第二讲 马尔可夫决策过程

这讲将介绍马尔可夫决策过程（Markov decision processes, MDP）。马尔可夫决策过程是顺序决策问题的经典数学描述，因此，MDP也是强化学习问题的数学描述和理论基础。在强化学习中，**马尔可夫决策过程是对完全可观测环境的描述**。几乎所有的强化学习问题都可以构造成马尔可夫决策过程。如，最优化控制主要是处理连续的马尔可夫过程；部分可观测问题也可以转换成马尔可夫决策问题。

本讲从介绍马尔可夫过程（Markov process，MP）开始，接着介绍马尔可夫奖励过程（Markov reward process，MRP），最后再介绍马尔可夫决策过程及其扩展。

2.1 马尔可夫过程（Markov process）

2.1.1 马尔可夫性质（Markov property）

通俗地说，马尔可夫性质是指，未来只与当前状态有关，与过去无关。其数学定义为：

$$P[S_{t+1}|S_t] = P[S_{t+1}|S_1, \dots, S_t]$$

从上式可知，当前状态包含了历史中的所有相关信息，因此，只要知道了当前状态，所有的历史信息就不再需要了。当前状态是下一个状态的充分统计数据。例如，等红绿灯时，假设你每一秒看一次灯，有了当前这一秒看到红绿灯的状态，就不需要前面看到的所有状态了，根据当前看到的状态，决定下一秒你是否继续等待。

对于一个马尔可夫状态s及其后续状态s'，**状态转移概率** 定义为：

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

状态转移概率矩阵 $\boldsymbol{P}$ 定义为从所有的状态s到所有的后续状态s'的转移概率，

$$\boldsymbol{P} = \begin{bmatrix} P_{11} & P_{12} & \dots & P_{1n} \\ P_{21} & P_{22} & \dots & P_{2n} \\ \dots & & & \\ \dots & & & \\ P_{n1} & P_{n2} & \dots & P_{nn} \end{bmatrix}$$

矩阵中行号表示当前状态s，列号表示到达的后续状态s'。每一行的和为1。

2.1.2 马尔可夫过程（Markov process）

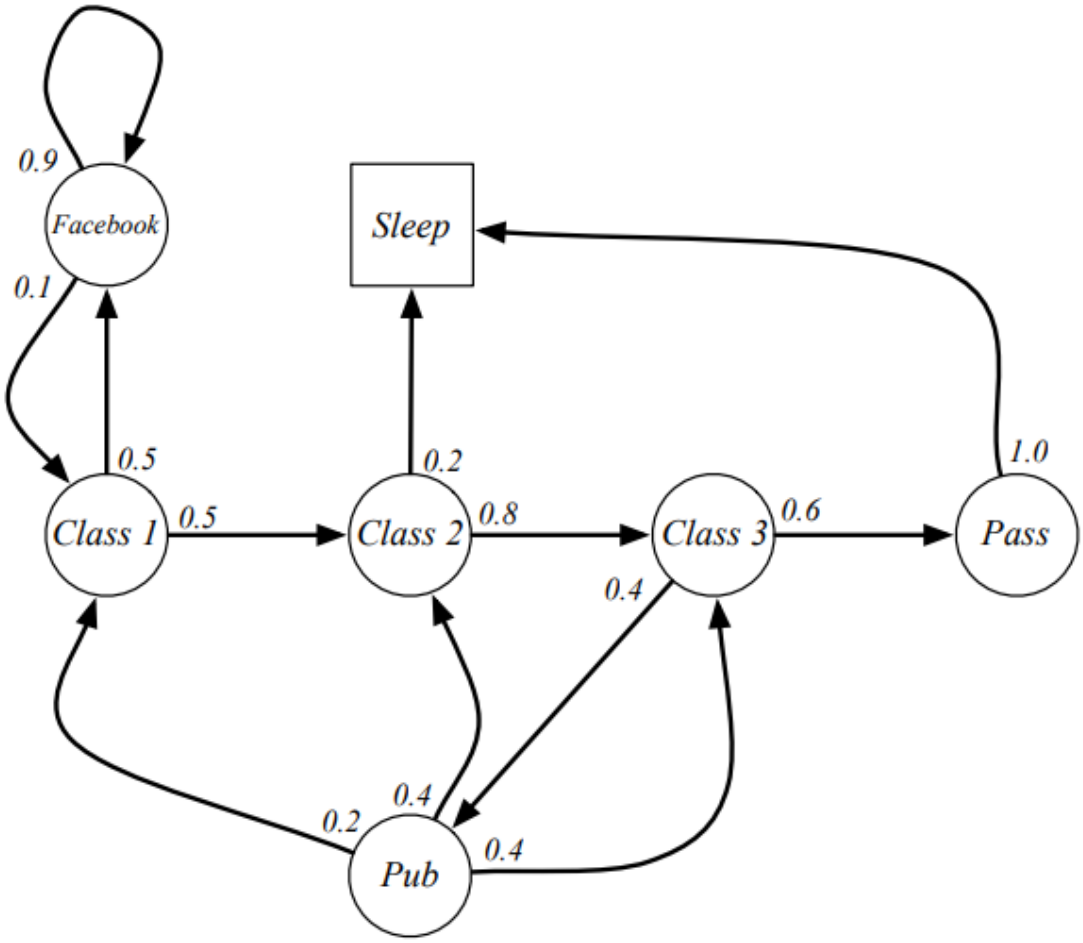
通俗地说，马尔科夫过程是一个无记忆的随机过程。例如，如具有马尔可夫性质的随机状态序列$S_1, S_2, \dots$。其定义如下：

马尔科夫过程又称为马尔科夫链(Markov Chain)，可以用一个元组<S,P>表示，其中

- S是有限数量的状态集
- P是状态转移概率矩阵

> 注：
> 1.任何状态只依赖于前一个状态（马尔可夫性质）
> 2.转移概率不会随时间变化

例2-1 学生马尔科夫链



上图中，圆圈表示学生所处的状态，方格Sleep是一个终止状态，终止状态也可以理解成以100%的概率转移到自己，所以一旦进入，就停留在此。箭头表示状态之间的转移，箭头上的数字表示状态转移概率。

例如，当学生处在第一节课（Class1）时，有50%的概率会上第2节课（Class2）；同时也有50%的概率不认真听课，进入到浏览facebook这个状态中。在刷facebook这个状态时，比较容易沉迷，有90%的概率在下一时刻继续浏览，有10%的概率返回到第一节课。当学生进入到第二节课（Class2）时，会有80%的概率继续参加第三节课（Class3），有20%的概率觉得课程较难而退出（Sleep）。当学生处于第三节课这个状态时，有60%的概率通过考试，继而100%的退出该课程，也有40%的概率去泡吧。泡吧之后，又分别有20%、40%、40%的概率返回值第一、二、三节课重新继续学习。

假设学生马尔科夫链从状态Class1开始，最终结束于Sleep。其间的过程根据状态转化图可以有很多种可

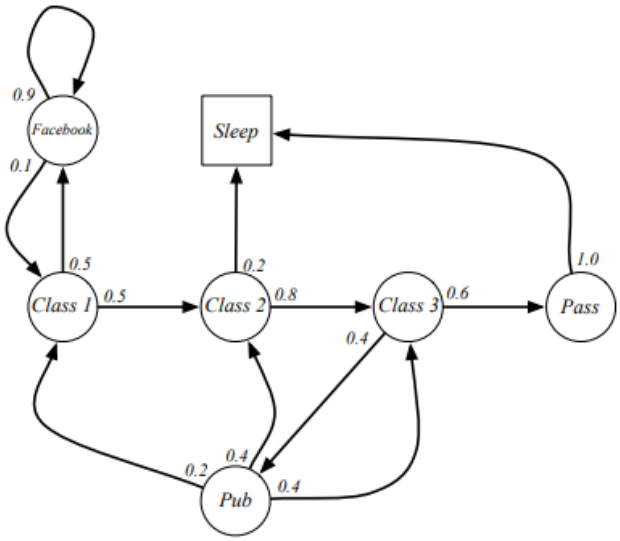
能性，这些都称为采样片段Sample Episodes。以下四个Episodes都是可能的：
C1 - C2 - C3 - Pass - Sleep

C1 - FB - FB - C1 - C2 - Sleep

C1 - C2 - C3 - Pub - C2 - C3 - Pass - Sleep

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

学生马尔科夫链的转移矩阵如下图



$$\mathcal{P} = \begin{array}{c} \\ C1 \\ C2 \\ C3 \\ Pass \\ Pub \\ FB \\ Sleep \end{array} \begin{array}{c} \begin{array}{ccccccc} C1 & C2 & C3 & Pass & Pub & FB & Sleep \end{array} \\ \begin{bmatrix} & 0.5 & & & & 0.5 & \\ & & 0.8 & & & & 0.2 \\ & & & 0.6 & 0.4 & & \\ & & & & & & 1.0 \\ 0.2 & 0.4 & 0.4 & & & & \\ 0.1 & & & & & 0.9 & \\ & & & & & & 1 \end{bmatrix} \end{array}$$

从转移矩阵中可以看到，每一行的和为1，即从某一当前状态出发，到其所有后续状态的转移概率之和为1。

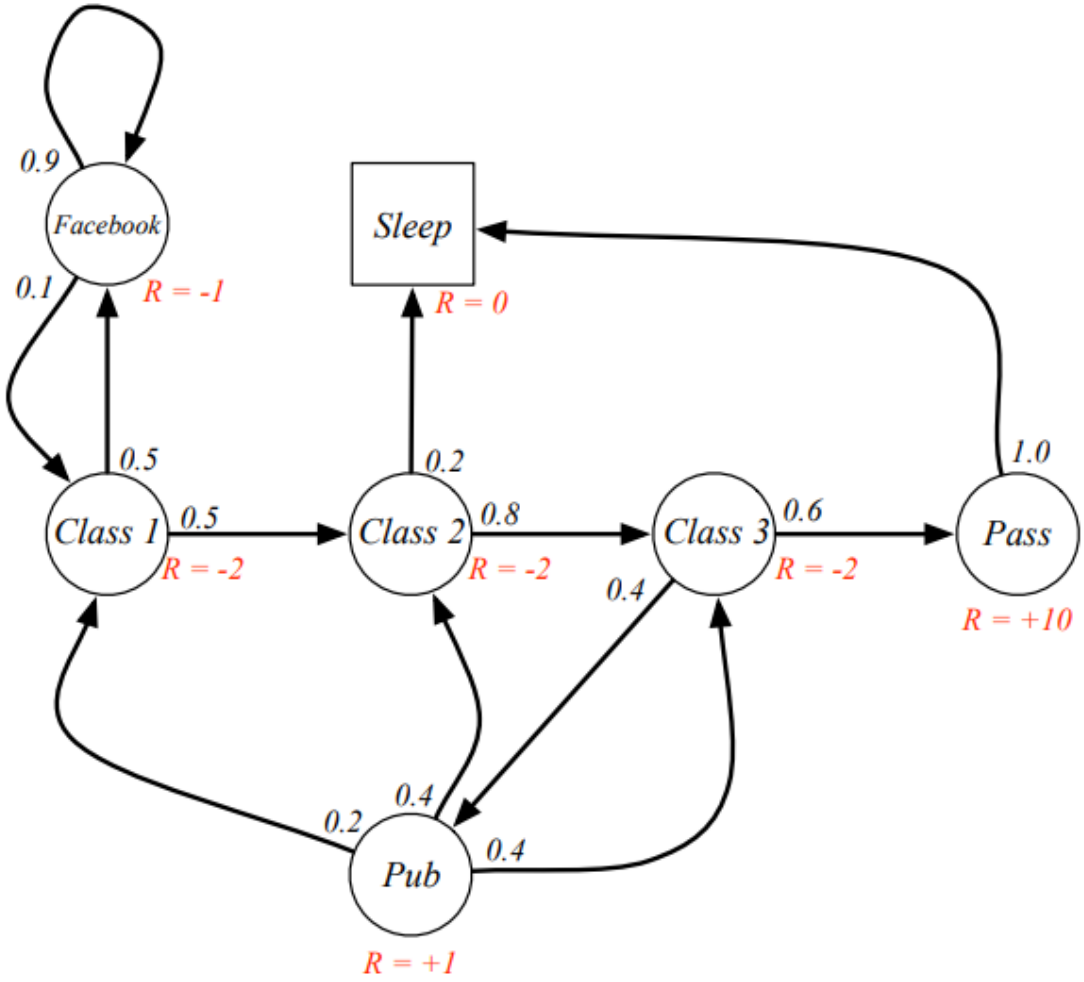
2.2 马尔可夫奖励过程（Markov reward process）

马尔科夫奖励过程在马尔科夫过程的基础上增加了**奖励R和折扣系数**γ 。其定义为：
马尔科夫奖励过程是一个元组$< S, P, R, \gamma >$，其中

- S是有限数量的状态集
- P是状态转移概率矩阵
- **R是一个奖励函数，**$R_s = E[R_{t+1}|S_t = s]$
- γ**是一个折扣因子，**$\gamma \in$ **[0, 1]**

