强化学习基本原理及编程实现06：提升学习效率的方法

郭宪

2019.11.3

人工智能学院
College of Artificial Intelligence

Nankai University

提升学习效率的方法

1. N 步时间差分方法

2. 资格迹方法-- TD(λ)

3. off-policy

第一部分：N 步时间差分方法（第七章）

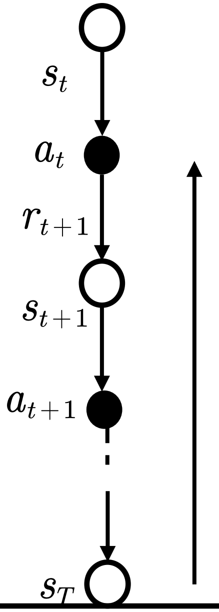
值函数估计过程

表格型值函数估计

Backup 值：

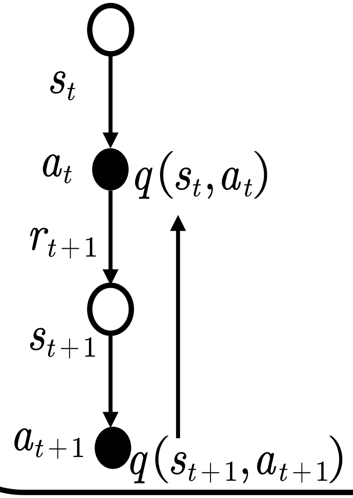
DP $Q_\pi(s,a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \sum_{a' \in A} \pi(a'|s')\, Q_\pi(s',a')$

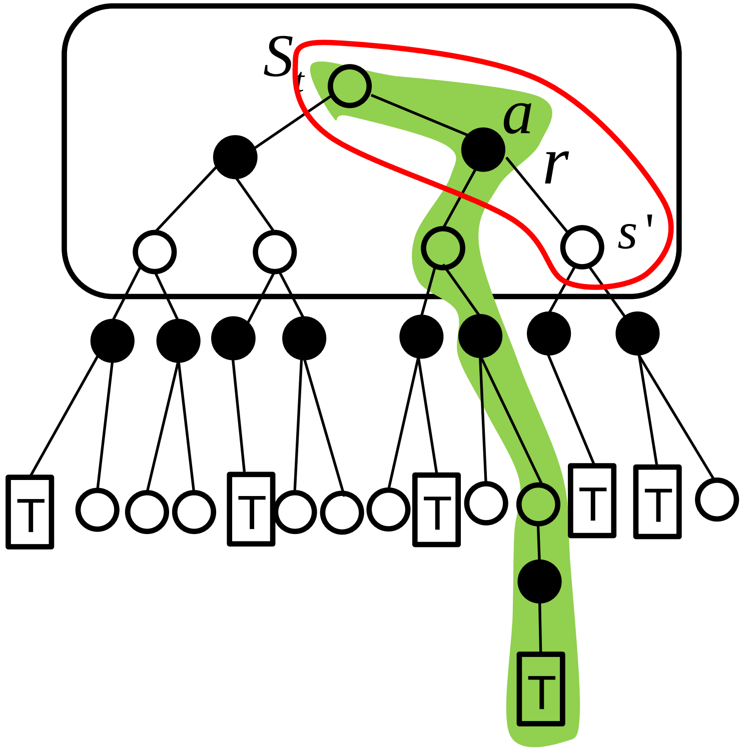
MC的回溯值为： $\sum_{k=1}^{N} \gamma^{k-1} r_{t+k}$



TD回溯值为：

$r_{t+1} + \gamma Q(s_{t+1}, a_{t+1})$



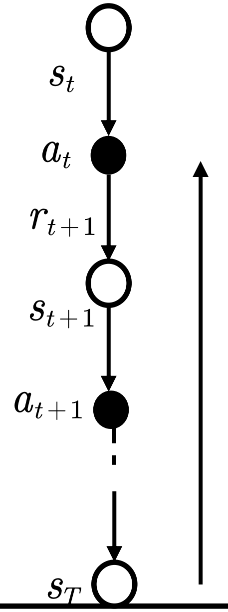


值函数估计过程

Backup 值：

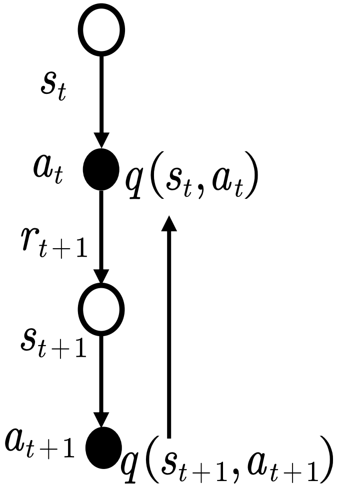
DP $Q_\pi(s,a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \sum_{a' \in A} \pi(a'|s')\, Q_\pi(s',a')$

