基础时序差分方法

除了蒙特卡洛方法之外，时序差分(Timeporal-Difference Learning) 方法也是一种model-free的强化学习方法，TD-Learning的思想也贯穿了Sarsa、Q-Learning等经典算法。因此在本文中我们将对TD-Learning做详细的推导与分析，希望在文章结束时我们能够回答下面的几个疑问，也是本人在初学TD-Learning时的困惑：

1. TD-Learning的基本原理是什么?
2. TD-Learning的本质是什么 ?
3. TD-Learning相较于蒙特卡洛方法有什么区别和优势?

在正式介绍TD-Learning之前，我们先引入Robbins-Monro算法 (简称RM算法)，这个算法就是要找到方程的根，即我们要找到一个 ，使得：

当然，假如的函数表达式已知，那么我们还能通过许多其他的方法找到方程的根。然而，引入RM算法的重要原因就在于RM算法能够应对函数表达式未知、或者无法获取的情况下方程的求根问题。我们直接给出RM算法找方程根的迭代过程：

其中， 是更新步长，类似于深度学习中的学习率。 表示对函数 的带噪观测。这一点应该如何理解呢：试想我们现在有一个黑盒子 (神经网络)，这个黑盒子实际上实现了某种函数 ，但是我们并不能获取到的表达式，我们只能让输入到这个黑盒子里面，来观测经过这个黑盒子之后得到的输出 ，由于噪声等因素的存在，我们也无法完全准确地观测到 ，实际上观测得到的只是 的一个近似值 。

有了RM算法的背景，我们就可以正式进入TD-Learning的学习了。

我们先从0开始推导最基本的TD算法：回顾一下强化学习的基本步骤，我们有一个策略 ，首先需要做策略评估，来评价当前的这个策略是好是坏。**策略评估我们就通过求解贝尔曼公式来实现**。在做完策略评估之后，我们就需要进行策略改进。我们首先聚焦于策略评估，先给出贝尔曼公式：

其中，第一项可以展开为：

其中，第二项 可以展开为：

我们发现，第二项正好是下一状态 的state value的期望! 即：

因此，我们可以得到贝尔曼方程的另一种写法：

我们别忘了一开始的目标：策略评估。即求解上式的贝尔曼方程。因此我们可以定义一个函数：

所以问题转化为找到一个 ，使得 。然而，这个方程中有期望的存在，期望实际上是比较难准确观测到的，我们现在手头上只有一些经验样本 和 ，因此我们就用这些样本去近似期望值。所以我们便可以得到RM算法中所说的“带噪观测”，如下：

因此，利用RM算法，我们就可以得到 的迭代更新操作 (Update rule)：

其中， 表示算法在第 步对 的估计。 是在第 步对 的采样 (**注：因为在RM算法中，我们每更新一次 ，都会重新将 输入进函数 ，得到新的带噪观测  。所以在TD-Learning中我们每更新一次 ，都要重新对奖励、下一个状态进行采样**)

在上面的“注”中，蕴藏着TD-Learning的实施步骤：即为了能够顺利收敛到 ，我们必须一直反复从 出发，得到一系列经验 。然而，这在实际训练中是比较低效的，我们在实际训练的时候，依然是先直接生成一长串的轨迹：

然后依次访问轨迹中的每一个 ，利用TD-Learning的Update rule对当前状态 的state value更新一个step，然后再访问轨迹中的下一条经验。只要轨迹足够长，对各个状态的访问次数足够多，算法依然可以让各个状态的state value达到收敛。

其次，我们观察TD-Learning的Update rule，可以发现，为了计算 的步的更新时，需要依赖下一状态准确的state value 。然而在实际实现时**我们并不能保证在计算 状态的state value时，其下一个状态的state value已经收敛**。为了解决这一问题，我们还是利用刚刚的思想，用 去替换 。能够这样替换的直观理解是：尽管我使用的 和真实值 还是有差距，但是随着的增大， 是越来越接近 的。因此只要迭代的次数足够多， 依然可以收敛到 。

我们现在搞清楚了TD-Learning的基本原理，即文章开篇所提出的第一个问题，那么第二个问题也就迎刃而解了：

**TD算法的本质就是利用RM算法去求解贝尔曼方程!**

这里插一句题外话：目前版本的TD-Learning只能帮助我们求出在某一策略 下，各个状态所对应的state value。即在做Policy evaluation。还需要结合其他的手段求出action value之后才能够进行Policy improvement。

最后，回答一下本文开篇所提到的最后一个问题：TD-Learning相比于蒙特卡洛方法有什么特点?

1. **Online：**从上面的实现分析中我们可以知道，TD-Learning是跟环境交互一次，就可以那这次交互的经验更新一次state value。即它是实时学习的，而蒙特卡洛方法则需要完整的episode完了之后才可以进行学习，所以是offline的。
2. **TD-Learning适用于连续任务：**即如果某个任务从开始到结束的episode特别特别长，这样蒙特卡洛方法为了完成一次学习就需要等待大量的时间，而TD-Learning则没有这样的顾虑。
3. **Boostrapping：**TD-Learning对某一状态的state value的估计要依赖于其他状态 的state value的估计，因此是boostrapping的。蒙特卡洛方法对state value的估计不依赖于其他任何状态，因此是non-boostrapping的。

回顾一下，本文推导了最简单的TD-Learning形式，它实际上是在利用RM算法求解贝尔曼方程去估计state value。后面我们再看看如何利用TD-Learning去估计action value以及optimal action value。将会涉及到著名的Sarsa算法和Q-Learning。