强化学习基础篇（十七）时间差分预测

之前介绍的基于贝尔曼方程求解最优策略的前两种方法：动态规划法和蒙特卡洛法。动态规划法主要用于求解基于模型的强化学习任务，而蒙特卡洛法用于求解免模型的强化学习任务。虽然基于采样的蒙特卡洛法能够初步求解免模型强化学习任务，但因其自身所存在的一些不足，如数据方差大、收敛速度慢等，导致其在实际环境中的运行效果并不理想。

基于此，本文将会介绍能够更好地求解免模型强化学习任务的另一种方法—时间差分（Temporal-difference，TD）法。时间差分法利用智能体在环境中时间步之间的时序差，学习由时间间隔产生的差分数据求解强化学习任务：另外，TD结合了动态规划法和蒙特卡洛方法优点，能够更准确、高效地求解强化学习任务，是目前强化学习求解的主要方法。

1、时间差分（Temporal-Difference）概述

虽然动态规划法能够较好地求解基于模型的强化学习任务，但在现实环境中，大多数强化学习任务都属于免模型类型，即不能够提供完备的环境知识。而通过基于采样的蒙特卡洛法，能够在一定程度上解决免模型强化学习任务求解方法的问题。蒙特卡洛法的求解需要等待每次实验结束才能进行，这导致蒙特卡洛法在现实环境中的学习效率难以满足实际任务需求。

为了更高效地求解免模型的强化学习任务，我们结合基于自举（Bootstrapping）方式的动态规划法和基于采样思想的蒙将卡洛法两者的优势。提出时间差分法。TD与MC方法类似，都是基于采样数据估计当前的价值函数。与MC不同的是，TD采用DP中的Bootstrapping方式计算当前的价值函数，而MC是在每次试验结束之后才能计算响应的价值函数。

Bootstrapping（自举）概念

“Bootstrapping”这个概念表示在当前值函数的计算过程中，会利用到后续的状态值函数或动作值函数，即利用到后续的状态或<状态-动作>对，

2、时间差分（Temporal-Difference）预测原理

蒙特卡洛法对多次采样后经验轨迹的奖励进行平均，并将平均后的奖励作为累积奖励G_t的近似期望。需要特别注意的是，累积奖励的平均计算是在一个经验轨迹收集完成之后开展。其更新过程中：

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

MC利用实际的奖励G_t作为目标来更新状态值，并且状态值的更新过程能够增量式地进行。其中，α为学习率，G_t为执行了个时间步t后的实际奖励，是基于某一策略状态值的无偏估计。

在时间差分学习中，算法在估计某一状态值时，使用关于该状态的即时奖励R_{t+1}和下步的状态值V_{t+1}乘以衰减系数γ进行更新，最简单的时间差分法称为$TD(0)$，其更新过程如下：

$$V(S_t) \leftarrow V(S_t) + \alpha(R_{t+1} + \gamma V(S_{t+1}) - V(S_t))$$

其中$R_{t+1} + \gamma V(S_{t+1})$为时间差分目标（TD Target），其代替了MC中的G_t，其表示预测的实际奖励。

这里定义时间差分误差(TD Error)为$\delta_t = R_{t+1} + \gamma V(S_{t+1}) - V(S_t)$，其用于状态值函数的估计。

此外，关于时间差分目标（TD Target），主要分为两种情况：

- 普通时间差分目标：即$R_{t+1} + \gamma V(S_{t+1})$，基于下一状态的预测值计算当前奖励预测值，是当前状态实际价值的有偏估计。
- 真实时间差分目标：即$R_{t+1} + \gamma V_{\pi}(S_{t+1})$，基于下一时间步状态的实际价值计算当前奖励预测值，是当前状态实际价值的无偏估计。

时间差分法类似于蒙特卡洛法，需要模拟多次采样的经验轨迹来获得期望的状态值函数估计。当采样足够多时，状态值函数的估计便能够收敛于真实的状态值。

3. 无偏估计（Unbiased Estimate)与有偏估计（Biased Estimate）以及方差权衡

无偏估计（Unbiased Estimate）

无偏估计指在多次重复实验下，计算的平均数接近估计参数的真实值。

实际上，无偏估计是用样未统计量来估计总体参数的一种无偏推断，估计量的数学期望等于被估计参数的真实值。此估计量被称为被估计参数的无偏估计，即具有无偏性性，是一种用于评价估计量优良性的准则。

在MC方法中使用的回报$G_t = R_{t+1} + \gamma R_{t+2} + \ldots + \gamma^{T-1} R_T$就是对$v_\pi(S_t)$的无偏估计。

真实时间差分目标$R_{t+1} + \gamma V_\pi(S_{t+1})$也是对$v_\pi(S_t)$的无偏估计。

有偏估计（Biased Estimate）

有偏差计与无偏估计相反，是指由样本值求的估计值与待估计参数的真实值之间有系统误差，其期望值不是待估参教的真值。

时间差分目标$R_{t+1} + \gamma V(S_{t+1})$是对$v_\pi(S_t)$的有偏估计。

方差(Variance)

