强化学习原理：
——无模型(MC&TD)强化学习算法

杨博渊

yby@nankai.edu.cn

2023.03.10

提纲

- 无模型强化学习概述
- 无模型预测
  - 蒙特卡洛
  - 时间差分
  - TD(λ)
- 无模型控制
  - On-policy蒙特卡洛
  - On-policy时间差分
  - Off-policy学习
- 总结

无模型强化学习概述

- 上次课
  - 动态规划
  - 求解已知的MDP
- 本次&下次课
  - 无模型预测
  - 估计一个未知MDP的值函数
  - 无模型控制
  - 最优化一个未知MDP的值函数

动态规划算法

算法1：策略迭代算法

[1] 输入：状态转移概率 $P_{ss'}^{a}$ 回报函数 R_s^a ， 折扣因子 γ

初始化值函数：$V(s)=0$ 初始化策略 π_0

[2] Repeat l=0,1,…

[3] find V^{π_l} Policy evaluation

[4] $\pi_{l+1}(s) \in \arg\max_a q^{\pi_l}(s,a)$ Policy improvement

[5] Until $\pi_{l+1}=\pi_l$

[6] 输出： $\pi^*=\pi_l$

值迭代算法

[1] 输入： 状态转移概率 $P_{ss'}^{a}$ 回报函数 R_s^a ， 折扣因子 γ

初始化值函数：$v(s)=0$ 初始化策略 π_0

[2] Repeat l=0,1,…

[3] for every s do

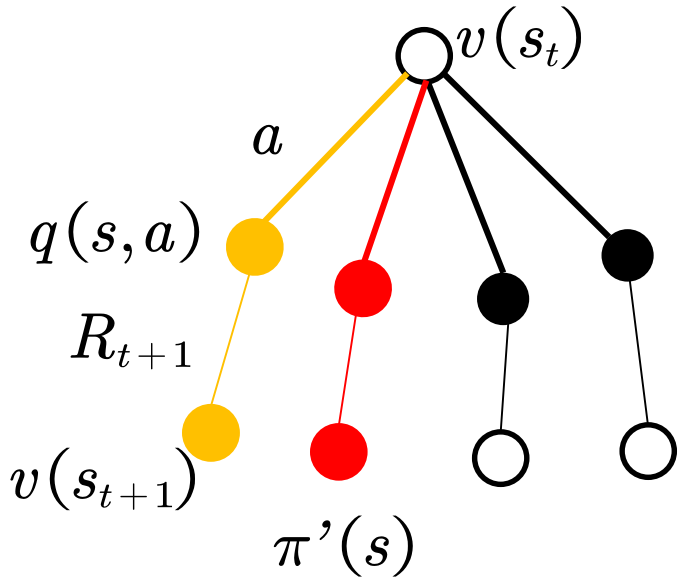
[4] $v_{l+1}(s)=\max_a R_s^a+\gamma\sum_{s'\in S}P_{ss'}^{a}v_l(s')$

[5] Until $v_{l+1}=v_l$

[6] 输出： $\pi(s)=\arg\max_a R_s^a+\gamma\sum_{s'\in S}P_{ss'}^{a}v_l(s')$

动态规划：策略改进证明

$$
\begin{aligned}
& v_\pi(s) \leqslant q_\pi(s, \pi'(s)) \\
& = \mathbb{E}[R_{t+1} + \gamma v_\pi(s_{t+1}) | s_t = s,\ a_t = \pi'(s)] \\
& = \mathbb{E}_{\pi'}[R_{t+1} + \gamma v_\pi(s_{t+1}) | s_t = s] \\
& \leqslant \mathbb{E}_{\pi'}[R_{t+1} + \gamma q_\pi(s_{t+1}, \pi'(s_{t+1})) | s_t = s] \\
& = \mathbb{E}_{\pi'}[R_{t+1} + \gamma \mathbb{E}_{\pi'}[R_{t+2} + \gamma v_\pi(s_{t+1}) | s_{t+1}, a_{t+1} = \pi'(s_{t+1})] | s_t = s] \\
& = \mathbb{E}_{\pi'}[R_{t+1} + \gamma R_{t+2} + \gamma^2 v_\pi(s_{t+2}) | s_t = s] \\
& \leqslant \mathbb{E}_{\pi'}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 v_\pi(s_{t+2}) | s_t = s] \\
& \vdots \\
& \leqslant \mathbb{E}_{\pi'}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots | s_t = s] \\
& = v_{\pi'}(s)
\end{aligned}
$$



最直观的一个改进策略是什么？　　贪婪策略！

无模型强化学习概述

一些可以建模为MDPs的问题：

- 电梯
- 帆船操控
- 自动泊车
- 生物反应器
- 直升飞机
- 航空物流
- 足球世界杯
- 地震
- 投资管理
- 蛋白质折叠
- 机器人行走
- 围棋

以上大部分问题：

- MDP模型未知，经验也被采样
- MDP模型已知，规模太大

无模型强化学习可以解决这些问题

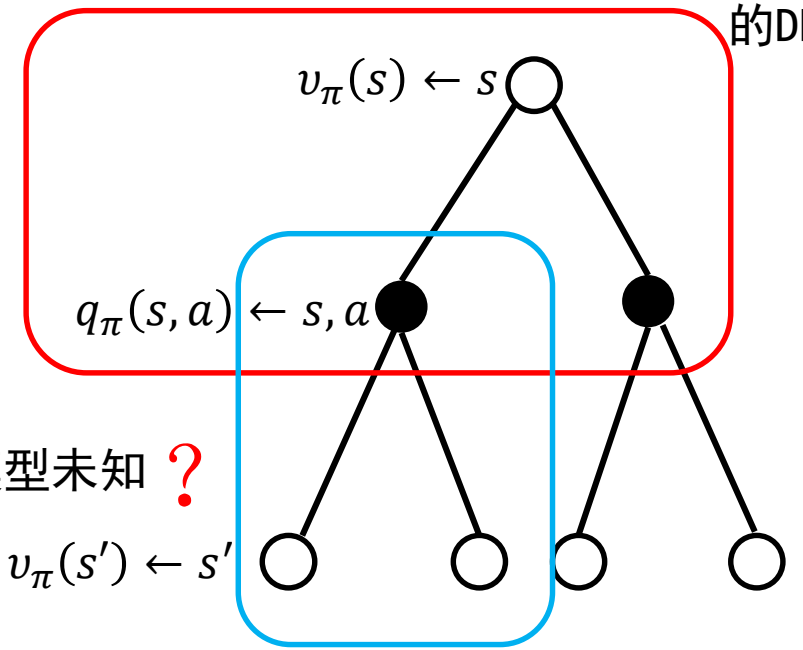
提纲

- 无模型强化学习概述
- 无模型预测
  - 蒙特卡洛
  - 时间差分
  - TD(λ)
- 无模型控制
  - On-policy蒙特卡洛
  - On-policy时间差分
  - Off-policy学习
- 总结

蒙特卡洛预测

给定策略 π 构造值函数：

基于模型已知的DP方法



$$v_\pi(s) = \sum_{a \in A} \pi(a|s) \left(R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_\pi(s') \right)$$

当智能体采用策略 π 时，累积回报服从一个概率分布，累积回报在状态s处的期望值定义为值函数：

$$v_\pi(s) = E_\pi[G_t | S_t = s] = E_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s \right]$$

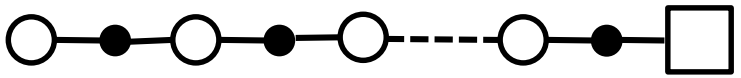
状态-行为值函数：

$$q_\pi(s,a) = E_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a \right]$$

蒙特卡洛方法利用经验平均代替随机变量的期望：

一次实验（an episode）： $S_1, A_1, R_2, \cdots, S_k \sim \pi$

终止状态：



计算状态s后的折扣回报返回值：

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

蒙特卡洛预测

动态规划策略评估算法

输入：需要评估的策略 π 状态转移概率 $P_{ss'}^{a}$ 回报函数 R_s^a ，折扣因子 γ

初始化值函数：$V(s)=0$

一次状态扫描

Repeat k=0,1,…

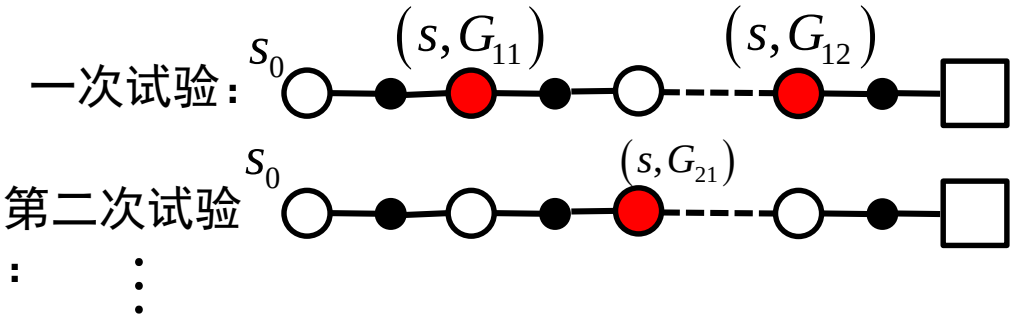
for every s do

$$v_{k+1}(s)=\sum_{a\in A}\pi(a|s)\left(R_s^a+\gamma\sum_{s'\in S}P_{ss'}^{a}v_k(s')\right)$$

