强化学习

第四讲：无模型预测学习

教师：赵冬斌　朱圆恒　张启超

中国科学院大学
中国科学院自动化研究所

April 8, 2021

上一节课回顾：动态规划

- 价值迭代
- 策略迭代
- 广义策略迭代

上一节课回顾：动态规划

- 价值迭代
- 策略迭代
- 广义策略迭代
  - 贝尔曼最优方程

$$V_*(s) = \max_a \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a V_*(s') \right)$$

  - 最优策略

$$\pi_*(s) = \arg\max_{a \in \mathcal{A}} \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_*(s') \right)$$

上一节课回顾：动态规划

- 价值迭代
- 策略迭代
- 广义策略迭代
  - 贝尔曼最优方程

$$V_*(s) = \max_a \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a V_*(s') \right)$$

  - 最优策略

$$\pi_*(s) = \arg\max_{a \in \mathcal{A}} \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_*(s') \right)$$

1. 求解非线性算子 max
2. 模型已知
3. 足够的计算空间

策略评估

- 许多场景需要对智能体的某一策略进行评估，确定该策略的好坏
  - e.g. 某一围棋程序的水平，智能汽车的驾驶水平
- 动态规划方法：求解贝尔曼期望方程

$$V_\pi(s) = \sum_{a\in\mathcal{A}} \pi(a|s)\left(\mathcal{R}_s^a + \gamma \sum_{s'\in\mathcal{S}} \mathcal{P}_{ss'}^a V_\pi(s')\right)$$

  - 矩阵形式求解

$$\mathcal{V}_\pi = (I - \gamma\mathcal{P}^\pi)^{-1}\mathcal{R}^\pi$$

  - 迭代计算

$$V_{l+1}(s) = \sum_{a\in\mathcal{A}} \pi(a|s)\left(\mathcal{R}_s^a + \gamma \sum_{s'\in\mathcal{S}} \mathcal{P}_{ss'}^a V_l(s')\right)$$

策略评估

- 许多场景需要对智能体的某一策略进行评估，确定该策略的好坏
  - e.g. 某一围棋程序的水平，智能汽车的驾驶水平
- 动态规划方法：求解贝尔曼期望方程

$$V_\pi(s) = \sum_{a\in\mathcal{A}} \pi(a|s)\left(\mathcal{R}_s^a + \gamma \sum_{s'\in\mathcal{S}} \mathcal{P}_{ss'}^a V_\pi(s')\right)$$

  - 矩阵形式求解

$$\mathcal{V}_\pi = (I - \gamma\mathcal{P}^\pi)^{-1}\mathcal{R}^\pi$$

  - 迭代计算

$$V_{l+1}(s) = \sum_{a\in\mathcal{A}} \pi(a|s)\left(\mathcal{R}_s^a + \gamma \sum_{s'\in\mathcal{S}} \mathcal{P}_{ss'}^a V_l(s')\right)$$

- 共同的不足：对模型依赖
  - 要么是 MDP 问题的模型已知 $\mathcal{R}, \mathcal{P}$
  - 要么智能体对环境建模 $\hat{\mathcal{R}}, \hat{\mathcal{P}}$

蒙特卡洛方法

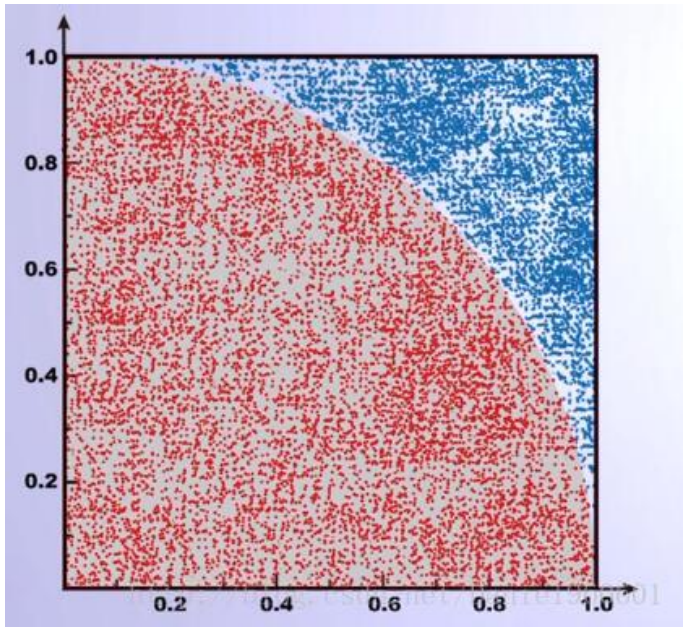
举例: 计算圆周率 π =?

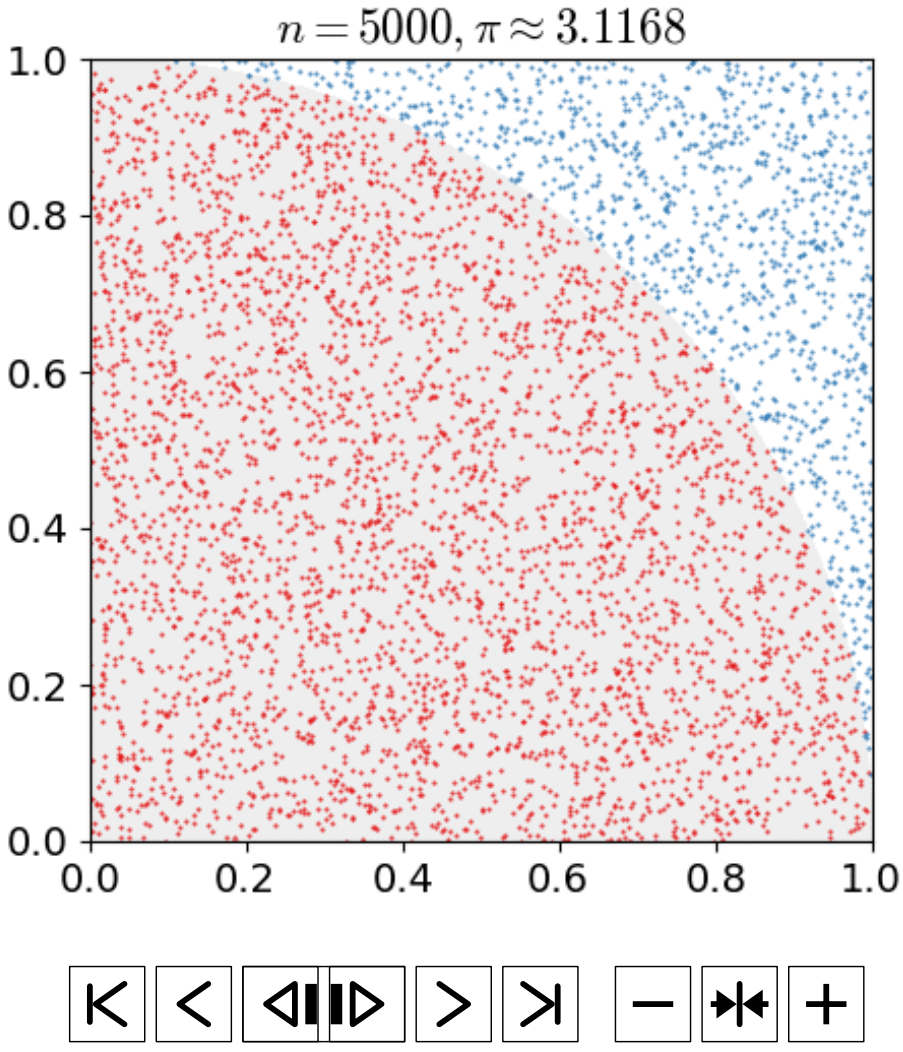
圆周率是圆的周长与直径的比值 $(l = \pi d)$, 也等于圆形面积与半径平方之比 $(A = \pi r^2)$

举例: 计算圆周率 $\pi = ?$

圆周率是圆的周长与直径的比值 $(l = \pi d)$, 也等于圆形面积与半径平方之比 $(A = \pi r^2)$

- **蒙特卡洛方法** 计算圆周率
  1. 构造一个单位正方形和一个单位圆的 1/4
  2. 向整个区域内随机投掷点，根据点到原点的距离判断点是落在 1/4 的圆内还是在圆外
  3. 根据落在两个不同区域的点数，求两个区域的比值
  4. 求出 1/4 单位圆的面积，从而求出圆周率





- 使用蒙特卡罗方法估算圆周率. 放置 30000 个随机点后, 的估算值与真实值相差 0.07%

蒙特卡洛方法 Monte-Carlo, MC

- MC 方法直接根据观测的轨迹学习
- MC 学习特点
  - 无模型 (model-free) 方法: 不需要 MDP 的转移/奖励函数
  - 根据完整的轨迹学习: 没有自举
- 基于最简单的思想: 价值 = 平均 (期望) 回报
- 运行 MC 方法通常要求:
  - 所有轨迹都到达终止状态 $\{s_0, a_0, \ldots, s_{terminal}\}$
  - 或者轨迹足够长 $\{s_0, a_0, \ldots, s_T\}, T \gg 1$

蒙特卡洛策略评估

- 目标: 从策略 π 产生的轨迹中学习 V_π

$$s_0, a_0, r_1, \ldots, s_k \sim \pi$$

- 回顾：回报的定义是所有折扣奖励和

$$G_t = r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{T-1} r_t$$

- 回顾：价值的定义是回报的期望

$$V_\pi(S) = \mathbb{E}_\pi[G_t | s_t = S]$$

- 蒙特卡洛策略评估方法使用回报的**经验均值**作为回报的**期望**

首次经过的 MC 策略评估

- 为了评估状态 s
- 在一次轨迹中 首次 遇到状态 s 时 (假定时刻为 t)
  - 计数加一：$N(s) \leftarrow N(s) + 1$
  - 累加 t 时刻之后的轨迹回报：$S(s) \leftarrow S(s) + G_t$
- 估计的价值等于回报均值 $V(s) = S(s)/N(s)$