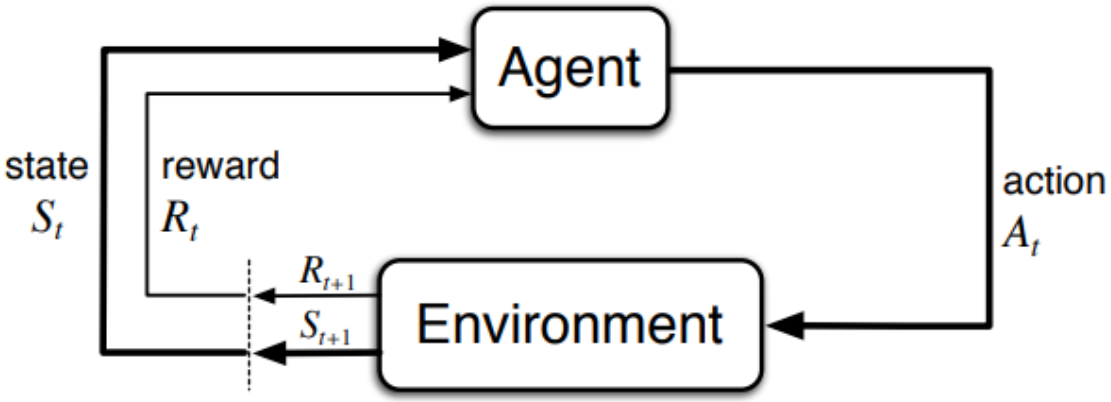
下图是一个马尔可夫奖励过程的示例，在**马尔科夫过程**的基础上针对每一个状态增加了相应的奖励，这里没有给定折扣因子，在实际应用中，可以根据具体问题，选择一个折扣因子。

例2-2 学生马尔可夫奖励过程



2.2.1 奖励函数

R是一个奖励函数。在当前状态S_t下,执行动作a_t,进入到下一个时刻(t+1)能获得的奖励期望。R约定为离开该状态获得的奖励，而不是进入该状态获得的奖励，即S_t对应奖励为R_{t+1}。这个约定主要原因：因为状态是从S_0 开始的，智能体未采取动作之前，环境有一个初始状态。S_0的奖励为离开S_0的奖励，离开S_0后进入下一个状态。如下图所示，离开状态S_t，环境更新状态，进入到t+1时刻。

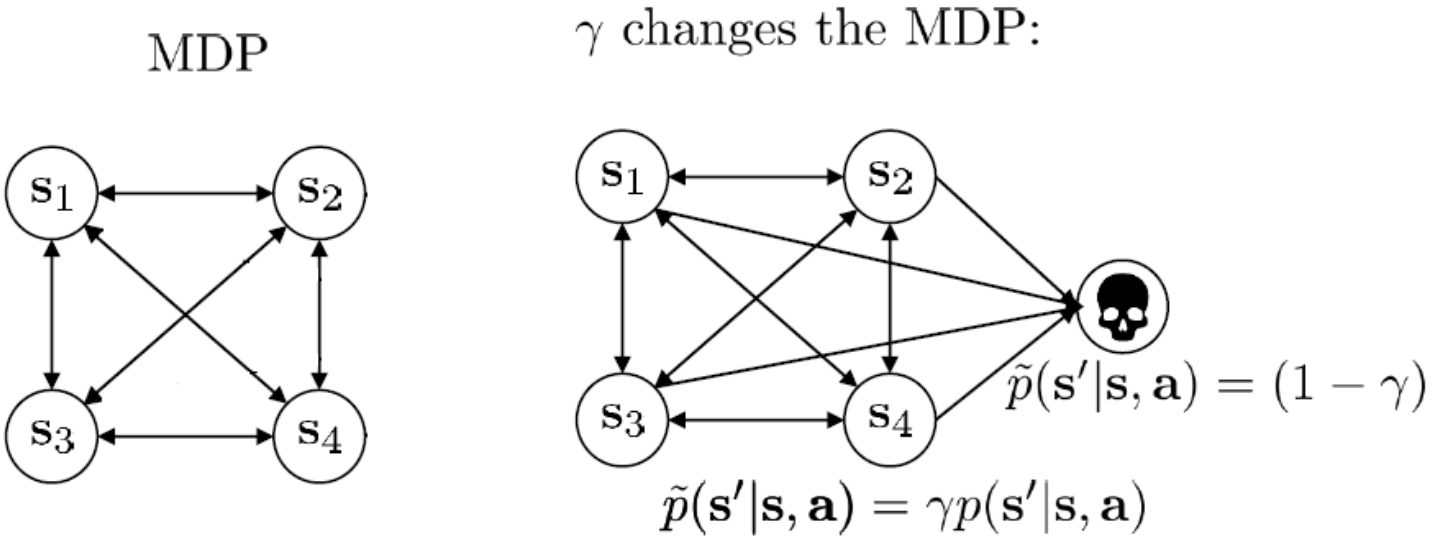


2.2.2 折扣因子（Discount factor）

在大部分马尔可夫奖励过程和马尔可夫决策过程中，都引入折扣因子$\gamma \in [0,1]$，主要原因如下：

- 首先，数学上的方便表示，在带环的马尔可夫过程中，可以避免陷入无限循环，达到收敛。
- 其次，随着时间的推移，远期利益的不确定性越来越大，符合人类对于眼前利益的追求。
- 再次，在金融上，近期的奖励可能比远期的奖励获得更多的利息，越早获得奖励，收益越高。

在另一门课（CS285）中，Levine给了一个很有意思的角度，下图左边的MDP，可以在四个状态中任意转移，构成无限MDP。在处理这个MDP问题时，为了避免无限循环，所以引入折扣因子。如果你想要找到另一个不同的MDP，不需要折扣因子，却能得到像左边带折扣因子的MDP一样的效果。这个不带折扣因子的MDP，可以通过左边带折扣因子的MDP引入一个额外的事件。这个事件需要能够描述：比起一百万年后中五百万彩票，你更希望现在中五百万。这是因为人生充满未知，你现在活着的概率比以后活着的概率大，所以人类追求眼前利益归根结底是对死亡的恐惧。因此我们对智能引入对死亡的恐惧，即增加一个死亡状态。因此，得到了右边的MDP。



我们可以这样来理解右边这个MDP，智能体是害怕死亡的，因此不想进入到死亡状态。假设S1，S2, S3, S4在每一个时间步都有（$1-\gamma$）的概率进入死亡，如果智能体死了，就不再获得奖励了。因此，智能体在S1，S2, S3, S4之间的转移概率为γ，这与左边带折扣因子的MDP是等价的。

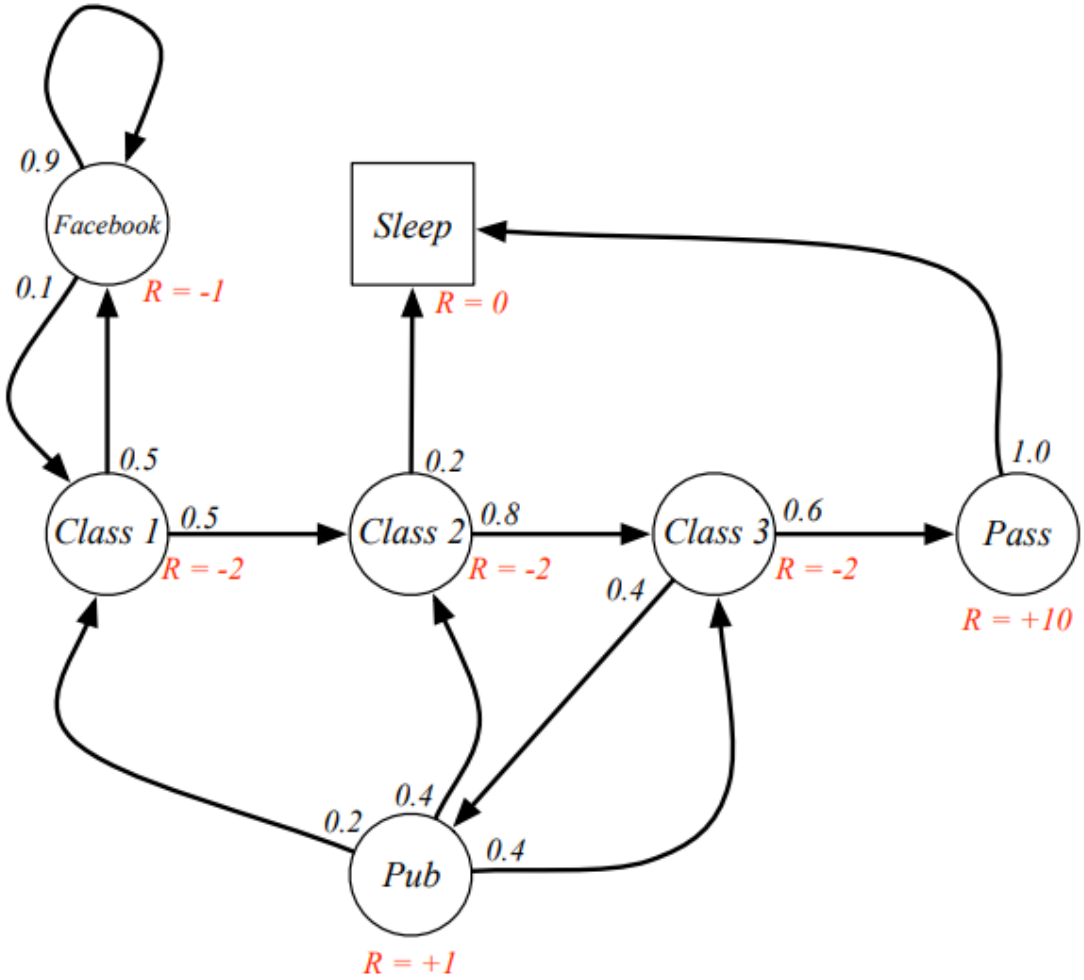
2.2.3 回报（Return）

回报（或译为“收益”）G_t为从时间步t开始带折扣的奖励总和。公式如下：

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

其中折扣因子γ体现了未来的奖励在当前时刻的价值比例（即G_t中的加权系数,权重即比例）。从t时刻起，经历了k+1时间步后，获得的奖励R_{t+k+1}在t时刻的体现出的价值是$\gamma^k R_{t+k+1}$，γ接近0，则表明智能体趋向于眼前的利益；γ接近1则表明智能体偏重考虑长远的利益。

例2-3，举例说明收获的计算，马尔可夫奖励过程如下图，



各状态对应的奖励如下表：

状态	C1	C2	C3	Pass	Pub	FB	Sleep
奖励	-2	-2	-2	10	1	-1	0

从马尔可夫奖励过程中采样如下4个马尔科夫链，计算当γ= 1/2时，在t=1时刻（$S_1 = C_1$）时状态S_1的采样收获分别为：

马尔科夫链	收获（G_t）
C1 C2 C3 Pass Sleep	$G_t = -2 - 2 * 1/2 - 2 * 1/4 + 10 * 1/8 = -2.25$

马尔科夫链	收获（G_t)
C1 FB FB C1 C2 Sleep	$G_t = -2 - 1 * 1/2 - 1 * 1/4 - 2 * 1/8 - 2 * 1/16 = -3.125$
C1 C2 C3 Pub C2 C3 Pass Sleep	$G_t = -2 - 1 * 1/2 - 2 * 1/4 + 1 * 1/8 - 2 * 1/16 \ldots = -3.141$
C1 FB FB C1 C2 C3 Pub C1 FB FB FB C1 C2 C3 Pub C2 Sleep	$G_t = -2 - 1 * 1/2 - 1 * 1/4 - 2 * 1/8 - 2 * 1/16 \ldots = -3.20$

