强化学习

Lecturer: 李帅

Note Taker: 邱一航

绪论

sequential decision making → Decision AI

observation → agent → action

↑ reward

environment (state) — change

History = Observation$_1$, Reward$_1$, Action$_1$, O_2, R_2, A_2, ⋯ O_t, R_t

↓

Decide A_t based on History.

State: 描述环境的所有信息. 用于确定将发生的事 (O_t, A_t, R_t)

Assumption. $S_t = f(H_t)$. 认为可以从 H_t 中判断出 S_t.

policy: $\pi: \mathcal{S} \to \Delta(\mathcal{A})$

- deterministic: $a = \pi(s)$
- stochastic: $\pi(a|s) = Pr[A_t = a \mid S_t = s]$

reward: $\mathcal{R}(s, a, s')$ ~ 感知到什么是"好"的 (for long term)

utility of a sequence $[r_0, r_1, \cdots]$

more vs less

now vs later → discounting γ. [e.g. interest]

$(0 < \gamma \leq 1)$

$r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots$ = Utility

Thm. $[a_1, a_2, \cdots] \succ [b_1, b_2, \cdots] \Leftrightarrow [r, a_1, a_2, \cdots] \succ [r, b_1, b_2, \cdots]$

value function predict reward in the future → decide whether the policy is good or not.

$$Q_\pi(s,a) = \mathbb{E}_\pi\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots \mid S_t = s,\ a_t = a\right]$$

$$= \mathbb{E}_\pi\left[R_{t+1} + \gamma Q_\pi(s', a') \mid S_t = s,\ a_t = a\right]$$

model predict the next state

$\mathcal{P}^a_{ss'} = Pr\left[S_{t+1} = s' \mid S_t = s,\ A_t = a\right]$ 转移概率

$\mathcal{R}^a_s = \mathbb{E}\left[R_{t+1} \mid S_t = s,\ A_t = a\right]$ 预测 reward (一些问题中是确定且已知的)

RL

- Based on value: $a^* = \underset{a}{\text{argmax}}\ Q_\pi(s,a)$ → policy: hidden
- Based on policy: explicit form, without value
- Arctor-Critic 学习/模仿

$(s,a) \to Q(s,a)$ large-scale! } Deep RL. (DL as value function)

↑ 近似两者之间的函数 ~ DL.

e.g. NeurIPS 2018 $\begin{bmatrix} s \\ a \end{bmatrix} \to$ DL $\to$ [probabilities of different actions] (DL as a policy)

Actually what we want of Q is

$$Q^*(s,a) = \max_\pi Q_\pi(s,a) = \mathbb{E}\left[R_{t+1} + \gamma Q^*\right] = \mathbb{E}\left[R_{t+1} + \gamma \max_\pi Q_\pi\right]$$

Q-Learning

$$Q_\theta(s_t,a_t) \leftarrow (1-\alpha)\, Q_\theta(s_t,a_t) + \alpha\left(r_t + \gamma \max_{a'} Q_\theta(s_{t+1},a')\right)$$

$$\leftarrow Q_\theta(s_t,a_t) + \alpha\left(r_t + \gamma \max_{a'} Q_\theta(s_{t+1},a') - Q_\theta(s_t,a)\right)$$

θ 参数

$$\mathcal{L}(\theta) = \left(Q_\theta - \underbrace{(r+\gamma Q)}_{\text{target. 也包括 } Q.}\right)^2$$

- Q_θ: evaluation network
- Q: target network. 是 Q_θ 几轮前的结果. 更新慢一些

前沿
- Simulator. 基于模拟模型的强化学习
- goal-oriented <robotics> ~ 长期任务挑战 → 分割成短期. 简单任务 (中间态)
- 模仿学习 (Imitation Learning)
- 多智能体强化学习 ~ 非稳态环境 How to collaborate?

exploitation v.s. exploration

π 当前策略. → 保持已知的最优 (exploitation)

→ 探索更多决策. 但未必最优. (exploration)

多臂老虎机

$\mathcal{A} = \{a^1, a^2, \cdots a^K\}$. $\quad R(r,a^i) = Pr[r \mid a^i]$

AIM: $\max \sum_{t=1}^{T} r_t$. $\quad r_t \sim R(\cdot, a_t)$.

$$Q_n(a^i) = \frac{r_1 + \cdots + r_{n-1}}{n-1} \qquad Q_{n+1}(a^i) = Q_n(a^i) + \frac{1}{n}(r_n - Q_n)$$

regret

$$R(a^i) = Q^* - Q(a^i)$$

$$= \max_{a^i \in \mathcal{A}} Q(a^i) - Q(a^i).$$

$$Q(a^i) = \mathbb{E}_{r \sim Pr[r|a^i]}(r).$$

total regret $\quad \sigma_R = \mathbb{E}_{a\sim\pi}\left[\sum_{t=1}^{T} R(a_t^i)\right]$