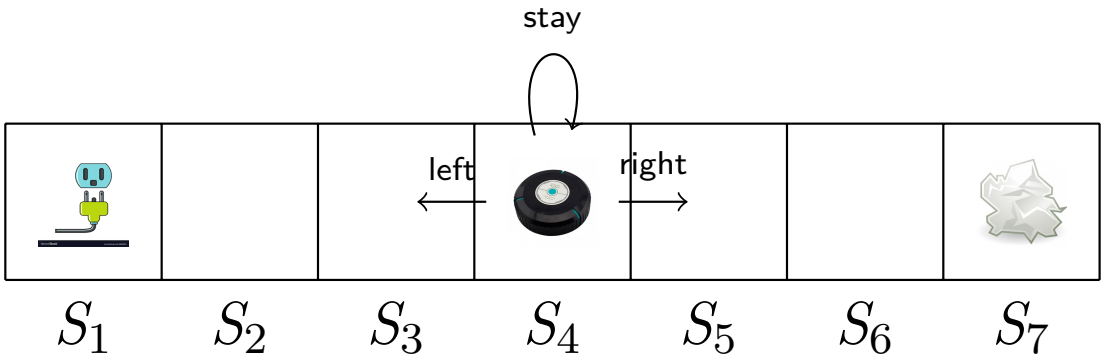
根据**大数定理**，当 $N(s) \to \infty$ 时，$V(s) \to V_\pi(s)$

每次经过的 MC 策略评估

- 为了评估状态 s
- 在一次轨迹中 每次 遇到状态 s 时 (假定时刻为 t)
  - 计数加一：$N(s) \leftarrow N(s) + 1$
  - 累加 t 时刻之后的轨迹回报：$S(s) \leftarrow S(s) + G_t$
- 估计的价值等于回报均值 $V(s) = S(s)/N(s)$

同样当 $N(s) \to \infty$ 时，$V(s) \to v_\pi(s)$

举例: 扫地机器人



- 策略: 机器人在所有位置以相同的概率随机向左或向右

$$\pi(left|s) = 0.5, \quad \pi(right|s) = 0.5$$

- 矩阵求解价值函数

$$V_\pi = [2.1322, 0.9883, 0.7856, 1.3311, 3.1443, 7.9520, 20.3332]^T$$

- 迭代计算结果

$$V_{22} = [2.1305, 0.9865, 0.7837, 1.3290, 3.1421, 7.9497, 20.3308]^T$$

蒙特卡洛方法

- 首次经过 MC 方法: 从某一状态 S_i 出发, 在策略 π 下运行 M 步, 重复 N 次实验
- 对 0 时刻的回报 G_0 来说，$t = M$ 时刻之后的累加奖励在折扣因子作用下最多等于

$$\Big| \sum_{t=M}^{\infty} \gamma^M r_{t+1} \Big| \leq \frac{\gamma^M}{1-\gamma} |\mathcal{R}|_{\max}$$

  - e.g. M $= 100$, $|\mathcal{R}|_{\max} = 10$, $\gamma = 0.7$, $\frac{\gamma^{100}}{1-\gamma} R_{\max} = 10^{-14}$
  - 也就是说计算 G_0 时只用 99 步的轨迹 $r_1 + \gamma r_2 + \cdots + \gamma^{99} r_{100}$ 误差也不超过 10^{-14}

不同重复次数下的实验结果

v_π	S_1	S_2	S_3	S_4	S_5	S_6	S_7
$N=1$	1.8763	0.9538	0.4634	2.7009	0.0114	0.1611	30.5428
$N=10$	2.1897	1.0364	0.5345	0.7806	4.7444	5.1964	20.9025
$N=100$	2.1543	0.8048	0.6088	1.0549	3.7364	8.4185	20.6725
$N=1000$	2.1540	0.9934	0.7048	1.3888	3.2423	7.9002	20.4140
$N=5000$	2.1305	0.9931	0.7459	1.3317	3.1891	7.8808	20.3318
$N=10000$	2.1328	0.9780	0.7657	1.3524	3.1077	7.9140	20.2370

$$V_\pi = [2.1322, 0.9883, 0.7856, 1.3311, 3.1443, 7.9520, 20.3332]^T$$

增量式计算均值

- 对一组不断产生新数据的序列 $x_1, x_2, \ldots, x_{k-1}, x_k$
- 可以增量式地计算当前观测的均值 $\mu_1, \mu_2, \ldots, \mu_{k-1}, \mu_k$

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

- 只需要记录前一时刻的均值和当前时刻的观测量

增量式的 MC 更新

- 根据轨迹 $s_0, a_0, r_1, \ldots, s_T$ 对 $V(s)$ 增量式地更新
- e.g. 状态 s_t，回报是 G_t

$$N(s_t) \leftarrow N(s_t) + 1$$

$$V(s_t) \leftarrow V(s_t) + \frac{1}{N(s_t)}(G_t - V(s_t))$$

  - 随着状态遍历次数的增加，$1/N(s_t) \to 0$
  - 学习后期，观测量对结果影响不大

增量式的 MC 更新

- 根据轨迹 $s_0, a_0, r_1, \ldots, s_T$ 对 $V(s)$ 增量式地更新
- e.g. 状态 s_t，回报是 G_t

$$N(s_t) \leftarrow N(s_t) + 1$$

$$V(s_t) \leftarrow V(s_t) + \frac{1}{N(s_t)}(G_t - V(s_t))$$

  - 随着状态遍历次数的增加，$1/N(s_t) \to 0$
  - 学习后期，观测量对结果影响不大

- 如果环境是动态、不断变化的，更希望是能够随时跟踪当前不断变化的均值

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

  - 使用固定的学习率 α

- 策略的价值定义

$$V_\pi(s_t) = \mathbb{E}_\pi[G_t], \quad a_k \sim \pi(s_k)$$

$$\Downarrow$$

- 蒙特卡策略评估方法

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

  - 用观测代替期望，更新 V

- 策略的价值定义

$$V_\pi(s_t) = \mathbb{E}_\pi[G_t], \quad a_k \sim \pi(s_k)$$

$$\Downarrow$$

- 蒙特卡策略评估方法

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

  - 用观测代替期望，更新 V

- 策略价值的另一个定义：贝尔曼期望方程

$$V_\pi(s_t) = \mathbb{E}_\pi[r_{t+1} + \gamma V_\pi(s_{t+1})], \quad a_t \sim \pi(s_t)$$

$$\Downarrow$$

- 策略的价值定义

$$V_\pi(s_t) = \mathbb{E}_\pi[G_t], \quad a_k \sim \pi(s_k)$$

$$\Downarrow$$

- 蒙特卡策略评估方法

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

  - 用观测代替期望，更新 V

- 策略价值的另一个定义：贝尔曼期望方程

$$V_\pi(s_t) = \mathbb{E}_\pi[r_{t+1} + \gamma V_\pi(s_{t+1})], \quad a_t \sim \pi(s_t)$$

$$\Downarrow$$

- 用观测代替期望，更新 V

$$V(s_t) \leftarrow V(s_t) + \alpha\big(r_{t+1} + \gamma V_\pi(s_{t+1}) - V(s_t)\big)$$

  - 时间差分方法

时间差分学习

时间差分学习 Temporal Difference (TD) Learnin

- TD 方法根据智能体经历的轨迹学习
- **无模型的**: 不知道 MDP 问题的转移和奖励函数
- 特点是:
  - 可以根据**非完整**的轨迹学习, 借助**自举法**
  - 根据一个猜测值更新另一个猜测值

时间差分学习 Temporal Difference (TD) Learnin

- TD 方法根据智能体经历的轨迹学习
- **无模型的**: 不知道 MDP 问题的转移和奖励函数
- 特点是:
  - 可以根据**非完整**的轨迹学习, 借助**自举法**
  - 根据一个猜测值更新另一个猜测值

自举法 bootstrapping

统计学上自举法是一种通过对样本进行重采样得到的估计总体的方法。
通俗地讲是通过**自身的** 力量，自己把自己抬起来

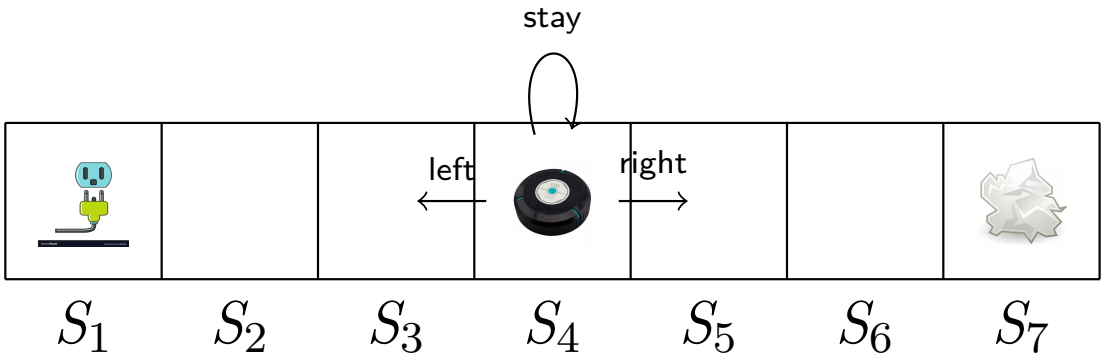
- 当前已有一个猜测的价值 V，期望减少和真实 V_π 的误差
- 由它自身定义的 $r_{t+1} + \gamma V(s_{t+1})$ 作为对 $V(s_t)$ 更精确的估计

