法律声明

- 本课件包括：演示文稿，示例，代码，题库，视频和声音等，小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。

关注 小象学院

第2讲　马尔科夫决策过程

强化学习

主讲人：叶梓

上海交通大学博士

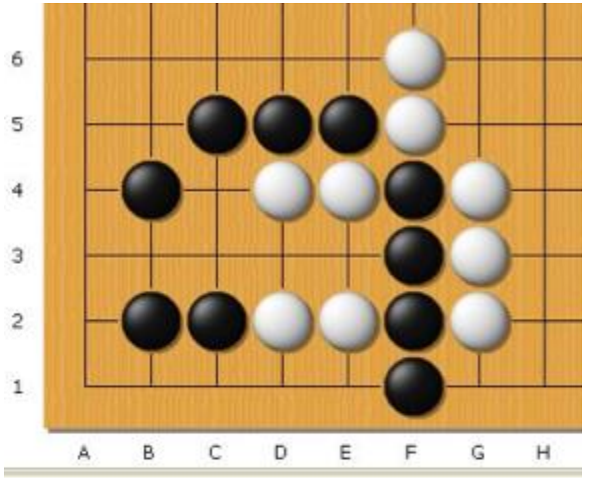
主要研究方向：机器学习、深度学习、人工智能

本章内容

- 马尔科夫性质
- 马尔科夫奖励/决策过程
- 策略
- 状态值函数/状态-行为值函数
- Bellman方程
- 最优价值函数
- 最优策略
- 示例

马尔科夫性质

- 马尔科夫性：系统的下一个状态只与当前状态有关，与以前状态无关。
- 定义：一个状态S_t是马尔科夫的，当且仅当：
  - $P(S_{t+1}|S_t) = P(S_{t+1}|S_1, \cdots, S_t)$
- 特点：
  - 当前状态蕴含所有相关的历史信息
  - 一旦当前状态已知，历史信息将会被抛弃



马尔科夫过程

- 马尔科夫过程，即该过程中所有的状态均满足马尔科夫性，它可以表示为一个二元组。
- 它包含了状态集和状态转移矩阵。

Definition

A *Markov Process* (or *Markov Chain*) is a tuple $\langle \mathcal{S}, \mathcal{P} \rangle$

- $\mathcal{S}$ is a (finite) set of states
- $\mathcal{P}$ is a state transition probability matrix,
  $\mathcal{P}_{ss'} = \mathbb{P}\left[S_{t+1} = s' \mid S_t = s\right]$

状态转移概率矩阵

- 系统有N个状态，P描述各种状态下向其他状态转移的概率矩阵。

定义为：

$$\begin{pmatrix} P_{11} & P_{12} \ldots P_{1N} \\ P_{21} & P_{22} \ldots P_{2N} \\ \vdots & \vdots \quad\quad \vdots \\ P_{N1} & P_{N2} \ldots P_{NN} \end{pmatrix}$$

- 状态转移概率矩阵是一个N阶方阵，满足概率矩阵性质

1） $P_{ij} \geq 0$, 非负性性质

2） $\sum_j P_{ij} = 1$, 每行的和为1。

马尔科夫奖励过程MRP

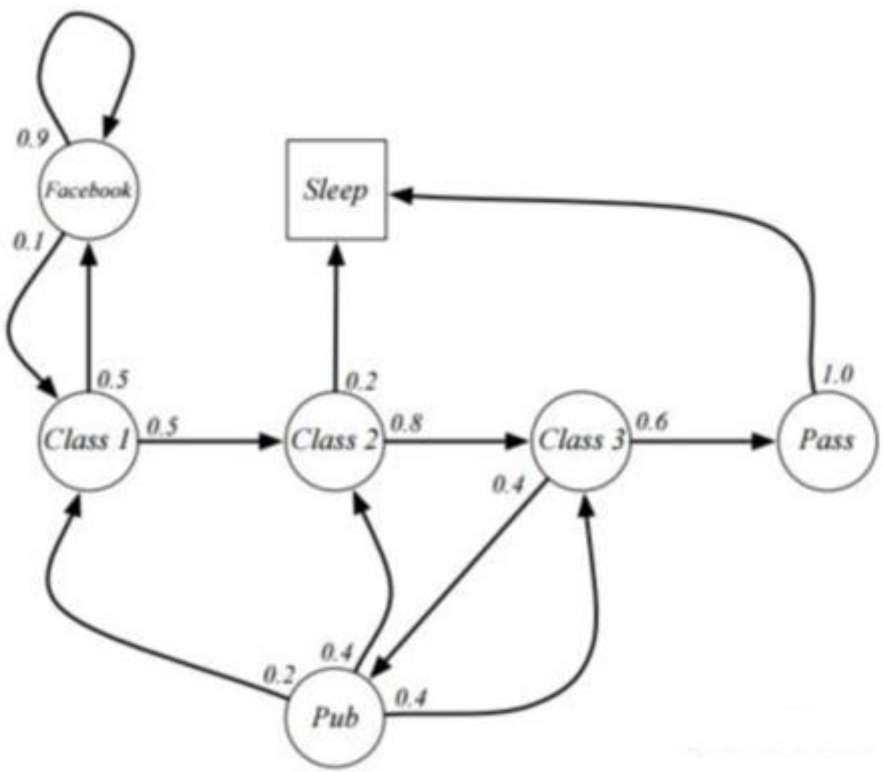
- 当马尔科夫过程是四元组：M = <S, P, R, γ >时，称作马尔科夫奖励过程。
- 与马尔科夫过程相比，马尔科夫奖励过程（MRP）又多了一个**奖励函数**以及一个**折扣因子**。

Definition

A *Markov Reward Process* is a tuple $\langle \mathcal{S}, \mathcal{P}, \mathcal{R}, \gamma \rangle$

- $\mathcal{S}$ is a finite set of states
- $\mathcal{P}$ is a state transition probability matrix,
  $\mathcal{P}_{ss'} = \mathbb{P}\left[S_{t+1} = s' \mid S_t = s\right]$
- $\mathcal{R}$ is a reward function, $\mathcal{R}_s = \mathbb{E}\left[R_{t+1} \mid S_t = s\right]$
- γ is a discount factor, $\gamma \in [0, 1]$

马尔科夫奖励过程示例

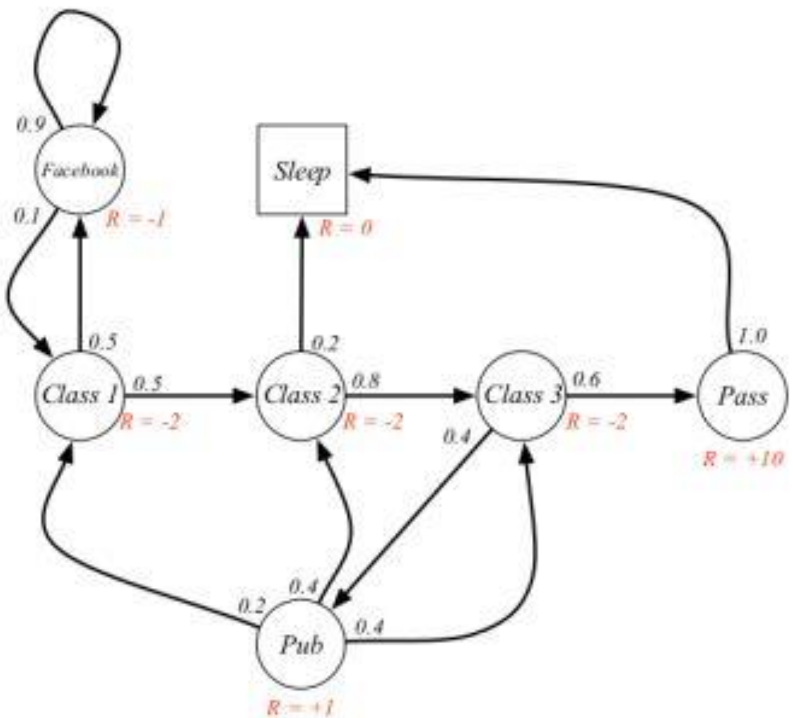


一个Student的例子

- 圆圈表示学生所处的状态：
- 方格Sleep是一个终止状态，或者可以描述成自循环的状态，也就是Sleep状态的下一个状态100%的几率还是自己。
- 箭头表示状态之间的转移，箭头上的数字表示当前转移的概率。