从上面例子可以看出，当$\gamma = 0$时，马尔科奖励过程的回报等于状态的奖励（上表中都等于C1的奖励）。

2.2.4 价值函数（Value Function）

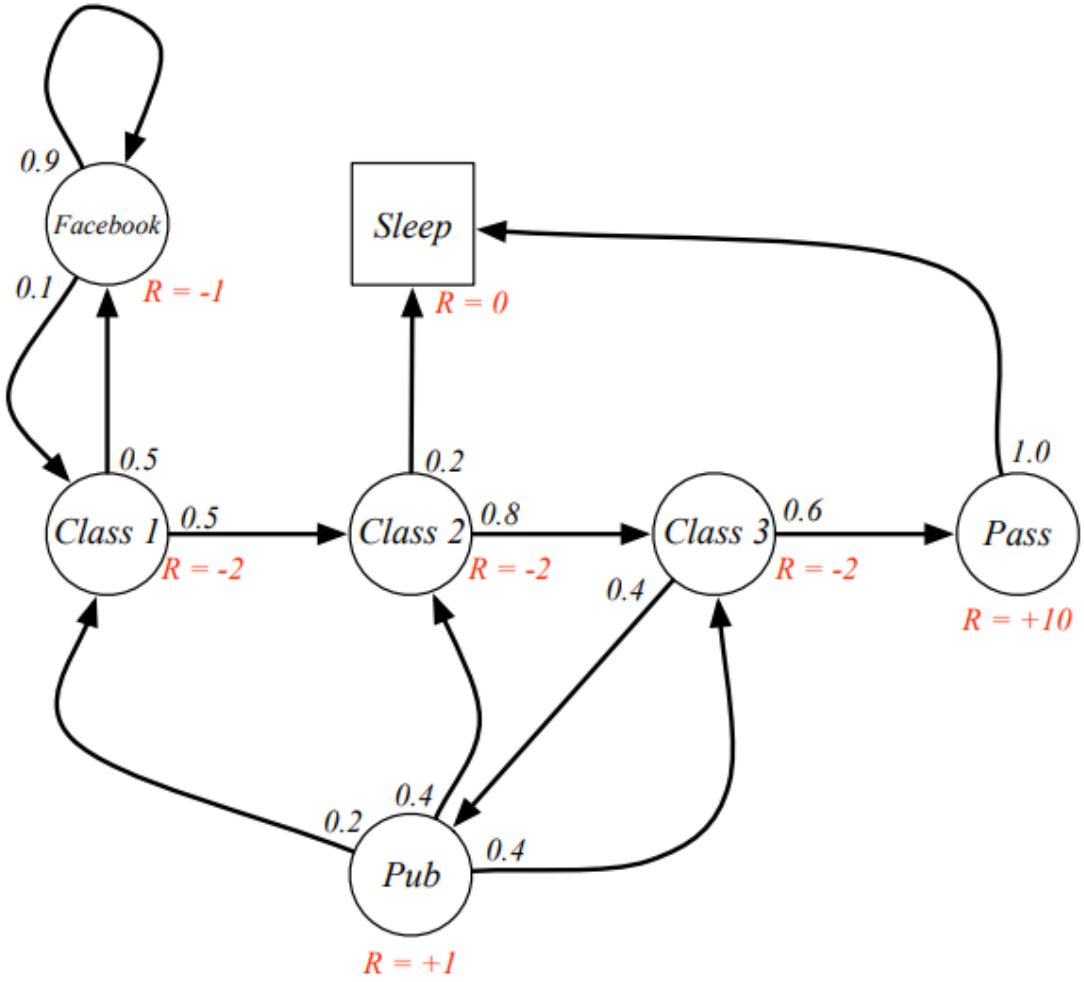
状态价值函数（state-value function）的定义：马尔可奖励过程的状态价值函数$V(s)$是从该状态s开始的**回报的期望**，即

$$v(s) = \mathbb{E}[G_t|S_t = s]$$

价值函数$V(s)$给出了某一状态的长期价值。

> 注：价值函数其实包括状态价值函数V(s)和动作价值函数Q(s,a),也分别称为V值和Q值，如果没有特别强调，价值函数一般指状态价值函数V(s)，即V值。Q值将在本讲后面介绍。

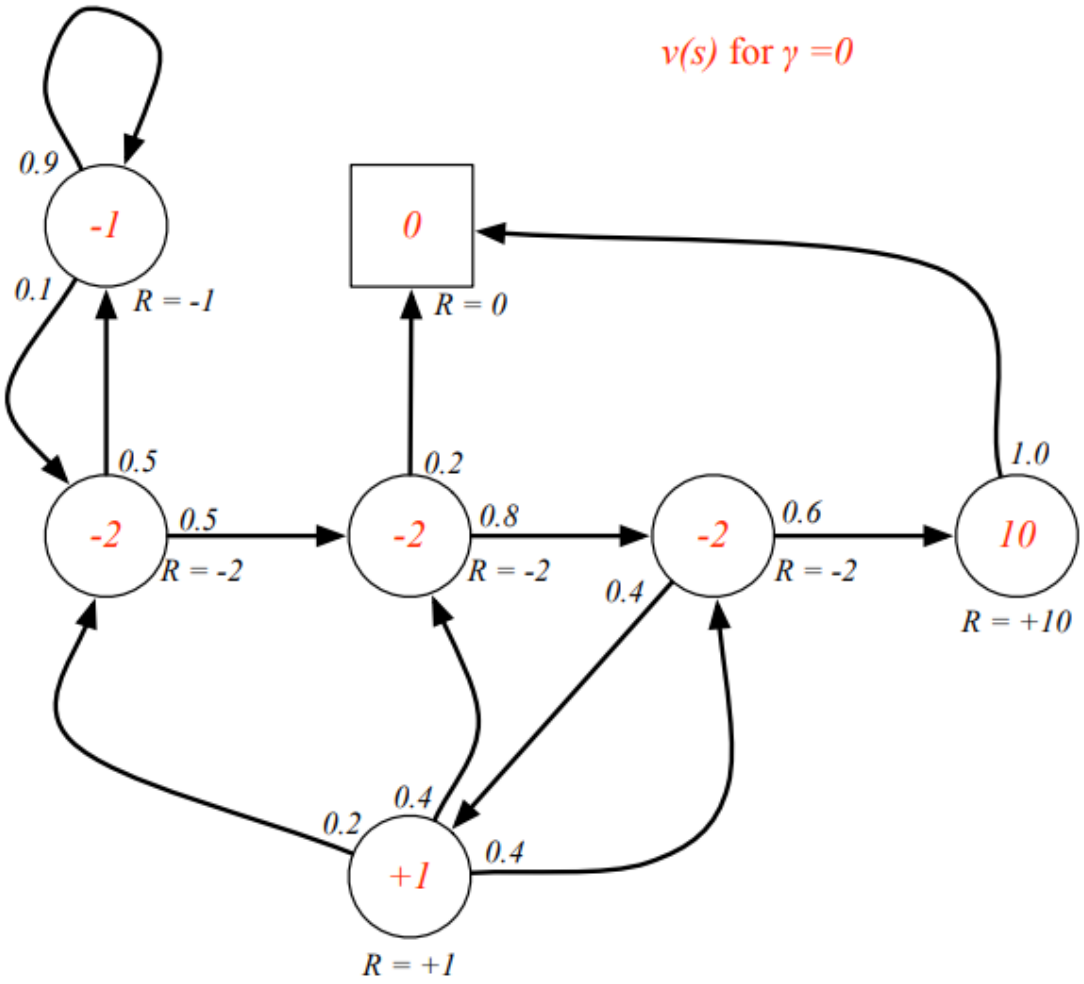
例2-4，举例说明价值的计算，马尔可夫奖励过程如下图，



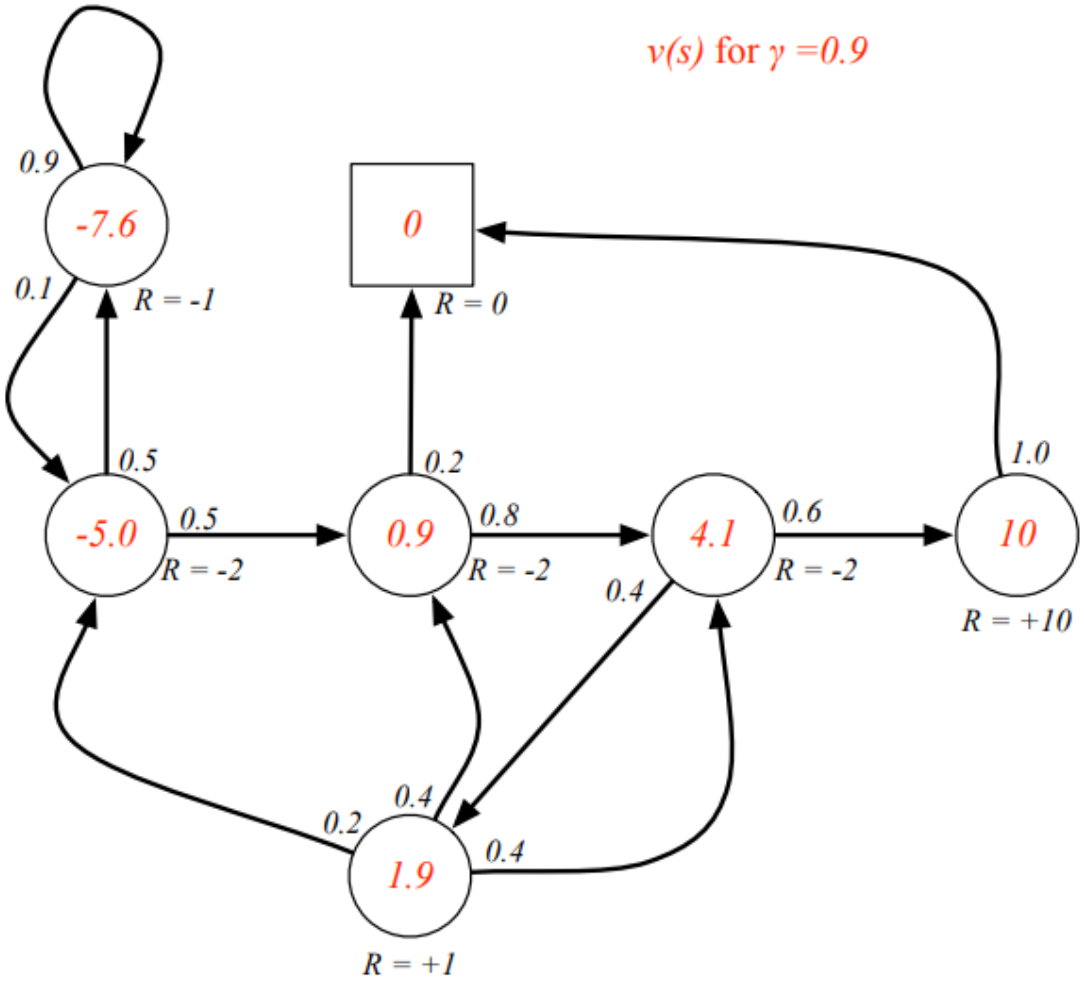
从例2-3中可知，当$\gamma = 0$时，各状态回报就等于该状态的奖励，因此各状态的价值等于该状态的即时奖励。当$\gamma \neq 0$时,各状态的价值需要通过价值函数的定义计算得到，这里给出γ分别为0, 0.9,和1三种情况下各状态的价值，如下图所示。

各状态圈内的数字表示该状态的价值，圈外的R=-2等表示的是该状态的即时奖励。

当$\gamma = 0$时，V(s)如圈内数值所示:



当$\gamma = 0.9$时，V（=(s)如圈内数值所示:



现在计算上图中状态的价值函数，状态FB，Pub,C1,C2,C3,Pass,Sleep的价值函数的初始值，可以初始化为0，或者其对应的奖励（本部分的计算可以看完后面的贝尔曼方程后，再回头看，理解更加清晰）。
计算公式如下：
FB状态时，有0.9的概率进入FB,0.1的概率进入到C1，因此

$$V(FB') = 0.9 * [-1 + \gamma * V(FB)] + 0.1 * [-1 + \gamma * V(C1)]$$

Pub状态时，有0.2的概率进入C1,0.4的概率进入到C2，0.4概率进入C3，因此

$$V(Pub') = 0.2 * [1 + \gamma * V(C1)] + 0.4 * [1 + \gamma * V\ \ (C2)] + 0.4 * [1 + \gamma * V(C3)]$$

同理，C1，C2，C3, Pass状态的价值函数分别为

$$V(C1') = 0.5 * [-2 + \gamma * V(C2)] + 0.5 * [-2 + \gamma * V(FB)]$$

$$V(C2') = 0.2 * [-2 + \gamma * V(Sleep)] + 0.8 * [-2 + \gamma * V(C3)]$$

$$V(C3') = 0.6 * [-2 + \gamma * V(Pass)] + 0.4 * [-2 + \gamma * V(pub)]$$

$$V(Pass) = 1 * (10 + \gamma * V(Sleep))$$

现在根据上面公式，进行迭代计算V值。各状态的初始V值都设0，算出各状态的后续状态的V(S')值，本次迭代算出来的V(S')值，作为下一次迭代各状态的当前状态的V(S），在下一次迭代中，根据V（S)的值（即上次迭代的V（S'）的值）继续更新各状态的后续状态的V(S')值，不断迭代，直到V值趋于稳定。

1.第一步迭代

V(FB),V(Pub),V(C1),V(C2),V(C3),V(Pass)都为0

$$V(FB') = 0.9 * [-1 + 0.9 * 0] + 0.1 * [-1 + 0.9 * 0] = -1$$

$$V(Pub') = 0.2 * [1 + 0.9 * 0] + 0.4 * [1 + 0.9 * 0] + 0.4 * [1 + 0.9 * 0] = 1$$

$$V(C1') = 0.5 * [-2 + 0.9 * 0] + 0.5 * [-2 + 0.9 * 0] = -2$$

$$V(C2') = 0.2 * [-2 + 0.9 * 0] + 0.8 * [-2 + 0.9 * 0] = -2$$

$$V(C3') = 0.6 * [-2 + 0.9 * 0] + 0.4 * [-2 + 0.9 * 0] = -2$$

$$V(Pass) = 1 * (10 + 0.9 * 0) = 10$$

2.第二步迭代

V(FB)=-1, V(Pub)=1, V(C1)=-2, V(C2)=-2, V(C3)=-2, V(Pass)=10

$$V(FB') = 0.9 * [-1 + 0.9 * -1] + 0.1 * [-2 + 0.9 * -2] = -1.99$$

$$V(Pub') = 0.2 * [1 + 0.9 * -2] + 0.4 * [1 + 0.9 * -2] + 0.4 * [1 + 0.9 * -2] = -0.8$$

$$V(C1') = 0.5 * [-2 + 0.9 * -1] + 0.5 * [-2 + 0.9 * -2] = -3.35$$

$$V(C2') = 0.2 * [-2 + 0.9 * 0] + 0.8 * [-2 + 0.9 * -2] = -3.44$$

$$V(C3') = 0.6 * [-2 + 0.9 * 10] + 0.4 * [-2 + 0.9 * 1] = -1.64$$

$$V(Pass) = 1 * (10 + 0.9 * 0) = 10$$

3.第三步迭代

V(FB)=-1.99, V(Pub)=-0.8, V(C1)=-3.35, V(C2)=-3.44, V(C3)=-1.64, V(Pass)=10

. . .

