而不需要考虑更早之前的历史数据，从而近似满足马尔可夫性质，这样的过程也叫做**部分可观测马尔可夫决策过程（Partially Observable Markov Decision Process, POMDP）**。

# 状态转移矩阵

通常，马尔可夫决策过程通常指有限马尔可夫决策过程（finite MDP），即状态空间和动作空间都是有限的。如果状态空间或动作空间是无限的，通常需要采用其他方法进行建模，例如连续时间马尔可夫过程等。

既然状态数有限，就可以用一种状态流向图的形式表示智能体与环境交互过程中的走向。如图 2 所示，图中每个曲线箭头表示指向自己，对于状态 $s_1$ 来说，有 $0.2$ 的概率继续保持在 $s_1$ 状态，同时也有 $0.4$ 和 $0.4$ 的概率转移到状态 $s_2$ 和 $s_3$。同理，其他状态之间也有类似的转移概率。

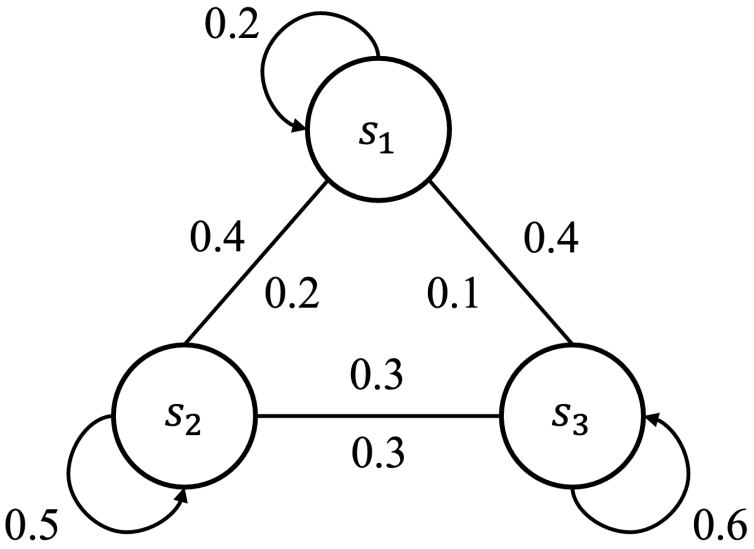


图 2: 马尔可夫链

注意，图 2 中并没有包含动作和奖励等元素，因此严格来说它表示的是**马尔可夫链（Markov Chain）**，又叫做离散时间的马尔可夫过程（Markov Process），但它与马尔可夫决策过程有着密切的联系，都是基于马尔可夫性质构建的。

我们用一个概率来表示状态之间的切换，如式 (4) 所示。

$$P_{ss'} = P(S_{t+1} = s' | S_t = s) \tag{4}$$

即当前状态是 $s$ 时，下一个状态是 $s'$的概率，其中大写的 $S$ 表示所有状态的集合，即 $S = \{s_1, s_2, s_3\}$。例如，$P_{12} = P(S_{t+1} = s_2 | S_t = s_1) = 0.4$ 表示当前时步的状态是 $s_1$，下一个时步切换到 $s_2$ 的概率为 $0.4$。

拓展到所有状态，可以把这些概率绘制成一个状态转移表，如表 1 所示。

**表 1: 马尔可夫状态转移表**

| | | | |
|---|---|---|---|
| $S_t = s_1$ | 0.2 | 0.4 | 0.4 |
| $S_t = s_2$ | 0.2 | 0.5 | 0.6 |

| | $S_{t+1}=s_1$ | $S_{t+1}=s_1$ | $S_{t+1}=s_3$ |
|---|---|---|---|
| $S_t=s_3$ | 0.1 | 0.3 | 0.6 |

在数学上也可以用矩阵来表示，如式 (5) 所示。

$$P_{ss'}=\begin{bmatrix}0.2 & 0.4 & 0.4\\ 0.2 & 0.5 & 0.3\\ 0.1 & 0.3 & 0.6\end{bmatrix} \tag{5}$$

