Reinforcement Learning

MDP

马尔科夫过程

$< S, P >$

马尔科夫奖励过程

$< S, P, r, \gamma >$

回报

$$G_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k}$$

价值函数

$$\begin{aligned} V(s) &= \mathbb{E}[G_t|S_t = s] \\ &= \mathbb{E}[R_t + \gamma V(S_{t+1})|S_t = s] \end{aligned}$$

贝尔曼方程:

$$= r(s) + \gamma \sum_{s' \in S} P(s'|s)V(s')$$

马尔科夫决策过程

$< S, A, P(s'|s, a), r(s, a), \gamma >$

策略

$$\pi(a|s) = P(A_t = a|S_t = s)$$

状态价值函数

$$V^\pi(s) = \mathbb{E}_\pi[G_t|S_t = s]$$

动作价值函数

$$Q^\pi(s, a) = \mathbb{E}_\pi[G_t|S_t = s, A_t = a]$$

$$\boxed{\begin{aligned} V^\pi &= \sum_{a \in A} \pi(a|s) Q^\pi(s, a) \\ Q^\pi &= r(s, a) + \gamma \sum_{s' \in S} P(s'|s, a) V^\pi(s') \end{aligned}}$$

贝尔曼期望方程

$$\begin{aligned} Q^\pi(s, a) &= \mathbb{E}_\pi[R_t + \gamma Q^\pi(s', a')|S_t = s, A_t = a] \\ &= r(s, a) + \gamma \sum_{s' \in S} P(s'|s, a) \sum_{a' \in A} \pi(a'|s') Q^\pi(s', a') \end{aligned}$$

$$\begin{aligned} V^\pi(s) &= \mathbb{E}_\pi[R_t + \gamma V^\pi(s')|S_t = s] \\ &= \sum_{a \in A} \pi(a|s) \left(r(s, a) + \gamma \sum_{s' \in S} P(s'|s, a) V^\pi(s') \right) \end{aligned}$$

策略卡洛方法

MDP状态价值

用策略在 MDP 上采样很多条序列，计算从这个状态出发的回报再求其期望

$$V^\pi(s) = \mathbb{E}_\pi[G_t|S_t = s] \approx \frac{1}{N} \sum_{i=1}^{N} G_t^{(i)}$$

一条序列只计算一次回报，也就是说条序列第一次出现该状态是计算回报的重积奖励，而后面再出现该状态时，该状态就被忽略了

(1) 使用策略 π 采样若干条序列:

$$s_0^{(1)} \xrightarrow{a_0^{(1)}} r_0^{(1)}, s_1^{(1)} \xrightarrow{a_1^{(1)}} r_1^{(1)}, s_2^{(1)} \xrightarrow{a_2^{(1)}} \cdots \xrightarrow{a_{T-1}^{(1)}} r_{T-1}^{(1)}, s_T^{(1)}$$

(2) 对每一条序列中的每一时间步 t 的状态 s 进行以下操作:

a. 更新状态 s 的计数器 $N(s) \leftarrow N(s) + 1$;

b. 更新状态 s 的总回报 $M(s) \leftarrow M(s) + G_t$;

(3) 每一个状态的价值被估计为回报的期望

$V(s) = M(s)/N(s)$

根据大数定律，当 $N(s) \to \infty$ 时，有 $V(s) \to V^\pi(s)$，所以还有一种增量更新方法:

a. $N(s) \leftarrow N(s) + 1$;

b. $V(s) \leftarrow V(s) + \frac{1}{N(s)}(G - V(s))$;

离线不同的方法

贝尔曼最优方程

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

value-based

DP

策略迭代

策略评估

$$V^{k+1}(s) = \sum_{a \in A} \pi(a|s) \left(r(s, a) + \gamma \sum_{s' \in S} P(s'|s, a) V^k(s') \right)$$

当 $k \to \infty$ 时，序列 $\{V^k\}$ 会收敛到 V^π

实际中，当 $\max_{s \in S} |V^{k+1}(s) - V^k(s)| \leq \epsilon$ 时，结束策略评估

策略提升

$$\begin{aligned} &\text{if} \quad \exists \quad \pi' \\ &\text{s.t.} \quad \forall s \in S, \quad Q^\pi(s, \pi'(s)) \geq V^\pi(s) \\ &\text{then} \quad V^{\pi'}(s) \geq V^\pi(s) \\ &\pi'(s) = \arg\max_a Q^\pi(s, a) = \arg\max_a \{r(s, a) + \gamma \sum_{s'} P(s'|s, a) V^\pi(s')\} \end{aligned}$$

$$\pi^0 \xrightarrow{\text{策略评估}} V^{\pi^0} \xrightarrow{\text{策略提升}} \pi^1 \xrightarrow{\text{策略评估}} V^{\pi^1} \xrightarrow{\text{策略提升}} \pi^2 \xrightarrow{\text{策略评估}} \cdots \xrightarrow{\text{策略提升}} \pi^*$$

