概念

强化学习研究的是智能体agent与环境之间交互的任务，也就是让agent像人类一样通过试错，不断地学习在不同的环境下做出最优的动作，而不是有监督地直接告诉agent在什么环境下应该做出什么动作。既然与环境交互，必然会有反馈，也即回报。

强化学习和监督学习的区别主要有以下两点：

1. 强化学习是试错学习(Trail-and-error)，由于没有直接的指导信息，智能体要以不断与环境进行交互，通过试错的方式来获得最佳策略。（可以认为是交互产生样本）
2. 延迟回报，强化学习的指导信息很少，而且往往是在事后（最后一个状态）才给出的，这就导致了一个问题，就是获得正回报或者负回报以后，如何将回报分配给前面的状态。

MDP

显然MDP是一个离散的过程，当前的执行的动作的回报依赖于未来的回报（延迟回报），一个使得在任意时刻和状态下的长期回报都是最大的策略是我们最终需要得到的。因此最简单的就是累计回报

$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \ldots = \sum_{k=t+1}^{\infty} R_k$ 但实际上我们一般会用下面更通用的公式来代替：

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

马尔可夫决策过程

一个有限马尔可夫决策过程由一个四元组构成 M=(S,A,P,R)。如上所述，S表示状态集空间，A表示动作集空间，P表示状态转移概率矩阵，R表示期望回报值。

给定当前状态和动作，转移到另一个状态的概率计算：

$$p\left(s'|s,a\right) = \Pr\{S_{t+1} = s'|S_t = s, A_t = a\} \in \mathbf{P}$$

在给定状态、动作和下一个状态的前提下，回报的计算：

$$r\left(s,a,s'\right) = \mathbf{E}\left[R_{t+1}|S_t = s, A_t = a, S_{t+1} = s'\right] \in \mathbf{R}$$

值函数及贝尔曼公式

增强学习的最终结果是找到一个环境到动作的映射—即策略$\pi(a|s)$，在几乎所有的强化学习理论中都会定义值函数来表示给定策略下期望的未来回报，并将值函数作为评估学习效果的指标。值函数有多种定义，目前常见的是将值函数直接定义为未来回报的期望：

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

上面表示的是在某个策略π下，当环境处于状态s时未来回报的期望，因此又叫做状态值函数(state-value function for policy)，只跟当前状态有关。同样，我们也可以定义动作值函数(action-value function for policy)，如下：

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

值函数和动作值函数的区别在于动作值是在给定动作下的期望回报，从形式上看，都可以写成递归的形式：

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

$$=\mathbf{E}_\pi\left[R_{t+1}+\gamma\sum_{k=0}^{\infty}\gamma^k R_{t+k+2}|S_t=s\right]$$

$$=\sum_a\pi(a|s)\cdot\mathbf{E}_\pi\left[R_{t+1}+\gamma\sum_{k=0}^{\infty}\gamma^k R_{t+k+2}|S_t=s,A_t\right]$$

$$=\sum_a\pi(a|s)\sum_{s'}p\left(s'|s,a\right)\left[r\left(s,a,s'\right)+\gamma\mathbf{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+2}|S_{t+1}=s'\right]\right]$$

$$=\sum_a\pi(a|s)\sum_{s'}p\left(s'|s,a\right)\left[r\left(s,a,s'\right)+\gamma v_\pi\left(s'\right)\right]$$

因此当前状态下的回报可以用下一状态下的值函数求解，这个公式就是贝尔曼公式。

同理状态值函数也可以递归求解：

$$\begin{aligned}q_\pi(s,a)&=\mathbf{E}_\pi\left[G_t|S_t=s,A_t=a\right]\\&=\mathbf{E}_\pi\left[R_{t+1}+\gamma\sum_{k=0}^{\infty}\gamma^k R_{t+k+2}|S_t=s,A_t=a\right]\\&=\sum_{s'}p\left(s'|s,a\right)\left[r\left(s,a,s'\right)+\gamma v_\pi\left(s'\right)\right]\end{aligned}$$

根据贝叶斯公式可以把两者关联：

$$v_\pi(s)=\sum_a\pi(a|s)q_\pi(s,a)$$

最优值函数及贝尔曼最优公式

我们的目标就是使得任意时刻未来回报的期望值都是最大的，为此目标可以是：

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

策略可能有多个，但是最优的状态值函数和动作值函数是一致的：

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

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

目标函数有了，最重要的就是如何求解最优策略$\pi_*(a|s)$，由于状态之间的转义是依赖动作的，因此不需要求解状态转义的概率。

贝尔曼公式与贝尔曼最优公式是MDP求解的基础，下面主要介绍几种MDP求解的方法。

参考链接