理论部分

本次报告的理论部分介绍有限马尔科夫决策过程与强化学习的理论知识、单步时序差分算法的理论知识和论文《xxx》中具体使用的算法三部分内容。

有限马尔科夫决策过程(finite MDP)

基本概念

MDP（马尔科夫决策过程）是基于马尔科夫性假设的一种决策序列模型，是一种通过交互式学习来实现目标的理论框架。MDP的原理在最优控制、强化学习等领域得到了广泛的应用。它包含**智能体(agent)**、**环境(environment)**、**状态(state)**、**动作(action)**、**收益(reward)** 5大要素。每个时刻，智能体在当前状态 s 下选择一个动作 a 后，环境对动作做出相应，反馈给智能体新的状态 s' 和一个收益 r，这一过程发生的可能性定义为概率[1]

$$p(s', r \mid s, a) \triangleq \Pr\{S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a\} \tag{1}$$

由此定义可以看出，下一时刻智能体的状态与收益只与当前时刻的状态和动作有关，与历史信息无关，体现了MDP状态转移的马尔科夫性。

有限MDP是指智能体状态、动作和收益的集合($\mathcal{S}$、$\mathcal{A}$、$\mathcal{R}$)都只含有有限个元素的MDP。$\mathcal{S}$、$\mathcal{A}$、$\mathcal{R}$ 刻画了MDP的静态特性，而概率 $p(s', r \mid s, a)$ 刻画了MDP的动态特性。它们共同确定了这个MDP决策过程中"环境"的数学内涵。

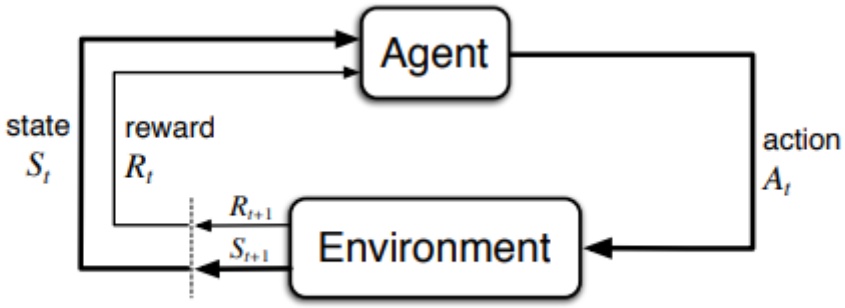

Figure 3.1: The agent–environment interaction in a Markov decision process.

在一个MDP过程里，智能体的目标是给出一个当前环境下的最优策略。**策略**是指智能体处于某个状态 s 时，它选择各种可能的行动 a 的概率，即策略 π 定义为

$$\pi(a|s) \triangleq \Pr\{A_t = a | S_t = s\} \tag{2}$$

那么什么策略是最优的，这取决于我们实际问题中要实现的目标是什么。这个目标我们用**回报**来描述。一般情况下，我们定义时刻 t 的回报 G_t 为：

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

其中 $0 \leq \gamma \leq 1$ 是一个参数，被称为**折扣率**。 当 $\gamma = 1$ 时，(3)式是容易理解的，即从此刻起未来所有收益的总和就是这一时刻的回报。之所以要加入折扣率，一方面是数学上可以使求和式易于收敛，另一方面则是实际问题中，当前的收益是确定的，而未来的收益基于模型的预测，是不确定的，价值相对更低，因此通过折扣率 γ 来修正智能体综合考虑当下与未来的权重大小。

有了回报的定义，最优策略就可以用期望得到的回报最大来描述，即我们希望找到的**最优策略** π_* 满足

$$\mathbb{E}_{\pi_*}\left[G_t \mid S_t = s\right] \geq \mathbb{E}_{\pi}\left[G_t \mid S_t = s\right] \qquad \forall \pi,\ s \tag{4}$$

（注：根据Bellman最优原理，这个最优策略是存在的。）

求解最优策略

为了求解最优策略，我们需要定义策略 π 的两个价值函数：

我们定义**策略 π 的状态价值函数** $v_\pi(s)$ 为：

$$v_\pi(s) \triangleq \mathbb{E}_\pi\left[G_t \mid S_t = s\right] \tag{5}$$

表示智能体在状态 s 下，按照策略 π 决策所得到的期望回报。

定义**策略 π 的动作价值函数** $q_\pi(s,a)$ 为：

$$q_\pi(s,a) \triangleq \mathbb{E}_\pi\left[G_t \mid S_t = s, A_t = a\right] \tag{6}$$

表示智能体在状态 s 下执行动作 a 后，按照策略 π 决策所得到的期望回报。

计算状态价值函数 $v_\pi(s)$ 可以使用**贝尔曼方程**：

$$\begin{aligned} v_\pi(s) &\doteq \mathbb{E}_\pi\left[G_t \mid S_t = s\right] \\ &= \mathbb{E}_\pi\left[R_{t+1} + \gamma G_{t+1} \mid S_t = s\right] \\ &= \sum_a \pi(a \mid s) \sum_{s'} \sum_r p\left(s', r \mid s, a\right)\left[r + \gamma \mathbb{E}_\pi\left[G_{t+1} \mid S_{t+1} = s'\right]\right] \\ &= \sum_a \pi(a \mid s) \sum_{s', r} p\left(s', r \mid s, a\right)\left[r + \gamma v_\pi\left(s'\right)\right], \qquad \text{for all } s \in \mathcal{S} \end{aligned} \tag{7}$$