MC的回溯值为： $\sum_{k=1}^{N} \gamma^{k-1} r_{t+k}$

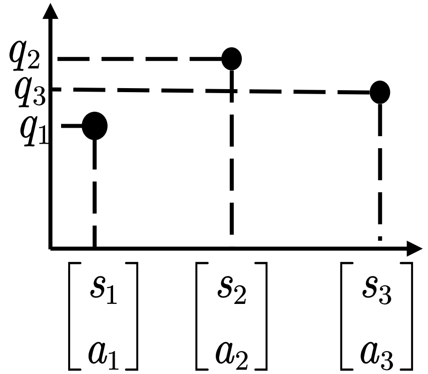


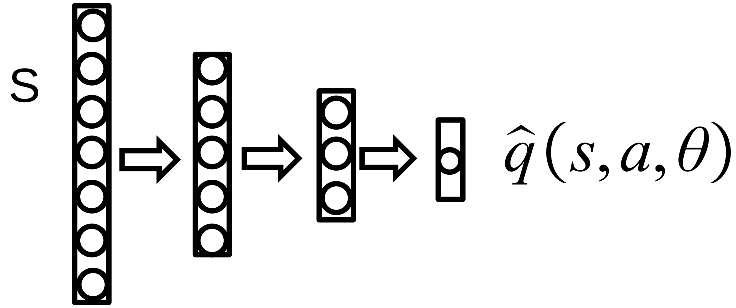
TD回溯值为：

$r_{t+1} + \gamma Q(s_{t+1}, a_{t+1})$



函数逼近： $\hat{q}(s,a,\theta)$





训练目标： $\arg\min_{\theta} \in \left(q(s,a) - \hat{q}(s,a,\theta)\right)^2$

强化学习：在线学习

值函数的评估

增量式MC方法估计值函数：

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

$\Rightarrow$ G_t 是值函数 $v_\pi(S_t)$ 的无偏估计

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

最简单的时间差分学习算法：TD(0)

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

$\Rightarrow$ 真实的TD目标 $R_{t+1} + \gamma v_\pi(S_{t+1})$ 是无偏估计，但 $R_{t+1} + \gamma V(S_{t+1})$ 是有偏估计

$R_{t+1} + \gamma V(S_{t+1})$ 称为TD目标

TD目标 $R_{t+1} + \gamma v_\pi(S_{t+1})$ 的方差比MC的返回值 G_t 要小很多。因为MC的返回值依赖于很多随机动作，转移概率和回报。TD目标仅依赖于一个随机动作，转移概率和回报。

N步预测

值函数估计：

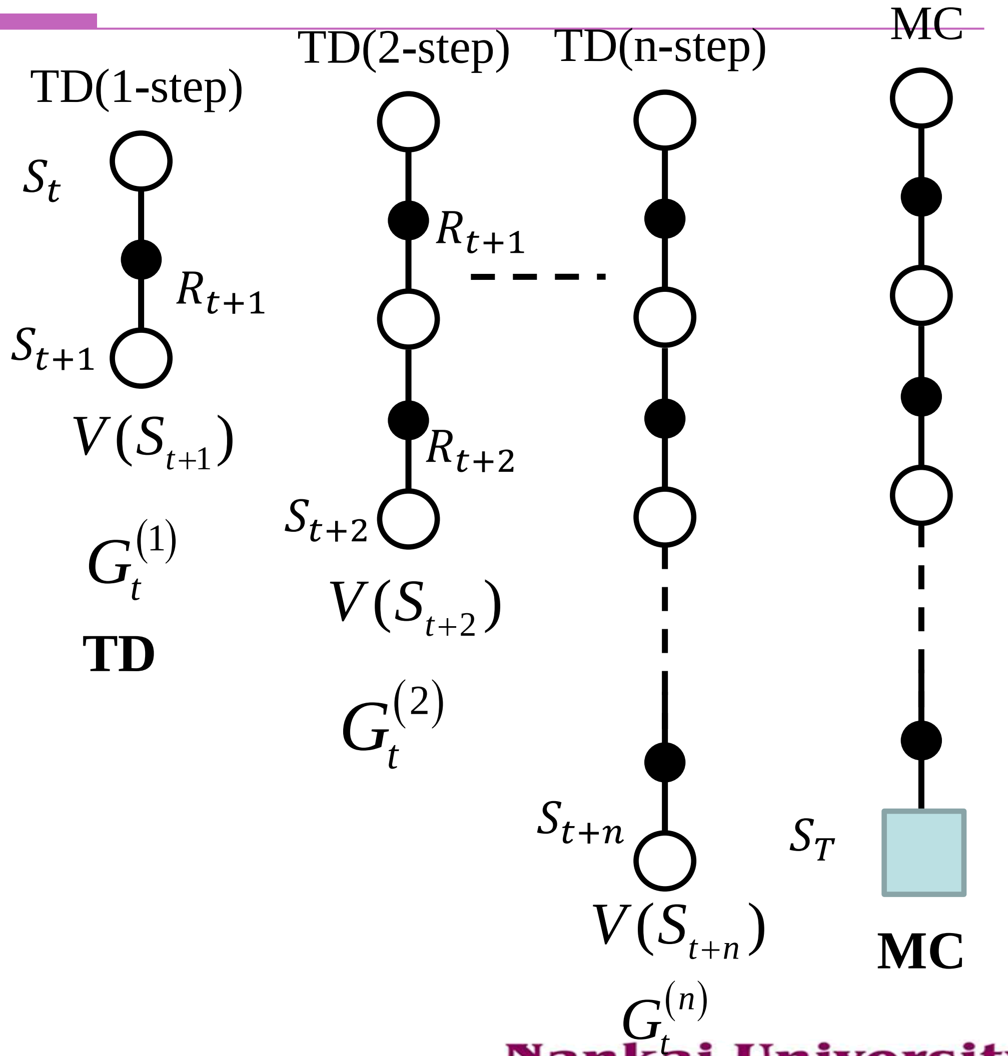
$$V(S_t) \leftarrow V(S_t) + \alpha\left(G_t - V(S_t)\right)$$