TD(0) 学习算法

1: 给定策略 π, 初始状态分布 D_0, $V(s) = 0, \forall s \in \mathcal{S}$, 学习率 α, $t = 0$
2: **loop**
3: **if** $s_t = S_{terminal}$ 或 s_t 未初始化 **then**
4: 初始化 $s_t \sim D_0$
5: **end if**
6: 采样动作 $a_t \sim \pi(s_t)$, 执行 a_t 后观察 r_{t+1}, s_{t+1}
7: 根据 $(s_t, a_t, r_{t+1}, s_{t+1})$ 更新
$V(s_t) \leftarrow V(s_t) + \alpha\big(r_{t+1} + \gamma V(s_{t+1}) - V(s_t)\big)$
8: $t \leftarrow t + 1$
9: **end loop**

- $r_{t+1} + \gamma V(s_{t+1})$ 称为 TD 目标
- $\delta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t)$ 称为 TD 误差

举例: 扫地机器人



- 策略: $\pi(left|s) = 0.5, \quad \pi(right|s) = 0.5$
- 矩阵求解价值函数

$$v_\pi = [2.1322, 0.9883, 0.7856, 1.3311, 3.1443, 7.9520, 20.3332]^T$$

- 迭代计算结果

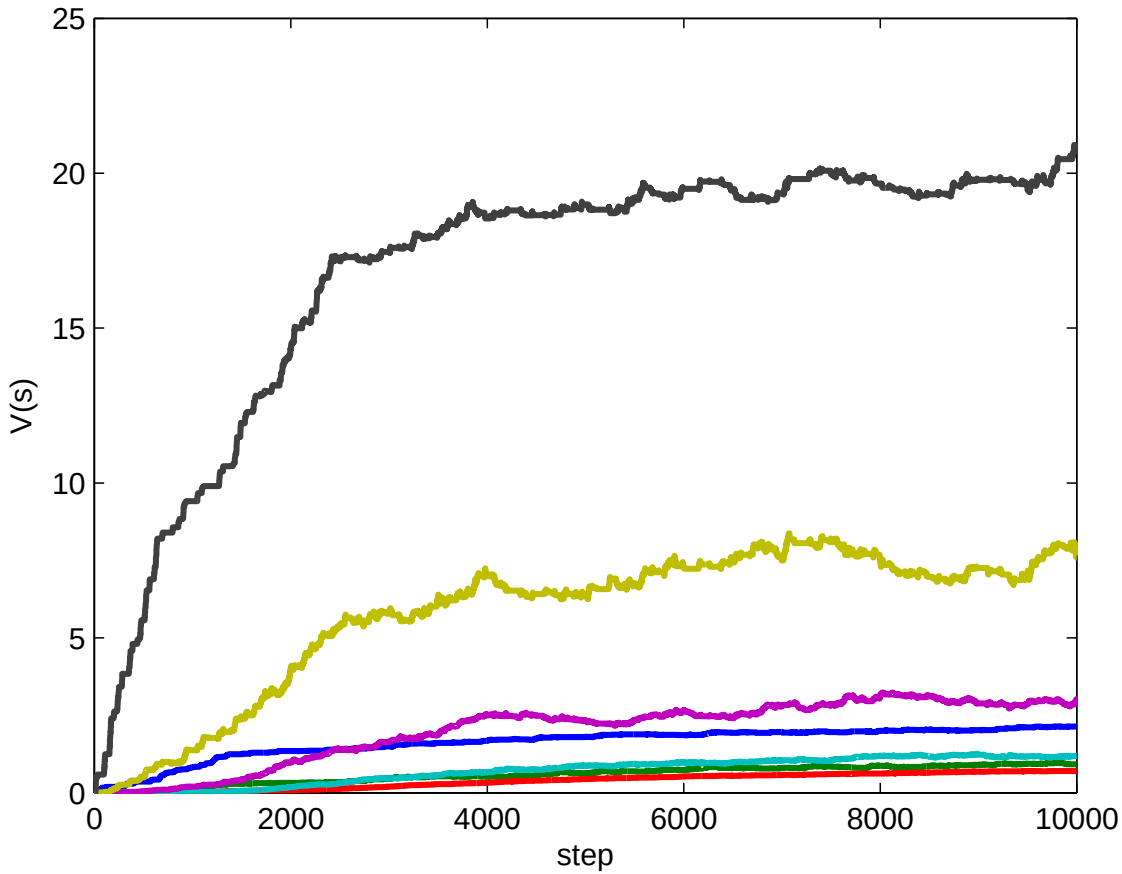
$$v_{22} = [2.1305, 0.9865, 0.7837, 1.3290, 3.1421, 7.9497, 20.3308]$$

- 蒙特卡洛方法

$$v_{10000} = [2.1328, 0.9780, 0.7657, 1.3524, 3.1077, 7.9140, 20.2370]$$

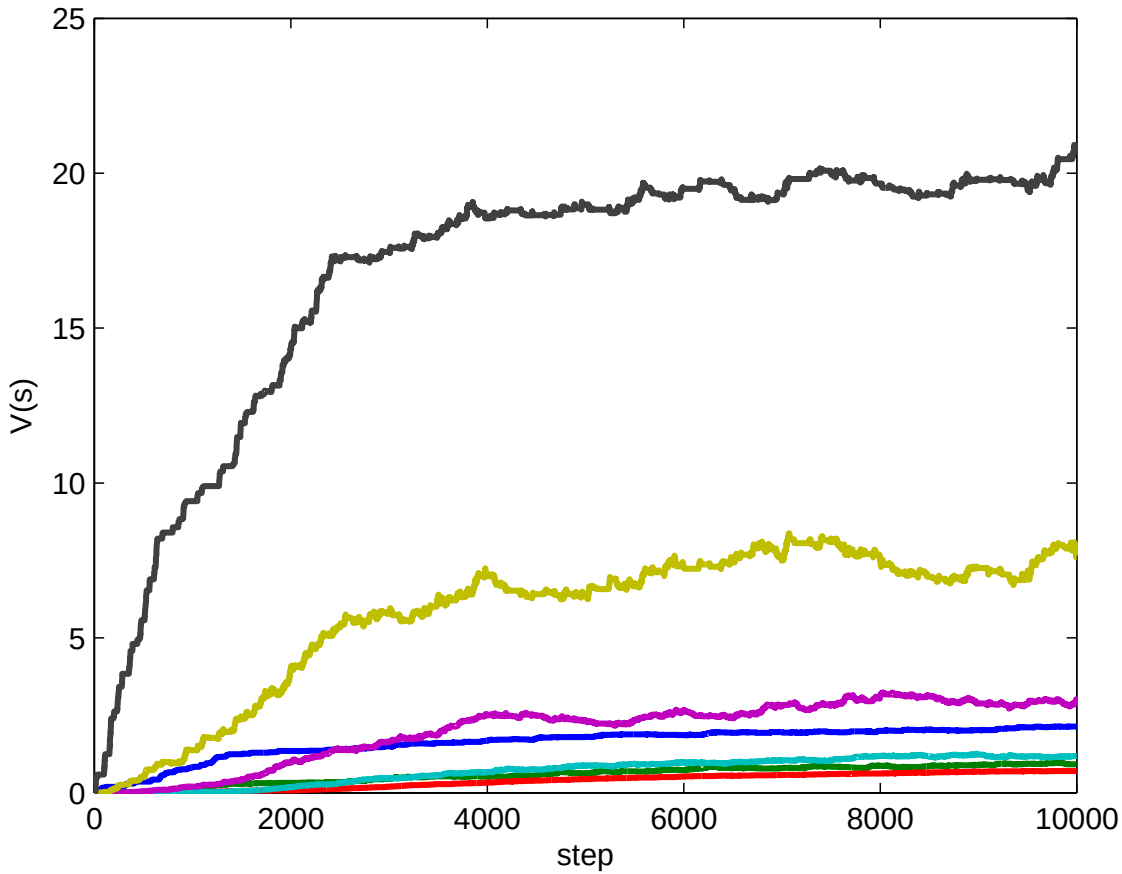
TD 学习结果 (1)

- 初始化 $V(s) = 0,\ D_0 = S_1,\ \alpha = 0.01$
- 学习时长 $N = 10000$



TD 学习结果 (1)

