

ch1: Markov Decision Process

Static Concepts:

state(S) - policy(π) - action(A) - model(p) - state(S')

reward(R') from transition

return(G) from trajectory

value(V+Q)

策略与价值:

①reward、return、value。

策略与价值一一对应，价值用来评估一个策略的好坏。

②价值比较，即策略比较，引出策略改进定理。

③强化学习的终极目标，求取最优策略，

最优策略不唯一，最优价值唯一。

最优动作价值，意味着选取这个动作，未来回报的期望最大。

④奖励r线性变换，V+Q跟随线性变换，改变了最优价值，不一定改变greedy最优策略。

⑤迭代时，最优策略可能已经稳定了，但是对应的最优价值还没稳定。

⑥从终止状态反向迭代，更新价值，速度更快。但是哪里是终止状态？上帝视角了。

$$p(s', r|s, a) = \Pr S_t = s', R_t = r | S_{t-1} = s, A_{t-1} = a$$

$$\sum_{s' \in S} \sum_{r \in R} p(s', r|s, a) = 1$$

$$p(s'|s, a) = \sum_{r \in R} p(s', r|s, a)$$

$$r(s, a) = \sum_{s' \in S} \sum_{r \in R} (p(s', r|s, a) * r)$$

ch2: Bellman Equations

Static Relationship

实质：描述状态值之间的静态关系（单项形式、矩阵形式）

求解：（矩阵求逆、数值迭代） — （policy-evaluation）

$$\begin{aligned}
 v_{\pi}(s) &= E_{\pi}[G_t | S_t = s] && (Definition) \\
 &= E_{\pi}[R_{t+1} + \gamma G_{t+1} | S_t = s] && (TD - 0) \\
 &= E_{\pi}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots | S_t = s] && (TD - n) (TD - \infty = MC) \\
 &= \sum_{a \in A} \pi(a|s) \sum_{s', r} p(s', r | s, a) * [r + \gamma E_{\pi}[G_{t+1} | S_{t+1} = s']] \\
 &= \sum_{a \in A} \pi(a|s) \sum_{s', r} p(s', r | s, a) * [r + \gamma v_{\pi}(s')] && (BEs) \\
 &= \sum_{a \in A} \pi(a|s) * q_{\pi}(s, a)
 \end{aligned}$$

$$\begin{aligned}
 q_{\pi}(s, a) &= E_{\pi}[G_t | S_t = s, A_t = a] && (Definition) \\
 &= E_{\pi}[R_{t+1} + \gamma G_{t+1} | S_t = s, A_t = a] && (TD - 0) \\
 &= E_{\pi}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots | S_t = s, A_t = a] && (TD - n) (TD - \infty = MC) \\
 &= \sum_{s', r} p(s', r | s, a) * [r + \gamma E_{\pi}[G_{t+1} | S_{t+1} = s']] \\
 &= \sum_{s', r} p(s', r | s, a) * [r + \gamma v_{\pi}(s')] \\
 &= \sum_{s', r} p(s', r | s, a) * [r + \gamma \sum_{a' \in A} \pi(a' | s') * q_{\pi}(s', a')] && (BEs)
 \end{aligned}$$

Policy-Comparison:

$$\pi' \geq \pi \quad \longleftrightarrow \quad v_{\pi'}(s) \geq v_{\pi}(s) \quad \forall s \in S$$

Policy-Improvement:

$$E_{\pi'}[q_{\pi}(s, \pi'(s))] \geq v_{\pi}(s) = E_{\pi}[q_{\pi}(s, \pi(s))] \quad \forall s \in S$$

Bellman Optimal Equations:

$$\begin{aligned} v_*(s) &= \max_{\pi} v_{\pi}(s) && (Definition) \\ &= \max_{\pi} (r_{\pi} + \gamma P_{\pi} v) && (BOEs) \\ &= \max_{a \in A} q_{\pi^*}(s, a) && \forall s \in S \end{aligned}$$

Contraction Mapping Theorem (迭代收敛至，唯一的不动点)
贝尔曼最优方程的迭代收缩过程，即是value iteration算法

ch3: Dynamic Programming

理解:

Model-based. Dynamics with Model p .

