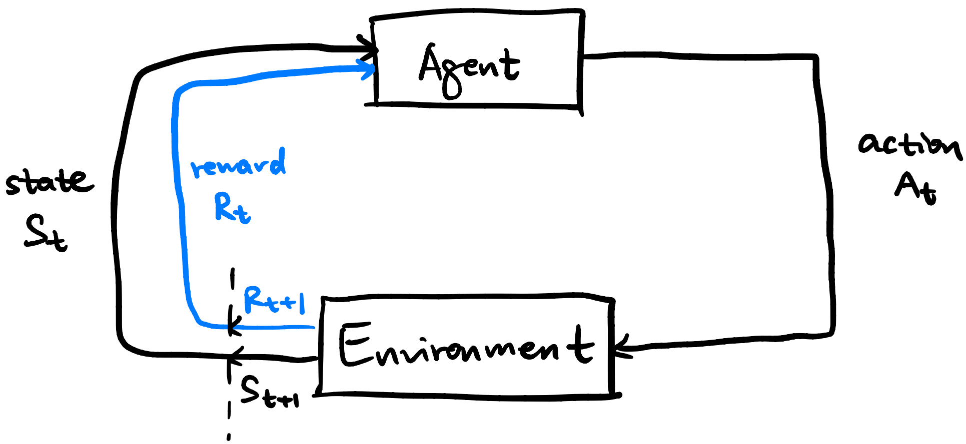
single-agent RL

- Markov Decision Process (MDP) ☆
- Dynamic Programming (DP)
- Monte Carlo (MC)
- Temporal Difference and Q-learning (TD) ☆
- Policy Gradient

Theoretical framework: MDP



logic: environment has state, agent observes state and make an action, he receives reward based on his state-action pair. However, his action also changes the state so at the next time step his new reward is based on his new state etc.

The agent's objective: maximize aggregate reward at all time steps.

⇓ Main Difficulty in RL!

Greedy strategy fails! If too shortsighted, only choose the action that brings highest current reward, might fall into a very bad state and get very low future rewards!

e.g. Consider bandit problem, RL problem with no state transition, agent just continuously make actions and receive reward (same as bandit in the casino).

At time t, make action A_t, assume reward

$$R_t|_{A_t=a} \sim N(\underbrace{q_*(a)}_{\text{mean reward for action } a \text{ unkown to agent}}, 1)$$

assume 4 bandits, A_t can take value $\{1,2,3,4\}$ (which to select)

$\boxed{1}$	$\boxed{2}$	$\boxed{3}$	$\boxed{4}$
$q_*(1)=0$	$q_*(2)=3$	$q_*(3)=2$	$q_*(4)=10$

obviously, optimal strategy is to always select bandit 4

but if we use completely greedy strategy,

$A_0 = 1 \Rightarrow R_0 = -0.5$ (realization of $N(0,1)$)

$A_1 = 2 \Rightarrow R_1 = 3.1$ (from $N(3,1)$)

we will think that choosing bandit 2 is a lot better, so we stick to bandit 2 and miss bandit 4.

even with no state transition, greedy is not a good strategy (ϵ-greedy instead)

$\Downarrow$

trade-off $\begin{cases} \text{exploitation (maximize the reward)} \\ \text{exploration (know about what happens for other actions)} \end{cases}$

If always exploit, might miss a better action
If always explore, might have low total reward

$\Downarrow$

motivation of a lot of
algorithm and concepts

Basic Setting of MDP:

State $s \in \mathcal{S}$, action $a \in \mathcal{A}(s)$, reward $r \in \mathcal{R} \subseteq \mathbb{R}$

the set of available actions might depend on the state

Infinite time-horizon

Finite MDP for simplicity: $|\mathcal{S}| < \infty$, $\forall s \in \mathcal{S}$, $|\mathcal{A}(s)| < \infty$, $|\mathcal{R}| < \infty$

denote $\mathcal{A} = \bigcup_{s \in \mathcal{S}} \mathcal{A}(s)$ is still finite

(the set of all possible actions regardless of the state)

dynamics $\quad p(s', r \mid s, a) \triangleq \mathbb{P}(S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a)$

(time-homogeneous)

given the action $\{A_t\}$, $\{S_t\}$ is Markov, transition from S_{t-1} to S_t only depends on the value of S_{t-1}.