但是时间差分目标（TD target）比G_t具有更低的方差（variance），回报G_t是基于许多额随机动作，转移概率以及回报而得到的，方差较大。而时间差分目标（TD target）依赖的是单个动作，转移概率以及回报而得到的，方差较小。

所以，总结一下：

a. MC具有高方差，为无偏估计

- 具有良好的收敛性（甚至在值函数近似的场景也有良好的收敛性）
- MC对初始值得设置并不敏感
- 理解容易，易于使用

b. TD具有低方差，为有偏估计

- 其效率通常高于MC方法
- $TD(0)$可以收敛到$v_\pi(S_t)$（但是在值函数近似的场景不总是能收敛到$v_\pi(S_t)$）
- TD对初始值得设置较为敏感

4、时间差分（Temporal-Difference）在开车回家场景的示例

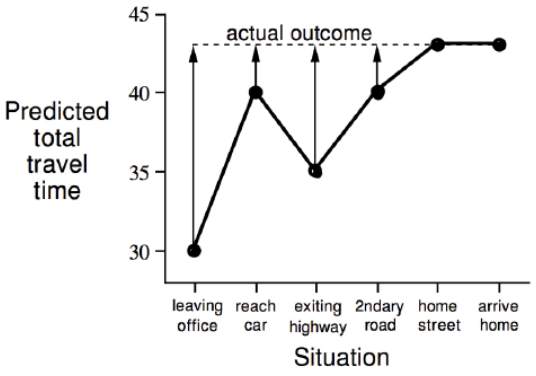
这里举一个开车回家的例子比较TD与MC方法的差异：

开车回家：当你每天从工作地点开车回家时，你会估计一下路上要花多久时间。当你离开办公室时，你会注意离开的时间、今天是星期几、当日天气，以及任何其他可能相关的因素。这个星期五，你在晚上六点整离开办公室，估计回家需要花费30分钟。当你到达你的车旁时，时间是6：05，这时却开始下雨了。在雨中开车通常比较慢，所以你重新估计，觉得到家还需要35分钟，即总共需要40分钟。15分钟后，你很快开完了高选路段，下高速后你将总时间的估计值减少到35分钟。不幸的是，这时你被堵在一辆缓慢的卡年后面，且道路太窄不能超车。最终你不得不跟着卡车，直到 6：40才开到你居住的街道，3分钟后你终于到家。在这个场景下，状态、时间和时长预测序列如下：

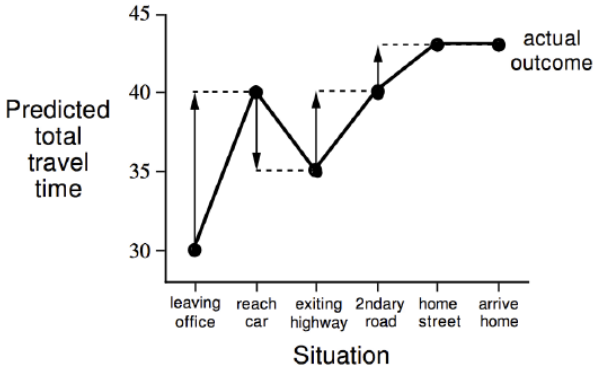
State	Elapsed Time (minutes)	Predicted Time to Go	Predicted Total Time
leaving office	0	30	30
reach car, raining	5	35	40
exit highway	20	15	35
behind truck	30	10	40
home street	40	3	43
arrive home	43	0	43

在这个例子中，收益是每一段行程消耗的时间。过程不加折扣（γ=1），因此每个状态的回报就是从这个状态开始直到回家实际经过的总时间。每个状态的价值是剩余时间的期望值。第二列数字给出了遇到的每个状态的价值的当前估计值。

Changes recommended by Monte Carlo methods (α=1)



Changes recommended by TD methods (α=1)



上图（左）所示，一种描述蒙特卡洛方法的步骤的简单办法是在时间轴上画出车总耗时的预测值（最后一列数据）。箭头表示的是常量α MC方法对预测值的改变（α=1）。这个值正是每个状态的价值的估计值（预估的剩余时间）与实际回报（真实的剩余时间）之差。

上图（右）所示，是使用TD方法对这个过程的描述，这些信息可以汇总为如下的表：

单位：分钟

状态	已经耗时	既往经验预计		MC 更新（$\alpha=1$）		TD 更新（$\alpha=1$）	
		仍需耗时	总耗时	仍需耗时	总耗时	仍需耗时	总耗时
离开办公室	0	30	30	43	43	40	40
取车时下雨	5	35	40	38	43	30	35
驱离高速	20	15	35	23	43	20	40
跟在卡车后	30	10	40	13	43	13	43
家附近街区	40	3	43	3	43	3	43
返回家中	43	0	43	0	43	0	43

想象一下作为个体的你如何预测下班后开车回家这个行程所花费的时间。在回家的路上你会依次经过一段高速公路、普通公路、和你家附近街区三段路程。由于你经常开车上下班，在下班的路上多次碰到过各种情形，比如取车的时候发现下雨，高速路况的好坏、普通公路是否堵车等等。在每一种状态下时，你对还需要多久才能到家都有一个经验性的估计。上表中的"既往经验预计（仍需耗时）"列给出了这个经验估计，这个经验估计基本反映了各个状态对应的价值，通常你对下班回家总耗时的预估是30分钟。

