第二章

强化学习与有监督的学习的对比

特性	强化学习	有监督学习
核心反馈机制	**评估性反馈** 这种反馈只表明刚才做的动作“好”还是“坏”（给多少分），但不告诉你是好在哪里，也不告诉你最好的动作是什么	**指导性反馈** 这种反馈直接给出“正确动作”是什么（标准答案），并且这个正确答案与你刚才采取的动作无关
数据来源	**通过交互产生** 智能体必须主动与环境交互，通过“试错”来产生数据和经验	**外部提供** 学习基于外部监督者提供的、既有的“带标注训练集”
是否依赖当前动作	**依赖** 反馈取决于你采取了什么动作。采取不同的动作，得到的反馈（收益）是不同的 。	**不依赖** 正确的标签（Label）是客观存在的，无论你预测什么，标准答案都是同一个 。
主要目标	**最大化收益信号** 寻找能让长期总收益最大化的策略 。	**推断与泛化** 学习如何根据训练集，对新的、未见过的情境做出正确的分类或预测 。
试探的需求	**必须试探** 因为不知道哪个动作最好，必须主动尝试不同的动作来发现高收益的选项 。	**无需试探** 不需要通过试错来发现答案，而是通过修正误差来逼近老师给出的标准答案。
典型应用场景	交互式问题、未知领域（无法获得所有正确示例）、序列决策问题	模式分类、人工神经网络、系统辨识等有明确正确答案的任务

2.1 一个k臂赌博机问题

问题背景：一个有k个控制杆的赌博机，拉动不同的控制杆收益满足不同的分布

目的：在某一段时间内最大化总收益的期望

> 为什么是总收益的期望而不是总收益：
>
> 因为我们无法把握收益，只能把握收益的期望。并且根据大数定律。我们最大化**期望**，是因为在概率环境中，**最大化期望是实现最大化未来实际总收益的唯一科学路径**。期望是因，实际收益是果。

名词解释：

- t时刻选择的动作记为A_t,相应的收益记为R_t
- k个动作，每个动作在被选中时都有一个期望收益/平均收益，这个收益就叫做这个动作的**价值**,动作a的价值记为$q_*(a)$定义为：

$$q_*(a) = E[R_t|A_t = a] \tag{1}$$

这个一般就是理论上的值，实际上我们也没法求这个的期望，只能对他进行一个估计。我们将动作a在t时刻的价值估计为$Q_t(a)$

我们希望这估计值越接近$q_*(a)$越好。任意时刻都会有价值估计最高的动作，而这个动作就被成为**贪心动作**。

- 开发：选择贪心动作，开发出更多的信息
- 试探：特意去选择那些非贪心的、甚至看起来较差的动作 。收集更多信息，改善对这些动作价值的**估计**

> 如何平衡好“开发”和“试探”这很重要。

2.2 动作-价值方法

在 k 臂赌博机问题中，如何具体地去**估计每个动作的价值**，以及如何利用这些估计值来选择动作

估计动作价值的方法

采样平均方法估计动作价值:

$$Q_t(a) = \frac{t\text{时刻之前执行动作}a\text{获得的总收益}}{t\text{时刻之前执行动作}a\text{的总次数}} = \frac{\sum_{i=1}^{t-1} R_i \cdot \mathbb{I}_{A_i=a}}{\sum_{i=1}^{t-1} \mathbb{I}_{A_i=a}} \tag{2}$$

其中，$\mathbb{I}_{\text{predicate}}$ 表示随机变量，当 predicate 为真时其值为 1，反之为 0。当分母为 0 时，我们将 $Q_t(a)$ 定义为某个默认值，例如 $Q_t(a) = 0$ 。当分母趋向无穷大时，根据大数定律，$Q_t(a)$ 会收敛到 $q_*(a)$。

选择动作的方法

1. 贪心方法

   选择当前估计价值最高的那个动作

$$A_t \doteq \text{argmax}_a Q_t(a) \tag{3}$$

2. ϵ-贪心方法
   - 以概率 $1-\epsilon$选择估计价值最高的那个动作
   - 以概率 ϵ随机选择一个动作

2.3 10臂赌博机问题

这一节是实验，要理解这个实验图像是这样的原因

2.4 增量实现

对于传统的求解方式：

$$Q_n = \frac{\sum_{i=1}^{n-1} R_i}{n-1} \tag{4}$$

这种传统的方式的弊端是需要存储历史数据

- 占内存
- 计算效率低

现在我们采用增量的方式来实现，推导如下

$$Q_{n+1} = \frac{\sum_{i=1}^{n} R_i}{n} \tag{5}$$

这个的意思是当动作a被选择n+1次之后动作a的价值的估计值。现在我们思考怎么能在Q_{n+1}的式子当中放上Q_n

$$\begin{aligned} nQ_{n+1} &= (n-1)Q_n + R_n \\ Q_{n+1} &= (1-\frac{1}{n})Q_n + \frac{1}{n}R_n \\ Q_{n+1} &= Q_n + \frac{1}{n}[R_n - Q_n] \end{aligned} \tag{6}$$

