第六讲：无模型方法三——多步自举

主讲人 陈达贵

清华大学自动化系

在读硕士

深蓝学院

shenlanxueyuan.com

强化学习理论与实践

2019-1-11

目录

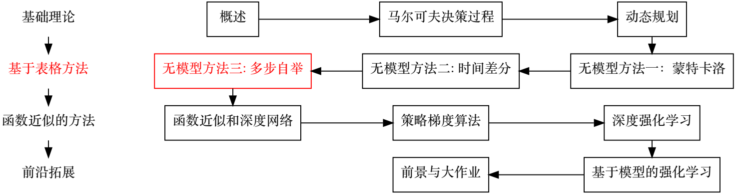
1 本章简介

2 多步自举

3 TD(λ)

4 TD(λ) 优化算法

章节目录


基础理论
基于表格方法
函数近似的方法
前沿拓展
概述
马尔可夫决策过程
动态规划
无模型方法一：蒙特卡洛
无模型方法二: 时间差分
无模型方法三: 多步自举
函数近似和深度网络
策略梯度算法
深度强化学习
基于模型的强化学习
前景与大作业

本章目录

1 本章简介

2 多步自举

3 TD(λ)
- TD(λ) 简介
- 资格迹
- TD(λ) 的两种视角的关系

4 TD(λ) 优化算法

无模型方法对比

MC 和 TD(0) 算法存在不同的优劣

- 是否从完整片段学习?
- 偏差和方差的权衡?
- 是否利用马尔可夫性？
- 浅备份? 深备份？
- 是否使用了自举?
- ...

更通用的方法

MC 和 TD(0) 都是无模型的方法，能不能有统一的方法呢？

- 多步自举
- TD(λ)，$\lambda \in [0, 1]$
- 资格迹

目录

1 本章简介

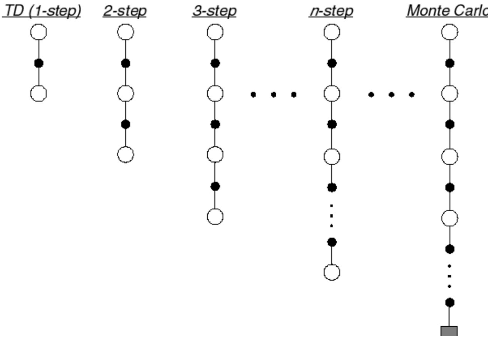
2 多步自举

3 TD(λ)

4 TD(λ) 优化算法

多步 TD

- 之前的 TD(0) 算法其实是一步 TD，即采样 1 步，然后利用对应的后继状态的值函数进行备份
- 多步的 TD 备份图如下:



n 步回报值

- 考虑下面的 n 步回报值对于 $n = 1, 2, \cdots, \infty$

$$n = 1 \quad \text{(TD)} \qquad G_t^{(1)} = R_{t+1} + \gamma V(S_{t+1})$$

$$n = 2 \qquad G_t^{(2)} = R_{t+1} + \gamma R_{t+2} + \gamma^2 V(S_{t+2})$$

$$n = \infty \quad \text{(MC)} \qquad G_t^{(\infty)} = R_{t+1} + \gamma R_{t+2} + \ldots + \gamma^{T-t-1} R_T$$

- 我们可以定义 n 步回报值

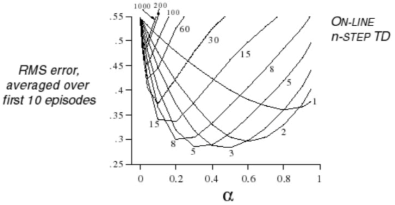
$$G_t^{(n)} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n V(S_{t+n})$$

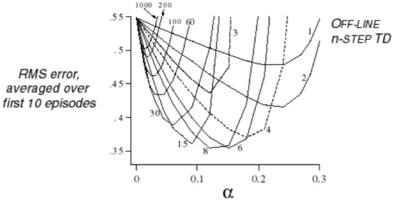
- n 步 TD 学习

$$V(S_t) \leftarrow V(S_t) + \alpha \left(G_t^{(n)} - V(S_t) \right)$$

具体几步更好?

