强化学习2022
第3节

涉及知识点：
无模型的强化学习、蒙特卡洛方法、蒙特卡洛价值预测、重要性采样、时序差分学习

值函数估计

张伟楠 – 上海交通大学

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. 规划与学习
6. 参数化的值函数和策略
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. 模仿学习
11. 离线强化学习
12. 参数化动作空间
13. 目标导向的强化学习
14. 多智能体强化学习
15. 强化学习大模型
16. 技术交流与回顾

无模型的强化学习

张伟楠 – 上海交通大学

无模型的强化学习（Model-free RL）

- 在现实问题中，通常没有明确地给出状态转移和奖励函数
  - 例如，我们仅能观察到部分片段（episodes）

Episode 1: $s_0^{(1)} \xrightarrow[R(s_0)^{(1)}]{a_0^{(1)}} s_1^{(1)} \xrightarrow[R(s_1)^{(1)}]{a_1^{(1)}} s_2^{(1)} \xrightarrow[R(s_2)^{(1)}]{a_2^{(1)}} s_3^{(1)} \dots s_T^{(1)}$

Episode 2: $s_0^{(2)} \xrightarrow[R(s_0)^{(2)}]{a_0^{(2)}} s_1^{(2)} \xrightarrow[R(s_1)^{(2)}]{a_1^{(2)}} s_2^{(2)} \xrightarrow[R(s_2)^{(2)}]{a_2^{(2)}} s_3^{(2)} \dots s_T^{(2)}$

- 模型无关的强化学习直接从经验中学习值（value）和策略（policy），而无需构建马尔可夫决策过程模型（MDP）
- 关键步骤：（1）估计值函数；（2）优化策略

值函数估计

- 在基于模型的强化学习（MDP）中，值函数能够通过动态规划计算获得

$$\begin{aligned} V^{\pi}(s) &= \mathbb{E}[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots | s_0 = s, \pi] \\ &= R(s) + \gamma \sum_{s' \in S} P_{s\pi(s)}(s') V^{\pi}(s') \end{aligned}$$

- 在模型无关的强化学习中
  - 我们无法直接获得 P_{sa} 和 R
  - 但是，我们拥有一系列可以用来估计值函数的经验

Episode 1: $s_0^{(1)} \xrightarrow[R(s_0)^{(1)}]{a_0^{(1)}} s_1^{(1)} \xrightarrow[R(s_1)^{(1)}]{a_1^{(1)}} s_2^{(1)} \xrightarrow[R(s_2)^{(1)}]{a_2^{(1)}} s_3^{(1)} \ldots s_T^{(1)}$

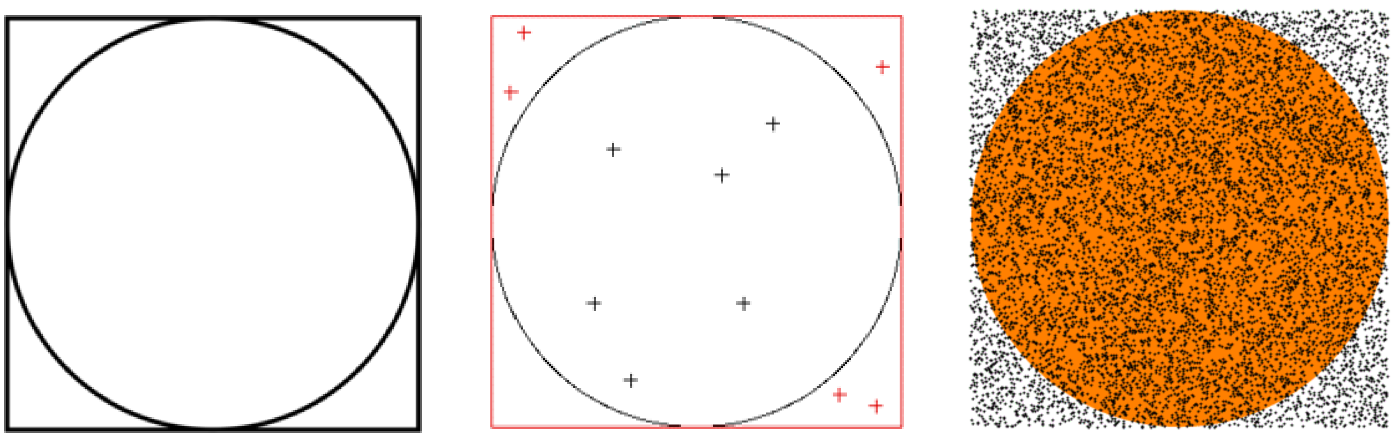
Episode 2: $s_0^{(2)} \xrightarrow[R(s_0)^{(2)}]{a_0^{(2)}} s_1^{(2)} \xrightarrow[R(s_1)^{(2)}]{a_1^{(2)}} s_2^{(2)} \xrightarrow[R(s_2)^{(2)}]{a_2^{(2)}} s_3^{(2)} \ldots s_T^{(2)}$