马尔科夫奖励过程示例

- 该学生马尔科夫奖励过程的即时奖励如左图：
- 状态转移概率矩阵如右图：



States	C1	C2	C3	Pass	Pub	FB	Sleep
Rewards	-2	-2	-2	10	1	-1	0
C1		0.5				0.5	
C2			0.8				0.2
C3				0.6	0.4		
Pass							1
Pub	0.2	0.4	0.4				
FB	0.1					0.9	
Sleep							1

马尔科夫决策过程 MDP

- 一个马尔科夫决策过程常由一个五元组（tuple）描述。
- 与MRP相比，马尔科夫决策过程（MDP）加入了动作集，
- 该过程中的状态并不是自发地按照某个概率进行转移，而是通过选择某个动作来进行转移的。

S
状态States的集合

R
一个由状态和动作到实数的映射，称为奖励方程Reward Function

A
动作Actions的集合

γ
一个大于等于0，小于1的值，称为折扣因子Discount Factor

$\{P_{ss'}^{a}\}$
状态转移概率矩阵，表示在状态s的情况下，执行动作a，然后转移到状态s'的概率。

形式化描述：马尔科夫过程

- 用于描述强化学习的马尔科夫过程是五元组 M = <S, A, P, R, γ >。
- 其中：
  - S是有限的状态集；A是有限的行动集。
  - P = S×A×S→ [0,1]是状态转换模型，$P(s,a,s')$ 表示状态s下执行行动a到达状态s'的概率，且满足$\sum_{s'} P(s,a,s') = 1$。或记作：$P_{ss'}^{a}$=P[S_{t+1}=s'|S_t=s, A_t=a]
  - R = S×A→ **R**(实数)，是即时奖励函数，$R(s,a)$表示状态s下执行行动a后可以得到的即时奖励的期望。

    $\mathcal{R}$ is a reward function, $\mathcal{R}_s^a = \mathbb{E}\left[R_{t+1} \mid S_t = s, A_t = a\right]$
  - 0≤γ ≤ 1。

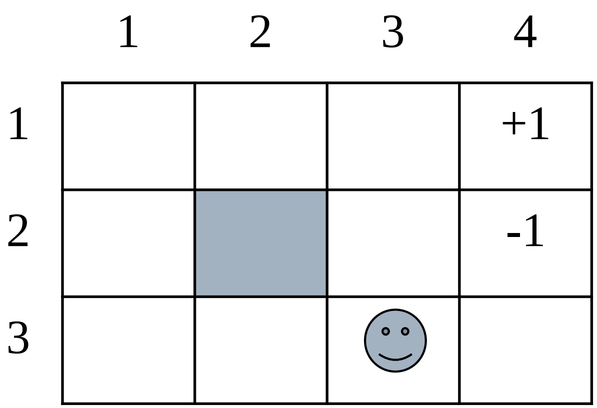
马尔科夫决策过程的示例

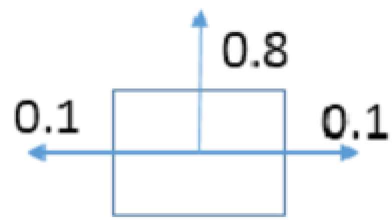
- 机器人在网格世界中，灰色单元是一个障碍（不能位于该单元）。在该问题中，MDP 的各元组是:
- S：机器人可以在11个网格中的任何一个，集合S 对应11个可能到达的位置。
- A：机器人可以做的动作
  - 向东(E) 向南(S) 向西(W) 向北(N)。
- R：机器人获得的奖励
  - 目的地是格子(4,1)，用r = +1来奖励；
  - 要避开格子(4,2)，用r = -1来惩罚；
  - 其他格子r = -0.02（模拟耗电，避免绕圈）。

	1	2	3	4
1				+1
2				-1
3				

马尔科夫决策过程的示例

- $P_{ss'}^{a}$：假设机器人的行为不那么精准，在执行相关动作a后有可能会走偏方向。
- $P_{ss'}^{a}$也隐含体现了马尔科夫性质：一个随机过程的未来状态的条件概率分布仅依赖于**当前状态**与**该状态下的动作**。
- $P_{(3,3)(3,2)}^{N} = 0.8$；$P_{(3,3)(4,3)}^{N} = 0.1$；$P_{(3,3)(2,3)}^{N} = 0.1$





$$\sum_{s'} P_{ss'}^{a} = 1$$

折扣系数

- 马尔科夫过程的三种情况

T=1，
greedy case
此算法是退化的却很重要

1<T<∞，
finite horizon case
每个状态的收益折减都一样

T=∞，
infinite-horizon case

- 1、只考虑下一步动作带来的影响，而不会考虑之后一系列动作带来的影响。虽然简单，很多情况下它就是最优解。
- 2、finite-horizon 的处理方式是最符合实际情况的。
- 3、在这种情况下，γ的取值很重要，因为需要保证计算结果是收敛的。

