第十四讲 强化学习简介

牟克典

2021年6月11日

概要

1 马尔可夫决策过程

- 学习器（Learner）与环境(Environment)进行交互， 以达到特定目标(Goal)
  - 学习器(Agent)对环境施加动作(Action)
  - 获得两类信息:
    - 环境的当前状态（State）
    - 回报(Reward)

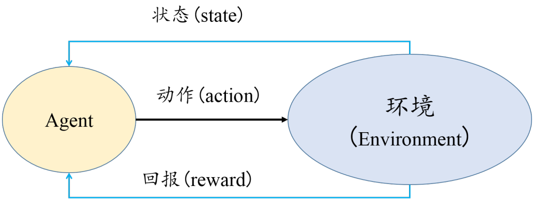

Figure: 强化学习场景

强化学习

- 学习器（Learner）与环境(Environment)进行交互， 以达到特定目标(Goal)
  - 学习器(learner, agent)对环境施加动作(Action)
  - 获得两类信息:
    - 环境的当前状态（State）
    - 回报(Reward)
  - 目标： 希望能获得最大回报
  - 学习任务： 确定能获得最大回报的策略(Policy)
- Agent获得的回报是即时的，环境不提供长期回报
- Agent面临探索-利用(Exploration versus exploitation)困境

- 马尔可夫决策过程模型(Markov decision process (MDP) model)：刻画环境和与环境的交互
- 马尔可夫决策过程MDP由如下定义：
  - 状态集 S,
  - 初始状态 $s_0 \in S$,
  - 动作集A,
  - 目的状态（集）$s' = \delta(s, a)$上的转移概率分布$P[s'|s, a]$,
  - 回报（集）$r' = r(s, a)$上的回报概率分布 $P[r'|s, a]$.
- 对离散时间模型来说， 在决策周期(回合)点 $\{0, \cdots, T\}$采取动作.
- 如果T有限， 则称MDP具有有限决策时域.
- 如果S和 A都有限， 则称是MDP有限的.

- $\Delta(A)$: 动作集A上的概率分布集.
- 策略(Policy)： 从状态集到 $\Delta(A)$的映射 $\pi : S \to \Delta(A)$.
  - 如果对任一$s \in S$， 都有惟一动作$a \in A$使得$\pi(s)(a) = 1$, 则称策略π是确定的.
  - 此时可以用从S到 A的映射来识别π, 并且用$\pi(s)$来表示上述的动作a.
- 上述策略不依赖于时间， 也被称为平稳策略(stationary policy).
- 更一般地， 我们定义非平稳策略(non-stationary policy) 为映射$\pi_t : S \to \Delta(A)$的序列.

Agent依确定策略π沿特定状态序列$s_0, \cdots, s_T$的回报为：

- $T < \infty$: $\sum_{t=0}^{T} r(s_t, \pi(s_t))$;
- $T = \infty$: $\sum_{t=0}^{+\infty} \gamma^t r(s_t, \pi(s_t))$, 其中$\gamma \in [0, 1)$.

策略π在状态$s \in S$的策略值(Policy value)$V_\pi(s)$定义为:

- $T < \infty$: $V_\pi(s) = E_{a_t \sim \pi(s_t)}\left[\sum_{t=0}^{T} r(s_t, a_t) | s_0 = s\right]$;
- $T = \infty$: $V_\pi(s) = E_{a_t \sim \pi(s_t)}\left[\sum_{t=0}^{+\infty} \gamma^t r(s_t, a_t) | s_0 = s\right]$, 其中$\gamma \in [0, 1)$.

策略π^*是最优策略(Optimal policy)是指对任一策略 π和状态$s \in S$, 都有

$$V_{\pi^*}(s) \geq V_\pi(s).$$

状态-动作值函数(State-action value function)

$$
\begin{aligned}
& Q_\pi(s,a) \\
= \; & E[r(s,a)] + E_{a_t\sim\pi(s_t)}\left[\sum_{t=1}^{+\infty}\gamma^t r(s_t,a_t)|s_0=s,a_0=a\right] \\
= \; & E\left[r(s,a)+\gamma V_\pi(s_1)|s_0=s,a_0=a\right].
\end{aligned}
$$

状态-动作值函数与策略值的关系

$$
E_{a\sim\pi(s)}[Q_\pi(s,a)] = V_\pi(s).
$$

Policy Improvement Theorem

对任意两个策略π, π'而言， 都有

$$
\begin{aligned}
& (\forall s \in S, E_{a \sim \pi'(s)}[Q_\pi(s,a)] \geq E_{a \sim \pi(s)}[Q_\pi(s,a)]) \\
\Longrightarrow \quad & (\forall s \in S, V_{\pi'}(s) \geq V_\pi(s)) .
\end{aligned}
$$

Bellman's optimality condition

策略π是最优策略的充要条件是对任意满足 $\pi(s)(a) > 0$ 的 $(s,a) \in S \times A$ 有

$$a \in \operatorname*{argmax}_{a' \in A} Q_\pi(s, a').$$

Existence of an optimal deterministic policy

任何一有限MDP都存在一个最优确定策略.

