第五讲：无模型方法二——时间差分

主讲人 陈达贵

清华大学自动化系

在读硕士

深蓝学院

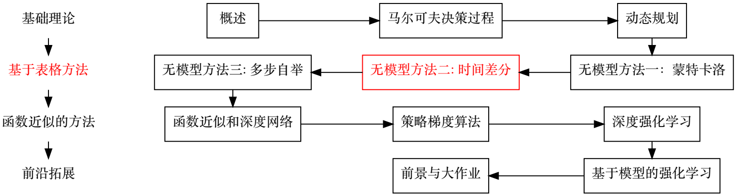
shenlanxueyuan.com

强化学习理论与实践

2019-1-4

目录

章节目录


基础理论
基于表格方法
函数近似的方法
前沿拓展
概述
马尔可夫决策过程
动态规划
无模型方法一：蒙特卡洛
无模型方法二: 时间差分
无模型方法三: 多步自举
函数近似和深度网络
策略梯度算法
深度强化学习
基于模型的强化学习
前景与大作业

本章目录

时间差分算法简介

- 强化学习中最核心也是最新奇的想法
- 混合了 DP 和 MC
  - 和 MC 类似，TD 也从历史经验中学习
  - 和 DP 类似，使用后继状态的值函数更新当前状态的值函数
- 属于无模型方法
  - 未知 $\mathcal{P}, \mathcal{R}$，需要交互，样本备份，需要充分的探索...
- 同时利用了采样和贝尔曼方程
- 可以从不完整的片段中学习（通过自举法）
  - 可同时应用于片段性任务和连续性任务
- 通过估计来更新估计

自举法[1]

- (bootstrapping) 又名拔靴法、自助法
- 通过对样本进行重采样得到的估计总体的方法
- 不用自举法
  - 样本 $\rightarrow$ 总体
- 使用自举法
  - 重采样样本 $\rightarrow$ 样本 (重采样多次可以估计分布)
  - 样本 $\rightarrow$ 总体
- 强化学习中的自举法
  - 利用一个估计去更新另一个估计

[1] https://en.wikipedia.org/wiki/Bootstrapping_(statistics)

目录

目录

时间差分策略评价

- 目的：给定策略 π，求其对应的值函数 v_π
- 增量式 MC
  - 用实际回报值G_t去更新值函数 $V(S_t)$

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

- 时间差分算法 (Temporal-difference，TD)
  - 使用估计的回报值$R_{t+1} + \gamma V(S_{t+1})$去更新值函数 $V(S_t)$ (TD(0))

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

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

时间差分策略评价算法

算法 1 基于表格的 TD(0) 策略评价算法

repeat(对于每个片段)
  初始化状态 S
  repeat(对于片段中的每一步)
    通过 $\pi(\cdot|S)$ 采样 A
    执行动作 A，观测 R, S'
    $V(S) \leftarrow V(S) + \alpha[R + \gamma V(S') - V(S)]$
    $S \leftarrow S'$
  until S 是终止状态
until 收敛

目录

TD 和 DP

- DP 利用了贝尔曼方程去解强化学习问题

$$V(s) \leftarrow \mathbb{E}\left[R + \gamma V(S')|s\right]$$

- TD 也利用了贝尔曼方程，但是做了以下几点改动
  - 全宽备份 → 样本备份：$s \to S$，并去掉期望符号[2]

