组合优化理论

第10章 强化学习

主讲教师：陈安龙

第10章 强化学习

1.强化学习概念与模型

2.马尔科夫策略过程

3.动态规划寻优策略

一、简介

机器学习是人工智能的一个分支，在近30多年已发展为一门多领域交叉学科，涉及概率论、统计学、逼近论、凸分析、计算复杂性理论等的学科。

强化（reinforcement）学习是机器学习的一个子领域，其灵感来源于心理学中的行为主义理论，即**智能体如何在环境给予的奖励或惩罚**的刺激下，逐步形成对刺激的预期，产生能获得最大利益的习惯性行为。它强调如何基于环境而行动，以使系统行为从环境中获得的**累积奖励值最大**。

由于它具有普适性而被很多领域进行研究，例如：自动驾驶,博弈论、控制论、运筹学、信息论、仿真优化、多主体系统学习、群体智能、统计学以及遗传算法。

强化学习的基本概念

策略（policy）：定义了agents在特定的时间特定的环境下的行为方式，可以视为是从环境状态到行为的映射，常用 π 来表示。策略可以分为两类：

（1）确定性的policy（Deterministic policy）：$a=\pi(s)$ $a=\pi(s)$

（2）随机性的policy（Stochastic policy）：$\pi(a|s)=P[A_t=a|S_t=s]$

其中，t 是时间点，t=0,1,2,3,……

$S_t \in S$，S 是环境状态的集合，S_t代表时刻t 的状态，s 代表其中某个特定的状态；$A_t \in A(S_t)$，$A(S_t)$是在状态S_t下的actions的集合，A_t代表时刻t的行为，a代表其中某个特定的行为。

强化学习的基本概念

奖励信号（a reward signal）：是一个标量值，是每个时间步中环境根据agent的行为返回给agent的信号，reward定义了在该情景下执行该行为的好坏，agent可以根据reward来调整自己的policy，常用R来表示。

值函数（value function）：Reward定义的是立即的收益，而value function定义的是长期的收益，它可以看作是累计的reward，常用*v*来表示。

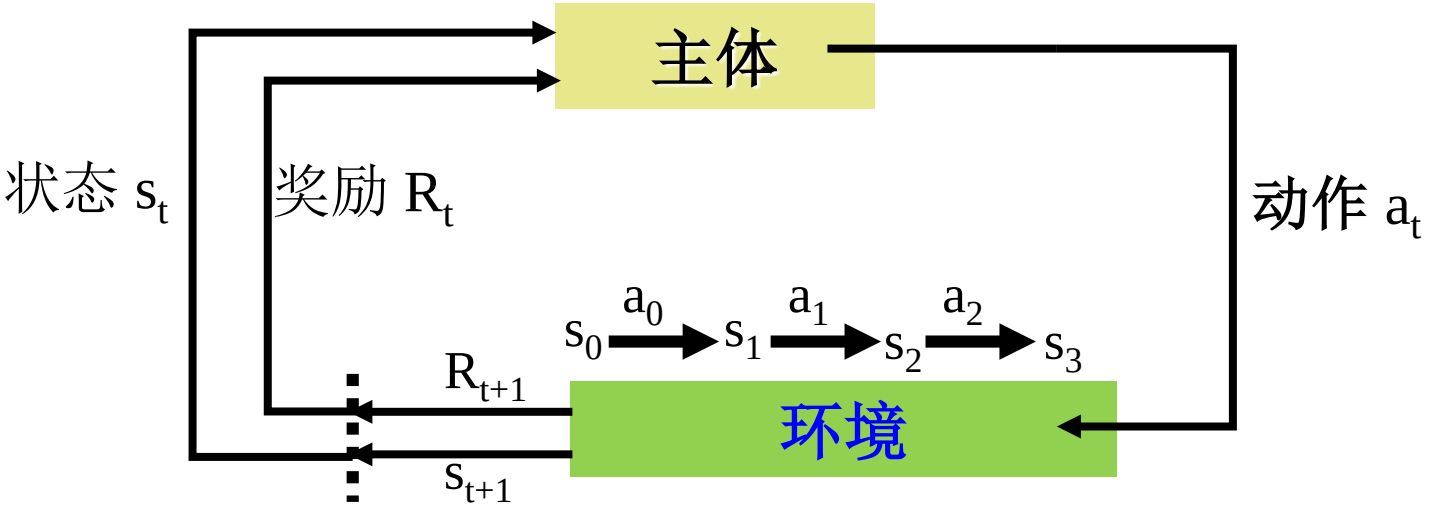
探索（exploration）：智能体agent选择之前未执行过的actions，探索新的动作选择或状态空间新的部分。

利用（exploitation）：智能体agent选择已执行过的actions，利用目前得到的知识，从而对已知的actions的模型进行完善。

强化学习模型

主要包含四个元素，agent，环境状态，行动，奖励；

强化学习的目标：是获得最多的累计奖励



其中，t 是时间点，t=0,1,2,3,……$S_t \in S$ S是环境状态的集合；

$A_t \in A(S_t)$，$A(S_t)$是在状态S_t下的actions的集合；

$R_{t+1} \in R$，在t时刻agent采取一个动作后都会收到一个回报值

$R_{t+1} \in R$，然后接收一个新状态S_{t+1}。

强化学习模型（续）

长期回报可以用每个时刻的立即回报来表示：

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} \cdots = \sum_{k=t+1}^{\infty} R_k$$

一般会用下面更通用的公式来代替：

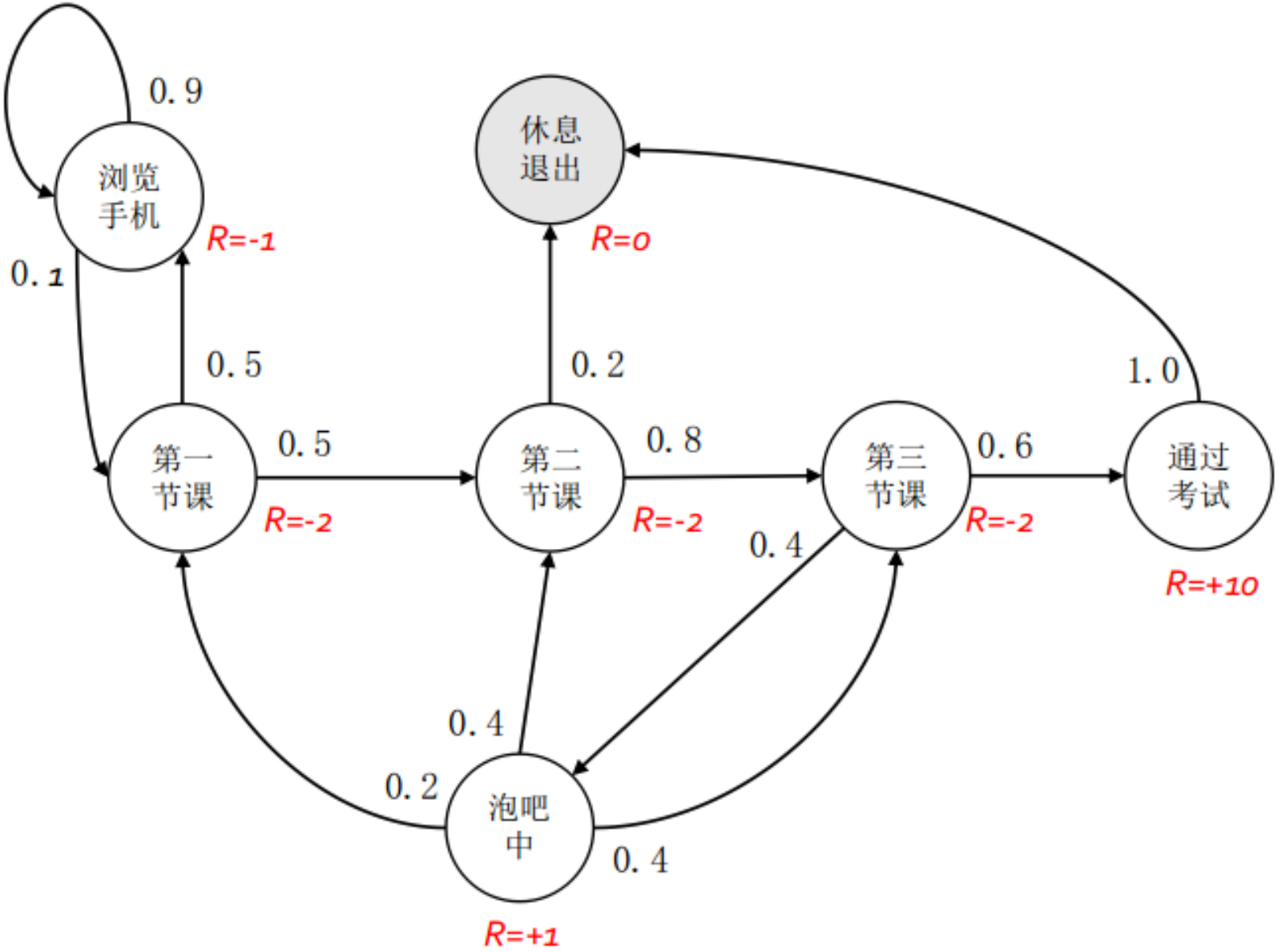
$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} \cdots + \gamma^{T-t-1} R_T = \sum_{k=0}^{T-t-1} \gamma^k R_{t+k+1}$$