(given S_t, A_t, future $R_{t+1}, S_{t+1}, A_{t+1}, \cdots \perp\!\!\!\perp$ past $S_0, A_0, R_1, \cdots, A_{t-1}$)

seeing current state s, making action a, the prob. that next state is s' and get immediate reward r

Calculations:

$$p(s' \mid s, a) \triangleq \mathbb{P}(S_t = s' \mid S_{t-1} = s, A_{t-1} = a)$$

$$= \sum_r p(s', r \mid s, a)$$

state-transition prob

$$r(s,a) \triangleq \mathbb{E}(R_t \mid S_{t-1}=s, A_{t-1}=a)$$ expected reward for state-action pair

$$= \sum_r r \cdot p(r \mid s,a)$$

$$= \sum_r r \cdot \sum_{s'} p(s', r \mid s, a)$$

expected reward for state-action-next state triple

$$r(s,a,s') \triangleq \mathbb{E}(R_t \mid S_{t-1}=s, A_{t-1}=a, S_t=s')$$

$$= \sum_r r \cdot p(r \mid s,a,s')$$

$$= \sum_r r \cdot \frac{p(s', r \mid s,a)}{p(s' \mid s,a)} = \frac{\sum_r r \cdot p(s', r \mid s,a)}{\sum_r p(s', r \mid s,a)}$$

so dynamics provides expression for everything we are interested in.

Modelling the environment

Modelling the agent

Agent obj: maximize the sum of rewards, in infinite horizon setting there's convergence problems so use discounting

$$G_t \triangleq \sum_{k=0}^{\infty} \underbrace{\gamma^k}_{\text{discount rate}} \cdot R_{t+k+1}$$

return at time t

(aggregating all rewards $\geq$ time t)

notice that R_{t+1} is the reward received based on S_t, A_t.

Always write $G_t = R_{t+1} + \gamma \cdot G_{t+1}$.

Agent obj: maximize $\mathbb{E} G_0$

The agent makes decision based on a policy

$$\pi(a|s) = \mathbb{P}(A_t = a \mid S_t = s) \quad \text{(time-homogeneous)}$$

a conditional distribution on A, meaning on seeing current state s, react with action a with some probability.

Why not always deterministically? Because of exploration!

At this point, it's possible to compute

$$\mathbb{E}(R_{t+1} | S_t) = \sum_r r \cdot \mathbb{P}(R_{t+1} = r | S_t)$$

policy connects state and action

$$= \sum_r r \cdot \sum_a \mathbb{P}(A_t = a | S_t) \cdot \mathbb{P}(R_{t+1} = r | S_t, A_t = a)$$

$$= \sum_r r \cdot \sum_a \pi(a | S_t) \cdot \sum_{s'} p(s', r | S_t, a)$$

Value Function:

$$v_\pi(s) \triangleq \mathbb{E}_\pi(G_t | S_t = s)$$

state value func for policy π

$$q_\pi(s, a) \triangleq \mathbb{E}_\pi(G_t | S_t = s, A_t = a)$$

state-action value func for policy π

Connection of v_π, q_π ?

Obviously, $v_\pi(s) = \sum_a \mathbb{P}(A_t = a | S_t = s) \cdot \mathbb{E}_\pi(G_t | S_t = s, A_t = a)$

$$= \sum_a \pi(a | s) \cdot q_\pi(s, a)$$

On the other hand,

$$q_\pi(s, a) = \mathbb{E}_\pi(R_{t+1} + \gamma \cdot G_{t+1} | S_t = s, A_t = a)$$

$$= \underbrace{r(s, a)}_{\text{calculated!}} + \gamma \cdot \mathbb{E}_\pi(G_{t+1} | S_t = s, A_t = a)$$

$$= \sum_r r \cdot \sum_{s'} p(s', r \mid s, a) + \gamma \cdot \sum_{s'} \mathbb{E}_\pi \left(G_{t+1} \mid S_{t+1} = s', S_t = s, A_t = a \right) \cdot \mathbb{P}(S_{t+1} = s' \mid S_t = s, A_t = a)$$