价值迭代

$$\begin{aligned} V^{k+1}(s &= \max_{a \in A} \{r(s, a) + \gamma \sum_{s' \in S} P(s'|s, a) V^k(s)\} \\ \text{when} \quad V &^{k+1} = V^k \\ \pi(s) &= \arg\max_a \{r(s, a) + \gamma \sum_{s'} P(s'|s, a) V^{k+1}(s)\} \end{aligned}$$

TD

价值函数的蒙特卡洛

$$V(s_t) \leftarrow V(s_t) + \alpha[G - V(s_t)]$$

$$\boxed{\begin{aligned} V^\pi(s &= \mathbb{E}_\pi[G_t|S_t = s] \\ &= \mathbb{E}_\pi[R_t + \gamma V^\pi(S_{t+1})|S_t = s] \end{aligned}}$$

蒙特卡洛必须等整个序列采集完之后才能计算回报 G_t 时序差分只需要当前步结束即可计算

蒙特卡洛是无偏的，但有较大的方差，时序差分是有偏的，但方差小

价值函数的时序差分

$$V(s_t) \leftarrow V(s_t) + \alpha[r_t + \gamma V(s_{t+1}) - V(s_t)]$$

其中，$r_t + \gamma V(s_{t+1}) - V(s_t)$ 称为时序差分

Sarsa

初始化 $Q(s, a)$

for 序列 $e \leftarrow 1 \to E$ **do:**

得到初始状态 s

用 $\epsilon-$ 贪婪策略根据 Q 选择当前状态 s 下的动作 a

for 时间步 $t \leftarrow 1 \to T$ **do:**

得到环境反馈 r, s'

用 $\epsilon-$ 贪婪策略根据 Q 选择当前状态 s' 下的动作 a'

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

$s \leftarrow s', a \leftarrow a'$

end for

end for

在线策略

行为策略（采样数据表的策略）与

目标策略（用这些数据更新的策略）相同。

如 a 和 a' 都有当前策略 $\epsilon - Greedy(Q)$ 采样得到

多步Sarsa

使用n步的奖励，然后使用之后状态的价值估计

$$G_t = r_t + \gamma r_{t+1} + \cdots + \gamma^n Q(s_{t+n}, a_{t+n})$$

动作价值函数更新

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma r_{t+1} + \cdots + \gamma^n Q(s_{t+n}, a_{t+n}) - Q(s_t, a_t)]$$

结合二者的优势

Q-Learning

初始化 $Q(s, a)$

for 序列 $e \leftarrow 1 \to E$ **do:**

得到初始状态 s

用 $\epsilon-$ 贪婪策略根据 Q 选择当前状态 s 下的动作 a

for 时间步 $t \leftarrow 1 \to T$ **do:**

得到环境反馈 r, s'

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

$s \leftarrow s'$

end for

end for

离线策略

行为策略（采样数据表的策略）与

目标策略（用这些数据更新的策略）不同。

如 a 由行为策略 $\epsilon - Greedy(Q)$ 采样得到，

a' 由当前策略 $\max(Q)$ 采样得到。

policy-based

一般方法

目标函数

$$J(\theta) = E_\beta[V_e(S)]$$

策略梯度定理

$$\frac{\partial J(\theta)}{\partial \theta} = \nabla_\theta J(\theta) = \mathbb{E}_\beta[\mathbb{E}_{A \sim \pi(\cdot|S;\theta)}[\frac{\partial \ln \pi(A|S;\theta)}{\partial \theta} \cdot Q_\pi(S, A)]]$$

随机梯度——策略梯度的无偏估计

$$g(s, a; \theta) \triangleq Q_\pi(s, a) \cdot \nabla_\theta \ln \pi(a|s; \theta)$$