当迭代次数到30次左右时，各状态的值趋于稳定（即迭代次数增加只有微小的变化），如上图中圆圈内的数值。其python代码如下，可以改变迭代次数n来观察值的更新。

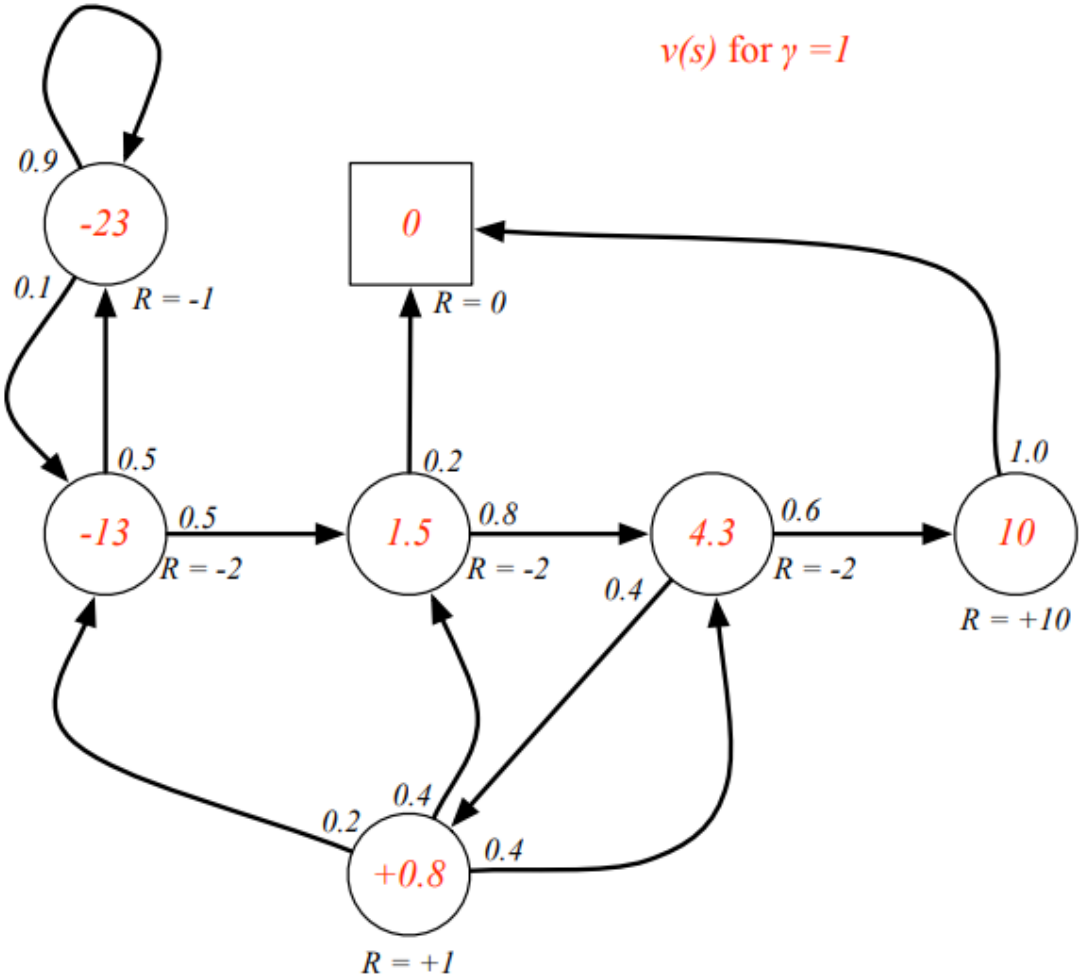
```
//initialization
V_fb = 0;
V_pub =0;
V_c1 = 0;
V_c2 = 0;
V_c3 = 0;
V_pass = 0;
V_slp = 0;
gamma =0.9;

//Iteratively computing Value function
for n in  range(1,30):
    V_fb_next  = 0.9*(-1+gamma*V_fb) + 0.1*(-1+gamma*V_c1)
    V_pub_next = 0.2*(1+gamma*V_c1) + 0.4*(1+gamma*V_c2)+0.4*(1+gamma*V_c3)
    V_c1_next  = 0.5*(-2+gamma*V_c2) + 0.5*(-2+gamma*V_fb)
    V_c2_next  = 0.2*(-2+gamma*V_slp) + 0.8*(-2+gamma*V_c3)
    V_c3_next  = 0.6*(-2+gamma*V_pass) + 0.4*(-2+gamma*V_pub)
    v_pass_next = 1*(10 + 0.9*V_slp)

    V_fb = V_fb_next;
    V_pub = V_pub_next;
    V_c1 = V_c1_next;
    V_c2 = V_c2_next;
    V_c3 = V_c3_next;
    v_pass =v_pass_next

print("fb = ",V_fb);
print("pub=",V_pub);
print("V_c1 =",V_c1)
print("V_c2 =",V_c2)
print("V_c3 =",V_c3)
```

当$\gamma = 1$时，V(s)如圈内数值所示:



价值函数的计算非常重要，解决RL问题的途径之一就是通过计算价值函数，然后隐式地得到策略，如第一讲的走迷宫的例子中基于价值的方法。

2.2.5 MRP的贝尔曼方程（Bellman Equation for MRPs）

状态的价值函数可以分解成两部分，如下：

- 该状态的即时奖励R_{t+1}
- 带折扣的该状态的后续状态的价值函数
  推导过程如下：

$$
\begin{aligned}
v(s) &= \mathbb{E}\left[G_t \mid S_t = s\right] \\
&= \mathbb{E}\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots \mid S_t = s\right] \\
&= \mathbb{E}\left[R_{t+1} + \gamma(R_{t+2} + R_{t+3} + \ldots) \mid S_t = s\right] \\
&= \mathbb{E}\left[R_{t+1} + \gamma G_{t+1} \mid S_t = s\right] \\
&= \mathbb{E}\left[R_{t+1} + \gamma V(S_{t+1}) \mid S_t = s\right]
\end{aligned}
$$

上式推导过程比较简单，只是最后两个等式的推导过程中，将G_{t+1}变成了$v(S_{t+1})$，这是基于回报的期望等于回报的期望的期望，即$\mathbb{E}[\mathbb{E}[G_{t+1} \mid S_{t+1}]|s_t] = \mathbb{E}[G_{t+1}|s_t]$。现在来证明这个等式。在证明这个等式之

前，先给出两个定义。

> 定义：如果 X 和 Y 都是离散型随机变量，则条件期望（Conditional Expectation）$E[Y|X=x]$的定义如下式所示：
>
> $$\mathrm{E}[Y \mid X=x]=\sum_{y} yP(Y=y \mid X=x)$$

> 定义：全期望（Law of total expectation） 也被称为叠期望（law of iterated expectations(LIE)）如果X是随机变量，其期望为$E(X)$，Y为相同概率空间上的任意随机变量，则有
>
> $$E[X]=E[E[X|Y]]$$

> 全期望的特殊情况：如果 A_i 是样本空间的有限或可数的划分(partition)，则全期望公式可以写成如下形式：
>
> $$\mathrm{E}(X)=\sum_{i} \mathrm{E}\left(X \mid A_i\right) \mathrm{P}\left(A_i\right)$$

现在证明$E[V(S_{t+1})|s_t]=E[E[G_{t+1} \mid S_{t+1}]|s_t]=E[G_{t+1}|s_t]$,为了证明简洁，令$s=s_t, g'=G_{t+1}, s'=s_{t+1}$,则有

$$\begin{aligned}
\mathbb{E}\left[\mathbb{E}\left[G_{t+1} \mid s_{t+1}\right] \mid s_t\right] &= \mathbb{E}\left[\mathbb{E}\left[g' \mid s'\right] \mid s\right] \\
&= \mathbb{E}\left[\sum_{g'} g'\, p\left(g' \mid s'\right) \mid s\right] \\
&= \sum_{s'} \sum_{g'} g'\, p\left(g' \mid s', s\right) p\left(s' \mid s\right) \\
&= \sum_{s'} \sum_{g'} \frac{g' p\left(g' \mid s', s\right) p\left(s' \mid s\right) p(s)}{p(s)} \\
&= \sum_{s'} \sum_{g'} \frac{g' p\left(g' \mid s', s\right) p\left(s', s\right)}{p(s)} \\
&= \sum_{s'} \sum_{g'} \frac{g' p\left(g', s', s\right)}{p(s)} \\
&= \sum_{s'} \sum_{g'} g' p\left(g', s' \mid s\right) \\
&= \sum_{g'} \sum_{s'} g' p\left(g', s' \mid s\right) \\
&= \sum_{g'} g' p\left(g' \mid s\right) \\
&= \mathbb{E}\left[g' \mid s\right] = \mathbb{E}\left[G_{t+1} \mid s_t\right]
\end{aligned}$$

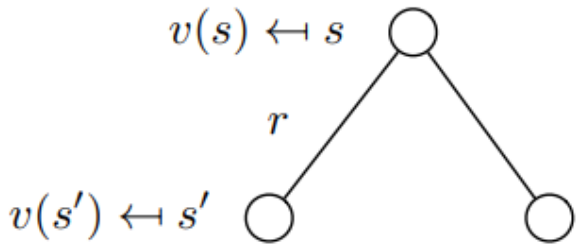
所以，MRP的贝尔曼方程为：

$$
\begin{aligned}
v(s) &= \mathbb{E}\left[R_{t+1} + \gamma V(S_{t+1}) \mid S_t = s\right] \\
&= \mathbb{E}\left[R_{t+1} \mid S_t = s\right] + \gamma \mathbb{E}[V(S_{t+1}) \mid S_t = s] \\
&= R(s) + \gamma \sum_{s_{t+1} \in S} p(s_{t+1}|s) V(s_{t+1})
\end{aligned}
$$

贝尔曼方程定义了当前状态s和下一个状态s_{t+1}的迭代关系，即当前状态的价值函数可以通过下一个状态的价值函数来计算。其实，在例2-4中的价值函数的迭代计算用的就是贝尔曼方程。

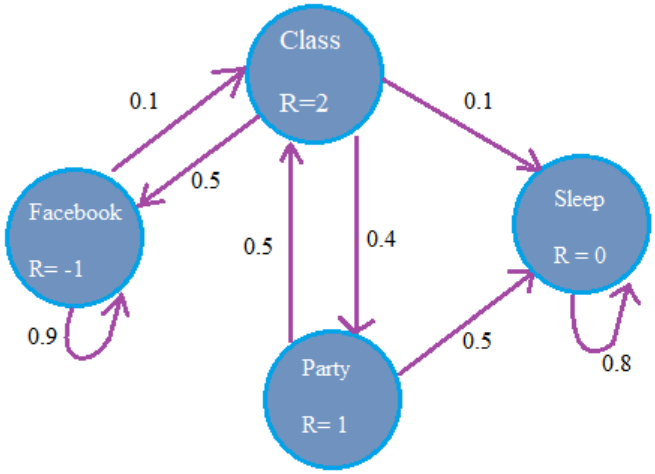
贝尔曼方程的图形解释如下图所示，图中圆圈表示当前状态s和所有可能的下一个状态s'，两个状态之间的连线表示离开当前状态的奖励 r。