其中$\gamma \in [0,1]$称为回报折扣因子，表明了未来的回报相对于当前回报的重要程度。

$\gamma=0$时，相当于只考虑立即回报不考虑长期回报；

$\gamma=1$时，将长期回报和立即回报看得同等重要。

$T \in [1,\infty]$表示完成一次实验过程的总步数，$T=\infty$和$\gamma=1$不能同时满足，否则长期回报将无法收敛。



下面给出了学生马尔科夫过程中四个状态序列的开始状态“第一节课”的收获值的计算，选取 S1 =“第一节课”，$\gamma = 0.5$。

- C1 C2 C3 Pass Sleep

$$G1 = -2 + (-2) * 1/2 + (-2) * 1/4 + 10 * 1/8 + 0 * 1/16 = -2.25$$

- C1 FB FB C1 C2 Sleep

$$G1 = -2 + (-1) * 1/2 + (-1) * 1/4 + (-2) * 1/8 + (-2) * 1/16 + 0 * 1/32 = -3.125$$

- C1 C2 C3 Pub C2 C3 Pass Sleep

$$G1 = -2 + (-2) * 1/2 + (-2) * 1/4 + 1 * 1/8 + (-2) * 1/16 + ... = -3.41$$

- C1 FB FB C1 C2 C3 Pub C1 FB FB FB C1 C2 C3 Pub C2 Sleep

$$G1 = -2 + (-1) * 1/2 + (-1) * 1/4 + (-2) * 1/8 + (-2) * 1/16 + (-2) * 1/32 + ... = -3.20$$

马尔科夫奖励过程

在一个时序过程中，如果 t + 1 时刻的状态仅取决于 t 时刻的状态 S_t 而与 t 时刻之前的任何状态都无关时，则认为 t 时刻的状态 S_t 具有马尔科夫性 (Markov property)

具备了马尔科夫性的随机过程称为马尔科夫过程 (Markov process)

马尔科夫过程的核心是状态转移概率矩阵：

$$P_{ss^*} = P(S_{t+1} = s^* \mid S_t = s)$$

从任意一个状态 s 到其所有后继状态 s* 的状态转概率

$$P = form \begin{bmatrix} p_{11} & \cdots & p_{1n} \\ \vdots & \cdots & \vdots \\ p_{n1} & \cdots & p_{nn} \end{bmatrix} \overset{to}{}$$

马尔科夫奖励过程（Markov reward process,MRP）是由 ⟨S, P, R, γ⟩ 构成的一个元组，其中：

S 是一个有限状态集

P 是集合中状态转移概率矩阵：$P_{ss'} = P[S_{t+1} = s' | S_t = s]$

R 是一个奖励函数:$R_s = E[R_{t+1} | S_t = s]$

γ 是一个衰减因子:$\gamma \in [0, 1]$

价值函数建立了从状态到价值的映射，是马尔科夫奖励过程中状态收获的期望。

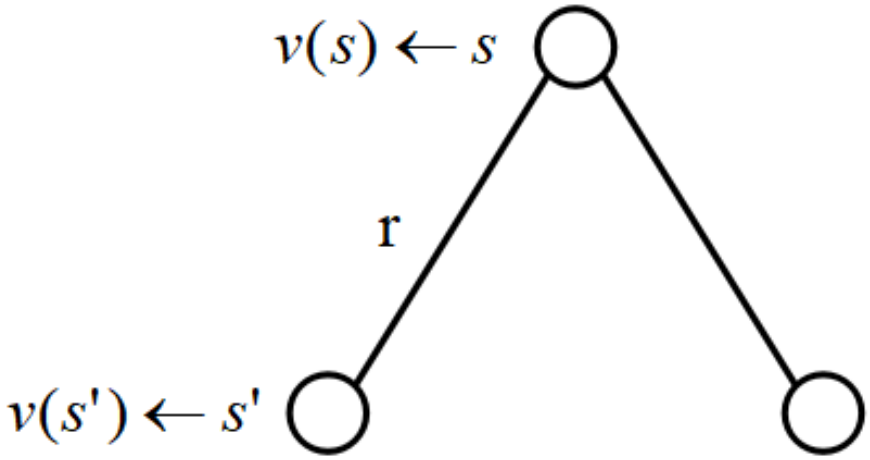
$$v(s) = E(G_t \mid S_t = s) = E(R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} \cdots \mid S_t = s)$$

$$= E(R_{t+1} + \gamma(R_{t+2} + \gamma^1 R_{t+3} \cdots) \mid S_t = s)$$

$$= E(R_{t+1} + \gamma G_{t+1} \mid S_t = s)$$

$$= E(R_{t+1} + \gamma v(S_{t+1}) \mid S_t = s)$$

根据马尔科夫奖励过程的定义，R_{t+1} 的期望就是其自身，因为每次离开同一个状态得到的奖励都是一个固定的值。而下一时刻状态价值的期望，可以根据下一时刻状态的概率分布得到。如果用 s′ 表示 s 状态下一时刻任一可能的状态：



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

上式称为马尔科夫奖励过程中的贝尔曼方程（Bellman equation）

上述Bellman 方程可以写成如下矩阵的形式：

$$v = R + \gamma P v$$

$$\begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix} = \begin{bmatrix} R_1 \\ \vdots \\ R_n \end{bmatrix} + \gamma \begin{bmatrix} p_{11} & \cdots & p_{1n} \\ \vdots & \cdots & \vdots \\ p_{n1} & \cdots & p_{nn} \end{bmatrix} \times \begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix}$$