长期回报

- 有限期任务和无限期任务可以统一起来

回报
- 无限期折扣回报
- 有限期折扣回报

- 无限期回报是：

$$R_t = \sum_{k=0}^{\infty} r_{t+k+1}$$

k=∞ 和 γ=1 不能同时出现。

- 无限期折扣回报是：

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

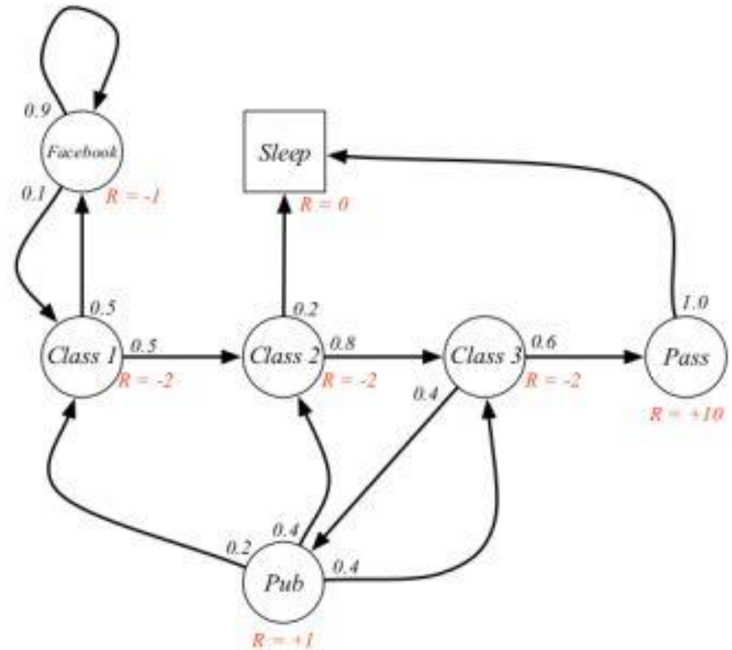
R是总回报，r是即时回报

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

G是总回报，R是即时回报

计算一下长期回报

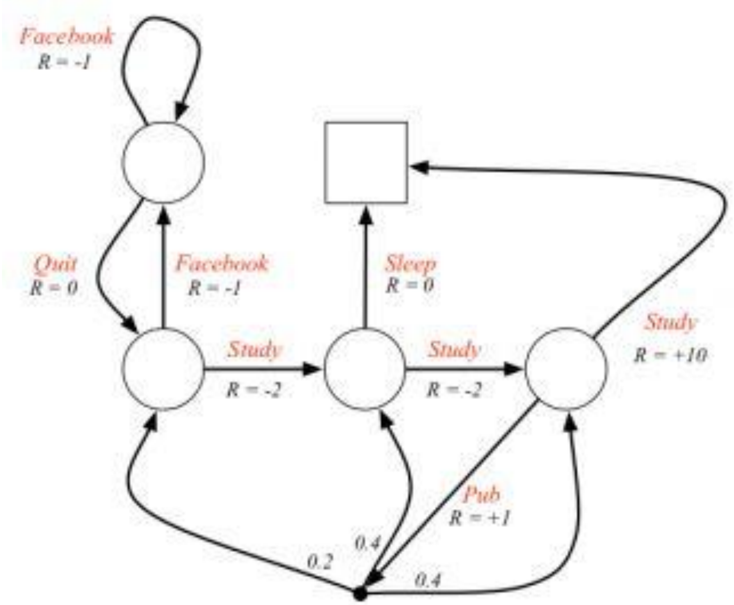
- 假定从class1开始，经历了不同的路径，则各路径长期回报分别按以下各式来计算。
- 其中，$\gamma = 1/2$



C1 C2 C3 Pass Sleep	$v_1 = -2 - 2 * \frac{1}{2} - 2 * \frac{1}{4} + 10 * \frac{1}{8}$	$= -2.25$
C1 FB FB C1 C2 Sleep	$v_1 = -2 - 1 * \frac{1}{2} - 1 * \frac{1}{4} - 2 * \frac{1}{8} - 2 * \frac{1}{16}$	$= -3.125$
C1 C2 C3 Pub C2 C3 Pass Sleep	$v_1 = -2 - 2 * \frac{1}{2} - 2 * \frac{1}{4} + 1 * \frac{1}{8} - 2 * \frac{1}{16} \dots$	$= -3.41$
C1 FB FB C1 C2 C3 Pub C1 ...	$v_1 = -2 - 1 * \frac{1}{2} - 1 * \frac{1}{4} - 2 * \frac{1}{8} - 2 * \frac{1}{16} \dots$	$= -3.20$

马尔科夫决策过程示例

- 图中红色的文字表示的是采取的行为，而不是先前的状态名。
- 对比之前的学生MRP示例可以发现，即时奖励与行为对应了，同一个状态下采取不同的行为得到的即时奖励是不一样的。
- 由于引入了Action，为避免与状态名混淆，因此此图没有给出各状态的名称；
- 此图还把Pass和Sleep状态合并成一个终止状态；
- 另外当选择“Pub”这个动作时，**主动进入了一个临时**状态（图中用黑色小实点表示），随后**被动的**被环境按照其动力学分配到另外三个状态。



小象学院 ChinaHadoop.cn

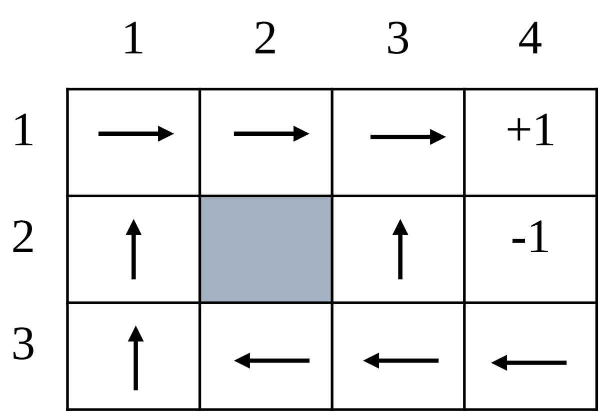
策略（Policy）

- 策略，规定了在每个可能的状态，Agent应该采取的动作的概率分布。
- 策略是强化学习的核心部分，策略的好坏最终决定了Agent的行动和整体性能。
- 策略可以是随机的，也可以是确定的。

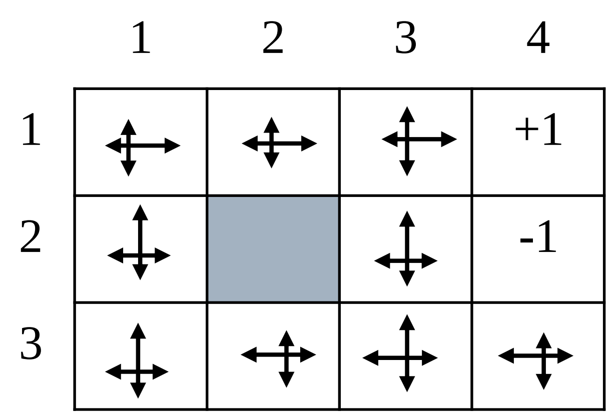
策略

- 一个策略(policy)函数定义为π:S→A，即输入为状态S，输出为A，
- 亦即策略π(s)=a，告诉机器人在状态s下该执行的动作是什么。

确定策略



不确定策略



允许策略与最优策略

- 关于任意状态所能选择的策略组成的集合F，称为允许策略集合，π∈ F。
- 在允许策略集合中找出使问题具有最优效果的策略π^*，称为最优策略。

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

贪婪策略

$$\pi(a|s)\leftarrow\begin{cases}1-\varepsilon+\dfrac{\varepsilon}{|A(s)|} & if\ a=arg\max_a Q(s,a)\\ \dfrac{\varepsilon}{|A(s)|} & if\ a\neq arg\max_a Q(s,a)\end{cases}$$

ε-greedy策略

MDP与MP、MRP的关系

可以从一个MDP模型中恢复MP与MRP

如果策略是已知的

- Given an MDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ and a policy π
- The state sequence $S_1, S_2, ...$ is a Markov process $\langle \mathcal{S}, \mathcal{P}^\pi \rangle$
- The state and reward sequence $S_1, R_2, S_2, ...$ is a Markov reward process $\langle \mathcal{S}, \mathcal{P}^\pi, \mathcal{R}^\pi, \gamma \rangle$
- where

以策略的可能性作为权重，进行通加。

$$\mathcal{P}^\pi_{s,s'} = \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{P}^a_{ss'}$$

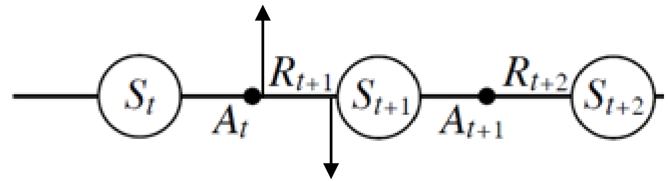
$$\mathcal{R}^\pi_s = \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{R}^a_s$$

MDP中的两重随机性

- 策略的随机性（Agent主观的）
  - 可能的动作：认真听课、开小差
  - 认真听课，还是开小差——掷骰子决定

- 状态转移概率的随机性（与环境交互导致的）
  - 认真听课：
    - 可能考过了(90%)，也可能考挂了(10%)
  - 开小差：
    - 可能考过了(1%)，也可能考挂了(99%)

R是谁的函数？



- 即时奖励R_{t+1}依赖于S_t和A_t的函数，还是依赖于S_{t+1}？先得奖励再转移，还是反之？

  $\mathcal{R}$ is a reward function, $\mathcal{R}_s^a = \mathbb{E}\left[R_{t+1} \mid S_t = s, A_t = a\right]$