- 初始化 $V(s) = 0$, $D_0 = S_1$, $\alpha = 0.01$
- 学习时长 $N = 10000$

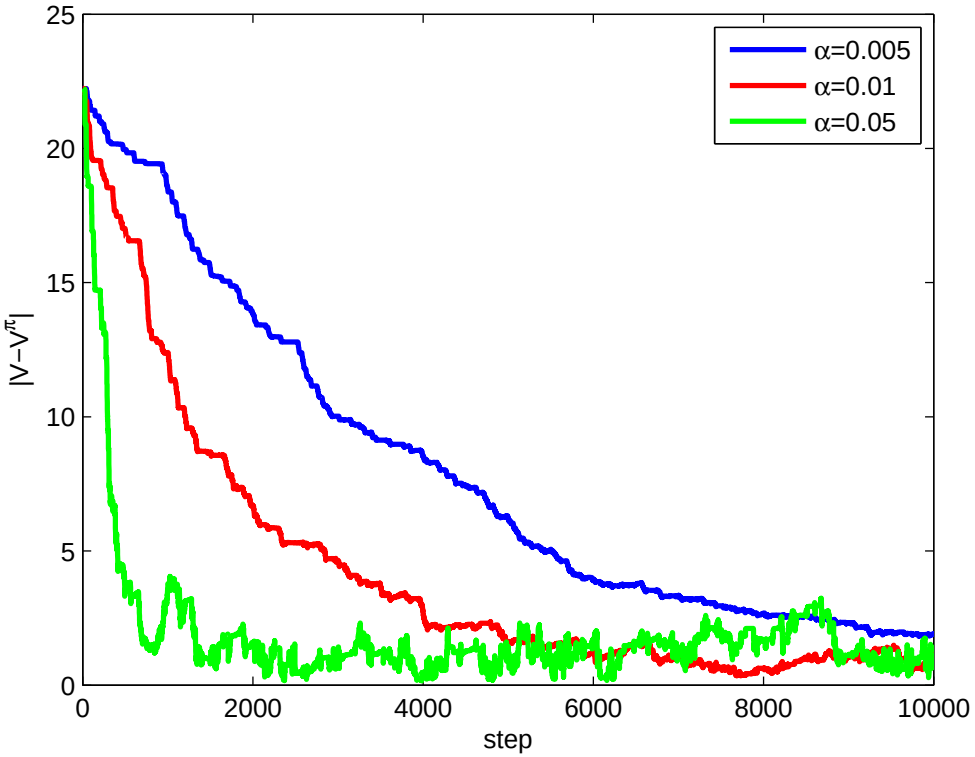


- 同 MC 方法一样，观测越多，越接近真实结果

TD 学习结果 (2)

$$V(s_t) \leftarrow V(s_t) + \alpha\big(r_{t+1} + \gamma V(s_{t+1}) - V(s_t)\big)$$

- 考虑不同的学习率 $\alpha = 0.005, 0.01, 0.05$
- 观测学习过程中 V 和真实 V_π 之间的误差



1. α 小，学习慢，曲线平滑
2. α 大，学习快，震荡明显

MC 和 TD 对比 (1)

- TD 可以在智能体运行过程中的每一时刻在线更新
- MC 需要完整的轨迹计算出回报后更新
- TD 可以根据不完整的轨迹学习
- MC 要求轨迹达到终止状态或序列足够长

偏差/方差的权衡

- 回报 $G_t = r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{T-1} r_t$ 是策略价值 $V_\pi(s_t)$ 的 无偏估计
  - $V_\pi(s_t) = \mathbb{E}_\pi[G_t]$
- 真实的 TD 目标 $r_{t+1} + \gamma V_\pi(s_{t+1})$ 是 $V_\pi(s_t)$ 的 无偏估计
  - $V_\pi(s_t) = \mathbb{E}_\pi[r_{t+1} + \gamma V_\pi(s_{t+1})]$
- 实际的 TD 目标 $r_{t+1} + \gamma V(s_{t+1})$ 是 $V_\pi(s_t)$ 的 有偏估计
- 但是 TD 目标比回报具有 更小的方差

偏差/方差的权衡

- 回报 $G_t = r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{T-1} r_t$ 是策略价值 $V_\pi(s_t)$ 的 无偏估计
  - $V_\pi(s_t) = \mathbb{E}_\pi[G_t]$
- 真实的 TD 目标 $r_{t+1} + \gamma V_\pi(s_{t+1})$ 是 $V_\pi(s_t)$ 的 无偏估计
  - $V_\pi(s_t) = \mathbb{E}_\pi[r_{t+1} + \gamma V_\pi(s_{t+1})]$
- 实际的 TD 目标 $r_{t+1} + \gamma V(s_{t+1})$ 是 $V_\pi(s_t)$ 的 有偏估计
- 但是 TD 目标比回报具有 更小的方差
  - 回报的计算涉及 整个轨迹 上的随机动作，转移状态，奖励 (随机因素多)
  - TD 目标只包含 一个时刻 的随机动作，转移状态，奖励 (随机因素少)

MC 和 TD 对比 (2)

- MC 是 高方差, 零偏差
  - 好的收敛性
  - (即使是使用逼近器也能保证收敛)
  - 与价值函数初始值无关
  - 原理简单, 使用方便
- TD 是 低方差, 有偏差
  - 通常情况是比 MC 更有效
  - TD(0) 能够收敛到 $V_\pi(s)$
  - (但是与逼近器结合后没有收敛保证)
  - 受价值函数初始值影响

MC 和 TD 的批量学习

- 对有限 MDPs 问题, MC 和 TD 都是收敛的:
  当观测量 $\to \infty$ 时, $V(s) \to V_\pi(s)$
- 但是如果我们只有有限的经验, 如何根据这些有限的经验提高学习的效果?

$$
\begin{gathered}
s_0^1, a_0^1, r_1^1, \ldots, s_{T_1}^1 \\
\vdots \\
s_0^K, a_0^K, r_1^K, \ldots, s_{T_K}^K
\end{gathered}
$$

MC 和 TD 的批量学习

- 对有限 MDPs 问题, MC 和 TD 都是收敛的:
  当观测量 $\to \infty$ 时, $V(s) \to V_\pi(s)$
- 但是如果我们只有有限的经验, 如何根据这些有限的经验提高学习的效果?

$$s_0^1, a_0^1, r_1^1, \ldots, s_{T_1}^1$$
$$\vdots$$
$$s_0^K, a_0^K, r_1^K, \ldots, s_{T_K}^K$$

- 批量学习:
  - 反复地从上述经验中随机选取选取轨迹 $k \in [1, K]$
  - 对轨迹 k 使用 MC 或 TD(0) 学习

举例: AB 问题

MDP: 两个状态 A, B; 无折扣; 8 段轨迹的经验

- A, 0, B, 0
- B, 1
- B, 1
- B, 1
- B, 1
- B, 1
- B, 1
- B, 0

那么 $V(A), V(B)$ 等于多少

- 每次经过的 MC 结果
  - $V(A) = mean(G_t(A)) = 0$
  - $V(B) = mean(G_t(B)) = 0.75$
- TD(0) 结果

$$V(s_t) \leftarrow V(s_t) + \alpha(r_{t+1} + \gamma V(s_{t+1}) - V(s_t))$$

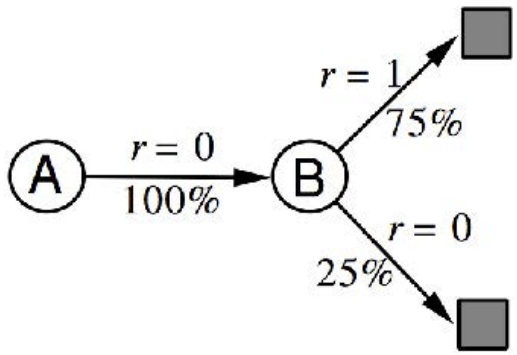
  - $V(B) = 0.75$
  - $V(A) = 0.75$

分析

- MC 收敛结果对应最小二乘误差
  - 最佳匹配观测 (s_t, G_t) 的回报

$$\sum_{k=1}^{K}\sum_{t=1}^{T_k}(G_t^k - V(s_t^k))^2$$

- TD(0) 收敛结果对应最大似然马尔可夫模型
  - 最佳匹配观测数据 $(s_t, a_t, r_{t+1}, s_{t+1})$ 的 MDP 模型 $\langle \mathcal{S}, \mathcal{A}, \hat{\mathcal{P}}, \hat{\mathcal{R}}, \gamma \rangle$



$$\hat{\mathcal{P}}_{s,s'}^a = \frac{1}{N(s,a)}\sum_{k=1}^{K}\sum_{t=1}^{T_k}\mathcal{I}(s_t^k, a_t^k, s_{t+1}^k = s, a, s')$$

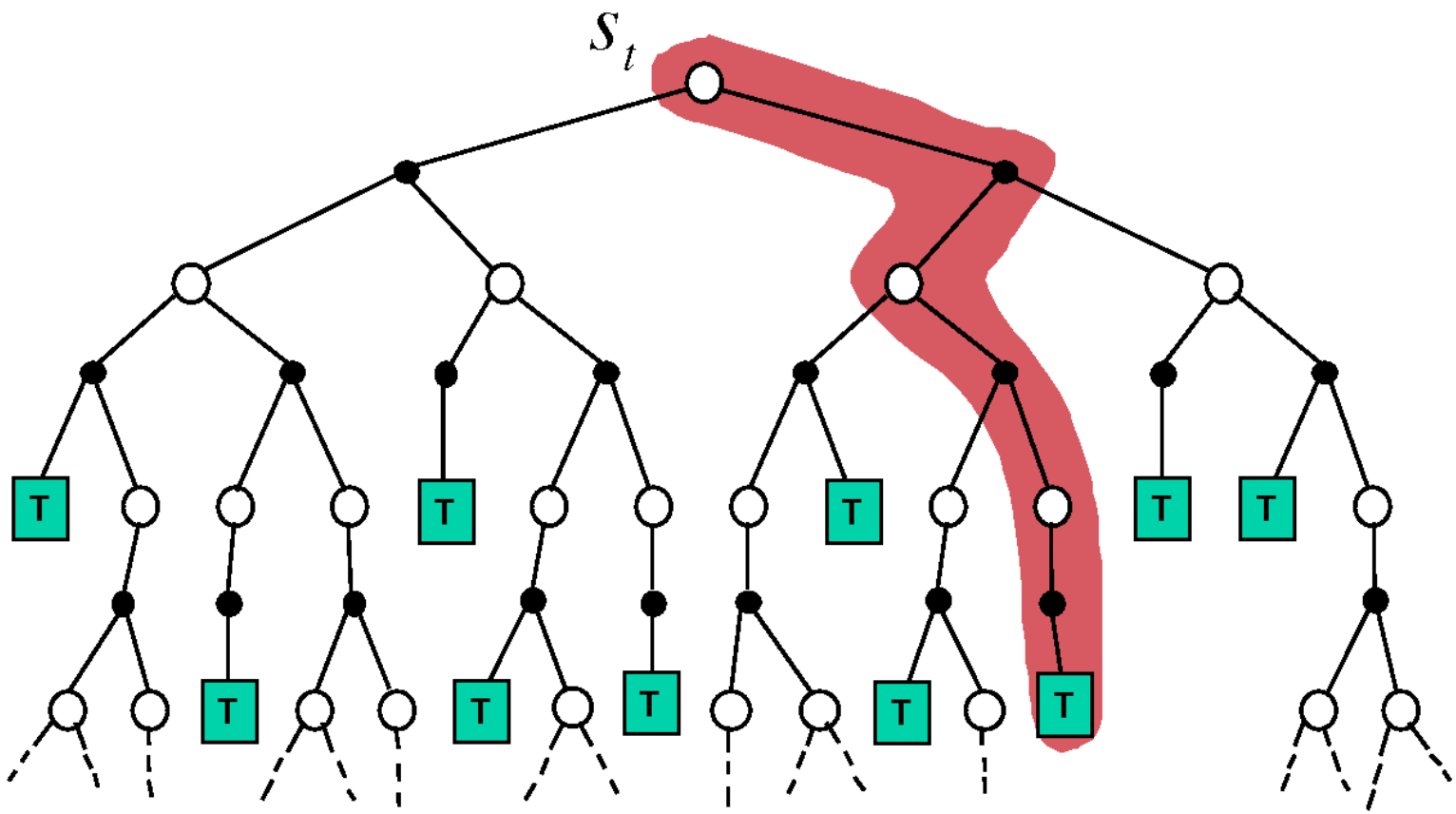
$$\hat{\mathcal{R}}_s^a = \frac{1}{N(s,a)}\sum_{k=1}^{K}\sum_{t=1}^{T_k}\mathcal{I}(s_t^k, a_t^k = s, a) r_t^k$$