$R_{t+2}, R_{t+3}, \cdots$

Markov

$\mathbb{E}_\pi (G_{t+1} \mid S_{t+1} = s') = V_\pi(s')$

$$= \sum_r r \cdot \sum_{s'} p(s', r \mid s, a) + \gamma \cdot \sum_{s'} V_\pi(s') \cdot \sum_r p(s', r \mid s, a)$$

$$= \sum_{r, s'} \left(r + \gamma \cdot V_\pi(s') \right) \cdot p(s', r \mid s, a)$$

MDP time-homogeneous Markov + policy time-homogeneous + infinite time horizon

$\Downarrow$

q_π, V_π time-homoge~ (not depend on t)

that's why consider inf time horizon (recall stochastic control)

Bellman Consistency Equation:

Describe what condition V_π, q_π has to satisfy,

✻ : consider $S_t \to S_{t+1}$ then can write back as value func!

$$V_\pi(s) = \mathbb{E}(R_{t+1} | S_t = s) + \gamma \cdot \mathbb{E}_\pi(G_{t+1} | S_t = s)$$

future reward following policy π with current state s

$$= \sum_r r \cdot \mathbb{P}(R_{t+1} = r | S_t = s) + \gamma \cdot \sum_{s', a} p(s', a | s) \cdot \mathbb{E}_\pi(G_{t+1} | S_t = s, A_t = a, S_{t+1} = s')$$

consider action

Markov!

$$= \sum_r r \cdot \sum_a \mathbb{P}(A_t = a | S_t = s) \cdot p(r | s, a) + \gamma \cdot \sum_{s', a} \underbrace{p(s', a | s)}_{= \pi(a|s) \cdot p(s'|a,s)} \cdot \mathbb{E}_\pi(G_{t+1} | S_{t+1} = s')$$

$$= \sum_r r \cdot \sum_a \pi(a|s) \sum_{s'} p(s', r | s, a) + \gamma \cdot \sum_{s', a} \pi(a|s) \cdot \sum_r p(s', r | s, a) \cdot V_\pi(s')$$

$$= \underbrace{\sum_a \pi(a|s)}_{\text{average w.r.t. policy based on current state}} \underbrace{\sum_{s', r} p(s', r | s, a)}_{\text{average w.r.t. dynamics}} \cdot \left[\underbrace{r}_{\text{immediate reward}} + \underbrace{\gamma V_\pi(s')}_{\text{discounted future reward}} \right]$$

Remark: One naturally thinks about if it's possible to solve out V_π from Bellman consistency equation, actually one can prove that solution $\exists$ and is unique. (Exercise)

However, one would need a given policy π and the dynamics p, so it depends on knowledge of the model.

Bellman consistency equation for q_π:

$$
\begin{aligned}
q_\pi(s,a) &= \mathbb{E}_\pi(G_t \mid S_t=s, A_t=a) \\
&= \mathbb{E}(R_{t+1} \mid S_t=s, A_t=a) + \gamma \cdot \mathbb{E}_\pi(G_{t+1} \mid S_t=s, A_t=a) \\
&= \sum_r r \cdot p(r \mid s,a) + \gamma \cdot \sum_{s',a'} \mathbb{P}_\pi(S_{t+1}=s', A_{t+1}=a' \mid S_t=s, A_t=a) \\
&\quad \cdot \underbrace{\mathbb{E}_\pi(G_{t+1} \mid S_t=s, A_t=a, S_{t+1}=s', A_{t+1}=a')}_{q_\pi(s',a')} \\
&= \sum_r r \cdot \sum_{s'} p(s',r \mid s,a) + \gamma \sum_{s',a'} p(s' \mid s,a) \cdot \pi(a' \mid s') \cdot q_\pi(s',a') \\
&= \underbrace{\sum_{r,s'} p(s',r \mid s,a)}_{\text{average w.r.t. dynamics}} \Big[\underbrace{r}_{\text{immediate reward}} + \underbrace{\gamma \sum_{a'} \pi(a' \mid s')\, q_\pi(s',a')}_{\text{discounted future reward}} \Big]
\end{aligned}
$$

Optimal Value Function

Since the obj is to maximize $\mathbb{E}G_0$, if a policy has higher state value $V_\pi(s)$ for $\forall s \in S$, it's a better policy. Does there exist the optimal policy?