- 按定义来看，R_{t+1}好像直接与S_t和A_t有关，经常写成r(s,a)；
- 但实际上R_{t+1}可以实现为进入S_{t+1}后（的同时）立刻到了R_{t+1}。当$P_{ss'}^a$不为1时，这样处理也比较容易。
- 正因为$P_{ss'}^a$的存在，才使得R_s^a是一个期望值。

插播一点数学知识：期望

- 离散型随机变量X的取值为$X_1, X_2, X_3, \ldots, X_n$；$p(X_1), p(X_2), p(X_3), \ldots, p(X_n)$ 为X_i对应取值的概率。

$$E(X) = X_1 * p(X_1) + X_2 * p(X_2) + \ldots + X_n * p(X_n)$$

$$E(X) = \sum_{k=1}^{\infty} x_k p_k$$

随机变量取值，基于其出现概率加权和

- 设连续性随机变量X的概率密度函数为f(x)，若积分绝对收敛，则称积分的值为随机变量的数学期望。

$$E(X) = \int_{-\infty}^{\infty} x f(x) dx$$

值函数（Value Function）

- 强化学习往往又具有延迟回报的特点：
  - 如果在第n步输掉了棋，
  - 那么只有状态s_n和动作a_n获得了即时奖励$r(s_n,a_n) = -1$，
  - 前面的所有状态立即奖励均为0。
- 所以对于之前的任意状态s和动作a，即时奖励函数r(s,a)无法说明策略的好坏。
- 因而需要定义值函数(value function)来表明当前状态下策略π的长期影响。

状态值函数

- **状态值函数**是指Agent在状态s_t根据策略π采取后续动作所得到的**累积回报的期望**，记为$V(s_t)$。

累计回报的期望

$$v_\pi(s) = E_\pi[G_t|S_t = s] = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s\right]$$

- 其中，累计回报的定义如下：

各个step的即时奖励的折后和

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

状态-行为值函数

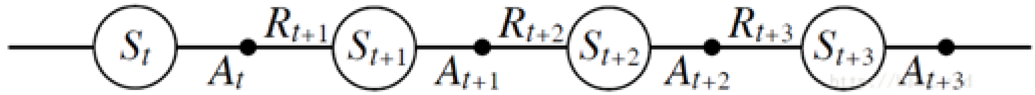
- 状态-行为值函数$q_\pi(s,a)$是指Agent从状态s出发，采取行为a后，
  - 特别注意，这个action不一定是依据π产生的！！
- 然后按照策略π采取行为得到的累计回报的期望。

这里仍是累计回报的期望

$$q_\pi(s,a) = E_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1} \middle| S_t = s, A_t = a\right]$$

区别是已知条件是s和a

Bellman方程



- $v_\pi(s) = E_\pi[G_t|S_t=s]$
  - $v_\pi(s) = E_\pi[R_{t+1}+\gamma R_{t+2}+\gamma^2 R_{t+3}+\ldots|S_t=s]$
  - $v_\pi(s) = E_\pi[R_{t+1}+\gamma(R_{t+2}+\gamma R_{t+3}+\ldots)|S_t=s]$
  - $v_\pi(s) = E_\pi[R_{t+1}+\gamma G_{t+1}|S_t=s]$
- $v_\pi(s) = E_\pi[R_{t+1}+\gamma v_\pi(S_{t+1})|S_t=s]$

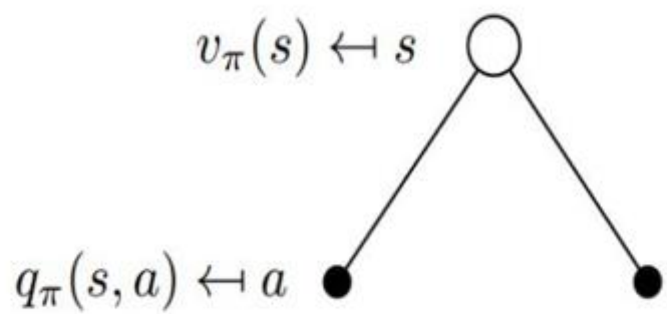
- 同理可推导出：
- $q_\pi(s,a) = E_\pi[R_{t+1}+\gamma q_\pi(S_{t+1}, A_{t+1})|S_t=s, A_t=a]$

Bellman方程

- 某个状态的值函数等于该状态下所有状态行为值函数$q_\pi(s,a)$的加权和。
- 这里的权重就是该状态下采取该行为的概率，即策略$\pi(a|s)$。

s状态下，所有可能的a

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

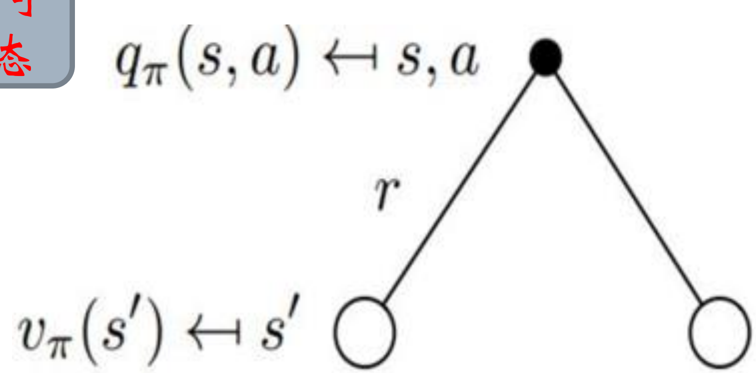


Bellman方程

- 状态行为值函数等于该状态、该行为执行后的即时奖励（的期望），加上它所导致的所有下一步状态的折减后状态值函数$v_\pi(s)$的加权和。
- 这里的权重是该状态下、该行为所导致的下一步状态的概率，即状态转移概率矩阵。
- 其中，$P_{ss'}^{a}$=P[S_{t+1}=s′|S_t=s, A_t=a]

a行为下，所有可能转移到的状态

$$q_\pi(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_\pi(s')$$



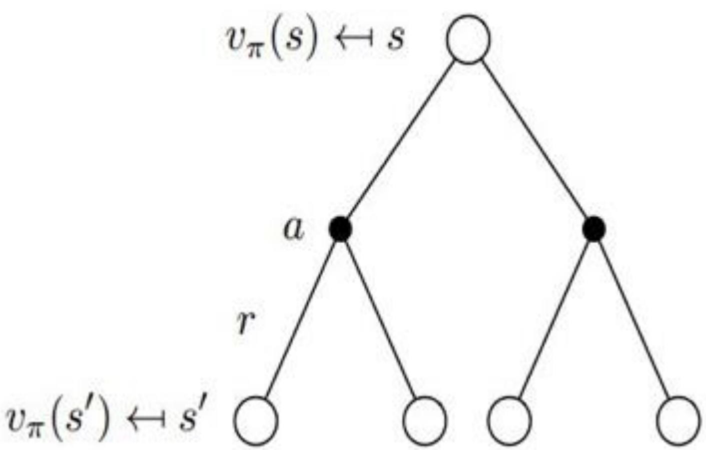
R_s^a = E[R_{t+1}|S_t=s,A_t=a]

