强化学习

菠萝包萝卜

2025 年 9 月 22 日

1 基础概念

1.1 状态 State

Definition 1.1. 状态 (state) 是指智能体相对于环境的一个状态 s_i ，可能状态的全体 $\mathcal{S} = \{s_i\}_i$ 称为状态空间 (全集) state space。

1.2 行动 Action

Definition 1.2. 行动 (action) 是指在每个状态下有一些可以进行的行为，每个这样的行为称为当前状态的一个行动，用字母 a_j 表示；类似地，可能的行动全体 $\mathcal{A}(s_i) = \{a_j\}_j$ 称为状态 s_i 下的行动空间；值得一提的是每个状态下能采取的行动不一定是相同的，所以上面的定义式中是有个 s_i 的自变量的。

1.3 状态转移 State Transition

Definition 1.3. 当我们在一定状态下采取了某种行动，智能体会从当前状态转变为行动后的另一种状态，这样的一个过程就是状态转移；它实际上是定义了智能体和环境的一种交互行为。

状态转移可以用表格来表示，在某个 s_i 下做出了一个行动 a_j ，会得到一个新的 $s \in \mathcal{S}$ ，这样的一个二元函数通过表格变能表出，但是它只能表示确定性的情况，并不泛用.

如此，我们引进了状态转移概率 (State transition probability)，通过概率来表示状态转移。

1.4 策略 Policy

Definition 1.4. 在交互过程中会因不同的行为来产生不同的路径来得到结果，这些诸多路径中孰好孰坏我们暂时是不知道的。智能体会在某个状态选择一个行动以便达到目标，所有这些状态-行动对便组成了策略，换句话说策略会告诉智能体在某个状态下采取什么行动。

为了描述策略，我们使用数理表示，仍然可以利用概率来表示，一般我们用 $\pi(\cdot|\cdot)$ 来表示策略。

当然，由于策略给定了，我们在某状态下做出某种行动的概率也就给定了，这个时候这样的条件概率视为一个二元函数，它可以同时表示确定情况和不确定情况，然后做个表格就能表示出这样的策略全体。

1.5 奖励 Reward

奖励是强化学习中最重要的概念之一。

> **Definition 1.5.** 所谓奖励 (reward)，是一个实数，它是在采取某种行动 (action) 后所得到的一个值，它满足:
> 如果奖励是正的，就代表鼓励这个行为的发生；
> 反之奖励是负的，就代表对这个行为的发生实施惩罚。

Remark. 如果奖励为零代表什么——就是对行为既不鼓励也不惩罚。

能否把正的奖励视为鼓励——事实上是可以的，如果正数代表惩罚，那么我们就要采取措施来得到更少的惩罚。

奖励是一种人机交互的手段，通过它可以引导智能体应该怎么做，不该怎么做。

1.5.1 表格表示法

对于某种状态，采取某种措施会获得一个 reward，我们把这些状态和这些措施列称表格，表格中间填充对应的 reward 值就能表示出这些情况的 reward。

1.5.2 数学表示法——条件概率

如果在状态 s_i ，我们选择了行动 a_j，得到的奖励（由状态和行动决定了）是 r_{cur}，那么我们就有

$$p(r = r_{cur}|s_i, a_j) = 1, p(r \neq r_{cur}|s_i, a_j) = 0$$

Remark. 刚才的说法其实是确定性的奖励；但是我们的奖励可能是随机的，举个例子，当你学习十分刻苦，你会在成绩上体现出 reward，但是这个 reward 具体是多少是不确定。

1.6 轨迹 Trajectory; 回报 Return

> **Definition 1.6.** 轨迹 (trajectory) 是一个状态-行动-奖励的链，它表述了从一个状态经过若干次行动，每次行动都会得到奖励，并逐次转移到另外一个状态的的过程。

> **Definition 1.7.** 回报 (return) 是在某一个轨迹中的所有奖励之和。

回报可以用来评估一个策略是好还是坏。

> **Definition 1.8.** 回报衰减 (discounted return) 试图解决一个问题：当我们达到目标后可能智能体还会有所行动，继续获得奖励，可能会造成回报发散。我们引入衰减率 (discount rate) $\gamma \in (0, 1]$，来对每步行动奖励乘上重复次数次方的衰减率 γ^i ，进行求和后才是衰减回报。

> **Definition 1.9.** 回合 (episode) 它指的是从环境的初始状态开始，智能体与环境交互直到达到某个终止状态的整个过程，对比轨迹，回合是开始到结束的一个完整轨迹，但是轨迹不一定是回合，它可以节选自某个回合。

有些任务是没有结束状态的，我们称为持续性任务 (continuing tasks)；事实上，我们会以一种数学方法把持续性任务转化为回合制任务。

选择一：把目标状态视为吸收态；所谓吸收态，只要智能体达到了吸收态，他就不会再离开这个状态了，或者说没有再可选用的 action，之后所有的 reward 全为 0。

