强化学习基础篇（十五）蒙特卡洛预测

1、 Model-free方法

通过贝尔曼方程求解最优策略π^*有3种基本方法：动态规划法、蒙特卡洛法和时间差分法。前面我们介绍了如何利用动态规划法去求解环境知识完备（即马尔可夫决策过程已知）的强化学习任务。简而言之，首先通过策略评估计算给定策略π的优劣程度，然后采用策略迭代算法获得基于策略π的最优价值函数$v_\pi^*(s)$，并根据最优价值函数$v_\pi^*(s)$确定最优策略π^*；出于效率的考虑，也可以采用值迭代算法来获得最优价值函数$v_\pi^*(s)$和最优策略π^*。
在实际任务中，环境知识完备性这一先决条件较难满足，也就意味着大量的强化学习任务难以直接采用动态规划法进行求解。对于环境知识不完备的MDP，即转移矩阵P以及奖励R未知的研究领域称为无模型（Model-Free）方法。Model-free方法的基本方法就是使用蒙特卡洛法（Monte Carlo，MC）和时间差分法（Temporal-difference，TD）。

此外Model-free方法可以分为两个大方面：预测（prediction）与控制（Control）

- Model-free预测是评估一个未知MDP的值函数。
- Model-free控制是优化一个未知MDP的值函数。

本文会主要介绍蒙特卡洛预测方法。

2、什么是蒙特卡洛方法

“蒙特卡洛”这一名字来源于摩纳哥的城市蒙特卡洛（MonteCarlo）。该方法由著名的美国计算机科学家冯·诺伊曼和S.M.乌拉姆在20世纪40年代第二次世界大战中研制原子弹（"曼哈顿计划”）时首先提出。
蒙特卡洛法是一种基于采样的算法名称，依靠重复随机抽样来获得数值结果的计算方法，其核心理念是使用随机性来解决原则上为确定性的问题。通俗而言，蒙特卡洛法采样越多，结果就越近似最优解，即通过多次采样逼近最优解。
举个简单的例子。去果园摘苹果，规则是每次只能摘一个苹果，并且手中只能留下一个苹果，最后走出果园的时候也只能带走一个苹果，目标是使得最后拿出果园的苹果最大。可以达成这样一个共识：进入果园后每次摘一个大苹果，看到比该苹果更大的则替换原来的苹果。基于上述共识，可以保证每次摘到的苹果都至少不比上一次摘到的苹果小。如果摘苹果的次数越多，挑出来的苹果就越大，但无法确保最后摘到的苹果一定是最大的，除非把整个果园的苹果都摘一遍。即尽量找较大的，但不保证是最大的。采样次数越多，结果就越近似最优解，这种方法就属于蒙特卡洛法。\

蒙特卡洛法能够处理免模型的任务，究其原因是无须依赖环境的完备知识（Environment backup），只需收集从环境中进行采样得到的经验轨迹（Experience episode），基于经验轨迹集数据的计算，可求解最优策略。

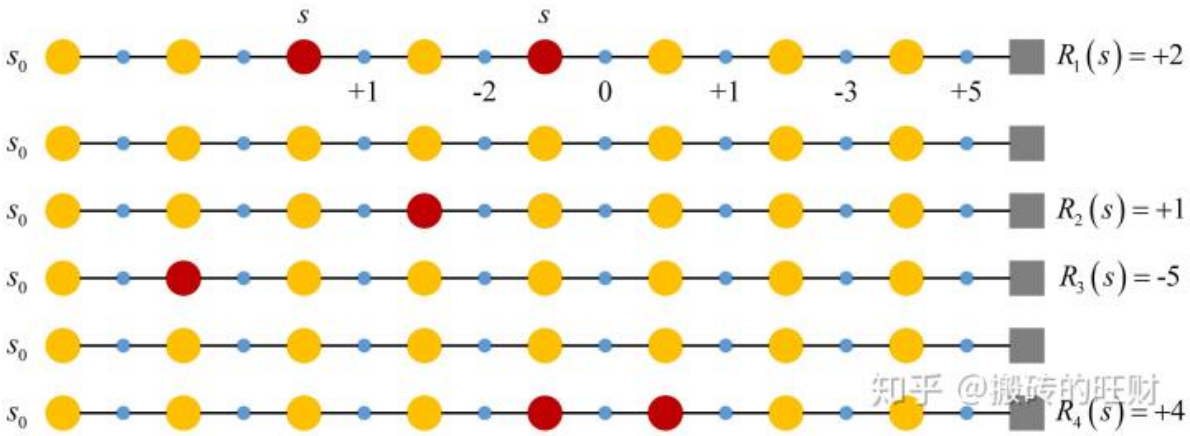
蒙特卡洛在强化学习中应用的核心主要包含以下几点：

- MC方法是直接从经验轨迹当中直接进行学习。
- MC方法是一种model-free方法，即没有MDP的转移概率P以及奖励R的先验知识。
- MC方法从完整的经验轨迹中学习，不使用bootstrapping方法。
- MC方法简单得使用这个思想：价值=平均回报。
- 缺点在于只能应用于一定有终结点的按幕分的MDP过程。

