强化学习实验报告

多臂赌博机与gymnasium

张恒硕

2212266

人工智能学院

2025 年 3 月 6 日

南開大學
Nankai University

目 录

图 片

表 格

1 实验目的

1. 以多臂赌博机程序为例，了解三种探索-利用平衡策略。
2. 将多臂赌博机程序由三臂改成四臂，进行调试，对比三种动作选择策略。
3. 挑选一个gymnasium包中的环境，分析其MDP元素。

2 基础知识

2.1 探索-利用平衡策略

EPSILON-GREEDY策略

以ϵ的概率进行随机探索，以$1-\epsilon$的概率选择当前估计都最优动作。
特点是简单高效，但依赖固定ϵ。

UPPER CONFIDENCE BOUND(UCB)策略

通过置信上限平衡探索和利用，有限不确定性高的动作。

$$UCB(a) = Q(a) + C\sqrt{\frac{\ln N}{n(a)}}$$

其中，C是控制探索强度的超参数，N是当前轮数，$n(a)$是动作a被选次数。
特点是自适应平衡探索与利用。

BOLTZMANN策略

基于温度参数生成概率分布。

$$P(a) = \frac{e^{Q(a)/\tau}}{\sum_{a'} e^{Q(a)/\tau}}$$

其中，τ是温度参数，其偏大时偏向探索，偏小时趋向贪心。
特点是温度敏感，可以动态调整探索强度。

2.2 马尔可夫决策过程（MDP）

1. 状态s：智能体当前的处境或信息。
2. 动作a：智能体在每个状态下可执行的操作。
3. 转移概率p：动作a导致从s转移到s'的概率，
   $P(s' \mid s, a) = \mathbb{P}[S_{t+1} = s' \mid S_t = s, A_t = a]$。
4. 奖励r：在状态s执行动作a后，智能体获得的即时奖励。
5. 折扣因子γ：衡量未来奖励的重要性。

3 多臂赌博机

3.1 代码改动

只添加了一个新的臂，并修改了相应代码。在设置q值时，四种动作依次为1、1.5、2、2.5。改动较为简单，这里不做分析。

3.2 实验结果与分析

四臂赌博机的实验结果如下图图1 on the following page和下表表1 on the next page：

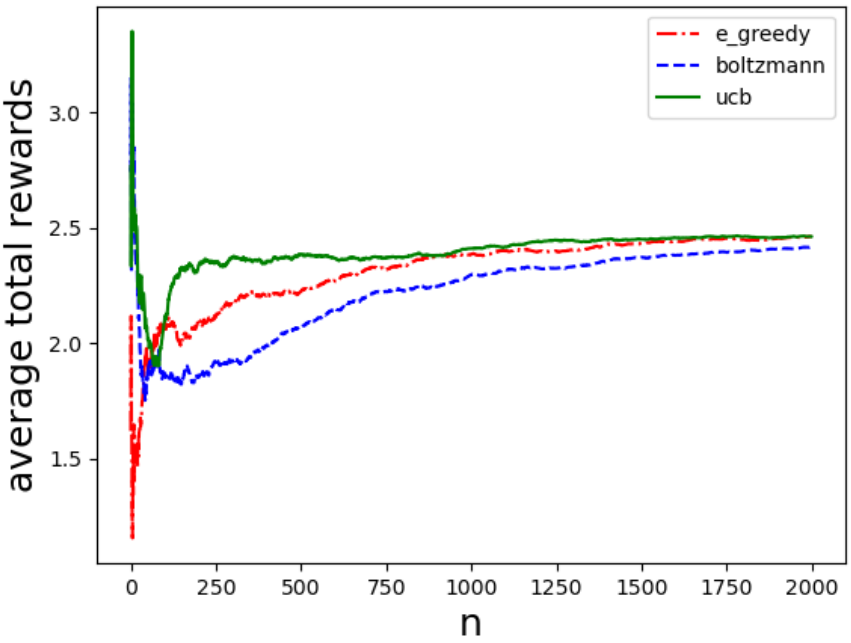

图 1: 三种策略对比

表 1: 三种策略的 Q 值估计对比

动作	实际值	Epsilon-Greedy	UCB	Boltzmann
a_1	1	0.7960018931	1.0158071311	0.795347734
a_2	1.5	1.4402378437	1.5875736916	1.432605369
a_3	2	1.6579154524	2.0985780328	1.957860961 02
a_4	2.5	2.5171363619 08	2.4815038519 45	2.5063682117 95

可以获得以下信息：

1. 三种方法的估计值都近似实际值，但在近似程度上有差异。
2. Epsilon-Greedy中，非最大奖励动作的三种动作的选择次数接近，这是不同于另外两种策略的，这与它探索是随机选取的方式有关。