蒙特卡洛方法

讲师：张伟楠 – 上海交通大学

蒙特卡洛方法

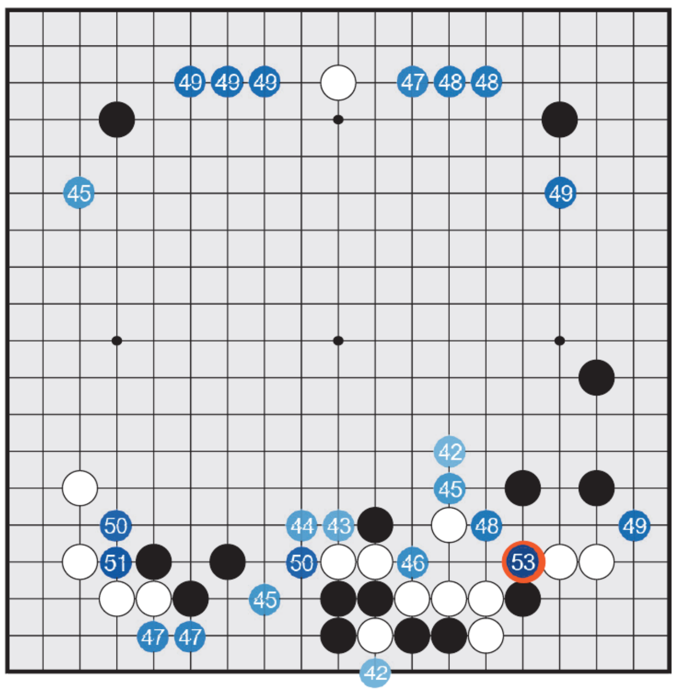
- 蒙特卡洛方法（Monte-Carlo methods）是一类广泛的计算算法。生活中处处都是MC方法。
  - 依赖于重复随机抽样来获得数值结果
- 例如，计算圆的面积

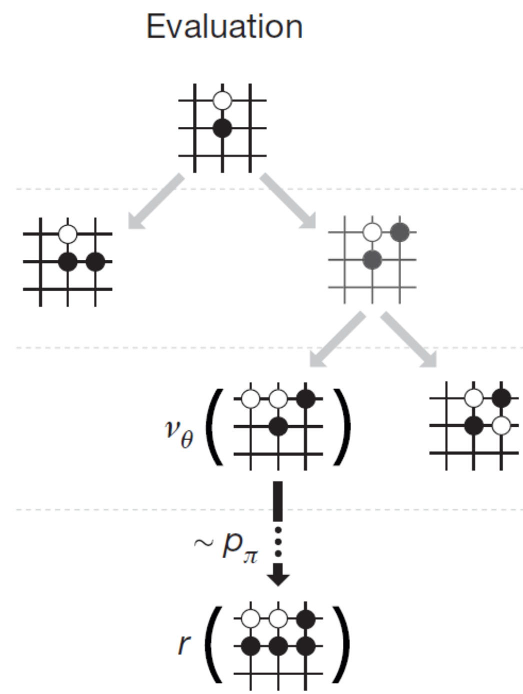


$$\text{Circle Surface} = \text{Square Surface} \times \frac{\#\text{points in circle}}{\#\text{points in total}}$$

蒙特卡洛方法

- 围棋对弈：估计当前状态下的胜率





$$\text{Win Rate}(s) = \frac{\#\text{win simulation cases started from } s}{\#\text{simulation cases started from } s \text{ in total}}$$

蒙特卡洛价值预测

讲师：张伟楠 – 上海交通大学

蒙特卡洛价值估计

- 目标：从策略 π 下的经验片段学习 V^{π}

$$s_0^{(i)} \xrightarrow[R_1^{(i)}]{a_0^{(i)}} s_1^{(i)} \xrightarrow[R_2^{(i)}]{a_1^{(i)}} s_2^{(i)} \xrightarrow[R_3^{(i)}]{a_2^{(i)}} s_3^{(i)} \ldots s_T^{(i)} \sim \pi$$

- 回顾：累计奖励（return）是总折扣奖励

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

- 回顾：值函数（value function）是期望累计奖励

$$\begin{aligned} V^{\pi}(s) &= \mathbb{E}[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots | s_0 = s, \pi] \\ &= \mathbb{E}[G_t | s_t = s, \pi] \\ &\simeq \frac{1}{N} \sum_{i=1}^{N} G_t^{(i)} \end{aligned}$$

- 使用策略π从状态s采样N个片段
- 计算平均累计奖励

- 蒙特卡洛策略评估使用经验均值累计奖励而不是期望累计奖励

蒙特卡洛价值估计

实现

- 使用策略π采样片段

$$s_0^{(i)} \xrightarrow[R_1^{(i)}]{a_0^{(i)}} s_1^{(i)} \xrightarrow[R_2^{(i)}]{a_1^{(i)}} s_2^{(i)} \xrightarrow[R_3^{(i)}]{a_2^{(i)}} s_3^{(i)} \ldots s_T^{(i)} \sim \pi$$

- 在一个片段中的每个时间步长t的状态s都被访问
  - 增量计数器 $N(s) \leftarrow N(s) + 1$
  - 增量总累计奖励 $S(s) \leftarrow S(s) + G_t$
  - 价值被估计为累计奖励的均值 $V(s) = S(s)/N(s)$
  - 由大数定率有

$$V(s) \rightarrow V^{\pi}(s) \text{ as } N(s) \rightarrow \infty$$

增量蒙特卡洛更新

- 每个片段结束后逐步更新$V(s)$
- 对于每个状态S_t和对应累计奖励G_t

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

- 对于非稳定的问题（即，环境会随时间发生变化），我们可以跟踪一个现阶段的平均值（即，不考虑过久之前的片段）

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

蒙特卡洛值估计

思路：
$$V(S_t) \simeq \frac{1}{N}\sum_{i=1}^{N} G_t^{(i)}$$

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

- 蒙特卡洛方法：直接从经验片段进行学习
- 蒙特卡洛是模型无关的：未知马尔可夫决策过程的状态转移/奖励
- 蒙特卡洛从完整的片段中进行学习：没有使用bootstrapping的方法
- 蒙特卡洛采用最简单的思想：值（value） = 平均累计奖励（mean return）
- 注意：只能将蒙特卡洛方法应用于有限长度的马尔可夫决策过程中
  - 即，所有的片段都有终止状态