理论上，该方程可以直接求解：

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

计算这类问题的时间复杂度是 $O(n^3)$，其中 n 是状态的数量。

马尔科夫决策过程

马尔科夫决策过程（Markov decision process, MDP）是由

⟨S, A, P, R, γ⟩ 构成的一个元组，其中：

S 是一个有限状态集

A 是一个有限行为集

P 是集合中基于行为的状态转移概率矩阵：

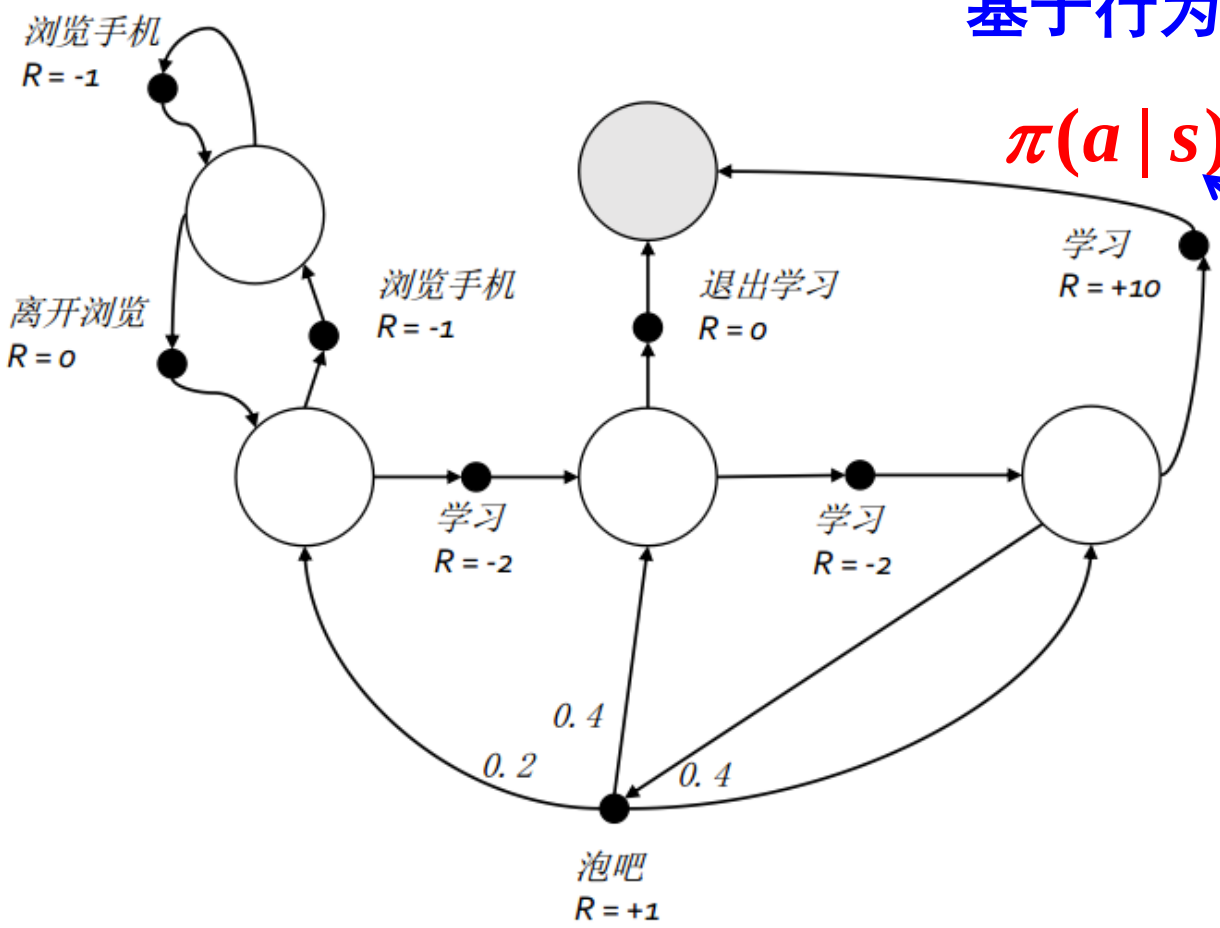
$$P_{ss'}^{a} = P\left[S_{t+1} = s' \mid S_t = s, A_t = a\right]$$

R 是基于状态和行为的奖励函数：

$$R_{ss'}^{a} = E\left[R_{t+1} \mid S_t = s, A_t = a\right]$$

γ 是一个衰减因子：$\gamma \in [0, 1]$

下面给出了学生马尔科夫决策过程的状态转化图。图中依然用空心圆圈表示状态，增加一类黑色实心圆圈表示个体的行为。根据马尔科夫决策过程的定义，奖励和状态转移概率均与行为直接相关，同一个状态下采取不同的行为得到的奖励是不一样的。



基于行为集合的一个概率分布：

$$\pi(a \mid s) = P\left[A_t = a \mid S_t = s\right]$$

好汉不提当年勇

当给定一个马尔科夫决策过程：$M = \langle S, A, P, R, \gamma \rangle$ 和一个策略 π，那么状态序列 $S_1, S_2, \ldots$ 是一个符合马尔科夫过程 $\langle S, P_\pi \rangle$ 的采样。类似的，联合状态和奖励的序列 $S_1, R_2, S_2, R_3, \ldots$ 是一个符合马尔科夫奖励过程 $\langle S, P_\pi, R_\pi, \gamma \rangle$ 的采样，并且在这个奖励过程中满足下面两个方程：

$$P^{\pi}_{s,s'} = \sum_{a \in A} \pi(a \mid s) P^{a}_{ss'}$$

$$R^{\pi}_{s} = \sum_{a \in A} \pi(a \mid s) R^{a}_{s}$$

上述公式体现了马尔科夫决策过程中一个策略对应了一个马尔科夫过程和一个马尔科夫奖励过程。同一个马尔科夫决策过程，不同的策略会产生不同的马尔科夫（奖励）过程，进而会有不同的状态价值函数。因此在马尔科夫决策过程中，有必要扩展先前定义的价值函数。

价值函数 $v_\pi(s)$ 是在马尔科夫决策过程下基于策略 π 的状态价值函数，表示从状态 s开始，遵循当前策略 π 时所获得的收获的期望：

$$v_\pi(s) = E\left[G_t \mid S_t = s\right]$$

由于引入了行为，为了描述同一状态下采取不同行为的价值，定义一个基于策略π 的行为价值函数 $q_\pi(s, a)$，表示在遵循策略 π 时，对当前状态 s 执行某一具体行为 a 所能的到的收获的期望：

$$q_\pi(s,a) = E\left[G_t \mid S_t = s, A_t = a\right]$$

行为价值（函数）是状态行为对价值(函数);而状态价值（函数）或价值（函数）多用于表示单纯基于状态的价值（函数）。

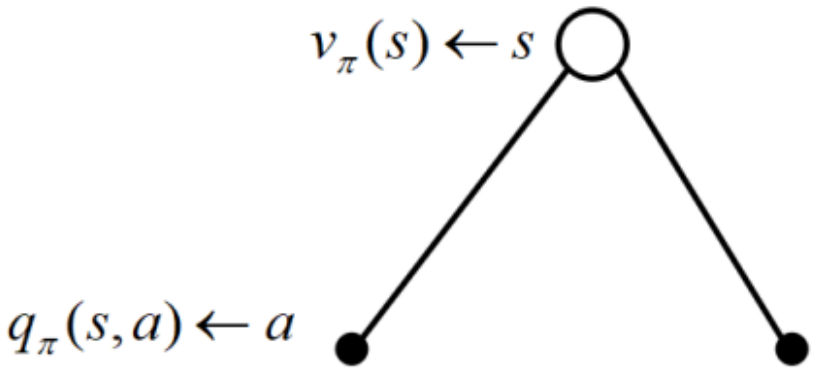
定义了基于策略 π 的状态价值函数和行为价值函数后，依据贝尔曼方程，可以得到如下两个贝尔曼期望方程：