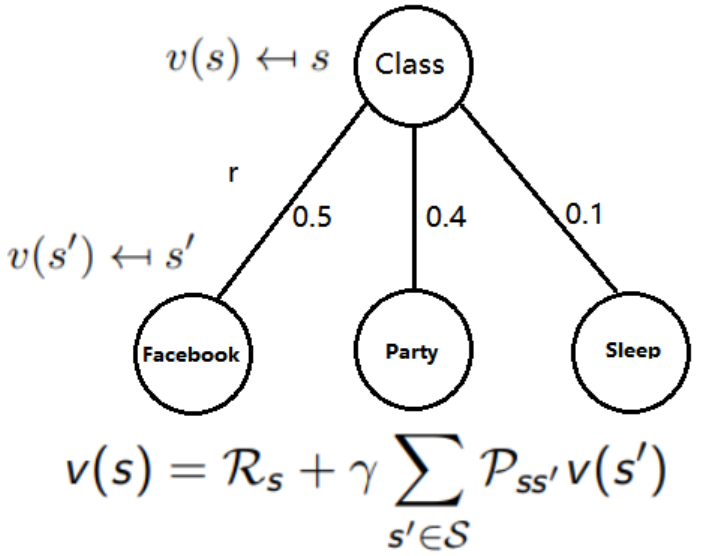
$$v(s) = \mathbb{E}\left[R_{t+1} + \gamma v(S_{t+1}) \mid S_t = s\right]$$



$$v(s) = \mathcal{R}_s + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'} v(s')$$

例2-5 利用Bellman方程计算马尔可夫奖励过程中的状态的价值如下图：





$$v(s) = \mathcal{R}_s + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'} v(s')$$

V(Class)= 2+ γ* [0.5*V(Facebook)+0.4*V(Party) + 0.1*V(Sleep)]

贝尔曼方程的矩阵形式

马尔可夫奖励过程的状态构成了状态空间，因此贝尔曼方程可以写成矩阵形式，

$$V = R + \gamma PV$$

其中，V是一个列向量，每一项表示一个状态。具体展开形式如下：

$$\begin{bmatrix} V(s_1) \\ V(s_2) \\ \vdots \\ V(s_N) \end{bmatrix} = \begin{bmatrix} R(s_1) \\ R(s_2) \\ \vdots \\ R(s_N) \end{bmatrix} + \gamma \begin{bmatrix} P(s_1 \mid s_1) & P(s_2 \mid s_1) & \dots & P(s_N \mid s_1) \\ P(s_1 \mid s_2) & P(s_2 \mid s_2) & \dots & P(s_N \mid s_2) \\ \vdots & \vdots & \ddots & \vdots \\ P(s_1 \mid s_N) & P(s_2 \mid s_N) & \dots & P(s_N \mid s_N) \end{bmatrix} \begin{bmatrix} V(s_1) \\ V(s_2) \\ \vdots \\ V(s_N) \end{bmatrix}$$

贝尔曼方程是一个线性方程组，因此可以直接得到解析解：

$$\begin{aligned} V &= R + \gamma PV \\ IV &= R + \gamma PV \\ (I - \gamma P)V &= R \\ V &= (I - \gamma P)^{-1} R \end{aligned}$$

从上式可知，可以通过矩阵逆运算直接求解方程，但矩阵求逆的复杂度为$O\ (n^3)$, n为状态数。因此，直接求解仅适用于状态空间规模小的MRP。状态空间规模大的MRP的求解通常使用迭代法。常用的迭代方法有：动态规划(Dynamic Programming)、蒙特卡洛评估(Monte-Carlo evaluation)、时序差分学(Temporal-Difference)等。我们将在后面学习这些方法。

2.3 马尔可夫决策过程（Markov decision process）

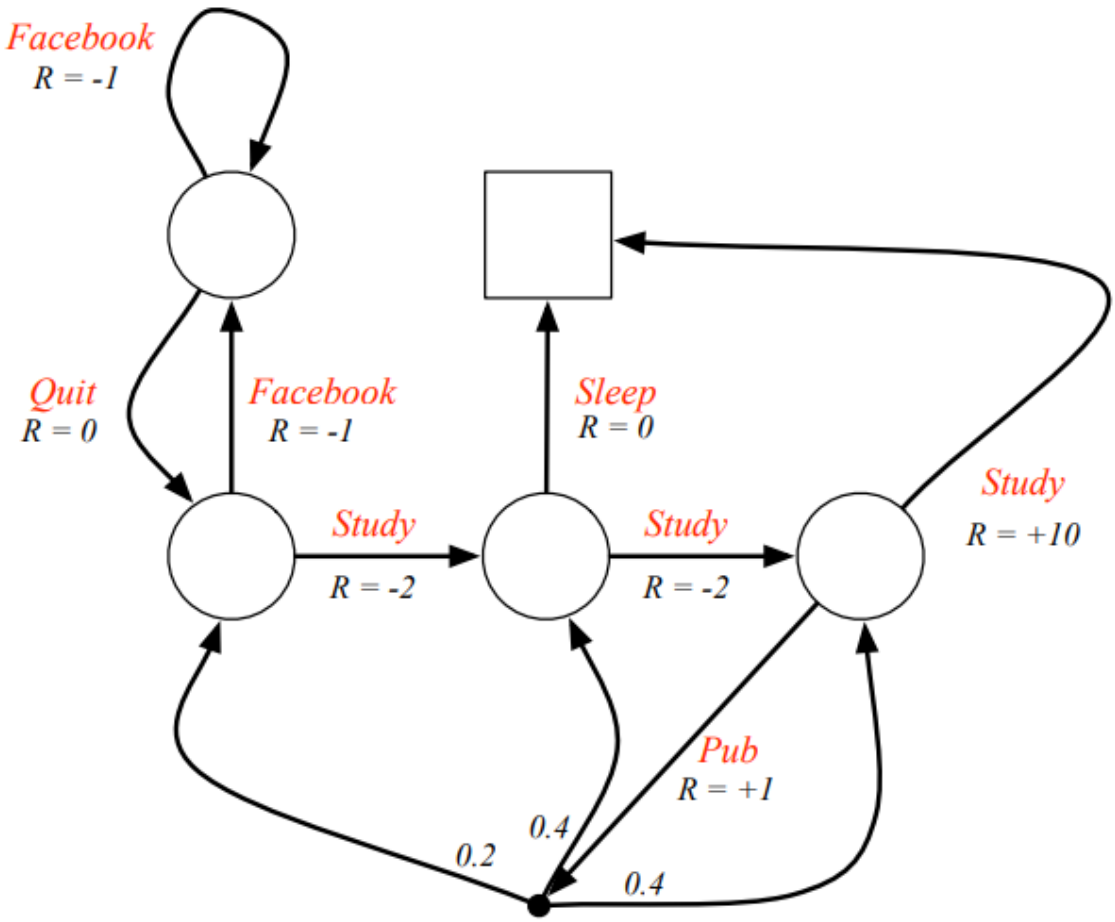
马尔可夫决策过程是在马尔可夫奖励过程的基础上加入了决策，即增加了动作。其定义为：
马尔科夫决策过程是一个元组$< S, A, P, R, \gamma >$，其中

- S是有限数量的状态集
- **A是有限数量的动作集**
- P是状态转移概率矩阵
  $P_{ss'}^{a} = \mathbb{P}[S_{t+1} = s' | S_t = s, A_t = a]$
- R是一个奖励函数，$R_s^a = E[R_{t+1} | S_t = s, A_t = a]$
- γ是一个折扣因子，$\gamma \in [0, 1]$

从上面定义可以看出，马尔可夫决策过程的**状态转移概率和奖励函数不仅取决于智能体当前状体，还取决于智能体选取的动作**。而马尔可夫奖励过程仅取决于当前状态。

例2-6 学生马尔可夫决策过程

图中红色的文字表示学生采取的动作，而不是MRP时的状态名。对比之前的学生MRP示例可以发现，即时奖励与动作有关了，同一个状态下采取不同的动作得到的即时奖励是不一样的。由于引入了动作，容易与状态名称混淆，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"去查阅文献"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，也就是说此时Agent没有选择权决定去哪一个状态。



2.3.1 策略（Policy）

定义：策略π是给定状态时，关于动作a的分布，即

$$\pi(a|s) = \mathbb{P}[A_t = a|S_t = s]$$

一个策略完整地定义了智能体的行为方式，即策略定义了智能体在各种状态下可能采取的动作，以及在各种状态下采取各种动作的概率。MDP的策略仅与当前的状态有关，与历史信息无关；同时某一确定的策略是静态的，与时间无关；但是个体可以随着时间更新策略。

当给定一个MDP: $M =< S, A, P, R, \gamma >$ 和一个策略π时，状态序列$S_1, S_2, \ldots$是一个马尔科夫过程$< S, P^{\pi} >$；同样，状态和奖励序列 $S_1, R_2, S_2, \ldots$是一个马尔科夫奖励过程 $< S, P^{\pi}, R^{\pi}, \gamma >$，并且在这个奖励过程中满足下面两个方程：

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

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

从上面两个方程可知，已知马尔可夫决策过程和策略π，可以把马尔可夫决策过程转换成马尔可夫奖励过程。在马尔可夫决策过程中，状态转移函数P(s'|s, a) 基于当前的状态以及当前的动作。而策略$\pi(a|s)$是给定状态时，关于动作a的分布,且已知。那么我们就可以通过策略求关于状态s的边缘分布，从而状态转移函数和奖励函数都变为只关于状态的函数。

2.3.2 价值函数（Value Function）

定义：MDP的状态价值函数（state-value Function）$v_{\pi}(s)$，是从状态s开始，执行策略所获得的收获的期望；或者说在执行当前策略π时，衡量智能体处在状态s时的价值大小。数学定义如下：

$$V_{\pi}(s) = \mathbb{E}_{\pi}[G_t|S_t = s]$$

定义: 行为价值函数(action-value function) $Q_{\pi}(s, a)$（又称为Q函数），是从状态s开始，采取动作a, 执行策略所获得的收获的期望；或者说在遵循当前策略π时，衡量对当前状态执行动作a的价值大小。行为价值函数一般都是与某一特定的状态相对应的，更精细的描述是状态行为对的价值函数。行为价值函数的数学定义如下：

$$Q_{\pi}(s, a) = \mathbb{E}_{\pi}[G_t|S_t = s, A_t = a]$$

V和q的关系

状态价值函数$v_{\pi}(s)$可以通过行为价值函数$Q_{\pi}(s, a)$来定义，状态价值函数定义为行为价值函数关于动作a的期望，数学描述为：

$$V_{\pi}(s) = \mathbb{E}_{a \sim \pi(a|s)}[Q_{\pi}(s, a)] = \sum_{a \in A} \pi(a|s) Q_{\pi}(s, a)$$

行为价值函数$Q_\pi(s,a)$描述的是状态s时，采取动作a的价值函数，是状态-动作对（s,a）的价值；而状态价值函数$v_\pi(s)$是在状态s时的价值函数。上式两者关系可以理解为，$v_\pi(s)$等于$Q_\pi(s,a)$，在状态s时，根据策略采取所有可能动作a的平均性能动作$\bar{a}$和状态s的Q值，即可以理解为$v_\pi(s)=Q_\pi(s,\bar{a})$,所以$v_\pi(s)$描述的是状态$s$下，平均性能动作的价值，因此，我们只关注状态。

2.3.3 贝尔曼期望方程（Bellman Expectation Equation）

与MRP的价值函数的贝尔曼方程相似，MDP的状态价值函数和行为价值函数，可以分解为即时奖励加上带折扣的后续状态的价值函数或行为价值函数，如下：

$$V_\pi(s)=\mathbb{E}_\pi[R_{t+1}+\gamma V_\pi(S_{t+1})|S_t=s]$$
$$Q_\pi(s,a)=\mathbb{E}_\pi[R_{t+1}+\gamma q_\pi(S_{t+1},A_{t+1})|S_t=s,A_t=a]$$

V函数贝尔曼方程的推导见MRP部分，这里对Q函数贝尔曼方程进行推导：

$$\begin{aligned}
Q_\pi(s,a)&=\mathbb{E}\left[G_t\mid s_t=s,a_t=a\right]\\
&=\mathbb{E}\left[R_{t+1}+\gamma R_{t+2}+\gamma^2R_{t+3}+\ldots\mid s_t=s,a_t=a\right]\\
&=\mathbb{E}\left[R_{t+1}+\gamma(R_{t+2}+\gamma R_{t+3}+\gamma^2R_{t+4}+\ldots)\mid s_t=s,a_t=a\right]\\
&=\mathbb{E}\left[R_{t+1}|s_t=s,a_t=a\right]+\gamma\mathbb{E}[G_{t+1}|s_t=s,a_t=a]\\
&=R(s,a)+\gamma\mathbb{E}[V(s_{t+1})|s_t=s,a_t=a]\\
&=R(s,a)+\gamma\sum_{s'\in S}P\left(s'\mid s,a\right)V\left(s'\right)
\end{aligned}$$