end for

Until $v_{k+1}=v_k$

输出： $v(s)$



蒙特卡罗方法利用经验平均代替随机变量的期望。

First visit MC策略评估： $v(s)=\dfrac{G_{11}(s)+G_{21}(s)+\cdots}{N(s)}$

every visit MC策略评估：

$$v(s)=\frac{G_{11}(s)+G_{12}(s)+\cdots G_{21}(s)+\cdots}{N(s)}$$

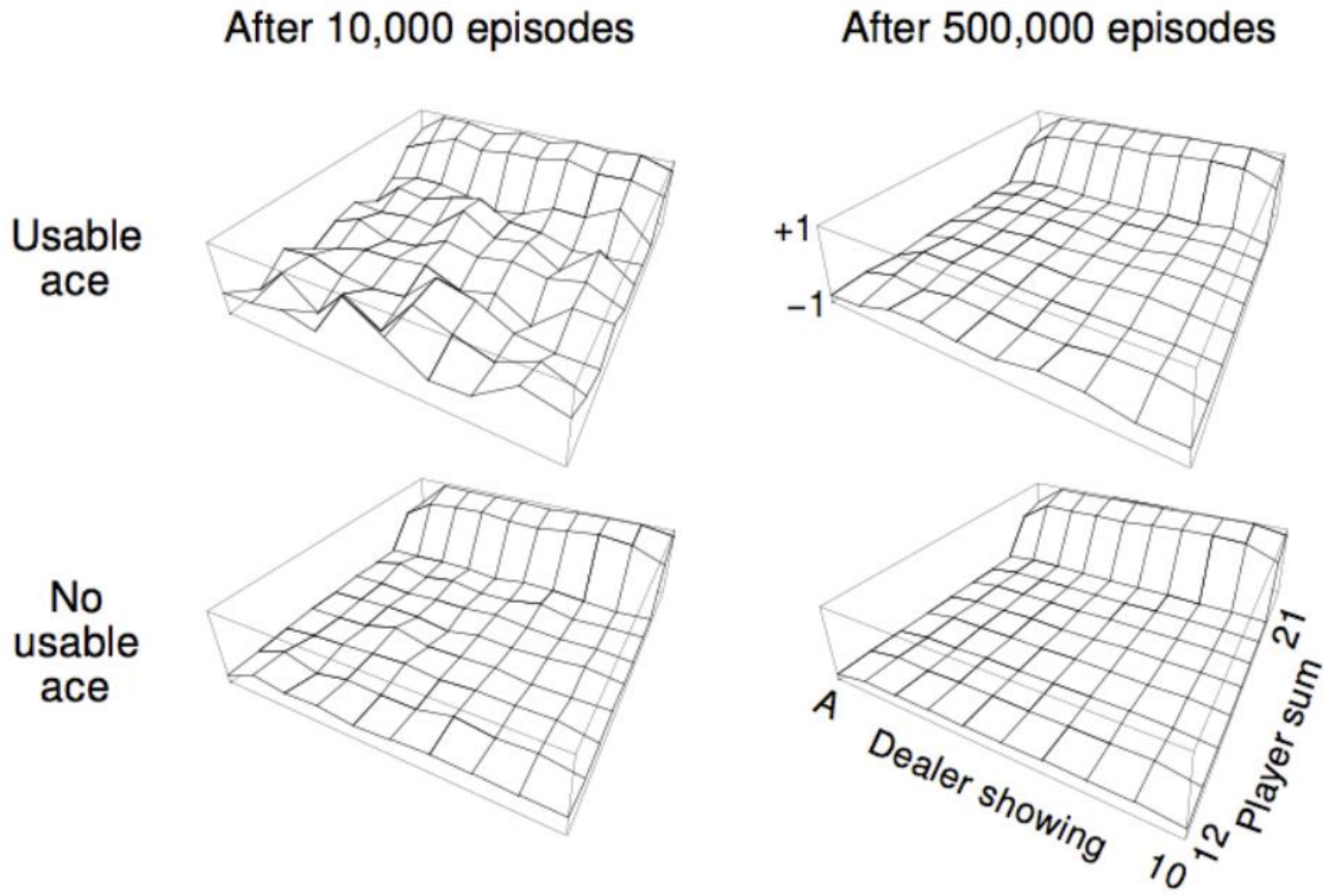
根据大数定律：$v(s)\to v_\pi(s)\ as\ N(s)\to\infty$

蒙特卡洛:21点

蒙特卡洛:21点

庄家总序列	玩家最终序列	玩家获得回报	当前估计的状态值
4, 10, 3	Q, 5, 5	+1	+1
4, J, 7	9, 6, 8	-1	0
4, 10, 2, 8	7, 8, 9	+1	0.333
4, 4, 2, 7	J, 5, Q,	-1	0
4, 9	5, K, 4, 8	-1	-0.2
4, 3, K	8, 7, 5	+1	0
…	…	…	…

蒙特卡洛:21点


After 10,000 episodes
After 500,000 episodes
Usable ace
No usable ace
+1
−1
A
Dealer showing
10
12
Player sum
21

蒙特卡洛:增量式更新

增量式计算:

$$
\begin{aligned}
Q_{n+1} &= \frac{1}{n}\sum_{i=1}^{n} R_i \\
&= \frac{1}{n}\left(R_n + \sum_{i=1}^{n-1} R_i\right) \\
&= \frac{1}{n}\left(R_n + (n-1)\frac{1}{n-1}\sum_{i=1}^{n-1} R_i\right) \\
&= \frac{1}{n}(R_n + (n-1)Q_n) \\
&= \frac{1}{n}(R_n + nQ_n - Q_n) \\
&= Q_n + \frac{1}{n}[R_n - Q_n]
\end{aligned}
$$

- 对于每一个状态S_t, 有回报G_t

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

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

- 非静态问题:

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

提纲

- 无模型强化学习概述
- 无模型预测
  - 蒙特卡洛
  - 时间差分
  - TD(λ)
- 无模型控制
  - On-policy蒙特卡洛
  - On-policy时间差分
  - Off-policy学习
- 总结

时间差分学习

增量式MC方法估计值函数：

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

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

最简单的时间差分学习算法：TD(0)

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

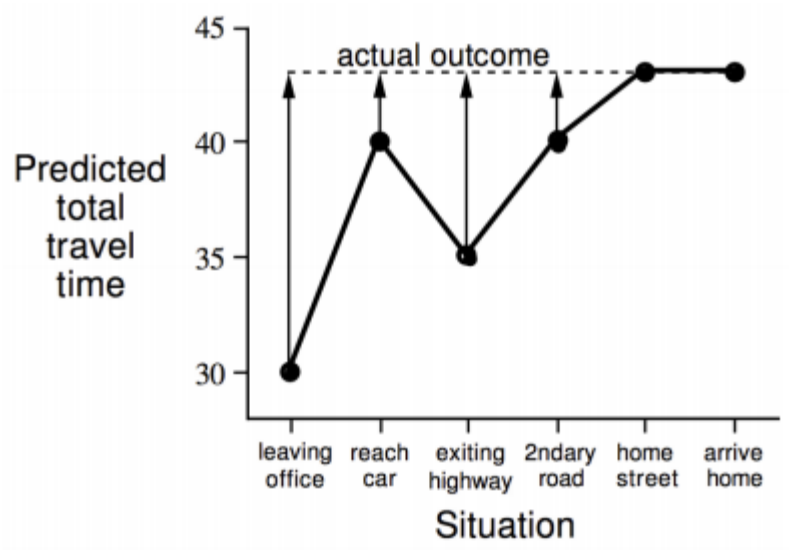
$R_{t+1} + \gamma V(S_{t+1})$ 称为TD目标

时间差分学习：驾车返家

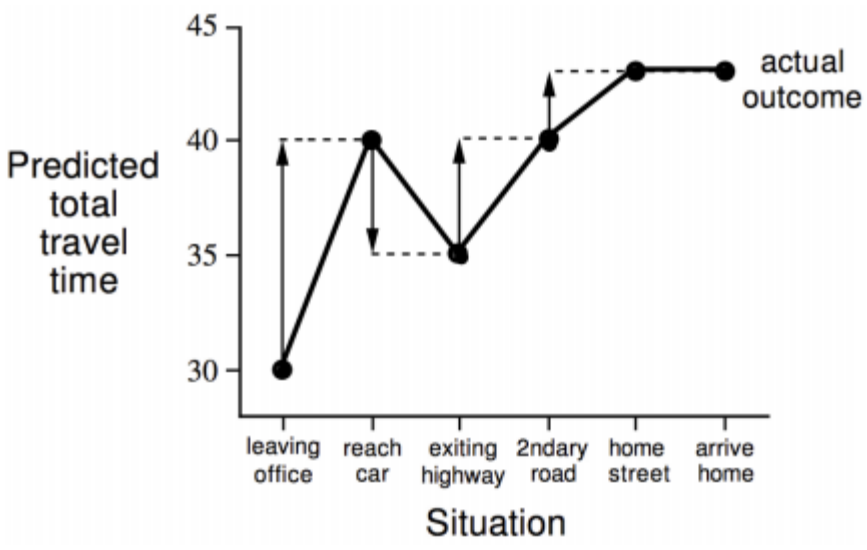
状态	已消耗时间（分）	预计仍需耗时（分）	预测总耗时（分）
离开办公室	0	30	30
取车，发现下雨	5	35	40
离开高速公路	20	15	35
被迫跟在卡车后面	30	10	40
到达家所在街区	40	3	43
进入家门	43	0	43