$$G_t^{(1)} = R_{t+1} + \gamma V\left(S_{t+1}\right)$$

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

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

$$\text{MC}: G_t^{(mc)} = R_{t+1} + \gamma R_{t+2} + \cdots$$

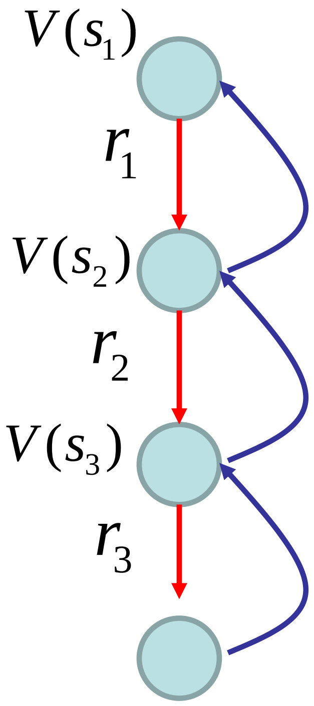


N步时间差分算法

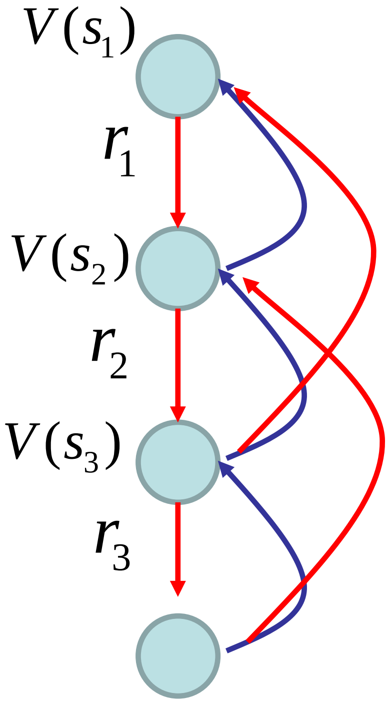
n-step TD for estimating $V \approx v_\pi$

Input: a policy π
Algorithm parameters: step size $\alpha \in (0,1]$, a positive integer n
Initialize $V(s)$ arbitrarily, for all $s \in \mathcal{S}$
All store and access operations (for S_t and R_t) can take their index mod $n+1$

Loop for each episode:
  Initialize and store $S_0 \neq$ terminal
  $T \leftarrow \infty$
  Loop for $t = 0, 1, 2, \ldots$:
  | If $t < T$, then:
  | Take an action according to $\pi(\cdot|S_t)$
  | Observe and store the next reward as R_{t+1} and the next state as S_{t+1}
  | If S_{t+1} is terminal, then $T \leftarrow t+1$
  | $\tau \leftarrow t - n + 1$ (τ is the time whose state's estimate is being updated)
  | If $\tau \geq 0$:
  | $G \leftarrow \sum_{i=\tau+1}^{\min(\tau+n,T)} \gamma^{i-\tau-1} R_i$
  | If $\tau + n < T$, then: $G \leftarrow G + \gamma^n V(S_{\tau+n})$ $(G_{\tau:\tau+n})$
  | $V(S_\tau) \leftarrow V(S_\tau) + \alpha\,[G - V(S_\tau)]$
  Until $\tau = T - 1$