$$\begin{cases} \forall s \in S, \quad V_*(s) \triangleq \sup_\pi V_\pi(s) \\ \forall s \in S,\ a \in A, \quad q_*(s) \triangleq \sup_\pi q_\pi(s,a) \end{cases}$$

optimal value function

as pointwise sup

optimal policy (not nece unique)

Thm: $\exists$ deterministic policy π_* s.t.

$$\forall s \in S,\ \forall a \in A,\quad V_{\pi^*}(s) = V_*(s), \quad q_{\pi^*}(s,a) = q_*(s,a)$$

value of optimal policy

optimal value

Surprisingly, pointwise sup of value function is just the value function of optimal policy!

local, for each $s \in S$, sup may be approx by different π

global optimal

Pf:

means take this action w.p. 1 (deterministic policy)

$$\pi_*(s) \triangleq \arg\sup_a \mathbb{E}\left[R_{t+1} + \gamma \cdot V_*(S_{t+1}) \,\middle|\, S_t = s, A_t = a \right]$$

whenever at state s, put all prob mass on the action that maximizes expectn of sum of immediate and discounted future reward.

$\forall s \in S$, $V_{\pi_*}(s) \le V_*(s)$ by def of V_*.

On the other hand, $\forall s \in S$,

$$V_*(s) = \sup_{\pi} \left\{ \mathbb{E}_{\pi}\left(R_{t+1} + \gamma G_{t+1} \mid S_t = s\right) \right\}$$

tower property

$$= \sup_{\pi} \left\{ \mathbb{E}_{\pi}\left[R_{t+1} + \gamma \cdot \mathbb{E}_{\pi}\left(G_{t+1} \mid S_t = s, A_t, S_{t+1}\right) \mid S_t = s \right] \right\}$$

change to π'

$$\le \sup_{\pi} \left\{ \mathbb{E}_{\pi}\left[R_{t+1} + \gamma \cdot \underbrace{\sup_{\pi'} \mathbb{E}_{\pi'}\left(G_{t+1} \mid S_t = s, A_t, S_{t+1}\right)}_{\text{Markov} \; = \; V_*(S_{t+1})} \mid S_t = s \right] \right\}$$

$$= \sup_{\pi} \left\{ \mathbb{E}_{\pi}\left[R_{t+1} + \gamma \cdot V_*(S_{t+1}) \mid S_t = s \right] \right\}$$

$$= \sup_{\pi} \left\{ \mathbb{E}\left[R_{t+1} + \gamma \cdot V_*(S_{t+1}) \mid S_t = s, A_t \sim \underbrace{\pi(\cdot \mid s)}_{\text{policy appears here}} \right] \right\}$$

def of $\tilde{\pi}$

$$\le \mathbb{E}_{\pi_*}\left[R_{t+1} + \gamma \cdot V_*(S_{t+1}) \mid S_t = s \right]$$

$$\Downarrow$$

$$V_*(S_t) \le \mathbb{E}_{\pi_*}\left[R_{t+1} + \gamma \cdot V_*(S_{t+1}) \mid S_t \right]$$

$\Downarrow$ iteratively

$$V_*(S_t) \le \mathbb{E}_{\pi_*}\left(R_{t+1} + \gamma \cdot \underbrace{\mathbb{E}_{\pi_*}\left[R_{t+2} + \gamma \cdot V_*(S_{t+2}) \mid S_{t+1} \right]}_{\text{upper bound of } V_*(S_{t+1})} \mid S_t \right)$$

$$= \mathbb{E}_{\pi_*}\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 \cdot V_*(S_{t+2}) \mid S_t \right]$$

So $V_*(S_t) \leq \cdots \leq \mathbb{E}_{\pi_*}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots | S_t]$
$\leq V_{\pi_*}(S_t)$

proves $\forall s \in S$, $V_*(s) = V_{\pi_*}(s)$.
and π_* is deterministic optimal policy.

For q: recall $q_\pi(s,a) = \sum_{r,s'} [r + \gamma V_\pi(s')] \cdot p(s', r | s, a)$
(connection of q_π, V_π)

take sup w.r.t. π on both sides PW

$$q_*(s,a) \leq \sum_{r,s'} [r + \gamma V_*(s')] \cdot p(s', r | s, a)$$
(sup goes inside sum)
$$= \sum_{r,s'} [r + \gamma \cdot V_{\pi_*}(s')] \cdot p(s', r | s, a)$$
$$= q_{\pi_*}(s,a)$$

proves that it's also optimal w.r.t. q.