MC 和 TD 对比 (3)

- TD 能够利用马尔可夫性
  - 需要知道 s_t, a_t 后面是谁 (s_{t+1})
  - 在 马尔可夫环境下更有效
- MC 不能利用马尔可夫性
  - 不需要知道 s_t 后面是谁 (s_{t+1})，只要知道 G_t 就行
  - 在 非马尔可夫环境下同样有效

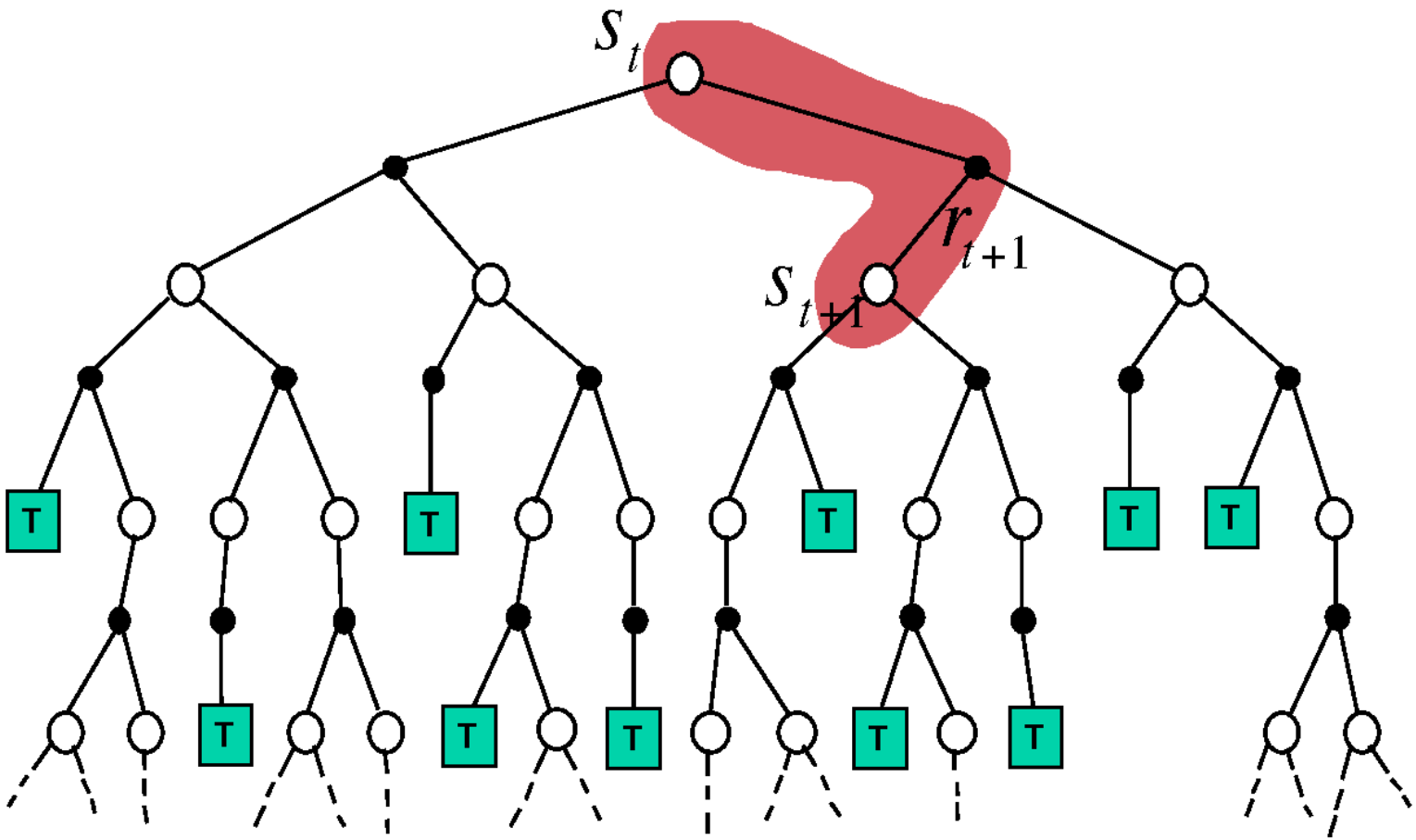
蒙特卡洛方法

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



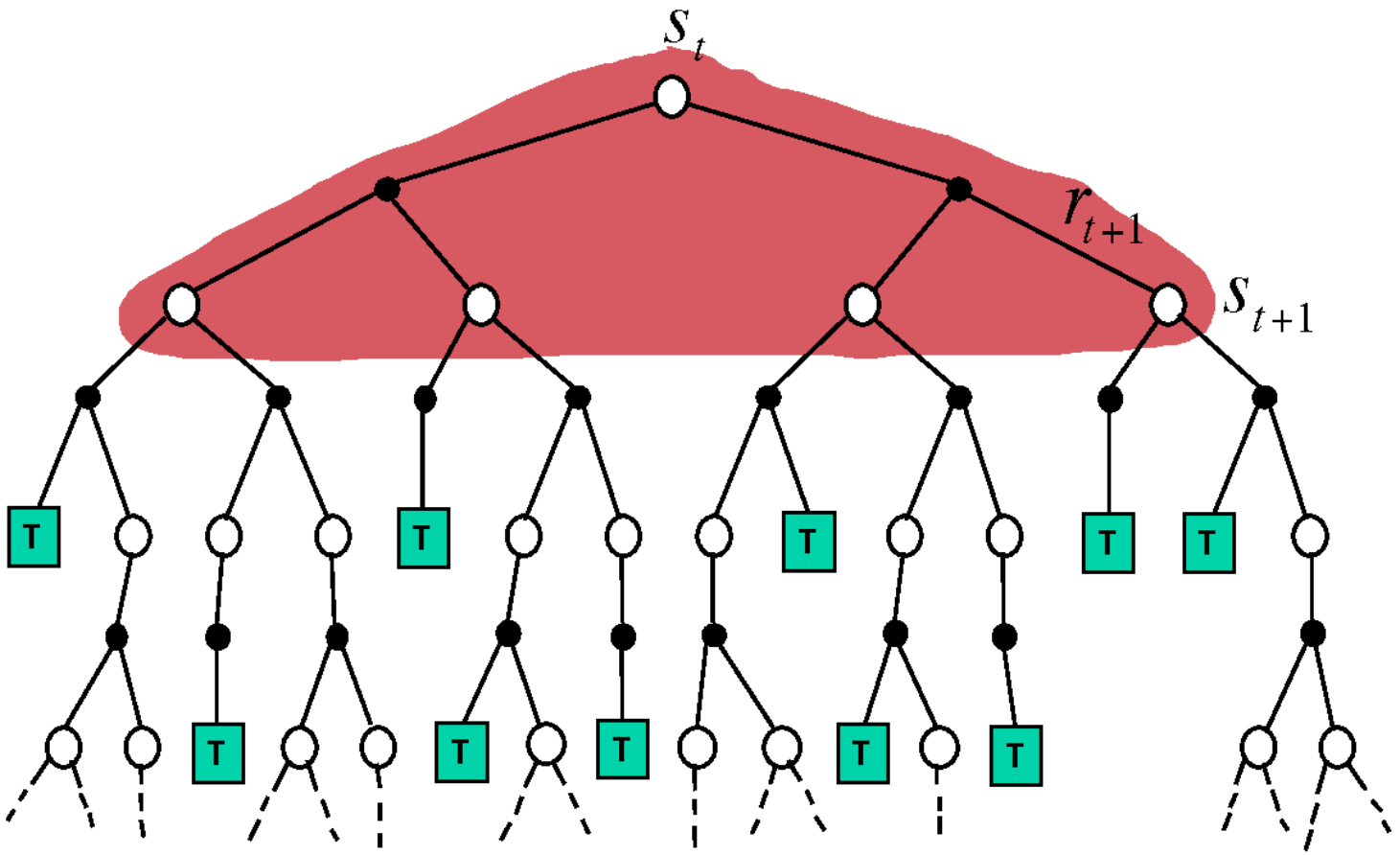
时间差分方法

$$V(s_t) \leftarrow V(s_t) + \alpha\big(r_{t+1} + \gamma V(s_{t+1}) - V(S_t)\big)$$