①已知模型 p , 给定策略 π , 解BEs, 得到价值 V 。

两种解法: 矩阵求逆、数值迭代。

(1) Policy Iteration:

Policy Evaluation: (Matrix)

$$v_{\pi_k} = r_{\pi_k} + \gamma P_{\pi_k} v_{\pi_k}$$

Policy Improvement:

$$\pi_{k+1} = \arg \max_{\pi} (r_{\pi} + \gamma P_{\pi} v_k)$$

(2) Value Iteration:

$$v_{k+1} = \max_{\pi} (r_{\pi} + \gamma P_{\pi} v_k)$$

Policy Update:

$$\pi_{k+1} = \arg \max_{\pi} (r_{\pi} + \gamma P_{\pi} v_k)$$

Value Update:

$$v_{k+1} = r_{\pi_{k+1}} + \gamma P_{\pi_{k+1}} v_k$$

(3) Turncated Iteration:

值迭代有限次数 (介于1次与无穷次之间) ;

值也未稳定, 就进行策略改进。

ch4: Monte Carlo

Sample:

$$\begin{aligned}v_{\pi}(s) &= E_{\pi}[G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots | S_t = s] & (TD- \text{ or } MC) \\q_{\pi}(s, a) &= E_{\pi}[G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots | S_t = s, A_t = a] & (TD- \text{ or } MC)\end{aligned}$$

理解:

Model-free. Dynamics with Trajectory.

- ①采样进行估计，是依据概率论的大数定理。
- ②episode长度（探索半径是否覆盖有价值的终点？）对估值影响，最优价值是否反向传播。
- ③估计的更新方式，非增长式（等着一起算）和增长式（来一个算一个）。
- ④epsilon关乎采样策略的探索性和最优性，
e大则探索性强、最优性弱，e小则探索性弱、最优性强，
- ⑤如果epsilon大到一定程度，可能会导致epsilon-greedy与最优greedy不一致。

(1) MC-Basic

二次循环，遍历所有(s, a)；在某个策略下，每对足够采样，非增长式估计相应Q值。
策略相应的，一套稳定Q值下，策略改进。
迭代之。

(2) MC-Exploring-Starts

起始分布，覆盖(s, a)全集。

Pi下，充分利用每一个trajectory里的所有(s, a)对，访问，增长式估计相应Q值。
每一个trajectory结束后，Q值未必稳定，就进行策略改进。
迭代之。

(3) MC-epsilon-greedy

过程分布，覆盖(s, a)全集，通过策略的stochastic实现。

e-Pi下，充分利用每一个trajectory里的所有(s, a)对，访问，增长式估计相应Q值。
每一个trajectory结束后，Q值未必稳定，就进行策略改进，生成e-Pi。
迭代之。

*stochastic approximation

理解:

以某形式的公式，为理论依据，进行实际采样与近似估计。

(1) Incremental-Estimation:

$$w_k = \frac{1}{k} \sum_{i=1}^k x_i \qquad w_{k-1} = \frac{1}{k-1} \sum_{i=1}^{k-1} x_i$$

$$w_k = \frac{1}{k} [(k-1)w_{k-1} + x_k] = w_{k-1} + \frac{1}{k} [x_k - w_{k-1}]$$

(2) Robbins-Monro:

$$g(w) = 0 \quad (\text{g is unknown, w is input, 0 is output})$$

$$g(w) = \nabla_w L(w) = 0 \quad (\text{w is parameters})$$

$$g(w) = L(w) - C = 0$$

$$w^* \text{ is the solution (Convergence Condition)}$$

$$w_{k+1} = w_k + a_k [\tilde{g}(w_k, \eta_k) - 0] \qquad \text{error}$$

$$= w_k + a_k (g(w_k) + \eta_k)$$

$$\text{iteration :} \quad \{w_k\} + \{\tilde{g}_k\} + \{a_k\}$$

(3) Optimazation:

$$\min_w J(w) = E[f(w, X)]$$

$$\Rightarrow \Rightarrow \quad \nabla_w E[f(w, X)] = 0$$

$$\Rightarrow \Rightarrow \quad \text{Gradient :} \quad \text{InputSpace,} \quad \text{direction} + \text{magnitude}$$

GD + (Mini)Batch GD + Stochastic GD:

$$w_{k+1} = w_k - \alpha_k \nabla_w E[f(w_k, X)] = w_k - \alpha_k E[\nabla_w f(w_k, X)]$$

$$w_{k+1} = w_k - \alpha_k \frac{1}{n} \sum_{i=1}^n \nabla_w f(w_k, x_i)$$

$$w_{k+1} = w_k - \alpha_k \nabla_w f(w_k, x_k)$$

ch5: Temporal Difference

理解:

Model-free. Dynamics with Transition.

