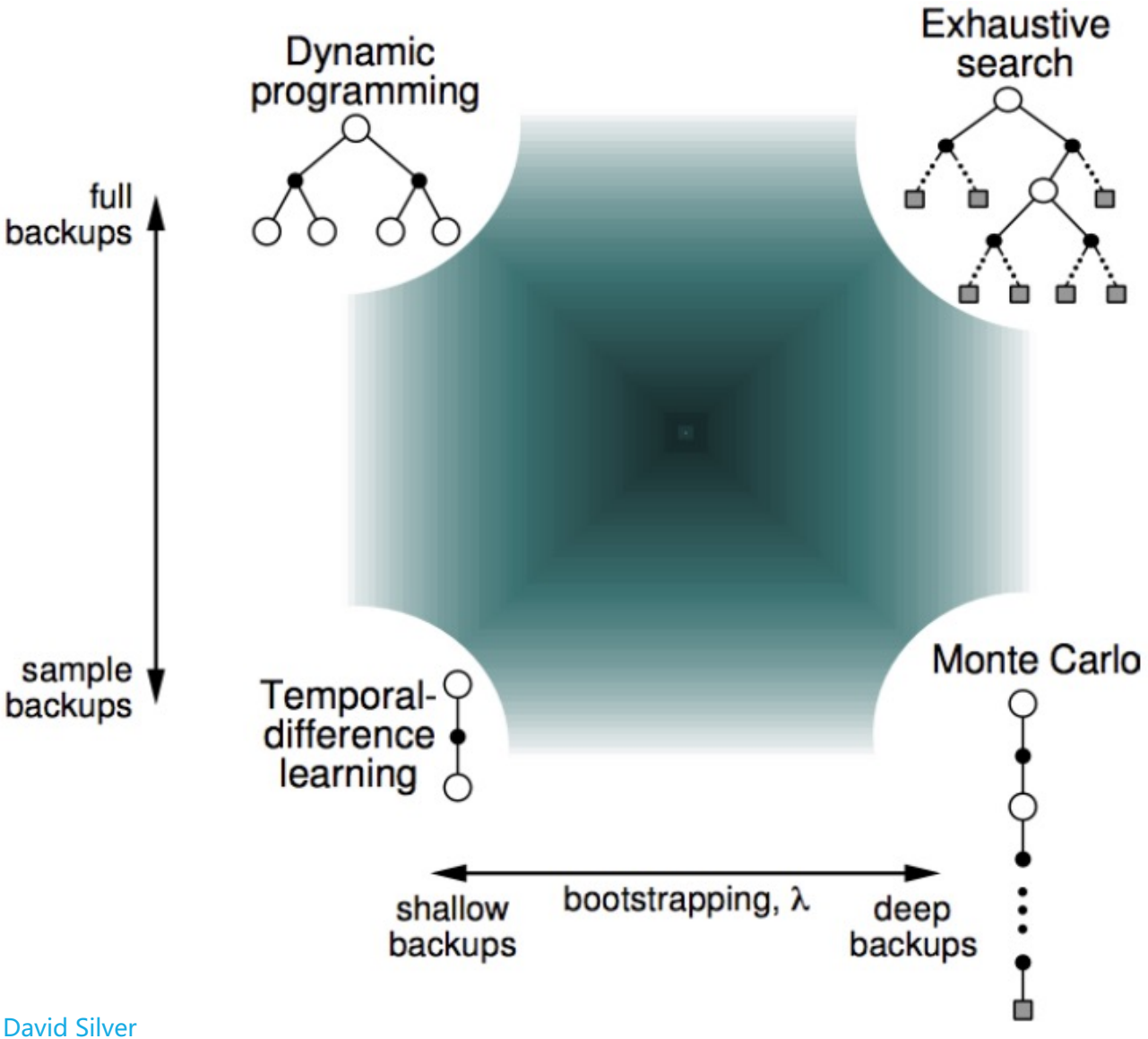
总览强化学习值函数估计多种方法



Slide credit: David Silver

强化学习2022
第4节

涉及知识点：
SARSA、Q学习算法及其收敛性、多步自助法

无模型控制方法

课程大纲

强化学习基础部分

1. 强化学习、探索与利用
2. MDP和动态规划
3. 值函数估计
4. 无模型控制方法
5. 参数化的值函数和策略
6. 规划与学习
7. 深度强化学习价值方法
8. 深度强化学习策略方法

强化学习前沿部分

9. 基于模型的深度强化学习
10. 离线强化学习
11. 模仿学习
12. 参数化动作空间
13. 多智能体强化学习基础
14. 多智能体强化学习前沿
15. 强化学习的应用
16. 技术交流与回顾

从知道什么是好的，到如何做好行动

- 从知道什么是好的：估计$V^{\pi}(S_t)$

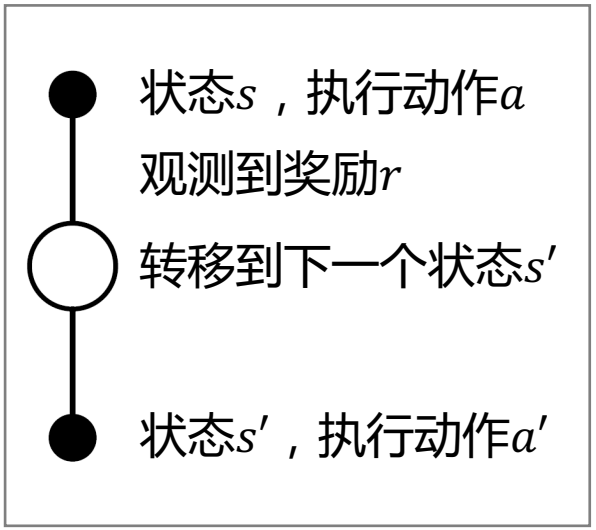
$$V(S_t) \leftarrow V(S_t) + \alpha\big(G_t - V(S_t)\big)$$

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

- 基于V函数，如何选择好的行动？

$$\pi(s) = \arg\max_{a \in A} \sum_{s' \in S} P_{sa}(s')V(s')$$

需要知道环境模型



- 基于Q函数，如何选择好的行动？

$$\pi(s) = \arg\max_{a \in A} Q(s, a)$$

  - 因此，估计Q函数对直接做行动（控制）有直接的作用

SARSA

SARSA

- 对于当前策略执行的每个（状态-动作-奖励-状态-动作）元组

状态s，执行动作a

观测到奖励r

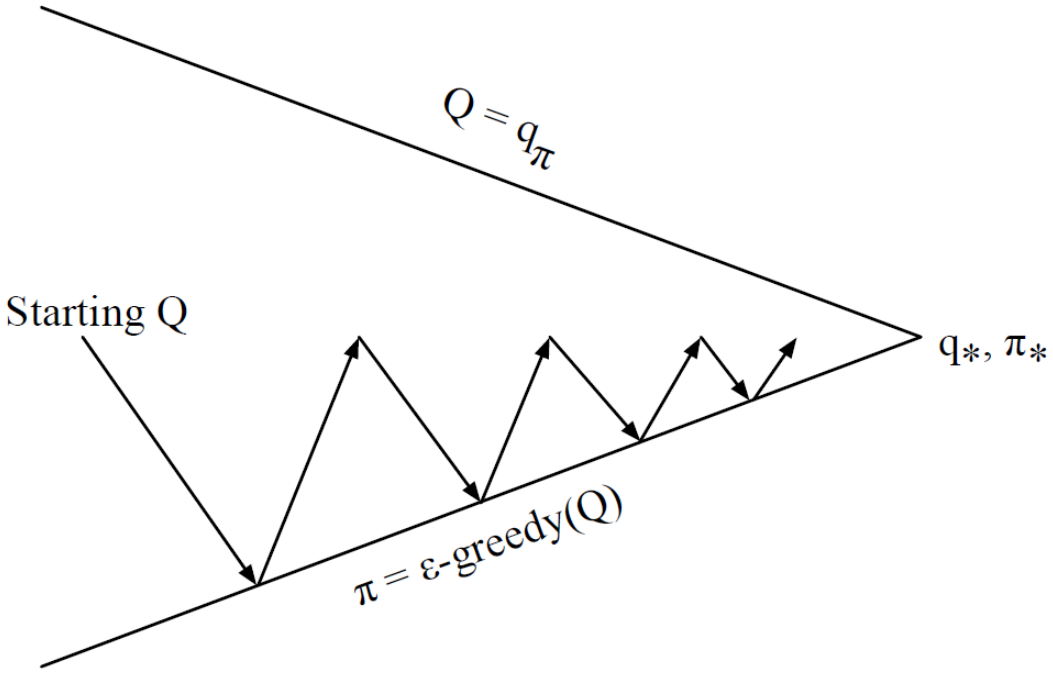
转移到下一个状态s'

状态s'，执行动作a'

- SARSA更新状态-动作值函数为

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma Q(s',a') - Q(s,a))$$

使用SARSA的在线策略控制



- 每个时间步长：
  - 策略评估：SARSA $Q(s,a) \leftarrow Q(s,a) + \alpha\big(r + \gamma Q(s',a') - Q(s,a)\big)$
  - 策略改进：ε-greedy策略改进