$$V(S) \leftarrow R + \gamma V(S')$$

  - 增加学习率

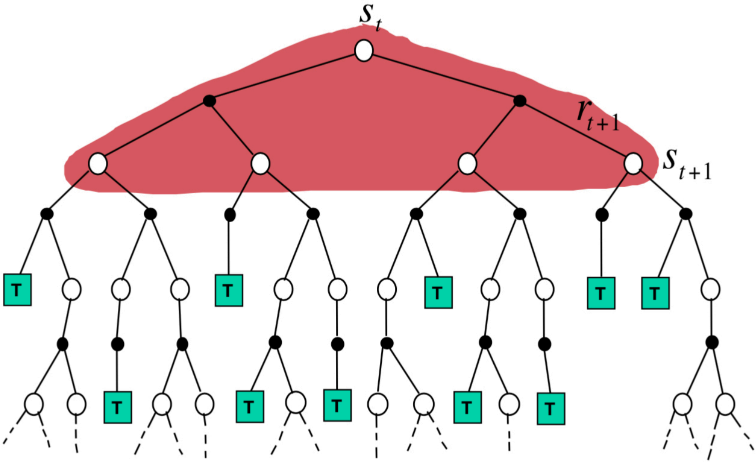
$$V(S) \leftarrow V(S) + \alpha(R + \gamma V(S') - V(S))$$

- 收敛后 $V(S) \stackrel{\mathbb{E}}{=} R + \gamma V(S')$
- 利用 TD 目标和当前值函数的差 (前后时间) 指导学习——时间差分

[2] 求期望有两种手段，一种是利用概率密度函数加权求和（DP），另一种是利用采样去估计 (TD, MC)

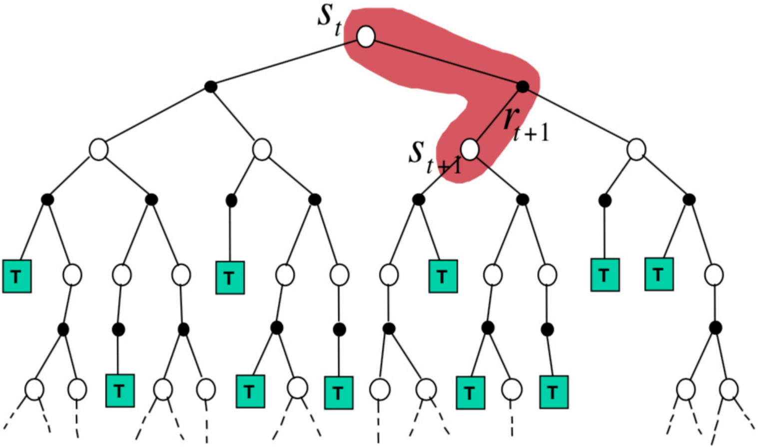
DP 备份

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



TD 备份

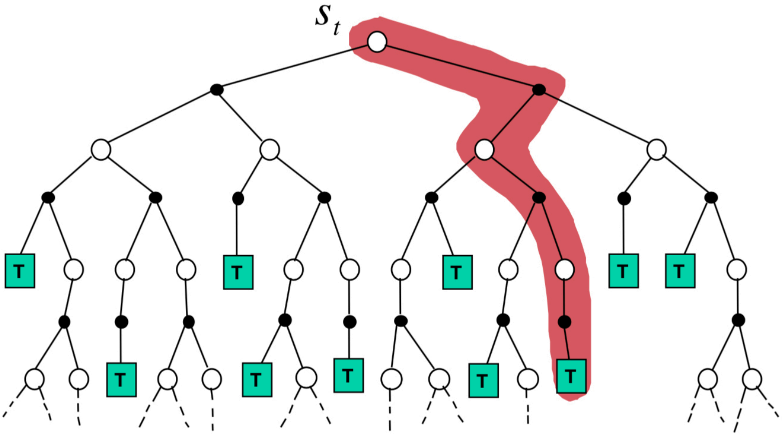
$$V(S_t) \leftarrow V(S_t) + \alpha(R_{t+1} + \gamma V(S_{t+1}) - V(S_t))$$



