

马尔可夫决策过程

强化学习的目标是给定一个马尔可夫决策过程，寻找最优策略

马尔可夫过程

马尔可夫过程是一个二元组 $\langle S, P \rangle$ 。S是有限状态集，P是状态转移矩阵。马尔可夫过程是一个无记忆的随机过程，每个状态序列 S_1, S_2 具有**马尔可夫性质**，即未来只与现在有关而与过去无关。

状态转移矩阵

状态转移矩阵 (State Transition Matrix)

P 定义了从所有**状态** s 转移到所有**后继状态** s' 的**概率**

	S_1	S_2	S_3
S_1	0.2	0.3	0.4
S_2	0.5	0.1	0.4
S_3	0.3	0.6	0.2

$$P_{ss'} = P[S_{t+1} = s' | S_t = s]$$

$$P = \begin{bmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & \ddots & \vdots \\ P_{n1} & \cdots & P_{nn} \end{bmatrix}$$



马尔可夫奖励过程

由四元组 $\langle S, P, R, \gamma \rangle$ 组成

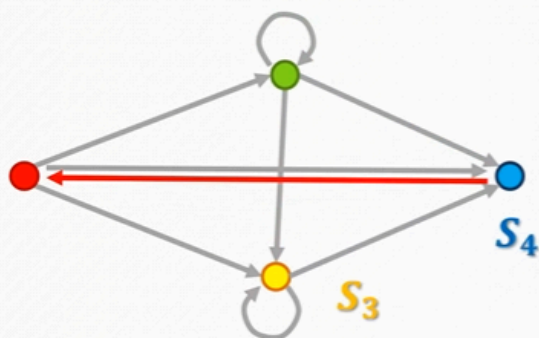
马尔可夫奖励过程是一个**四元组** $\langle S, P, R, \gamma \rangle$

- S : (有限) 状态集
- P : 状态转移概率矩阵 $P_{ss'} = P[S_{t+1} = s' | S_t = s]$
- R : 奖励函数 $R_S = E[R_{t+1} | S_t = s]$
- γ : 折扣因子/衰减系数 $\gamma \in [0, 1]$



设当前状态为 $S_t = s$,由 s 到 S_{t+1} 集合的奖励值集合为 R_{t+1} , R_S 就是集合 R_{t+1} 的期望。 R_S 是一个期望，但 R_{t+1} 不是，他只是代表到达 $t+1$ 时刻的某一个状态点的奖励。

- R : 奖励函数 $R_S = E[R_{t+1} | S_t = s]$



一定分布

$$R_{S_1} = E[R_{t+1} | S_t = S_1]$$

$$R_{S_2} = E[R_{t+1} | S_t = S_2]$$

$$R_{S_3} = E[R_{t+1} | S_t = S_3]$$

$$R_{S_4} = E[R_{t+1} | S_t = S_4]$$

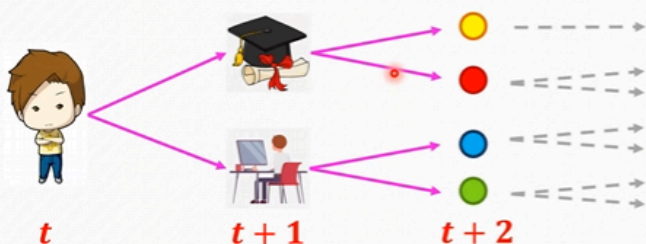
折扣因子 γ

折扣因子 γ 是用来计算当前状态的总回报的，表示离的越远的时刻的价值越少。注意，这里的 G_t 只是某一条路线的折扣奖励，并不是当前状态的最终价值。

- γ : 折扣因子/衰减系数 $\gamma \in [0, 1]$

回报 (Return): G_t 是从时间 t 开始的**总折扣奖励**

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$



表示所有**奖励在当前的价值**

- 未来是不确定的
- 未来很长

6

值函数 $V(S)$

$V(S)$ 表示状态 S 的长期价值，即综合了未来所有路线的折扣奖励，算出期望。

$$V(s) = E[G_t | S_t = s]$$

贝尔曼方程

对于 G_t 可以采用递归的方式表示，即 $G_t = R_{t+1} + \gamma G_{t+1}$ 。因为原式是在 s 状态的，而要求 $E[G(S_{t+1})]$ 要在 s' 状态下，所以多了个转移矩阵。