重要性采样

讲师：张伟楠 – 上海交通大学

重要性采样

- 估计一个不同分布的期望

$$
\begin{aligned}
\mathbb{E}_{x\sim p}[f(x)] &= \int_x p(x)f(x)dx \\
&= \int_x q(x)\frac{p(x)}{q(x)}f(x)dx \\
&= \mathbb{E}_{x\sim q}\left[\frac{p(x)}{q(x)}f(x)\right]
\end{aligned}
$$

- 将每个实例的权重重新分配为 $\beta(x) = \frac{p(x)}{q(x)}$

使用重要性采样的离线策略蒙特卡洛

- 使用策略μ产生的累计奖励评估策略π
- 根据两个策略之间的重要性比率（importance ratio）对累计奖励G_t加权
- 每个片段乘以重要性比率

$$\{s_1, a_1, r_2, s_2, a_2, \dots, s_T\} \sim \mu$$

$$G_t^{\pi/\mu} = \frac{\pi(a_t|s_t)}{\mu(a_t|s_t)} \frac{\pi(a_{t+1}|s_{t+1})}{\mu(a_{t+1}|s_{t+1})} \cdots \frac{\pi(a_T|s_T)}{\mu(a_T|s_T)} G_t$$

使用重要性采样的离线策略蒙特卡洛

- 更新值函数以逼近修正的累计奖励值

$$V(s_t) \leftarrow V(s_t) + \alpha \left(G_t^{\pi/\mu} - V(s_t) \right)$$

无法在π非零而μ为零时使用

重要性采样将显著增大方差（variance）

使用重要性采样的离线策略时序差分

- 使用策略μ产生的时序差分目标评估策略π
- 根据重要性采样对时序差分目标$r + \gamma V(s')$加权
- 仅需要一步来进行重要性采样修正

$$V(s_t) \leftarrow V(s_t) + \alpha \left(\underbrace{\frac{\pi(a_t|s_t)}{\mu(a_t|s_t)}}_{\text{重要性采样修正}} \underbrace{\left(r_{t+1} + \gamma V(s_{t+1})\right)}_{\text{时序差分目标}} - V(s_t) \right)$$

具有比蒙特卡洛重要性采样更低的方差

策略仅需在单步中被近似

时序差分学习

讲师：张伟楠 – 上海交通大学

时序差分学习（Temporal Difference Learning）

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = R_{t+1} + \gamma V(S_{t+1})$$

$$V(S_t) \leftarrow V(S_t) + \alpha\big(\underbrace{R_{t+1}}_{\text{观测值}} + \underbrace{\gamma V(S_{t+1})}_{\text{对未来的猜测}} - V(S_t)\big)$$

- 时序差分方法直接从经验片段中进行学习
- 时序差分是模型无关的
  - 不需要预先获取马尔可夫决策过程的状态转移/奖励
- 通过bootstrapping，时序差分从不完整的片段中学习
- 时序差分更新当前预测值使之接近估计累计奖励（非真实值）

蒙特卡洛 vs. 时序差分（MC vs. TD）

相同的目标：从策略π下的经验片段学习V^{π}

- 增量地进行每次蒙特卡洛过程（MC）
  - 更新值函数$V(S_t)$使之接近准确累计奖励G_t

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

- 最简单的时序差分学习算法（TD）：
  - 更新$V(S_t)$使之接近估计累计奖励$R_{t+1} + \gamma V(S_{t+1})$

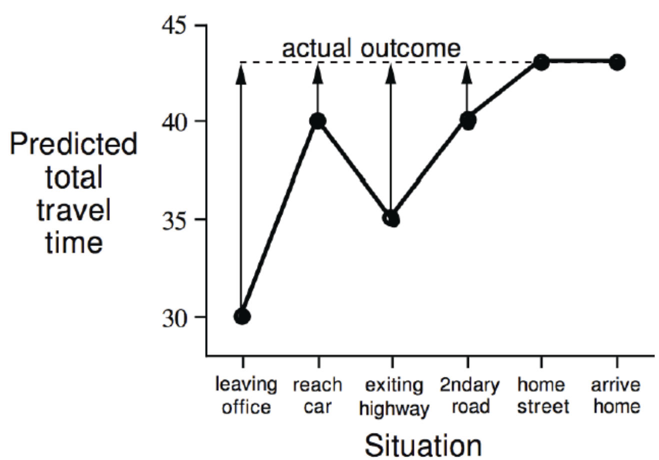
$$V(S_t) \leftarrow V(S_t) + \alpha\big(R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\big)$$

  - 时序差分目标：$R_{t+1} + \gamma V(S_{t+1})$
  - 时序差分误差：$\delta_t = R_{t+1} + \gamma V(S_{t+1}) - V(S_t)$