$$v_\pi(s) = E(R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s)$$

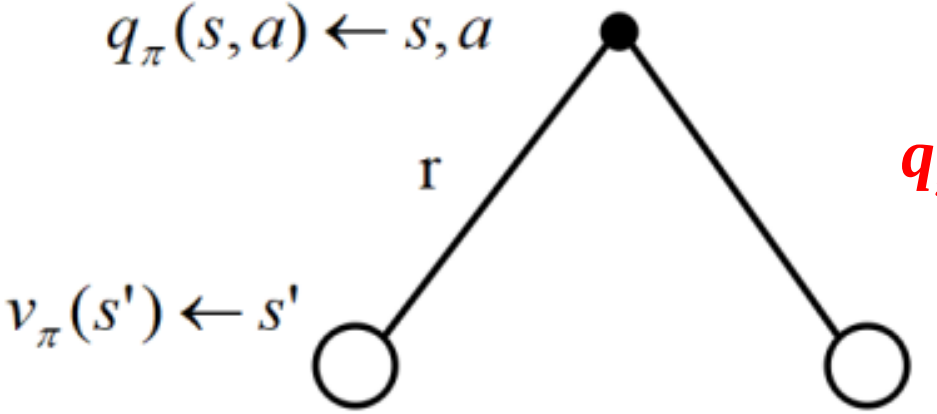
$$q_\pi(s) = E(R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a)$$

由于行为是连接马尔科夫决策过程中状态转换的桥梁，一个行为的价值与状态的价值关系紧密。一个状态的价值可以用该状态下所有行为价值来表达：

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

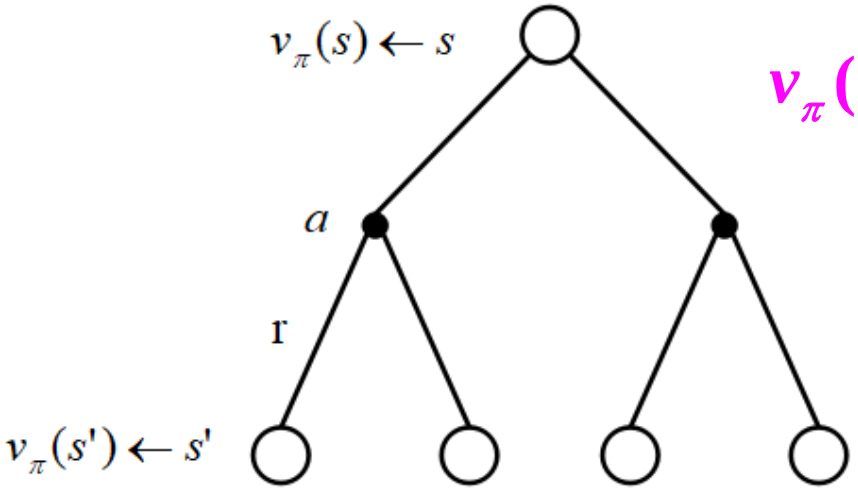


一个行为的价值可以用该行为所能到达的后续状态的价值来表达：



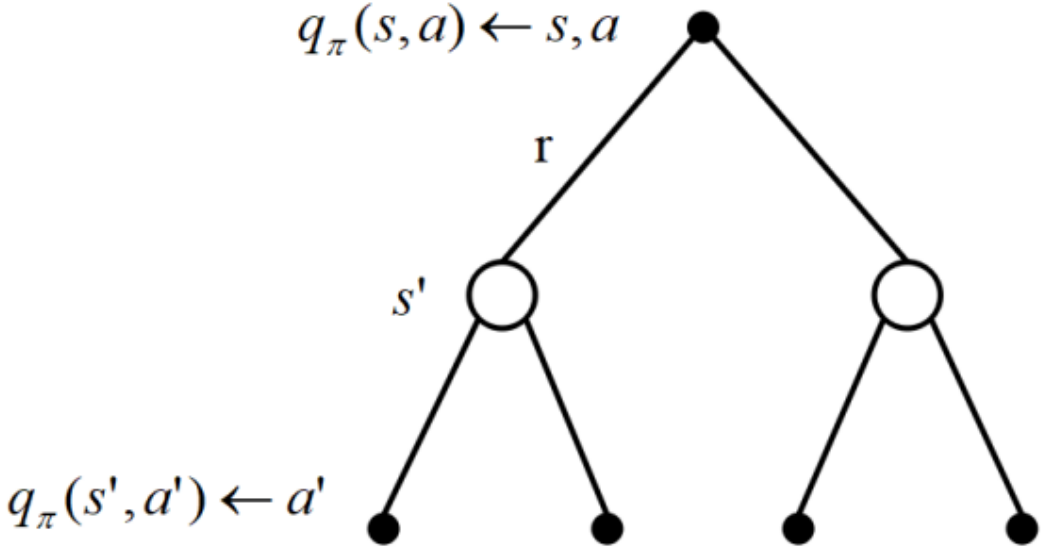
$$q_\pi(s,a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_\pi(s')$$

如果把上二式组合起来，可以得到下面的结果：



$$v_\pi(s) = \sum_{a \in A} \pi(a \mid s) \left(R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_\pi(s') \right)$$