需要自己实验，经验认为 3-10 步左右，要好于 $TD(0)$ 和 MC





n 步 TD 策略评价

算法 1 n 步 TD 策略评价

repeat（对于每一个片段）
    repeat 对于片段中的每一步
        根据 $\pi(\cdot, S_t)$ 选择动作 A_t
        执行动作 A_t，观察到 R_{t+1}, S_{t+1}，并将其存储起来
        if $\tau = t - n + 1 \geq 0$ **then**
            $G \leftarrow \sum_{i=\tau+1}^{\min(\tau+n,T)} \gamma^{i-\tau-1} R_i$
            If $\tau + n < T$, then $G \leftarrow G + \gamma^n V(S_{\tau+n})$
            $V(S_\tau) \leftarrow V(S_\tau) + \alpha \left[G - V(S_\tau)\right]$
        end if
    until 直到终止状态
until 收敛

两个注意的点：

- 为了计算 n 步回报值，需要维护 R, S 的存储空间
- 对于后继状态不足 n 个的，使用 MC 目标值

目录

1 本章简介

2 多步自举

3 TD(λ)
- TD(λ) 简介
- 资格迹
- TD(λ) 的两种视角的关系

4 TD(λ) 优化算法

目录

1 本章简介

2 多步自举

3 TD(λ)
- TD(λ) 简介
- 资格迹
- TD(λ) 的两种视角的关系

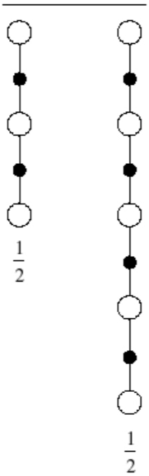
4 TD(λ) 优化算法

将 n 步回报值平均

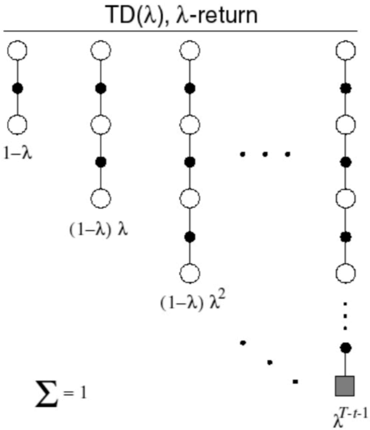
- 不同的 n 下的 n 步回报值效果不同
- 将不同 n 下的 n 步回报值做加权平均，也能构成一个有效的回报值
- 比如：平均 2 步回报值和 4 步回报值

$$\frac{1}{2}G^{(2)}+\frac{1}{2}G^{(4)}$$

- 混合了两种信息
- 能不能有效地混合所有的 n 步回报值？



λ 回报值



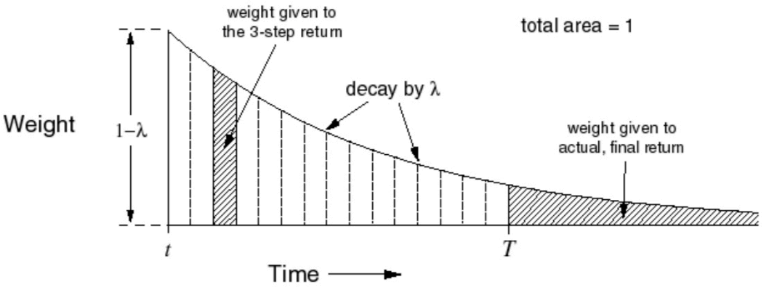
- λ 回报值混合了所有的 n 步回报值 $G_t^{(n)}$
- 使用了权重 $(1-\lambda)\lambda^{n-1}$

$$G_t^{\lambda} = (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}G_t^{(n)}$$

- $\lambda=0$，退化成 TD(0)；$\lambda=1$，退化成 MC
- TD(λ) 更新公式

$$V(S_t) \leftarrow V(S_t)+\alpha\left(G_t^{\lambda}-V(S_t)\right)$$

TD(λ) 加权函数

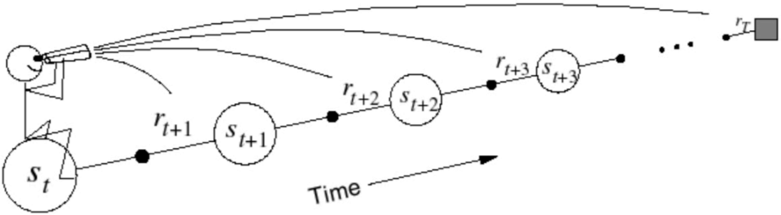


$$G_t^{\lambda} = (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}G_t^{(n)}$$