我们观察这个式子，它的形式我们可以理解为

$$\text{新的估计值} = \text{旧的估计值} + \text{步长} \times [\text{目标} - \text{旧的估计值}] \tag{7}$$

- “目标”= 这次更新时，我们希望当前估计去逼近的那个值。
- 这个里面$\frac{1}{n}$我们可以理解为步长，更普适的式子是$Q_{n+1} = Q_n + \alpha_t(a)(R_n - Q_n)$，步长可以随着时间进行变化.

2.5 跟踪一个非平稳过程

- 平稳过程：环境不随时间变化——对应收敛，我希望估计 Q 最终稳定到一个固定值
- 非平稳过程：环境不随时间变化——对应跟踪，我希望估计 Q 能跟着真实值移动（哪怕永远有波动）

前边我们使用的步长$\alpha = \frac{1}{n}$不适合于固定步长，因为学到后面你"变得很固执"：新数据几乎推不动 Q。在非平稳环境里，希望**近期数据更重要**，否则你会一直被"很久以前"的回报拖着走，跟不上变化。那么我们就要采用固定的步长。

$$Q_{n+1} = Q_n + \alpha[R_n - Q_n] \tag{8}$$

为什么固定步长之后能更加反应近期数据？原因是Q_{n+1}是由R_i和Q_1的加权平均值。并且越近期的数据权重越大。西现在我们可以进行计算,通过迭代$Q_{n+1} = (1-\alpha)Q_n + \alpha R_n$

$$\begin{aligned} Q_{n+1} &= \alpha R_n + (1-\alpha)\alpha R_{n-1} + (1-\alpha)^2 \alpha R_{n-2} + \cdots + (1-\alpha)^{n-1}\alpha R_1 + (1-\alpha)^n Q_1 \\ Q_{n+1} &= (1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i \end{aligned} \tag{9}$$

> 在平稳条件下，若对每个动作 a，它被选择次数 $N_t(a) \to \infty$，
> 并且该动作的步长序列 $\{\alpha_n(a)\}$ 满足
>
> $$\sum_{n=1}^{\infty} \alpha_n(a) = \infty, \qquad \sum_{n=1}^{\infty} \alpha_n^2(a) < \infty, \tag{10}$$
>
> 则 $Q(a)$ 的随机逼近更新在常见假设下可保证收敛（到相应的期望值/不动点）。
>
> - 条件1：不能衰减得太慢，否则收敛不到呢就不动了
> - 条件2：不能让噪声抖动太大，要不不好收敛。

2.6 乐观初始值

之前我们讲的ε**-贪心策略**相比于单纯的**贪心策略**更适合试探。现在我们不使用ε**-贪心策略**也可以进行试探。

假如我们在设置初始值Q_1的过程中将初始值设置地足够大，那么智能体在该策略下将会进行更多的**试探**

> 实验结果表明，采用乐观初始值的方法虽然在早期收敛较慢，但能够促使智能体进行更充分的探索，从而在长期内获得更优的策略性能。

2.7 基于置信度上界的动作选择

> 2.6节的乐观初始值探索主要发生在**初期**一旦估计"冷静下来"，探索就**自然消失**。对 **非平稳问题** 或 **长期不确定性** 并不理想，强烈依赖 Q_1 的设置于是作者在 2.7 节提出一个更"理性"的问题：**能不能根据"我们对某个动作有多不确定"，来决定是否再去探索它?**

本节就是在讲一个公式：

$$A_t \doteq \underset{a}{\operatorname{argmax}} \left[Q_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}} \right] \tag{11}$$

- 其中c是一个大于0的数，它控制着试探程度

2.8 梯度赌博机算法

> 梯度赌博机算法不再估计具体的价值，而是学习对每个动作的**数值化偏好**，它唯一的意义在于相对大小。

- **偏好函数** $H_t(a)$：算法为每个动作 a 维护一个偏好值 $H_t(a)$。这个值本身并没有具体的"收益"含义（比如它不是说这个动作能赚多少分），它唯一的意义在于**相对大小**
- 偏好越高，被选概率越大
- 动作选择的概率为$\pi_t(a)$