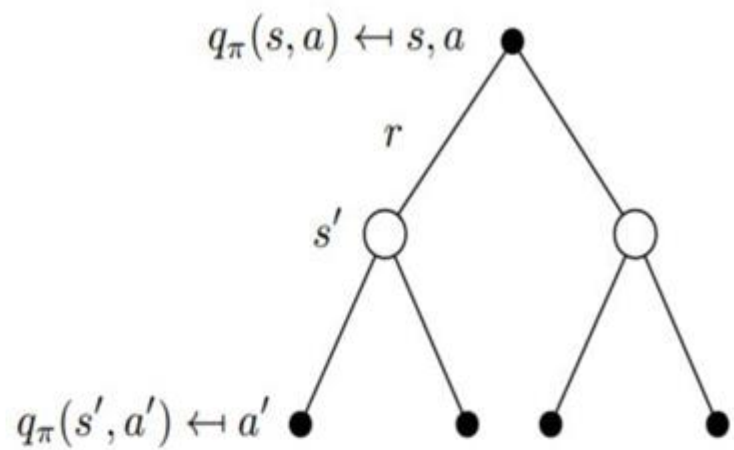
Bellman方程

- Bellman方程其实是$v_\pi(s)$和$q_\pi(s,a)$自身以及相互之间的递推关系。

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

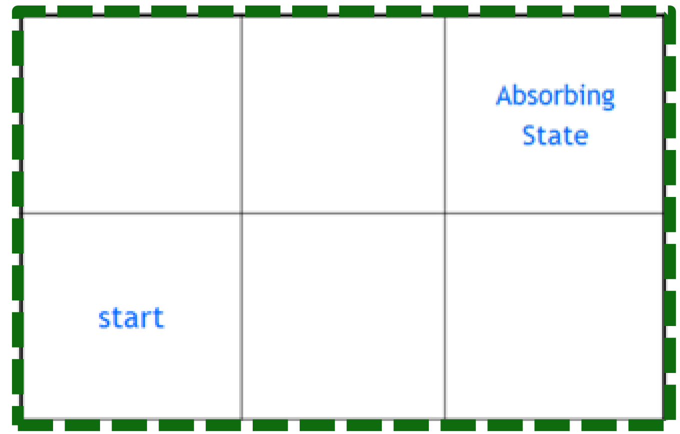
$$q_\pi(s,a) = R_s^a + \gamma\sum_{s'\in S}P_{ss'}^a\sum_{a'\in A}\pi(a'|s')\,q_\pi(s',a')$$





值函数与Q函数的计算

- 如图所示是一个格子世界，假设agent从左下角的start点出发，
- 右上角为目标位置，称为吸收状态(Absorbing state)。
- 对于进入吸收态的动作，给予立即回报100，对其他动作则给予0回报。
- 折减因子γ值我们选择0.9。



值函数与Q函数的计算

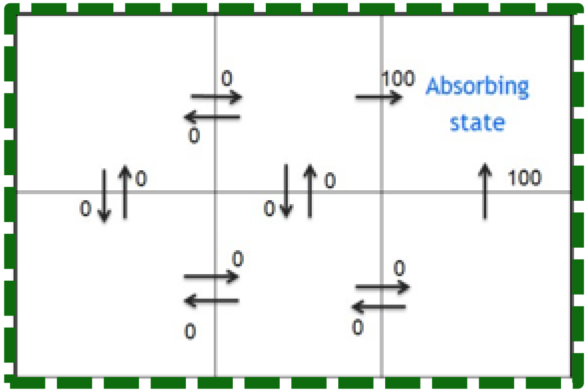
- 记第i行，第j列的状态为s_{ij}，在每个状态，
- 有四种上下左右四种可选的动作，分别记为au, ad, al, ar。
- 并认为状态按动作a选择的方向转移的概率为1。

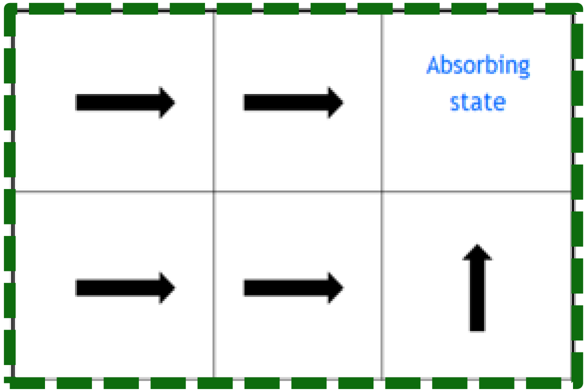
s_{11}	s_{12}	s_{13}
s_{21}	s_{22}	s_{23}

当s和a确定时，$P_{ss'}^{a}$={1}

值函数与Q函数的计算

- 由于状态转移概率是1，每组(s,a)对应了唯一的s'。
- 即时奖励函数r(s'|s,a)可以简记为r(s,a)
- 如图所示，每个格子代表一个状态s，箭头则代表动作a，旁边的数字代表立即奖励，可以看到只有进入目标位置的动作获得了奖励100，其他动作都获得了0奖励。





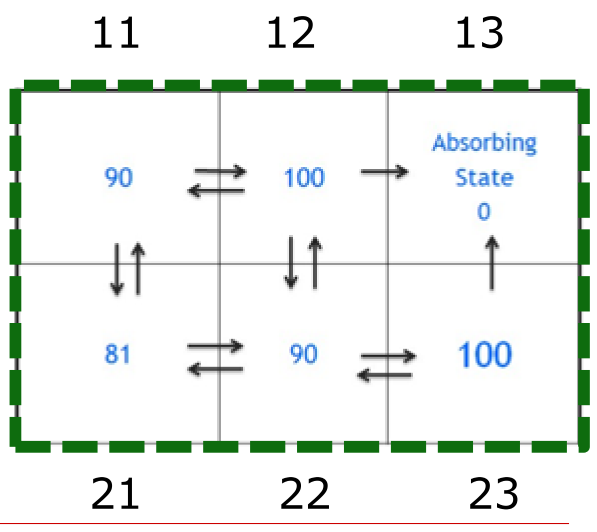
- 即r(s12,ar) = r(s23,au) =100。
- 假定已知策略如左下图。

值函数与Q函数的计算

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

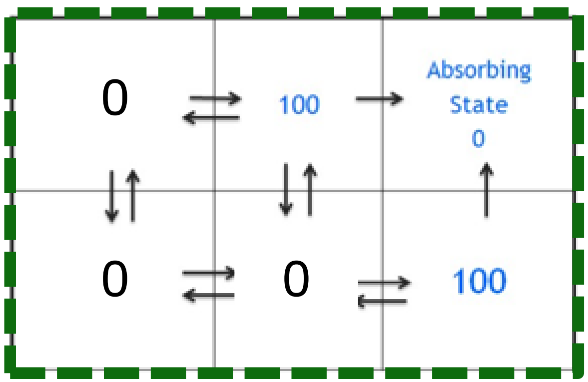
- 值函数$V_\pi(s)$的计算：根据V_π的表达式，即时奖励r(s,a)和策略π，可得：

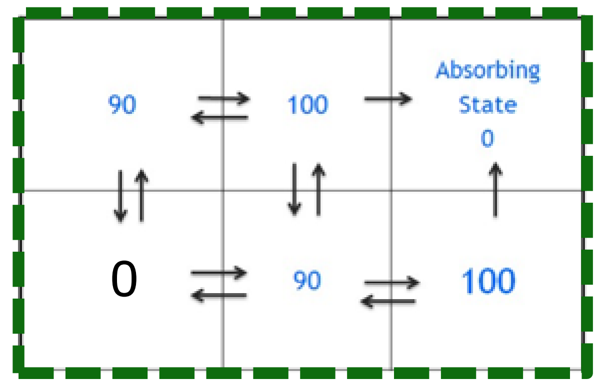
- $V_\pi(s_{12}) = r(s_{12},ar) = r(s_{13}|s_{12},ar) = 100$
- $V_\pi(s_{11}) = r(s_{11},ar)+\gamma*V_\pi(s_{12}) = 0+0.9*100 = 90$
- $V_\pi(s_{23}) = r(s_{23},au) = 100$
- $V_\pi(s_{22}) = r(s_{22},ar)+\gamma*V_\pi(s_{23}) = 90$
- $V_\pi(s_{21}) = r(s_{21},ar)+\gamma*V_\pi(s_{22}) = 81$