目录

MC 备份

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

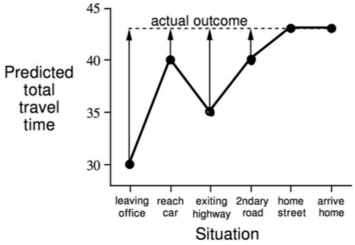


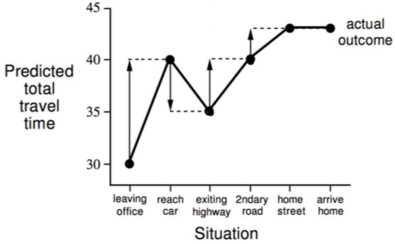
Driving Home Example

State	Elapsed Time (minutes)	Predicted Time to Go	Predicted Total Time
leaving office	0	30	30
reach car, raining	5	35	40
exit highway	20	15	35
behind truck	30	10	40
home street	40	3	43
arrive home	43	0	43

Driving Home Example——MC 和 TD

- 左：MC 方法 $(\alpha = 1)$
- 右：TD 方法 $(\alpha = 1)$



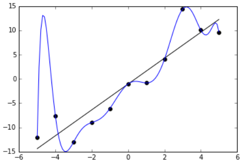


TD 和 MC 的优缺点 (1)

- TD 算法在知道结果之前学习
  - TD 算法在每一步之后都能在线学习
  - MC 算法必须等待回报值得到之后才能学习
- TD 算法即便没有最终结果也能学习
  - TD 算法能够从不完整序列中学习
  - MC 算法仅仅能够从完整序列中学习
  - TD 算法适用于连续性任务和片段性任务
  - MC 算法仅仅适用于片段性任务
- TD 算法有多个驱动力
  - MC 算法只有奖励值作为更新的驱动力
  - TD 算法有奖励值和状态转移作为更新的驱动力

偏差/方差权衡

- 在监督学习中，偏差/方差有另外的理解——欠拟合和过拟合
  - 偏差大 (欠拟合): 预测值和样本之间的差
  - 方差大 (过拟合): 样本值之间的方差, 学出的模型适用性差
- 方差大意味着样本的置信度较差
- 不同的机器学习方法会在两者之间做权衡 (trade-off)



RL 中的偏差/方差权衡

- 回报值 $G_t = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{T-t-1} R_T$ 是值函数 $v_\pi(S_t)$ 的无偏估计
- 真实的 TD 目标值 $R_{t+1} + \gamma v_\pi(S_{t+1})$ 是值函数 $v_\pi(S_t)$ 的无偏估计
- 使用的 TD 目标值 $R_{t+1} + \gamma V(S_{t+1})$ 是值函数 $v_\pi(S_t)$ 的有偏估计
- TD 目标值的方差要远小于回报值
    - 回报值依赖于很多随机变量 $A_t, S_{t+1}, R_{t+1}, A_{t+1}, S_{t+2}, R_{t+2}, \cdots$
    - TD 目标值仅仅依赖于一个随机序列 A_t, S_{t+1}, R_{t+1}

TD 和 MC 的优缺点 (2)

- MC 有高方差，零偏差
    - 收敛性较好 (即使采用函数逼近)
    - 对初始值不太敏感
    - 简单，容易理解和使用
    - 随着样本数量的增加，方差逐渐减少，趋近于 0
- TD 有低方差，和一些偏差
    - 通常比 MC 效率更高
    - 表格法下 TD(0) 收敛到 $v_\pi(s)$(函数逼近时不一定)
    - 对初始值更敏感
    - 随着样本数量的增加，偏差逐渐减少，趋近于 0

批 (batch)MC 和 TD

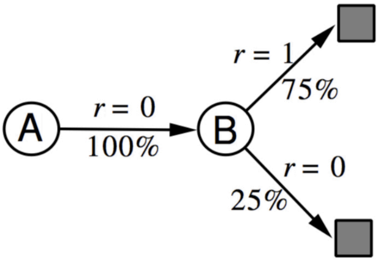
- 随着经验 $\to \infty$, MC 和 TD 收敛: $V(s) \to v_\pi(s)$
- 但是当经验有限时，两者的收敛有什么区别?
    - 比如重复采样了 K 条轨迹
    - 对于每一条轨迹 $k \in [1, K]$ 分别运用 MC 和 TD(0) 算法

AB Example

两个状态 A, B, $\gamma = 1$，采样了 8 条轨迹

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

求 $V(A), V(B)$?