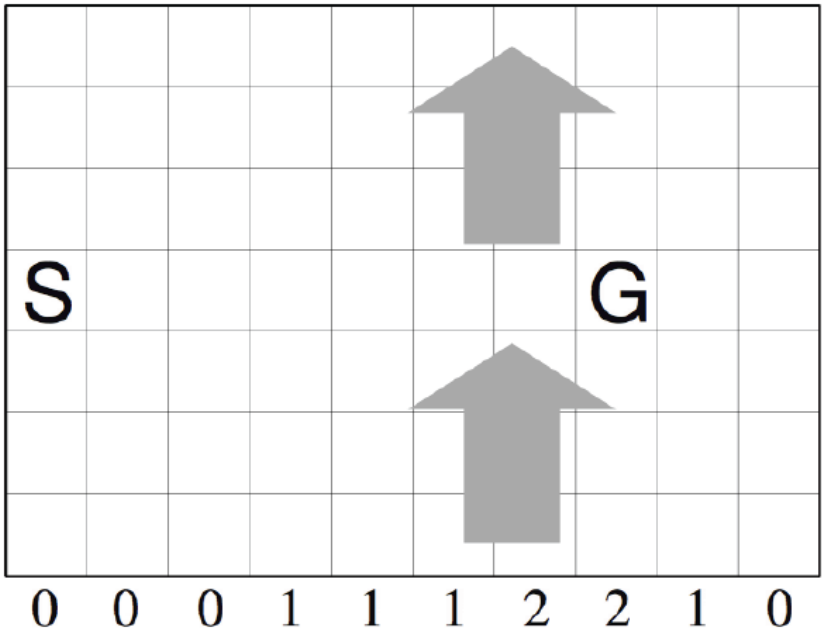
SARSA算法

Sarsa: An on-policy TD control algorithm

Initialize $Q(s,a), \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(terminal\text{-}state, \cdot) = 0$
Repeat (for each episode):
  Initialize S
  Choose A from S using policy derived from Q (e.g., ϵ-greedy)
  Repeat (for each step of episode):
    Take action A, observe R, S'
    Choose A' from S' using policy derived from Q (e.g., ϵ-greedy)
    $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma Q(S',A') - Q(S,A)\big]$
    $S \leftarrow S'$; $A \leftarrow A'$;
  until S is terminal

注：在线策略时序差分控制（on-policy TD control）使用当前策略进行动作采样。即，SARSA算法中的两个“A”都是由当前策略选择的

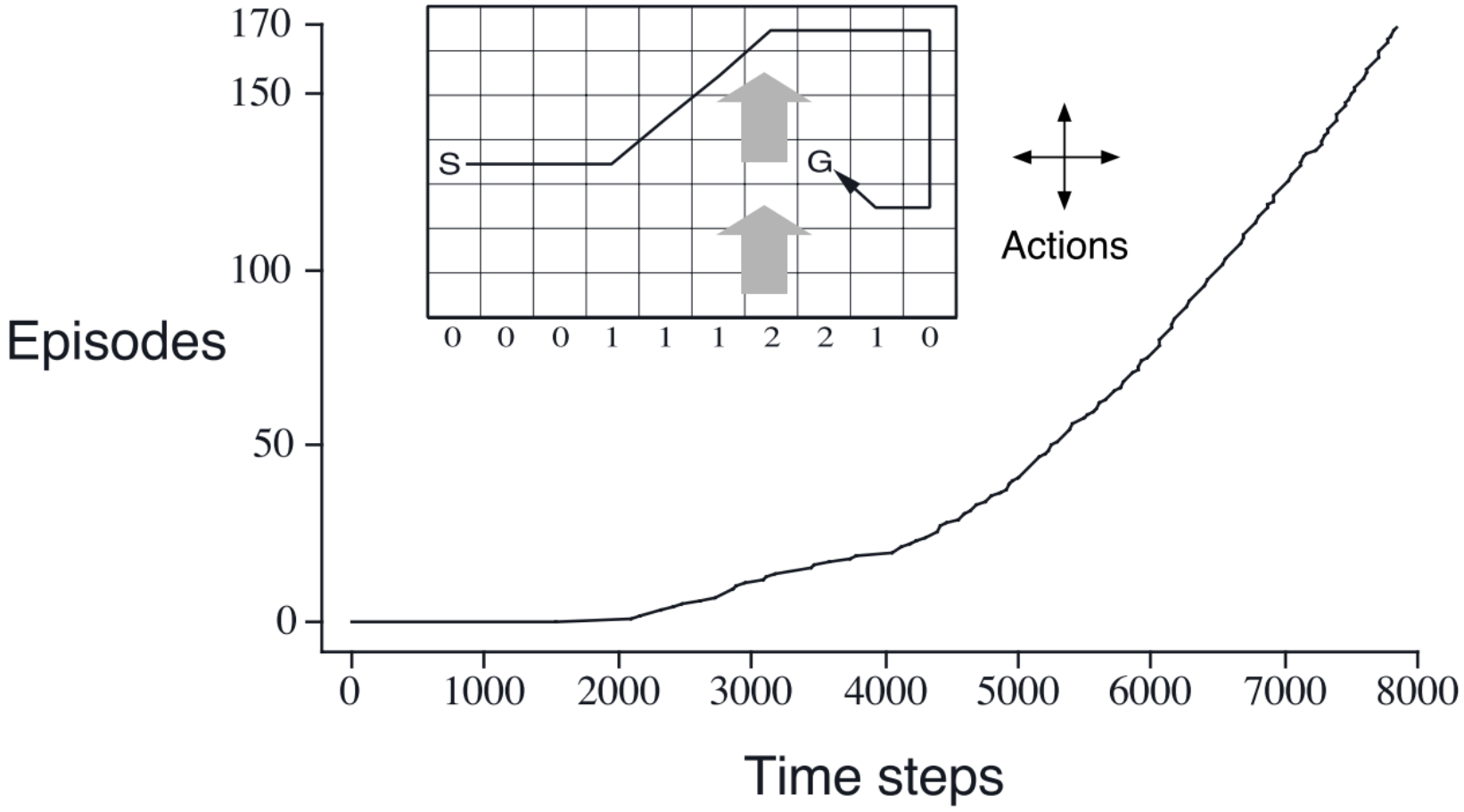
SARSA示例：Windy Gridworld



standard moves

- 每步的奖励 = -1，直到智能体抵达目标网格
- 无折扣因子

SARSA示例：Windy Gridworld



注意：随着训练的进行，SARSA策略越来越快速地抵达目标

Q 学习算法及其收敛性

目录

Contents


01
Q 学习

Q 学习

- 学习状态-动作值函数 $Q(s,a) \in \mathbb{R}$，不直接优化策略
- 一种离线策略（off-policy）学习方法

策略函数，一般是给定的策略，$\mu(\cdot|\, s_t,) \in \mathbb{R}^{|A|}$

动作空间，$a \sim A$

$$Q(s_t,a_t) = \sum_{t=0}^{T} \gamma^t R(s_t,a_t), a_t \sim \mu(s_t)$$

奖励函数，$R(s_t,a_t) \in \mathbb{R}$

迭代式：$Q(s_t,a_t) = R(s_t,a_t) + \gamma Q(s_{t+1},a_{t+1})$

离线策略学习

什么是离线策略学习

- 目标策略 $\pi(a|s)$ 进行值函数评估（ $V^{\pi}(s)$或$Q^{\pi}(s,a)$ ）
- 行为策略 $\mu(a|s)$ 收集数据： $\{s_1, a_1, r_2, s_2, a_2, \dots, S_T\}\sim\mu$

为什么使用离线策略学习

- 平衡探索（ exploration ）和利用（ exploitation ）
- 通过观察人类或其他智能体学习策略
- 重用旧策略所产生的经验
- 遵循探索策略时学习最优策略
- 遵循一个策略时学习多个策略
- Policy Transfer
  - Collective Noise Contrastive Estimation for Policy Transfer Learning. AAAI 2016

Q 学习

- 无需重要性采样（为什么？）
- 根据行为策略选择动作 $a_t \sim \mu(\cdot|s_t)$
- 根据目标策略选择后续动作 $a'_{t+1} \sim \pi(\cdot|s_t)$
  - 目标$Q^{\star}(s_t, a_t) = r_t + \gamma Q(s_{t+1}, a'_{t+1})$
- 更新$Q(s_t, a_t)$的值以逼近目标状态-动作值

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left(r_{t+1} + \gamma Q(s_{t+1}, a'_{t+1}) - Q(s_t, a_t)\right)$$

策略π的动作，而非策略μ

使用Q 学习的离线策略控制

- 允许行为策略和目标策略都进行改进
- 目标策略π是关于$Q(s,a)$的贪心策略

$$\pi(s_{t+1}) = \arg\max_{a'} Q(s_{t+1}, a')$$

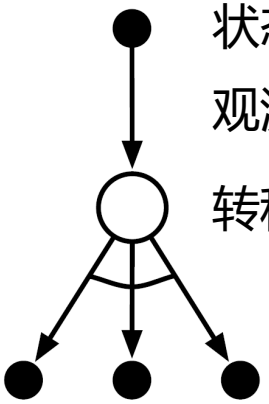
- 行为策略μ是关于$Q(s,a)$的ε-贪心策略
- Q-学习目标函数可以简化为

$$\begin{aligned} r_{t+1} + \gamma Q(s_{t+1}, a'_{t+1}) &= r_{t+1} + \gamma Q(s_{t+1}, \arg\max_{a'_{t+1}} Q(s_{t+1}, a'_{t+1})) \\ &= r_{t+1} + \gamma \max_{a'_{t+1}} Q(s_{t+1}, a'_{t+1}) \end{aligned}$$