①TD时序差分: 在不同时刻, 对同一个量的估计, 有差, 利用差改进估计。

②没有模型 p 、只有数据 t , 进行估计:

MC利用整条trajectory, 估计 V 、 Q , 离线/无偏/大方差;

TD利用片段transition, 估计 V 、 Q , 在线/有偏/小方差。

③SARSA, 用TD估计某个 P_i 的 Q 。

④Q-Learning, 用TD直接估计 Q^* 。(异轨, 从数据中提取共性MDP- Q^* 信息)

⑤行为策略采集数据, 利用数据计算TD-target (数据组织形式),

目标策略的估计值趋向TD-target, 即TD-target代表着目标策略。

⑥on-policy: 用一个策略和环境交互, 得到experience, 估计这个策略, 改进这个策略。

再用改进的策略进行循环, 交互、估计、改进, 直至最优 (这个系列的最优)。

off-policy: 一个策略和环境交互, 得到experience,

另一个策略被估计、被改进。(目标策略不易采集数据, 所以需要行为策略, 异轨)

experience应当具有一些性质, 能够架通两个策略:

行为策略能够采到, 目标策略能够使用。

*Temporal Difference:

time	model	s	s'
t	w(t)	v(s; w(t))	v(s'; w(t))
t+1	w(t+1)	r+v(s'; w(t))	
t	w(t)	q(s, a; w(t))	q(s', a'; w(t))
t+1	w(t+1)	r+q(s', a'; w(t))	
t	w(t)	q(s, a; w(t))	max(q(s', A))
t+1	w(t+1)	r+max(q(s', A))	

(1) TD-V: s-a-r-s

$$v_{t+1}(s_t) = v_t(s_t) + \alpha_t[r_{t+1} + \gamma v_t(s_{t+1}) - v_t(s_t)]$$

$$v_{t+1}(s) = v_t(s) \quad \forall s \neq s_t$$

(2) TD-Q: SA-R-SA

Policy Evaluation:

$$q_{t+1}(s_t, a_t) = q_t(s_t, a_t) + \alpha_t(s_t, a_t)[r_{t+1} + \gamma q_t(s_{t+1}, a_{t+1}) - q_t(s_t, a_t)]$$

$$q_{t+1}(s, a) = q_t(s, a) \quad \forall (s, a) \neq (s_t, a_t)$$

Policy Improvement:

某个策略下的Q值，可能还未估计稳定，
就可以进行策略提升。最终目标提升至Q*。

(3) TD-Q*: Q*-Learning (s-a-r-s)

Value Improvement:

$$q_{t+1}(s_t, a_t) = q_t(s_t, a_t) + \alpha_t(s_t, a_t)[r_{t+1} + \gamma \max_a q_t(s_{t+1}, a) - q_t(s_t, a_t)]$$

$$q_{t+1}(s, a) = q_t(s, a) \quad \forall (s, a) \neq (s_t, a_t)$$

*(DP+MC+TD)-Summary:

value	DP-model(p)	TD-data(t)
v	Policy-Evaluation(vBEs)	TD(0/n/...)
v*	Value-iteration(vBOEs)	-
q	Policy-Evaluation(qBEs)	TD(0)=SARSA
q*	Value-iteration(qBOEs)	Q*-Learning

ch6: Value Function Approximation

理解:

①函数拟合的优势: 用更少的参数量 (存储), 估计, 更多的状态量 (泛化);
原因在于, 函数这种表达形式, 存储了更本质的信息。

函数拟合的劣势: 存在精度损失。

函数拟合的关键: 最优的结构 + 最优的参数, 提升拟合准确程度。

②状态分布(.): 平均分布U、稳态分布D ($d + (pi+p) = d$)。

优化时, SGD将不同分布吸收拉平了。

③某个真实分布中, 函数的一系列真值, 形成一个超面;

经过各种采样, 得到超面的部分观测真值;

建立一个模型, 并优化参数, 用来拟合未完全观测到的超面。

④自举, 让不均匀的高低估扩展, 可能导致最优性发生变化。

(1) Objective Function:

$$L(w) = E_{S \sim (\cdot)} [(v_{\pi}(S) - \tilde{v}(S; w))^2]$$

(2) Optimazation:

$$\begin{aligned} w_{k+1} &= w_k - \alpha_k \nabla_w L(w_k) \\ w_{t+1} &= w_t + \alpha_t (v_{\pi}(s_t) - \tilde{v}(s_t; w_t)) \nabla_w \tilde{v}(s_t; w_t) \end{aligned}$$