对最优策略π^*来说,

$$\forall s \in S,\ \pi^*(s) = \operatorname*{argmax}_{a\in A} Q_{\pi^*}(s,a).$$

且

$$V^*(s) = Q_{\pi^*}(s,\pi^*(s)).$$

则

$$\forall s \in S,\ V^*(s) = \max_{a\in A}\left\{E[r(s,a)] + \gamma \sum_{s'\in S} P[s'|s,a]V^*(s')\right\}.$$

Bellman equations

无限决策时域MDP的策略π在状态$s \in S$的策略值$V_\pi(s)$满足如下线性方程:

$$\forall s \in S,\ \ V_\pi(s) = E_{a_1 \sim \pi(s)}[r(s, a_1)] + \gamma \sum_{s' \in S} P[s' | s, \pi(s)] V_\pi(s')$$

Proof.

$$\begin{aligned} V_\pi(s) &= E\left[\sum_{t=0}^{+\infty} \gamma^t r(s_t, \pi(s_t)) | s_0 = s\right] \\ &= E[r(s, \pi(s))] + \gamma E\left[\sum_{t=0}^{+\infty} \gamma^t r(s_{t+1}, \pi(s_{t+1})) | s_0 = s\right] \end{aligned}$$

$$
\begin{aligned}
&= E[r(s,\pi(s))] + \gamma E\left[\sum_{t=0}^{+\infty} \gamma^t r(s_{t+1},\pi(s_{t+1}))|s_0 = s\right] \\
&= E[r(s,\pi(s))] + \gamma E\left[\sum_{t=0}^{+\infty} \gamma^t r(s_{t+1},\pi(s_{t+1}))|s_1 = \delta(s,\pi(s))\right] \\
&= E[r(s,\pi(s))] + \gamma E\left[V_\pi(\delta(s,\pi(s)))\right]. \quad \square
\end{aligned}
$$

进一步， 令

- $\mathbf{P}$为转移概率矩阵，其中$\mathbf{P}_{s,s'} = P[s'|s,\pi(s)]$;
- $\mathbf{V}$为策略值向量， 其中$\mathbf{V}_s = V_\pi(s)$;
- $\mathbf{R}$为回报向量， 其中$\mathbf{R}_s = E[r(s,\pi(s))]$,

则Bellman方程可以写成

$$\mathbf{V} = \mathbf{R} + \gamma \mathbf{P}\mathbf{V}.$$

对有限MDP来说， Bellman方程有惟一解：

$$\mathbf{V}_0 = (\mathbf{I} - \gamma \mathbf{P})^{-1} \mathbf{R}.$$

Proof. 由Bellman方程

$$\mathbf{V} = \mathbf{R} + \gamma \mathbf{P} \mathbf{V}$$

可得

$$(\mathbf{I} - \gamma \mathbf{P}) \mathbf{V} = \mathbf{R}.$$

考虑

$$\| \mathbf{P} \|_\infty = \max_s \sum_{s'} |\mathbf{P}_{ss'}| = 1.$$

由此可得

$$\| \gamma \mathbf{P} \|_\infty = \gamma < 1.$$

故而$\gamma \mathbf{P}$的特征值小于1， 因此$(\mathbf{I} - \gamma \mathbf{P})$可逆。 □

概要

1. 马尔可夫决策过程

2. 规划算法
   - 值迭代算法
   - 策略迭代算法
   - 线性规划

我们假定环境是已知的:

- 对任何 $s, s' \in S$和 $a \in A$来说, $P[s'|s, a]$ 和 $E[r(s, a)]$都已知.

三种规划算法:

- 值迭代算法
- 策略迭代算法
- 线性规划算法

值迭代算法的出发点:

$$\forall s \in S,\ V^*(s) = \max_{a \in A} \left\{ E[r(s,a)] + \gamma \sum_{s' \in S} P[s'|s,a] V^*(s') \right\}.$$

- 对向量$\mathbf{V} \in R^{|S|}$，我们用 $V(s)$来表示$\mathbf{V}$的第s个分量.
- 定义映射$\Phi : R^{|S|} \to R^{|S|}$如下:

$$\forall s \in S,\ [\Phi(V)](s) = \max_{a \in A} \left\{ E[r(s,a)] + \gamma \sum_{s' \in S} P[s'|s,a] V(s') \right\}.$$

- 方程中关于$a \in A$的最大化定义了一个策略π.