行为的价值也可以用如下来表示：

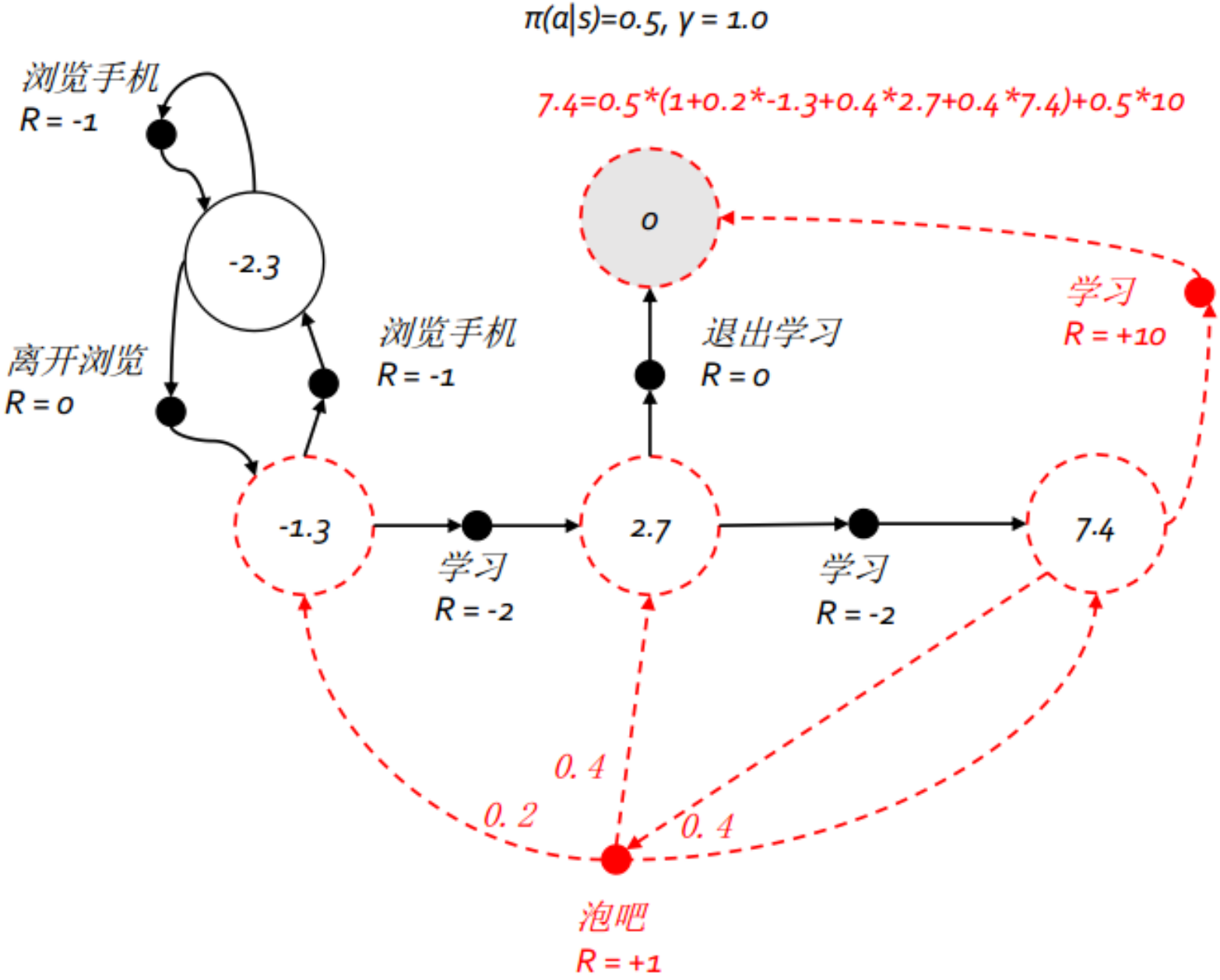


$$q_\pi(s,a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \left(\sum_{a \in A} \pi(a' \mid s')\, q_\pi(s',a') \right)$$

给出了一个给定策略下学生马尔科夫决策过程的价值函数。每一个状态下都有且仅有2 个实质可发生的行为，采用策略是两种行为以均等 (各 0.5) 的概率被选择执行，同时衰减因子 γ = 1。图中状态“第三节课”在该策略下的价值为 7.4。

$$v_\pi(s) = \sum_{a \in A} \pi(a \mid s)\left(R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_\pi(s')\right)$$

$$v_\pi(s) = \sum_{a \in A} \pi(a \mid s) \left(R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_\pi(s') \right)$$



定义：最优状态价值函数（optimal value function）是所有策略下产生的众多状态价值函数中的最大者：

$$v_* = \max v_\pi(s)$$

定义：最优行为价值函数（optimal action-value function）是所有策略下产生的众多行为价值函数中的最大者：

$$q_*(s,a) = \max q_\pi(s,a)$$

定义：策略 π 优于 $\pi'(\pi \geqslant \pi')$，如果对于有限状态集里的任意一个状态 s，不等式：$v_\pi(s) \geqslant v_{\pi'}(s)$ 成立。

结论：

对于任何马尔科夫决策过程，存在一个最优策略 $\pi*$ 优于或至少不差于所有其它策略。

一个马尔科夫决策过程可能存在不止一个最优策略，但最优策略下的状态价值函数均等于最优状态价值函数：

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

最优策略下的行为价值函数均等于最优行为价值函数：

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

最优策略可以通过最大化最优行为价值函数 q*(s, a) 来获得：

$$\pi_*(s,a)=\begin{cases}1 & if\ \arg\max_{a\in A} q_\pi(s,a)\\ 0 & other\end{cases}$$

最优策略在面对每一个状态时将总是选择能够带来最大最优行为价值的行为。当得到 q*(s, a)，最优策略也就找到了。因此求解强化学习问题就转变为了求解最优行为价值函数问题。

状态 s 的最优价值可以由下面的贝尔曼最优方程得到：

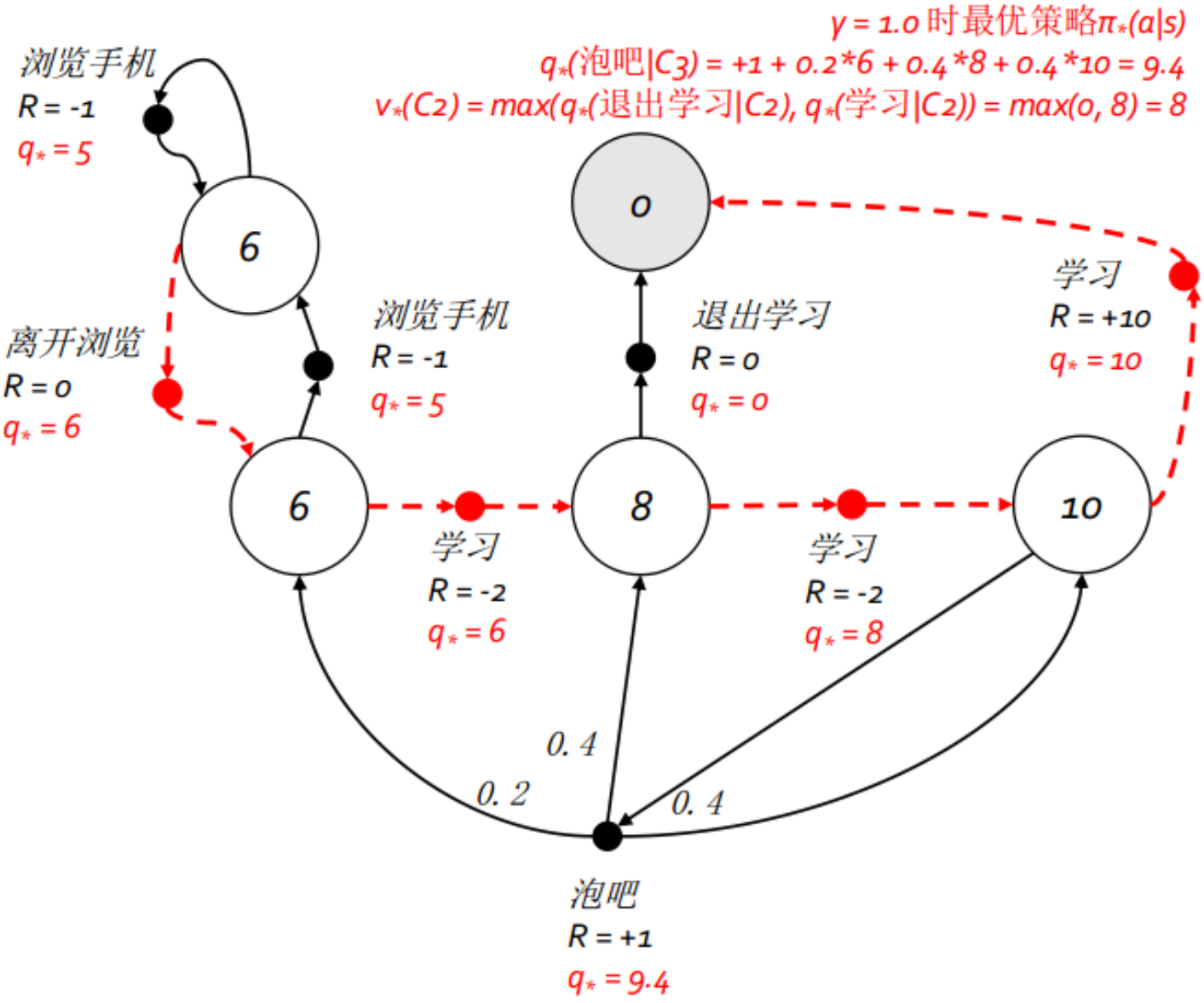
$$v_\pi(s)=\sum_{a\in A}\pi(a\mid s)\,q_\pi(s,a)$$

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

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

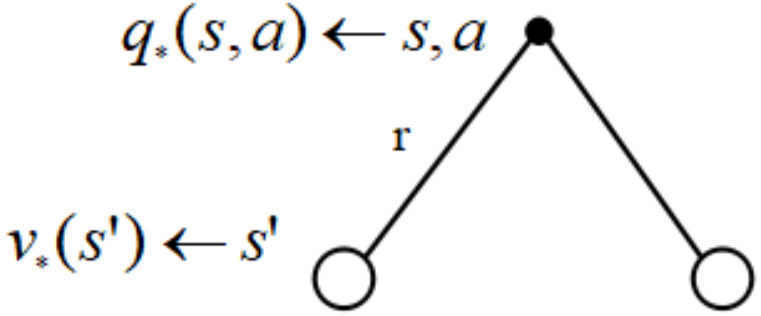
$$\Longrightarrow v_*(s)=\max q_*(s,a)$$