TD(λ) 的两种视角

- 前向视角 (forward-view): 主要是为了理解 TD(λ)
- 后向视角 (backward-view): 真正的实用算法

TD(λ) 的前向视角



- 通过使用 λ 回报值来更新值函数
- 前向视角使用将来的数据 $R_{t+1}, S_{t+1}, \cdots$ 来计算 G_t^λ
- 类似于 MC，只能从完整的片段学习

目录

1 本章简介

2 多步自举

3 TD(λ)
- TD(λ) 简介
- 资格迹
- TD(λ) 的两种视角的关系

4 TD(λ) 优化算法

TD(λ) 的后向视角

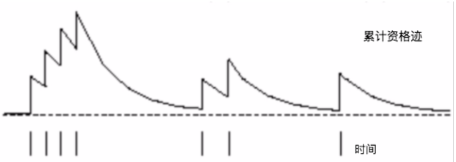
- 前向视角提供理论
- 后向视角提供实用算法
- 通过后向视角，可以实现
  - 在线更新
  - 每步更新
  - 从不完整状态更新

注: <u>通过后向视角验证了：同一个算法使用不同的执行方法可以获得不同的计算收益</u>

资格迹 (Eligibility Traces)

- 信度分配 (Credit assignment) 问题：到底是钟声还是灯光造成了最后的震动
- 频率启发式：归因到频数最高的状态
- 近因启发式：归因到最近的状态
- 资格迹是两者的结合

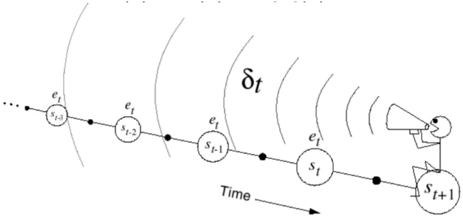
$$E_0(s) = 0$$
$$E_t(s) = \gamma\lambda E_{t-1}(s) + \mathbf{1}(S_t = s)$$



后向视角的 TD(λ)

- 对于每一个状态 s，维护一个资格迹 $E(s)$
- 更新值函数 $V(s)$ 时，会更新每一个状态 s
- 使用 TD 误差 δ_t 和资格迹 $E_t(s)$

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



- 资格迹本质上是记录了所有状态 s 对后继状态 S_{t+1} 的贡献度，被用来对 TD 误差进行加权

目录

1 本章简介

2 多步自举

3 TD(λ)
- TD(λ) 简介
- 资格迹
- TD(λ) 的两种视角的关系

4 TD(λ) 优化算法

TD(λ) 与 TD(0)

- 当 $\lambda = 0$ 时，只有当前状态会被更新

$$E_t(s) = \mathbf{1}(S_t = s)$$
$$V(s) \leftarrow V(s) + \alpha\delta_t E_t(s)$$

- 等价于 TD(0) 的更新公式

$$V(S_t) \leftarrow V(S_t) + \alpha\delta_t‘$$

TD(λ) 和 MC

- 当 $\lambda = 1$ 时，信度分配会被延迟到终止状态
- 这里考虑片段性任务，而且考虑离线更新
- 考虑一个片段整体的情况下，TD(1) 总更新量等价于 MC
  - 在每一步更新上可能有差距
- 对 s 的总更新量
  - TD(λ) 前向视角：$\sum_{t=1}^{T} \alpha(G_t^\lambda - V(S_t))\mathbf{1}(S_t = s)$
  - TD(λ) 后向视角：$\sum_{t=1}^{T} \alpha\delta_t E_t(s)$

TD(1) 和 MC

- 考虑一个片段，其中在时间 k 处，状态 s 会被访问
- TD(1) 算法中，对 s 的资格迹如下

$$
\begin{aligned}
E_t(s) &= \gamma E_{t-1}(s) + \mathbf{1}(S_t = s) \\
&= \begin{cases} 0 & \text{if } t < k \\ \gamma^{t-k} & \text{if } t \geq k \end{cases}
\end{aligned}
$$

- TD(1) 在线更新累计误差

$$
\sum_{t=1}^{T-1} \alpha \delta_t E_t(s) = \alpha \sum_{t=k}^{T-1} \gamma^{t-k} \delta_t = \alpha (G_k - V(S_k))
$$