我们引进如下矩阵/向量表示

- $\mathbf{P}_\pi$: 其中$(\mathbf{P}_\pi)_{ss'} = P[s'|s,\pi(s)]$;
- $\mathbf{R}_\pi$: 其中$((\mathbf{R}_\pi)_s = E[r(s,\pi(s))]$,

将

$$\forall s \in S,\ [\Phi(V)](s) = \max_{a \in A} \left\{ E[r(s,a)] + \gamma \sum_{s' \in S} P[s'|s,a] V(s') \right\}.$$

重新表示成

$$\Phi(\mathbf{V}) = \max_\pi \left\{ \mathbf{R}_\pi + \gamma \mathbf{P}_\pi \mathbf{V} \right\}.$$

值迭代算法

ValueIteration($\mathbf{V}_0$)

```
V ← V_0
while ‖ V − Φ(V) ‖ ≥ (1−γ)ε/γ do
      V ← Φ(V)
return Φ(V)
```

$$\mathbf{V} \leftarrow \mathbf{V}_0$$
$$\textbf{while} \parallel \mathbf{V} - \Phi(\mathbf{V}) \parallel \geq \frac{(1-\gamma)\epsilon}{\gamma} \textbf{ do}$$
$$\quad \mathbf{V} \leftarrow \Phi(\mathbf{V})$$
$$\textbf{return } \Phi(\mathbf{V})$$

- 这里$\mathbf{V}_0 \in R^{|S|}$是随机选定的迭代初值(初始策略);
- ϵ是逼近阈值.

值迭代算法的收敛性

对任给的初值$\mathbf{V}_0$, 由$\mathbf{V}_{n+1} = \Phi(\mathbf{V}_n)$定义的序列收敛到$\mathbf{V}^*$.

值迭代算法的收敛性证明

Proof. 对任意$s \in S$和$\mathbf{V} \in R^{|S|}$，令 $a^*(s)$是定义$\Phi(\mathbf{V})(s)$的最大化运算所对应的动作，则对任意$s \in S$和$\mathbf{U} \in R^{|S|}$，

$$
\begin{aligned}
& \Phi(\mathbf{V})(s) - \Phi(\mathbf{U})(s) \\
\leq\ & \Phi(\mathbf{V})(s) - \left(E[r(s, a^*(s))] + \gamma \sum_{s' \in S} P[s'|s, a^*(s)] U(s') \right) \\
=\ & \gamma \sum_{s' \in S} P[s'|s, a^*(s)][V(s') - U(s')] \\
\leq\ & \gamma \sum_{s' \in S} P[s'|s, a^*(s)] \parallel \mathbf{V} - \mathbf{U} \parallel_\infty = \gamma \parallel \mathbf{V} - \mathbf{U} \parallel_\infty .
\end{aligned}
$$

类似可以得到

$$
\Phi(\mathbf{U})(s) - \Phi(\mathbf{V})(s) \leq \gamma \parallel \mathbf{V} - \mathbf{U} \parallel_\infty .
$$

由此可以得到

$$\forall s \in S,\ |\Phi(\mathbf{V})(s) - \Phi(\mathbf{U})(s)| \leq \gamma \parallel \mathbf{V} - \mathbf{U} \parallel_\infty .$$

进而

$$\parallel \Phi(\mathbf{V}) - \Phi(\mathbf{U}) \parallel_\infty \leq \gamma \parallel \mathbf{V} - \mathbf{U} \parallel_\infty,$$

即 Φ对$\|\cdot\|_\infty$来说是γ-Lipschitz的.
对最优策略π^*来说,

$$\forall s \in S,\ V^*(s) = \max_{a \in A} \left\{ E[r(s,a)] + \gamma \sum_{s' \in S} P[s'|s,a] V^*(s') \right\},$$

即

$$\mathbf{V}^* = \Phi(\mathbf{V}^*),$$

则对任意$n \in N$,

$$\begin{aligned} \| \mathbf{V}^* - \mathbf{V}^{n+1} \|_\infty = & \| \Phi(\mathbf{V}^*) - \Phi(\mathbf{V}_n) \|_\infty \\ \leq & \gamma \| \mathbf{V}^* - \mathbf{V}_n \|_\infty \\ \leq & \gamma^{n+1} \| \mathbf{V}^* - \mathbf{V}_0 \|_\infty . \end{aligned}$$

考虑到$\gamma \in (0, 1)$, 因此由

$$\mathbf{V}_{n+1} = \Phi(\mathbf{V}_n)$$

定义的序列收敛到$\mathbf{V}^*$. □

关于ϵ最优逼近与迭代次数

$$\begin{aligned}\| \mathbf{V}^* - \mathbf{V}_{n+1} \|_\infty \leq & \| \mathbf{V}^* - \Phi(\mathbf{V}_{n+1}) \|_\infty + \| \Phi(\mathbf{V}_{n+1}) - \mathbf{V}_{n+1} \|_\infty \\ = & \| \mathbf{V}^* - \Phi(\mathbf{V}_{n+1}) \|_\infty + \| \Phi(\mathbf{V}_{n+1}) - \Phi(\mathbf{V}_n) \|_\infty \\ \leq & \gamma \| \mathbf{V}^* - \mathbf{V}_{n+1} \|_\infty + \gamma \| \mathbf{V}_{n+1} - \mathbf{V}_n \|_\infty\end{aligned}$$