求解 $|\mathcal{S}|$ 个线性方程组成的方程组得到。

求解最优策略，就相当于最大化价值函数，通过类似的推导，我们可以得到求解最优价值函数的**贝尔曼最优方程**：

$$\begin{aligned} v_*(s) &= \max_a \mathbb{E}\left[R_{t+1} + \gamma v_*\left(S_{t+1}\right) \mid S_t = s, A_t = a\right] \\ &= \max_a \sum_{s', r} p\left(s', r \mid s, a\right)\left[r + \gamma v_*\left(s'\right)\right] \end{aligned} \tag{8.1}$$

$$\begin{aligned} q_*(s,a) &= \mathbb{E}\left[R_{t+1} + \gamma \max_{a'} q_*\left(S_{t+1}, a'\right) \mid S_t = s, A_t = a\right] \\ &= \sum_{s', r} p\left(s', r \mid s, a\right)\left[r + \gamma \max_{a'} q_*\left(s', a'\right)\right]. \end{aligned} \tag{8.2}$$

一旦求出了最优的价值函数，就可以轻易得到最优策略。具体而言，在得到 v_* 后，可以通过单步搜索的贪心策略得到全局最优动作，从而得到最优策略 π_* ； 在得到 q_* 后，只需要选择取值最大的 $q_*(s,a)$ 就可以得到状态 s 时的全局最优动作 a ，从而得到最优策略 π_* 。

基本问题

对于实际中的MDP模型，我们关注两类问题：**预测问题**和**控制问题**。预测问题是指，当智能体的当前策略 π 确定时，如何求出该策略的价值函数。控制问题是指，对于给定的MDP模型，寻找出智能体的最优策略。解决预测问题是解决控制问题的前提，或者说，预测问题可以看作控制问题的一个子问题。一个经典的求解MDP模型控制问题的框架叫做广义策略迭代(GPI)。GPI的思想便是通过交替的完成对当前策略价值函数的计算和根据当前策略的价值函数改进策略两个步骤实现最优策略的求解。在本次实验中，我们将问题的一部分建模为了MDP模型的预测问题。

单步时序差分算法(TD(0))

虽然在理论上，求解MDP问题有着明确的公式，但观察(7)、(8.1)、(8.2)可以发现，精确使用贝尔曼方程求解需要的计算量与状态空间的大小有关。由于实际问题中状态空间 $|\mathcal{S}|$ 巨大，因此在实际中使用贝尔曼最优方程精确求解几乎是不可能的。我们必须开发其他算法降低求解的复杂度。单步时序差分算法(TD(0)) 就是一种常见的求解方法。

TD(0)方法是一种迭代法，它遵循迭代方法的一般框架：

$$\text{新估计值} \leftarrow \text{旧估计值} + \text{步长} \times \left[\text{目标} - \text{旧估计值}\right]$$

对于状态价值函数 $v_\pi(s)$，理想的"目标"自然是 $v_\pi(s)$ 的真实值 $\mathbb{E}_\pi\left[G_t \mid S_t = s\right]$。然而这个值的计算涉及求期望和计算回报 G_t 两个困难。TD(0)的基本思想是：用采样近似期望，用当前价值代替最优价值，即

$$\begin{aligned}\mathbb{E}_\pi\left[G_t \mid S_t = S\right] &= \mathbb{E}_\pi\left[R_{t+1} + \gamma G_{t+1} \mid S_t = S\right] \\ &\approx \mathbb{E}_\pi\left[R_{t+1} + \gamma v_\pi\left(S_{t+1}\right) \mid S_t = S\right] \\ &\approx R + \gamma V\left(S'\right)\end{aligned}$$

其中 R, S' 分别是在状态 S 时根据策略 π 进行一次模拟采样后得到的收益和转移到的新状态。

因此最终的计算公式为：

$$V(S) \leftarrow V(S) + \alpha\left[R + \gamma V\left(S'\right) - V(S)\right]$$

完整的TD(0)预测算法如下图：

Tabular TD(0) for estimating v_π

Input: the policy π to be evaluated
Algorithm parameter: step size $\alpha \in (0, 1]$
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop for each episode:
  Initialize S
  Loop for each step of episode:
    $A \leftarrow$ action given by π for S
    Take action A, observe R, S'
    $V(S) \leftarrow V(S) + \alpha\left[R + \gamma V(S') - V(S)\right]$
    $S \leftarrow S'$
  until S is terminal

我们将TD(0)算法中价值更新的部分 $R_{t+1} + \gamma V\left(S_{t+1}\right) - V(S_t)$ 称作**TD误差**，记为 δ_t。我们可以推导出：

$$\begin{aligned} G_t - V(S_t) &= R_{t+1} + \gamma G\left(S_{t+1}\right) - V(S_t) + \gamma V\left(S_{t+1}\right) - \gamma V\left(S_{t+1}\right) \\ &= \delta_t + \gamma\left(G_{t+1} - V(S_{t+1})\right) \end{aligned}$$

即用TD误差可以建立起当前时刻与下一时刻算法对价值函数的估计的偏差之间的关系。

TD(0)算法可以看作是动态规划方法（用当前价值代替最优价值）和蒙特卡洛方法（用采样近似期望）思想的结合，很好地继承了两种方法各自的优点。相比与依靠纯模拟的蒙特卡洛方法，TD(0)算法不需要等待一个完整模拟的结束，只需要模拟一个时刻就可以立即得到估计值；相比与动态规划方法，TD(0)算法不需要一个环境模型，即描述收益和下一状态联合概率分布的模型。TD(0)的收敛性也具有理论保证。对于任意的策略 π， TD(0)都已经被证明能够收敛到 v_π。

[1] Montague P R. Reinforcement learning: an introduction, by Sutton, RS and Barto, AG[J]. Trends in cognitive sciences, 1999, 3(9): 360.