3. UCB是三种方法中收敛最快的，可以发现，它选取低奖励动作的概率较小，这与它的选取公式有关。

4. Boltzmann的收敛速度和结果都是最差的，这与τ值的选取有关，若果能动态地实现温度曲线式的参数变化，将能取得更好的结果。

4 GYMNASIUM

加载了$MountainCarContinuous-v0$环境，其对应的源码位于continuous_mountain_car.py。这是一个小车爬坡的模拟过程，通过设定小车的运动，使其爬上正确方向的坡并到达指定位置。

4.1 源码分析

状态空间

```
self.observation_space = spaces.Box(
    low=self.low_state, high=self.high_state, dtype=np.float32
)
where:
self.low_state = np.array([-1.2, -0.07])
self.high_state = np.array([0.6, 0.07])
```

本部分定义小车可行的位置空间为$[-1.2, 0.6]$，速度空间为$[-0.07, 0.07]$。

动作空间

```
self.action_space = spaces.Box(
    low=self.min_action, high=self.max_action, shape=(1,), dtype=np.float32
)
where:
self.min_action = -1.0, self.max_action = 1.0
```

本部分限制小车动作在$[-1.0, 1.0]$范围内。

奖励函数

```
reward = 100.0 if terminated else -math.pow(action[0], 2) * 0.1
```

本部分设定了小车运动过程中的奖励。当位置$\geqslant 0.45$且速度$\geqslant$ goal_velocity时，奖励为$+100$，成功抵达目标。 在抵达前，每一步奖励为$-0.1(\text{action}^2)$，其为负值惩罚，随动作数增加，以督促小车尽快抵达目标。

状态转移

本部分定义小车的状态转移为确定性转移，其速度和位置更新公式如下:

$$\text{velocity}_{t+1} = \text{clamp}(\text{velocity}_t + \text{force} * \text{power} - 0.0025 * \cos(3 * \text{position}_t), [-0.07, 0.07])$$

$$\text{position}_{t+1} = \text{clamp}(\text{position}_t + \text{velocity}_{t+1}, [-1.2, 0.6])$$

另外，还对碰撞进行了处理：若位置达到边界且速度方向与边界相反，则速度置零。

折扣率 源码中未显式定义。

初始状态

```
self.state = np.array([
    self.np_random.uniform(low=low, high=high),
    0.0
])
```

本部分初始化了小车状态，其位置均匀分布在$[-0.6, -0.4]$， 速度固定为0。

4.2 实验结果与分析

运行过程如下图图2所示：

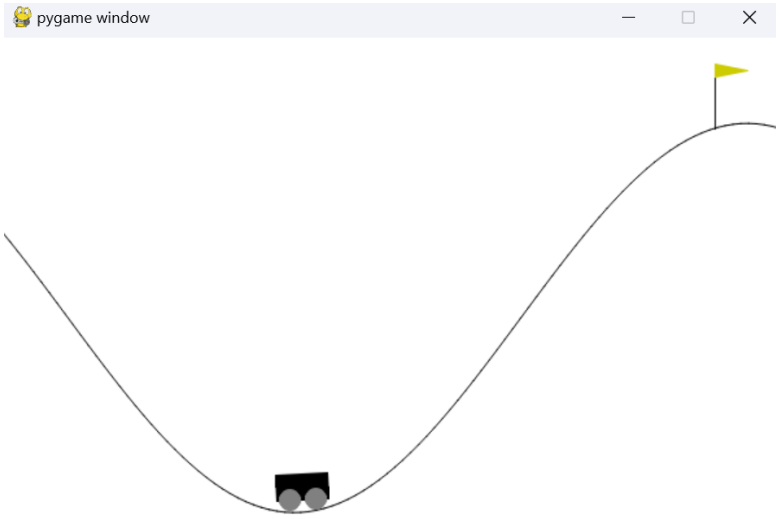

图 2: 小车爬坡（MountainCarContinuous-vo）

结合代码和实验过程可以获得以下信息：

1. 奖励与折扣率：本实验为使小车尽快抵达目标，设定过程动作的奖励为负，且随动作数增加，这是一种变向的折扣率。
2. 违规动作处理：本实验中对于碰壁有所处理，属于理论课中介绍的禁止与惩罚两种方式的前者。
3. 合理设置状态空间和动作空间：如本实验，对小车的速度上限有所限制，这对保证实验的连续性和合理性有益。

5 实验总结

通过本次实验的学习，掌握了两方面知识。一方面是从多臂赌博机出发，了解了三种探索-利用平衡策略，这是强化学习中动作选择的基础策略。另一方面，通过阅读gymnasium包中特定环境的源代码，深化了对马尔可夫链的认识，并初步了解了这个常用的强化学习包。