- Q-学习更新方式

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_{t+1} + \gamma \max_{a'_{t+1}} Q(s_{t+1}, a'_{t+1}) - Q(s_t, a_t))$$

Q 学习控制算法

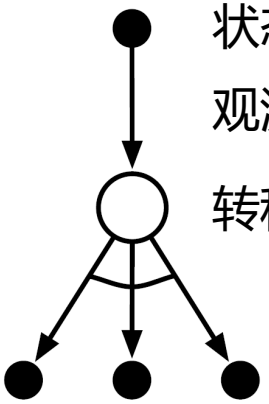


$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_{t+1} + \gamma \max_{a'_{t+1}} Q(s_{t+1}, a'_{t+1}) - Q(s_t, a_t))$$

- 定理：Q-学习控制收敛到最优状态-动作值函数

$$Q(s, a) \rightarrow Q^*(s, a)$$

Q 学习控制算法



$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t))$$

- 为什么不需要重要性采样？
  - 使用了状态-动作值函数而不是使用状态值函数

02

收敛性证明

Q 学习的收敛性

收缩算子（contraction operator）

- $Q(s,a) = r(s,a) + \gamma \max_{a'} Q(s',a')$
- 定义 H 算子：$HQ = r(s,a) + \gamma \mathbb{E}_{s' \sim p(\cdot|s,a)}[\max_{a'} Q(s',a')]$
- 最优值函数 $Q^\star$ 是 H 的不动点，意味着：$Q^\star = HQ^\star$

Melo, Francisco S. "Convergence of Q-learning: A simple proof." *Institute Of Systems and Robotics, Tech. Rep* (2001): 1-4.

Q 学习的收敛性

- 直接从Q函数证明

$$(\mathbf{H}q)(x,a) = \sum_{y\in\mathcal{X}} \mathsf{P}_a(x,y)\left[r(x,a,y) + \gamma \max_{b\in\mathcal{A}} q(y,b)\right]$$

$$\begin{aligned}
&\|\mathbf{H}q_1 - \mathbf{H}q_2\|_\infty \\
&= \max_{x,a} \left| \sum_{y\in\mathcal{X}} \mathsf{P}_a(x,y)\left[r(x,a,y) + \gamma \max_{b\in\mathcal{A}} q_1(y,b) - r(x,a,y) + \gamma \max_{b\in\mathcal{A}} q_2(y,b)\right] \right| \\
&= \max_{x,a} \gamma \left| \sum_{y\in\mathcal{X}} \mathsf{P}_a(x,y)\left[\max_{b\in\mathcal{A}} q_1(y,b) - \max_{b\in\mathcal{A}} q_2(y,b)\right] \right| \\
&\le \max_{x,a} \gamma \sum_{y\in\mathcal{X}} \mathsf{P}_a(x,y) \left| \max_{b\in\mathcal{A}} q_1(y,b) - \max_{b\in\mathcal{A}} q_2(y,b) \right| \\
&\le \max_{x,a} \gamma \sum_{y\in\mathcal{X}} \mathsf{P}_a(x,y) \max_{z,b} |q_1(z,b) - q_2(z,b)| \\
&= \max_{x,a} \gamma \sum_{y\in\mathcal{X}} \mathsf{P}_a(x,y) \|q_1 - q_2\|_\infty \\
&= \gamma \|q_1 - q_2\|_\infty .
\end{aligned}$$

$$\|\mathbf{H}q_1 - \mathbf{H}q_2\|_\infty \le \gamma \|q_1 - q_2\|_\infty$$

Melo, Francisco S. "Convergence of Q-learning: A simple proof." *Institute Of Systems and Robotics, Tech. Rep* (2001): 1-4.

动手学强化学习-Q learning收敛性证明：
https://hrl.boyuai.com/chapter/1/%E6%97%B6%E5%BA%8F%E5%B7%AE%E5%88%86%E7%AE%97%E6%B3%95

柯西数列

数列的柯西收敛准则

数列$\{x_n\}$收敛的充分必要条件是：对于任意给定的正数ε，总存在正整数N，使得当n>N,m>N时有

$$|x_n - x_m| < \varepsilon$$

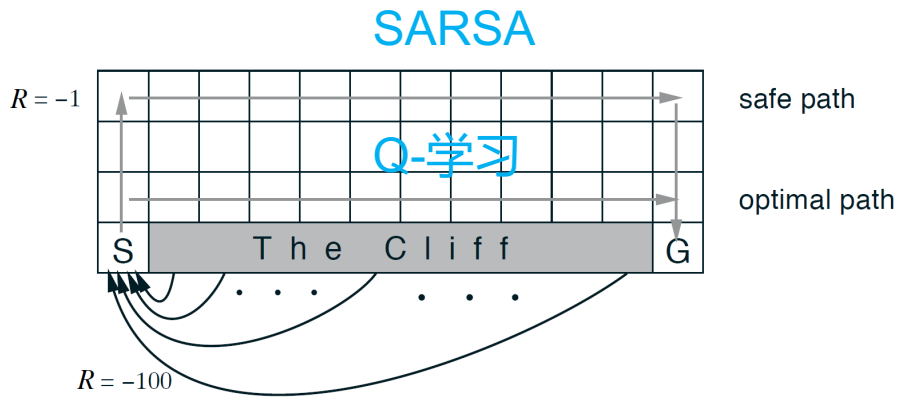
我们把满足该条件的{x_n}称为柯西序列，那么上述定理可表述成：数列{x_n}收敛，当且仅当它是一个柯西序列。

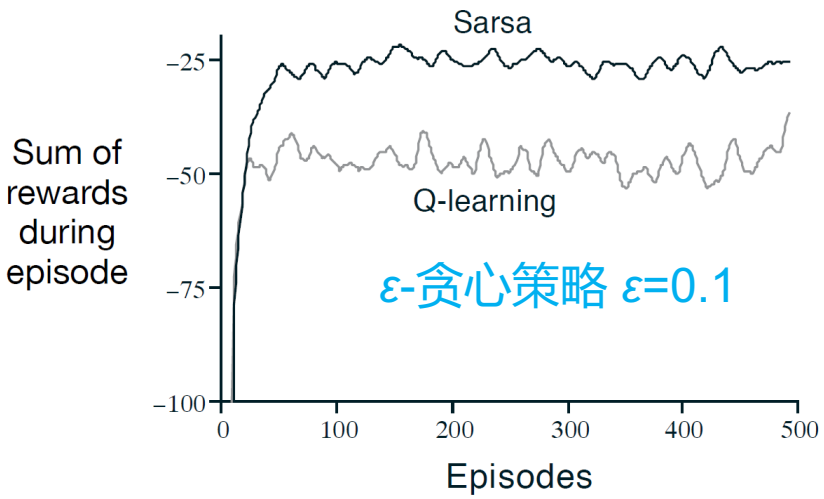
该准则的几何意义表示，数列{x_n}收敛的充分必要条件是：该数列中的元素随着序数的增加而愈发靠近，即足够靠后的任意两项都无限接近。

$$\|\mathbf{H}q_1 - \mathbf{H}q_2\|_\infty \leq \gamma \|q_1 - q_2\|_\infty$$

SARSA与Q 学习对比实验

- 悬崖边行走（Cliff-walking）
  - 无折扣的奖励
  - 片段式的任务
  - 所有移动奖励 = -1
  - 踏入悬崖区域会产生-100奖励并将智能体送回开始处
- 为什么会有图示结果？





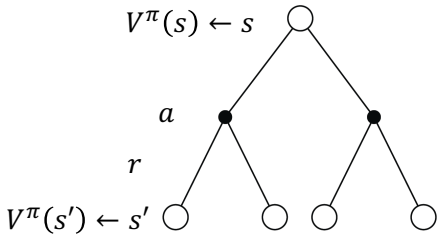
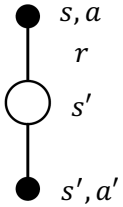
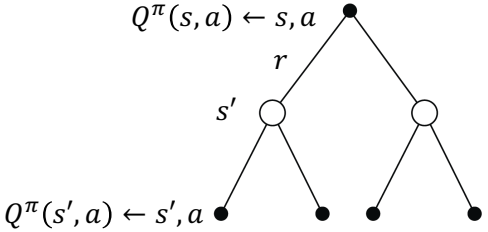
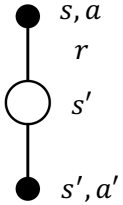
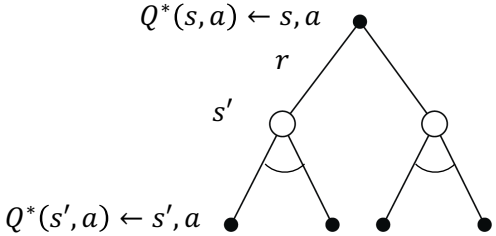
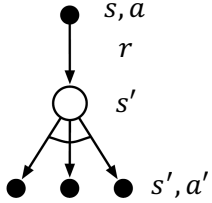
多步自助法