进一步可得

$$\| \mathbf{V}^* - \mathbf{V}_{n+1} \|_\infty \leq \frac{\gamma}{1-\gamma} \| \mathbf{V}_{n+1} - \mathbf{V}_n \|_\infty$$

如果 $\| \mathbf{V}_{n+1} - \mathbf{V}_n \|_\infty < \frac{(1-\gamma)}{\gamma}\epsilon$, 则

$$\| \mathbf{V}^* - \mathbf{V}_{n+1} \|_\infty \leq \frac{\gamma}{1-\gamma} \times \frac{(1-\gamma)}{\gamma}\epsilon = \epsilon.$$

另一方面：

$$\| \mathbf{V}_{n+1} - \mathbf{V}_n \|_\infty = \| \Phi(\mathbf{V}_n) - \Phi(\mathbf{V}_{n-1}) \|_\infty$$
$$\leq \gamma \| \mathbf{V}_n - \mathbf{V}_{n-1} \|_\infty \leq \gamma^n \| \Phi(\mathbf{V}_0) - \mathbf{V}_0 \|_\infty .$$

如果n是使得

$$\frac{(1-\gamma)}{\gamma}\epsilon \leq \| \mathbf{V}_{n+1} - \mathbf{V}_n \|_\infty \leq \gamma^n \| \Phi(\mathbf{V}_0) - \mathbf{V}_0 \|_\infty$$

成立的最大整数，则$n \leq O(\log \frac{1}{\epsilon})$.

策略迭代算法

PolicyIteration(π_0)

```
π ← π0
π′ ← NIL
while (π ≠ π′) do
      V ← Vπ
      π′ ← π
      π ← argmaxπ{Rπ + γPπV}
return π
```

- 这里π_0是随机选定的迭代初值(初始策略).
- $\mathbf{V}_\pi$由$(\mathbf{I}-\gamma\mathbf{P}_\pi)\mathbf{V}=\mathbf{R}_\pi$计算.

由策略迭代算法得到的策略值序列$(\mathbf{V}_n)_{n\in N}$, 则对任意$n \in N$, 下列不等式成立:

$$\mathbf{V}_n \le \mathbf{V}_{n+1} \le \mathbf{V}^*.$$

Proof. 令π_{n+1}为第n轮迭代得到的策略改进, 则$(\mathbf{I}-\gamma\mathbf{P}_{\pi_{n+1}})^{-1}$保序, 即对任意$\mathbf{X}, \mathbf{Y} \in R^{|S|}$, 如果$(\mathbf{Y}-\mathbf{X}) \ge \mathbf{0}$, 则

$$(\mathbf{I}-\gamma\mathbf{P}_{\pi_{n+1}})^{-1}(\mathbf{Y}-\mathbf{X}) \ge \mathbf{0}.$$

对策略π_{n+1}来说,

$$\mathbf{R}_{\pi_{n+1}} + \gamma\mathbf{P}_{\pi_{n+1}}\mathbf{V}_n \ge \mathbf{R}_{\pi_n} + \gamma\mathbf{P}_{\pi_n}\mathbf{V}_n = \mathbf{V}_n,$$

则

$$\mathbf{R}_{\pi_{n+1}} \ge (\mathbf{I}-\gamma\mathbf{P}_{\pi_{n+1}})\mathbf{V}_n.$$

$$\mathbf{R}_{\pi_{n+1}} \geq (\mathbf{I} - \gamma \mathbf{P}_{\pi_{n+1}})\mathbf{V}_n.$$

考虑到$(\mathbf{I} - \gamma \mathbf{P}_{\pi_{n+1}})^{-1}$的保序性，则

$$\mathbf{V}_{n+1} = (\mathbf{I} - \gamma \mathbf{P}_{\pi_{n+1}})^{-1}\mathbf{R}_{\pi_{n+1}} \geq \mathbf{V}_n. \quad \square$$

设$(\mathbf{U}_n)_{n\in N}$为值迭代算法得到的策略值序列， 设$(\mathbf{V}_n)_{n\in N}$为策略迭代算法得到的策略值序列. 如果$\mathbf{U}_0 = \mathbf{V}_0$， 则

$$\forall n \in N,\ \mathbf{U}_n \le \mathbf{V}_n \le \mathbf{V}^*.$$

Proof. 我们首先证明Φ的单调性. 设$\mathbf{U}$和 $\mathbf{V}$使得 $\mathbf{U} \le \mathbf{V}$且 π是使得$\Phi(\mathbf{U}) = \mathbf{R}_\pi + \gamma \mathbf{P}_\pi \mathbf{U}$成立的策略, 则

$$\Phi(\mathbf{U}) \le \mathbf{R}_\pi + \gamma \mathbf{P}_\pi \mathbf{V} \le \max_{\pi'}\{\mathbf{R}_{\pi'} + \gamma \mathbf{P}_{\pi'} \mathbf{V}\} = \Phi(\mathbf{V}).$$