时间差分学习：驾车返家

Changes recommended by Monte Carlo methods (α=1)



Changes recommended by TD methods (α=1)



TD与MC对比

TD 在知道结果之前可以学习，MC必须等到最后结果才能学习。

TD 可以在没有结果时学习，可以在持续进行的环境里学习。

TD与MC偏差与方差平衡

增量式MC方法估计值函数：

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

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

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

最简单的时间差分学习算法：TD(0)

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

$R_{t+1} + \gamma V(S_{t+1})$ 称为TD目标

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

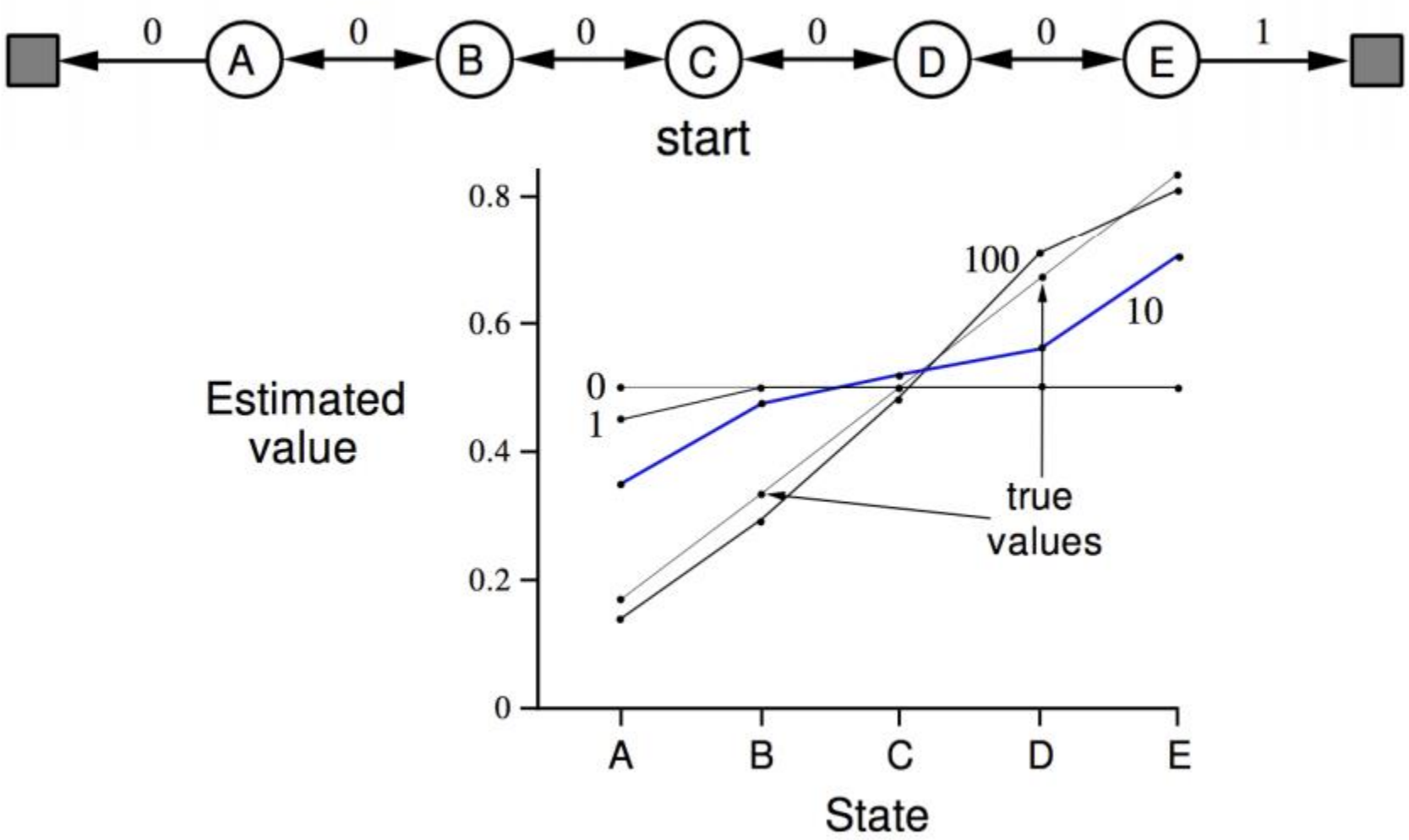
TD目标 $R_{t+1} + \gamma\upsilon_\pi(S_{t+1})$ 的方差比MC的返回值 G_t 要小很多。因为MC的返回值依赖于很多随机动作，转移概率和回报。TD目标仅依赖于一个随机动作，转移概率和回报。

TD与MC对比

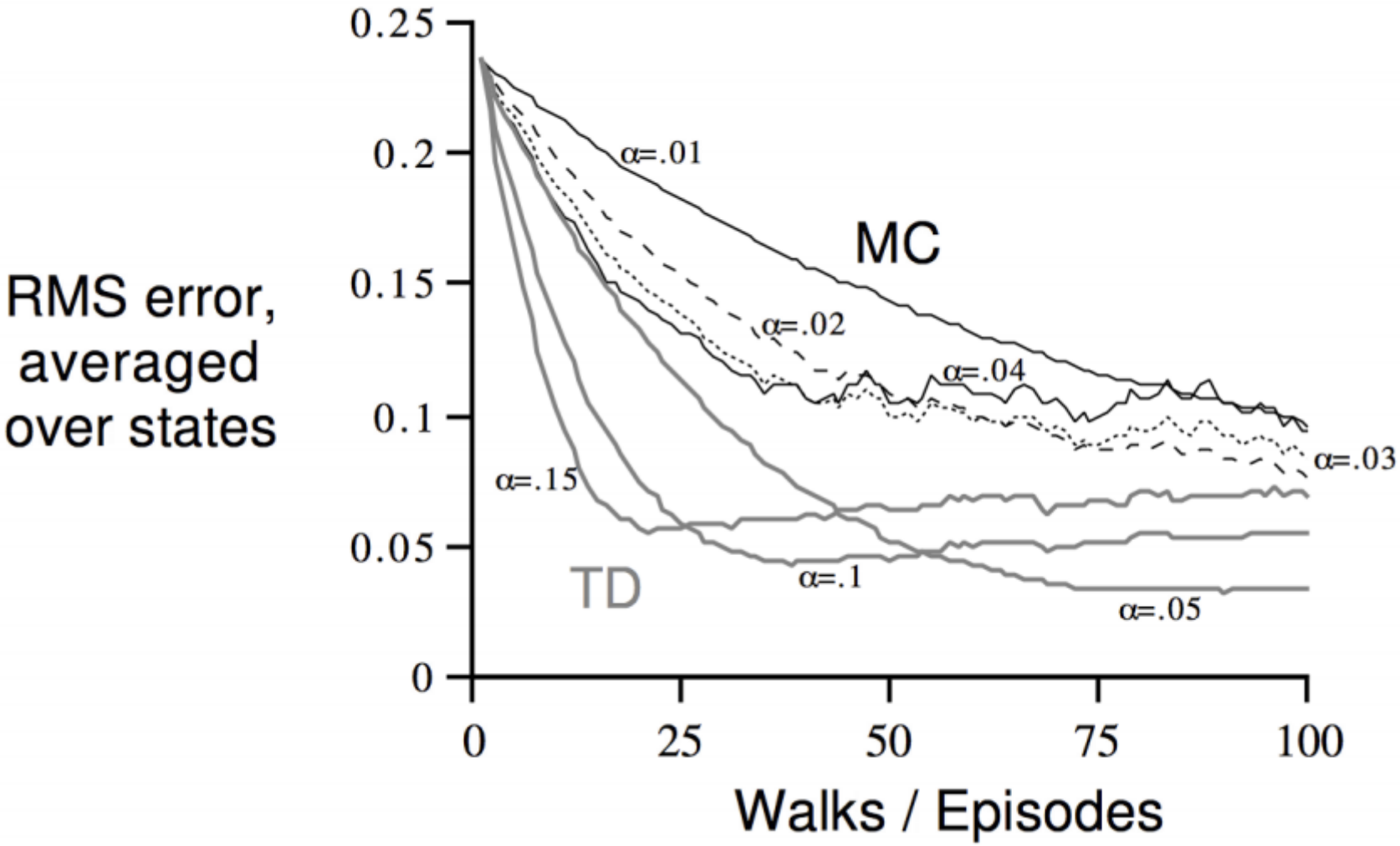
MC 没有偏差（bias），但有着较大的方差（Variance），且对初始值不敏感；