确定性等价估计 (Certainty Equivalence estimate)

- MC 收敛到最小均方误差的解
  - 是对样本回报值的最佳拟合

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

  - 在 AB Example 中， $V(A) = 0$
- TD(0) 收敛到最大似然马尔可夫模型中的解
  - 是对马尔可夫链的最佳拟合，假设了数据是来自 $\mathcal{P}, \mathcal{R}$

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

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

  - 在 AB Example 中， $V(A) = 0 + V(B) = 0.75$
  - 等价于内在动态过程是确定性的估计

MC 和 TD 的优缺点 (3)

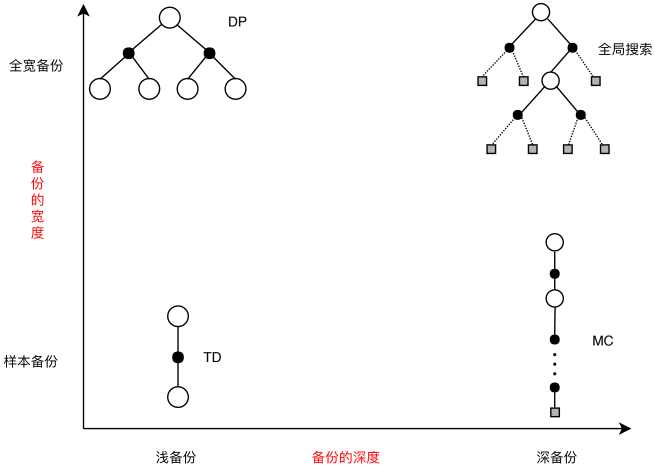
- TD 利用了马尔可夫性
  - 一般来说 TD 在马尔可夫环境中更有效
- MC 没有利用马尔可夫性
  - 一般对非马尔可夫环境更有效

目录

自举和采样

- 自举: 使用随机变量的估计去更新
  - MC 没有自举
  - DP 和 TD 都有自举
- 采样: 通过样本估计期望
  - MC 和 TD 采样
  - DP 不采样

备份



目录

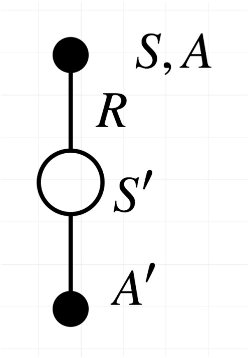
目录

TD 中的策略迭代

- 广义策略迭代
  - **策略评价**: TD 策略评价，$Q = q_\pi$
  - **策略提升**: ε-贪婪策略提升
- TD 优化相比 MC 优化有几点好处
  - 低方差
  - 在线更新 (online)
  - 不完整序列

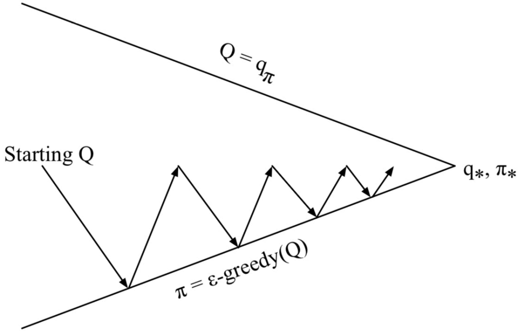
目录

SARSA 备份



$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\left(R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)\right)$$

Sarsa



在每个时间步骤(值迭代)