- 一直到片段结束

$$
\delta_k + \gamma \delta_{k+1} + \gamma^2 \delta_{k+2} + \cdots + \gamma^{T-1-k} \delta_{T-1}
$$

TD(1) 和 MC

- 当 $\lambda = 1$ 时，TD 误差的和能够简化为 MC 误差

$$
\begin{aligned}
&\delta_t + \gamma\delta_{t+1} + \gamma^2\delta_{t+2} + \ldots + \gamma^{T-1-t}\delta_{T-1} \\
= &R_{t+1} + \gamma V(S_{t+1}) - V(S_t) \\
&+ \gamma R_{t+2} + \gamma^2 V(S_{t+2}) - \gamma V(S_{t+1}) \\
&+ \gamma^2 R_{t+3} + \gamma^3 V(S_{t+3}) - \gamma^2 V(S_{t+2}) \\
&\quad \vdots \\
&+ \gamma^{T-1-t} R_T + \gamma^{T-t} V(S_T) - \gamma^{T-1-t} V(S_{T-1}) \\
= &R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots + \gamma^{T-1-t} R_T - V(S_t) \\
= &G_t - V(S_t)
\end{aligned}
$$

TD(1) 和 MC

- TD(1) 等价于每次拜访的 MC 算法
- 区别是：在线误差累计，每步更新
- 如果 TD(1) 也等到片段结束后离线更新
- 那么 TD(1) 就是 MC

对 TD(λ) 化简

前向视角和后向视角下的误差等价

$$
\begin{aligned}
G_t^\lambda - V(S_t) = & -V(S_t) + (1-\lambda)\lambda^0 (R_{t+1} + \gamma V(S_{t+1})) \\
& + (1-\lambda)\lambda^1 (R_{t+1} + \gamma R_{t+2} + \gamma^2 V(S_{t+2})) \\
& + (1-\lambda)\lambda^2 (R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 V(S_{t+3})) \\
& + \ldots \\
= & -V(S_t) + (\gamma\lambda)^0 (R_{t+1} + \gamma V(S_{t+1}) - \gamma\lambda V(S_{t+1})) \\
& + (\gamma\lambda)^1 (R_{t+2} + \gamma V(S_{t+2}) - \gamma\lambda V(S_{t+2})) \\
& + (\gamma\lambda)^2 (R_{t+3} + \gamma V(S_{t+3}) - \gamma\lambda V(S_{t+3})) \\
& + \ldots \\
= & \quad (\gamma\lambda)^0 (R_{t+1} + \gamma V(S_{t+1}) - V(S_t)) \\
& + (\gamma\lambda)^1 (R_{t+2} + \gamma V(S_{t+2}) - V(S_{t+1})) \\
& + (\gamma\lambda)^2 (R_{t+3} + \gamma V(S_{t+3}) - V(S_{t+2})) \\
& + \ldots \\
= & \ \delta_t + \gamma\lambda\delta_{t+1} + (\gamma\lambda)^2 \delta_{t+2} + \ldots
\end{aligned}
$$

前向视角和后向视角的 TD(λ)

- 考虑一个片段，其中在时间 k 处，状态 s 会被访问
- TD(λ) 的资格迹

$$
\begin{aligned}
E_t(s) &= \gamma\lambda E_{t-1}(s) + \mathbf{1}(S_t = s) \\
&= \begin{cases} 0 & \text{if } t < k \\ (\gamma\lambda)^{t-k} & \text{if } t \geq k \end{cases}
\end{aligned}
$$

- 后向视角 TD(λ) 在线更新累计误差

$$
\sum_{t=1}^{T} \alpha\delta_t E_t(s) = \alpha \sum_{t=k}^{T} (\gamma\lambda)^{t-k}\delta_t = \alpha\left(G_k^\lambda - V(S_k)\right)
$$

- 截止到片段结束时，能收集到 λ 回报值误差
- 对 s 的多次拜访，$E_t(s)$ 会收集多次误差

两种视角下的等价性

离线更新

- 在整个片段里累计更新误差
- 在片段结束后统一更新
- 离线更新下，前向视角和后向视角等价

在线更新

- TD(λ) 在片段的每一步更新
- 此时前向视角和后向视角有一点不同
- 可以通过对资格迹进行一些修正，使两者完全等价
- True Online TD(lambda) - Harm van Seijen, Richard S. Sutton