TD 小方差variance, 但有一定程度的偏差，对初始值较敏感，通常比 MC 更高效；；

时间差分学习：随机行走


0
A
0
B
0
C
0
D
0
E
1
start
Estimated value
0.8
0.6
0.4
0.2
0
100
10
0
1
true values
A
B
C
D
E
State

时间差分学习：随机行走


RMS error, averaged over states
0.25
0.2
0.15
0.1
0.05
0
0
25
50
75
100
Walks / Episodes
MC
TD
α=.01
α=.02
α=.04
α=.03
α=.15
α=.1
α=.05

批量MC与TD

MC与TD收敛：$V(s) \to v_\pi(s)$，当样本 $\to \infty$

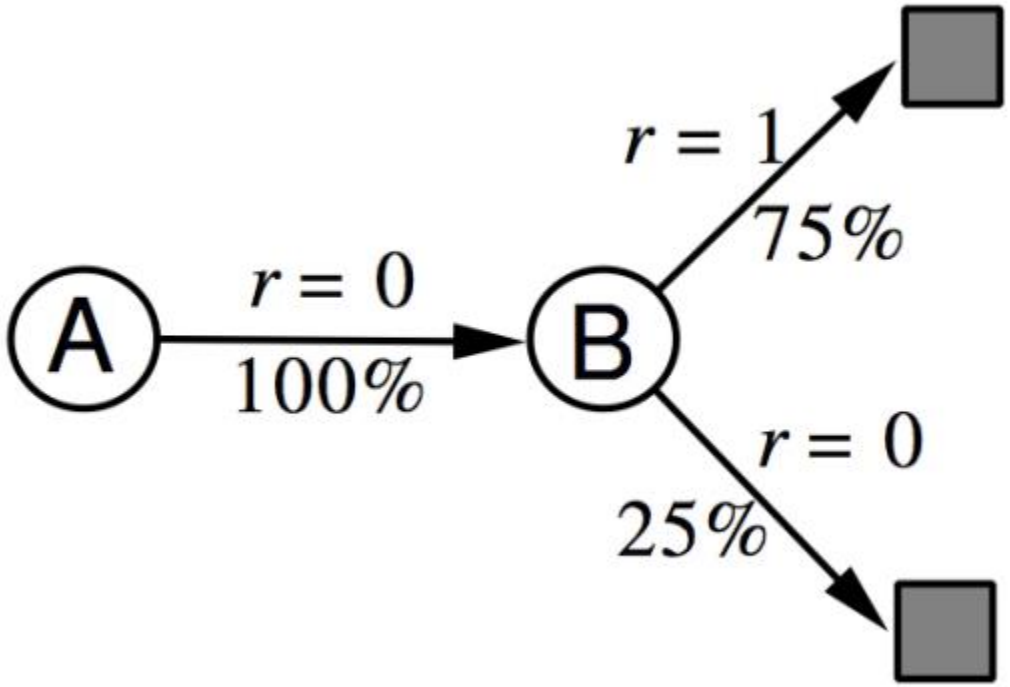
对于有限样本的批量求解呢？

$$s_1^1, a_1^1, r_2^1, \ldots, s_{T_1}^1$$
$$\vdots$$
$$s_1^K, a_1^K, r_2^K, \ldots, s_{T_K}^K$$