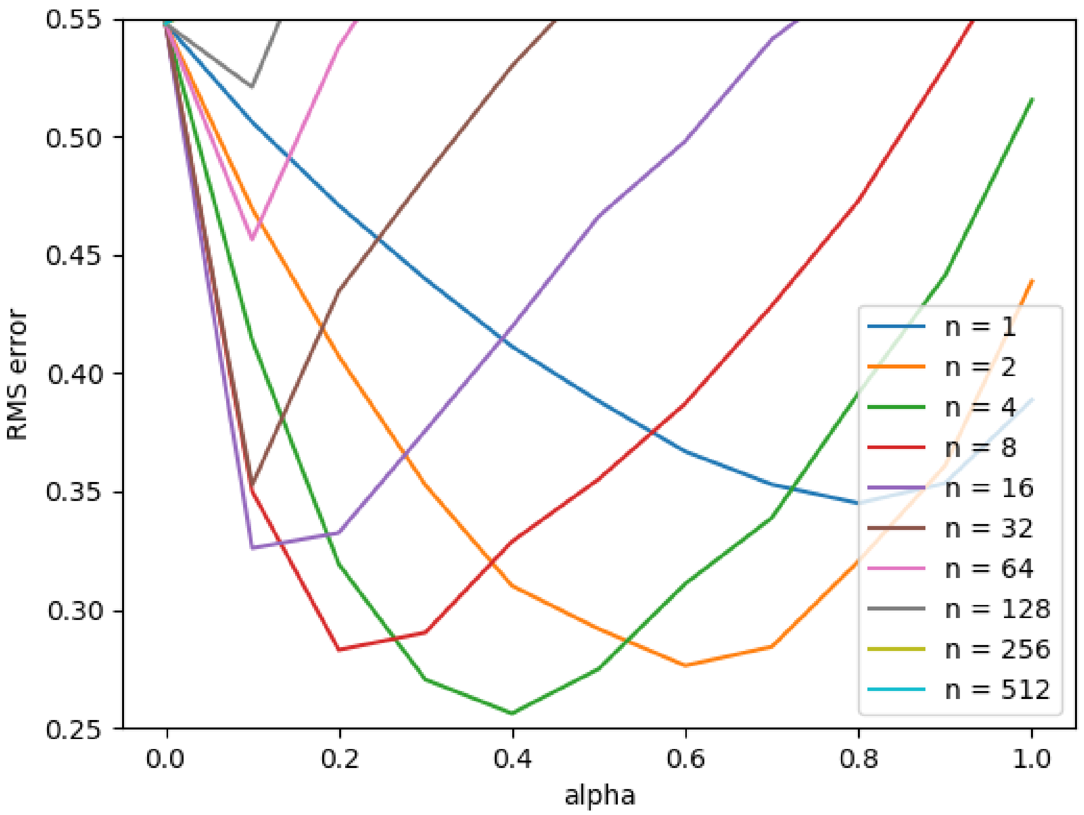


单步预测



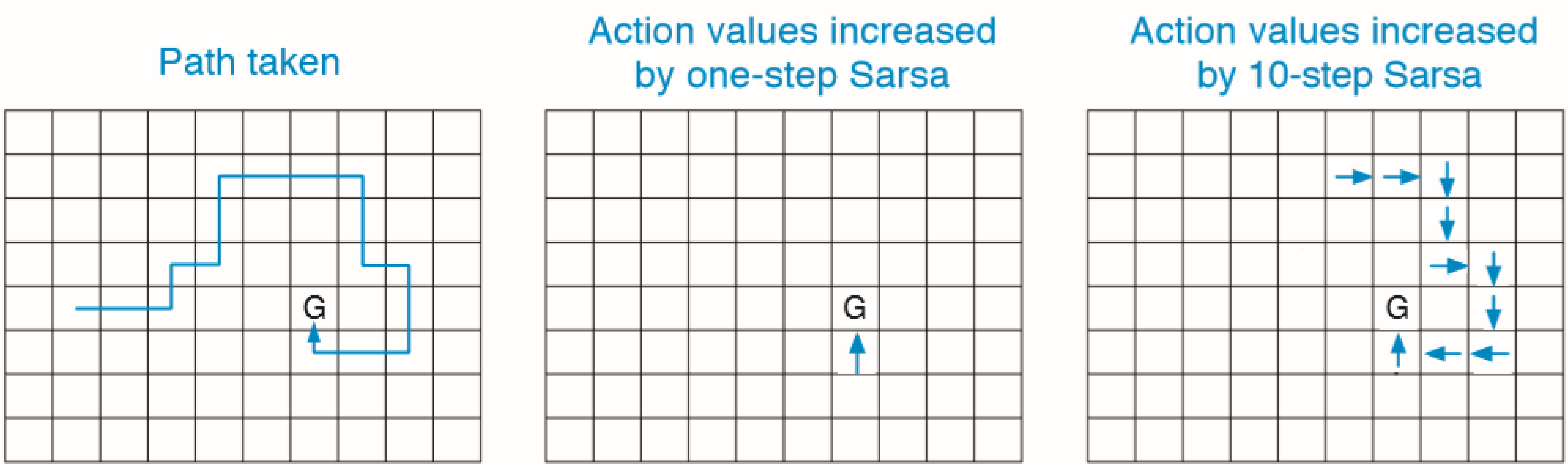
两步预测

N步时间差分算法



0.55
0.50
0.45
0.40
0.35
0.30
0.25
RMS error
0.0
0.2
0.4
0.6
0.8
1.0
alpha
n = 1
n = 2
n = 4
n = 8
n = 16
n = 32
n = 64
n = 128
n = 256
n = 512