经验轨迹（Episodes）

经验轨迹是智能体通过与环境交互获得的状态、动作、奖励的样本序列。

举个例子：



每条经验轨迹（episode）就是一条从起始状态到结束状态的经历。例如在走迷宫，一条episode就是从你开始进入迷宫，到最后走出迷宫的路径。首先我们要得到的是某一个状态s的平均收获。所以我们说的episode要经过状态s。所以上图中第二条和第五条路径没有经过状态s，对于s 来说就不能使用它了。完整的经验轨迹不要求起始状态一定是某一个特定的状态，但是要求智能体最终进入环境认可的某一个终止状态。理论上完整的状态序列越多，结果越准确。

有终结点的按幕分的MDP过程

无论采用哪个策略，都会在有限时间内到达终点并获得回报。就像上面那个图，每条样本都会最终到达终点。现实中，我们的棋类游戏，都会在有限步数以后达到输赢或者平局的结果并获得相应的回报。

价值=平均回报

其实从字面理解就是求均值的意思。就是状态s在每一个样本中收获的回报均值。

比如，现评估某状态s的价值函数。我们采样了两个经验轨迹，从一个经验轨迹里面得到的回报是5，然后下一个经验轨迹里面的得到的回报是7，我们可以从起始状态来评估此状态的价值函数为：

$$\frac{5+7}{2}=6$$

3. 蒙特卡洛策略评估

蒙特卡洛法采用时间步有限的、完整的经验轨迹，其所产生的经验信息可推导出每个状态的平均奖励，以此来代替奖励的期望（即目标状态值）。换言之，在给定的策略π下，蒙特卡洛法从一系列完整的经验轨迹中学习该策略下的状态值函数$v_\pi(s)$。
当模型环境未知（Model-free）时，智能体根据策略π进行采样，从起始状态s_0出发，执行该策略T步后达到一个终止状态s_T，从而获得一条完整的经验轨迹。

$$s_0,a_0,r_1,s_1,a_1,r_2,\ldots,s_{T-1},a_{T-1},r_T,s_T\sim\pi$$

对于t时刻的状态s，未来折扣累积奖励为：

$$G_t=r_t+\gamma r_{t+1}+\ldots+\gamma^{T-1}r_{t+T}$$

蒙特卡洛法利用经验轨迹的平均未来折扣累积奖励G作为状态值的期望。

$$G=average(G_1+G_2+\ldots+G_T)$$

而强化学习的目标是求解最优策略π^*，得到最优策略的一个常用方法是求解状态值函数$v_\pi(s)$的期望。如果采样的经验轨迹样本足够多，就可以精确估计出状态s上下遵循策略π的期望，即状态值函数$v_\pi(s)$。

$$v_\pi(s) = \mathbb{E}[G|s \in S]$$

当根据策略π收集到的经验轨迹样本趋于无穷多时，得到的状态值$v_\pi(s)$也就无限接近于真实的状态值。

4.First Visit以及Every Visit蒙特卡洛评估方法

假设智能体收集了大量基于某一策略π下运行到状态s的经验轨迹，就可直接估计在该策略下的状态值$v(s)$。然而在状态转移过程中，可能发生一个状态经过一定的转移后又一次或多次返回该状态的情况，即状态发生多次重复，这会给实际的状态值估算带来噪声。
例如，羊群在A地吃草为状态s_A，在B地吃草为状态s_B，显然羊群会根据草地的肥沃程度在不同的地方迁徙吃草，但是会出现羊群重复回到同一个地方吃草的情况，例如重新回到A地吃草，即重复返回到状态s_A。
因此，在一个经验轨迹里，需要对同一经验轨迹中重复出现的状态进行处理，主要有如下两种方法，一种是首次访问蒙特卡洛策略评估（First-Visit Monte-Carlo Policy Evaluation）另一种是每次访问蒙特卡洛策略评估 （Every-Visit Monte-Carlo Policy Evaluation）。

首次访问蒙特卡洛策略评估（First-Visit Monte-Carlo Policy Evaluation）

在给定一个策略，使用一系列完整经验轨迹评估某一个状态s时，对于每一个经验轨迹，仅当该状态s首次出现的时间t列入计算：

- 状态出现的次数加1： $N(s) \leftarrow N(s) + 1$
- 总回报更新： $S(s) \leftarrow S(s) + G_t$
- 使用平均回报更新值函数： $V(s) = S(s)/N(s)$
- 由大数定理可以得到，当$N(s)$趋于无穷大时： $V(s) \rightarrow v_\pi(s)$

每次访问蒙特卡洛策略评估（First-Visit Monte-Carlo Policy Evaluation）

在给定一个策略，使用一系列完整经验轨迹评估某一个状态s时，对于每一个经验轨迹，仅当该状态s每次出现在状态转移链时，例如，一次在时刻t_1，一次在时刻t_2，则两次对应的G_{t_1},G_{t_2}都应该纳入对$v(s)$的计算

- 状态出现的次数加1： $N(s) \leftarrow N(s) + 1$ (每次访问状态s)
- 总回报更新： $S(s) \leftarrow S(s) + G_t$(每次访问状态$s$)
- 使用平均回报更新值函数： $V(s) = S(s)/N(s)$
- 由大数定理可以得到，当$N(s)$趋于无穷大时： $V(s) \rightarrow v_\pi(s)$

5.累进更新平均值 （Incremental Mean）

不论是First-Visit还是Every-Visit，在计算回报均值时，都是利用总回报除以状态s的总访问次数的，我们能否对均值进行增量式的求取?