MC与TD：AB例子

现有两个状态(A和B)， 8个完整Episode

Episode	状态转移及奖励
1	A:0, B:0
2	B:1
3	B:1
4	B:1
5	B:1
6	B:1
7	B:1
8	B:0



V(A), V(B)？

批量MC与TD

MC算法试图收敛至一个能够最小化状态值与实际回报的均方差的解

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

TD算法则收敛至一个根据已有经验构建的最大可能的马尔科夫模型的状态值

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

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

TD与MC对比

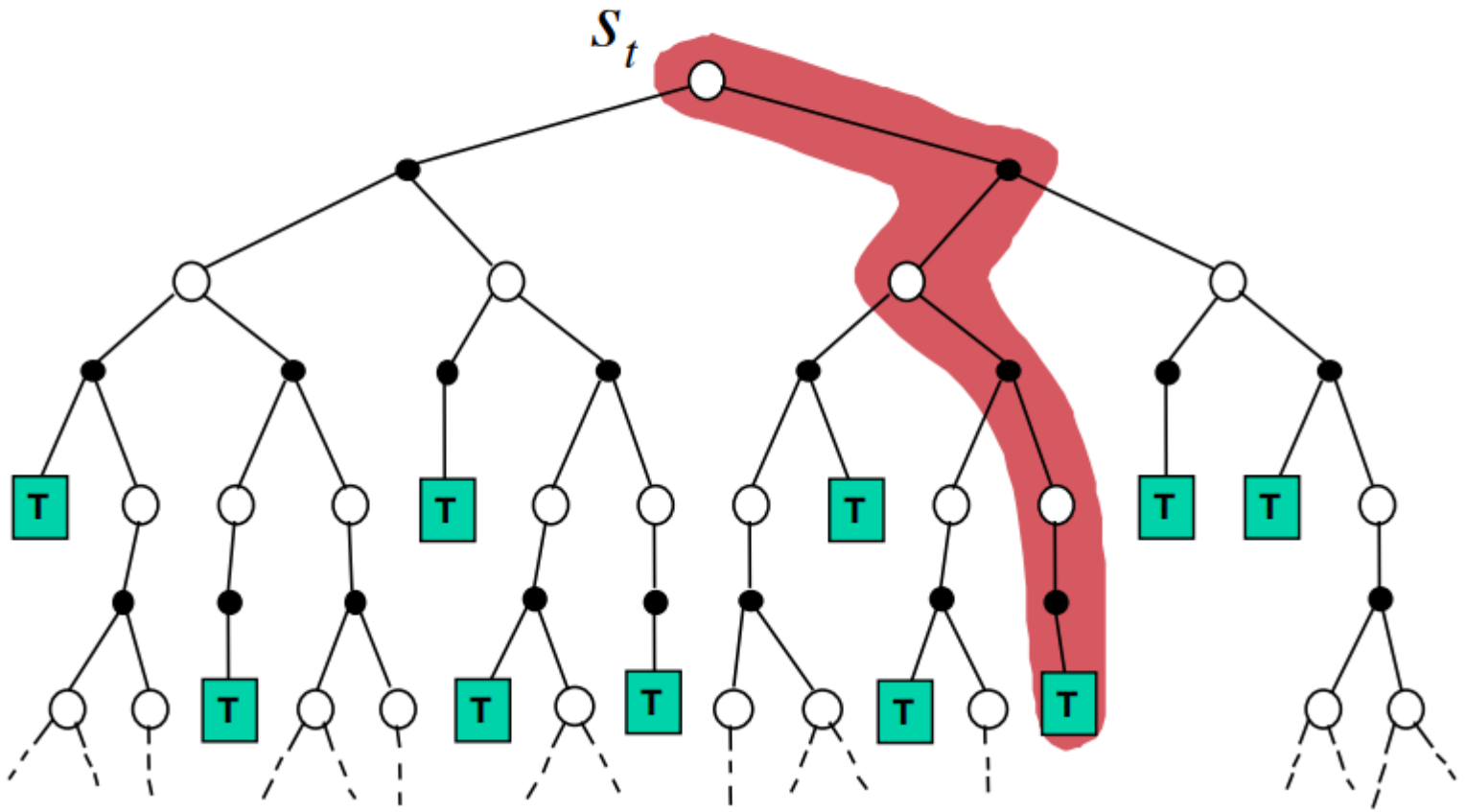
TD算法使用了MDP问题的马尔科夫属性，在Markov 环境下更有效

MC算法并不利用马尔科夫属性，通常在非Markov环境下更有效

小结：三种算法MC,TD,DP

MC回溯

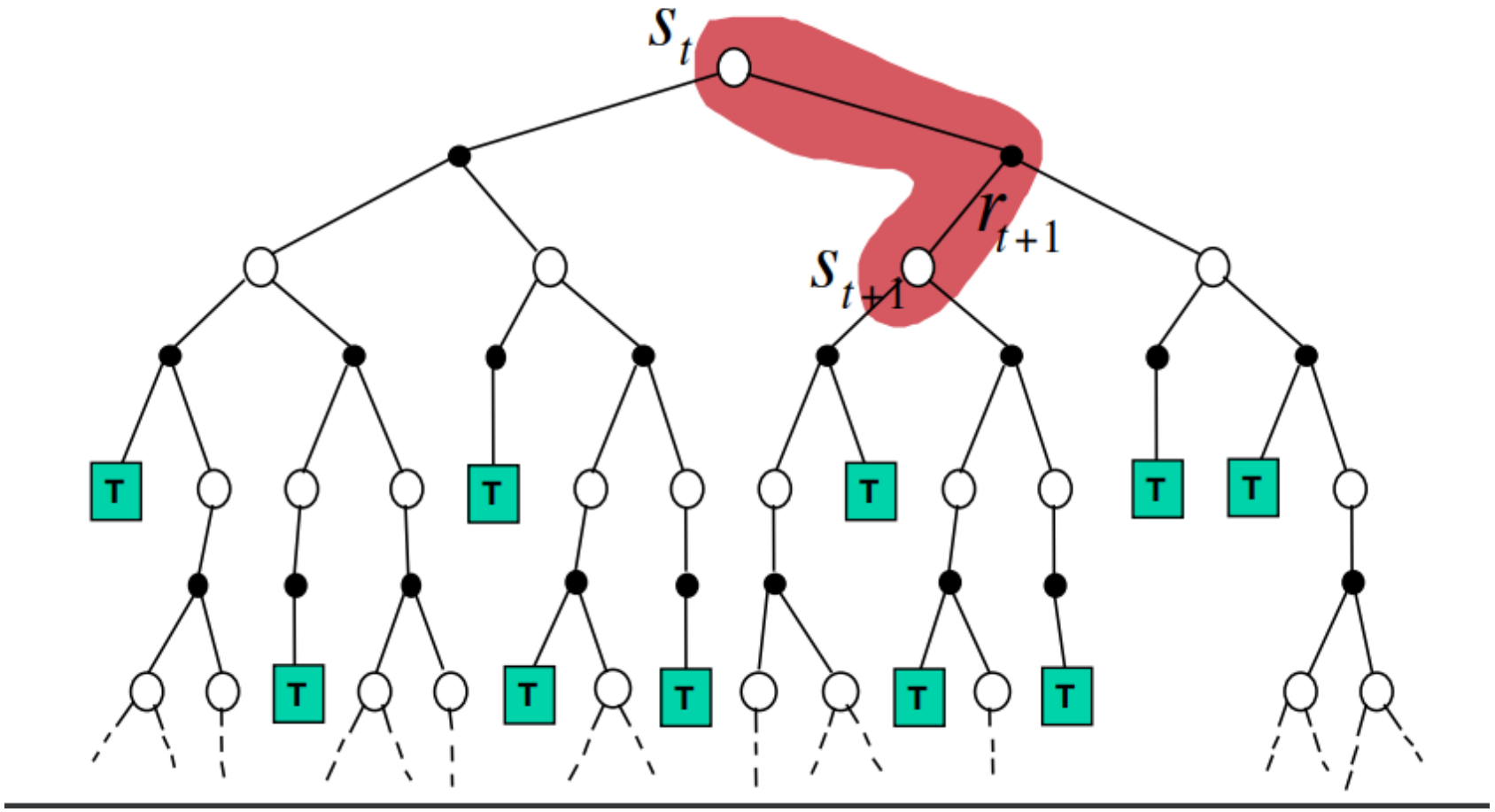
$$V(S_t) \leftarrow V(S_t) + \alpha\left(G_t - V(S_t)\right)$$



小结：三种算法MC,TD,DP

TD回溯

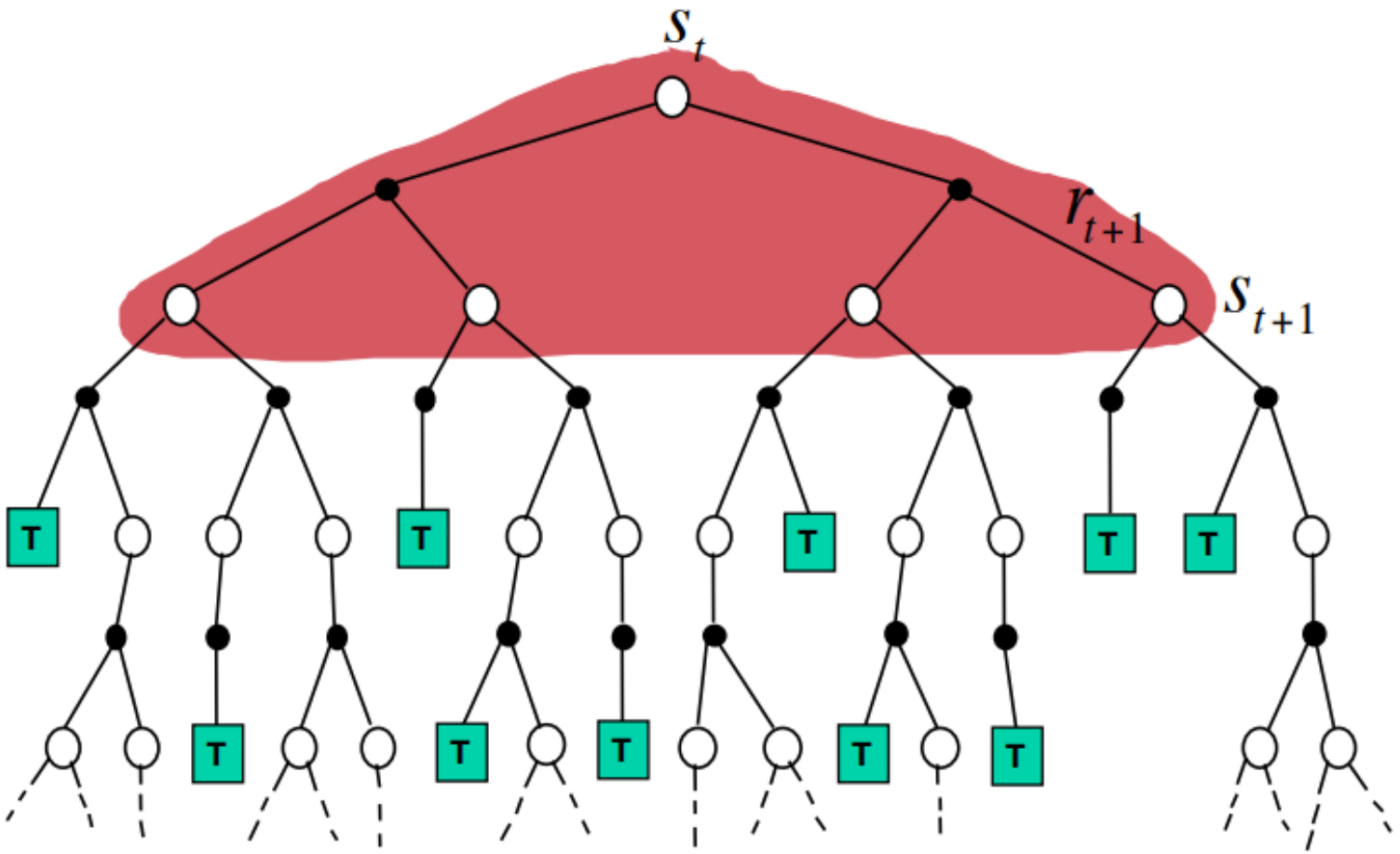
$$V(S_t) \leftarrow V(S_t) + \alpha\left(R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\right)$$



小结：三种算法MC,TD,DP

DP回溯

$$V(S_t) \leftarrow \mathbb{E}_\pi \left[R_{t+1} + \gamma V(S_{t+1}) \right]$$



小结：三种算法MC,TD,DP

MC：利用采样平均回报逼近期望

$$\upsilon_\pi(s) = E_\pi[G_t \mid S_t = s]$$

$$= E_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1} \mid S_t = s\right]$$