N步Sarsa



N步Sarsa比单步Sarsa要快，多步Sarsa能学到更多的知识

第二部分：资格迹方法-- TD(λ)

N步预测

值函数估计：

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

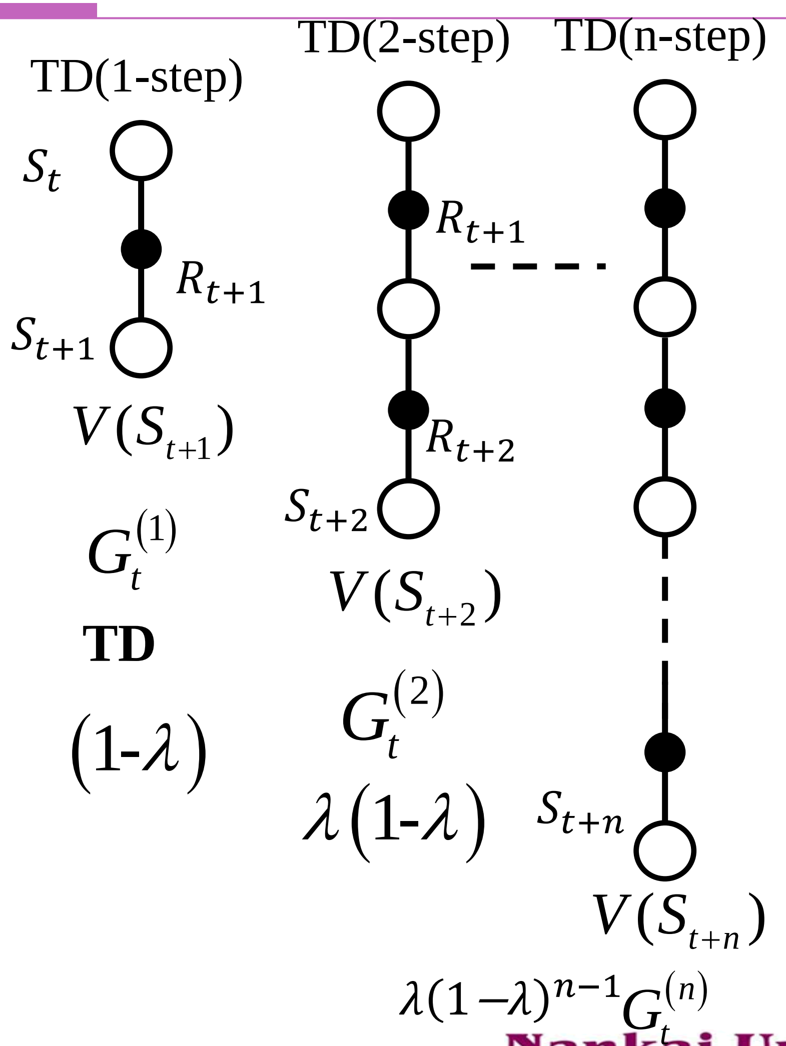
$$G_t^{(1)} = R_{t+1} + \gamma V(S_{t+1})$$

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

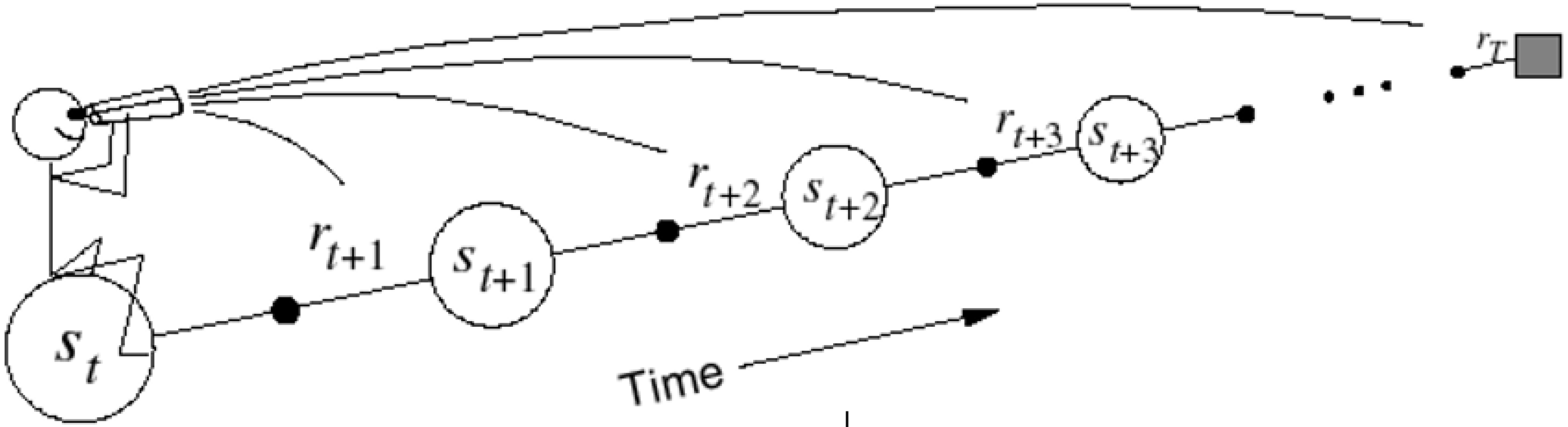
$$G_t^{(n)} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n V(S_{t+n})$$