$$V(S) = E[R_{t+1} + \gamma G(s_{t+1}) | S_t = s]$$

$$V(S) = R_s + \gamma \sum_{s' \in S} P_{ss'} V(s')$$

矩阵形式

这里需要解释一下为什么两边看成一个V?

- 首先所有的状态本质上都是对自己进行迭代更新，因为状态集是不变的。在标量形式下分开写是为了表达递归关系。
- 贝尔曼方程想表达一个平衡状态下的值函数，即当V达到平衡时（收敛），左右两侧一样

贝尔曼方程矩阵形式

$$V(s) = R_s + \gamma \sum_{s' \in S} P_{ss'} V(s')$$
$$\begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix}_{n \times 1} = \begin{bmatrix} R_1 \\ \vdots \\ R_n \end{bmatrix}_{n \times 1} + \gamma \begin{bmatrix} P_{11} & \dots & P_{1n} \\ \vdots & & \\ P_{n1} & \dots & P_{nn} \end{bmatrix}_{n \times n} \begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix}_{n \times 1}$$

$$\underline{V} = R + \gamma P V$$

$$\underline{(1 - \gamma P)} V = R$$

$$\underline{V} = (1 - \gamma P)^{-1} R$$



10

马尔可夫决策过程

由一个五元组 $\langle S, A, P, R, \gamma \rangle$ 表示，其中A是有限动作集。

- **S**: (有限) 状态集
- **A**: (有限) 动作集
- **P**: 状态转移概率矩阵 $P_{ss'}^a = P[S_{t+1} = s' | S_t = s, A_t = a]$
- **R**: 奖励函数 $R_s^a = E[R_{t+1} | S_t = s, A_t = a]$
- **γ** : 折扣因子/衰减系数 $\gamma \in [0, 1]$



11

策略

π 是一个随机变量，他的分布代表所有动作。

π 是给定状态的**动作分布** $\pi(a|s) = P[A_t = a | S_t = s]$ 随机变量

- 策略完全决定智能体行为
- MDP策略依赖于当前状态(无关历史)
- 策略是固定的(无关时间) $A_t \sim \pi(\cdot | S_t)$, 任意 $t > 0$



当前S下采用不同a的可能性 * 采用每个a后到不同S'的可能性 = 在策略 π 下从S到S'的可能性

给定一个马尔可夫**决策过程** $M = \langle S, A, P, R, \gamma \rangle$ 和**策略** π
 其可转化为马尔可夫**过程**和马尔可夫**奖励过程**
 $\langle S, P \rangle$ $\langle S, P, R, \gamma \rangle$

$$P_{s,s'}^{\pi} = \sum_{a \in A} \pi(a|s) P_{s,s'}^a$$

$$R_s^{\pi} = \sum_{a \in A} \pi(a|s) R_s^a$$

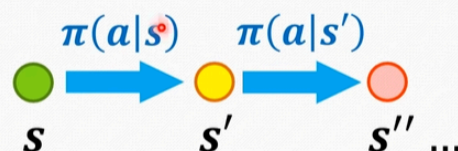


状态值函数

相比马尔可夫奖励过程的价值函数，马尔可夫决策过程是智能体自身做出了对应动作后从而引起状态的变化，而前两者则是随波逐流。

状态值函数 (State-value function)

$$\underline{v_{\pi}(s) = E_{\pi}[G_t | S_t = s]}$$

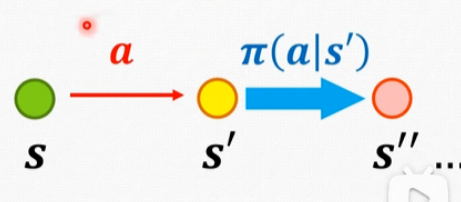


动作值函数

在状态 S 下，先采取策略 a 后回报的期望。注意，在 S 下采取状态 a 并不能到达某个确定的状态，还是具有状态转移矩阵 $P_{s,s'}^a$ 。其实就是在执行该动作后的状态值函数。