Bellman Optimality Equation :

Plug in $\pi = \pi_*$ in Bellman consistency equation

$$V_\pi(s) = \sum_a \pi(a|s) \cdot \sum_{s',r} p(s',r|s,a) \cdot \left[r + \gamma \cdot V_\pi(s')\right]$$

$$\Downarrow$$

$$V_*(s) = \sum_a \boxed{\pi_*(a|s)} \sum_{s',r} p(s',r|s,a) \cdot \left[r + \gamma \cdot V_*(s')\right]$$

$$q_\pi(s,a) = \sum_{s',r} p(s',r|s,a) \left[r + \gamma \cdot \sum_{a'} \pi(a'|s') \cdot q_\pi(s',a')\right]$$

$$\Downarrow$$

$$q_*(s,a) = \sum_{s',r} p(s',r|s,a) \left[r + \gamma \cdot \sum_{a'} \boxed{\pi_*(a'|s')} \cdot q_*(s',a')\right]$$

☆ : but it's not a good form since we don't know π_* in prior!

To get rid of π_*, discover relationship between V_*, q_*

$$V_*(s) = \sup_\pi V_\pi(s) = \sup_\pi \sum_a \pi(a|s) \cdot q_\pi(s,a)$$

$$\leq \sup_\pi \underbrace{\sum_a \pi(a|s) \cdot q_*(s,a)}_{\text{weighted average}}$$

$$= \max_a q_*(s,a)$$

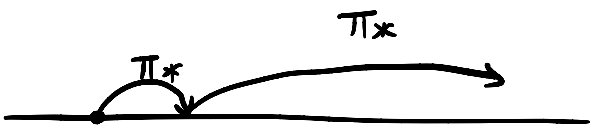
but $V_*(s) = V_{\pi_*}(s) < \max_a q_*(s,a)$ by contradiction

consider $\tilde{\pi}(s) \triangleq \arg\max_a q_*(s,a)$,

$$q_{\pi_*}(s, \tilde{\pi}(s)) = \max_a q_*(s,a) > V_{\pi_*}(s) = q_{\pi_*}(s, \pi_*(s))$$

$$\Downarrow$$

strictly better, contradiction with π_* being optimal policy!



$$\Downarrow$$

Policy Improvement Thm (proof not mentioned)

$$\tilde{\pi} = \pi_*$$

$$\Downarrow$$

$$V_*(s) = \max_a q_*(s,a)$$

☆: compact form of Bellman optimality equation

Useful Bellman Optimality Equation:

$$V_*(s) \overset{*}{=} \max_a q_*(s,a)$$

(HJB type)

$$= \max_a \sum_{r,s'} \left[r + \gamma \cdot V_*(s')\right] \cdot p(s', r \mid s, a)$$

recall

$$q_\pi(s,a) = \sum_{r,s'} \left(r + \gamma \cdot V_\pi(s')\right) \cdot p(s', r \mid s, a)$$

$$q_*(s,a) = \sum_{r,s'} \left[r + \gamma \cdot V_*(s')\right] \cdot p(s', r \mid s, a)$$

$$\overset{*}{=} \sum_{r,s'} \left[r + \gamma \cdot \max_{a'} q_*(s', a')\right] \cdot p(s', r \mid s, a)$$

$\pi_*(s) = \arg\max_a q_*(s,a)$ is optimal policy

$$= \arg\max_a \sum_{r,s'} \left[r + \gamma \cdot V_*(s')\right] \cdot p(s', r \mid s, a)$$

Correspondence:

① : Stochastic control (cts time, finite horizon, non-randomized policy

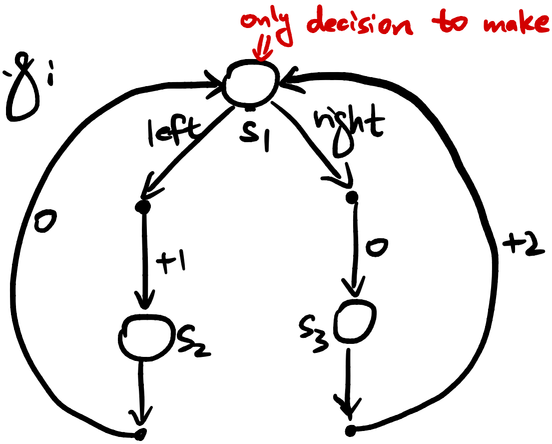
⇓

Solve control with RL

② : Statistical decision theory (no time evolution, recall the way deriving Bayes estimator, similar to optimal policy)

