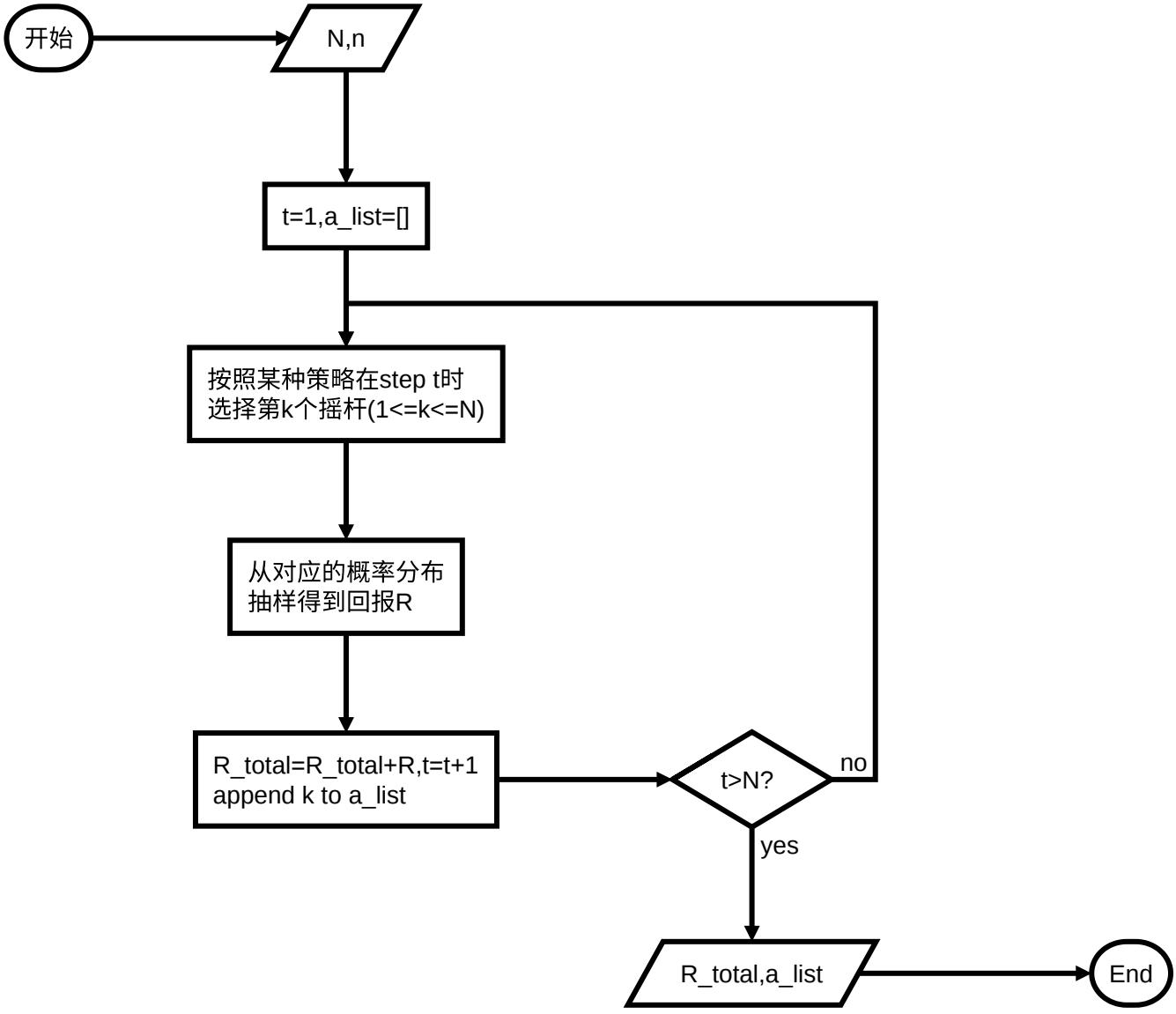
第二、三章读书笔记

1911667 闫纪甫

2.多臂赌博机

2.1 n-Armed Bandit Problem

n臂赌博机问题指的是有n个摇杆(n>1)，每次作出动作选择之后，比如摇其中的第k个，对应的赌博机就会给出一个数值回报。这个数值回报是从这个选择动作（摇第k个杆）对应的平稳概率分布中抽取出来的。目标是在做N次选择之后($time\ steps = N$)，使得总的回报最大。该问题写成流程图如下：