动作值函数 (Action-value function)

$$\underline{q_{\pi}(s, \underline{a}) = E_{\pi}[G_t | S_t = s, \underline{A_t = a}]}$$

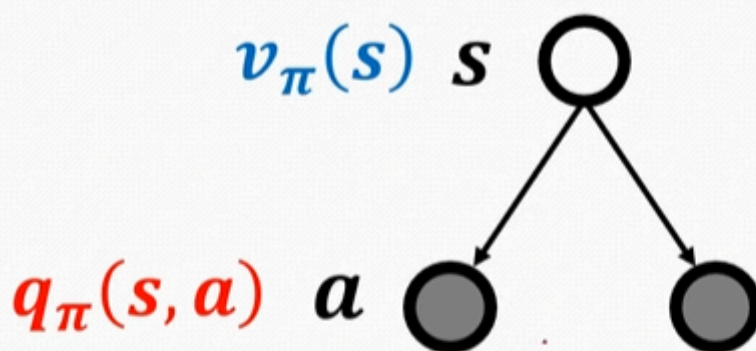


贝尔曼方程

$$\begin{aligned} V(s) &= \sum_{a \in A} \pi(a|s) [R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a V(s')] \\ &= \sum_{a \in A} \pi(a|s) q(s, a) \\ q(s, a) &= R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \sum_{a' \in A} \pi(a'|s) q(s', a') \\ &= R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v(s') \end{aligned}$$

状态值函数和动作值函数的关系

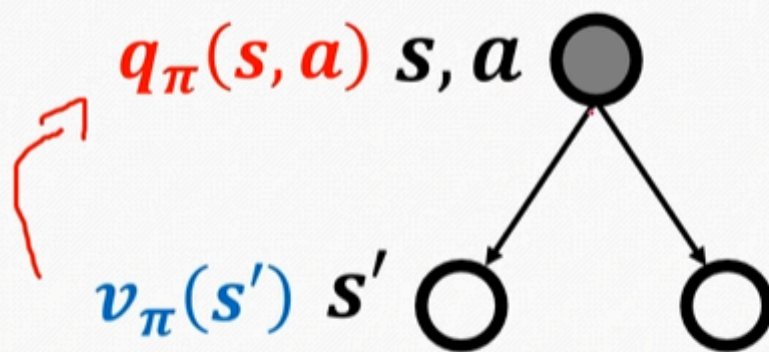
根据定义式可以推出：



$$v_{\pi}(s) = \sum_{a \in A} \pi(a|s) \underline{\underline{q_{\pi}(s, a)}}$$

某一个状态的价值可以用该
状态下**所有动作的价值**表述

根据贝尔曼的递归式子可以推出：



$$q_{\pi}(s, a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_{\pi}(s')$$

某一个动作的价值可以用该
状态后继状态的价值表述



17

最优策略

最优状态值函数 (Optimal state-value function)

$$v_*(s) = \max_{\pi} v_{\pi}(s)$$

最优动作值函数 (Optimal action-value function)

$$q_*(s, a) = \max_{\pi} q_{\pi}(s, a)$$

最优策略 (Optimal policy) • 存在一个最优策略, 使 $\pi_* \geq any \pi$

$\pi_* \geq any \pi$ • 所有最优策略都能取得最优**状态**值函数

注: 若 $v_{\pi'}(s) \geq v_{\pi}(s)$, 则 $\pi' > \pi$ • 所有最优策略都能取得最优**动作**值函数