策略网络提升

$$\theta \leftarrow \theta + \beta \cdot g(s, a; \theta)$$

带基线的策略梯度方法

带基线的策略梯度定理——b是不依赖于A的任意函数

$$\nabla_\theta J(\theta) = \mathbb{E}_\beta[\mathbb{E}_{A \sim \pi(\cdot|S;\theta)}[(Q_\pi(S, A) - b) \cdot \nabla_\theta \ln \pi(A|S; \theta)]]$$

随机梯度

$$g_b[(s, a; \theta) = [Q_\pi(S, A) - b] \cdot \nabla_\theta \ln \pi(A|S; \theta)$$

$$\boxed{\begin{aligned} &g_b(s, a; \theta) \text{ 是 } \nabla_\theta J(\theta) \text{ 的无偏估计} \\ &Bias = \mathbb{E}_{S,A}[g_b(s, a; \theta)] - \nabla_\theta J(\theta) = 0 \\ &b \text{ 的取值对 } g_b(s, a; \theta) \text{ 是有影响的} \\ &Var = \mathbb{E}_{S,A}[\|g_b(s, a; \theta) - \nabla_\theta J(\theta)\|^2] \\ &b \text{ 接近 } Q_\pi(s, a) \text{ 关于 } s \text{ 的均值，方差会比较小} \\ &\Rightarrow b = V_\pi(s) \text{ 是很好的基线} \end{aligned}}$$

REINFORCE

折扣回报

$$U_t = \sum_{k=t}^{n} \gamma^{k-t} \cdot R_k$$

动作价值是折扣回报的期望

$$Q_\pi(s_t, a_t) = \mathbb{E}[U_t|S_t = s_t, A_t = a_t]$$

用折扣回报的梯度替换不连近似动作价值

$$\hat{g}(s_T, a_t; \theta) = u_t \cdot \nabla_\theta \ln \pi(a_t|s_t; \theta)$$

策略网络提升

$$\theta_{new} \leftarrow \theta_{now} + \beta \cdot \sum_{t=1}^{n} \gamma^{t-1} \cdot \hat{g}(s_t, a_t; \theta_{now})$$

带基线的REINFORCE

策略网络

折扣回报

$$u_t = \sum_{k=t}^{n} \gamma^{k-t} \cdot r_k$$

基线——价值网络做出的预测

$$\hat{v}_t = v(s_t; \omega)$$

带基线的策略梯度

$$\hat{g}(s_t, a_t; \theta) = (u_t - \hat{v}_t) \cdot \nabla_\theta \ln \pi(a_t|s_t; \theta)$$

梯度上升

$$\theta \leftarrow \theta + \beta \cdot \hat{g}(s_t, a_t; \theta)$$

价值网络

损失函数

$$L(\omega) = \frac{1}{2n} \sum_{t=1}^{n} [v(s_t; \omega) - u_t]^2$$

损失函数的梯度

$$\nabla_\omega L(\omega) = \frac{1}{n} \sum_{t=1}^{n} [v(s_t; \omega) - u_t] \cdot \nabla_\omega v(s_t; \omega)$$

梯度下降

$$\omega \leftarrow \omega - \alpha \cdot \nabla_\omega L(\omega)$$

价值网络只起到基线的作用

Actor-Critic

策略网络

用价值网络近似动作价值

$$\hat{g}(s_t, a_t; \theta) \triangleq q(s, a_t; \omega) \cdot \nabla_\theta \ln \pi(a|s; \theta)$$

策略网络提升

$$\theta \leftarrow \theta + \beta \cdot \hat{g}(s, a_t; \theta)$$

价值网络

TD目标

$$\hat{y}_t \triangleq r_t + \gamma \cdot q(s_{t+1}, a_{t+1}; \omega)$$

损失函数

$$L(\omega) \triangleq \frac{1}{2}[q(s_t, a_t; \omega) - \hat{y}_t]^2$$

价值网络提升策略梯度

损失函数梯度

$$\nabla_\omega L(\omega) = [q(s_t, a_t; \omega) - \hat{y}_t] \cdot \nabla_\omega q(s_t, a_t; \omega)$$

梯度下降

$$\omega \leftarrow \omega - \alpha \cdot \nabla_\omega L(\omega)$$

Advantage Actor-Critic (A2C)

策略网络

贝尔曼公式

$$Q_\pi(s_t, a_t) = \mathbb{E}_{S_{t+1} \sim p(\cdot|s_t, a_t)}[R_t + \gamma \cdot V_\pi(S_{t+1})]$$