我们对n作归纳. 假设$\mathbf{U}_n \le \mathbf{V}_n$， 由$\Phi$的单调性可得

$$\mathbf{U}_{n+1} = \Phi(\mathbf{U}_n) \le \Phi(\mathbf{V}_n) = \max_{\pi}\{\mathbf{R}_\pi + \gamma \mathbf{P}_\pi \mathbf{V}_n\}.$$

令$\pi_{n+1} = \text{argmax}_\pi\{\mathbf{R}_\pi + \gamma\mathbf{P}_\pi\mathbf{V}_n\}$，则

$$\Phi(\mathbf{V}_n) = \mathbf{R}_{\pi_{n+1}} + \gamma\mathbf{P}_{\pi_{n+1}}\mathbf{V}_n \leq \mathbf{R}_{\pi_{n+1}} + \gamma\mathbf{P}_{\pi_{n+1}}\mathbf{V}_{n+1} = \mathbf{V}_{n+1},$$

因此 $\mathbf{U}_{n+1} \leq \mathbf{V}_{n+1}$. $\square$

注：

- 策略迭代次数通常少于值迭代次数;
- 策略迭代每次需要通过$(\mathbf{I} - \gamma\mathbf{P}_\pi)\mathbf{V} = \mathbf{R}_\pi$计算策略值.

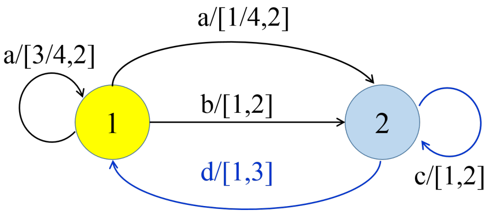

Figure: MDP

- 值迭代算法：

$$\mathbf{V}_{n+1}(1) = \max\{2 + \gamma(\frac{3}{4}\mathbf{V}_n(1) + \frac{1}{4}\mathbf{V}_n(2)), 2 + \gamma\mathbf{V}_n(2)\}$$
$$\mathbf{V}_{n+1}(2) = \max\{3 + \gamma\mathbf{V}_n(1), 2 + \gamma\mathbf{V}_n(2)\}.$$

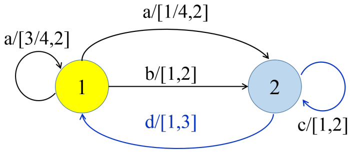

Figure: MDP

策略迭代算法：

- 初始策略π_0: $\pi_0(1) = b$, $\pi_0(2) = c$.
- 相应的策略值满足Bellman方程

$$\begin{cases} \mathbf{V}_{\pi_0}(1) = 2 + \gamma \mathbf{V}_{\pi_0}(2) \\ \mathbf{V}_{\pi_0}(2) = 2 + \gamma \mathbf{V}_{\pi_0}(2) \end{cases}$$

- $\mathbf{V}_{\pi_0}(1) = \mathbf{V}_{\pi_0}(2) = \frac{2}{1-\gamma}$.

回顾最优策略的策略值的表示:

$$\forall s \in S,\ \ V^*(s) = \max_{a \in A} \left\{ E[r(s,a)] + \gamma \sum_{s' \in S} P[s'|s,a] V^*(s') \right\}.$$

对任意一组给定的权值$\{\alpha(s) > 0\}_{s \in S}$, 则上述系列优化问题可以等价表示为如下线性规划问题:

$$\min_{\mathbf{V}} \sum_{s} \alpha(s) V(s)$$

$$s.t. \quad \forall s \in S, \forall a \in A, V(s) \geq E[r(s,a)] + \gamma \sum_{s' \in S} P[s'|s,a] V(s')$$

这个线性规划的行数为$|S||A|$, 列数为$|S|$。

$$\min_{\mathbf{V}} \sum_{s \in S} \alpha(s) V(s)$$

$$s.t. \quad \forall s \in S, \forall a \in A, V(s) \geq E[r(s,a)] + \gamma \sum_{s' \in S} P[s'|s,a] V(s')$$

我们考虑行数更少的对偶问题：

$$
\begin{aligned}
\max_{\mathbf{X}} \quad & \sum_{s\in S,a\in A} E[r(s,a)]x(s,a) \\
s.t. \quad & \forall s \in S, \sum_{a\in A} x(s',a) = \alpha(s') + \gamma \sum_{s\in S,a\in A} P[s'|s,a]x(s',a) \\
& \forall s \in S, \forall a \in A, x(s,a) \geq 0.
\end{aligned}
$$

概要

1. 马尔可夫决策过程
2. 规划算法
   - 值迭代算法
   - 策略迭代算法
   - 线性规划
3. 学习算法
   - TD(0)算法
   - Q-learning算法
   - SARSA算法
   - TD(λ)算法

考虑更一般的情形：刻画环境的MDP模型未知

- $P[s'|s,a]$未知
- $P[r'|s,a]$未知

如何求得最优策略?