学生马尔科夫决策过程最优策略



马尔科夫决策过程(1)

由贝尔曼最优方程得到：



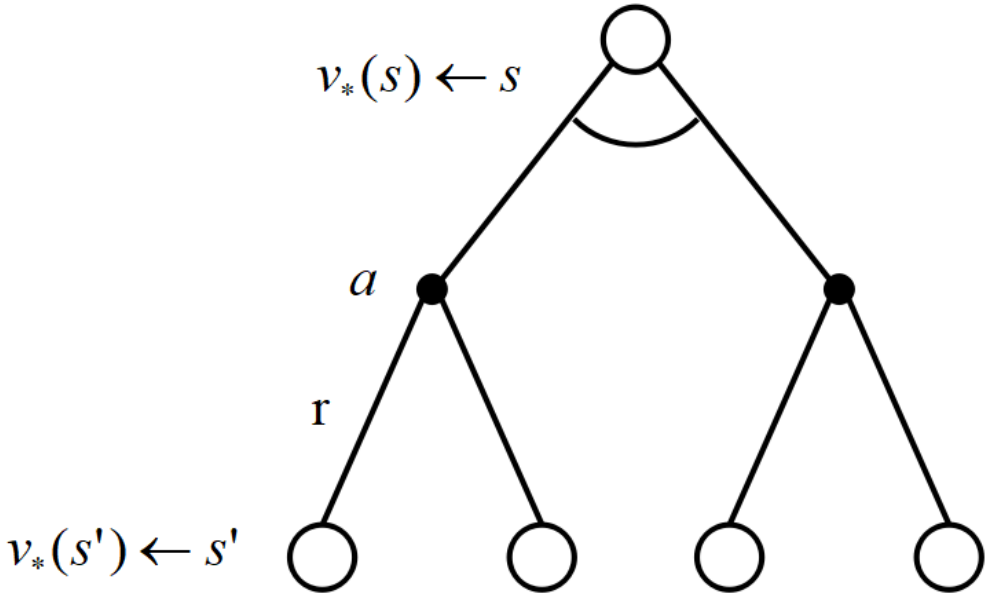
$$q_*(s,a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_*(s')$$

一个行为的最优价值由两部分组成，一部分是执行该行为后环境给予的确定的即时奖励，另一部分则由所有后续可能状态的最优状态价值按发生概率求和乘以衰减系数得到。

可以看出，某状态的最优价值等同于该状态下所有的行为价值中最大者。

马尔科夫决策过程(3)

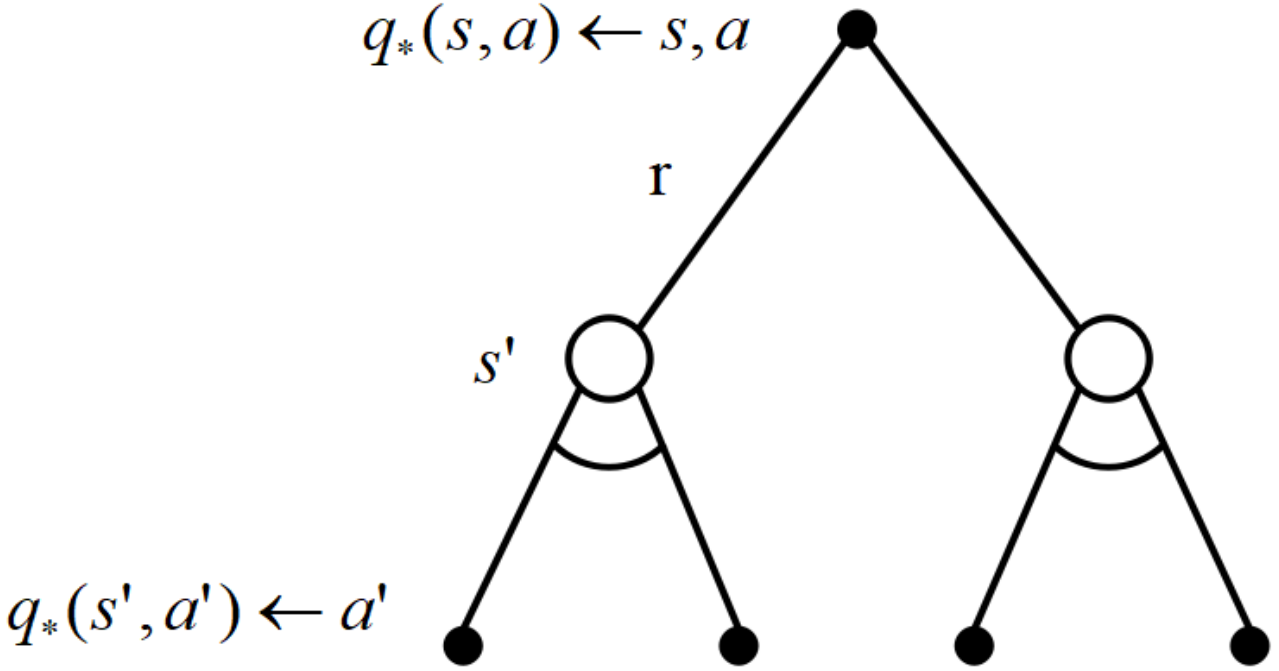
某一行为的最优行为价值可以由该行为可能进入的所有后续状态的最优状态价值来计算得到。



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

马尔科夫决策过程(3)

类似的，最优行为价值函数也可以由后续的最优行为价值函数来计算得到：



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

动态规划寻找最优策略(1)

就是在了解包括状态和行为空间、转移概率矩阵、奖励等信息的基础上判断一个给定策略的价值函数，或判断一个**策略的优劣并最终找到最优的策略和最优价值函数**

动态规划算法**把求解复杂问题分解为求解子问题**，通过求解子问题进而得到整个问题的解。在解决子问题的时候，其结果通常需要存储起来被用来解决后续复杂问题。

动态规划寻找最优策略(2)

预测和控制是规划的两个重要内容。预测是对给定策略的评估过程，控制是寻找一个最优策略的过程。对预测和控制的数学描述如下：

预测 (prediction)： 已知一个马尔科夫决策过程 **MDP** $\langle S, A, P, R, \gamma \rangle$ 和一个策略 π，或者是给定一个马尔科夫奖励过程 **MRP** $\langle S, P_\pi, R_\pi, \gamma \rangle$，求解基于该策略的价值函数 v_π。

控制 (control)： 已知一个马尔科夫决策过程 **MDP** $\langle S, A, P, R, \gamma \rangle$，求解最优价值函数 v_* 和最优策略 π_*。

策略评估

策略评估 (policy evaluation) 指计算给定策略下状态价值函数的过程。

使用同步迭代联合动态规划的算法：从任意一个状态价值函数开始，依据给定的策略，结合贝尔曼期望方程、状态转移概率和奖励同步迭代更新状态价值函数，直至其收敛，得到该策略下最终的状态价值函数。理解该算法的关键在于在一个迭代周期内如何更新每一个状态的价值。

策略评估

贝尔曼期望方程给出了如何根据状态转换关系中的后续状态 s′ 来计算当前状态 s 的价值，在同步迭代法中，使用上一个迭代周期 k 内的后续状态价值来计算更新当前迭代周期 k + 1内某状态 s 的价值：

$$v_{k+1}(s)=\sum_{a\in A}\pi(a|s)\left(R_s^a+\gamma\sum_{s'\in S}P_{ss'}^a v_k(s')\right)$$