选择二：把目标态视为策略的常态；智能体仍然可以离开目标状态并且在回到目标态的时候获得 +1 的奖励。

1.7 马尔科夫决策过程 MDP

Definition 1.10. 马尔可夫决策过程的关键要素:
首先是三个集合：状态空间 $\mathcal{S}$；行动空间 $\mathcal{A}(s), s \in \mathcal{S}$；奖励空间 $\mathcal{R}(s,a)$.
然后是概率分布：状态转移概率，在状态 s 进行行动 a 的条件下然后转化为状态 s' 的条件概率 $p(s'|s,a)$；奖励概率，在状态 s 进行行动 a 的条件下获得奖励 r 的概率 $p(r|s,a)$
策略：在状态 s，我们采取行动 a 的概率 $\pi(a|s)$
马尔可夫性: 无记忆性

$$p(s_{t+1}|a_{t+1}, s_t, \ldots, a_1, s_0) = p(s_{t+1}|a_{t+1}, s_t)$$
$$p(r_{t+1}|a_{t+1}, s_t, \ldots, a_1, s_0) = p(r_{t+1}|a_{t+1}, s_t)$$

2 贝尔曼公式

2.1 状态值 State Value

一些符号说明，考虑以下的一个单步过程

$$S_t \xrightarrow{A_t} R_{t+1}, S_{t+1}$$

$t, t+1$: 是离散时间实例
S_t : t 时刻的状态
A_t : 状态 S_t 下采取的行动
R_{t+1} : 进行 A_t 后所获得的奖励
S_{t+1} : S_t 进行动作 A_t 后转移的状态
值得一提的是 S_t, A_t, R_{t+1} 都是随机变 (向) 量，这样状态转移的过程又可以用以下的语言表示

$$\begin{aligned} S &\to A_t : \pi(A_t = a|S_t = s) \\ S_t, A_t &\to R_{t+1} : p(R_{t+1} = r|S_t = s, A_t = a) \\ S_t, A_t &\to S_{t+1} : p(S_{t+1} = s'|S_t = s, A_t = a) \end{aligned}$$

在这里，我们假设我们是知道模型的这些概率分布的。

再考虑以下的多步骤轨迹

$$S_t \xrightarrow{A_t} R_{t+1}, S_{t+1} \xrightarrow{A_{t+1}} R_{t+2}, S_{t+2} \xrightarrow{A_{t+2}} R_{t+1}, \ldots$$

衰减的回报是

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$$

它仍然是一个随机变量。

Definition 2.1. 我们称上述衰减回报 G_t 的期望是状态值 (函数):

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

Remark. 状态值是一个 s 的函数，这是由条件期望的定义所决定的；它也受约于策略 π，对于不同的策略，状态值函数可能不一样；它表示的是一个状态的值，如果它越大，说明策略是更优的，因为这意味着可能会获得更多的奖励。

2.2 贝尔曼公式

我们回到 G_t 的表达式

$$\begin{aligned} G_t &= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots, \\ &= R_{t+1} + \gamma(G_{t+1}) \end{aligned}$$

这样我们的状态值也可以是

$$\begin{aligned} v_\pi(s) &= \mathbb{E}[G_t|S_t = s] \\ &= \mathbb{E}[R_{t+1} + \gamma G_{t+1}|S_t = s] \\ &= \mathbb{E}[R_{t+1}|S_t = s] + \gamma \mathbb{E}[G_{t+1}|S_t = s] \end{aligned}$$

一方面前半部分我们可以由全期望公式，其称为即刻的奖励均值

$$\begin{aligned}\mathbb{E}[R_{t+1}|S_t=s]&=\sum_a \pi(a|s)\mathbb{E}[R_{t+1}|S_t=s,A_t=a]\\&=\sum_a \pi(a|s)\sum_r p(r|s,a)r\end{aligned}$$

后半部分称为未来奖励的均值，类似地

$$\begin{aligned}\mathbb{E}[G_{t+1}|S_t=s]&=\sum_{s'}\mathbb{E}[G_{t+1}|S_t=s,S_{t+1}=s']p(s'|s)\\&=\sum_{s'}\mathbb{E}[G_{t+1}|S_{t+1}=s']p(s'|s)\\&=\sum_{s'}v_\pi(s')p(s'|s)\\&=\sum_{s'}v_\pi(s')\sum_a p(s'|s,a)\pi(a|s)\end{aligned}$$