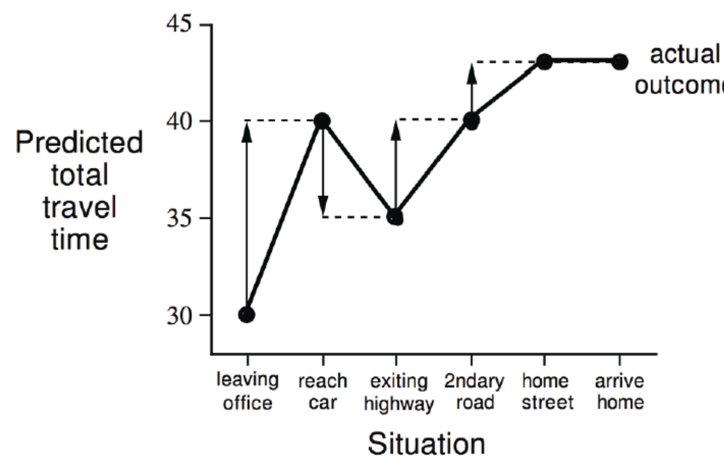
驾车回家的例子（MC vs. TD)

状态	经过的时间 (分钟)	预计所剩时间	预计总时间
离开公司	0	30	30
开始驾车， 下雨	5	35	40
离开高速公路	20	15	35
卡车后跟车	30	10	40
到达家所在街道	40	3	43
直奔家门	43	0	43

Changes recommended by Monte Carlo methods (α=1)



Changes recommended by TD methods (α=1)



蒙特卡洛（MC）和时序差分（TD）的优缺点

- 时序差分：能够在知道最后结果之前进行学习
  - 时序差分能够在每一步之后进行在线学习
  - 蒙特卡洛必须等待片段结束，直到累计奖励已知

- 时序差分：能够无需最后结果地进行学习
  - 时序差分能够从不完整的序列中学习
  - 蒙特卡洛只能从完整序列中学习
  - 时序差分在连续（无终止的）环境下工作
  - 蒙特卡洛只能在片段化的（有终止的）环境下工作

偏差（Bias）/方差（Variance）的权衡

- 累计奖励$G_t = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{T-1} R_T$是$V^{\pi}(S_t)$的无偏估计
- 时序差分真实目标$R_{t+1} + \gamma V^{\pi}(S_{t+1})$是$V^{\pi}(S_t)$的无偏估计
- 时序差分目标$R_{t+1} + \underbrace{\gamma V(S_{t+1})}_{\text{当前估计}}$是$V^{\pi}(S_t)$的有偏估计

- 时序差分目标具有比累计奖励更低的方差
  - 累计奖励——取决于多步随机动作，多步状态转移和多步奖励
  - 时序差分目标——取决于单步随机动作，单步状态转移和单步奖励

蒙特卡洛（MC）和时序差分（TD）的优缺点（2）

MC:

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

蒙特卡洛具有高方差，无偏差

- 良好的收敛性质
  - 使用函数近似时依然如此
- 对初始值不敏感
- 易于理解和使用

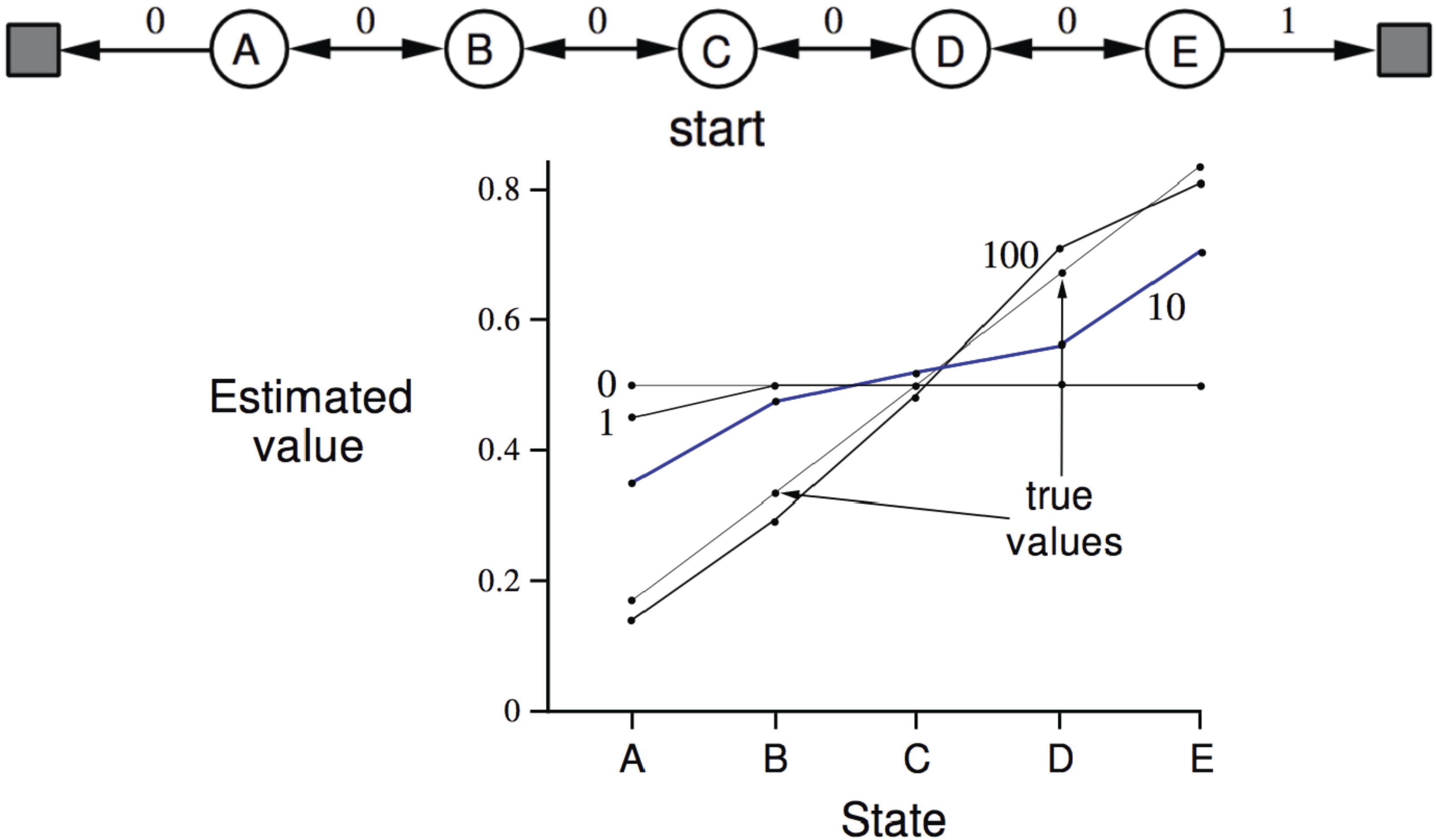
TD:

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

时序差分具有低方差，有偏差

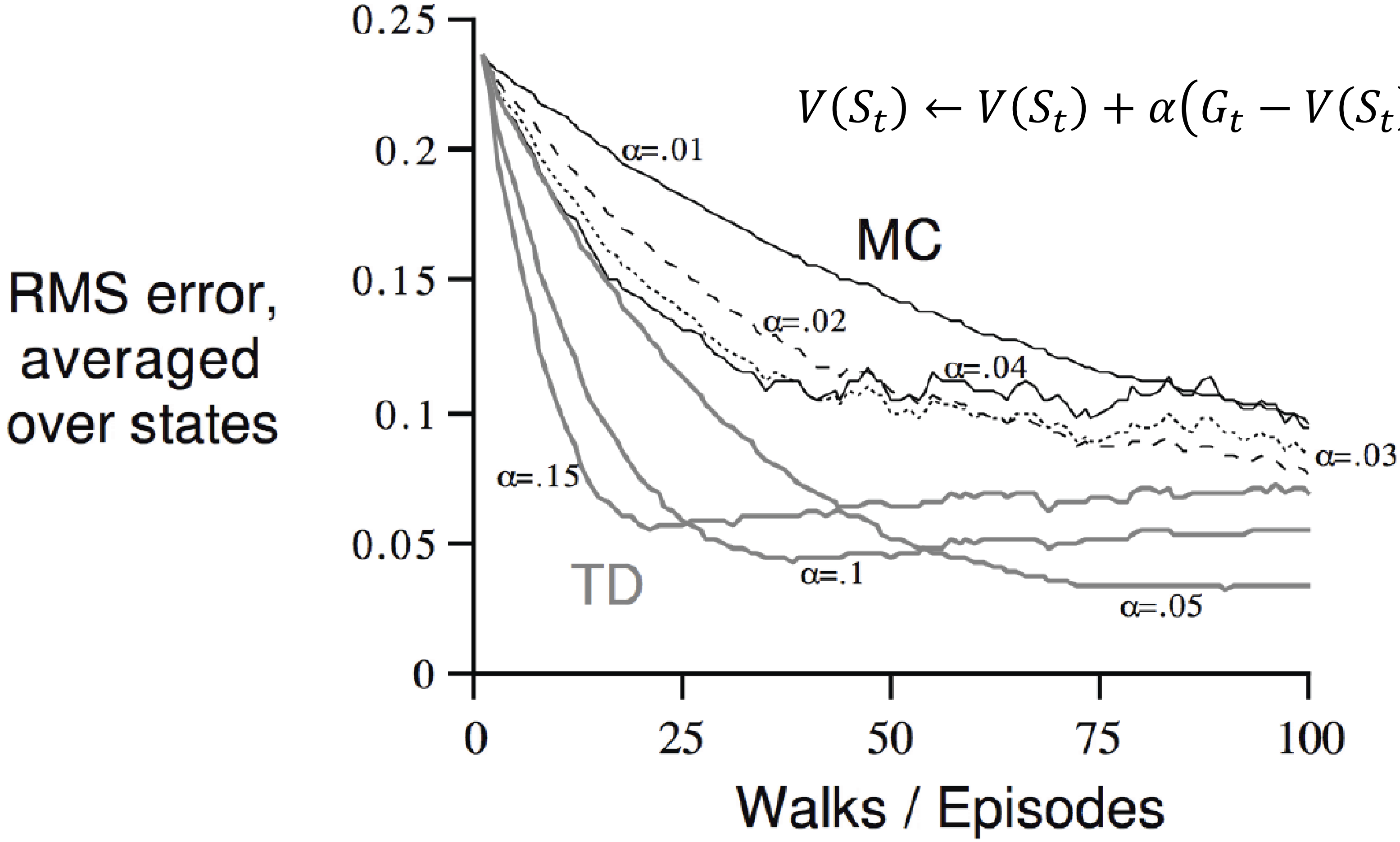
- 通常比蒙特卡洛更加高效
- 时序差分最终收敛到$V^{\pi}(S_t)$
  - 但使用函数近似并不总是如此
- 比蒙特卡洛对初始值更加敏感