可以对计算得到的新的状态价值函数再次进行迭代，直至状态函数收敛，也就是迭代计算得到每一个状态的新价值与原价值差别在一个很小的可接受范围内。

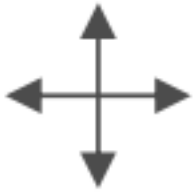
策略评估

贝尔曼期望方程给出了如何根据状态转换关系中的后续状态 s′ 来计算当前状态 s 的价值，在同步迭代法中，使用上一个迭代周期 k 内的后续状态价值来计算更新当前迭代周期 k + 1内某状态 s 的价值：

$$v_{k+1}(s)=\sum_{a\in A}\pi(a\mid s)\left(R_s^a+\gamma\sum_{s'\in S}P_{ss'}^a v_k(s')\right)$$

可以对计算得到的新的状态价值函数再次进行迭代，直至状态函数收敛，也就是迭代计算得到每一个状态的新价值与原价值差别在一个很小的可接受范围内。

策略评估示例



0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

- **即时奖励：**上图是一个九宫格，左上角和右下角是终点，它们的**reward是0**，其它的状态，**reward都是-1**。
- **状态空间：**除了灰色两个格子，其他都是非终点状态
- **动作空间：**在每个状态下，都有四种动作可以执行，分别是上下左右(东西南北)。

策略评估示例

- **转移概率：**任何想要离开格子的动作将保持其状态不变，也就是原地不动。其他时候都是直接移动到下一个状态。所以状态转移概率是确定性的。
- **折扣因子：**$\gamma=1$
- **当前策略：**在任何状态下，agent都采取随机策略，也就是它的动作是随机选择的，即：

$$\pi(e|*)=\pi(w|*)=\pi(s|*)=\pi(n|*)=0.25$$

问题：评估在这个九宫格里给定的策略。也就是说，在策略给定的情况下(这里是随机策略)，求解在该策略下所有状态的$v(s)$值。

策略评估示例

0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0

K=0

0.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	0.0

K=1

-1.0+0.25*0+0.25*0+0.25*0+0.25*0=-1.0

策略评估示例

0.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	0.0

K=1

0.0	-1.7	-2.0	-2.0
-1.7	-2.0	-2.0	-2.0
-2.0	-2.0	-2.0	-1.7
-2.0	-2.0	-1.7	0.0

K=2

-1.0+0.25*0+0.25*(-1.0)*3=-1.7

策略评估示例

0.0	-1.7	-2.0	-2.0
-1.7	-2.0	-2.0	-2.0
-2.0	-2.0	-2.0	-1.7
-2.0	-2.0	-1.7	0.0

K=2

0.0	-2.4	-2.9	-3.0
-2.4	-2.9	-3.0	-2.9
-2.9	-3.0	-2.9	-2.4
-3.0	-2.9	-2.4	0.0

K=3

-1.0+0.25*0+0.25*(-1.7)+0.25*(-2.0)*2=-2.4

策略评估示例

0.0	-2.4	-2.9	-3.0
-2.4	-2.9	-3.0	-2.9
-2.9	-3.0	-2.9	-2.4
-3.0	-2.9	-2.4	0.0

K=3

0.0	-6.1	-8.4	-9.0
-6.1	-7.7	-8.4	-8.4
-8.4	-8.4	-7.7	-6.1
-9.0	-8.4	-6.1	0.0

K=10

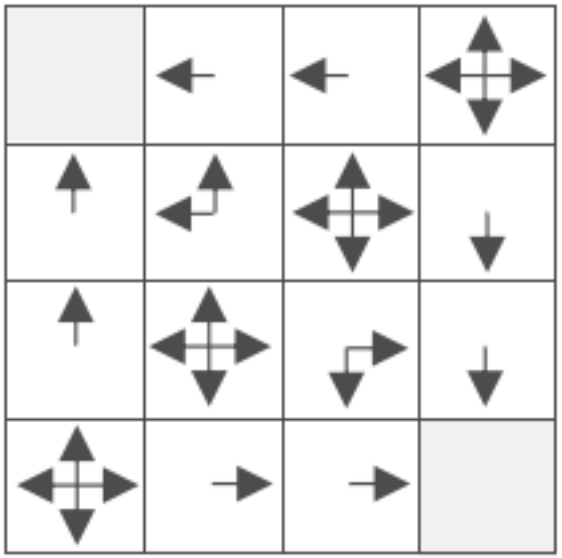
0.0	-14	-20	-22
-14	-18	-20	-20
-20	-20	-18	-14
-22	-20	-14	0.0

K=∞

策略迭代

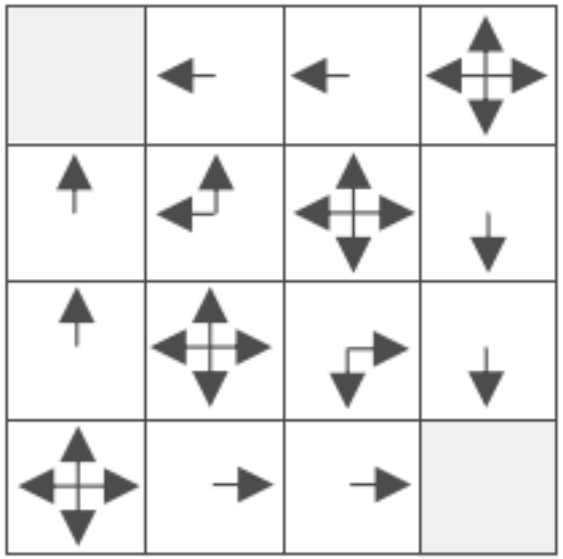
完成对一个策略的评估，将得到基于该策略下每一个状态的价值。很明显，不同状态对应的价值一般也不同，那么个体是否可以根据得到的价值状态来调整自己的行动策略呢，例如：考虑如下的贪心策略：个体在某个状态下选择的行为是其能够到达后续所有可能的状态中价值最大的那个状态。以均一随机策略下第 2 次迭代后产生的价值函数为例说明贪心婪策略。

0.0	-1.7	-2.0	-2.0
-1.7	-2.0	-2.0	-2.0
-2.0	-2.0	-2.0	-1.7
-2.0	-2.0	-1.7	0.0



策略迭代

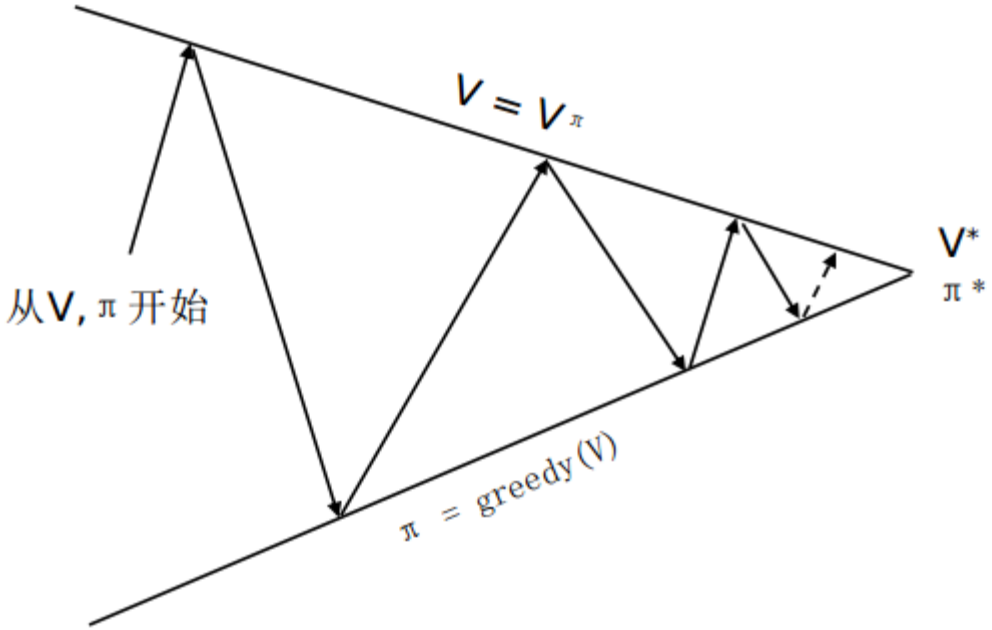
0.0	-1.7	-2.0	-2.0
-1.7	-2.0	-2.0	-2.0
-2.0	-2.0	-2.0	-1.7
-2.0	-2.0	-1.7	0.0