目录

Contents

01
多步时序差分预测

回顾：动态规划（DP）和时序差分（TD）的关系

	完全反向传播(DP)	采样反向传播(TD)
状态值函数的贝尔曼期望方程 $V^{\pi}(s)$	 迭代的策略评估	 时序差分学习
状态-动作值函数的贝尔曼期望方程 $Q^{\pi}(s,a)$	 $Q-$策略迭代	 SARSA
状态-动作值函数的贝尔曼最优方程 $Q^{*}(s,a)$	 $Q-$价值迭代	 $Q-$学习

回顾：动态规划（DP）和时序差分（TD）的关系

完全反向传播(DP)	采样反向传播(TD)
迭代的策略评估 $V(s) \leftarrow \mathbb{E}[r + \gamma V(s')\|s]$	时序差分学习 $V(s) \overset{\alpha}{\leftarrow} r + \gamma V(s')$
Q – 策略迭代 $Q(s,a) \leftarrow \mathbb{E}[r + \gamma Q(s',a')\|s,a]$	SARSA $Q(s,a) \overset{\alpha}{\leftarrow} r + \gamma Q(s',a')$
Q – 价值迭代 $Q(s,a) \leftarrow \mathbb{E}\left[r + \gamma \max_{a'} Q(s',a') \middle\| s,a\right]$	Q – 学习 $Q(s,a) \overset{\alpha}{\leftarrow} r + \gamma \max_{a'} Q(s',a')$

其中 $x \overset{\alpha}{\leftarrow} y \equiv x \leftarrow x + \alpha(y - x)$

回顾：蒙特卡洛方法&时序差分法

蒙特卡洛方法

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

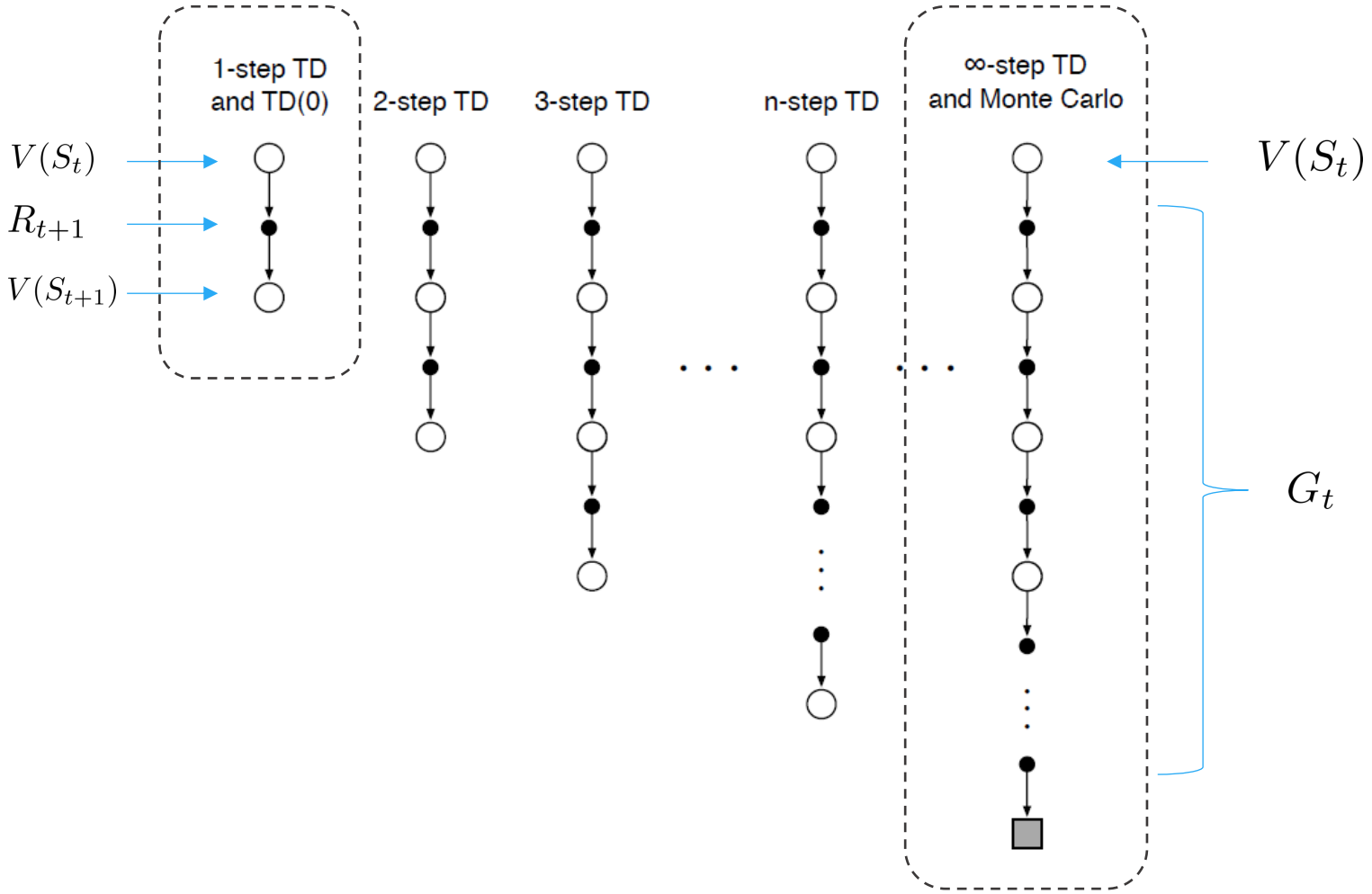
$$G_t = R_{t+1} + \gamma R_{t_2} + \gamma^2 R_{t+3} + ... + \gamma^{T-t-1} R_T$$

时序差分法

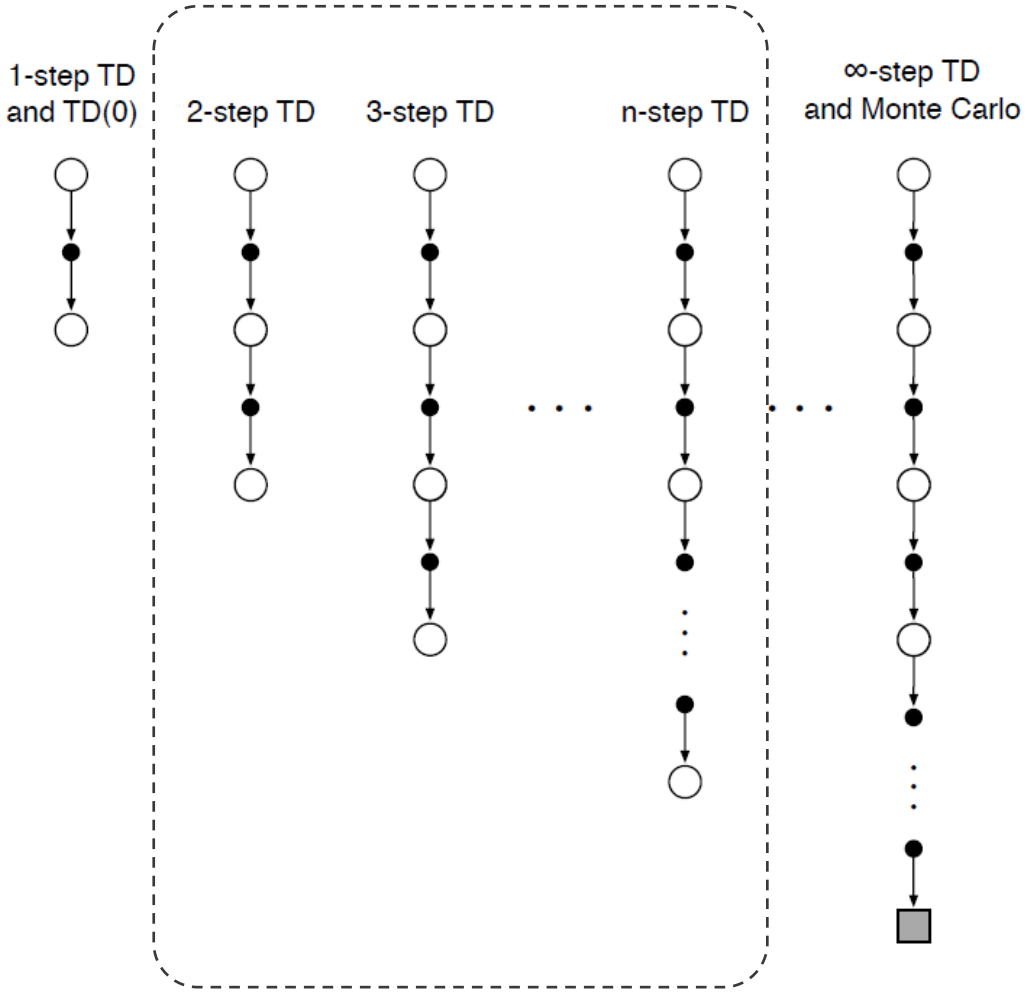
$$V(S_t) \leftarrow V(S_t) + \alpha(R_{t+1} + \gamma V(S_{t+1}) - V(S_t))$$