> **下面是偏好函数和动作选择的概率之间的关系：**
>
> 为了把"偏好"转化为具体的"选择概率"，算法使用了 **Softmax 分布**。
>
> 动作 a 在时刻 t 被选中的概率 $\pi_t(a)$ 计算公式为：
>
> $$Pr\{A_t = a\} \doteq \frac{e^{H_t(a)}}{\sum_{b=1}^{k} e^{H_t(b)}} \doteq \pi_t(a) \tag{12}$$
>
> 这意味着：

- 所有动作的概率之和为 1。
- 偏好 $H_t(a)$ 越大，分子 $e^{H_t(a)}$ 越大，该动作被选中的概率 $\pi_t(a)$ 就越高。

本节的核心是偏好函数$H_t(a)$的更新规则：

当选择了动作A_t并获得了收益 R_t 后

对于被选中的动作 A_t：

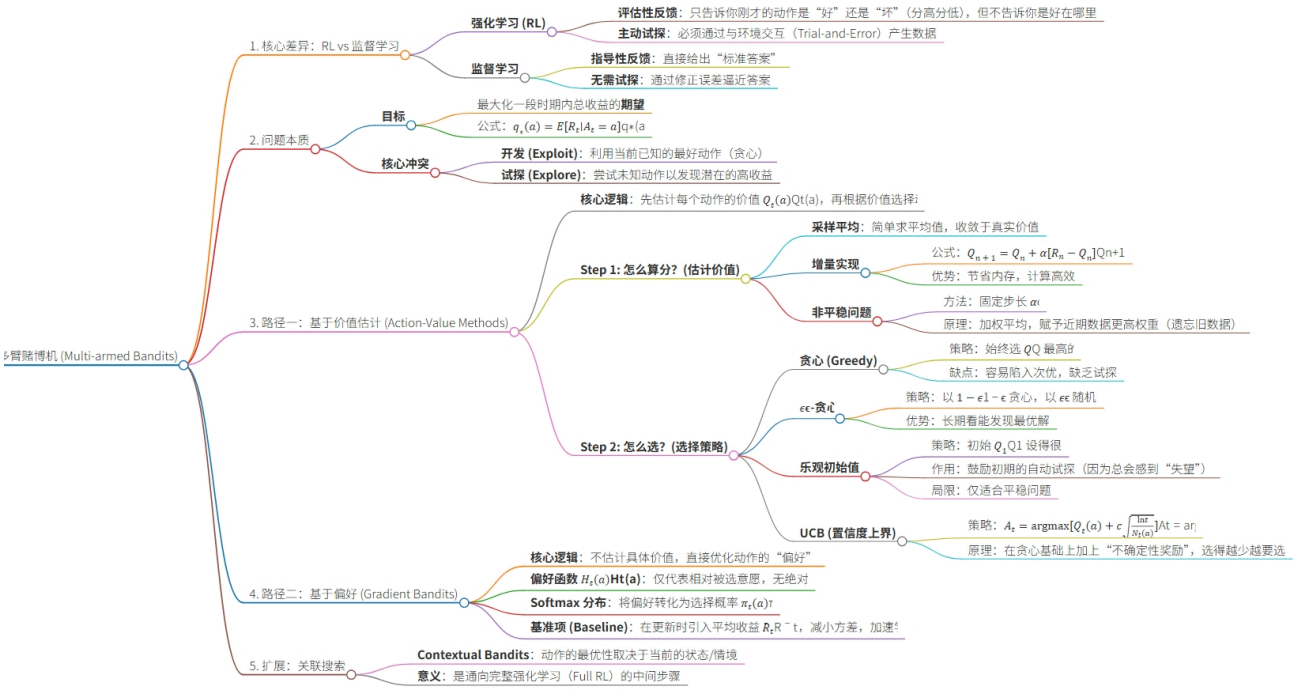
$$H_{t+1}(A_t) \doteq H_t(A_t) + \alpha(R_t - \overline{R}_t)(1 - \pi_t(A_t)) \tag{13}$$

对于未被选中的动作 a ($a \neq A_t$)：

$$H_{t+1}(a) \doteq H_t(a) - \alpha(R_t - \overline{R}_t)\pi_t(a) \tag{14}$$

2.9 关联搜索

这一节主要讲的就是最优动作会根据环境的不同而变换，所以针对这种情况改变相应的策略，但是具体怎么做本节还是没讲。



个人反思：这个东西的最终目的是选择动作，而选择动作有不同的策略（贪心，ε贪心）

其中一个重要的选择动作的方法就是估计动作的价值。（当然还有一种是通过偏好，那个和这个的“打法”不太一样）