优势函数(Advantage function)

$$Q_\pi(s, a) - V_\pi(s)$$

近似策略梯度

$$\begin{aligned} g(s_t, a_t; \theta) &= [Q_\pi(s_t, a_t) - V_\pi(s_t)] \cdot \nabla_\theta \ln \pi(a_t|s_t; \theta) \\ &= [\mathbb{E}_{S_{t+1}}[R_t + \gamma \cdot V_\pi(S_{t+1})] - V_\pi(s_t)] \cdot \nabla_\theta \ln \pi(a_t|s_t; \theta) \end{aligned}$$

蒙特卡洛近似

$$g_t \triangleq [r_t + \gamma \cdot V_\pi(s_{t+1}) - V_\pi(s_t)] \cdot \nabla_\theta \ln \pi(a_t|s_t; \theta)$$

用价值网络 $v(s, \omega)$ 替换状态价值函数 $V_\pi(s)$

$$\hat{g}(s_t, a_t; \theta) \triangleq [r_t + \gamma \cdot v_\pi(s_{t+1}; \omega)] - v_\pi(s_t; \omega)] \cdot \nabla_\theta \ln \pi(a_t|s_t; \theta)$$

策略网络提升

$$\theta \leftarrow \theta + \beta \cdot \hat{g}(s, a; \theta)$$

价值网络

贝尔曼公式

$$V_\pi(s_t) = \mathbb{E}_{A \sim \pi(\cdot|s_t; \theta)}[\mathbb{E}_{S_{t+1} \sim p(\cdot|s_t, A_t)}[R_t + \gamma \cdot V_\pi(S_{t+1})]]$$

TD目标

$$\hat{y}_t \triangleq r_t + \gamma \cdot v(s_{t+1}; \omega)$$

损失函数

$$L(\omega) \triangleq \frac{1}{2}[v(s_t; \omega) - \hat{y}_t]^2$$

损失函数的梯度

$$\nabla_\omega L(\omega) = [v(s_t; \omega) - \hat{y}_t] \cdot \nabla_\omega v(s_t; \omega)$$

梯度下降

$$\omega \leftarrow \omega - \alpha \cdot \nabla_\omega L(\omega)$$

TRPO

策略目标

Multi-agent

完全合作关系

multi-agent cooperative A2C(MAC-A2C)

策略网络

$$\pi(A^1|S; \theta')$$

动作A的概率密度函数

$$\pi(A|S; \theta^1, \cdots, \theta^m) \triangleq \pi(A^1|S; \theta^1) \times \cdots \times \pi(A^m|S; \theta^m)$$

合作关系 MARL 的策略梯度定理

$$\nabla_\theta J(\theta^1, \cdots, \theta^m) = \mathbb{E}_{S,A}[(Q_\pi(S, A) - b) \cdot \nabla_\theta \ln \pi(A^1|S; \theta')]$$

随机梯度——策略梯度的无偏估计

$$g'(s, a; \theta') \triangleq (Q_\pi(s, a) - V_\pi(s)) \cdot \nabla_\theta \ln \pi(a^1|s; \theta')$$

用 $r_t + \gamma \cdot v(s_{t+1}; \omega)$ 近似 $Q_\pi(s_t, a_t)$，

用 $v(s_t; \omega)$ 近似 $V_\pi(s_t)$

$$\hat{g}'(s_t, a_t^1; \theta') \triangleq (r_t + \gamma \cdot v(s_{t+1}; \omega) - v(s_t; \omega)) \cdot \nabla_\theta \ln \pi(a_t^1|s_t; \theta')$$

策略网络提升

$$\theta' \leftarrow \theta' + \beta \cdot \hat{g}'(s_t, a_t^1; \theta')$$

价值网络

参数共享 不能独立训练大模型，需要所有智能体的数据

观测

$$s = [o_1, \cdots, o_m]$$

TD目标

$$\hat{y}_t \triangleq r_t + \gamma \cdot v(s_{t+1}; \omega)$$

损失函数

$$L(\omega) \triangleq \frac{1}{2}[v(s_t; \omega) - \hat{y}_t]^2$$

损失函数的梯度

$$\nabla_\omega L(\omega) = [v(s_t; \omega) - \hat{y}_t] \cdot \nabla_\omega v(s_t; \omega)$$

梯度下降

$$\omega \leftarrow \omega - \alpha \cdot \nabla_\omega L(\omega)$$