随机游走的例子



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

随机游走的例子

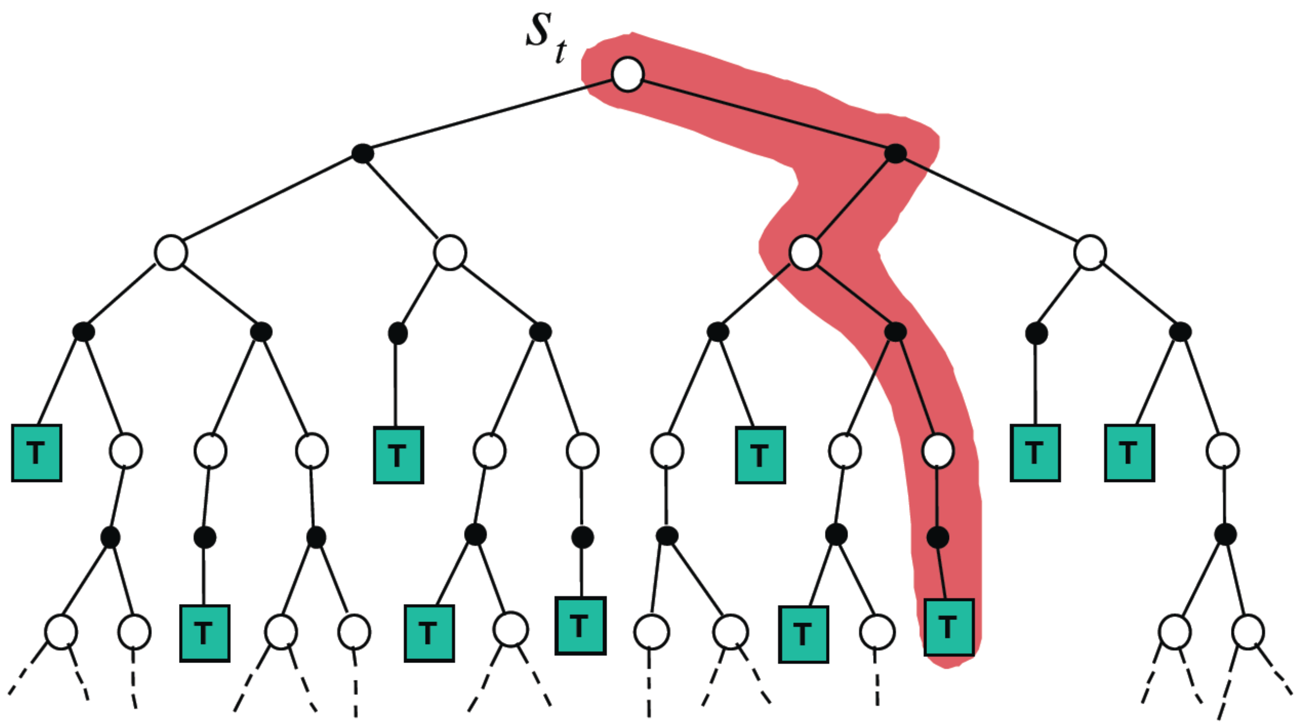


$$V(S_t) \leftarrow V(S_t) + \alpha\big(G_t - V(S_t)\big)$$

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

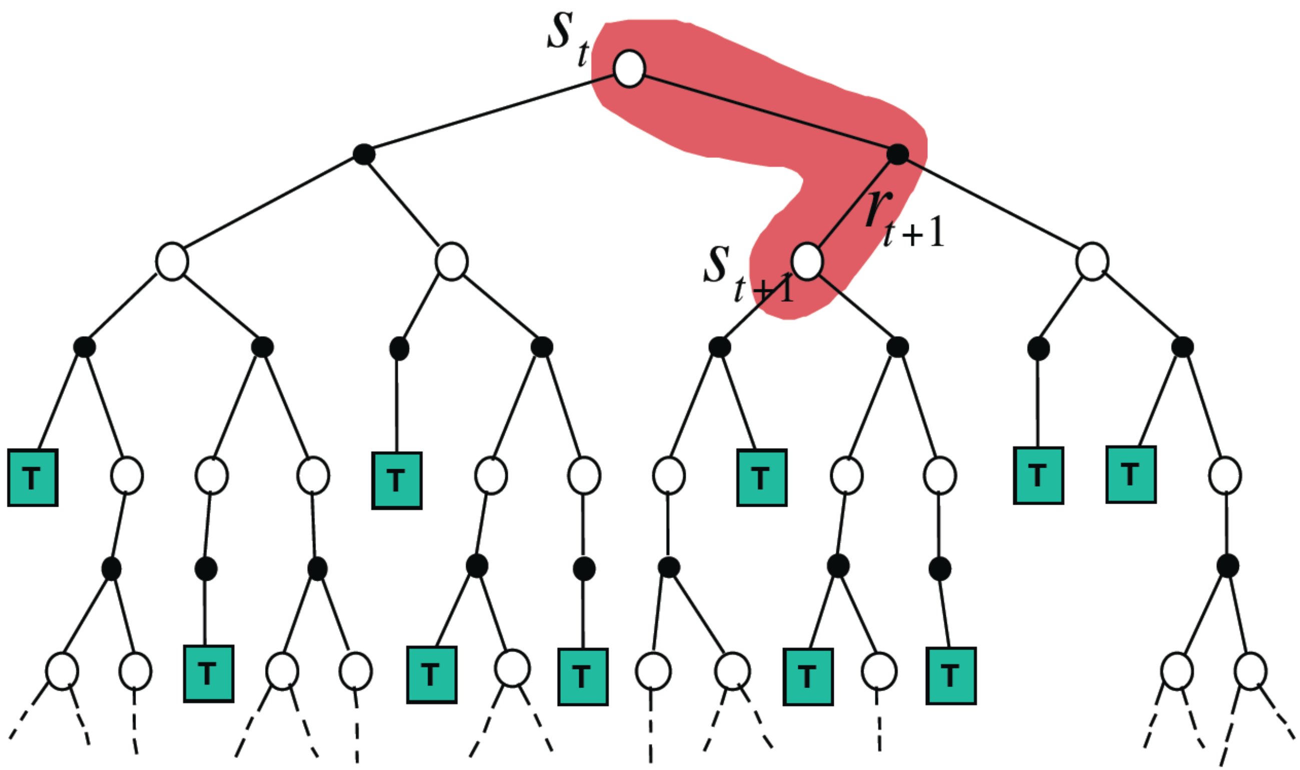
蒙特卡洛反向传播（Backup）

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