- **策略评价** Sarsa, $Q \approx q_\pi$
- **策略提升** ε-贪婪策略提升

Sarsa 算法

算法 2 Sarsa 算法

初始化 $Q(s,a), \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$，且 $Q(\text{终止状态}, \cdot) = 0$
repeat(对于每个片段)
  初始化状态 S
  根据 Q 选择一个在 S 处的动作 A，(e.g. 使用 ε-贪婪策略)
  repeat(对于片段中的每一步)
    执行动作 A，观测 R, S'
    根据 Q 选择一个在 S' 处的动作 A'，(e.g. 使用 ε-贪婪策略)
    $Q(S,A) \leftarrow Q(S,A) + \alpha\left(R + \gamma Q(S',A') - Q(S,A)\right)$
    $S \leftarrow S'$; $A \leftarrow A'$
  until S 是终止状态
until 收敛

为什么是在策略的？

- 执行的动作 A 是来自当前 Q 值下的 ε-贪婪策略
- 构建 TD 目标值是的动作 A' 是来自当前 Q 值下的 ε-贪婪策略
- 这两者是同一个策略

Sarsa 收敛性

定理

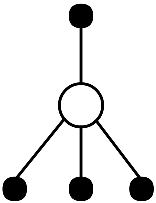
在满足以下条件时，Sarsa 算法收敛到最优的状态动作值函数 $Q(s,a) \to q_*(s,a)$

- 策略序列 $\pi_t(a|s)$ 满足 GLIE
- 步长序列 α_t 是一个 Robbins-Monro 序列

$$\sum_{t=1}^{\infty} \alpha_t = \infty \qquad \sum_{t=1}^{\infty} \alpha_t^2 < \infty$$

- GLIE 保证了
  - 充分的探索
  - 策略最终收敛到贪婪的策略
- Robbins-Monro 保证了
  - 步长足够大，足以克服任意初始值
  - 步长足够小，最终收敛 (常量步长不满足)

期望 Sarsa



$$
\begin{aligned}
Q(S_t, A_t) &\leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \mathbb{E}\left[Q(S_{t+1}, A_{t+1}) | S_{t+1}\right] - Q(S_t, A_t)\right] \\
&\leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \sum_a \pi(a|S_{t+1}) Q(S_{t+1}, a) - Q(S_t, A_t)\right]
\end{aligned}
$$

期望 Sarsa

- 减少了由于 A' 的选择带来的方差
- 在相同更新步数时，期望 Sarsa 比 Sarsa 的通用性更好
- 可以在在策略和离策略中切换
  - 在策略：TD 目标值中的 $R_{t+1} + \gamma \sum_a \pi(a|S_{t+1})Q(S_{t+1}, a)$ 中的策略 π 和采样的策略是同一个策略
  - 离策略：TD 目标值中的 $R_{t+1} + \gamma \sum_a \pi(a|S_{t+1})Q(S_{t+1}, a)$ 中的策略 π 和采样的策略是不同的策略
- 一种特殊情况，TD 目标值中的策略选择贪婪策略，采样的策略选用 ε-贪婪策略——Q 学习

目录

离策略 TD 评价

- 使用行为策略 μ 生成样本，然后评价目标策略 π
- 需要利用重要性采样对 TD 目标值 $R+\gamma V(S')$ 进行加权
- 跟 MC 算法不同，仅仅只需要一次重要性采样率去矫正偏差

$$V(S_t) \leftarrow V(S_t) + \alpha \left(\frac{\pi(A_t|S_t)}{\mu(A_t|S_t)} (R_{t+1} + \gamma V(S_{t+1})) - V(S_t) \right)$$

- 比 MC 的重要性采样方差小得多

对 Q 函数的离策略学习

- 离策略的 TD 算法对比 MC 算法，重要性采样率的因子数减小到一步
- 是否能减少到 0 步？
- 对 Q 函数的离策略学习不需要重要性采样
- 执行的动作 A_t 来自策略 μ，通过 A_t 与环境交互得到样本 S_{t+1}, R_{t+1}，在已知 S_t, A_t 的情况下的重要性采样率为

$$\frac{\mathbb{P}_\pi[S_{t+1}, R_{t+1}|S_t, A_t]}{\mathbb{P}_\mu[S_{t+1}, R_{t+1}|S_t, A_t]} = 1$$