$\min \sigma_R \iff \max \mathbb{E}_{a\sim\pi}\left[\sum_{t=1}^{T} Q(a_t^i)\right]$ 一直exploit / 一直explore ~ 线性递增

$\sigma_R = \mathbb{E}\left[\sum_{t=1}^{T} Q(a^*) - \sum_{t=1}^{T} Q(a_t)\right]$

$\frac{\sigma_R}{T} \to 0 \iff Q(a_t) \to Q(a^*)$

$\sigma_R(T) = o(T)$. sublinear

$\lim_{T\to\infty} \sigma_R \geq \log T \sum_{a|\Delta a>0} \frac{\Delta a}{D_{KL}(R(r|a)\|R^*(r|a))} \sim \log T \sum_{a|\Delta_a>0} \frac{1}{\Delta_a}$.

<explore>
- $\log T$ not optimal
- $T - \log T$ optimal

greedy: 看看目前数据里哪个 arm 最好，用它. ⇒ **ε-greedy**: 以 ε 概率选择 $U(1, N)$. 衰减 ε 能指数下降

乐观初始化 reward $\in (0, 1)$. → 初始化: $Q(a^i) >> 1$. 如用 5 或 10

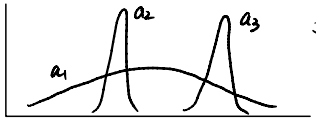
增量更新 $Q(a^i)$, 从 5 降到 $(0, 1)$ 范围. → 未 pull 的 arm 保持很高的 $Q(\cdot)$.

→ 各个 arm 都要被 pull 多次才能回到较正常的估值

$\min\{1, \frac{c|A|}{d^2 t}\}$ need knowledge

⇒ 每个 arm 都被 pull 了较多次数 (exploration).

动作的价值分布



如何平衡 exploitation vs exploration

不确定性大的 $Q(a^i)$ 有探索价值

UCB Hoeffding 不等式. $Pr[\mathbb{E}[x] > \bar{x}_t + u] \leq e^{-2tu^2}$ for $x \in [0,1]$

估计上置信界 $\hat{U}(a^i)$ ⇒ $Q_t(a) \leq \hat{Q}_t(a) + \hat{U}_t(a)$ 以高概率 $(1-p)$ 成立

$a^* = \underset{a\in A}{\operatorname{argmax}}\ \hat{Q}_t(a) + \hat{U}_t(a)$. ↖ exploitation

$e^{-2N_t(a)U_t(a)^2} = p \Rightarrow \hat{U}_t(a) = \sqrt{-\frac{\log p}{2N_t(a)}}$ exploration

Hoeffding

Thompson Sampling 根据每个动作成为最优动作的概率来选择. 根据概率分布采样，选择 $\operatorname{argmax} \tilde{Q}(\cdot)$ ← 采样.

each arm → Init: Gaussian $(0, 1)$

posterior ⇒ Gaussian $\left(\hat{\alpha}_a(t), \frac{1}{1+T_a(t)}\right)$ ← 采样越多越确信

Markov Decision Process

Markov 性 the future is independent of the past when given the present:

状态从历史中获得了充分信息. 不用管历史（S_t 吸收了 H_t 中所有信息）

$$Pr[S_{t+1} | S_t] = Pr[S_{t+1} | S_1, \cdots S_t]$$

$$Pr[S_{t+1} | S_t, A_t]$$

$\pi: \mathcal{H}_t \to \mathcal{A}$ $\xrightarrow{\text{Markov Property}}$ $\pi: \mathcal{S} \to \mathcal{A}$

MDP: $(\mathcal{S}, \mathcal{A}, \{P_{s,a}\}, \gamma, \mathcal{R})$. for RL: P_{sa} and R: unknown

AGENT: a_0, $R(s_0, a_0)$, a_1, $R(s_1, a_1)$, ...

STATE: $s_0 \xrightarrow{P_{s_0,a_0}} s_1 \xrightarrow{P_{s_1,a_1}} s_2 \to \cdots$

TOTAL REWARD: $R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + \cdots$ 多数情况 R 只与 s 有关.