假设你现在又下班准备回家了，当花费了5分钟从办公室到车旁时，发现下雨了。此时根据既往经验，估计还需要35分钟才能到家，因此整个行程将耗费40分钟。随后你进入了高速公路，高速公路路况非常好，你一共仅用了20分钟就离开了高速公路，通常根据经验你只再需要 15 分钟就能到家，加上已经过去的 20 分钟，你将这次返家预计总耗时修正为35分钟，比先前的估计少了5分钟。但是当你进入普通公路时，发现交通流量较大，你不得不跟在一辆卡车后面龟速行驶，这个时候距离出发已经过去30分钟了，根据以往你路径此段的经验，你还需要10分钟才能到家，那么现在你对于回家总耗时的预估又回到了40分钟。最后你在出发 40分钟后到达了家附近的街区，根据经验，还需要3分钟就能到家，此后没有再出现新的情况，最终你在43分钟的时候到达家中。经历过这一次的下班回家，你对于处在途中各种状态下返家的还需耗时（对应于各状态的价值）有了新的估计，但分别使用MC算法和TD算法得到的对于各状态返家还需耗时的更新结果和更新时机都是不一样的。

如果使用MC算法，在整个驾车返家的过程中，你对于所处的每一个状态，例如"取车时下雨"，"离开高速公路"，"被迫跟在卡车后"、"进入街区"等时，都不会立即更新这些状态对应的返家还需耗时的估计，这些状态的返家仍需耗时仍然分别是先前的35分钟、15分钟、10分钟和3分钟。但是当你到家发现整个行程耗时43分钟后，通过用实际总耗时减去到达某状态的已耗时，你发现在本次返家过程中在实际到达上述各状态时，仍需时间则分别变成了：38分钟（43-5）、23 分钟（43-20）、13 分钟（43-30）和3分钟（43-40）。如果选择修正系数为1，那么这些新的耗时将成为今后你在各状态时的预估返家仍需耗时，相应的整个行程的预估耗时被更新为43分钟。

单位：分钟

状态	已经耗时	既往经验预计		MC 更新（$\alpha=1$）		TD 更新（$\alpha=1$）	
		仍需耗时	总耗时	仍需耗时	总耗时	仍需耗时	总耗时
离开办公室	0	30	30	43	43	40	40
取车时下雨	5	35	40	38	43	30	35
驱离高速	20	15	35	23	43	20	40
跟在卡车后	30	10	40	13	43	13	43
家附近街区	40	3	43	3	43	3	43
返回家中	43	0	43	0	43	0	43

如果使用TD算法，则又是另外一回事，当取车发现下雨时，同样根据既往经验你会认为还需要35分钟才能返家，此时，你将立刻更新对于返家总耗时的估计，为仍需的35分钟加上你离开办公室到取车现场花费的5分钟，即40分钟。同样道理，当驶离高速公路，根据经验，你对到家还需时间的预计为15分钟，但由于之前你在高速上较为顺利，节省了不少时间，在第20分钟时已经驶离高速，实际从取车到驶离高速只花费了15分钟，则此时你又立刻更新了从取车时下雨到到家所需的时间为30分钟，而整个回家所需时间更新为35分钟。当你在驶离高速在普通公路上又行驶了10分钟被堵，你预计还需10分钟才能返家时，你对于刚才驶离高速公路返家还需耗时又做了更新，将不再是根据既往经验预估的15分钟，而是现在的20分钟，加上从出发到驶离高速已花费的20分钟，整个行程耗时预估因此被更新为40分钟。直到你花费了40分钟只到达家附近的街区还预计有3分钟才能到家时，你更新了在普通公路上对于返家还需耗时的预计为13分钟。最终你按预计3分钟后进入家门，不再更新剩下的仍需耗时。

通过这个例子我们可以知道：

a. TD可以在知道最终结果之前就进行学习。

- 我们可以使用TD进行在线学习，每一个时间步之后就可以进行学习。
- MC确是必须等到当前幕介绍，例如必须开车到家后知道最终的时间才能进行学习。

b. TD也可以在没有最终输出的场景下进行学习。

- 就算是序列不完整，TD也是可以学习的，而MC方法必须依赖于完整的序列。
- TD方法可以应用于连续的环境任务（没有结束点），而MC方法比必须应用于具有结束点的环境。

5、表格型$TD(0)$算法伪代码

Tabular TD(0) for estimating v_π

Input: the policy π to be evaluated
Algorithm parameter: step size $\alpha \in (0, 1]$
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop for each episode:
  Initialize S
  Loop for each step of episode:
    $A \leftarrow$ action given by π for S
    Take action A, observe R, S'
    $V(S) \leftarrow V(S) + \alpha\big[R + \gamma V(S') - V(S)\big]$
    $S \leftarrow S'$
  until S is terminal