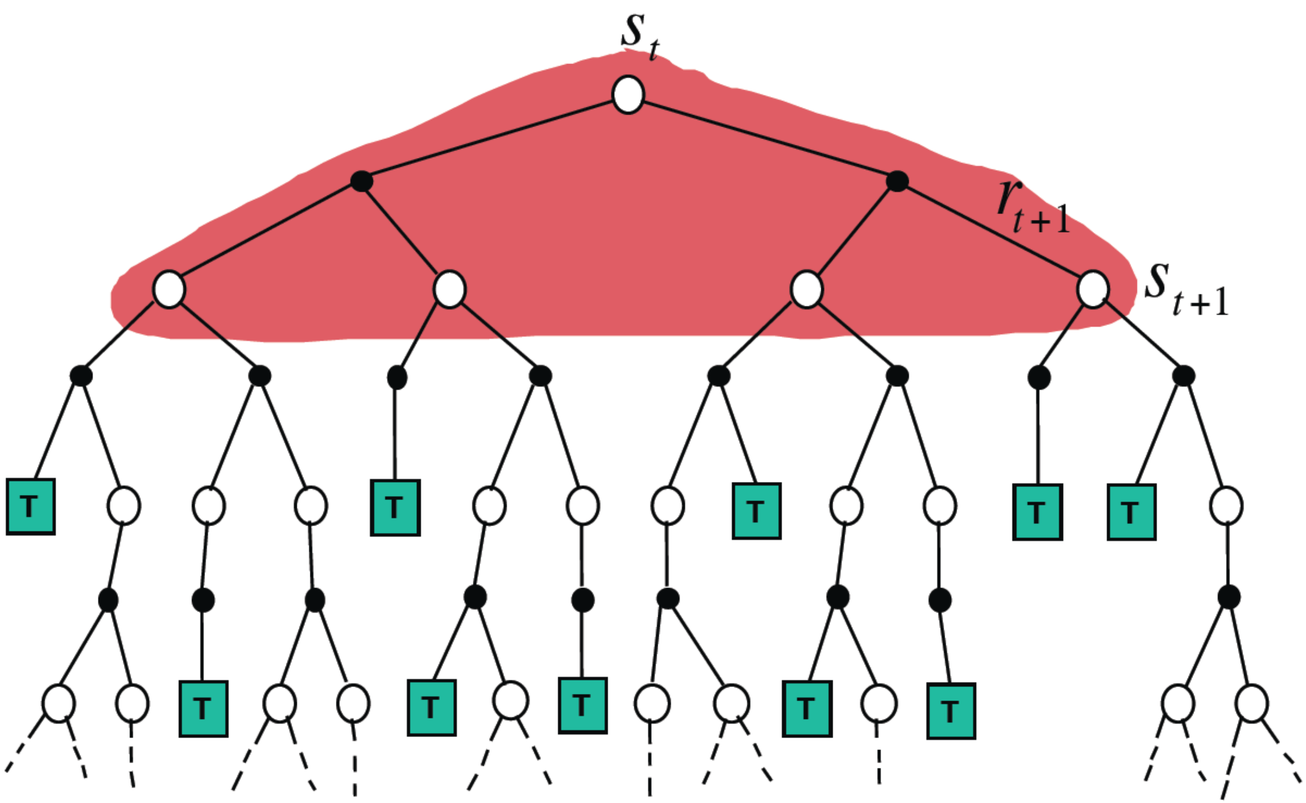
时序差分反向传播（Backup）

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



动态规划反向传播（Backup）

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



多步时序查分学习

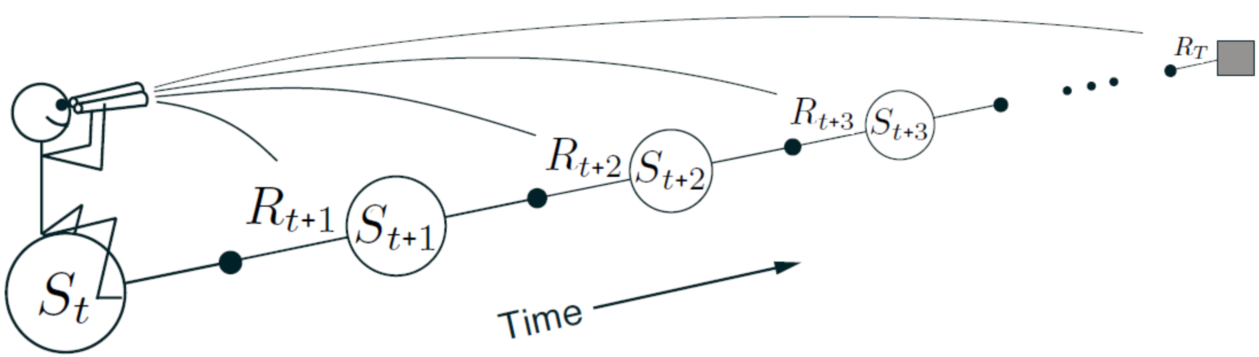
- 对于有时间约束的情况，我们可以跳过n步预测的部分，直接进入模型无关的控制

- 定义n步累计奖励

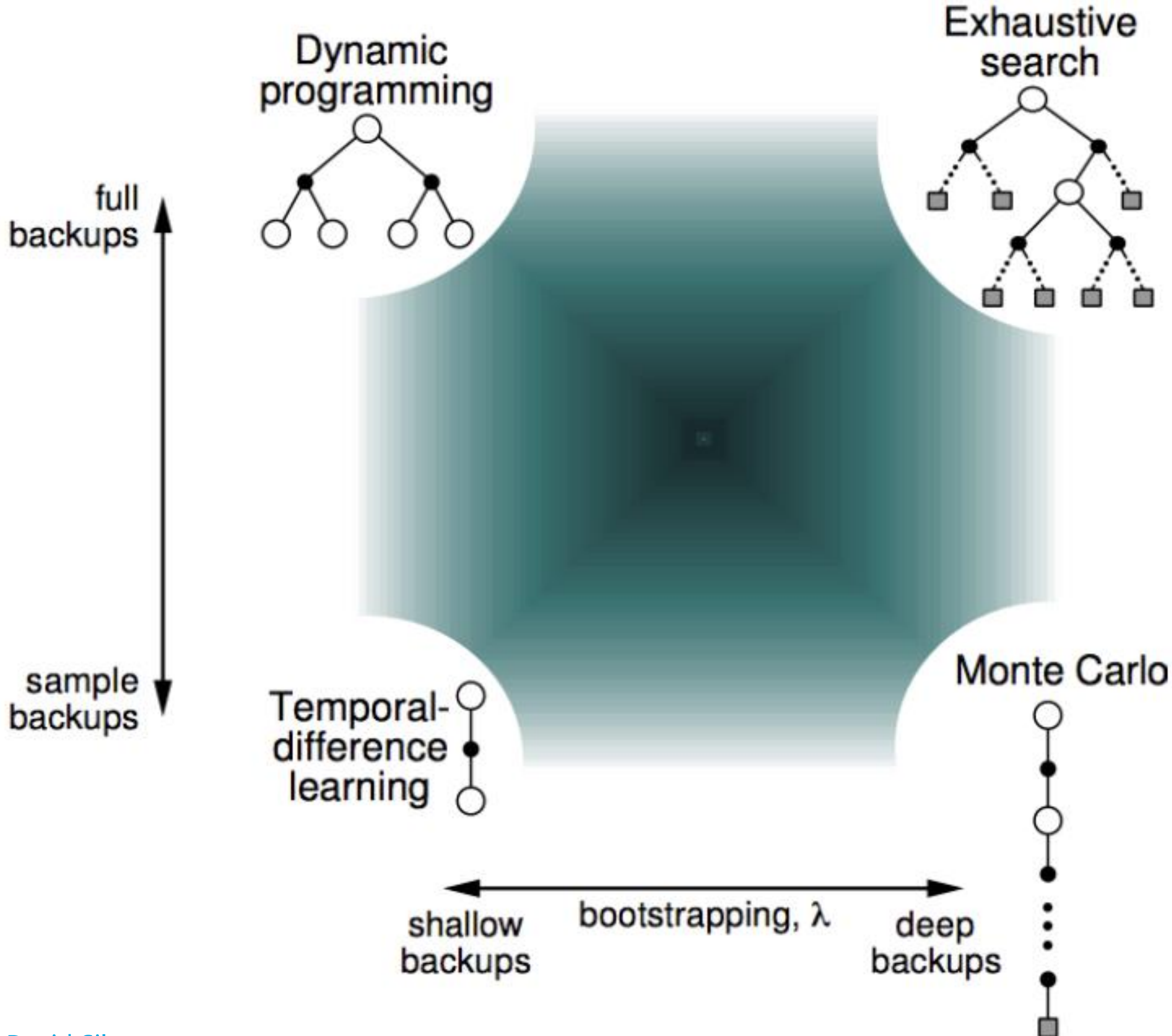
$$G_t^{(n)} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n V(S_{t+n})$$

- n步时序差分学习

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



总览强化学习值函数估计多种方法



Slide credit: David Silver

值函数估计总结

- 无模型的强化学习在黑盒环境下使用
- 要优化智能体策略，首要任务则是精准、高效地估计状态或者(状态、动作)的价值
- 在黑盒环境下，值函数的估计方法主要包括蒙特卡洛方法和时序差分法
- 蒙特卡洛方法通过采样到底的方式直接估计价值函数
- 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
- 实际使用中，时序差分方法更加常见

THANK YOU