Given an environment (MDP in this case), different strategies samples a different distribution of (s, a).

occupancy measure $\rho^\pi(s,a) = \sum_{t=0}^{T} \gamma^t Pr[s_t = s, a_t = a | s_0, \pi]$

$$\rho^\pi(s) = \sum_{t=0}^{T} \gamma^t Pr[s_t = s | s_0, \pi] = \sum_{t=0}^{T} \gamma^t Pr[s_t = s | s_0, \pi] \sum_{a'} \pi(a_t = a' | s_t = s)$$

$$= \sum_a \sum_{t=0}^{T} \gamma^t Pr[s_t = s, a_t = a | s_0, \pi] = \sum_a \rho^\pi(s,a)$$

Thms. $\rho^{\pi_1} = \rho^{\pi_2}$ iff. $\pi_1 = \pi_2$. $\pi_\rho(a|s) = \frac{\rho(s,a)}{\sum_{a'} \rho(s,a')}$

TOTAL REWARD. $V(\pi) = \mathbb{E}_\pi[R(s,a)]$

已知 $P_{s,a}$ 和 R 求解策略 也可以直接解 $v^\pi = r + \gamma P v^\pi$

$$V^\pi(s) = \mathbb{E}_\pi[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots | s_0 = s]$$

$$= R(s) + \gamma \sum_{s' \in \mathcal{S}} P_{s,\pi(s)}(s') V^\pi(s') = R(s) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,\pi(s))} V^\pi(s')$$

value iteration 要求 $v_1 = f(v_0)$ is γ-Lipschitz (contracting). i.e. $\|f(v_0) - f(v_0')\| \le \gamma \|v_0 - v_0'\|$.

$v_0 \to v^{(1)} = f(v_0) \to \cdots \to v^{(n)} = f(v^{(n-1)})$

$v_0' \to v^{(1)\prime} = f(v_0') \to \cdots \to v^{(n)\prime} = f(v^{(n-1)\prime})$

$\|v^{(n)} - v^{(n)\prime}\| \le \gamma^n \|v_0 - v_0'\|$ (收敛到同一点) $\to 0$.

Bellman Equation $V^*(s) = \max_\pi V^\pi(s) \Rightarrow V^*(\pi) = R(s) + \gamma \max_{a \in \mathcal{A}} \mathbb{E}_{s' \sim P(\cdot|s,a)} V^*(s)$ (still contracting.)

policy iteration $\pi'(s) = \operatorname{argmax}_a R(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} V^\pi(s)$. $\rightsquigarrow$ Is $V^{\pi'} \ge V^\pi$?

$\pi = \pi^*$. $V^{\pi'} = V^\pi$

$\pi \ne \pi^*$. $V^{\pi'} > V^\pi$.

(converge faster)

* 从经验中学习 MDP 模型 → 动态规划 (v/p iter.)

MODEL-FREE RL

(model-based)

通常不会给出状态转移和 reward ⟶ ① 估计模型 MDP ② Sampling (model-free)

用频率逼近概率

Monte-Carlo $V^\pi(s) \approx \frac{1}{N}\sum_{i=1}^{N} G_t^{(i)}$. $\quad G_t = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{T-1} R_T$

增量MC更新 $N(S_t)++$.

$V(S_t) \leftarrow V(S_t) + \frac{1}{N(S_t)}\left(G_t - V(S_t)\right)$. ← 从完整的片段中学习

(只能用于有限长度 MDP)

对非稳定问题 ~ 只追踪现阶段平均值

以 α 衰减.

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

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

评估: 使用 μ 策略产生的奖励评估 π 策略 重要性比率: $\frac{\pi(s)}{\mu(s)}$.

$$\{s_1, a_1, r_1, s_2, a_2, r_2, \cdots s_T\} \sim \mu \Rightarrow G_t^{\pi/\mu} = \frac{\pi(a_t|s_t)}{\mu(a_t|s_t)}\frac{\pi(a_{t+1}|s_{t+1})}{\mu(a_{t+1}|s_{t+1})}\cdots\frac{\pi(a_T|s_T)}{\mu(a_T|s_T)} G_t$$

使用重要性采样的离线MC $V(S_t) \leftarrow V(S_t) + \alpha\left(G_t^{\pi/\mu} - V(S_t)\right)$

时序差分学习 (TD)

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

$V(S_t) \leftarrow V(S_t) + \alpha\left(R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\right)$ ← 直接从经验片段中学习.

observation ↑ prediction for future ↖

$\text{sample} = R(s,\pi(s),s') + \gamma V^\pi(s') \Rightarrow V^\pi(s) \leftarrow (1-\alpha)V^\pi(s) + \alpha\cdot\text{sample}$ ~ sample 为 V^π 的无偏估计.

① Gradient Descent View i.e. $V^\pi(s) \leftarrow V^\pi(s) + \alpha\cdot\left[\text{sample} - V^\pi(s)\right]$

$\text{error}(V) = \frac{1}{2}(\text{sample} - V^\pi(s))^2 \longrightarrow \nabla \text{error}$.