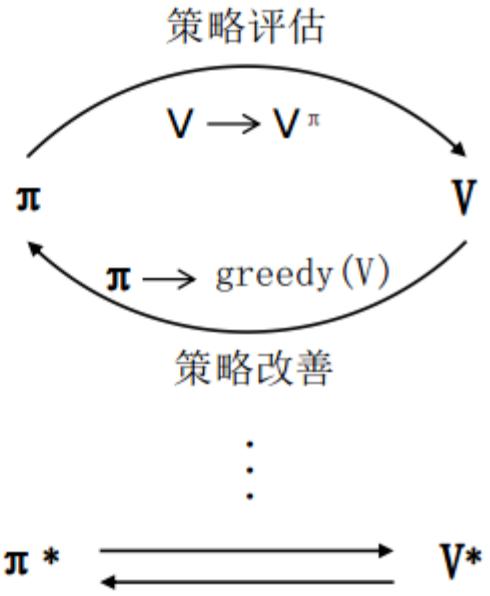


在这个小型方格世界中，新的贪心策略比之前的均一随机策略要优秀不少，至少在靠近终止状态的几个状态中，个体将有一个明确的行为，而不再是随机行为了。从均一随机策略下的价值函数中产生了新的更优秀的策略，这是一个策略改善的过程。

策略迭代

依据新的策略 π′ 会得到一个新的价值函数，并产生新的贪心策略，如此重复循环迭代将最终得到最优价值函数 $v*$ 和最优策略 $\pi*$。策略在循环迭代中得到更新改善的过程称为策略迭代（**policy iteration**）。下图 直观地显示了策略迭代的过程。





基于贪心策略的迭代将收敛于最优策略

考虑一个依据确定性策略 π 对任意状态 s 产生的行为 $a = \pi(s)$，贪心策略在同样的状态 s下会得到新行为：$a' = \pi'(s)$，其中：

$$\pi'(s) = \arg\max_{a \in A} q_\pi(s,a)$$

假如个体在与环境交互的仅下一步采取该贪心策略产生的行为，而在后续步骤仍采取基于原策略产生的行为，那么下面的（不）等式成立：

$$q_\pi(s,\pi'(s)) = \max_{a \in A} q_\pi(s,a) \geq q_\pi(s,\pi(a)) = v_\pi(s)$$

$$q_\pi(s,\pi'(s)) = \max_{a\in A} q_\pi(s,a) \geq q_\pi(s,\pi(a)) = v_\pi(s)$$

由于上式中的 s 对状态集 S 中的所有状态都成立，那么针对状态 s 的所有后续状态均使用贪心策略产生的行为，不等式：$v_{\pi'} \geq v_\pi(s)$ 将成立。这表明新策略下状态价值函数总不劣于原策略下的状态价值函数。该步的推导如下：

$$\begin{aligned} v_\pi(s) &\leq q_\pi(s,\pi'(s)) = E_{\pi'}(R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s) \\ &\leq E_{\pi'}(R_{t+1} + \gamma q_\pi(S_{t+1},\pi'(S_{t+1})) \mid S_t = s) \\ &\leq E_{\pi'}(R_{t+1} + \gamma R_{t+2} + \gamma^2 q_\pi(S_{t+2},\pi'(S_{t+2})) \mid S_t = s) \\ &\leq E_{\pi'}(R_{t+1} + \gamma R_{t+2} + \cdots \mid S_t = s) \\ &\leq v_{\pi'}(s) \end{aligned}$$

如果在某一个迭代周期内，状态价值函数不再改善，即：

$$q_\pi(s,\pi'(s)) = \max_{a\in A} q_\pi(s,a) = q_\pi(s,\pi(s)) = v_\pi(s)$$

那么就满足了贝尔曼最优方程的描述：

$$v_\pi = \max_{a\in A} q_\pi(s,a)$$

此时，对于所有状态集内的状态 s ∈ S，满足：$v_\pi(s) = v_*(s)$，这表明此时的策略 π 即为最优策略。

价值迭代

任何一个最优策略可以分为两个阶段，首先该策略要能产生当前状态下的最优行为，其次对于该最优行为到达的后续状态时该策略仍然是一个最优策略。

如果一个策略不能在当前状态下产生一个最优行为，或者这个策略在针对当前状态的后续状态时不能产生一个最优行为，那么这个策略就不是最优策略。

价值迭代

一个状态的最优价值可以由其后续状态的最优价值通过贝尔曼最优方程来计算：

$$v_*(s) = \max_{a \in A} \left(R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_\pi(s') \right)$$

迭代过程中价值函数更新的公式为：

$$v_{k+1}(s) = \max_{a \in A} \left(R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_k(s') \right)$$

贪心策略迭代示例

- **即时奖励：**如右图的九宫格，左上角是终点，它的**reward是0**，其它的状态，**reward都是-1**。
- **状态空间：**除了左上灰色格子，其他都是非终点状态
- **动作空间：**在每个状态下，都有四种动作可以执行，分别是上下左右(东西南北)。

g	A	B	C
D	E	F	G
H	I	J	K
L	M	N	O

Problem

贪心策略的价值迭代示例

$$v_{k+1}(s) = \max_{a \in A} \left(R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_k(s') \right)$$

g	A	B	C
D	E	F	G
H	I	J	K
L	M	N	O

0			

V_0

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

V_1

0	-1	-1	-1
-1	-1	-1	-1
-1	-1	-1	-1
-1	-1	-1	-1

V_2

0	-1	-2	-2
-1	-2	-2	-2
-2	-2	-2	-2
-2	-2	-2	-2

V_3

0	-1	-2	-3
-1	-2	-3	-3
-2	-3	-3	-3
-3	-3	-3	-3

V_4

0	-1	-2	-3
-1	-2	-3	-4
-2	-3	-4	-4
-3	-4	-4	-4

V_5

0	-1	-2	-3
-1	-2	-3	-4
-2	-3	-4	-5
-3	-4	-5	-5

V_6

0	-1	-2	-3
-1	-2	-3	-4
-2	-3	-4	-5
-3	-4	-5	-6

V_7

异步动态规划算法

- 原位动态规划 (in-place dynamic programming)：与同步动态规划算法通常对状态价值保留一个额外备份不同，原位动态规划则直接利用当前状态的后续状态的价值来更新当前状态的价值。
- 优先级动态规划 (prioritised sweeping)：对每一个状态进行优先级分级，优先级越高的状态其状态价值优先得到更新。通常使用贝尔曼误差来评估状态的优先级，贝尔曼误差被表示为新状态价值与前次计算得到的状态价值差的绝对值。
- 实时动态规划 (real-time dynamic programming)：使用个体与环境交互产生的实际经历来更新状态价值，对个体实际经历过的状态进行价值更新。经常访问过的状态将得到较高频次的价值更新，而与个体关系不密切、个体较少访问到的状态其价值得到更新的机会就较少。