这个矩阵就叫做**状态转移矩阵（State Transition Matrix）**，拓展到所有状态可表示为式 (6) 所示。

$$P_{ss'}=\begin{pmatrix}p_{11} & p_{12} & \cdots & p_{1n}\\ p_{21} & p_{22} & \cdots & p_{2n}\\ \vdots & \vdots & \ddots & \vdots\end{pmatrix} \tag{6}$$

其中 $n$ 表示状态数，注意从同一个状态出发转移到其他状态的概率之和是等于 1 的，即 $\sum_{j=1}^{n}p_{ij}=1$，$i=1,2,\cdots,n$。**状态转移矩阵是环境的一部分**，描述了环境状态之间的转移关系。

# 目标与回报

在强化学习中，智能体的目标是**通过与环境的交互，学习一个最优策略，使得在每个状态下选择的动作能够最大化累积的奖励**。这个累积的奖励通常被称为**回报（Return）**，如式 (7) 所示。

$$G_t=R_{t+1}+\gamma R_{t+2}+\gamma^2R_{t+3}+\cdots=\sum_{k=0}^{\infty}\gamma^kR_{t+k+1} \tag{7}$$

表示从时间步 $t$ 开始，未来所有奖励的加权和，其中 $\gamma$ 是折扣因子，在 0 到 1 之间。折扣因子的作用是用来控制未来奖励在当前决策中的重要性。当 $\gamma$ 接近 0 时，智能体更关注当前的奖励，而忽略未来的奖励；当 $\gamma$ 接近 1 时，智能体会更加重视未来的奖励。

折扣因子一方面在数学上确保回报 $G_t$ 的收敛性，另一方面也代表了时间价值，就像经济学中的货币折现，同样面值的货币现在拿到手中比未来拿到更有价值。此外，折扣因子还可以用来衡量智能体对长期回报的关注度，或者者说"能看到多远"，称之为有效视界（effective horizon），如式 (8) 所示。

$$H_{eff}=\frac{1}{1-\gamma} \tag{8}$$

当 $\gamma=0.9$ 时，$H_{eff}=10$，表示智能体主要关注未来 10 个时步内的奖励；当 $\gamma=0.99$ 时，$H_{eff}=100$，表示智能体关注未来 100 个时步内的奖励。

此外，当前时步的回报 $G_t$ 跟下一个时步 $G_{t+1}$ 的回报是有所关联的，即递归地定义，如式 (9) 所示。

$$\begin{aligned}G_t&\doteq r_{t+1}+\gamma r_{t+2}+\gamma^2r_{t+3}+\gamma^3r_{t+4}+\cdots\\&=r_{t+1}+\gamma\left(r_{t+2}+\gamma r_{t+3}+\gamma^2r_{t+4}+\cdots\right)\\&=r_{t+1}+\gamma G_{t+1}\end{aligned} \tag{9}$$

# 策略与价值

## 策略

策略 （Policy）表示智能体在每个状态下选择动作的规则或方法，用 $\pi$ 表示。如式 (10) 所示，策略是一个从状态到动作的映射或函数。

$$\pi(a|s) = P(A_t = a|S_t = s) \tag{10}$$

表示在状态 $s$ 下选择动作 $a$ 的概率分布。**策略可以是确定性的（deterministic），即在每个状态下总是选择同一个动作，或者是随机性的（stochastic），即在每个状态下根据一定的概率分布选择动作。**

## 状态价值

状态价值函数（State-Value Function）表示在给定状态下，按照某种策略 $\pi$ 进行决策所能获得的回报期望值，用 $V_\pi(s)$ 表示，如式 (11) 所示。

$$\begin{aligned} V_\pi(s) &= \mathbb{E}_\pi[R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \cdots | S_t = s] \\ &= \mathbb{E}_\pi[G_t | S_t = s] \end{aligned} \tag{11}$$

举例来说，假设智能体处于一个 $3 \times 3$ 的网格世界中，目标是走到右下角，对应获得 10 分的奖励，走到其他格子的奖励为 0，策略是随机选择一个方向（上、下、左、右）移动一步。