动态规划方法

$$V(s_t) \leftarrow \mathbb{E}_\pi[r_{t+1} + \gamma V(s_{t+1})]$$



自举和采样角度区别 MC/TD/DP

- 自举法：更新时包含一个猜测量
  - MC 不使用自举
  - TD 使用自举
  - DP 使用自举
- 采样法：使用采样的数据计算期望
  - MC 使用采样
  - TD 使用采样
  - DP 不使用采样

n-步回报

- TD 的学习目标：1 步回报

$$G_t^{(1)} = r_{t+1} + \gamma V(s_{t+1})$$

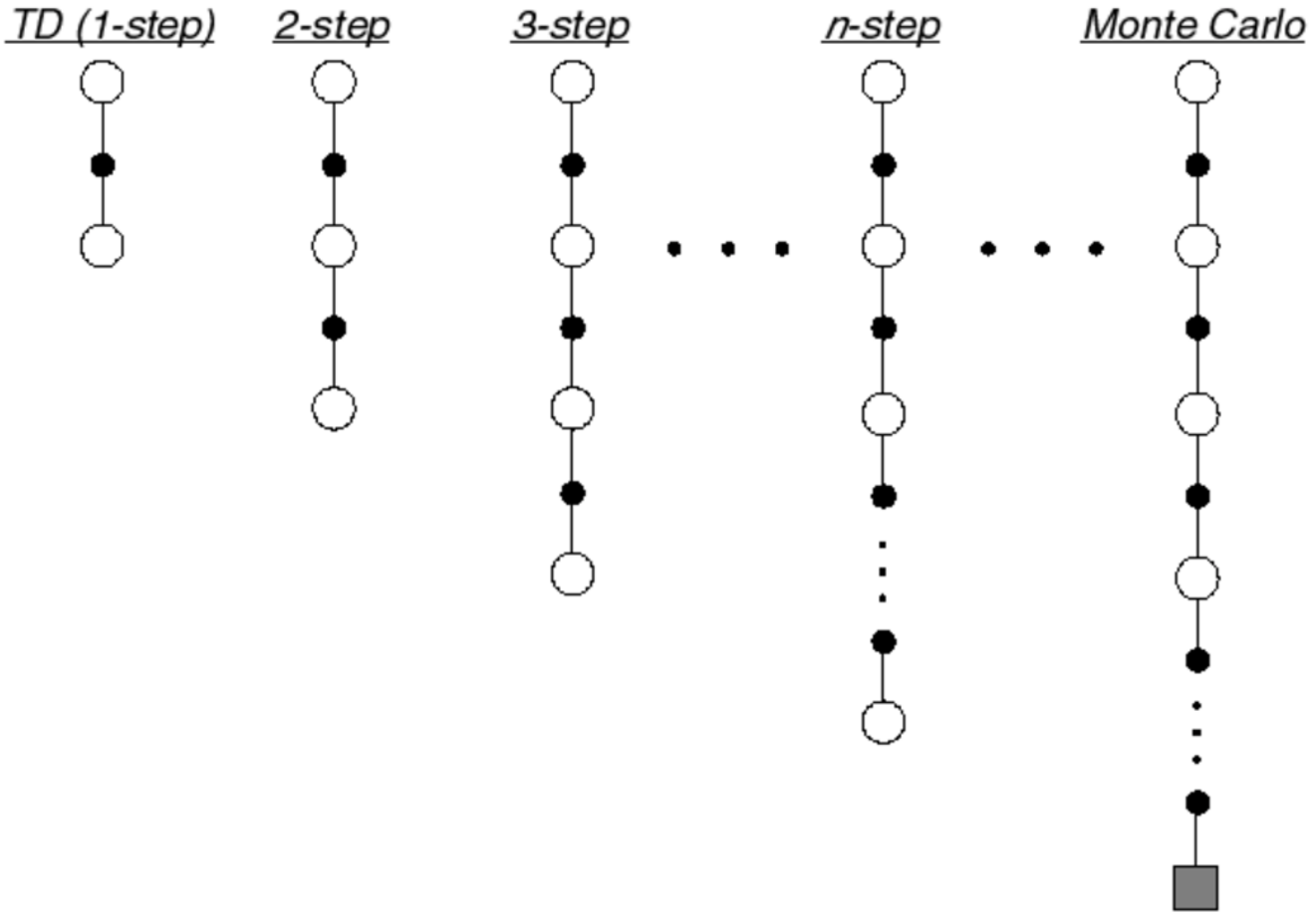
- MC 的学习目标：无穷步回报

$$G_t = r_{t+1} + \gamma r_{t+2} + \dots$$

- 是否可以两者结合，在 TD 学习中增加回报的计算步数？

n-步预测

- 让 TD 目标继续向后观测 n 步



n-步回报

- 考虑如下几种 n-步回报

$$
\begin{aligned}
&n=1 \quad (TD) && G_t^{(1)} = r_{t+1} + \gamma V(s_{t+1}) \\
&n=2 && G_t^{(2)} = r_{t+1} + \gamma r_{t+2} + \gamma^2 V(s_{t+2}) \\
&\vdots \\
&n=\infty \quad (MC) && G_t^{(\infty)} = r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{t+T-1} r_T
\end{aligned}
$$

- 定义 n-步的回报

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

- n-步时间差分学习

$$V(s_t) \leftarrow V(s_t) + \alpha\big(G_t^{(n)} - V(s_t)\big)$$

1-步回报的误差上界

- 1 步回报和真实 V_π 之间的期望误差：

$$\begin{aligned}\max_{s_t}\left|\mathbb{E}[G_t^{(1)}]-V_\pi(s_t)\right| &= \max_{s_t}\left|\mathbb{E}[r_{t+1}+\gamma V(s_{t+1})]-V_\pi(s_t)\right| \\ &= \max_{s_t}\left|[\mathcal{T}^\pi(V)](s_t)-V_\pi(s_t)\right| \\ &\leq \gamma\|V-V_\pi\|\end{aligned}$$

其中 $\mathcal{T}^\pi$ 是贝尔曼期望算子，$\mathcal{T}^\pi(V_\pi)=\mathcal{R}^\pi+\gamma\mathcal{P}^\pi V_\pi$

n-步回报的误差上界

- n-步回报的期望误差

$$
\begin{aligned}
&\max_{s_t} \left| \mathbb{E}[G_t^{(n)}] - V_\pi(s_t) \right| \\
=&\max_{s_t} \left| \mathbb{E}[r_{t+1} + \gamma r_{t+1} + \dots \gamma^n V(s_{t+n})] - V_\pi(s_t) \right| \\
=&\max_{s_t} \left| \mathbb{E}[r_{t+1} + \gamma \mathbb{E}[r_{t+2} + \dots + \gamma \mathbb{E}[r_{t+n} + \gamma V(s_{t+n})] \dots]] - V_\pi(s_t) \right| \\
=&\max_{s_t} \left| [\underbrace{\mathcal{T}^\pi \circ \dots \circ \mathcal{T}^\pi}_{n}(V)](s_t) - V_\pi(s_t) \right| \\
\leq&\gamma^n \| V - V_\pi \|
\end{aligned}
$$

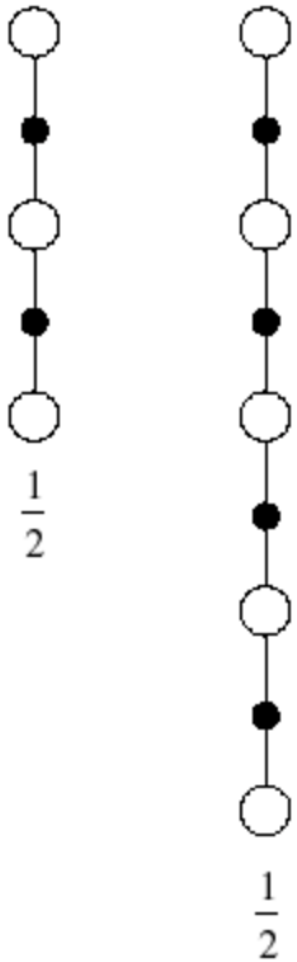
- n-步回报对应更准确的价值估计
  - n 越大，估计和真实价值之间的误差上界越低
  - 但是估计的方差会变大，e.g. $n \to \infty$ 变成 MC 方法
- n 较小时，价值估计准确性低，方差小
- n 较大时，价值估计准确性高，方差大