小结

离线更新	$\lambda = 0$	$\lambda \in (0,1)$	$\lambda = 1$
后向视角 前向视角	TD(0) $\Updownarrow$ TD(0)	TD(λ) $\Updownarrow$ 前向 TD(λ)	TD(1) $\Updownarrow$ MC
在线更新	$\lambda = 0$	$\lambda \in (0,1)$	$\lambda = 1$
后向视角 前向视角 真实在线更新	TD(0) $\Updownarrow$ TD(0) $\Updownarrow$ TD(0)	TD(λ) $\not\Updownarrow$ 前向 TD(λ) $\Updownarrow$ 真实在线 TD(λ)	TD(1) $\not\Updownarrow$ MC $\Updownarrow$ 真实在线 TD(1)

目录

1 本章简介

2 多步自举

3 TD(λ)

4 TD(λ) 优化算法

n 步 Sarsa

- 考虑下面的 n 步回报值对于 $n = 1, 2, \cdots, \infty$

$$
\begin{aligned}
&n = 1 \quad \text{(Sarsa)} && q_t^{(1)} = R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) \\
&n = 2 && q_t^{(2)} = R_{t+1} + \gamma R_{t+2} + \gamma^2 Q(S_{t+2}, A_{t+2}) \\
&n = \infty \quad \text{(MC)} && q_t^{(\infty)} = R_{t+1} + \gamma R_{t+2} + \ldots + \gamma^{T-t-1} R_T
\end{aligned}
$$

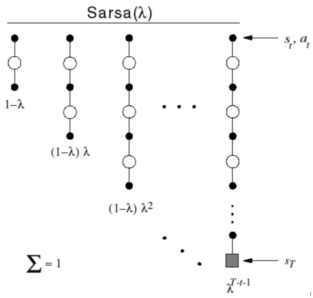
- 定义 n 步 Q 回报值

$$q_t^{(n)} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n Q(S_{t+n}, A_{t+n})$$

- n 步 Sarsa 使用 n 步回报值更新 $Q(s, a)$

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left(q_t^{(n)} - Q(S_t, A_t) \right)$$

前向视角的 Sarsa(λ)



- q^λ 回报值混合了所有的 n 步 Q 回报值 $q_t^{(n)}$
- 使用加权权重 $(1-\lambda)\lambda^{n-1}$

$$q_t^\lambda = (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}q_t^{(n)}$$

- 前向视角 Sarsa(λ)

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\left(q_t^\lambda - Q(S_t, A_t)\right)$$

后向视角的 Sarsa(λ)

- 使用资格迹在线更新
- 但是这里的资格迹是针对 $\langle s, a\rangle$

$$
\begin{aligned}
E_0(s,a) &= 0 \\
E_t(s,a) &= \gamma\lambda E_{t-1}(s,a) + \mathbf{1}(S_t = s, A_t = a)
\end{aligned}
$$

- 对所有的 s, a 更新 $Q(s,a)$
- 利用资格迹和 TD 误差更新

$$
\begin{aligned}
\delta_t &= R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t) \\
Q(s,a) &\leftarrow Q(s,a) + \alpha\delta_t E_t(s,a)
\end{aligned}
$$

Sarsa(λ) 算法

算法 2 Sarsa(λ) 算法

对于所有的 $s \in \mathcal{S}, a \in \mathcal{A}(s)$ 初始化 $Q(s, a)$
repeat（对于每一个片段）
    初始化 $E(s, a) = 0, \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$
    选择初始状态和初始动作 S, A
    repeat 对于片段中的每一步
        执行动作 A，观察到 R, S'
        根据 Q 选择一个在 S' 处的动作 A'（e.g. 使用 ε-贪婪策略）
        $\delta \leftarrow R + \gamma Q(S', A') - Q(S, A)$
        $E(S, A) \leftarrow E(S, A) + 1$
        for 对于所有的 $s \in \mathcal{S}, a \in \mathcal{A}(s)$ **do**
            $Q(s, a) \leftarrow Q(s, a) + \alpha \delta E(s, a)$
            $E(s, a) \leftarrow \gamma \lambda E(s, a)$
        end for
        $S \leftarrow S', A \leftarrow A'$
    until 直到终止状态
until 收敛