⇓

information theory methods for optimality in RL e.g.:



rewards shown are deterministic

two deter~ policy π_{left}, π_{right},

which is optimal?

pf: $S = \{s_1, s_2, s_3\}$ $A = \{left, right\}$,

Bellman optimality equation:

$$\begin{cases} V_*(s_2) = \gamma V_*(s_1) \\ V_*(s_3) = 2 + \gamma V_*(s_1) \\ V_*(s_1) = \max\{1 + \gamma V_*(s_2), \gamma V_*(s_3)\} \end{cases}$$

$$\Downarrow$$

$$V_*(s_1) = \max\{1 + \gamma^2 V_*(s_1), 2\gamma + \gamma^2 V_*(s_3)\}$$

$$\Downarrow$$

$$\begin{cases} \gamma \in [0, \frac{1}{2}), & \pi_l \text{ optimal} \\ \gamma = \frac{1}{2}, & \pi_l, \pi_r \text{ optimal} \\ \gamma \in (\frac{1}{2}, 1), & \pi_r \text{ optimal} \end{cases}$$

(intuitively match!)

DP: (model-based)

Policy iteration: current policy π_k, ① do policy evaluation (get V_{π_k} or q_{π_k} through Bellman consistency equation as fixed point iteration),
② policy improvement get better π_{k+1} $\quad \pi_{k+1}(s) = \arg\max_a q_{\pi_k}(s,a)$
repeat until convergence

Value iteration: directly solve V_* or q_* through fixed point iteration of Bellman consistency equation and construct π_*

Check: $\mathcal{T} v(s) = \max_a \mathbb{E}[R_{t+1} + \gamma \cdot v(S_{t+1}) \mid S_t = s, A_t = a]$
is contraction mapping

$\exists\, 0 \le k < 1,\ \forall v, v',\ \|\mathcal{T}v - \mathcal{T}v'\|_\infty \le k \|v - v'\|_\infty$

actually the discount rate γ

$\mathcal{T}$ is Bellman optimality operator

pros: easy to implement, efficient, fit with small problems

cons: model-based, can't deal with cts state/action space,

MC

Natural since v, q are conditional expectation, policy evaluation done by MC.

First-visit MC ES:

①: Exploring start, any (s,a) pos probability of selected as initial state (maintain exploration!)

②: For fixed policy π, experience: $S_0, A_0, R_1, \ldots$
Calculate returns at each time, find out the time of first visit to (s,a) and add the return at this time of first visit into list(s,a)
ensure i.i.d. MC samples (can be every-visit also)
(main a list for each state-action pair)

③: Update estimate for $q_\pi(s,a)$ as sample average of all numbers in list(s,a)

④: Construct greedy deter~ policy (after enough experience gained)
$\pi'(s) = \arg\max_a q_\pi(s,a)$ as policy improvement

⑤: Iterate until $\pi \to \pi_*$, $q \to q_*$

pros: model-free (pure experience)

cons: Inefficient, deter~ policy, X cts state action space
has to wait until end of episode to calculate return

TD:

MC: $V_\pi(s) = \mathbb{E}_\pi(G_t | S_t = s)$, sample G_t with condition $S_t = s$

Bellman consistency equation

TD: $V_\pi(s) = \mathbb{E}_\pi(R_{t+1} + \gamma V_\pi(S_{t+1}) | S_t = s)$, expect to see

$$\mathbb{E}_\pi(\underbrace{R_{t+1} + \gamma V_\pi(S_{t+1}) - V_\pi(S_t)}_{\text{temporal difference error } \delta_t} | S_t = s) = 0.$$

- $0 \Rightarrow V_\pi(s)$ is good
- $>0 \Rightarrow V_\pi(s)$ too small
- $<0 \Rightarrow V_\pi(s)$ too large

TD(0) (one-step TD): only policy evaluation

①: For fixed policy π, initialize state S, figure out action A given by π and S

②: Take action A, observe reward R, next state S'

learn the guess from the guess, bootstrap!