$Q_\pi(s,a)$和$V_\pi(s)$的关系

根据上式推导：

$$Q_\pi(s,a)=R(s,a)+\gamma\sum_{s'\in S}P\left(s'\mid s,a\right)V_\pi\left(s'\right)\tag{1}$$

根据上一小节给出的关系：

$$V_\pi(s)=\sum_{a\in A}\pi(a|s)Q_\pi(s,a)\tag{2}$$

把式（1）带入式（2）得：

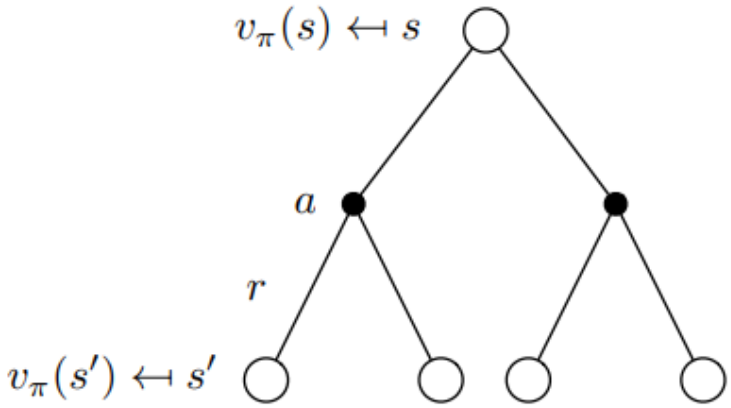
$$V_\pi(s)=\sum_{a\in A}\pi(a|s)\left(R(s,a)+\gamma\sum_{s'\in S}P\left(s'\mid s,a\right)V_\pi\left(s'\right)\right)\tag{3}$$

把式（2）带入式（1）得：

$$Q_\pi(s,a)=R(s,a)+\gamma\sum_{s'\in S}P\left(s'\mid s,a\right)\sum_{a\in A}\pi(a|s')Q_\pi(s',a)\tag{4}$$

2.3.4 备份图（backup diagram）

备份过程（又称为更新操作）是算法的图形化表示，通过图形表示状态，动作，状态转移，奖励等。下图中，空心较大圆圈表示状态，黑色实心小圆表示的是动作，连接状态和动作的线条仅仅把该状态以及该状态下可以采取的动作关联起来,黑色小圆和后续状态之间的连线为即时奖励r。

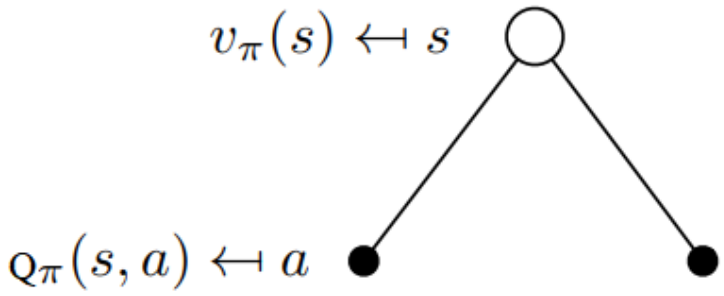


从上图可知，价值函数$v_\pi(s)$ 的更新是从后续状态s' 传回给黑色节点，再从黑色节点传回给根节点s。所以说，价值函数的备份过程（更新）通过当前状态的后续状态传回给当前状态。图示的关系构成了更新或备份操作的基础，而这些操作是强化学习方法的核心。

$Q_\pi(s,a)$和$V_\pi(s)$关系的备份图

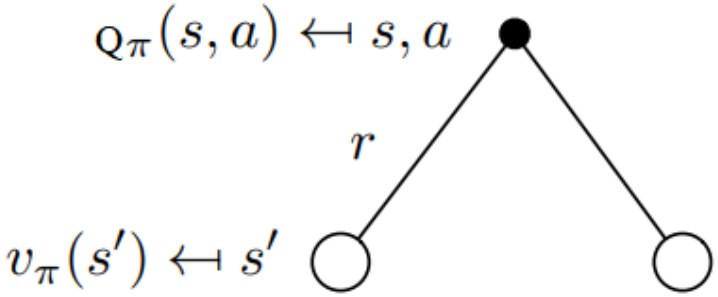
从下图（称为V值备份图）可以看出，在状态s时，遵循策略π后，状态s的价值体表示为在该状态下遵循某一策略而采取所有可能动作的动作价值（Q值）按该状态下动作发生概率（策略）的乘积求和，即

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

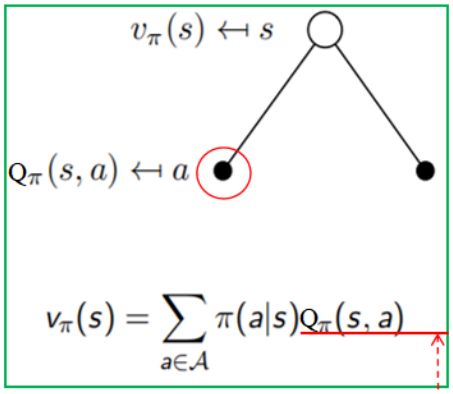


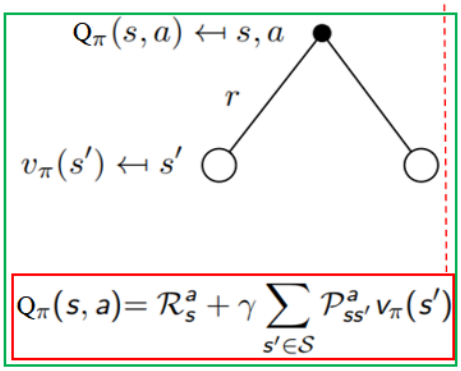
类似地，从下图（称为Q值备份图）可以看出，在状态s时，遵循策略π，采取动作a后，$(s，a)$的动作价值函数表示为在该状态下采取动作a后的即时奖励，加上带折扣求和，求和项为：该状态s下，采取动作a后转移到所有可能后续状态s’ 的转移概率和所有可能后续状态s’的状态价值函数相乘后，按所有可能后续状态求和，即

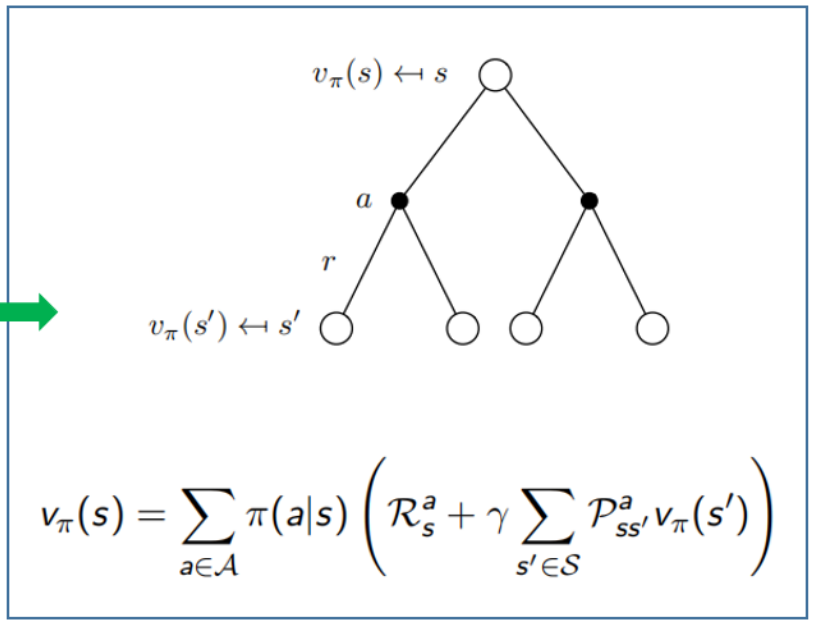
$$Q_\pi(s,a) = R(s,a) + \gamma \sum_{s' \in S} P\left(s' \mid s,a\right) V_\pi\left(s'\right)$$