有没有方法介于两者之间？

多步时序差分预测


1-step TD
and TD(0)
2-step TD
3-step TD
n-step TD
∞-step TD
and Monte Carlo
$V(S_t)$
R_{t+1}
$V(S_{t+1})$
$V(S_t)$
G_t

多步时序差分预测


1-step TD
and TD(0)
2-step TD
3-step TD
n-step TD
∞-step TD
and Monte Carlo

n 步累计奖励

- 考虑下列 n 步累计奖励，$n = 1,2,\ldots,\infty$

$n = 1$	（TD）	$G_t^{(1)} = R_{t+1} + \gamma V(S_{t+1})$
$n = 2$		$G_t^{(2)} = R_{t+1} + \gamma R_{t+2} + \gamma^2 V(S_{t+2})$
⋮		⋮
$n = \infty$	（MC）	$G_t^{(\infty)} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{T-1} R_T$

- 定义 n 步累计奖励

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

- n 步时序差分学习

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

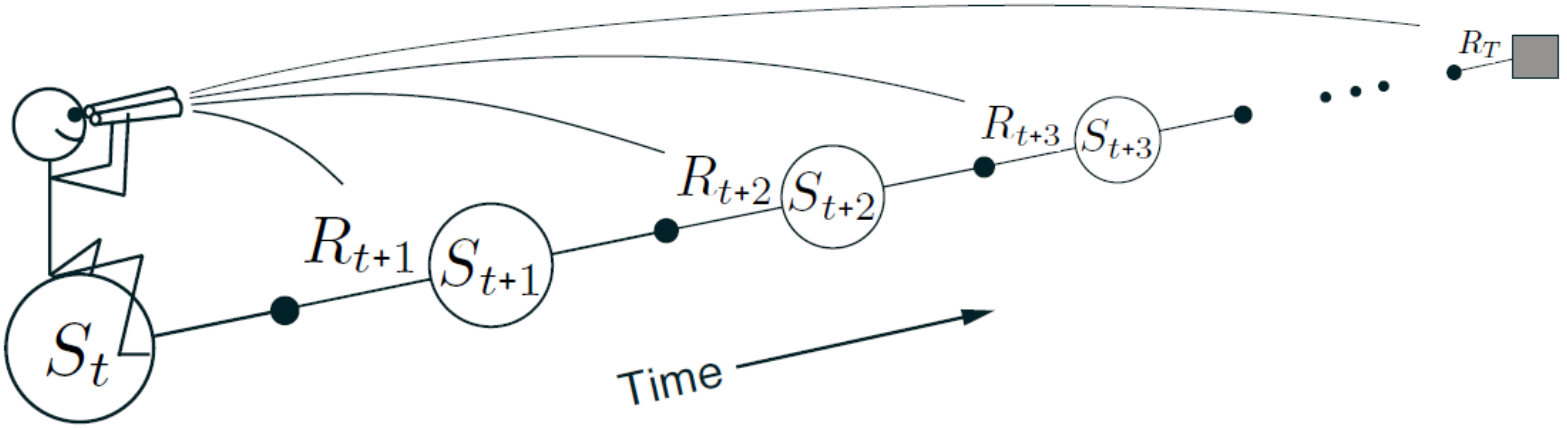
n 步累计奖励

- 定义 n 步累计奖励

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

- n 步时序差分学习

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



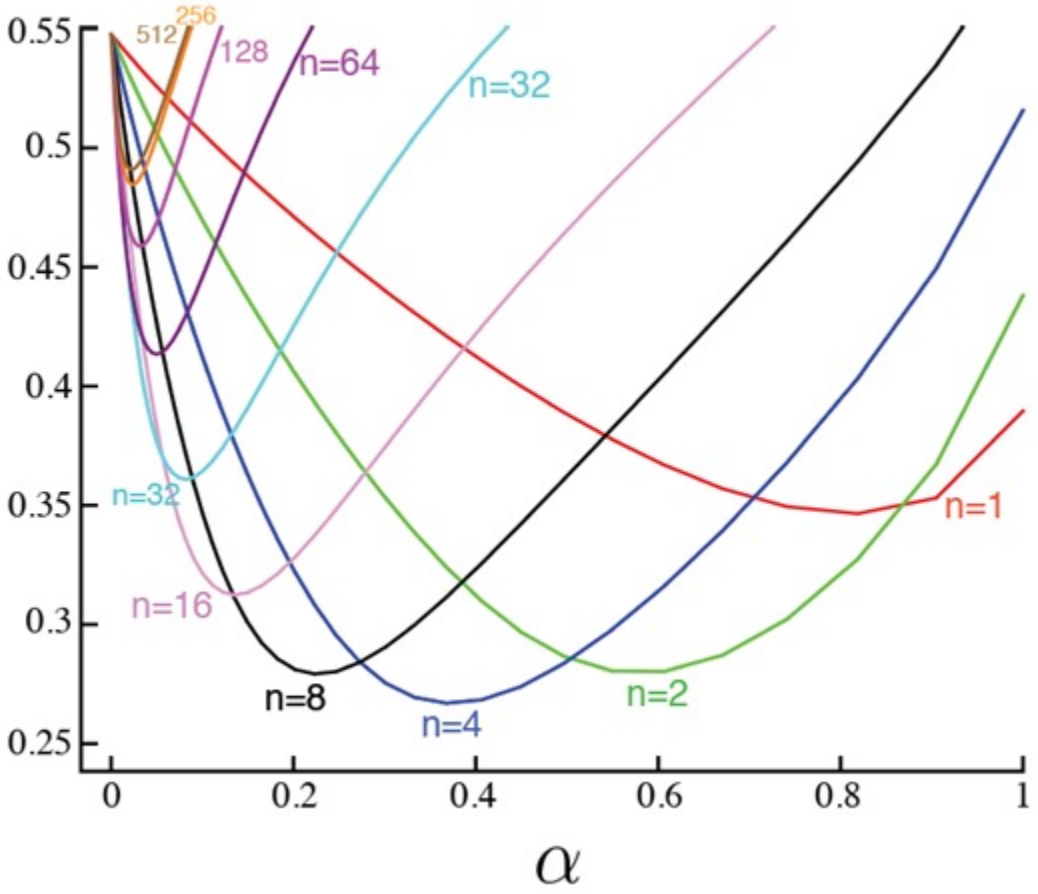
随机游走的例子里使用 n 步时序差分

随机游走

- 每个片段都从中间状态 C 开始。
- 每一时刻都有均等的概率向左走或者向右走。
- 到最左侧或者最右侧时，片段结束。
- 如果走到最右侧，得到的奖励为1；如果走到最左侧，得到的奖励为0。

多步时序差分法的表现

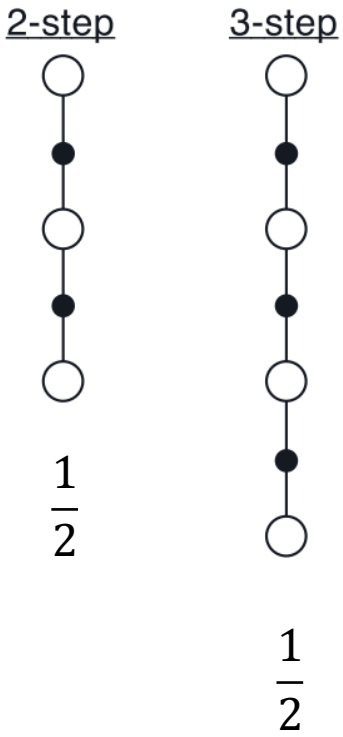
拥有19个状态的随机游走游戏中，在10个片段(episode)结束时的RMS误差