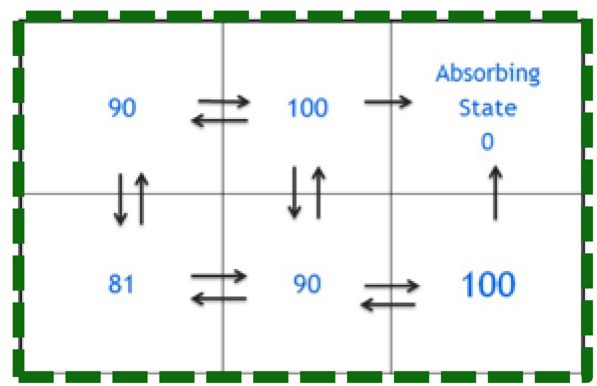


值函数与Q函数的计算

- 如果不是这样的计算顺序呢？







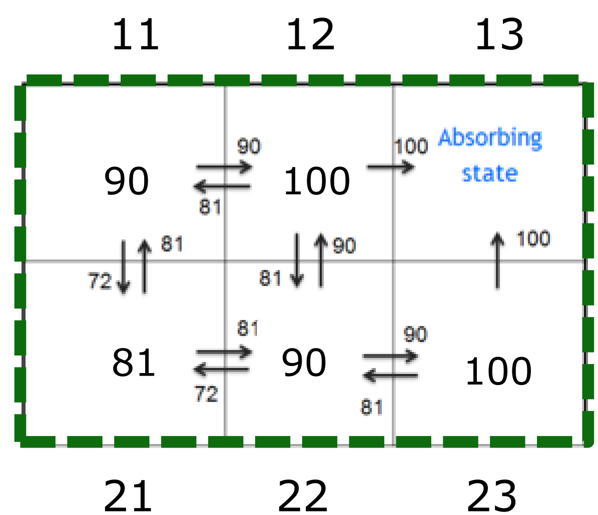
没关系！多迭代几次会得到相同的效果。

值函数与Q函数的计算

$$q_\pi(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_\pi(s')$$

- Q(s,a)值的计算
- 有了策略π和立即奖励函数r(s,a), Q_π(s,a)如何得到的呢?
- 对s_{11}计算Q函数（用到了上面V_π的结果）如下：
- $Q_\pi(s_{11},ar)$
  - $=r(s_{11},ar)+\gamma *V_\pi(s_{12}) = 0+0.9*100 = 90$
- $Q_\pi(s_{11},ad)$
  - $=r(s_{11},ad)+\gamma *V_\pi(s_{21}) = 72$

验证一下：$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) q_\pi(s,a)$

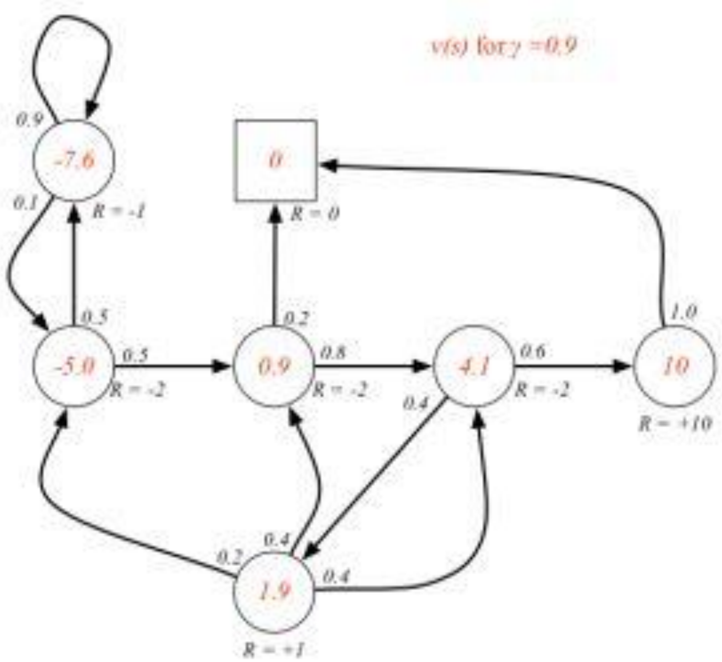


学生MRP问题的状态值

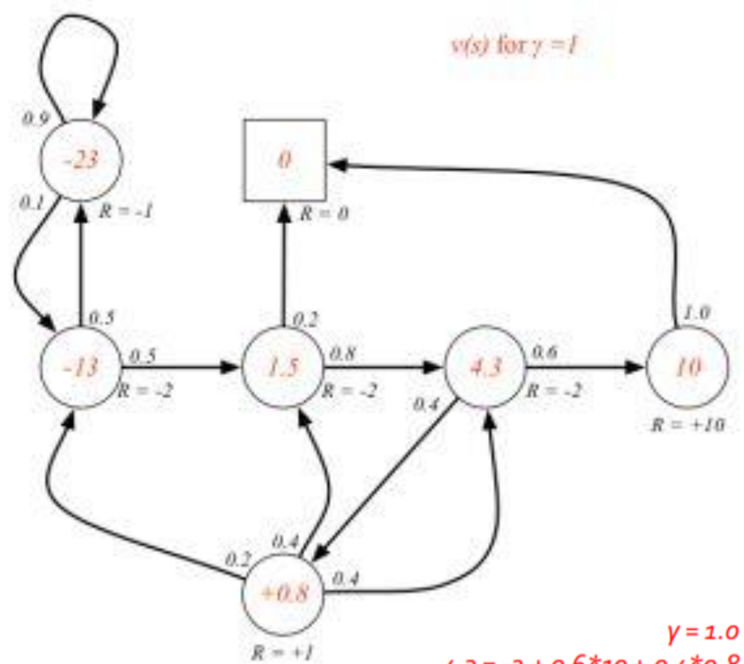
$$v_\pi(s) = \sum_{a \in A} \pi(a|s)\left(R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_\pi(s')\right)$$

- 各状态圈内的数字表示该状态的价值，圈外的R表示的是该状态的即时奖励。

γ=0.9



γ=1



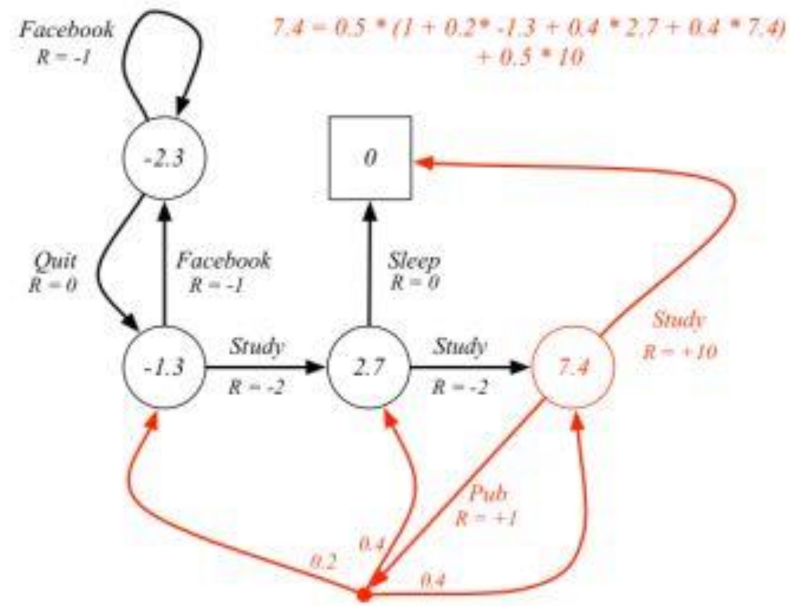
γ = 1.0
4.3 = -2 + 0.6*10 + 0.4*0.8

小象学院 ChinaHadoop.cn

值函数的计算

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

- 学生MDP中，红色圆圈的状态价值的计算：



- 学生Agent遵循的策略是随机策略，即所有可能的行为有相同的几率被选择执行。
- $\pi(a|s) = 0.5, \gamma = 1$
- $7.4 = 0.5(1 + 0.2 * (-1.3) + 0.4 * 2.7 + 0.4 * 7.4) + 0.5 * 10$