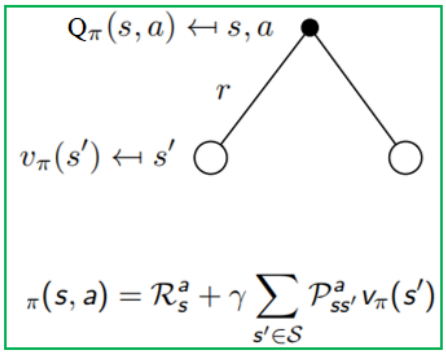
把Q值备份图带入到V值备份图，得到如下图所示的

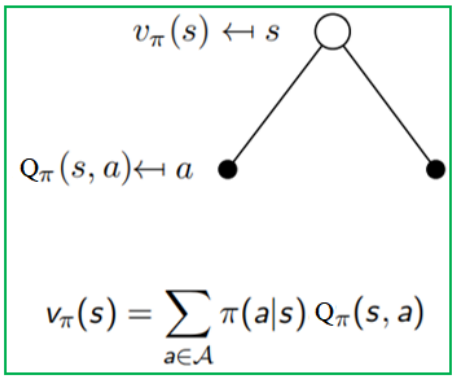


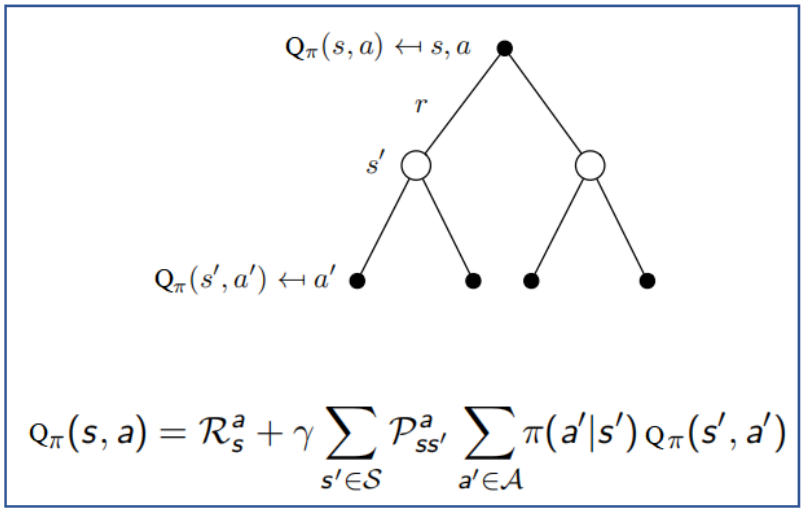




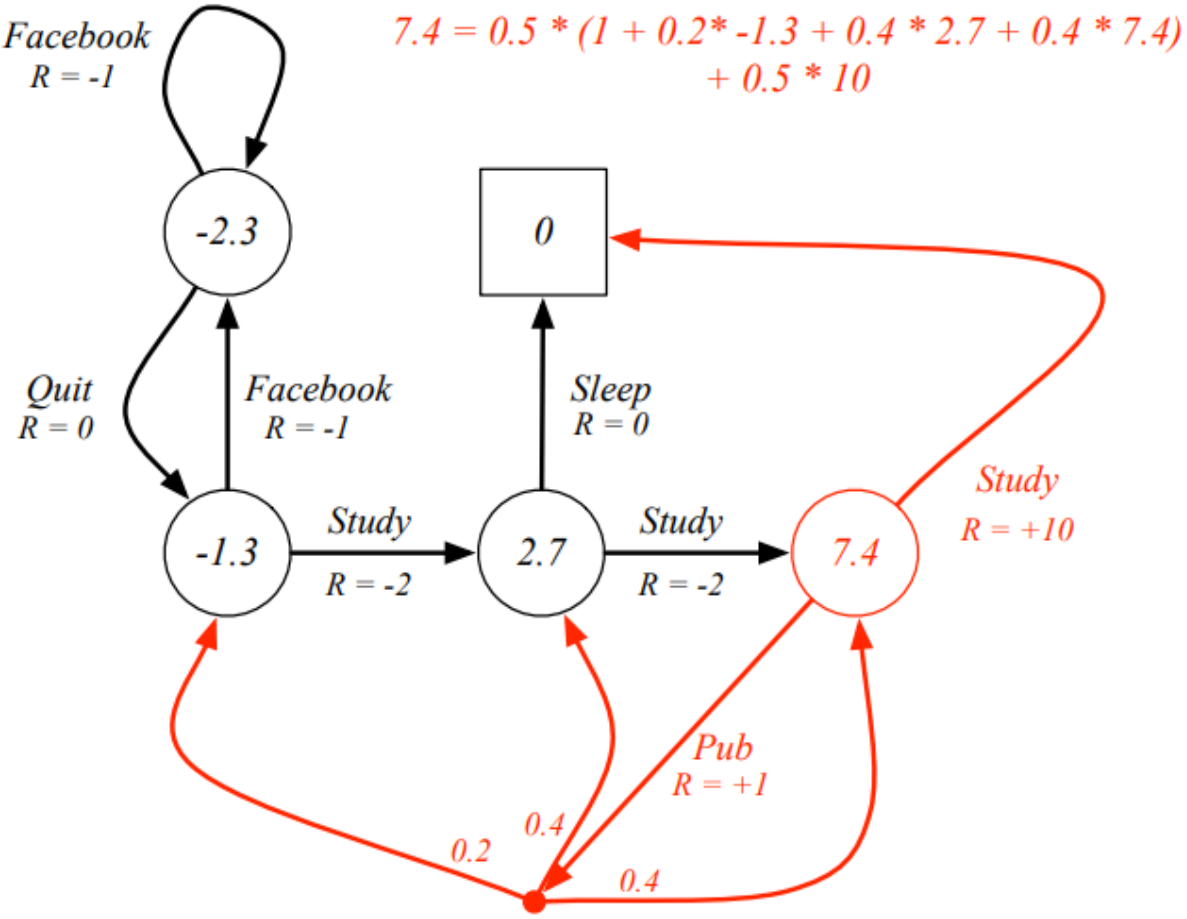
同理，把V值备份图带入到Q值备份图，得到如下图所示的







例2-7 学生MDP的贝尔曼期望方程



上图给出了红色空心圆圈状态的状态价值是如何计算的，遵循的策略为随机策略，即所有可能的动作等概率被执行。

贝尔曼期望方程的矩阵形式

MDP的贝尔曼期望方程的矩阵形式可以通过MRP的贝尔曼方程的的矩阵形式推导而来，其实就是增加了策略，

$$V_\pi = R^\pi + \gamma P^\pi V_\pi$$

解析解为：

$$V_\pi = (I - \gamma P^\pi)^{-1} R^\pi$$

2.3.5 最优价值函数（Optimal Value Function）

最优价值函数

最优状态价值函数$V_*(s)$ 指的是从策略产生的所有状态价值函数中，选取使得状态s价值最大的函数：

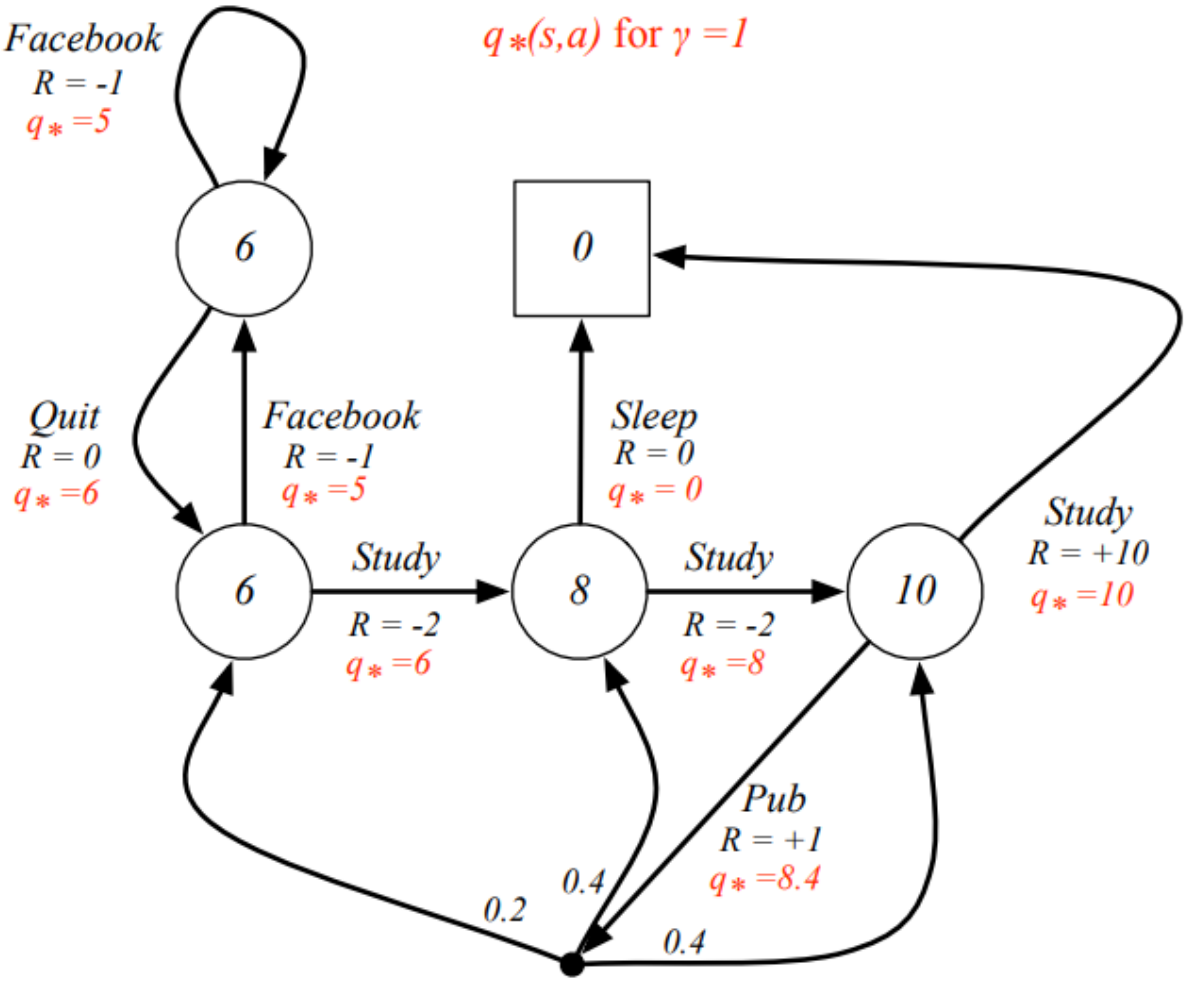
$$v_* = \max_\pi V_\pi(s)$$

类似的，最优行为价值函数$Q_*(s,a)$指的是从策略产生的所有行为价值函数中，选取使得状态动作对$<s,a>$价值最大的函数：

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

最优价值函数具体明确了MDP的最优可能表现，当我们知道了最优价值函数，也就知道了每个状态的最优价值，也就是解决了MDP问题，回顾第一讲的走迷宫问题中，基于价值的方法给出的状态价值就是最优价值，也就隐式地给出了策略。

例2-8 学生MDP 最优动作价值函数



上图中的Class3（圈内状态价值为10）采取Pub后，分别进入到Class1，Class2, Class3的最优动作价值函数计算如下：

$$q_* = R(C3, Pub) + \gamma(P^{Pub}_{C3,C1}V(C1) + P^{Pub}_{C3,C2}V(C2) + P^{Pub}_{C3,C3}V(C3))$$
$$q_* = 1 + 1 * (0.2 * 6 + 0.4 * 8 + 0.4 + 10) = 9.4$$

所以，个人认为上图中标的8.4（没有加即时奖励）是不对的。。

最优策略（optimal policy）

定义关于策略的偏序（partial ordering）：当对于任何状态s，遵循策略π的状态价值大于等于遵循策略 π' 的状态价值，则策略π优于策略 π’，即

$$\pi \geq \pi' \ if \ v_\pi(s) \geq v_{\pi'}(s) \ , \forall s$$

定理: 对于任何MDP，有下面性质：

- 存在一个最优策略，比任何其他策略更好或至少相等 $\pi_* \geq \pi, \ \forall \pi$
- 所有的最优策略有相同的最优价值函数$v_{\pi_*}(s) = v_*(s)$
- 所有的最优策略具有相同的行为价值函数$Q_{\pi_*}(s,a) = Q_*(s,a)$

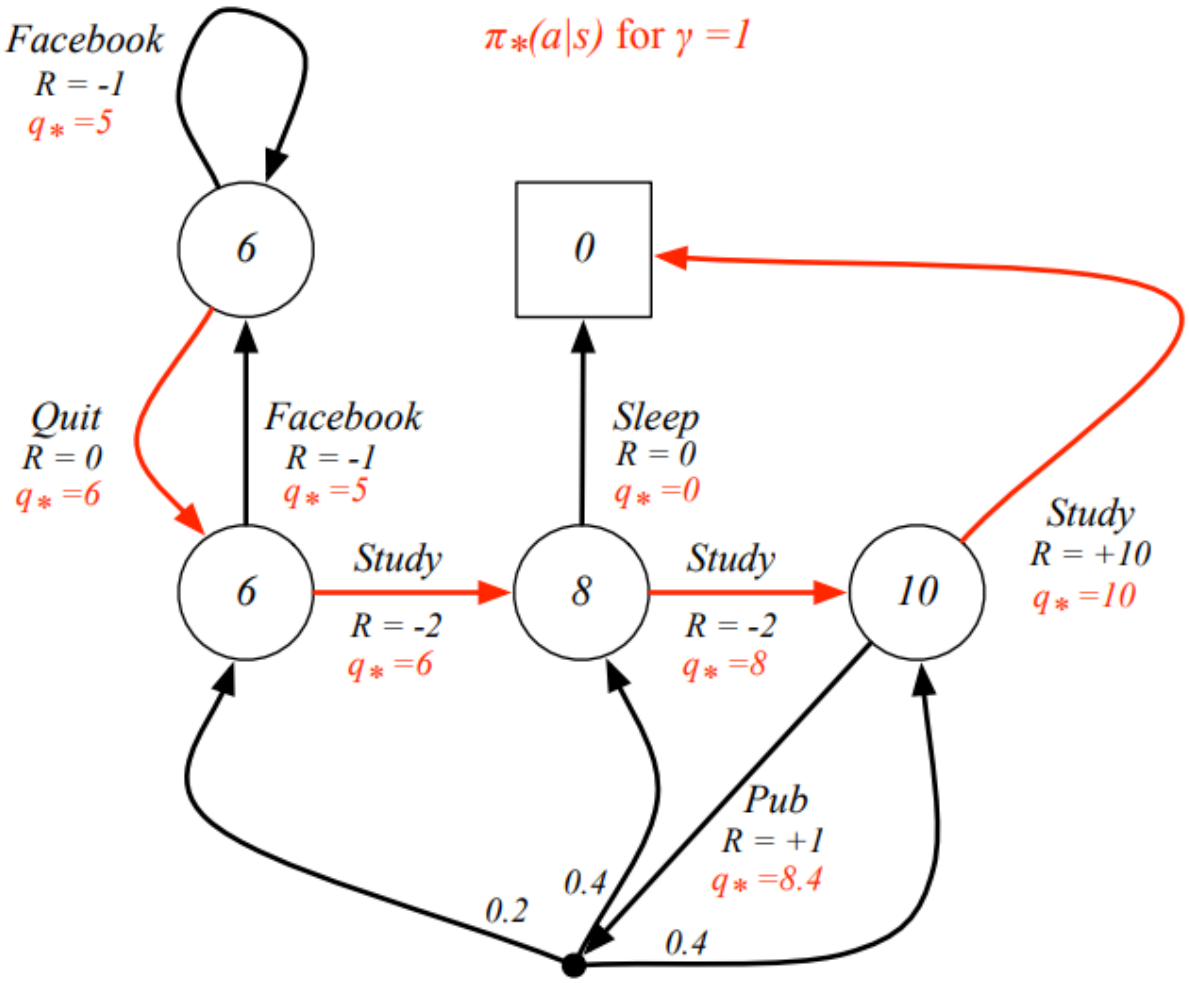
寻找一个最优策略

可以通过最大化最优行为价值函数来找到最优策略：

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

对于任何MDP问题，一定存在一个最优的且是确定的策略，但这并不意味着只有确定策略是最优的。也可以是随机策略是最优的，比如在某个状态s时，有两个动作a1,a2有相同的Q值,我们可以随机选择其中之一。 另外，如果我们知道最优行为价值函数，则通过最优Q值可以找到最优策略,即上式中采取使得最优Q值最大的动作。