平均 n 步累计奖励

- 我们可以进一步对不同 n 下的 n 步累计奖励求平均值
- 例如，求 2 步和 3 步时的平均累计奖励

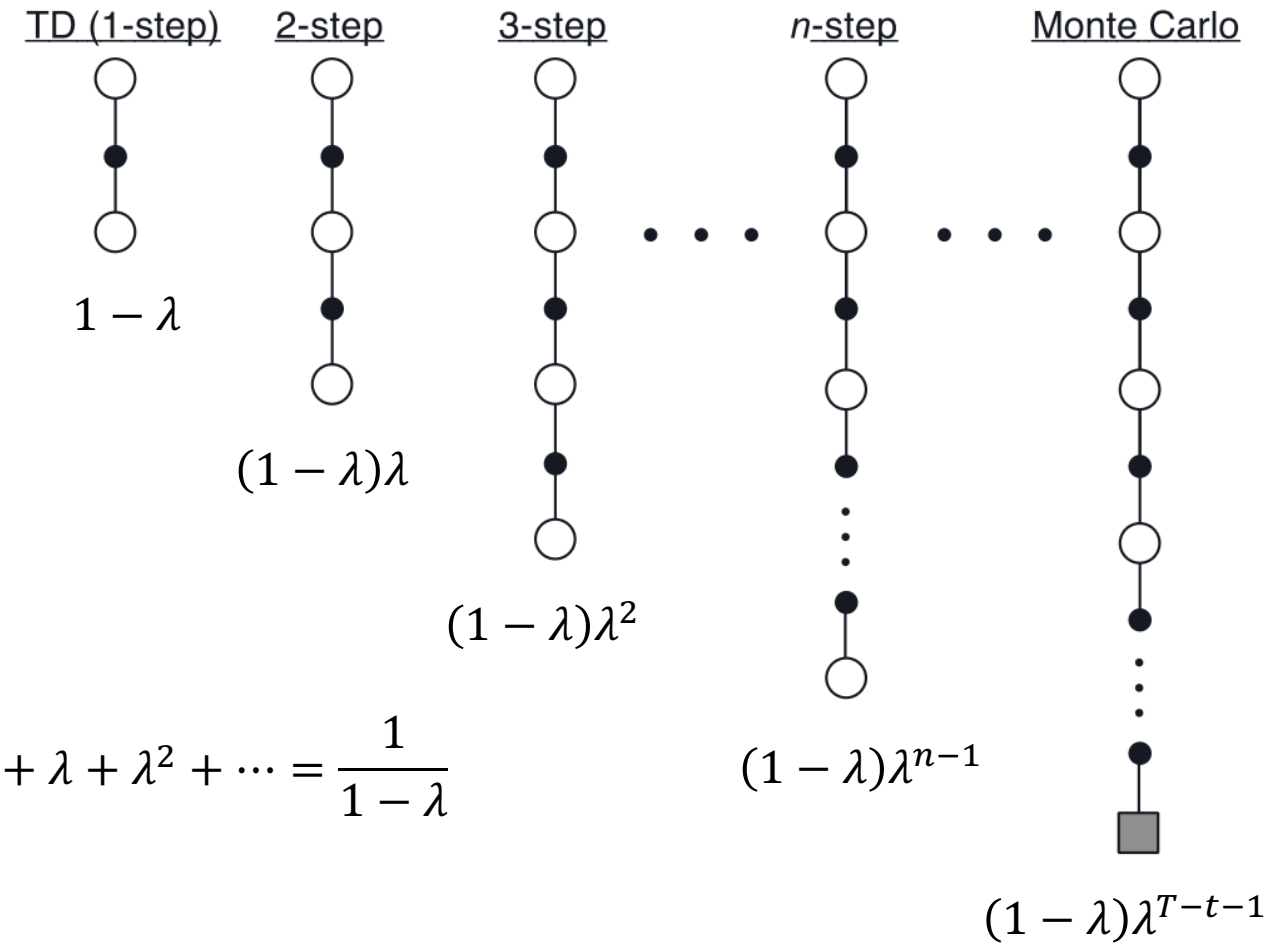
$$\frac{1}{2}G^{(2)} + \frac{1}{2}G^{(3)}$$



- 上式结合两种不同时间步长的信息
- 我们是否能够结合所有不同时间步长的信息呢？

使用平均 n 步累计奖励的 TD(λ) 算法

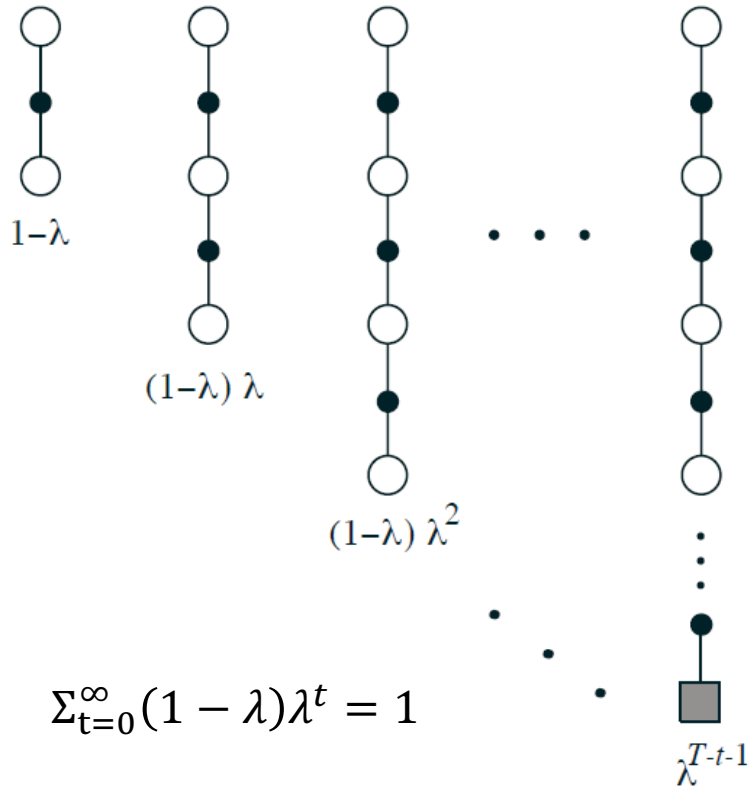
$\mathrm{TD}(\lambda), \lambda$ −累计奖励



$$1+\lambda+\lambda^2+\cdots=\frac{1}{1-\lambda}$$

使用平均 n 步累计奖励的 TD(λ) 算法

TD(λ), λ – 累计奖励



$\Sigma_{t=0}^{\infty}(1-\lambda)\lambda^t = 1$

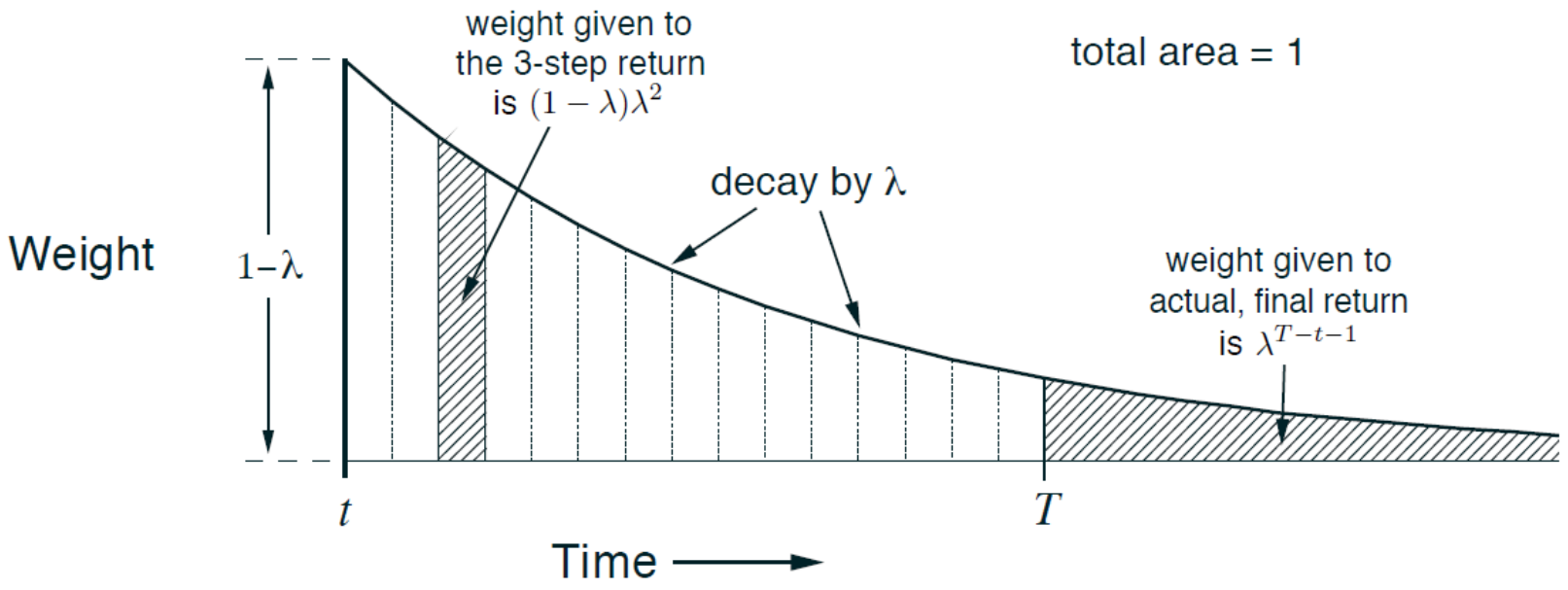
- λ – 累计奖励 G_t^{λ} 结合了所有 n 步累计奖励 $G_t^{(n)}$
- 使用权重 $(1-\lambda)\lambda^{n-1}$

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

- TD(λ)