均值化 n-步回报

- 对不同的 n 求它们 n-步回报的平均
- 例如对 2-步和 4-步回报求均值

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

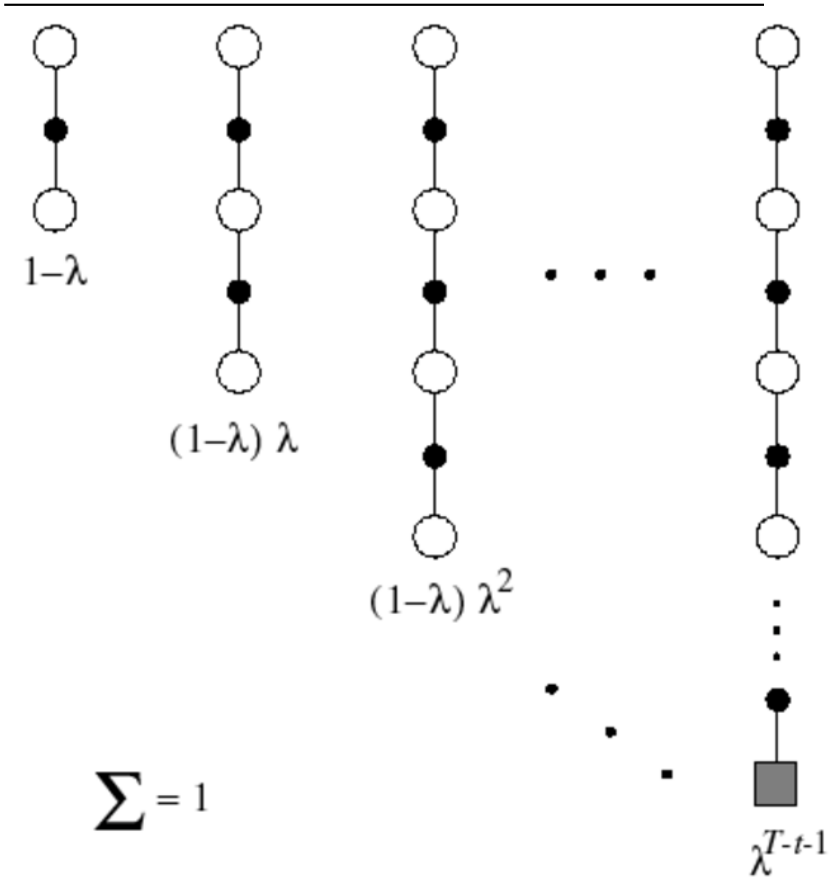
- 结果结合了来自两种步数的回报信息
- 那是否可以将所有步数的回报整合在一起?



TD(λ)

λ-回报

TD(λ), λ-回报



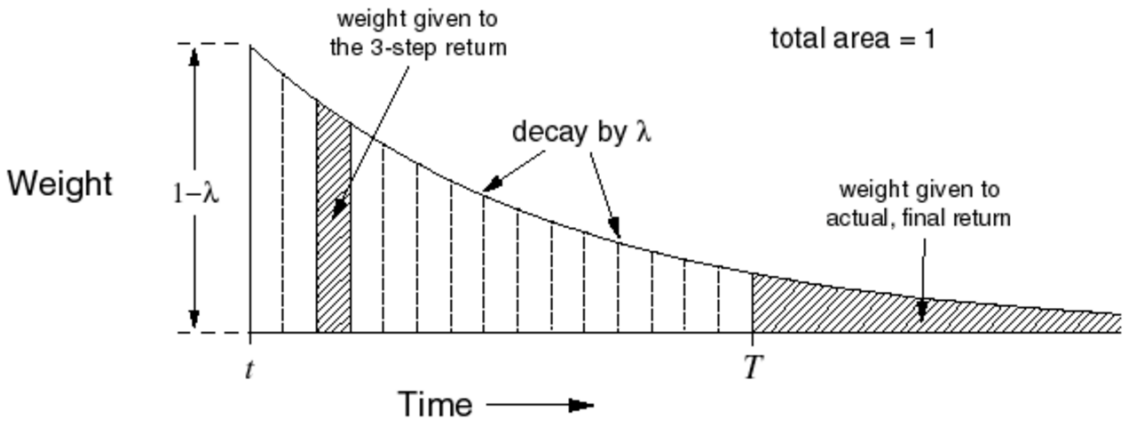
- λ-回报 G_t^λ 将所有的 n-步回报 $G_t^{(n)}$ 整合在一起，$0 < \lambda < 1$
- 对每项使用权重 $(1-\lambda)\lambda^{n-1}$

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

- 前向更新的 TD(λ) 形式

$$V(s_t) \leftarrow V(s_t) + \alpha\big(G_t^\lambda - V(s_t)\big)$$

TD(λ) 权重函数



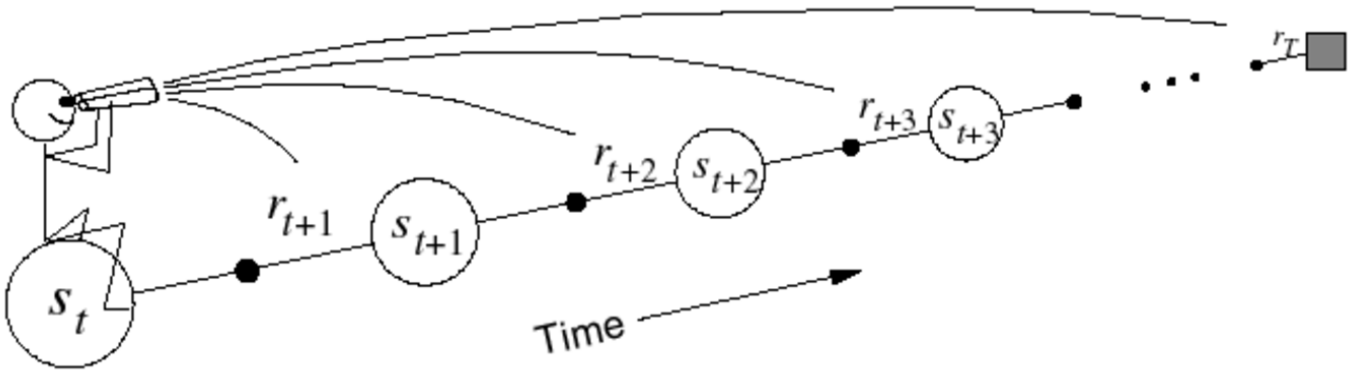
$$G_t^\lambda = (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}G_t^{(n)}$$ 无穷长轨迹

$$G_t^\lambda = (1-\lambda)\sum_{n=1}^{T-t-1}\lambda^{n-1}G_t^{(n)} + \lambda^{T-t-1}G_t$$ 轨迹在 T 时刻终止

- n 比较大的 $G_t^{(n)}$，权重 $(1-\lambda)\lambda^{n-1}$ 比较低
- 好处：既利用了长步数估计的精度，又降低了高方差的影响

前向 TD(λ)

$$V(s_t) \leftarrow V(s_t) + \alpha\big(G_t^{\lambda} - V(s_t)\big)$$



- 更新价值函数向 λ-回报逼近
- 需要未来时刻的观测计算 G_t^{λ}
- 与 MC 一样要求完整的轨迹
- 离线学习

- 前向 TD(λ) 无法在线学习的原因是：
  - 对当前时刻 $V(s_t)$ 的更新依赖未来时刻的观测
  - 只有到未来时刻才能知道如何对 $V(s_t)$ 更新
- 那么，能否定义一个变量代表状态 S 在过去被访问的信息
- 每个新时刻，利用新观测的数据，根据这一变量对所有状态进行更新？

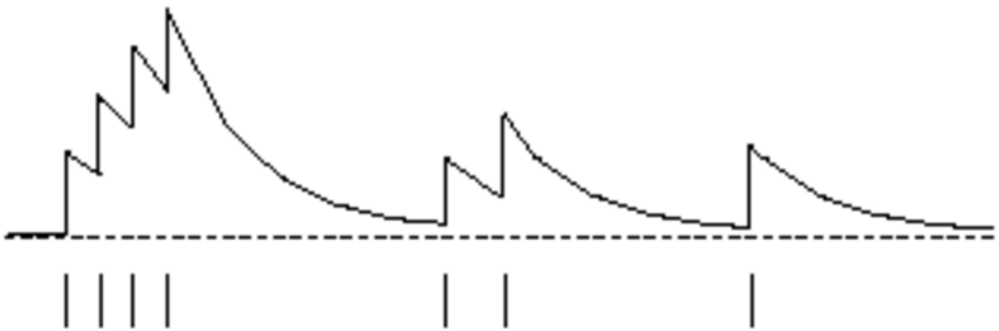
资格迹

资格迹 Eligibility Traces

- 这样的变量称为 资格迹
- 对有限 MDP 问题，会为每个状态定义一个资格迹
  $E_0(S) = 0, \forall S \in \mathcal{S}$
- 每个时刻对所有的资格迹衰减 $\gamma\lambda$，同时对观测到的状态的资格迹加 1

$$E_t(S) = \begin{cases} \gamma\lambda E_{t-1}(S) & \text{if } S \neq s_t \\ \gamma\lambda E_{t-1}(S) + 1 & \text{if } S = s_t \end{cases}$$
$$= \gamma\lambda E_{t-1}(S) + \mathcal{I}(S = ss_t)$$

  - 反映了状态被观测的次数和频率



该状态随时间变化的资格迹

沿时间轴状态 S 出现的次数

后向 TD(λ)

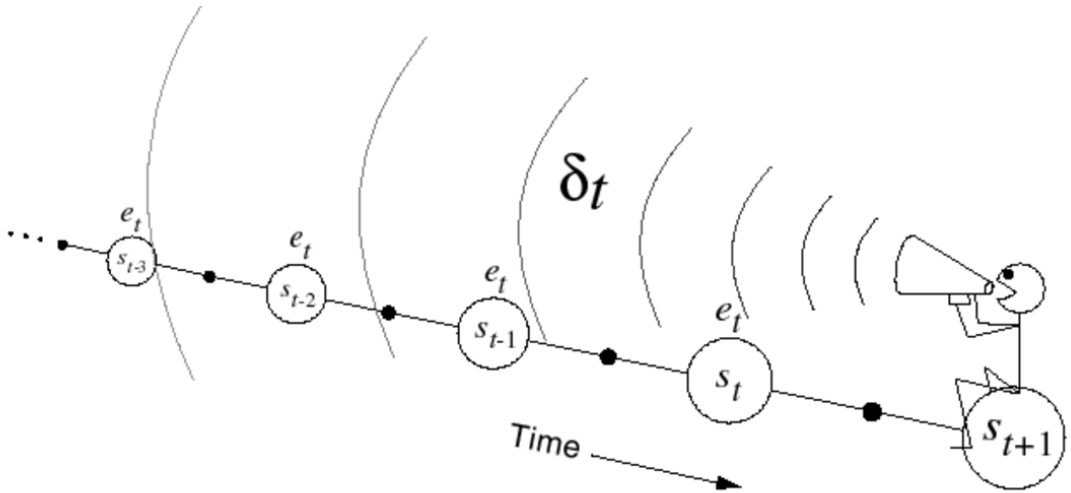
- 资格迹代表了出现一次强化信号时，需要对每个状态的更新程度
- 这里强化信号指的是每个时刻的 1 步 TD 误差

$$\delta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t)$$

- 所有状态的价值变化量等于 TD 误差乘上相应的资格迹

$$V(S) \leftarrow V(S) + \alpha \delta_t E_t(S), \quad \forall S \in \mathcal{S}$$

