文章标题

本章在上一章的基础上，进一步介绍了新的方法：**时序差分学习**（Temporal-Difference learning），简称 TD Learning。

时序差分学习可以看作蒙特卡罗（MC）和动态规划（DP）的一种结合：

- 和 MC 的相似之处在于，TD 方法从实际的经验来获取信息，无需获知环境的全部信息。
- 和 DP 的相似之处在于，TD 方法能够利用上之前已知的信息来做实时学习，无需等得到完整的收益反馈再进行估值更新。

本章同样基于 **GPI 模型**，来介绍基于时序差分学习（TD learning）的算法。

一、测试段落 1

我们先对比看看上一章的 MC 算法（固定 α）和本章的 TD 算法的核心公式：

constant-α MC:

$$V(S_t) \leftarrow V(S_t) + \alpha \left[G_t - V(S_t) \right]$$

TD(0) (one-step TD):

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

- MC 方法必须等待整个 episode 结束后得到 G_t 才能做一次更新。
- TD 方法则只需等到这一步结束，利用实时观测到的奖励值 R_{t+1} 和现有估计值 $V(S_{t+1})$ 来进行更新。

*这里的 TD(0) 方法指**单步 TD 方法**，括号里的 0 改为其他数字后又指其他算法，将会在后面章节介绍。*

我们在第二章讲过，这样的更新式可以更广义地写作

$$NewEstimate \leftarrow OldEstimate + StepSize \left[Target - OldEstimate \right]$$

将括号中的式子看作是一种误差，便可将这样的更新式看作是不断地在消除误差。我们将 TD 方法中的这个误差定义为 **TD error:**

$$\delta_t \doteq R_{t+1} + \gamma V(S_{t+1}) - V(S_t)$$

二、测试段落 2

2.1 优点

- 无须获知环境的具体模型。
- 通过一种在线的、完全实时的方式来进行增量更新。
- 如果 episodes 太长，或者是连续型任务，MC 方法将会有很严重的延迟问题，TD 方法能够解决这种问题。

2.2 收敛性

给定策略 π，满足一定条件的情况下，能够证明 TD(0) 方法能确保 v 收敛到 v_π：

$$\sum_{k=1}^{\infty} \alpha = \infty \qquad \text{and} \qquad \sum_{k=1}^{\infty} \alpha^2 < \infty$$

这个结果是来自随机逼近方面的理论，我们在第二章也提到过。需要注意的是，这只是其收敛的一个必要条件，一些情况下即使不满足，也一样能收敛。

2.3 效率对比

- 目前还没能从数学上证明哪个方法（TD & MC）收敛得更快。
- 实际情况下，对于随机性的任务，TD 方法通常收敛得比 constant-α MC 方法要快一些。