2.2 Action-Value Methods（four action selection methods) & incremental implementation

当采用定义的方法计算，即直接进行与环境交互得到许多奖励、然后求平均的方法。这也是蒙特卡洛方法的基础。状态a处的行为的值为

$$Q_t(a) = \frac{R_1 + R_2 + ... + R_{N_t(a)}}{N_t(a)}$$

根据大数定律：$Q_t(a) \to q(a)\ as\ N_t \to \infty$。可以估计q(a)的值。
而接下来有两种策略，第一种是选择当前值最大的那个行为作为最优策略,即贪心策略。即：

$$A_t = \arg\max_a Q_t(a)$$

贪心的策略选择就是要利用现有的知识，使得当前的回报最大。但是为了得到最优策略，有时需要偶尔进行探索。即是**探索-利用平衡策略**。其中比较简单的一种被称为ϵ-greedy动作选择法：

$$A_t^* = \begin{cases} \arg\max_a Q_t(a), \text{with probability } 1-\epsilon \\ \text{a random action, with probability } \epsilon \end{cases}$$

另外还有两种动作选择法，分别称为UCB动作选择和波尔兹曼动作选择：

$$A_t = \arg\max_a \left[Q_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}} \right]$$

$$P_r\{A_t = a\} = \frac{e^{H_t(a)}}{\sum_{b=1}^{k} e^{H_t(b)}} = \pi(a)$$

它们的参数分别是c和k。
当选择了目前的动作之后带来的回报是R，会进行状态更新迭代（增量实现）：

$$N(A) \leftarrow N(A) + 1$$
$$Q(A) \leftarrow Q(A) + \frac{1}{N(A)}(R - Q(A))$$

其中我们将参数$\alpha = \frac{1}{k} = \frac{1}{N(A)}$作为增量的步长。将当即汇报R与先前的估计值Q(A)的差值叫做一次估计时的**误差**。

2.3 Tracking a Nonstationary Problem

以上直接平均的策略在平稳状态可以有效使用。然而，在更多情况下，环境是非平稳的。也就是说，**来自近期的信息比更久远的信息更有效**，这与研究非线性问题而在一小段范围内线性化具有相似的思想。

于是，有一种估计方法让α不随时间步长增加而变化，即让其保持定常量$\alpha \in (0,1]$的状态。这种状态便于我们利用加权平均的思想处理该情况，对该状态的估计利用的信息越新，赋予其权重越大：

$$Q_{k+1} = (1-\alpha)^k Q_1 + \sum_{i=1}^{k} \alpha(1-\alpha)^{k-i} R_i$$

该方程也是利用增量式不断迭代得到的。
设置$\alpha_k(a) = \alpha$而非$\frac{1}{k}$固然会导致其部分收敛条件不满足。但是既然是在非平稳环境下，这种状态也是我们所需要的。

2.4 Optimistic Initial Values

设置一定的初始动作值Q_1对于强化学习的探索过程十分有帮助。在短期内，得到的回报可能会低于估计值，这反而会使得估计值收敛前各个动作都会被尝试几次。虽然短期内会降低累积的回报，但长期来看，这有利于学习到最佳动作，提高长期的总回报。

3.有限马尔可夫决策

3.1 The Agent–Environment Interface

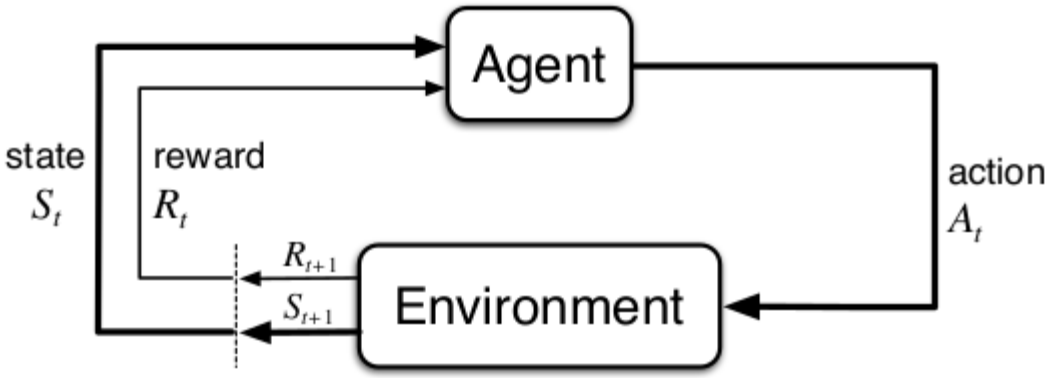


Figure 2: The agent–environment interaction in reinforcement learning.