- 后向 TD(λ) 会在每个时刻进行一次上述的更新过程



在线 TD(λ) 算法

任意初始化 $V(S)$
repeat {对每个 episode:}
  $E(S) = 0, \forall S \in \mathcal{S}$
  初始化 s
  repeat {对 episode 的每一步:}
    $a \sim \pi(s)$
    执行 a 后观测奖励 r 和下一状态 s'
    $\delta \leftarrow r + \gamma V(s') - V(s)$
    $E(s) \leftarrow E(s) + 1$
    for all $S \in \mathcal{S}$ **do**
      $V(S) \leftarrow V(S) + \alpha\delta E(S)$
      $E(S) \leftarrow \gamma\lambda E(S)$
    end for
    $s \leftarrow s'$
  until s 是终止状态
until

TD(0) 和 TD

- 当 $\lambda = 0$ 时, TD(λ) 变成

$$E_t(S) = \begin{cases} 0 & \text{if } S \neq s_t \\ 1 & \text{if } S = s_t \end{cases}$$

$$\Delta V_t(S) = \begin{cases} 0 & \text{if } S \neq s_t \\ \alpha \cdot \delta_t \cdot 1 & \text{if } S = s_t \end{cases}$$

- 等同于前面介绍的 TD 方法

$$V(s_t) \leftarrow V(s_t) + \alpha\delta_t$$

TD(1) 和 MC

- 考虑在一次轨迹中 S 在 k 时刻被唯一访问一次
- $\lambda = 1$ 下该状态的 TD(1) 资格迹会在被访问之后随时间衰减

$$
\begin{aligned}
E_t(S) &= \gamma E_{t-1}(S) + \mathcal{I}(S = s_t) \\
&= \begin{cases} 0 & \text{if } t < k \\ \gamma^{t-k} & \text{if } t \geq k \end{cases}
\end{aligned}
$$

- TD(1) 在整条轨迹上对 S 的累计更新量等于

$$
\begin{aligned}
\sum_{t=0}^{T} \alpha \delta_t E_t(S) &= \alpha \sum_{t=k}^{T} \gamma^{t-k} \delta_t \\
&= \delta_k + \gamma \delta_{k+1} + \gamma^2 \delta_{k+2} + \cdots + \gamma^{T-k} \delta_T
\end{aligned}
$$

- 假定 TD(1) 只在轨迹结束后对 V 更新 (离线学习)
- 累计量中的 V 保持不变

$$
\begin{aligned}
&\delta_k + \gamma\delta_{k+1} + \gamma^2\delta_{k+2} + \cdots + \gamma^{T-k}\delta_T \\
=&r_{k+1} + \gamma\, V(s_{k+1}) - V(s_k) \\
&+ \gamma r_{k+2} + \gamma^2\, V(s_{k+2}) - \gamma\, V(s_{k+1}) \\
&+ \gamma^2 r_{k+3} + \gamma^3\, V(s_{k+3}) - \gamma^2\, V(s_{k+2}) \\
&\vdots \\
&+ \gamma^{T-k} r_{T+1} - \gamma^{T-k}\, V(S_T) \\
=&r_{k+1} + \gamma r_{k+2} + \gamma^2 r_{k+3} + \cdots + \gamma^{T-k} r_{T+1} - V(S_k) \\
=&G_k - V(s_k)
\end{aligned}
$$

- 离线 TD(1) 方法在同一条轨迹对某一状态的累计更新量等于对该状态的 MC 更新

$$\sum_{t=0}^{T} \alpha \delta_t E_t(S) = \alpha\big(G_k - V(s_k)\big)$$

- 但是实际往往使用在线 TD(1) 方法：
  - 在线，增量式，学习效率高
  - 对比 MC 要求整条轨迹结束后再更新

前向 TD(λ) 和后向 TD(λ) 的等价性

定理

当使用离线更新时，前向 TD(λ) 和后向 TD(λ) 在同一条轨迹上的更新量是相同的

$$\sum_{t=0}^{T} \Delta V_t^{TD}(S) = \sum_{t=0}^{T} \Delta V_t^{\lambda}(s_t) \mathcal{I}_{S=s_t}, \quad \forall S \in \mathcal{S}$$

- 离线更新：在轨迹结束后根据轨迹上的观测量对 V 更新
- $\Delta V_t^{TD}(S)$ 代表后向 TD(λ) 在 t 时刻在 S 上的更新量：
  $\Delta V_t^{TD}(S) = \alpha \delta_t E_t(S)$
- $\Delta V_t^{\lambda}(s_t)$ 代表前向 TD(λ) 在 t 时刻对观测量 s_t 的更新：
  $\Delta V_t^{\lambda}(s_t) = \alpha(G_t^{\lambda} - V_t(s_t))$
- $\mathcal{I}_{S=s_t}$ 是指示函数，当 $S = s_t$ 时输出 1，否则输出 0

定理证明

- 首先轨迹上的资格迹等于

$$E_t(S) = \sum_{k=0}^{t} (\gamma\lambda)^{t-k} \mathcal{I}_{S=s_k}$$

- 因此后向 TD(λ) 轨迹上的更新量写成

$$\begin{aligned}
\sum_{t=0}^{T} \Delta V_t^{TD}(S) &= \sum_{t=0}^{T} \alpha\delta_t \sum_{k=0}^{t} (\gamma\lambda)^{t-k} \mathcal{I}_{S=s_k} \\
&= \sum_{k=0}^{T} \alpha \sum_{t=0}^{k} (\gamma\lambda)^{k-t} \mathcal{I}_{S=s_t} \delta_k \\
&= \sum_{t=0}^{T} \alpha \sum_{k=t}^{T} (\gamma\lambda)^{k-t} \mathcal{I}_{S=s_t} \delta_k \\
&= \sum_{t=0}^{T} \alpha \mathcal{I}_{S=s_t} \sum_{k=t}^{T} (\gamma\lambda)^{k-t} \delta_k
\end{aligned}$$

- 前向 TD(λ) 的一次更新等于

$$
\begin{aligned}
\frac{1}{\alpha}\Delta V_t^{\lambda}(s_t) =& G_t^{\lambda} - V_t(s_t) \\
=& - V_t(s_t) + (1-\lambda)\lambda^0[r_{t+1} + \gamma V_t(s_{t+1})] \\
&\quad + (1-\lambda)\lambda^1[r_{t+1} + \gamma r_{t+2} + \gamma^2 V_t(s_{t+2})] \\
&\quad + (1-\lambda)\lambda^2[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 V_t(s_{t+3})] \\
&\quad\quad \vdots \qquad \vdots \qquad \vdots \qquad \ddots \\
=& - V_t(s_t) \\
&+ (\gamma\lambda)^0[r_{t+1} + \gamma V_t(s_{t+1}) - \gamma\lambda V_t(s_{t+1})] \\
&+ (\gamma\lambda)^1[r_{t+2} + \gamma V_t(s_{t+2}) - \gamma\lambda V_t(s_{t+2})] \\
&+ (\gamma\lambda)^2[r_{t+3} + \gamma V_t(s_{t+3}) - \gamma\lambda V_t(s_{t+3})] \\
&\vdots \\
=& (\gamma\lambda)^0[r_{t+1} + \gamma V_t(s_{t+1}) - V_t(s_t)] \\
&+ (\gamma\lambda)^1[r_{t+2} + \gamma V_t(s_{t+2}) - V_t(s_{t+1})] \\
&+ (\gamma\lambda)^2[r_{t+3} + \gamma V_t(s_{t+3}) - V_t(s_{t+2})] \\
&\vdots
\end{aligned}
$$

$$\begin{aligned}\frac{1}{\alpha}\Delta V_t^{\lambda}(s_t) &= G_t^{\lambda} - V_t(s_t) \\ &= \sum_{k=t}^{\infty}(\gamma\lambda)^{k-t}\delta_k \qquad (\text{无穷长轨迹}) \\ &= \sum_{k=t}^{T}(\gamma\lambda)^{k-t}\delta_k \qquad (s_T \text{ 是终止状态})\end{aligned}$$

- 所以

$$\sum_{t=0}^{T}\Delta V_t^{TD}(S) = \sum_{t=0}^{T}\Delta V_t^{\lambda}(s_t)\mathcal{I}_{S=s_t}$$

前向和后向 TD(λ)

- 前向提供理论依据
- 后向给出算法实现
- 在离线更新时两者等价
- 但实际应用时往往使用 在线的后向 TD(λ) 方法
  - 在线学习
  - 每一时刻都更新
  - 可以适用轨迹中的一小段, 非完整的轨迹

总结

蒙特卡洛方法

时间差分学习

n-步回报

TD(λ)

资格迹