如果初始状态 $s_0$ 即起点在左上角，按照当前策略，平均可能需要 10 才能到达终点，每步奖励为 0，到达终点时获得 10 分奖励，设折扣因子 $\gamma = 0.9$，则从起点状态出发的状态价值计算如式 (12) 所示。

$$V_\pi(s_0) = 0 + 0.9^1 \times 0 + 0.9^2 \times 0 + \cdots + 0.9^{10} \times 10 \approx 3.49 \tag{12}$$

## 动作价值

动作价值函数（Action-Value Function）表示在给定状态 $s$ 和动作 $a$ 下，按照某种策略 $\pi$ 进行决策所能获得的回报期望值，用 $Q_\pi(s, a)$ 表示，如式 (13) 所示。

$$Q_\pi(s, a) = \mathbb{E}_\pi \left[ G_t \mid s_t = s, a_t = a \right] \tag{13}$$

## 状态价值与动作价值的关系

状态价值函数和动作价值函数之间存在密切的关系，如式 (14) 所示。

$$V_\pi(s) = \sum_{a \in A} \pi(a \mid s) Q_\pi(s, a) \tag{14}$$

换句话说，状态价值是对所有可能的动作价值的加权平均，而动作价值则是对特定动作的评价。

状态价值反映了策略本身的好坏，它不关心智能体在状态 $s$ 下选择了哪个具体动作，而是关注在该状态下按照策略 $\pi$ 进行决策所能获得的整体回报期望值。动作价值则更具体地反映了在特定状态下选择某个动作所能获得的回报期望值，即不仅考虑智能体所处的状态，还考虑了智能体在该状态下选择的具体动作。

# 有模型与无模型

在谈到有模型（Model-Based）与无模型（Model-Free）方法时，这里的模型通常指与智能体交互的环境模型，即对环境的状态转移概率和奖励函数进行建模。根据是否使用环境模型，强化学习方法可以分为有模型方法和无模型方法两大类。

有模型的方法利用环境模型来进行规划和决策，通常包括动态规划（Dynamic Programming）等方法。这些方法通过对环境的状态转移概率和奖励函数进行建模，能够在不与环境直接交互的情况下，预测未来的状态和奖励，从而制定最优策略。

无模型的方法则不依赖于环境模型，包括 Q-Learning 、SARSA 等算法，这些方法通过与环境的直接交互，学习状态价值函数或动作价值函数，从而逐步改进策略。无模型方法通常更适用于复杂或未知的环境，因为它们不需要对环境进行显式建模，在强化学习中应用更为广泛。

# 预测与控制

预测（Prediction）问题是指在给定策略 $\pi$ 的情况下，评估该策略的好坏，即计算状态价值函数 $V_\pi(s)$ 或动作价值函数 $Q_\pi(s,a)$。预测问题的目标是了解在当前策略下，智能体在不同状态下能够获得的回报期望值。用于预测任务的算法主要包括蒙特卡洛方法（Monte Carlo）、时序差分学习（Temporal Difference, TD）等，后续会详细介绍。

控制（Control）问题是指在给定环境模型的情况下，寻找最优策略 $\pi^*$，使得在每个状态下选择的动作能够最大化累积的回报。控制问题的目标是通过与环境的交互，学习一个最优策略，使得智能体能够在不同状态下获得最大的回报。用于控制任务的算法主要包括动态规划（Dynamic Programming）、Q学习（Q-Learning）、策略梯度方法（Policy Gradient）等，后续也会详细介绍。

复杂问题中通常需要同时解决预测和控制问题，即在学习最优策略的过程中，同时评估当前策略的好坏，从而不断改进策略，最终收敛到最优策略。

# 思考

**强化学习所解决的问题一定要严格满足马尔可夫性质吗？请举例说明。**

不一定。例如在围棋游戏场景中，不仅需要考虑当前棋子的位置，还需要考虑棋子的历史位置，因此不满足马尔可夫性质。但依然可以使用强化学习的方法进行求解，例如在 AlphaGO 论文中使用了蒙特卡洛树搜索算法来解决这个问题。在一些时序性场景中，也可以通过引入记忆单元来解决这个问题，例如在 DQN 算法中，使用了记忆单元来存储历史状态，从而解决了这个问题，尽管它也不满足马尔可夫性质。