- 基于模型的方法
  - 先从当前已经采取的动作得到的即时回报序列学得MDP模型，再求得策略
- 免模型方法(重点)
  - 直接学得动作策略.

定理1

令H从$\mathbb{R}^N$到$\mathbb{R}^N$的映射, $(w_t)_{t\in\mathbb{N}}$是$\mathbb{R}^N$中的随机变量序列, $(\alpha_t)_{t\in\mathbb{N}}$是实数列, $(x_t)_{t\in\mathbb{N}}$是如下定义的序列:

$$\forall s \in [1, N],\ x_{t+1}(s) = x_t(s) + \alpha_t(s)[H(x_t)(s) - x_t(s) + w_t(s)],$$

其中$x_0 \in \mathbb{R}^N$. 定义$\mathcal{F}_t = \{(x_{t'})_{t'\le t}, (w_{t'})_{t'\le t-1}, \alpha_{t'})_{t'\le t}\}$, 并假定下列条件满足:

- $\exists K_1, K_2 \in \mathbb{R}$: $E[w_t^2(s)|\mathcal{F}_t] \le K_1 + K_2 \parallel x_t \parallel^2$, 其中$\parallel \cdot \parallel$为某范数;
- $E[w_t|\mathcal{F}_t] = 0$;
- $\forall s \in [1, N],\ \sum_{t=0}^{\infty} \alpha_t = \infty,\ \sum_{t=0}^{\infty} \alpha_t^2 < \infty$;
- H 是具有不动点x^*的$\parallel \cdot \parallel_\infty$-压缩映射.

则x_t几乎处处收敛到x^*.

回顾计算策略π的策略值的Bellman线性方程

$$\begin{aligned} V_\pi(s) &= E[r(s,\pi(s))] + \gamma \sum_{s' \in S} P[s'|s,\pi(s)] V_\pi(s') \\ &= E_{s'}[r(s,\pi(s)) + \gamma V_\pi(s')|s]. \end{aligned}$$

TD(0)算法主要有如下两步:

- 抽样一个新状态s';
- 更新$V(s)$:

$$\begin{aligned} V(s) &\leftarrow (1-\alpha)V(s) + \alpha[r(s,\pi(s)) + \gamma V(s')] \\ &= V(s) + \alpha[r(s,\pi(s)) + \gamma V(s') - V(s)], \end{aligned}$$

这里α是访问状态s的次数的函数,

$$r(s,\pi(s)) + \gamma V(s') - V(s)$$

为V值的时序差分。

TD(0)()

V $\leftarrow$ **V**$_0$

for $t \leftarrow 0$ **to** T **do**

    $s \leftarrow \mathrm{SelectState}()$

    for each step of epoch t **do**

        $r' \leftarrow \mathrm{Reward}(s, \pi(s))$

        $s' \leftarrow \mathrm{Nextstate}(s, \pi(s))$

        $V(s) \leftarrow (1 - \alpha)V(s) + \alpha[r' + \gamma V(s')]$

        $s \leftarrow s'$

returen V

- 回顾对最优策略π^*来说,

$$\forall s \in S,\ \pi^*(s) = \operatorname*{argmax}_{a \in A} Q^*(s, a).$$

且 $V^*(s) = \max_{a\in} Q^*(s, a)$.

- 我们假定回报函数$r(\cdot, \cdot)$是确定的.
- 对Q^*来说,

$$\begin{aligned} Q^*(s, a) &= E[r(s, a)] + \gamma \sum_{s' \in S} P[s'|s, a] V^*(s') \\ &= E_{s'}[r(s, a) + \gamma \max_{a \in A} Q^*(s', a)] \end{aligned}$$

- 上述计算中涉及的分布未知.

Q-learning算法主要有如下两步:

- 抽样一个新状态s';
- 更新$Q(s,a)$:

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha[r(s,a) + \gamma \max_{a' \in A} Q(s',a')].$$

这里α是访问状态s的次数的函数.