例 2-9 学生MDP最优策略示例，图中的红色箭头表示最优策略

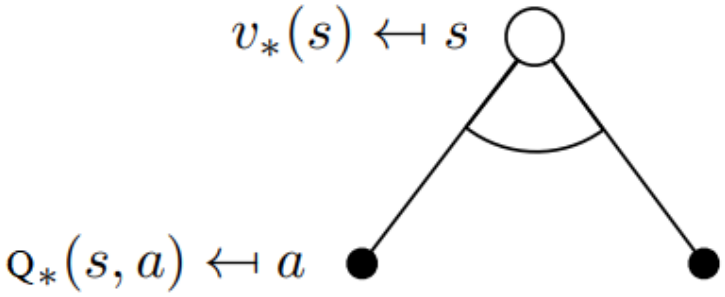


2.3.6 贝尔曼最优方程（Bellman Optimality Equation）

V^*的贝尔曼最优方程

一个状态的最优价值等于从该状态出发采取的所有动作产生的动作价值中最大的那个动作价值。

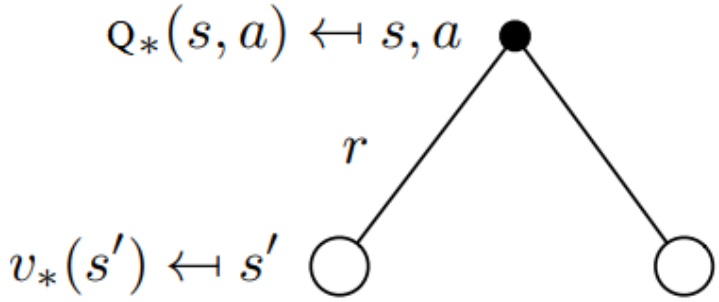
$$V_*(s) = \max_a Q_*(s,a)$$



Q^*的贝尔曼最优方程

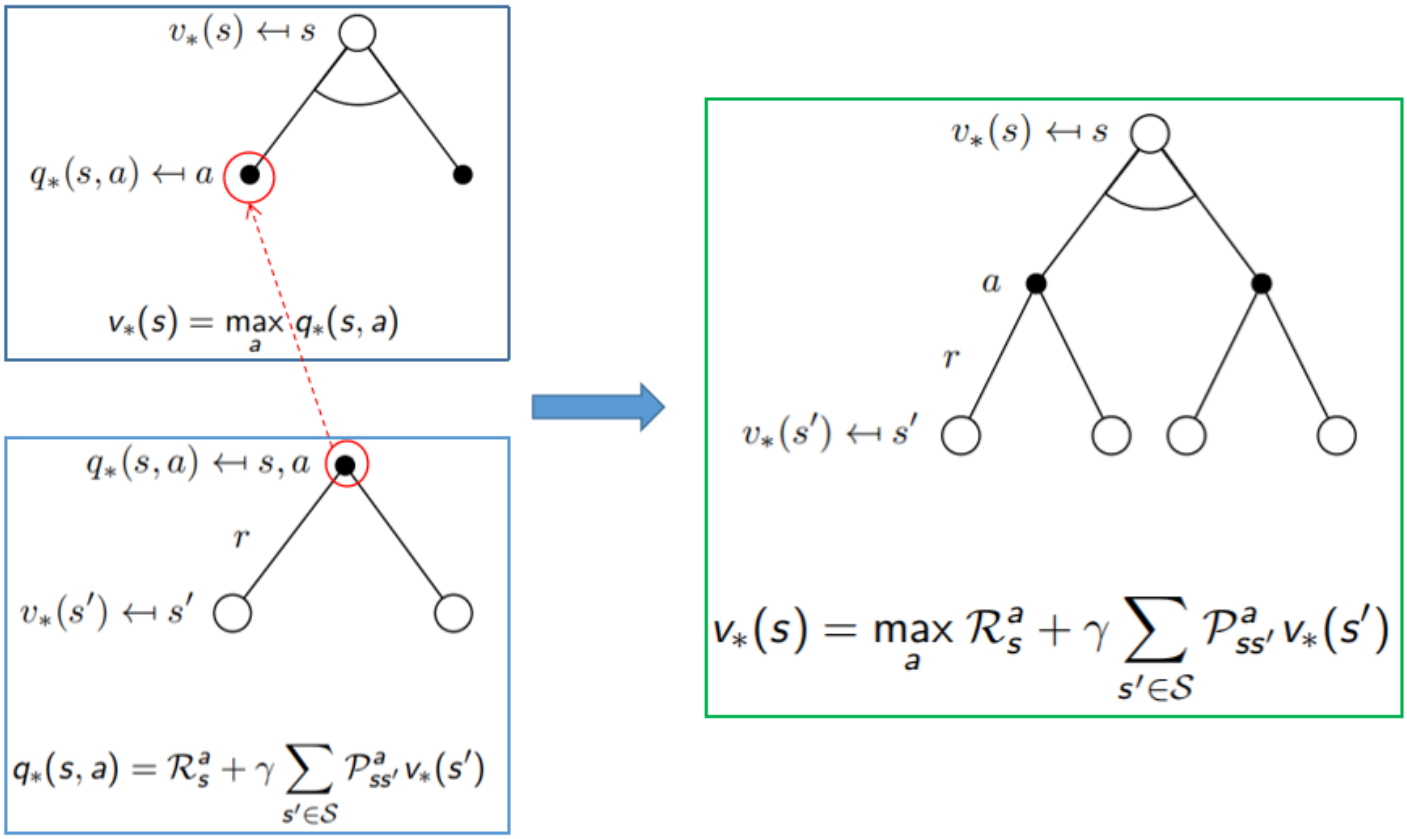
在某个状态s下，采取某个动作的最优动作价值由两部分组成，一部分是离开状态 s 的即时奖励，另一部分则是所有能到达的后续状态 s' 的最优状态价值与相应的转移概率的乘积后求和。

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

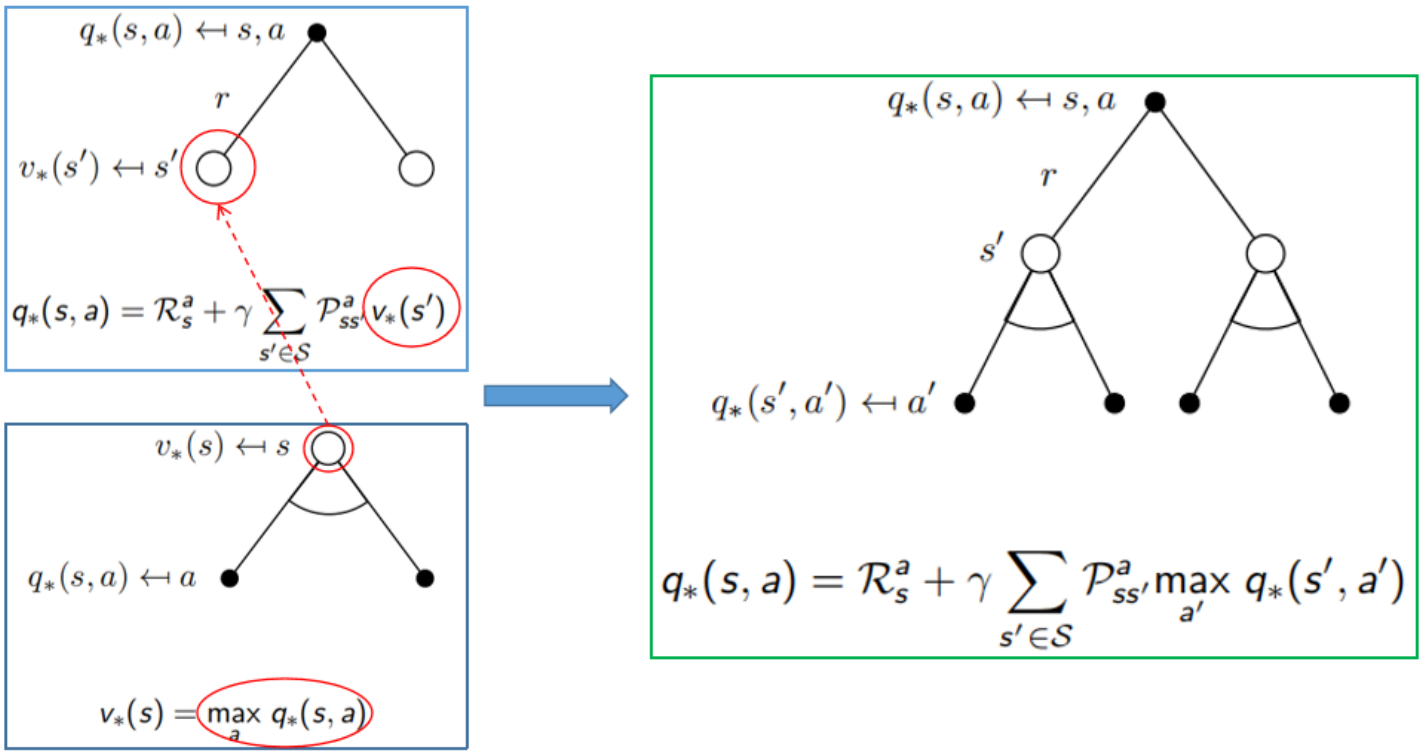


V^*的贝尔曼最优迭代方程

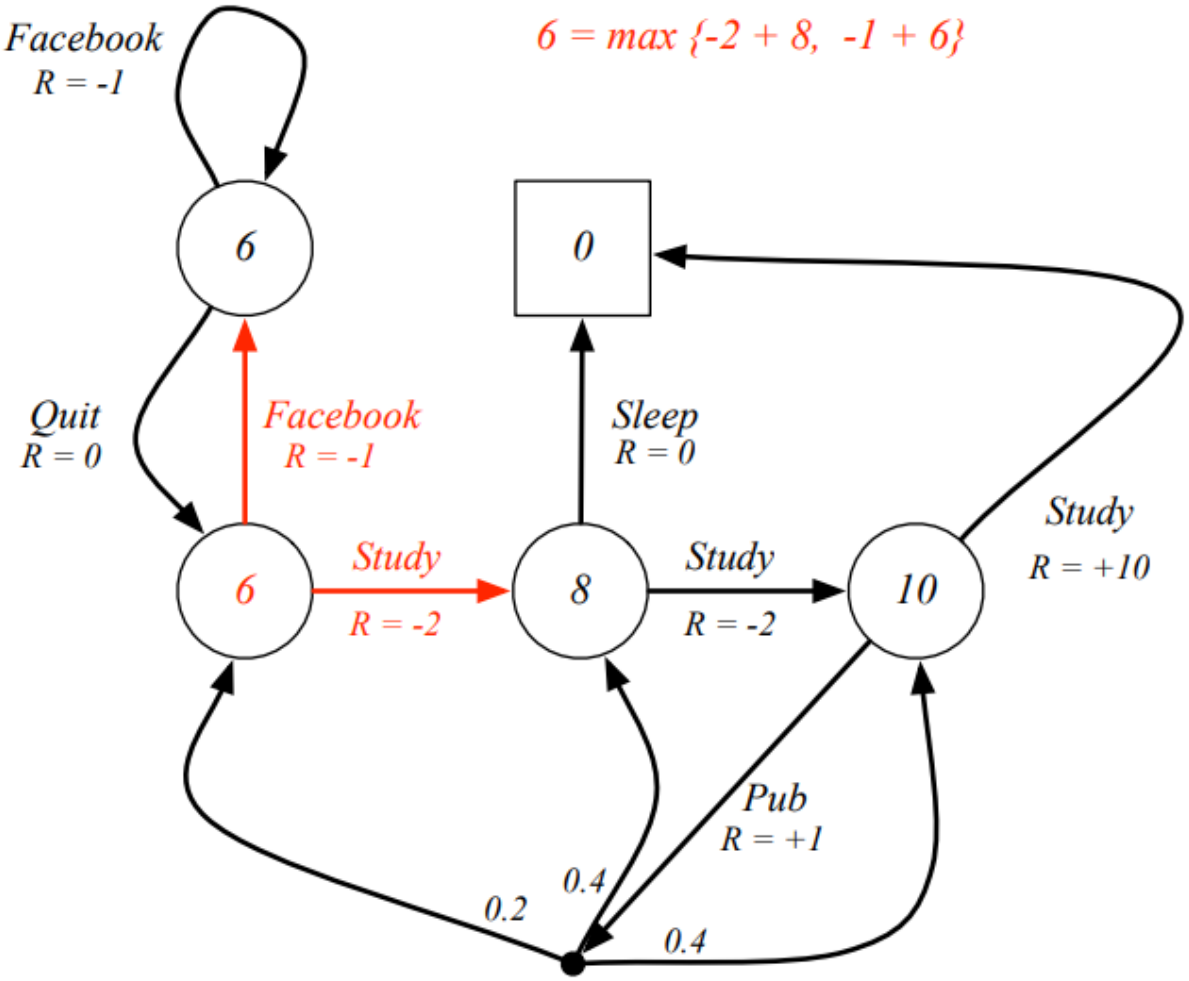
如前面的贝尔曼方程一样，通过上面两个最优贝尔曼方程的相互代入的方式（Q^*代入V^*，V^*代入Q^*），得到状态（动作）价值函数的最优贝尔曼迭代方程



Q^*的贝尔曼最优迭代方程



例2-10 学生MDP贝尔曼最优方程



贝尔曼最优方程的求解

Bellman最优方程是非线性的，通常没有闭式解，但可以通过一些迭代方法来求解：价值迭代、策略迭代、Q-learning、Sarsa等。将在后面几讲介绍这些方法。

2.3.7 马尔可夫决策过程的扩展

其他类别的MDP有：无限状态和连续MDP；部分可观测MDP；不带折扣的、平均奖励MDP等，具体介绍略，有兴趣的读者可以自己阅读相关资料。

参考

1.David Silver第2课
2.叶强《David Silver强化学习公开课中文讲解及实践》
3.《CS285》第6讲
4.其他网页资料