练习 5.12：赛车轨道 (Racetrack) ——离策略蒙特卡洛控制解题报告

作者：JEFF

2025 年 3 月 3 日

1 引言

本报告讨论了《*Reinforcement Learning: An Introduction*》（Sutton & Barto）第五章练习 5.12 “Racetrack” 任务的实现方法。该任务要求我们在离散化的赛车轨道上驾驶赛车，以最短时间到达终点线，同时避免冲出赛道边界。为此，我们采用了**离策略蒙特卡洛控制（Off-policy Monte Carlo Control）**的方法，对赛车进行策略改进与价值估计。

2 问题描述与环境设定

2.1 网格与赛车状态

题目将赛道抽象为一个离散网格 grid，其中每个网格单元可属于以下类型：

- `CELL_EDGE`（0）：赛道边界，赛车若到达此处即视为越界。
- `CELL_TRACK`（1）：赛道区域，可安全行驶。
- `CELL_START_LINE`（2）：起始线所在区域，赛车的初始位置从此区域随机选取。
- `CELL_FINISH_LINE`（3）：终点线区域，赛车驶过即回合结束。

赛车的状态由位置和速度组成：

$$S_t = \big(\text{position} = (r, c),\ \text{speed} = (v_x, v_y)\big).$$

其中，速度的两个分量 v_x, v_y 均限制在区间 $[0, 4]$（或题目所规定的最大值）。此外，赛车若因加速度调整导致速度分量超出该区间，则需将其截断（clamp）。

2.2 动作与随机噪声

每个时间步，赛车可以执行 9 种动作 (a_x, a_y)，其中

$$a_x, a_y \in \{-1, 0, 1\}.$$

该动作表示对当前速度在水平方向和垂直方向各增加 $-1, 0, +1$。根据题目描述，为增加难度，还需要在每个时间步以 0.1 的概率将加速度置为 $(0, 0)$，即本来想加速也可能失败，从而使赛车不发生速度变化。

2.3 奖励设计

为了鼓励赛车尽快到达终点，并惩罚越界、拖延时间等行为，本实验常设定：

$$r_t = \begin{cases} -1, & \text{若赛车在赛道内正常移动;} \\ 0, & \text{若赛车穿过终点线;} \\ -100, & \text{若赛车越界;} \end{cases}$$

若我们还考虑回合超时（例如超过 10,000 步）则给一次性额外惩罚 -200 并强制结束回合。

3 离策略蒙特卡洛控制原理

3.1 目标策略与行为策略

在离策略（Off-policy）控制中，我们将**目标策略** π 与**行为策略** b 区分开来：

- **目标策略** π：我们想要学习和评估的策略，通常是一个确定性贪心策略，即对动作价值函数 $Q(s, a)$ 取最大值的动作。
- **行为策略** b：实际在环境中产生数据的策略，需要保证对所有可能动作都有非零概率（覆盖性），常见做法是纯随机策略或 ϵ-soft 策略。

在本练习中，π 约为 “对当前 Q 贪心”，而 b 选为纯随机（9 种动作等概率），使得所有动作都能被探索到。

3.2 加权重要性采样（Weighted Importance Sampling）

离策略蒙特卡洛方法依赖于重要性采样（*Importance Sampling*）来对目标策略的价值进行无偏估计。设一条经历为

$$S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T, S_T,$$

其在行为策略 b 下发生的概率为

$$P_b(\tau) = \prod_{t=0}^{T-1} b(A_t \mid S_t),$$

而在目标策略 π 下发生的概率则为

$$P_\pi(\tau) = \prod_{t=0}^{T-1} \pi(A_t \mid S_t).$$

对于该条轨迹，我们定义重要性采样比率为

$$W_t \;=\; \prod_{k=0}^{t-1} \frac{\pi(A_k \mid S_k)}{b(A_k \mid S_k)}.$$

若在本实验中，b 是 9 种动作均匀随机，即 $b(a \mid s) = \frac{1}{9}$，而 π 对唯一最优动作的概率为 1，其余为 0，则

$$\frac{\pi(A_k \mid S_k)}{b(A_k \mid S_k)} = \begin{cases} 9, & \text{若} A_k = \arg\max_a Q(S_k, a), \\ 0, & \text{否则.} \end{cases}$$

这导致在回溯时，一旦发现行为策略所选动作与目标策略不符，后续权重即为 0，可直接 break 结束回溯。

3.3 价值函数更新

在蒙特卡洛框架下，我们在每条轨迹结束后计算回报

$$G_t = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{T-1-t} R_T.$$

然后对 $Q(S_t, A_t)$ 进行更新：

$$Q(S_t, A_t) \;\leftarrow\; Q(S_t, A_t) \;+\; \alpha\, W_t \,\big(G_t - Q(S_t, A_t)\big),$$

其中 $\alpha = \frac{1}{C(S_t, A_t)}$（或类似自适应步长），$C(S_t, A_t)$ 用来累积权重。

4 算法流程与思路

4.1 主要步骤

1. **初始化**：将动作价值函数 $Q(s, a)$ 设为某个初始值（如 -150），并初始化计数器 $C(s, a)$。

2. **重复若干回合**：

(a) 从**起始线**随机选一个位置作为赛车初始位置，并将速度置为 0。

(b) 按照**行为策略** b（纯随机）进行交互，直至**到达终点**或**超时**或**回到起点继续**（若撞墙）。记录状态、动作、奖励序列。

(c) 对该回合的轨迹从后往前计算回报 G，并根据重要性采样比率 W 更新 Q：

$$\begin{aligned} G &\leftarrow \gamma G + R_t, \\ C(S_t, A_t) &\leftarrow C(S_t, A_t) + W, \\ Q(S_t, A_t) &\leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)}\big(G - Q(S_t, A_t)\big). \end{aligned}$$

(d) **改进目标策略** π：对每个状态 s，令

$$\pi(s) = \arg\max_a \; Q(s, a).$$

(e) 若本次回合中动作 $A_t \neq \pi(S_t)$，则停止对之前时刻的回溯更新（因为后续权重将变为 0）。

(f) $W \leftarrow W \times \dfrac{\pi(A_t \mid S_t)}{b(A_t \mid S_t)}$，若动作匹配则乘 9，否则为 0。

4.2 轨迹插值检测与回合超时

本题在赛车位置更新时，需要对从 (r, c) 移动到 $(r - v_x, c + v_y)$ 的整段路径进行离散检查，防止“跳格”越过边界或终点。若在中途检测到越界或撞墙，则将赛车重置到起始线；若检测到穿过终点线，则本回合结束。

此外，为避免某些回合无限拖延，可设置最大步数（如 10,000 步）限制，一旦超过此数仍未到终点，则给予一次性负奖励（如 -200）并结束回合。

5 实验结果与结论

通过上述离策略蒙特卡洛控制算法，赛车最终会学到一条**近似最优的驾驶策略**，在较少时间步内成功到达终点。若在网格轨道 `grid1` 和 `grid2` 上分别训练并可视化轨迹，可观察到赛车在多次探索后逐渐找到绕过弯道、且避免出界的优质路线。

在实际实现中，纯随机行为策略的探索效率并不算高，但仍可在足够多回合下收敛到较优解。若要进一步提升效率，可改用更温和的 ϵ-greedy 策略或引入时间差分方法（Sarsa、Q-learning 等）。

6 参考文献

- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2nd Edition).

- 本课程作业相关说明与提示。