Q-Learning algorithm

$Q \leftarrow Q_0$ * initialization, e.g., $Q_0 = 0$
for $t \leftarrow 0$ **to** t **do**
    $s \leftarrow \mathrm{SelectState}()$
    for each step of epoch t **do**
        $a \leftarrow \mathrm{SelectAction}(\pi, s)$
        $r' \leftarrow \mathrm{Reward}(s, \pi(s))$
        $s' \leftarrow \mathrm{Nextstate}(s, \pi(s))$
        $Q(s,a) \leftarrow Q(s,a) + \alpha[r' + \gamma \max_{a'} Q(s', a') - Q(s,a)]$
        $s \leftarrow s'$
returen Q

Q-learning算法的收敛性

对一有限MDP, 假定$\forall s \in S$和$\forall a \in A$, $\alpha_t(s,a) \in [0,1]$ 且 $\sum_{t=0}^{+\infty} \alpha_t(s,a) = +\infty$, $\sum_{t=0}^{+\infty} \alpha_t^2(s,a) < +\infty$, 则Q-learning算法以概率1收敛到$Q^*$.

Proof. 我们以$(Q_t(s,a))_{t\geq 0}$表示由算法产生的在$(s,a) \in S \times A$点的状态-动作函数值序列, 则

$$Q_{t+1}(s_t,a_t) = Q_t(s_t,a_t) + \alpha[r(s_t,a_t) + \gamma \max_{a'} Q_t(s_{t+1},a') - Q_t(s_t,a_t)].$$

我们定义$s' = \text{Nextstate}(s,a)$, $\alpha_t(s,a)$为0 if $(s,a) \neq (s_t,a_t)$, 否则为$\alpha_t(s_t,a_t)$, 则对任意$s \in S$和$a \in A$, 我们将上式重写为

$$
\begin{aligned}
& Q_{t+1}(s,a) \\
= \;& Q_t(s,a) + \alpha_t(s,a)[r(s,a) + \gamma \max_{a'} Q_t(s',a') - Q_t(s,a)] \\
= \;& Q_t(s,a) + \alpha_t(s,a)[r(s,a) + \gamma E_{u\sim P[\cdot|s,a]}[\max_{a'} Q_t(u,a')] \\
- \;& \gamma E_{u\sim P[\cdot|s,a]}[\max_{a'} Q_t(u,a')] + \gamma \max_{a'} Q_t(s',a') - Q_t(s,a)] \\
= \;& Q_t(s,a) \\
+ \;& \alpha_t(s,a)[r(s,a) + \gamma E_{u\sim P[\cdot|s,a]}[\max_{a'} Q_t(u,a')] - Q_t(s,a)] \\
+ \;& \gamma \alpha_t(s,a)[\max_{a'} Q_t(s',a') - E_{u\sim P[\cdot|s,a]}[\max_{a'} Q_t(u,a')]].
\end{aligned}
$$

令$\mathbf{Q}_t$为$Q_t(s,a)$以为分量的向量，向量 $\mathbf{w}_t$的第s个分量为

$$
w_t(s) = \max_{a'} Q_t(s',a') - E_{u\sim P[\cdot|s,a]}[\max_{a'} Q_t(u,a')]
$$

向量$\mathbf{H}(\mathbf{Q}_t)$的分量$\mathbf{H}(\mathbf{Q}_t)(s,a)$定义为

$$\mathbf{H}(\mathbf{Q}_t)(s,a) = r(s,a) + \gamma E_{u \sim P[\cdot|s,a]}[\max_{a'} Q_t(u,a')].$$

则

$$\forall (s,a) \in S \times A,\ \mathbf{Q}_{t+1}(s,a) \\ = \mathbf{Q}_t(s,a) + \alpha_t(s,a)[\mathbf{H}(\mathbf{Q}_t)(s,a) - \mathbf{Q}_t(s,a) + \gamma \mathbf{w}_t(s)].$$

假设可知

- $\alpha_t(s,a) \in [0,1]$ 且 $\sum_{t=0}^{+\infty} \alpha_t(s,a) = +\infty$, $\sum_{t=0}^{+\infty} \alpha_t^2(s,a) < +\infty$.

- 由$\mathbf{w}_t$的定义可知，$E[\mathbf{w}_t|\mathcal{F}_t] = 0$.
- 对任意$s' \in S$,

$$\begin{aligned}|\mathbf{w}_t(s)| &\leq \max_{a'}|Q_t(s', a')| + |E_{u\sim P[\cdot|s,a]}[\max_{a'} Q_t(u, a')]| \\ &\leq 2\max_{s'}\max_{a'}|Q_t(s', a')| = 2 \parallel \mathbf{Q}_t \parallel_\infty .\end{aligned}$$

故

$$E[\mathbf{w}_t^2(s)|\mathcal{F}_t] \leq 4 \parallel \mathbf{Q}_t \parallel_\infty^2 .$$