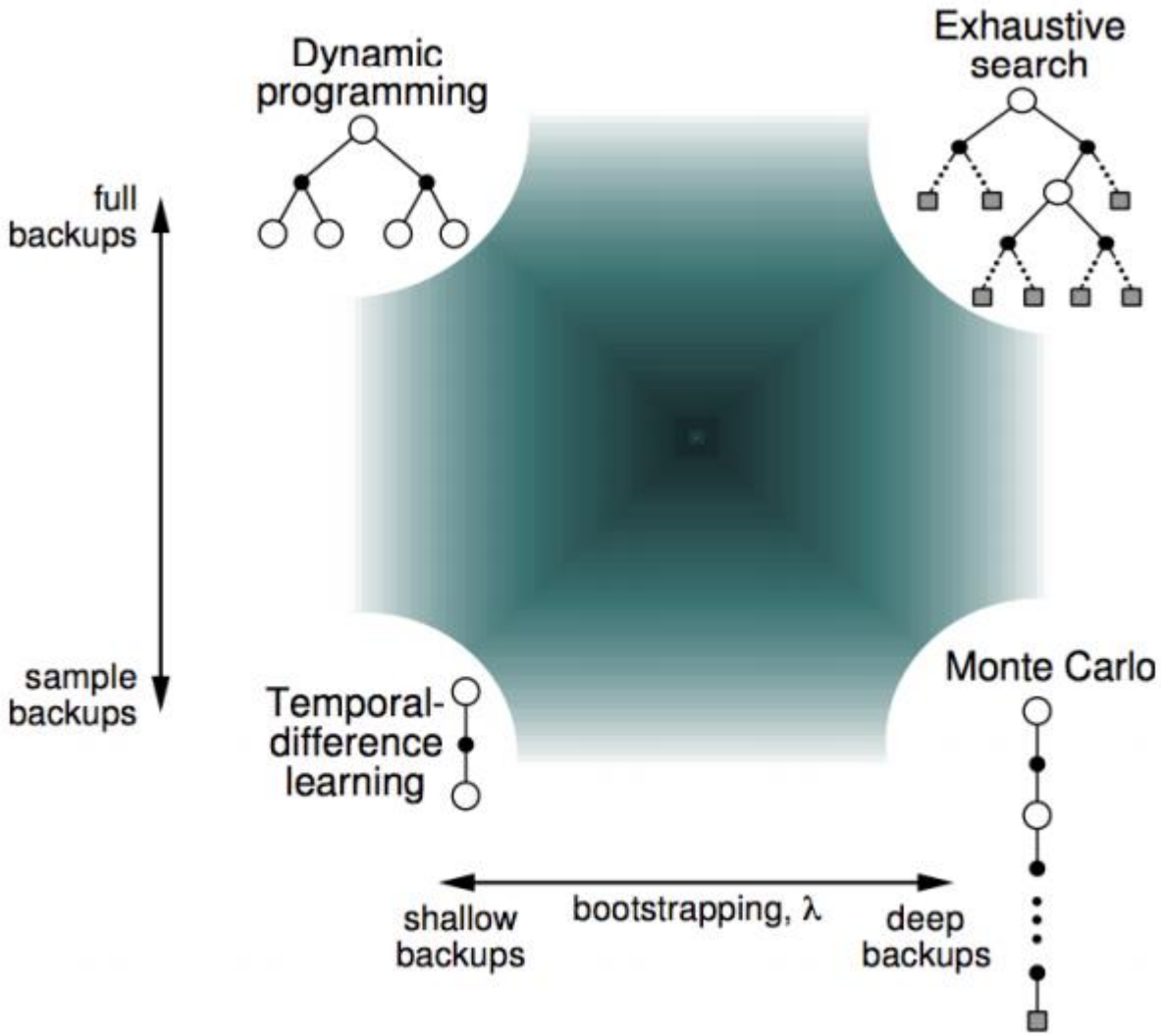
$$= E_\pi\left[R_{t+1} + \gamma\sum_{k=0}^{\infty}\gamma^k R_{t+k+2} \mid S_t = s\right]$$

$$= E_\pi\left[R_{t+1} + \gamma\upsilon_\pi(S_{t+1}) \mid S_t = s\right]$$

TD：联合了MC和DP，采样期望值，并利用真值的当前估计值 $V(S_{t+1})$

DP:期望值由模型来提供，但是利用真值的当前估计值 $V(S_{t+1})$

小结：三种算法MC,TD,DP

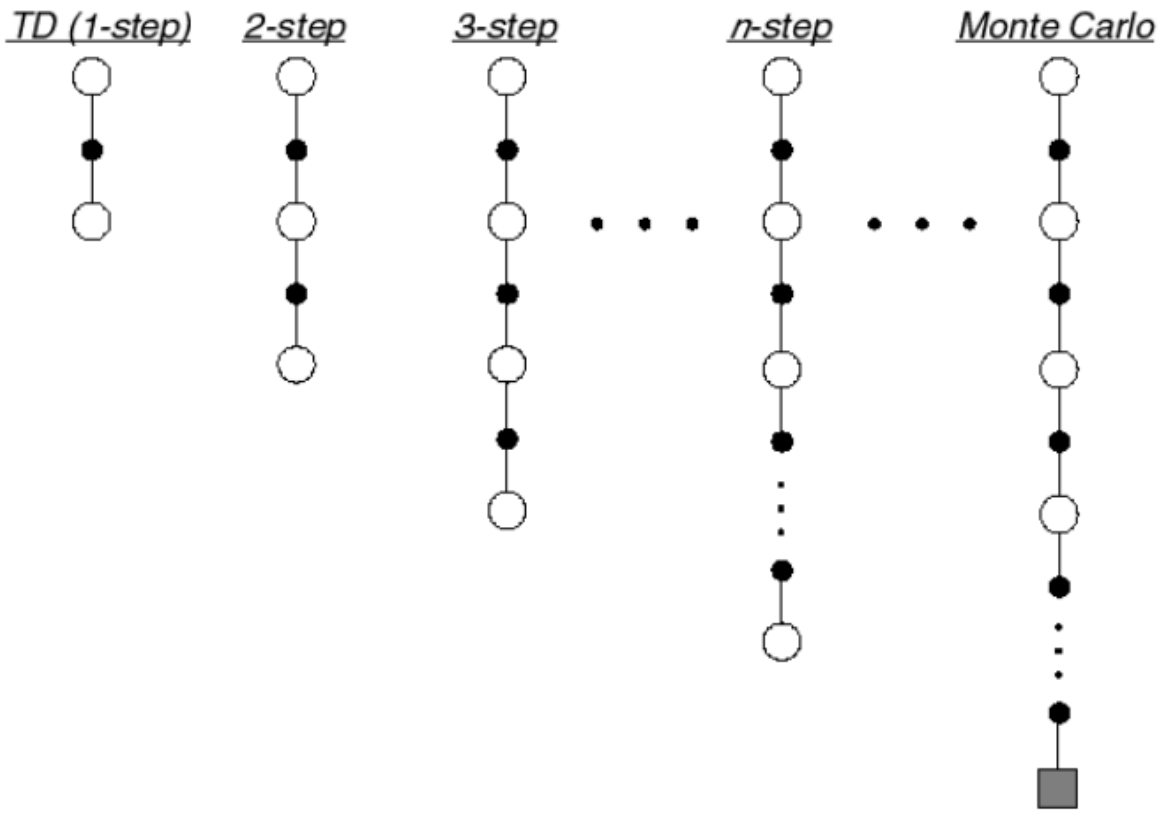


提纲

- 无模型强化学习概述
- 无模型预测
  - 蒙特卡洛
  - 时间差分
  - TD(λ)
- 无模型控制
  - On-policy蒙特卡洛
  - On-policy时间差分
  - Off-policy学习
- 总结