③: $V(S) \leftarrow V(S) + \alpha \cdot [R + \gamma V(S') - V(S)]$

α: learning rate, parameter

$S \leftarrow S'$ go to next state

④: loop until episode ends

- pros: model-free (better than DP), much more efficient (experience while update) — online (better than MC)
- cons: X cts state/action space

TD(0) policy evaluation is proved to converge to v_π w.p. 1 if stochastic approx scheme holds, i.e.

$$\begin{cases} \sum_n \alpha_n = \infty & \text{large enough, overcome fluctuation} \\ \sum_n \alpha_n^2 < \infty & \text{small enough, guarantee convergence} \end{cases}$$

for α_n as learning rate at time n

$\Downarrow$ Idea of TD(0) applied on estimating q_*, $Q \approx q_*$

SARSA (state-action-reward-state-action)

①: Init state S. Generate action A based on S and ε-greedy policy derived from $Q \approx q_*$

②: Take action A, observe reward R, next state S'

③: Generate A' based on S' and ε-greedy policy derived from Q.

④: (Bellman consistency equation)

$$Q(S,A) \leftarrow Q(S,A) + \alpha \cdot \boxed{R + \gamma Q(S',A') - Q(S,A)}$$

TD error for q_π

$$q_\pi(s,a) = \mathbb{E}_\pi\left[R_{t+1} + \gamma \cdot q_\pi(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a\right]$$

$S \leftarrow S'$, $A \leftarrow A'$ (next time step)

actually following this action

⑤: Loop until episode ends

on-policy: learning of Q based on A', generated by the policy which is constructed based on Q and we are actually following it!

Q-learning: $(Q \approx q_*)$

①: Init state S. Generate action A based on S and ε-greedy policy derived from Q. actually following the action here

②: Take action A, reward R, next state S'

(Bellman optimality equation)

③: $Q(S,A) \leftarrow Q(S,A) + \alpha \cdot \underbrace{\left[R + \gamma \cdot \max_a Q(S',a) - Q(S,A)\right]}_{\text{TD error}}$

$S \leftarrow S'$ (next time step)

④: Loop until episode ends

$$q_*(s,a) = \sum_{r,s'} \left[r + \gamma \cdot \max_{a'} q_*(s',a')\right] \cdot p(s', r \mid s,a)$$

write back as expectation

$$= \mathbb{E}\left[R_{t+1} \mid S_t = s, A_t = a\right] + \gamma \cdot \mathbb{E}\left(\max_{a'} q_*(S_{t+1}, a') \mid S_t = s, A_t = a\right)$$

$$= \mathbb{E}\left[R_{t+1} + \gamma \cdot \max_a q_*(S_{t+1}, a) \mid S_t = s, A_t = a\right]$$

Off-policy: uses the next action in the way of $\max_a Q(s', a)$, actually it's the greedy action w.r.t. Q. We are just estimating the return assuming a greedy policy were followed but not actually following this policy.

On/off-policy depends on if the action you use in TD error is exactly what you follow !

Policy Gradient Method:

Directly update policy π, can deal with problems where estimation of value function is infeasible.

Parametrize $\pi(a|s,\theta)$ with parameter θ, obj of agent is to get the optimal policy π to maximize

$$J(\theta) = \mathbb{E}_\pi(G_0 | S_0 = s_0)$$

Thm (Policy Gradient):

∇ w.r.t. θ

$$\nabla J(\theta) \propto \sum_{s \in S} \mu(s) \sum_{a \in A} q_\pi(s,a) \cdot \nabla \pi(a|s,\theta)$$

where μ is on-policy dist as prob meas. on S.

$$\eta(s) \triangleq \sum_{k=0}^{\infty} \gamma^k \cdot \mathbb{P}_\pi(S_k = s | S_0 = s_0), \quad \mu(s) \triangleq \frac{\eta(s)}{\sum_{s' \in S} \eta(s')}$$

(in continuing case, μ is stationary dist)

$\Downarrow$ proportion of time MDP has spent in state s

Naturally, shall update θ in the direction of $\nabla J(\theta)$

learning rate

$$\theta \leftarrow \theta + \alpha \cdot \sum_s \mu(s) \sum_a q_\pi(s,a) \cdot \nabla \pi(a|s,\theta)$$

how to calculate $\mu(s)$ and $q_\pi(s,a)$?