② Exponential Moving Average View. ~ $V_n = \alpha x_n + \alpha(1-\alpha)x_{n-1} + \cdots + \alpha(1-\alpha)^{n-2}x_2 + (1-\alpha)^{n-1}x_1$

"forgets the past".

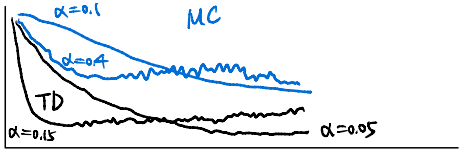
$\alpha = \frac{1}{N}$.

MC $\alpha\uparrow$ ~ new sample would greatly influence V. e.g. $\alpha = 0.04$ → 到25之前快速收敛. (光滑)

25之后剧烈振荡

(α 过大. new sample 的方差影响大)

TD



多步时序差分学习

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

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

γ^n：使 bias 减少 → bias 的主要来源.

How to improve policies?

↓

MC and TD can't.

$P(s,a,s')$: unknown! ⇒ Impossible to get $Q(s,a)$

$\pi(s)$ → we need to observe at state s. ← Fix s? Hard.

(开发者模式(x))

SARSA

~ learn $Q^\pi(\cdot)$.

$$Q^\pi(s,a) = \sum_{s'} P(s'|s,a) \cdot \left[R(s,a,s') + \gamma \cdot Q^\pi(s', \pi(s'))\right]$$

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left(r + \gamma Q(s',a') - Q(s,a)\right).$$

Improvement of π: ε-greedy $\begin{cases} \underset{a}{\operatorname{argmax}}\ Q(s,a) & \text{w.p. } 1-\varepsilon \\ U[\mathcal{A}] & \text{w.p. } \varepsilon. \end{cases}$

on-policy. ~ 使用当前策略进行动作采样. (基于当前策略选择下一个动作) [实时改变]

Q-Learning

AIM: Learn Q^*.

$$Q^*(s,a) = \sum_{s'} P(s'|s,a)\left[R(s,a,s') + \gamma \cdot \max_{a'} Q^*(s',a')\right]$$

{ 目标策略 π → 用以评估 V^π. Q^π
{ 行为策略 μ → 用以收集数据. $\{S_1, a_1, r_2, S_2, a_2, \cdots S_T\} \sim \mu$.

(在环境中尝试)

环境 $a_t \sim \mu(\cdot|s_t)$.

offline $a_{t+1} \sim \pi(\cdot|s_t)$. $Q^\pi(s_t,a_t) \leftarrow Q^\pi(s_t,a_t) + \alpha\left(r_{t+1} + \gamma Q^\pi(s_{t+1}, a_{t+1}) - Q(s_t,a_t)\right)$

用 μ 在环境中撞出来的 $S_t, a_t \to S_{t+1}$ 来更新 Q^π.

π: $\pi(s_{t+1}) = \underset{a'}{\operatorname{argmax}}\ Q^*(s_{t+1}, a')$

μ: ε-greedy policy of $Q^*(s,a)$.

$$r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) = r_{t+1} + \gamma Q\left(s_{t+1}, \underset{a_{t+1}}{\operatorname{argmax}}\ Q(s_{t+1}, a_{t+1})\right)$$

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

用以学习到 Q^*. (而非 Q^π).

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha\left(r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t,a_t)\right)$$

$Q \to Q^*$ eventually.

	完全反向传播（DP）全展开.	采样反向传播（TD）
$V^{\pi}(s)$.	value iteration	时序差分
$Q^{\pi}(s,a)$	Q - policy iteration	SARSA
$Q^{*}(s,a)$	Q - value iteration	Q - Learning

多步时序差分学习（reprise）

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

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

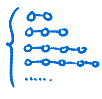
$\alpha \uparrow \sim$ forget more about past. 新采样影响更大.

$n \uparrow \cdots$ Monte Carlo. （能承受的 α 更小）

$n \downarrow \cdots$ TD （能承受的 α 更大）

$\sum \lambda^n = \frac{1}{1-\lambda}$

$$\Rightarrow G_t^{\lambda} = (1-\lambda) \sum_{n=1}^{\infty} \lambda^{n-1} G_t^{(n)} \sim$$ 结合平均不同步数下的奖励.



用 λ 控制一下方差

	variance
$1-\lambda$	小
$(1-\lambda)\lambda$	↓
$(1-\lambda)\lambda^2$	
$(1-\lambda)\lambda^3$	大

$\lambda = 1 \rightarrow$ Monte Carlo. $\lambda = 0 \rightarrow$ TD