对图2模型的理解

1. 在任意的离散时间步$t = 0, 1, 2, 3, ...$，智能体(agent)能够创建一个从环境(environment)给出的状态(state)到选择每一个环境状态下容许的动作(action)的概率的映射(mapping)关系。这种映射叫做该智能体在时间步t下的策略(policy)π_t。这也就定义出了条件概率：$\pi_t(a|s)$
2. 环境给智能体的奖励(reward)S_t，是在环境通过t时刻的自身状态和智能体给出的动作A_t计算出来的一个数值。智能体的唯一目标就是要在一个较长的时间段下，最大化总奖励。

细化

将总奖励进行规范化表达，常用折扣回报来量化t时刻后期望能够获得的回报(Return)：

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

其中，T指的是终止时刻。$\gamma \in [0, 1)$使得$T \to \infty$时的回报依然收敛。$\gamma = 0$可以理解为贪心策略。$\gamma = 1$即相当于没有折扣。

3.2 马尔可夫决策过程

> A state signal that succeeds in retaining all relevant information is said to have the Markov property.(informal)
> A state signal has the Markov property $\Leftrightarrow \Pr\{R_{t+1} = r, S_{t+1} = s' | S_t, A_t\} = \Pr\{R_{t+1} = r, S_{t+1} = s' | S_0, A_0, S_1, A_1, ..., S_t, A_t\}$

马尔可夫决策过程(MDP)是环境的性质。我们用$p(s', r|s, a)$表示$\Pr\{R_{t+1} = r, S_{t+1} = s' | S_t, A_t\}$。在具有马尔可夫性的环境中，因为给出蕴含当下的历史信息与只给出当下的信息是等价的，智能体只需要根据当前给出的信息就可以预测未来的状态和期望的奖励。
作为基础，可以为强化学习基本模型的一些术语进行量化：

- 预期的奖励

$$r(s, a) = \mathrm{E}[R_{t+1} | S_t = s, A_t = a] = \sum_{r \in \mathcal{R}} r \sum_{s' \in \mathcal{S}} p(s', r|s, a)$$

- 转移概率

$$p(s'|s, a) = \Pr\{S_{t+1} = s' | S_t = s, A_t = a\} = \sum_{r \in \mathcal{R}} p(s', r|s, a)$$

- 已知下一步状态下，预测的奖励

$$r(s, a, s') = \frac{\sum_{r \in \mathcal{R}} r p(s', r|s, a)}{p(s'|s, a)}$$

而$r(s, a, s')$在**状态转移图**中会得到应用。以上的前提就是来自环境的状态信号是可知的。

3.3 值函数及优化

对于MDP，定义策略π的值函数v_π为

$$v_\pi(s) = \mathrm{E}[G_t | S_t = s]$$

作出动作a的行为值函数q_π为

$$q_\pi(s,a)=\mathrm{E}[G_t|S_t=s,A_t=a]$$

于是得出，值函数就是行为值函数对策略的期望：

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

然而，在强化学习的马尔可夫决策过程中，更倾向用其递归形式：

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

应用递归形式，采用**回溯图**的数据结构更有利于在计算机中实现该公式。

写成递归形式，是为了利用**贝尔曼方程**将多阶段决策问题简化成多个单阶段决策问题。于是对某步回溯：

$$\begin{aligned}v_*(s)&=\max_{a\in\mathcal{A}(s)}q_{\pi_*}(s,a)\\&=\max_{a\in\mathcal{A}(s)}\sum_{s',r}p(s',r|s,a)[r+\gamma v_*(s')]\end{aligned}$$

$$q_*(s,a)=\sum_{s',r}p(s',r|s,a)[r+\gamma\max_{a'\in\mathcal{A}(s')}q_{\pi_*}(s',a')]$$