- 我们再证**H**是关于$\|\cdot\|_\infty$的γ-压缩.
  对任意$\mathbf{Q}_1,\mathbf{Q}_2\in\mathbb{R}^{|S|\times|A|}$和$(s,a)\in S\times A$,

$$
\begin{aligned}
&|\mathbf{H}(\mathbf{Q}_2)(s,a)-\mathbf{H}(\mathbf{Q}_1)(s,a)|\\
=&|\gamma E_{u\sim P[\cdot|s,a]}[\max_{a'}Q_2(u,a')-\max_{a'}Q_1(u,a')]|\\
\le&\gamma E_{u\sim P[\cdot|s,a]}[|\max_{a'}Q_2(u,a')-\max_{a'}Q_1(u,a')|]\\
\le&\gamma E_{u\sim P[\cdot|s,a]}[\max_{a'}|Q_2(u,a')-Q_1(u,a')|]\\
\le&\gamma\max_{u}\max_{a'}|Q_2(u,a')-Q_1(u,a')|\\
=&\gamma\|\mathbf{Q}_1-\mathbf{Q}_2\|_\infty
\end{aligned}
$$

- **H**是压缩函数，因此有不动点$\mathbf{Q}^*:\mathbf{H}(\mathbf{Q}^*)=\mathbf{Q}^*$.

由定理1可知，Q-learning算法收敛到Q^*. □

关于算法框架的第 5行 “ $a \leftarrow \mathrm{SelectAction}(\pi, s)$ ”：

- 从上述收敛性定理可以知道， 只要策略π可以保证每对(s, a)可以被访问无限次即可, 其他无特别规定.
- 一种可能就是由t 时刻的Q_t决定， 即

$$\mathrm{SelectAction}(\pi, s) = \operatorname*{argmax}_a Q_t(s, a).$$

但这不能保证所有的动作或所有的状态都被访问到.

- 标准的选择是ϵ-greedy policy:
  - 以$1-\epsilon$的概率选择贪心动作(greedy action), 即$\operatorname{argmax}_a Q_t(s, a)$;
  - 以ϵ的概率随机选择动作.

- 另一种选择：Boltzmann-exploration: 当前状态s下， 以如下概率选择动作a:

$$p_t(a|s,Q) = \frac{e^{\frac{Q(s,a)}{\tau_t}}}{\sum_{a' \in A} e^{\frac{Q(s,a')}{\tau_t}}},$$

这里τ_t是温度.

  - $t \to +\infty$时$\tau_t \to 0$, 以保证在t比较大时选择的是贪心动作.
  - 另一方面, τ_t也不能收敛太快， 比如可以选择$\frac{1}{\log(n_t(s))}$.

SARSA algorithm

$Q \leftarrow Q_0$ * initialization, e.g., $Q_0 = 0$

for $t \leftarrow 0$ **to** t **do**

$\quad s \leftarrow \mathrm{SelectState}()$

$\quad a \leftarrow \mathrm{SelectAction}(\pi(Q), s)$

$\quad$ **for** each step of epoch t **do**

$\quad\quad r' \leftarrow \mathrm{Reward}(s, a)$

$\quad\quad s' \leftarrow \mathrm{Nextstate}(s, a)$

$\quad\quad a' \leftarrow \mathrm{SelectAction}(\pi(Q), s')$

$\quad\quad Q(s,a) \leftarrow Q(s,a) + \alpha_t(s,a)[r' + \gamma Q(s',a') - Q(s,a)]$

$\quad\quad s \leftarrow s'$

$\quad\quad a \leftarrow a'$

returen Q

- TD(0) 和 Q-learning 算法都只基于即时回报.
- TD(λ)算法考虑多步的回报.

定义R_t^n如下:

$$R_t^n = r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{n-1} r_{t+n} + \gamma^n V(s_{t+n}).$$

考虑多步回报$\{R_t^n\}$上的几何分布, 定义R_t^λ:

$$R_t^\lambda = (1-\lambda)\sum_{n=0}^{+\infty} \lambda^n R_t^n,$$

这里$\lambda \in [0,1]$. 基于R_t^λ, 将TD(0)算法中的策略值更新改变为:

$$V(s) \leftarrow V(s) + \alpha(R_t^\lambda - V(s)).$$

TD(λ)()

V $\leftarrow$ **V**$_0$
e $\leftarrow$ **0**
for $t \leftarrow 0$ **to** T **do**
  $s \leftarrow \text{SelectState}()$
  for each step of epoch t **do**
    $s' \leftarrow \text{Nextstate}(\pi, s)$
    $\delta \leftarrow r(s, \pi(s)) + \lambda V(s') - V(s)$
    $e(s) \leftarrow \lambda e(s) + 1$
    for $u \in S$ **do**
      if $u \neq s$ **then**
        $e(u) \leftarrow \gamma\lambda e(u)$
      $V(u) \leftarrow V(u) + \alpha\delta e(u)$
    $s \leftarrow s'$
returen **V**

小结

- 马尔可夫决策过程模型
- 规划算法
  - 值迭代算法
  - 策略迭代算法
- 学习算法
  - TD(0)算法
  - Q-learning算法
  - SARSA算法
  - TD(λ)算法