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

使用平均 n 步累计奖励的 TD(λ) 算法

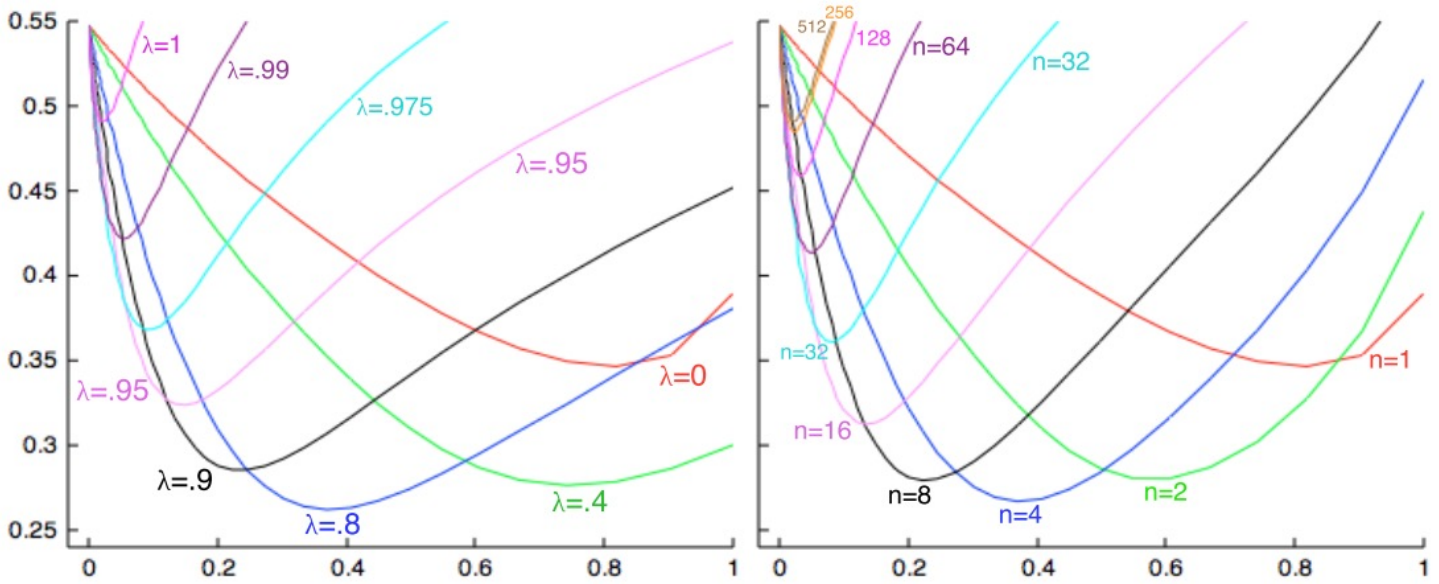


$$G_t^{\lambda} = (1-\lambda)\sum_{n=1}^{T-t-1}\lambda^{n-1}G_t^{(n)} + \lambda^{T-t-1}G_t$$

- 当 $\lambda = 1$ 时，$G_t^{\lambda} = G_t$，相当于蒙特卡洛方法
- 当 $\lambda = 0$ 时，$G_t^{\lambda} = G_t^{(1)}$，相当于单步时序差分方法

TD(λ) vs. n 步时序差分

前10个片段(episode)结束时的Offline RMS误差



19个状态的随机游走实验结果

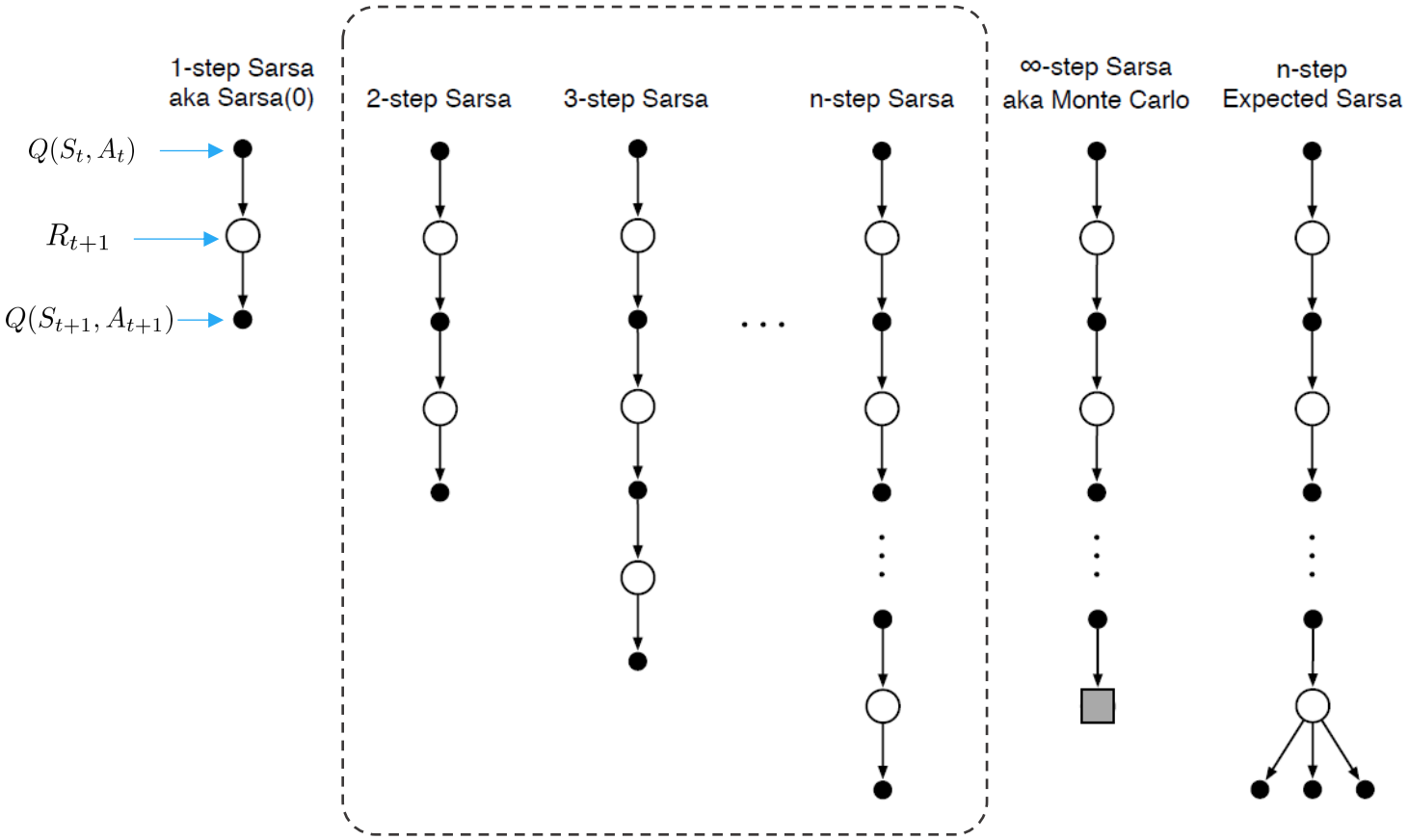
- 在使用最佳的 α 与 λ 值时，离线 λ – 累计奖励算法具有稍好一些的实验结果

02
多步Sarsa

多步Sarsa

- n 步的思想是否只能用在预测上？我们能不能把这种思想放在控制上？

将这种思想用在 Sarsa 上， 我们得到了 n 步 Sarsa算法



多步Sarsa

- 类似与 n 步TD，我们有：

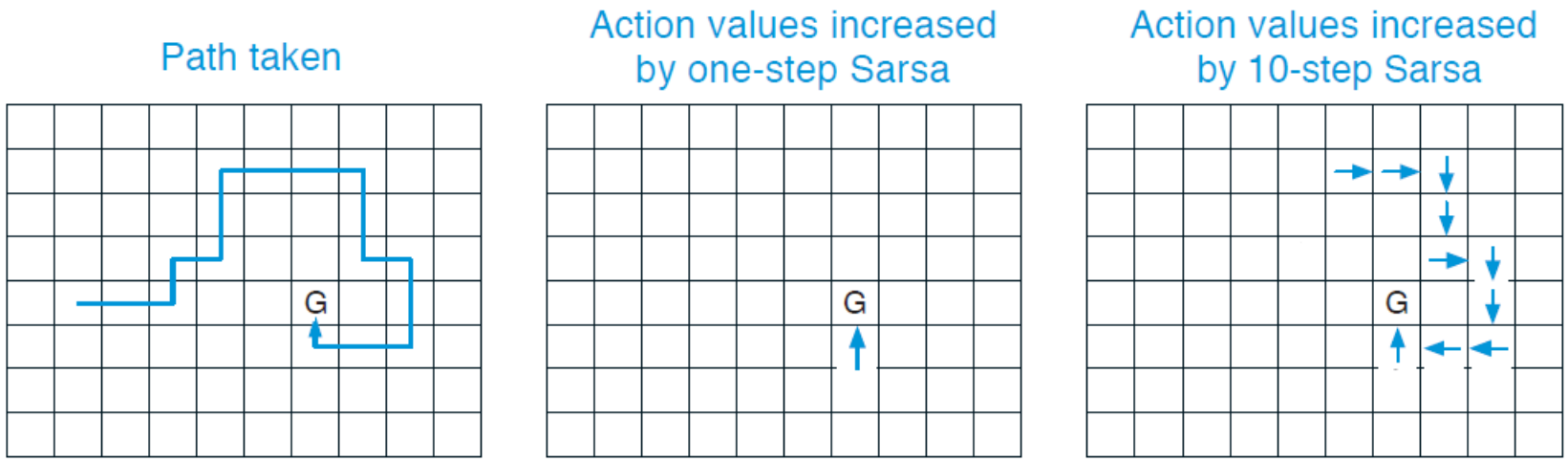
$$G_{t:t+n} \doteq R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^{n} Q_{t+n-1}\left(S_{t+n}, A_{t+n}\right)$$

$$where, n \geq 1, 0 \leq t < T - n$$

$$Q_{t+n}\left(S_t, A_t\right) \doteq Q_{t+n-1}\left(S_t, A_t\right) + \alpha\left[G_{t:t+n} - Q_{t+n-1}\left(S_t, A_t\right)\right]$$

$$where, 0 \leq t < T$$

多步Sarsa的例子


Path taken
G
Action values increased
by one-step Sarsa
G
Action values increased
by 10-step Sarsa
G