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

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



$TD(\lambda)$ 前向视角

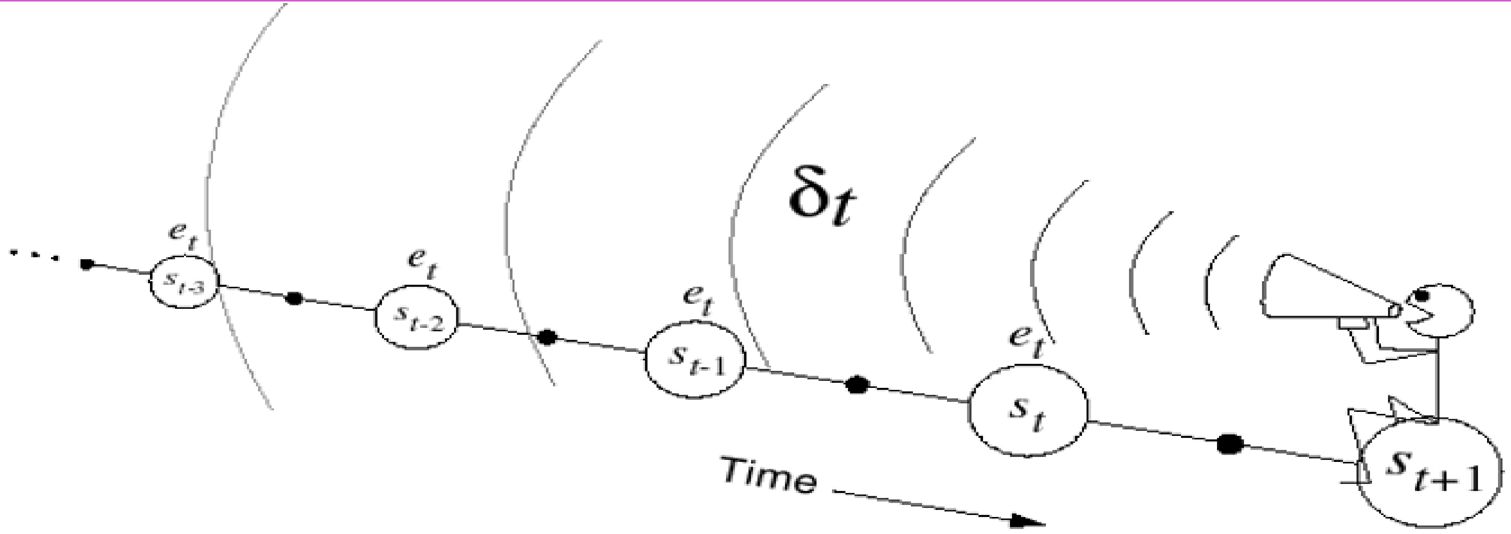


$$V(S_t) \leftarrow V(S_t) + \alpha\left(G_t^{(\lambda)} - V(S_t)\right)$$

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

为了得到当前值函数的估计需要将来的回报和值函数，因此像蒙特卡罗方法一样，只有整个实验结束后，才能计算得到。

$TD(\lambda)$ 后向视角



当前的TD偏差： $\delta_t = R_{t+1} + \gamma V(S_{t+1}) - V(S_t)$

对于每个状态 s ，值函数的改变量为：$V(s) \leftarrow V(s) + \alpha\delta_t \boxed{E_t(s)}$ 适合度轨迹

适合度轨迹定义： $E_t(s) = \begin{cases} \gamma\lambda E_{t-1}(s), & if\ s \neq s_t \\ \gamma\lambda E_{t-1}(s)+1, & if\ s = s_t \end{cases}$

资格迹的理解：表格型

访问状态流:

s_0													

资格迹:

1	0	0	0	0	0	0	0	0	0	0	0	0	0

访问状态流:

s_0	s_1												

资格迹:

$\lambda\gamma$	1	0	0	0	0	0	0	0	0	0	0	0	0

$TD(\lambda)$, $TD(0)$, $TD(1)$

当λ=0， 只有当前状态值更新：

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

当λ=1，状态 s 值函数的总更新与MC等价

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

对于一般的 λ 前向视角偏差等价于后向视角偏差

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

$Sarsa(\lambda)$算法

1. 初始化 $Q(s,a), \forall s \in S, a \in A(s)$,给定参数$\alpha$，$\gamma$
2. Repeat:

行动策略和评估策略都是ε贪婪策略

给定起始状态 s，并根据ε贪婪策略在状态 s 选择动作 a，对所有的 $s \in S, a \in A(s), E(s,a)=0$

Repeat(对于一幕的每一步)

(a) 根据 ε 贪婪策略在状态 s 选择动作 a，得到回报 r 和下一个状态s’，在状态 s’ 根据 ε 贪婪策略得到动作a’

(b) $\delta \leftarrow r+\gamma Q(s',a')-Q(s,a)$，$E(s,a) \leftarrow E(s,a)+1$

(c) 对所有的 $s \in S, a \in A(s)$: $Q(s,a) \leftarrow Q(s,a)+\alpha\delta E(s,a)$，$E(s,a) \leftarrow \gamma\lambda E(s,a)$

(d) s=s’，a=a’

Until s 是终止状态

Until 所有的$Q(s,a)$收敛

3. 输出最终策略：$\pi(s)=\arg\max_a Q(s,a)$

资格迹的理解：函数逼近型

资格迹为与权重维数相同的向量，为短期记忆，持续的时间少于一幕的长度，其作用是辅助整个学习过程，具体过程为：

$$z_{-1} \doteq \mathbf{0}$$

$$z_t \doteq \gamma\lambda z_{t-1} + \nabla \hat{v}(S_t, w_t)$$

时间差分误差为：

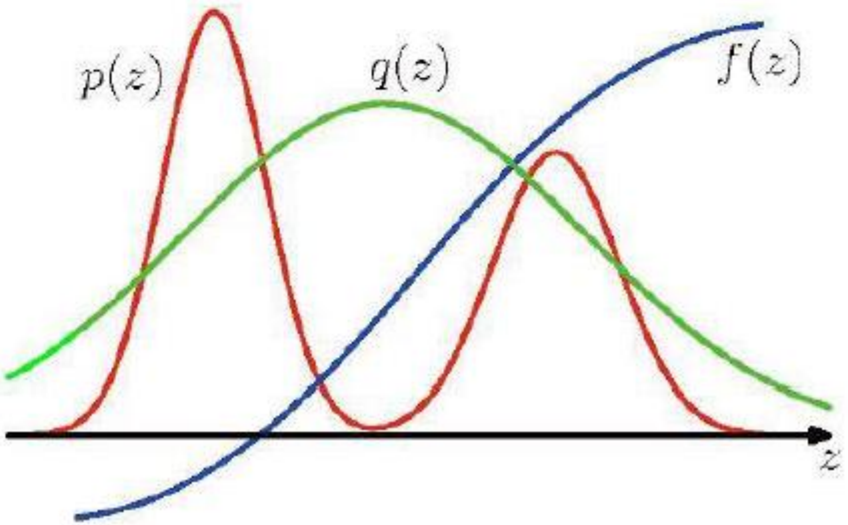
$$\delta_t \doteq R_{t+1} + \gamma\hat{v}(S_{t+1}, w_t) - \hat{v}(S_t, w_t)$$

权重更新方法为：

$$w_{t+1} \doteq w_t + \alpha\delta_t z_t$$

第三部分： off-policy

重要性采样



重要性采样

重要性采样求积分：

$$
\begin{aligned}
E[f] &= \int f(z)p(z)dz \\
&= \int f(z)\frac{p(z)}{q(z)}q(z)dz \\
&\approx \frac{1}{N}\sum_n \frac{p(z^n)}{q(z^n)} f(z^n),\ z^n \sim q(z)
\end{aligned}
$$

定义重要性权重： $\omega^n = p(z^n)/q(z^n)$

普通的重要性采样求积分： $E[f] = \frac{1}{N}\sum_n \omega^n f(z^n)$

重要性采样积分：无偏估计，但方差无穷大

减小方差的方法：加权重要性采样求积分

$$E[f] \approx \sum_{n=1}^{N} \frac{\omega^n}{\sum_{m=1}^{N}\omega^m} f(z^n)$$

MC 重要性采样

在策略 π 下，t 时刻后轨迹的概率为：

$$\Pr\left(A_t, S_{t+1}, \cdots, S_T\right) = \prod_{k=t}^{T-1} \pi\left(A_k \mid S_k\right) p\left(S_{k+1} \mid S_k, A_k\right)$$

在目标策略和行为策略下，每个回报都使用概率进行加权

$$\rho_t^T = \frac{\prod_{k=t}^{T-1} \pi\left(A_k \mid S_k\right) p\left(S_{k+1} \mid S_k, A_k\right)}{\prod_{k=t}^{T-1} \mu\left(A_k \mid S_k\right) p\left(S_{k+1} \mid S_k, A_k\right)} = \prod_{k=t}^{T-1} \frac{\pi\left(A_k \mid S_k\right)}{\mu\left(A_k \mid S_k\right)}$$