TD(λ)：n步预测

TD目标向前看n步



TD(λ)：n步预测

考虑n步的回报，n=1,2,∞

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

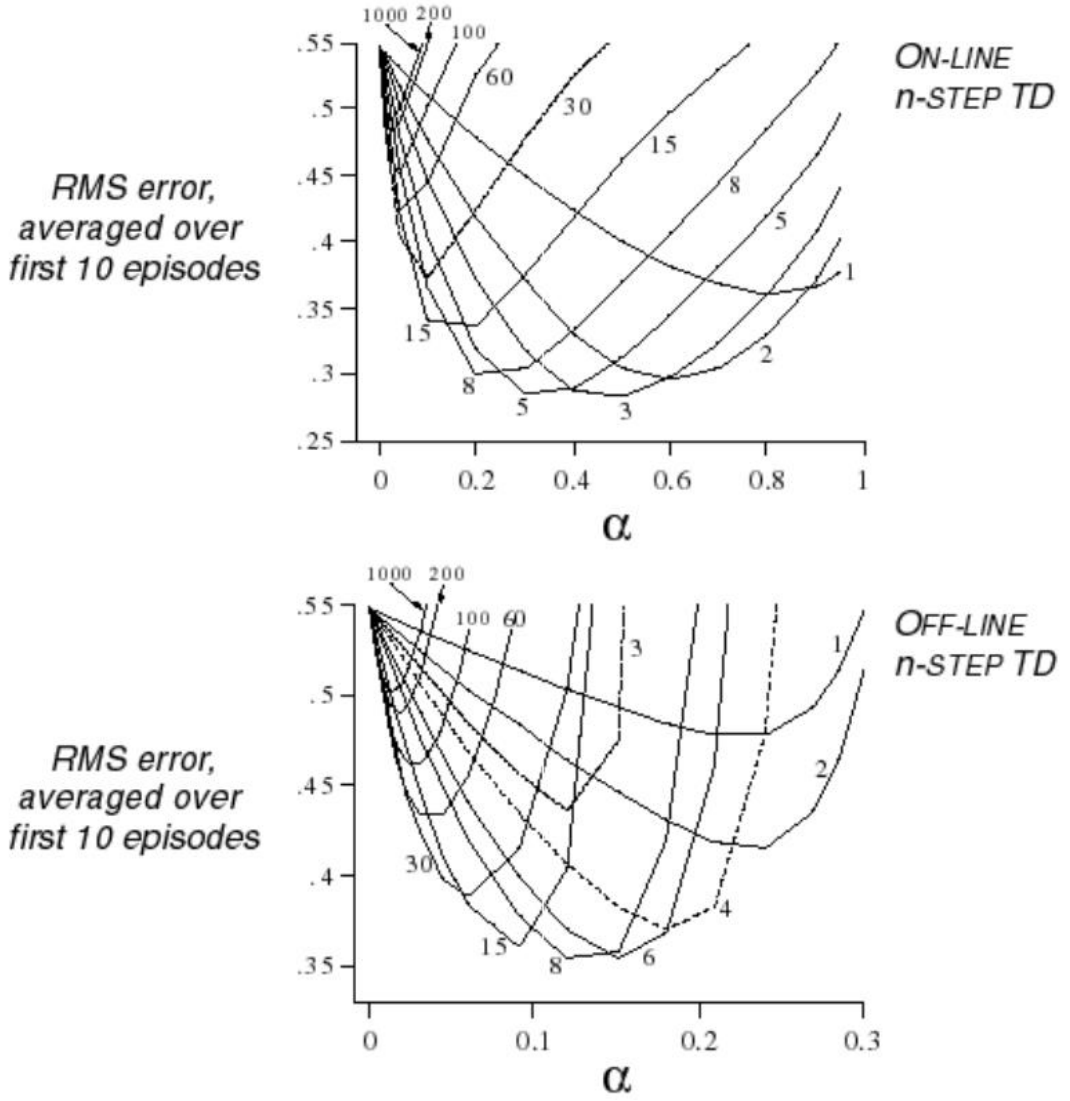
定义n步回报为

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

n步时间差分学习

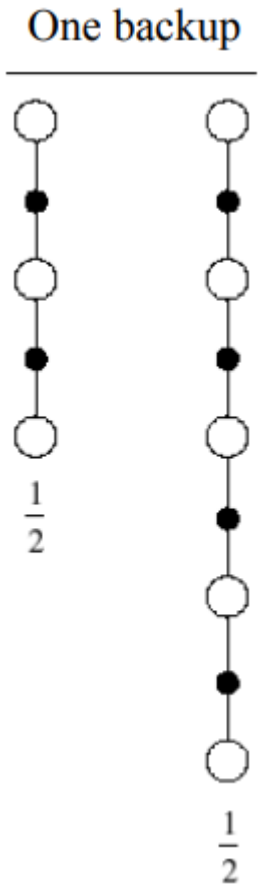
$$V(S_t) \leftarrow V(S_t) + \alpha \left(G_t^{(n)} - V(S_t) \right)$$

示例：大规模随机行走


RMS error, averaged over first 10 episodes
ON-LINE n-STEP TD
1000 200 100 60 30 15 8 5 1 15 8 5 3 2
.55 .5 .45 .4 .35 .3 .25
0 0.2 0.4 0.6 0.8 1
α
RMS error, averaged over first 10 episodes
OFF-LINE n-STEP TD
1000 200 100 60 3 1 2 30 4 15 8 6
.55 .5 .45 .4 .35
0 0.1 0.2 0.3
α

平均n步回报

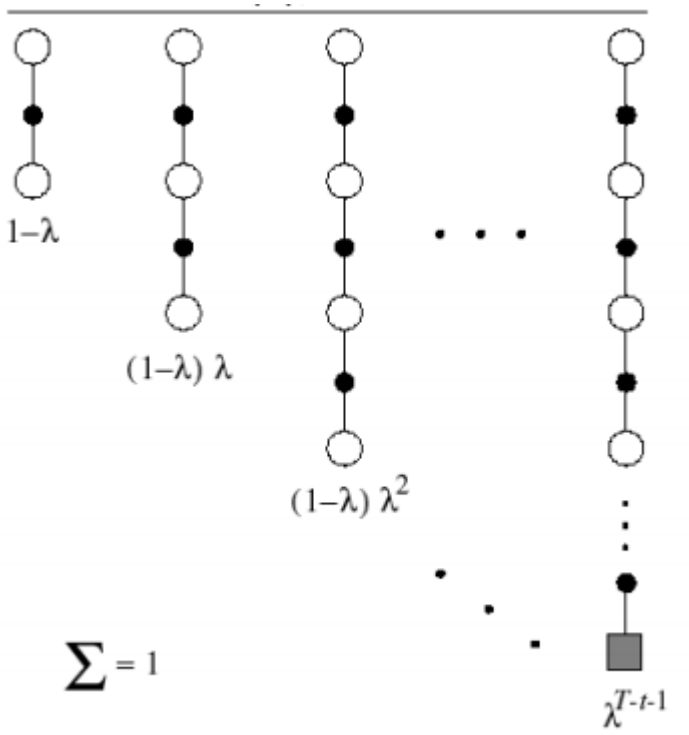
$$\frac{1}{2}G^{(2)} + \frac{1}{2}G^{(4)}$$



是否可以更高效的结合全部时间步的信息？

λ回报

TD(λ)，λ回报



λ回报 G_t^λ 结合了全部n步的回报 $G_t^{(n)}$

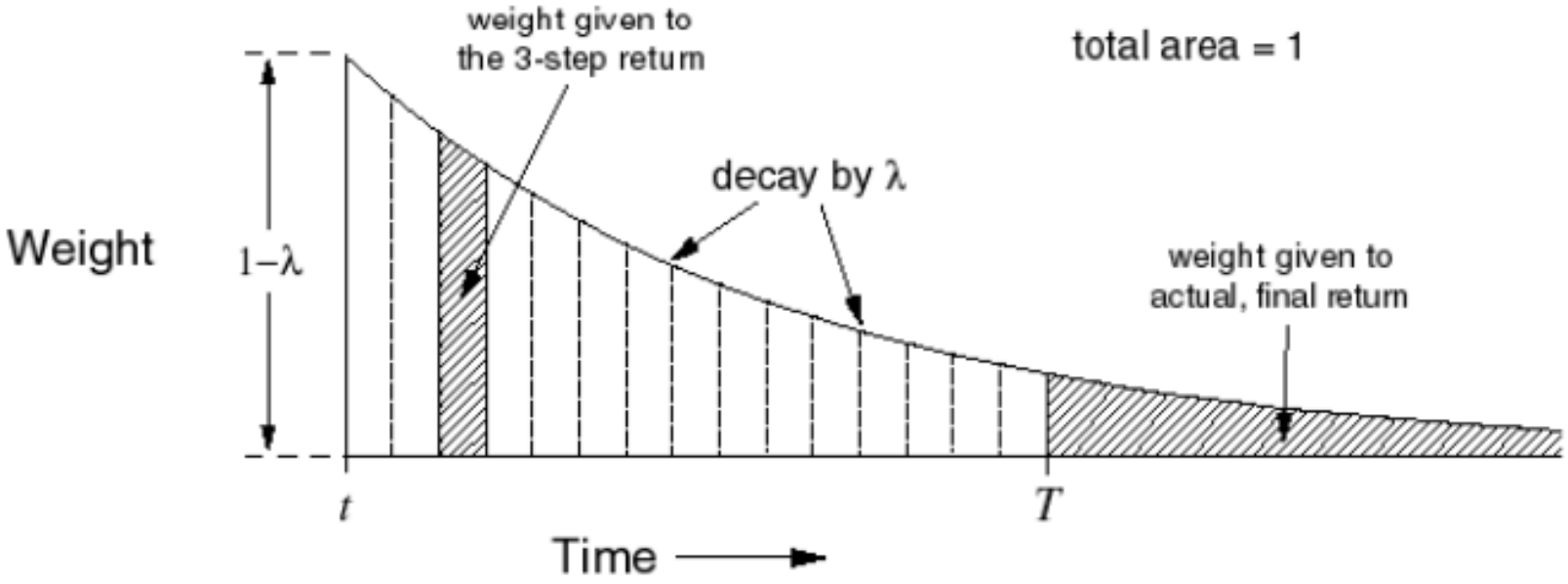
施加权重 $(1-\lambda)\lambda^{n-1}$

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

TD(λ)预测

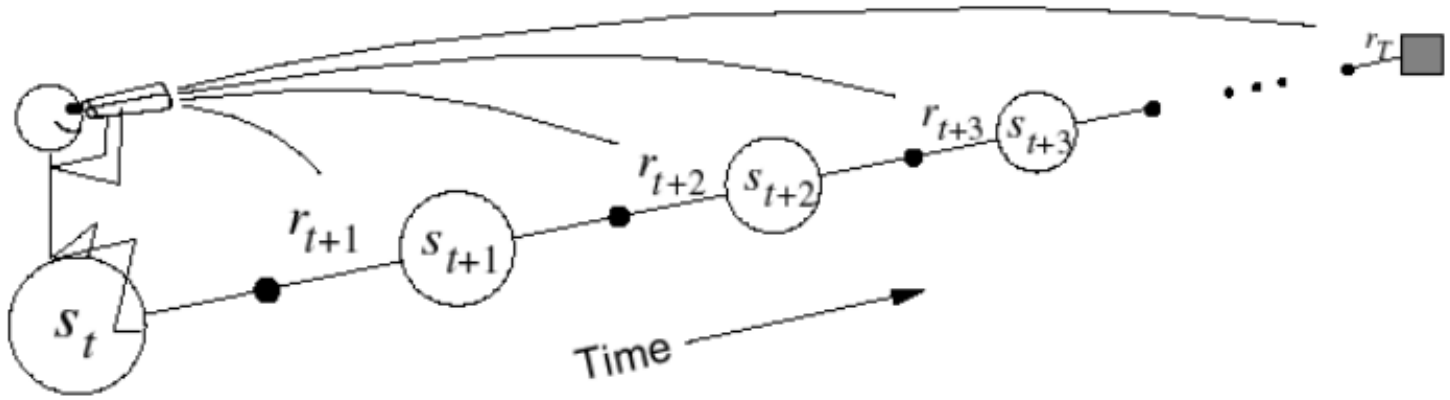
$$V(S_t) \leftarrow V(S_t) + \alpha\left(G_t^\lambda - V(S_t)\right)$$