(3) Algorithm:

$$\begin{aligned} w_{t+1} &= w_t + \alpha_t (g_t - \tilde{v}(s_t; w_t)) \nabla_w \tilde{v}(s_t; w_t) & MC + VA \\ w_{t+1} &= w_t + \alpha_t [r_{t+1} + \gamma \tilde{v}(s_{t+1}; w_t) - \tilde{v}(s_t; w_t)] \nabla_w \tilde{v}(s_t; w_t) & TD + VA \end{aligned}$$

(4) Approximator(w):

$$\begin{aligned} linear &= kx + b \\ nonlinear &= neural \ network \end{aligned}$$

(5) SARSA + VA: (SA-R-SA + q- + w)

Value Update(w):

$$w_{t+1} = w_t + \alpha_t [r_{t+1} + \gamma q(s_{t+1}, a_{t+1}; w_t) - q(s_t, a_t; w_t)] \nabla_w q(s_t, a_t; w_t)$$

Policy Improvement: epsilon-greedy

(6) Q*-Learning + VA: (S-A-R-S + q- + w)

Value Update(w):

$$w_{t+1} = w_t + \alpha_t [r_{t+1} + \gamma \max_a q(s_{t+1}, a; w_t) - q(s_t, a_t; w_t)] \nabla_w q(s_t, a_t; w_t)$$

Policy Improvement: (epsilon-)greedy

(7) Deep Q*-Learning: (Buffer(S-A-R-S) + NN(w)*2)
(continuous states + discrete actions)

Value Update(w):

$$w_{t+1} = w_t + \alpha_t [r_{t+1} + \gamma \max_a q(s_{t+1}, a; w_T) - q(s_t, a_t; w_t)] \nabla_w q(s_t, a_t; w_t)$$

Policy Improvement: (epsilon-)greedy

(8) Double Q*-Net: (Buffer(S-A-R-S) + NN(w)*2)

ch7: Policy Gradient

*Policy-Summary:

Policy		
Form	Distribution	Network Model (Continuous A ?)
Evaluation	Values (V + Q)	Performance Metrics
Optimization	(epsilon)-greedy	Gradient-Ascent

(1) Policy Network: (Storage + Generalization)

$\pi(a|s; \theta)$ from Distribution to Network Model

(2) Optimal Policy: (!= Optimal Values)

$\max_{\theta} \text{metrics}[\pi(a|s; \theta)]$ from Values to Metrics

(3) Distribution(d):

$d - \pi$: e.g. *Uniform Distribution* : $d(s) = 1/|S| \quad \forall s \in S$
 $d + \pi$: e.g. *Stationary Distribution* : $d_{\pi}^T \cdot P_{\pi} = d_{\pi}^T$
 d_0 : e.g. *Specific Distribution* : $d_0(s_0) = 1 \quad d_0(s) = 0 \quad \forall s \neq s_0$

(4) Metric-1: average state value (?) = average average return

$$\begin{aligned}
 J_1(\theta) &= \overline{v_{\pi}} = E_{S \sim d}[v_{\pi}(S)] = \sum_{s \in d} d(s) v_{\pi}(s) = d^T \cdot v_{\pi} \\
 &= E \left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} \right]
 \end{aligned}$$

(5) Metric-2: average one-step reward = average infinite reward

$$\begin{aligned}
 J_2(\theta) &= \overline{r_{\pi}} = E_{S \sim d}[r_{\pi}(S)] = \sum_{s \in d} d(s) r_{\pi}(s) \\
 &= \lim_{n \rightarrow \infty} \frac{1}{n} E \left[\sum_{k=1}^n R_{t+k} \right]
 \end{aligned}$$

(6)Metrics-Relationship:

$$\bar{r}_\pi = (1 - \gamma)\bar{v}_\pi$$

(7)Policy Gradient Theorem:

$$\begin{aligned}\nabla_\theta J(\theta) &= \sum_{s \in S} \eta(s) \sum_{a \in A} \nabla_\theta \pi(a|s; \theta) q_\pi(s, a) \\ &= E[\nabla_\theta \ln \pi(A|S; \theta) q_\pi(S, A)] \\ \theta_{t+1} &= \theta_t + \alpha \nabla_\theta \ln \pi(a_t|s_t; \theta_t) q_\pi(s_t, a_t) \\ &= \theta_t + \alpha \frac{q_\pi(s_t, a_t)}{\pi(a_t|s_t; \theta_t)} \nabla_\theta \pi(a_t|s_t; \theta_t)\end{aligned}$$

proof.