最优价值函数

- **最优状态值函数**
  最优值函数$v_*(s)$ 是在从所有策略产生的状态价值函数中，选取使状态s价值最大的函数：

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

- **最优状态行为值函数**
  最优状态行为值函数$q_*(s,a)$是从所有策略产生的行为价值函数中，选取是状态行为对<s,a>价值最大的函数:

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

贝尔曼最优方程

- 根据之前介绍的Bellman公式，可以分别得到最优状态值函数和最优状态-行动值函数的贝尔曼最优方程。

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

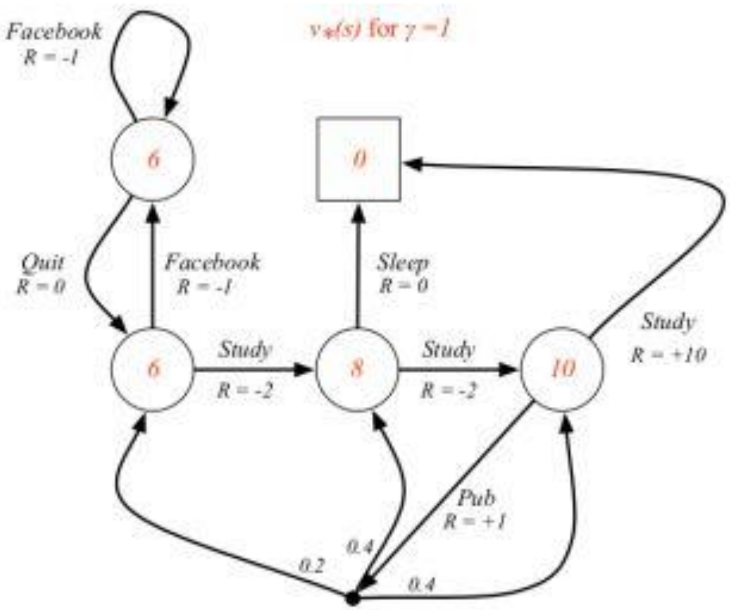
$$q^*(s,a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \max_{a'} q^*(s',a')$$

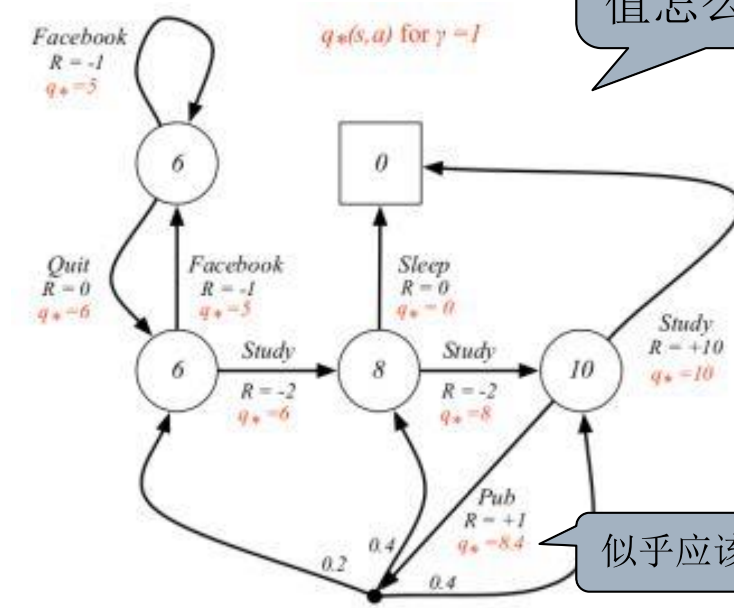
- 有了贝尔曼方程和贝尔曼最优方程后，就可以用动态规划等方法来求解MDP了。

最优状态价值与最优状态行为价值

- 学生MDP问题的最优状态价值和最优状态行为价值：

这里的状态值怎么算的？





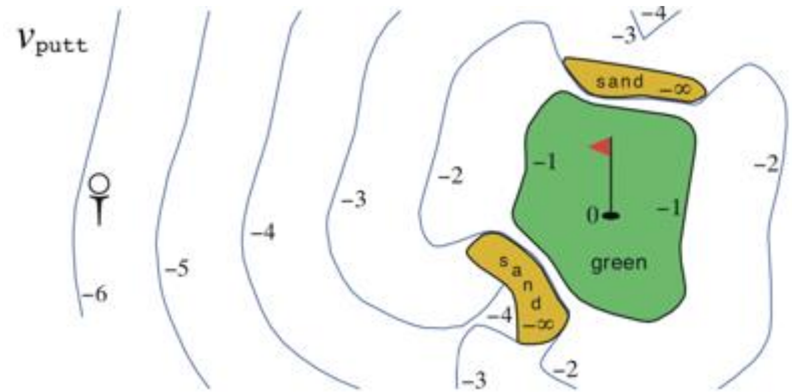
似乎应该是9.4

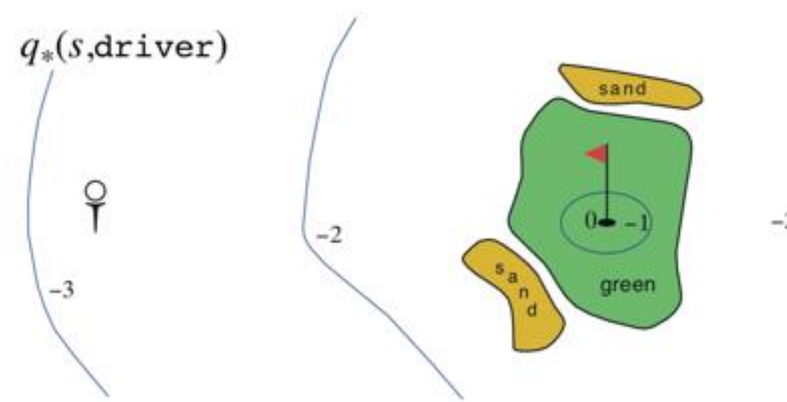
$$q_\pi(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_\pi(s')$$