03

使用重要性采样的多步离线学习法

回顾：采用重要性采样的离线学习

- 在离线学习的场景下，往往会有两个策略：
  - π: 通常是对于目前的动作值估计函数的一个贪婪策略（也有可能是一个随机策略）
  - b: 用来收集数据的，更有探索性的一个策略（常用 ϵ-greedy 来实现）

- 由于收集数据的策略和我们要实际优化的策略并不是同一个策略，我们需要加上一项修正（以值函数为例）：

$$\Delta V' = \frac{\pi(a_t|s_t)}{b(a_t|s_t)} \Delta V$$

回顾：采用重要性采样的离线学习

- 在重要性采样的角度对值函数的推导：

$$\mathbb{E}_{s\sim\pi}(f(s)) = \int p(s|\pi)f(s)ds = \int \frac{p(s|\pi)}{p(s|b)}p(s|b)f(s)ds = \mathbb{E}_{s\sim b}[\frac{p(s|\pi)}{p(s|b)}f(s)]$$

$$Pr(A_t, S_{t+1}, A_{t+1}, ..., S_h|S_t, A_{t:h} \sim b) = \Pi_{k=t}^{h-1} b(A_k|S_k)p(S_{k+1}|S_k, A_k)$$

$$Pr(A_t, S_{t+1}, A_{t+1}, ..., S_h|S_t, A_{t:h} \sim \pi) = \Pi_{k=t}^{h-1} \pi(A_k|S_k)p(S_{k+1}|S_k, A_k)$$

$$G_{t:t+n} = R_{t+1} + \gamma R_{t+2} + ... + \gamma^{n-1} R_{t+n} + \gamma^n V_{t+n-1}(S_{t+n})$$

$$\rho_{t:h} \doteq \prod_{k=t}^{\min(h,T-1)} \frac{\pi(A_k|S_k)}{b(A_k|S_k)}$$

$$V_{t+n}(S_t) \doteq V_{t+n-1}(S_t) + \alpha[\rho_{t:t+n-1} G_{t:t+n} - V_{t+n-1}(S_t)]$$

使用重要性采样的多步离线学习法

- 对于状态值函数 V：

$$V_{t+n}(S_t) \doteq V_{t+n-1}(S_t) + \alpha[\rho_{t:t+n-1} G_{t:t+n} - V_{t+n-1}(S_t)]$$

$$\rho_{t:h} \doteq \prod_{k=t}^{\min(h,T-1)} \frac{\pi(A_k|S_k)}{b(A_k|S_k)}$$

- 对于动作值函数 Q：

$$Q_{t+n}(S_t, A_t) \doteq Q_{t+n-1}(S_t, A_t) + \alpha[\rho_{t:t+n-1} G_{t:t+n} - Q_{t+n-1}(S_t, A_t)]$$

$$\rho_{t:h} \doteq \prod_{k=t}^{\min(h,T-1)} \frac{\pi(A_k|S_k)}{b(A_k|S_k)}$$

使用重要性采样的多步离线学习法

Algorithm 1: 离线n步Sarsa算法

Input: 任意探索性的策略b, Q函数, 目标策略π, 常数n

Output: 估计的Q函数, $Q \approx q_\pi$

初始化;

for 对每个*episode* **do**

  for $t = 0, 1, 2, \ldots$ **do**

    if $t < T$ **then**

      Take A_t, get (R_{t+1}, S_{t+1});

      if S_{t+1}是终止状态 **then**

        $T \leftarrow t + 1$;

      else

        根据策略$b(\cdot|S_{t+1})$选择A_{t+1}并保存;

    $\tau \leftarrow t - n + 1$ **if** $\tau \geq 0$ **then**

      $\rho \leftarrow \Pi_{i=\tau+1}^{\min(\tau+n-1, T-1)} \frac{\pi(A_i|S_i)}{b(A_i|S_i)}$;

      $G \leftarrow \Sigma_{i=\tau+1}^{\min(\tau+n, T)} \gamma^{i-\tau-1} R_i$;

      if $\tau + n < T$ **then**

        $G \leftarrow G + \gamma^n Q(S_{\tau+n}, A_{\tau+n})$;

      $Q(S_\tau, A_\tau) \leftarrow Q(S_\tau, A_\tau) + \alpha[\rho G - Q(S_\tau, A_\tau)]$

      更新π，确保其对Q具有贪婪性质;

Sarsa 算法框架

计算重要性采样系数

计算 n 步目标

更新 Q 函数

04

多步树回溯算法

多步树回溯算法 (N-step Tree Backup Algorithm)

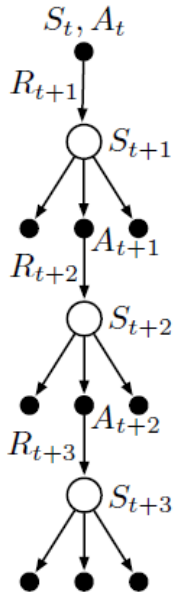
- 是否存在一种不需使用重要性采样的离线算法？
  - Q学习和期望 Sarsa (expected Sarsa)

- 我们可以将这种形式扩展到 n 步，称之为多步树回溯算法。
- 以 $n = 3$ 为例子：

$\pi(a|S_{t+1})$

$\pi(A_{t+1}|S_{t+1})\,\pi(a'|S_{t+2})$

$\pi(A_{t+1}|S_{t+1})\,\pi(A_{t+2}|S_{t+2})\,\pi(a''|S_{t+3})$



the 3-step tree-backup update

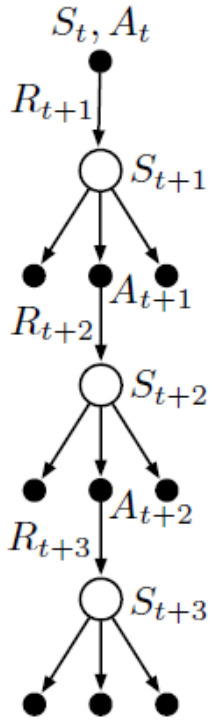
多步树回溯算法 (N-step Tree Backup Algorithm)

- 以 $n = 3$ 为例子：

$\pi(a|S_{t+1})$

$\pi\left(A_{t+1}|S_{t+1}\right)\pi\left(a'|S_{t+2}\right)$

$\pi\left(A_{t+1}|S_{t+1}\right)\pi\left(A_{t+2}|S_{t+2}\right)\pi\left(a''|S_{t+3}\right)$



the 3-step tree-backup update

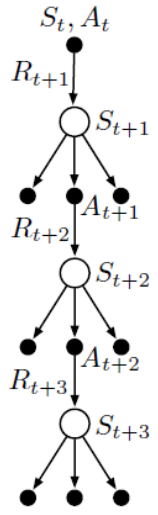
多步树回溯算法推导

- 对于 $n=1$：

$$G_{t:t+1} \doteq R_{t+1} + \gamma \sum_a \pi\left(a|S_{t+1}\right) Q_t\left(S_{t+1}, a\right)$$

- 对于 $n=2$：

$$\begin{aligned} G_{t:t+2} \doteq & R_{t+1} + \gamma \sum_{a \neq A_{t+1}} \pi\left(a|S_{t+1}\right) Q_{t+1}\left(S_{t+1}, a\right) \\ & + \gamma \pi\left(A_{t+1}|S_{t+1}\right)\left(R_{t+2} + \gamma \sum_a \pi\left(a|S_{t+2}\right) Q_{t+1}\left(S_{t+2}, a\right)\right) \\ = & R_{t+1} + \gamma \sum_{a \neq A_{t+1}} \pi\left(a|S_{t+1}\right) Q_{t+1}\left(S_{t+1}, a\right) + \gamma \pi\left(A_{t+1}|S_{t+1}\right) G_{t+1:t+2} \end{aligned}$$



the 3-step tree-backup update

- 推广到 n 步：

$$G_{t:t+n} \doteq R_{t+1} + \gamma \sum_{a \neq A_{t+1}} \pi\left(a|S_{t+1}\right) Q_{t+n-1}\left(S_{t+1}, a\right) + \gamma \pi\left(A_{t+1}|S_{t+1}\right) G_{t+1:t+n}$$

$$Q_{t+n}\left(S_t, A_t\right) \doteq Q_{t+n-1}\left(S_t, A_t\right) + \alpha\left[G_{t:t+n} - Q_{t+n-1}\left(S_t, A_t\right)\right]$$

模型无关强化学习总结

- 模型无关强化学习在黑盒环境下使用
- 蒙特卡洛方法通过采样直接估计价值函数
- 时序差分学习通过下一步的价值估计来更新当前一步的价值估计
- 模型无关强化学习的on-policy和off-policy算法区别
  - SARSA、Q学习
- 多步自助法可看作介于蒙特卡洛方法和单步时序差分方法之间的一种方法
- 讨论：模型无关强化学习和基于模型的强化学习方法之间的对比

THANK YOU