Turns out we don't need to calculate those but can write them in terms of expectation.

Assume $S_0 \sim \mu$ (stationary distribution) $\left(S_t \xrightarrow{d} \mu \ (t \to \infty) \text{ possible to argue if } (S_t) \text{ nice enough} \right)$

assume $\gamma = 1$, then

$$\underbrace{\sum_s \mu(s)}\sum_a q_\pi(s,a) \cdot \nabla \pi(a|s,\theta)$$

$$= \mathbb{E}_\pi \left[\sum_a q_\pi(S_t,a) \cdot \nabla \pi(a|S_t,\theta) \right]$$

$\left(q_\pi(S_t,a) = \mathbb{E}_\pi(G_t | S_t, A_t = a) \right)$

$$= \mathbb{E}_\pi \underbrace{\sum_a \pi(a|S_t,\theta)}\cdot \frac{\nabla \pi(a|S_t,\theta)}{\pi(a|S_t,\theta)} \cdot q_\pi(S_t,a)$$

$$= \mathbb{E}_\pi \mathbb{E}_{A_t \sim \pi(\cdot|S_t,\theta)} \left[\frac{\nabla \pi(A_t|S_t,\theta)}{\pi(A_t|S_t,\theta)} \cdot q_\pi(S_t,A_t) \middle| S_t \right]$$

$$= \mathbb{E}_\pi \left[\frac{\nabla \pi(A_t|S_t,\theta)}{\pi(A_t|S_t,\theta)} \cdot q_\pi(S_t,A_t) \right]$$

$$= \mathbb{E}_\pi \left[\nabla \log \pi(A_t|S_t,\theta) \cdot \mathbb{E}_\pi(G_t | S_t, A_t) \right]$$

$$= \mathbb{E}_\pi \left[G_t \cdot \nabla \log \pi(A_t|S_t,\theta) \right]$$

gradient has form of expectation!

$\Downarrow$

SGD

☆: $\theta \leftarrow \theta + \alpha \cdot G_t \cdot \nabla \log \pi(A_t|S_t,\theta)$

REINFORCE:

①: Experience $S_0, A_0, R_1, \cdots$ following current policy under parameter θ

②: At each time t, build up G_t and perform

$$\theta \leftarrow \theta + \alpha \cdot \gamma^t \cdot G_t \cdot \nabla \log \pi(A_t | S_t, \theta)$$

combine with DL: π can be approx by NN with softmax output layer.

Actor-Critic:

actor: approx policy, generating actions

critic: approx state value func to assess the action taken (TD error)

①: Current state S, generate action $A \sim \pi(\cdot|S,\theta)$
take A, get reward R, next state S'

②: TD error

$$\delta \leftarrow R + \gamma \hat{v}(S', w) - \boxed{\hat{v}(S, w)}$$

approx state value func, parametrized by parameter w

③: $\begin{cases} w \leftarrow w + \alpha^{w} \cdot \boxed{\delta \cdot \nabla_w \hat{v}(S_t, w)} \\ \theta \leftarrow \theta + \alpha^{\theta} \cdot \gamma^{t} \cdot \delta \cdot \nabla_\theta \log \pi(A_t | S_t, \theta) \end{cases}$

update critic

learning rate

update actor

④: $S \leftarrow S'$ (next time step)

DL: approx policy & value func with NN, so organize 2 NNs and ∇_w, ∇_θ can be derived easily numerically.

$\begin{cases} \text{Pros: model-free, cts } \cancel{\text{state}} \text{ action space, enough randomized policy, online} \\ \text{Cons: parametric form, time-consuming training} \end{cases}$

weighted mean-square error of value approx:

$$\overline{VE}(w) = \sum_s \mu(s) \left[v_\pi(s) - \hat{v}(s, w) \right]^2$$

so $\nabla_w \overline{VE}(w) \propto \mathbb{E}_\pi \left[\hat{v}(S_t, w) - G_t \right] \cdot \nabla_w \hat{v}(S_t, w)$

again use SGD idea avoid calculation of expectation

replace G_t with $R + \gamma \hat{v}(S', w)$ $\Leftarrow$

(TD idea)