- TD 目标值由之前的 $R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}), A_{t+1} \sim \mu(\cdot, S_t)$ 变成了 $R_{t+1} + \gamma Q(S_{t+1}, A'), A' \sim \pi(\cdot|S_t)$

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left(R_{t+1} + \gamma Q(S_{t+1}, A') - Q(S_t, A_t) \right)$$

- 如果把从 π 采样 A' 改成对 π 求期望得到离策略版的期望 Sarsa

Q 学习

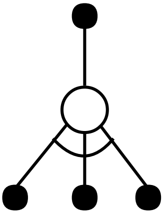
- 目标策略选择 $Q(s, a)$ 下的贪婪策略

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

- 行为策略 μ 选择 $Q(s, a)$ 下的 ε-贪婪策略
- Q 学习的 TD 目标值会得到简化

$$\begin{aligned} & R_{t+1} + \gamma Q(S_{t+1}, A') \\ = & R_{t+1} + \gamma Q(S_{t+1}, \arg\max_{a'} Q(S_{t+1}, a')) \\ = & R_{t+1} + \max_{a'} \gamma Q(S_{t+1}, a') \end{aligned}$$

Q 学习优化算法



$$Q(S,A) \leftarrow Q(S,A) + \alpha \left(R + \gamma \max_{a'} Q(S',a') - Q(S,A) \right)$$

定理

Q 学习优化算法会收敛到最优的状态动作值函数，$Q(s,a) \rightarrow q_*(s,a)$

Q 学习优化算法

算法 3 Q 学习算法

初始化 $Q(s,a), \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$，且 $Q(终止状态, \cdot) = 0$

repeat(对于每个片段)

  初始化状态 S

  repeat(对于片段中的每一步)

    根据 Q 选择一个在 S 处的动作 A，(e.g. 使用 ε-贪婪策略)

    执行动作 A，观测 R, S'

    $Q(S,A) \leftarrow Q(S,A) + \alpha\left(R + \gamma \max_a Q(S',a) - Q(S,A)\right)$

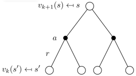
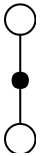
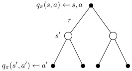
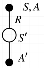
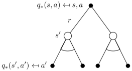
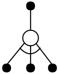
    $S \leftarrow S'$

  until S 是终止状态

until 收敛

目录

DP 和 TD 之间的关系

	全宽备份 (DP)	样本备份 (TD)
对于 $v_\pi(s)$ 的 贝尔曼期望方程	 迭代式策略评价	 TD 策略评价
对于 $q_\pi(s,a)$ 的 贝尔曼期望方程	 策略迭代 (Q)	 Sarsa
对于 $q_*(s,a)$ 的 贝尔曼最优方程	 值迭代 (Q)	 Q 学习

DP 和 TD 之间的关系

全宽备份 (DP)	样本备份 (TD)
迭代式策略评价 $V(s) \leftarrow \mathbb{E}[R + \gamma V(S') \mid s]$	TD 策略评价 $V(S) \overset{\alpha}{\leftarrow} R + \gamma V(S')$
策略迭代 (Q) $Q(s,a) \leftarrow \mathbb{E}[R + \gamma Q(S', A') \mid s, a]$	Sarsa $Q(S,A) \overset{\alpha}{\leftarrow} R + \gamma Q(S', A')$
值迭代 (Q) $Q(s,a) \leftarrow \mathbb{E}[R + \gamma \max_{a' \in \mathcal{A}} Q(S', a') \mid s, a]$	Q 学习 $Q(S,A) \overset{\alpha}{\leftarrow} R + \gamma \max_{a' \in \mathcal{A}} Q(S', a')$

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