普通重要性采样，值估计：

从t到T(t)的返回值

时间t后的第一次终止时刻

$$V(s) = \frac{\sum_{t \in \mathcal{T}(s)} \rho_t^{T(t)} G_t}{\left|\mathcal{T}(s)\right|}$$

状态s被访问过的所有时刻的集合

s s

t = 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19

$$\mathcal{T}(s) = \{4, 15\} \quad T(4) = 7, T(15) = 19$$

加权重要性采样，值估计：

$$V(s) = \frac{\sum_{t \in \mathcal{T}(s)} \rho_t^{T(t)} G_t}{\sum_{t \in \mathcal{T}(s)} \rho_t^{T(t)}}$$

Off-policy every visit MC

初始化，对于所有的

$s \in S, a \in A(s)$:

$Q(s,a) \leftarrow$ 任意

$C(s,a) \leftarrow 0$

$\pi(s) \leftarrow$ 相对于Q的贪婪策略

Repeat forever:

利用软策略μ产生一次实验：

$S_0, A_0, R_1, \cdots, S_{T-1}, A_{T-1}, R_T, S_T$

$G \leftarrow 0$

$W \leftarrow 1$

$For\ t = T-1, T-2, \cdots downto\ 0:$

$G \leftarrow \gamma G + R_{t+1}$

$C(S_t, A_t) \leftarrow C(S_t, A_t) + W$

策略评估

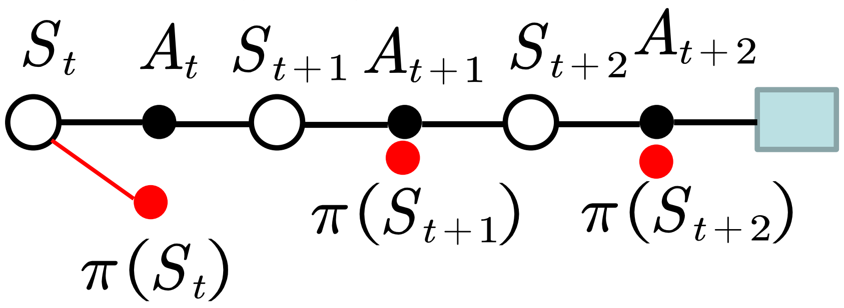
$$\boxed{Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)}\left[G - Q(S_t, A_t)\right]}$$

$$\boxed{\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)}$$

策略改善

如果$A_t \neq \pi(S_t)$则退出for循环

$$W \leftarrow W \frac{1}{\mu(A_t | S_t)}$$



Off-policy n-step Sarsa (7.3节）

Off-policy n-step Sarsa for estimating $Q \approx q_*$ or q_π

Input: an arbitrary behavior policy b such that $b(a|s) > 0$, for all $s \in \mathcal{S}, a \in \mathcal{A}$
Initialize $Q(s,a)$ arbitrarily, for all $s \in \mathcal{S}, a \in \mathcal{A}$
Initialize π to be greedy with respect to Q, or as a fixed given policy
Algorithm parameters: step size $\alpha \in (0,1]$, a positive integer n
All store and access operations (for S_t, A_t, and R_t) can take their index mod $n+1$

Loop for each episode:
  Initialize and store $S_0 \neq$ terminal
  Select and store an action $A_0 \sim b(\cdot|S_0)$
  $T \leftarrow \infty$
  Loop for $t = 0, 1, 2, \ldots$:
  | If $t < T$, then:
  | Take action A_t
  | Observe and store the next reward as R_{t+1} and the next state as S_{t+1}
  | If S_{t+1} is terminal, then:
  | $T \leftarrow t+1$
  | else:
  | Select and store an action $A_{t+1} \sim b(\cdot|S_{t+1})$
  | $\tau \leftarrow t - n + 1$ (τ is the time whose estimate is being updated)
  | If $\tau \geq 0$:
  | $\rho \leftarrow \prod_{i=\tau+1}^{\min(\tau+n-1,T-1)} \frac{\pi(A_i|S_i)}{b(A_i|S_i)}$ $(\rho_{\tau+1:t+n-1})$
  | $G \leftarrow \sum_{i=\tau+1}^{\min(\tau+n,T)} \gamma^{i-\tau-1} R_i$
  | If $\tau + n < T$, then: $G \leftarrow G + \gamma^n Q(S_{\tau+n}, A_{\tau+n})$ $(G_{\tau:\tau+n})$
  | $Q(S_\tau, A_\tau) \leftarrow Q(S_\tau, A_\tau) + \alpha\rho\,[G - Q(S_\tau, A_\tau)]$
  | If π is being learned, then ensure that $\pi(\cdot|S_\tau)$ is greedy wrt Q
  Until $\tau = T - 1$