这里假设每次的回报为x_j，在k次后平均回报为：

$$\mu_k = \frac{1}{k}\sum_{j=1}^{k} x_j$$

在$k-1$次后平均回报为：

$$\mu_{k-1} = \frac{1}{k-1}\sum_{j=1}^{k-1} x_j$$

这种增量式的推导过程可以按照如下过程进行：

$$\begin{aligned}\mu_k &= \frac{1}{k}\sum_{j=1}^{k} x_j \\ &= \frac{1}{k}(x_k + \sum_{j=1}^{k-1} x_j) \\ &= \frac{1}{k}(x_k + (k-1)\mu_{k-1}) \\ &= \mu_{k-1} + \frac{1}{k}(x_k - \mu_{k-1})\end{aligned}$$

6. 蒙特卡洛累进更新(Incremental Monte-Carlo Updates)

对于一系列经验轨迹：$S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T, S_T$

按照更新平均值的方法可以进行如下累进更新响应的价值函数：

$$\begin{aligned}&N(S_t) \leftarrow N(S_t) + 1 \\ &V(S_t) \leftarrow V(S_t) + \frac{1}{N(S_t)}(G_t - V(S_t)) \\ &w.r.t \quad G_t = R_t + \gamma R_{t+1} + \ldots + \gamma^{T-1} R_{t+T}\end{aligned}$$

这里另外说下静态问题和非静态问题的概念：

- 静态问题就是说我们的MDP是不变的，比如转移矩阵，比如奖励函数
- 非静态问题即随着时间的推移，MDP中的某些参数将会发生改变。

这里我们将MC方法变为增量式，我们可以扩展开改变$\frac{1}{N(S_t)}$，将他定义为一个类似于学习率的参数α，在处理非静态问题时，使用这个方法跟踪一个实时更新的平均值是非常有用的，可以扔掉那些已经计算过的episode信息。

引入参数α后的状态价值更新方法可以更改为：

$$V(S_t) \leftarrow V(S_t) + \alpha(G_t - V(S_t))$$

7. 蒙特卡洛预测算法伪代码

7.1 首次访问型MC预测算法，用于估计$V \approx v_\pi$

First-visit MC prediction, for estimating $V \approx v_\pi$

Input: a policy π to be evaluated
Initialize:
    $V(s) \in \mathbb{R}$, arbitrarily, for all $s \in \mathcal{S}$
    $Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$

Loop forever (for each episode):
    Generate an episode following π: $S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T$
    $G \leftarrow 0$
    Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
        $G \leftarrow \gamma G + R_{t+1}$
        Unless S_t appears in $S_0, S_1, \ldots, S_{t-1}$:
            Append G to $Returns(S_t)$
            $V(S_t) \leftarrow$ average$(Returns(S_t))$

7.2 蒙特卡洛ES（试探性出发），用于估计$\pi \approx \pi_*$

Monte Carlo ES (Exploring Starts), for estimating $\pi \approx \pi_*$

Initialize:
$\pi(s) \in \mathcal{A}(s)$ (arbitrarily), for all $s \in \mathcal{S}$
$Q(s,a) \in \mathbb{R}$ (arbitrarily), for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$
$Returns(s,a) \leftarrow$ empty list, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

Loop forever (for each episode):
Choose $S_0 \in \mathcal{S}$, $A_0 \in \mathcal{A}(S_0)$ randomly such that all pairs have probability > 0
Generate an episode from S_0, A_0, following π: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
$G \leftarrow 0$
Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
$G \leftarrow \gamma G + R_{t+1}$
Unless the pair S_t, A_t appears in $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:
Append G to $Returns(S_t, A_t)$
$Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$
$\pi(S_t) \leftarrow \text{argmax}_a\, Q(S_t, a)$

7.3 离轨策略MC预测算法（策略评估），用于估计$Q \approx q_\pi$

Off-policy MC prediction (policy evaluation) for estimating $Q \approx q_\pi$

Input: an arbitrary target policy π
Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
$Q(s,a) \in \mathbb{R}$ (arbitrarily)
$C(s,a) \leftarrow 0$

Loop forever (for each episode):
$b \leftarrow$ any policy with coverage of π
Generate an episode following b: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
$G \leftarrow 0$
$W \leftarrow 1$
Loop for each step of episode, $t = T-1, T-2, \ldots, 0$, while $W \neq 0$:
$G \leftarrow \gamma G + R_{t+1}$
$C(S_t, A_t) \leftarrow C(S_t, A_t) + W$
$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)} [G - Q(S_t, A_t)]$
$W \leftarrow W \frac{\pi(A_t|S_t)}{b(A_t|S_t)}$