**马尔可夫决策过程主要包含哪些要素?**

马尔可夫决策 $< S, A, R, P, \gamma >$ 主要包含状态空间 $S$、动作空间 $A$、奖励函数 $R$、状态转移矩阵 $P$、折扣因子 $\gamma$ 等要素，其中状态转移矩阵 $P$ 是环境的一部分，而其他要素是智能体的一部分。在实际应用中，通常还考虑值函数 $V$ 和策略函数 $\pi$ 等要素，值函数用于某个状态下的长期累积奖励，策略函数用于某个状态下的动作选择。

**马尔可夫决策过程引入折扣因子 $\gamma$ 的作用是什么？如何理解折扣因子的意义?**

折扣因子 $\gamma$ 的作用是用来控制未来奖励在当前决策中的重要性。它的取值范围在 $0$ 到 $1$ 之间。当 $\gamma$ 接近 $0$ 时，智能体更关注当前的奖励，而忽略未来的奖励；当 $\gamma$ 接近 $1$ 时，智能体会更加重视未来的奖励。一方面在数学上能够确保回报 $G_t$ 的收敛性，另一方面也代表了时间价值，就像经济学中的货币折现，同样面值的货币现在拿到手中比未来拿到更有价值。

**马尔可夫决策过程与金融科学中的马尔可夫链有什么区别与联系?**

马尔可夫链是一个随机过程，其下一个状态只依赖于当前状态而不受历史状态的影响，即满足马尔可夫性质。马尔可夫链由状态空间、初始状态分布和状态转移概率矩阵组成。马尔可夫决策过程是一种基于马尔可夫链的决策模型，它包含了状态、行动、转移概率、奖励、值函数和策略等要素。马尔可夫决策过程中的状态和状态转移概率满足马尔可夫性质，但区别在于它还包括了行动、奖励、值函数和策略等要素，用于描述在给定状态下代理如何选择行动以获得最大的长期奖励。

**有模型与免模型算法的区别？举一些相关的算法?**

有模型算法在学习过程中使用环境模型，即环境的转移函数和奖励函数，来推断出最优策略。这种算法会先学习环境模型，然后使用模型来生成策略。因此，有模型算法需要对环境进行建模，需要先了解环境的转移函数和奖励函数，例如动态规划等算法。免模型算法不需要环境模型，而是直接通过试错来学习最优策略。这种算法会通过与环境的交互来学习策略，不需要先了解环境的转移函数和奖励函数。免模型算法可以直接从经验中学习，因此更加灵活，例如Q-learning 、Sarsa 等算法。

**举例说明预测与控制的区别与联系。**

**区别**：预测任务主要是关注如何预测当前状态或动作的价值或概率分布等信息，而不涉及选择动作的问题；控制任务则是在预测的基础上，通过选择合适的动作来最大化累计奖励，即学习一个最优的策略。**联系**：预测任务是控制任务的基础，因为在控制任务中需要对当前状态或动作进行预测才能选择最优的动作；控制任务中的策略通常是根据预测任务中获得的状态或动作价值函数来得到的，因此预测任务对于学习最优策略是至关重要的。以赌博机问题为例，预测任务是估计每个赌博机的期望奖励（即价值函数），控制任务是选择最优的赌博机来最大化累计奖励。在预测任务中，我们可以使用多种算法来估计每个赌博机的期望奖励，如蒙特卡罗方法、时间差分方法等。在控制任务中，我们可以使用贪心策略或ε-贪心策略来选择赌博机，这些策略通常是根据预测任务中得到的每个赌博机的价值函数来确定的。因此，预测任务对于控制任务的实现至关重要。