我们对和式进行一个换序就有

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

它便是贝尔曼公式，中括号外是一个策略评估，而中括号内其实是一个动态模型。

我们把上式子的中括号拆开，并且再进行一下变换

$$\begin{aligned}v_\pi(s)&=\sum_a \pi(a|s)[\sum_r p(r|s,a)r+\gamma\sum_{s'}p(s'|s,a)v_\pi(s')]\\&=\sum_a \pi(a|s)\sum_r p(r|s,a)r+\gamma\sum_a\sum_{s'}\pi(a|s)p(s'|s,a)v_\pi(s')\\&=r_\pi(s)+\gamma\sum_{s'}p_\pi(s'|s)v_\pi(s')\end{aligned}$$

这里 $r_\pi(s)=\sum_a \pi(a|s)\sum_r p(r|s,a)r, p_\pi(s'|s)=\sum_a \pi(a|s)p(s'|s,a)$，把诸多的 s_i 的 $v_\pi(s_i)$ 等式拼起来

$$\boldsymbol{v_\pi}=\boldsymbol{r_\pi}+\gamma P_\pi \boldsymbol{v_\pi}$$

其中

$$\begin{aligned}&\boldsymbol{v_\pi}=[v_\pi(s_1),\dots,v_\pi(s_n)]^T\in\mathbb{R}^n\\&\boldsymbol{r_\pi}=[r_\pi(s_1),\dots,r_\pi(s_n)]^T\in\mathbb{R}^n\\&P_\pi\in\mathbb{R}^{n\times n},(P_\pi)_{ij}=p_\pi(s_j|s_i)\text{称为状态转移矩阵}\end{aligned}$$

2.3 利用贝尔曼公式解状态值

给定一个策略，找出相关的状态值就叫策略评估，这是强化学习的一个基础问题，是找到更好的策略的基石。

由于我们的方程组是

$$\boldsymbol{v_\pi}=\boldsymbol{r_\pi}+\gamma P_\pi \boldsymbol{v_\pi}$$

由线性方程组的解法是可以得到

$$\boldsymbol{v_\pi}=(I-\gamma P_p i)^{-1}\boldsymbol{r_\pi}$$

这样的解析解，但是这在实际运算中并不是个容易事，实际用数值逼近的办法来解

$$\boldsymbol{v_{k+1}}=\boldsymbol{r_\pi}+\gamma P_\pi v_k$$

Theorem 2.1. 上述更新的过程使得 $\boldsymbol{v_k}$ 收敛。

Proof. 定义误差 $\boldsymbol{\delta_k} = \boldsymbol{v_k} - \boldsymbol{v_\pi}$，我们只需说明 $\boldsymbol{\delta_k} \to \mathbf{0}$。

我们有 $\boldsymbol{v_{k+1}} = \boldsymbol{\delta_{k+1}} + \boldsymbol{v_\pi}$，和 $\boldsymbol{v_k} = \boldsymbol{\delta_k} + \boldsymbol{v_\pi}$，结合更新式 $\boldsymbol{v_{k+1}} = \boldsymbol{r_\pi} + \gamma P_\pi v_k$ 就能够得到

$$\boldsymbol{\delta_{k+1}} + \boldsymbol{v_\pi} = \boldsymbol{r_\pi} + \gamma P_\pi(\boldsymbol{\delta_k} + \boldsymbol{v_\pi})$$

重写一下就是

$$\boldsymbol{\delta_{k+1}} = -\boldsymbol{v_\pi} + \boldsymbol{r_\pi} + \gamma P_\pi \boldsymbol{\delta_k} + \gamma P_\pi \boldsymbol{v_\pi} = \gamma P_\pi \boldsymbol{\delta}_k$$

重复迭代就是

$$\boldsymbol{\delta_{k+1}} = \gamma^{k+1} P_\pi^{k+1} \boldsymbol{\delta_0}$$

注意到 $P_\pi \mathbf{1} = \mathbf{1}$ ，这样就有 $P_\pi^k \mathbf{1} = \mathbf{1}$ 对任意 $k \in \mathbb{N}$ 成立；另一方面，由于 $|\gamma| < 1$,k 充分大的时候 $\gamma^k \to 0$，由于 $\boldsymbol{\delta_0}$ 是迭代开始既定的，它的各项可以小于一个 M，这样

$$||\boldsymbol{\delta_{k+1}}|| \leq ||M\gamma^{k+1} P_\pi^{k+1} \mathbf{1}|| \to 0$$

□

2.4 行动值 Action Value

Definition 2.2. 所谓行动值 (action value)，它是指智能体从一个状态采取了某个行动后所能得到的平均回报

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

它是元组 (s,a) 的一个二元函数，且依旧受约于 π

我们容易得到

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

也就是说

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

同样回到贝尔曼公式的话可以看出

$$q_\pi(s,a) = \sum_r p(r|s,a) r + \gamma \sum_{s'} p(s'|s,a) v_\pi(s')$$