q: (1)MC (2)MC-BS (3)Advantage (4)TD

stochastic gradient ascent

exploitation + exploration

(8)REINFORCE:

$$\begin{aligned}\theta_{t+1} &= \theta_t + \alpha \nabla_\theta \ln \pi(a_t|s_t; \theta_t) q_t(s_t, a_t) \\ q_\pi(s_t, a_t) &: q_t(s_t, a_t)(MC)\end{aligned}$$

(9)REINFORCE-BS:

ch8: Actor Critic

(1) QAC: (SARSA-VA) + PG

$$\begin{aligned} w_{t+1} &= w_t + \alpha_w [r_{t+1} + \gamma q(s_{t+1}, a_{t+1}; w_t) - q(s_t, a_t; w_t)] \nabla_w q(s_t, a_t; w_t) \\ \theta_{t+1} &= \theta_t + \alpha_\theta \nabla_\theta \ln \pi(a_t | s_t; \theta_t) q(s_t, a_t; w_{t+1}) \end{aligned}$$

(2) A2C: (TD-VA) + PG

$$\begin{aligned} q_\pi(s_t, a_t) - BS &= q_\pi(s_t, a_t; w'_t) - v_\pi(s_t; w_t) \\ &= \delta_t = r_{t+1} + \gamma v(s_{t+1}; w_t) - v(s_t; w_t) \\ w_{t+1} &= w_t + \alpha_w \delta_t \nabla_w v(s_t; w_t) \\ \theta_{t+1} &= \theta_t + \alpha_\theta \nabla_\theta \ln \pi(a_t | s_t; \theta_t) \delta_t \end{aligned}$$

Importance Sampling:

$$\begin{aligned} E_{x \sim p}[f(x)] &= \int_x p(x) f(x) dx \\ &= \int_x q(x) \frac{p(x)}{q(x)} f(x) dx \\ &= E_{x \sim q} \left[\frac{p(x)}{q(x)} f(x) \right] \end{aligned} \quad \beta(x) = \frac{p(x)}{q(x)} = \frac{\pi(x)}{b(x)}$$

(3) Off-Policy A2C:

$$\begin{aligned} q_\pi(s_t, a_t) - BS &= q_\pi(s_t, a_t; w'_t) - v_\pi(s_t; w_t) \\ &= \delta_t = r_{t+1} + \gamma v(s_{t+1}; w_t) - v(s_t; w_t) \\ w_{t+1} &= w_t + \alpha_w \frac{\pi(a_t | s_t; \theta_t)}{b(a_t | s_t)} \delta_t \nabla_w v(s_t; w_t) \\ \theta_{t+1} &= \theta_t + \alpha_\theta \nabla_\theta \ln \pi(a_t | s_t; \theta_t) \frac{\pi(a_t | s_t; \theta_t)}{b(a_t | s_t)} \delta_t \end{aligned}$$

ch9: Deterministic PG

(1) DPG:

$$\mu(s; \theta) = a * \quad \mu(s, a; \theta) = 1$$

$$J(\theta) = E_{S \sim d_\mu} [v_\mu(S)] = \sum_{S \sim d_\mu} d_\mu(s) v_\mu(s) = d_\mu^T \cdot v_\mu$$

$$\nabla_\theta J(\theta) = \sum_{S \sim d_\mu} d_\mu(s) \nabla_\theta \mu(s; \theta) (\nabla_a q_\mu(s, a))|_{a=\mu(s)} \quad \text{proof.}$$

$$= E_{S \sim d_\mu} [\nabla_\theta \mu(S; \theta) (\nabla_a q_\mu(S, a))|_{a=\mu(S)}]$$

$$\delta_t = r_{t+1} + \gamma q(s_{t+1}, \mu(s_{t+1}; \theta_t); w_t) - q(s_t, a_t; w_t)$$

$$w_{t+1} = w_t + \alpha_w \delta_t \nabla_w q(s_t, a_t; w_t)$$

$$\theta_{t+1} = \theta_t + \alpha_\theta \nabla_\theta \mu(s_t; \theta_t) (\nabla_a q(s_t, a; w_{t+1}))|_{a=\mu(s_t)}$$