TD(λ)权重方程



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

前向视图TD(λ)

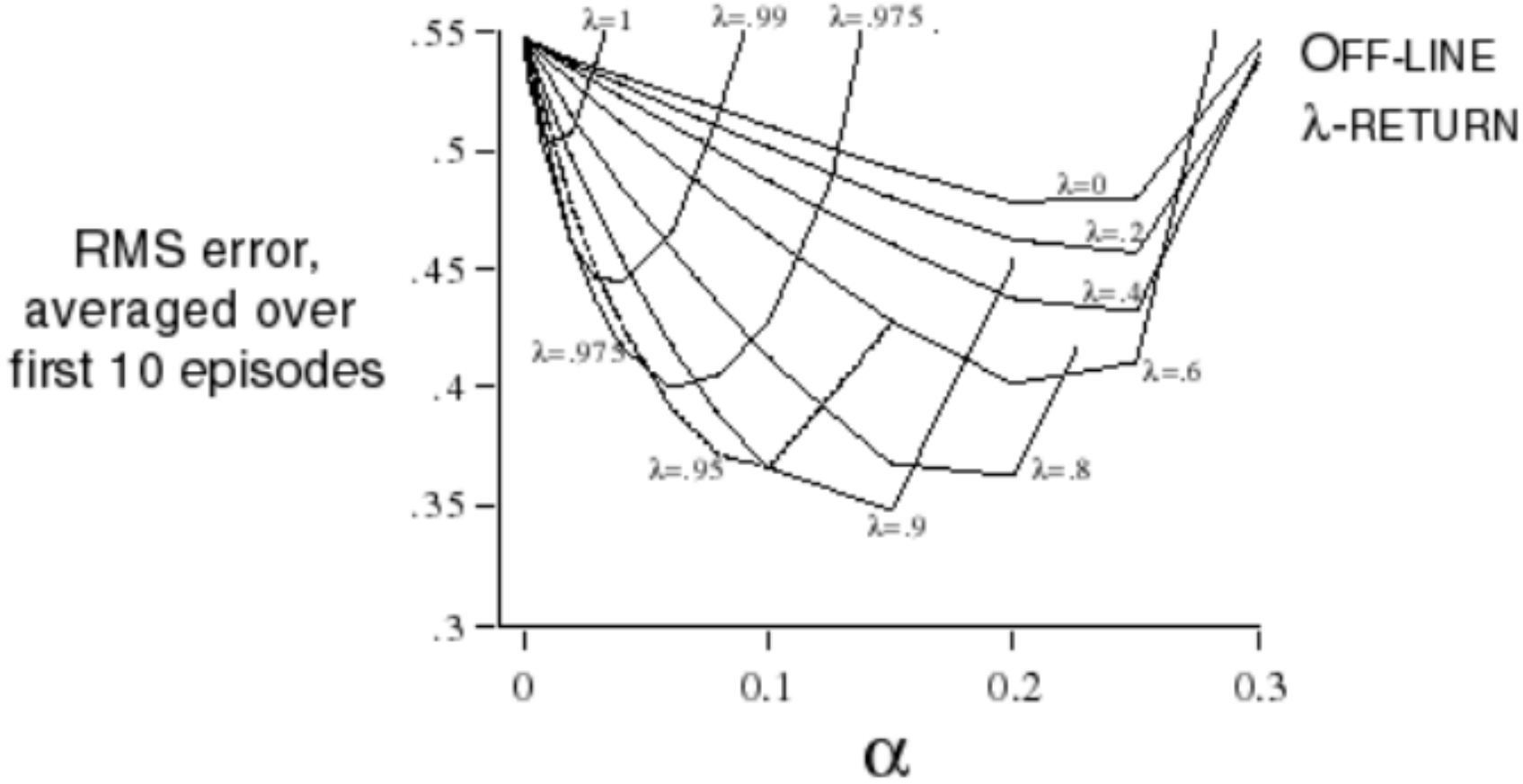


值函数向着λ回报更新

向前看（未来）λ回报

同MC一样，需要走完整个Episode

前向视图TD(λ)


RMS error,
averaged over
first 10 episodes
.55
.5
.45
.4
.35
.3
0
0.1
0.2
0.3
α
λ=1
λ=.99
λ=.975
λ=0
λ=.2
λ=.4
λ=.6
λ=.8
λ=.9
λ=.95
λ=.975
OFF-LINE
λ-RETURN

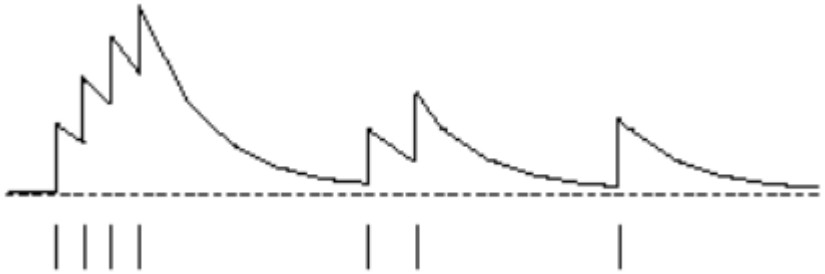
反向视图TD(λ)

响铃的因素较重要还是亮灯的因素更重要呢？

频率启发：将原因归因于出现频率最高的状态

就近启发：将原因归因于较近的几次状态

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



反向视图TD(λ)

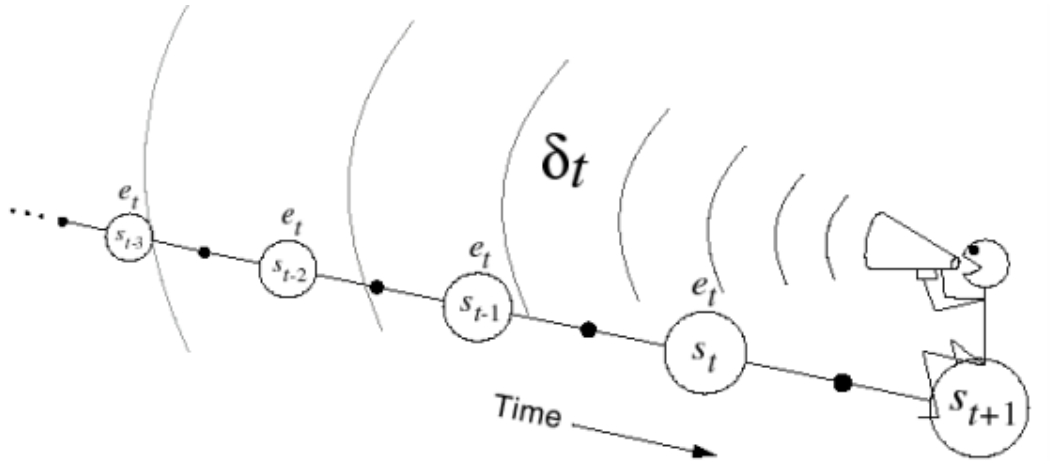
每一个状态都保持一个资格迹

更新每个状态的值

按比例加权TD误差和资格迹

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

$$V(s) \leftarrow V(s) + \alpha \delta_t E_t(s)$$



前项和反向视图TD(λ)小结

Offline updates	$\lambda = 0$	$\lambda \in (0,1)$	$\lambda = 1$
Backward view	TD(0)	TD(λ)	TD(1)
	$\Vert$	$\Vert$	$\Vert$
Forward view	TD(0)	Forward TD(λ)	MC
Online updates	$\lambda = 0$	$\lambda \in (0,1)$	$\lambda = 1$
Backward view	TD(0)	TD(λ)	TD(1)
	$\Vert$	$\nparallel$	$\nparallel$
Forward view	TD(0)	Forward TD(λ)	MC
	$\Vert$	$\Vert$	$\Vert$
Exact Online	TD(0)	Exact Online TD(λ)	Exact Online TD(1)

第四次作业

1.阅读《Reinforcement Learning： An Introduction》第五、六、十二章（预测部分）

2. 阅读MC方法和TD方法的实现