- 1+1*(0.2*6+0.4*8+0.4*10) = 9.4

最优值函数的作用

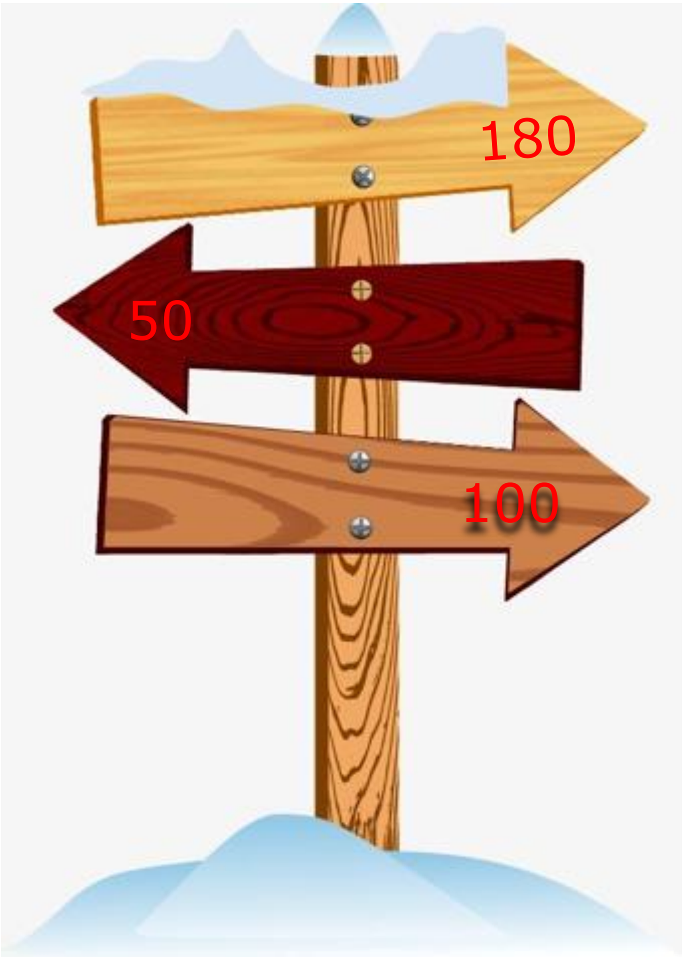
- 最优状态值函数的作用在于决定策略，有了值函数之后，策略的制定就很简单：
- 利用“贪心法”即可，每一个状态的行动策略都是往分值大的地方去。
- 最优状态行为值函数则更简单，选状态行为值函数大的动作即可。





最优值函数的作用

- 最优状态值函数
  - 跳房子
- 最优状态行为值函数
  - 指路牌



最优策略

- 定义策略之间的偏序关系$\pi \geq \pi'$，
  - if $v_{\pi}(s) \geq v_{\pi'}(s), \forall s$
- 对任意MDP：
  - 存在最优策略π_*，满足$\pi_* \geq \pi, \forall \pi$。
  - 所有最优策略的状态值函数都等于最优状态值函数$v_{\pi*}(s) = v_*(s)$
  - 所有的最优策略的状态行为值函数都等于最优状态行为值函数$q_{\pi*}(s,a) = q_*(s,a)$

寻找最优策略

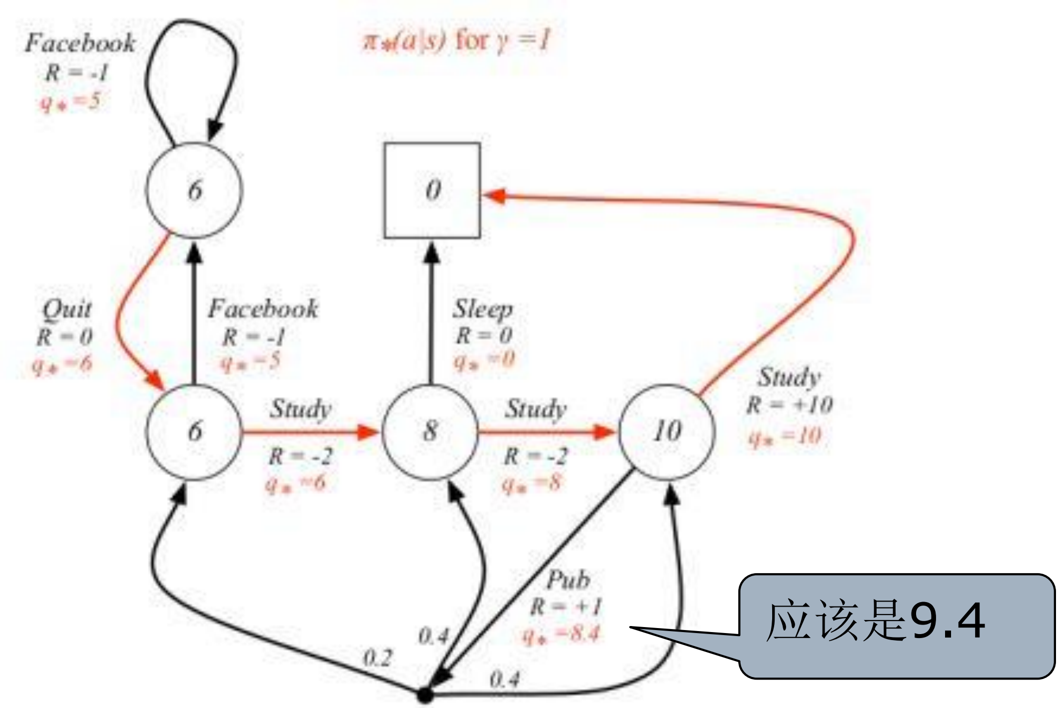
- 可以通过最大化最优行为价值函数来找到最优策略：

$$\pi_*(a|s) = \begin{cases} 1 & \text{if } a = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, q_*(s,a) \\ 0 & \textit{otherwise} \end{cases}$$

- 如前一页所述，对于任何MDP问题，总存在一个确定性的最优策略。
- 同时如果知道最优行为价值函数，则表明我们就可以找到最优策略。

最优策略示意

- 学生MDP问题的最优策略

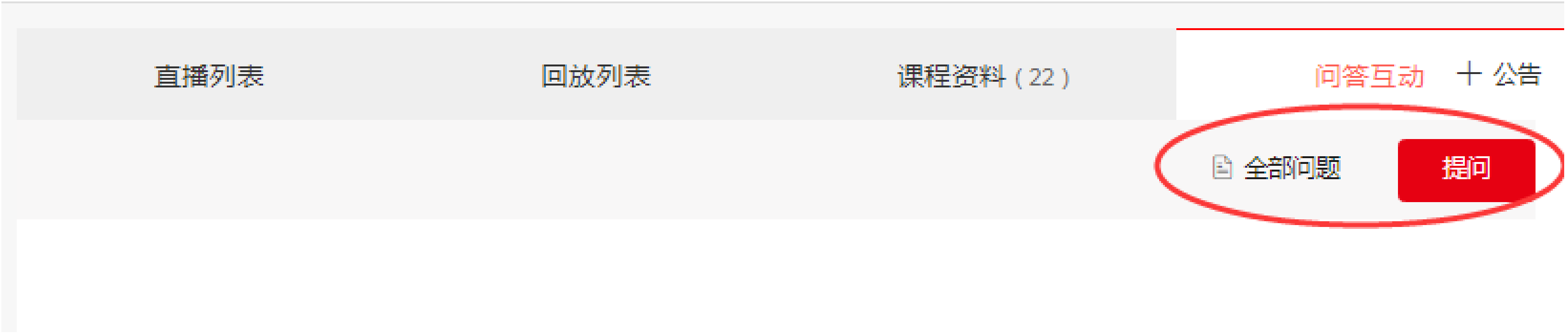


参考资料

- https://blog.csdn.net/zz_1215/article/details/44138823
- https://blog.csdn.net/VictoriaW/article/details/78839929
- https://zhuanlan.zhihu.com/p/28084942
- https://blog.csdn.net/u013745804/article/details/78206794

问答互动

在所报课的课程页面，
1、点击“全部问题”显示本课程所有学员提问的问题。
2、点击“提问”即可向该课程的老师和助教提问问题。



联系我们

小象学院：互联网新技术在线教育领航者

- 微信公众号：**小象学院**