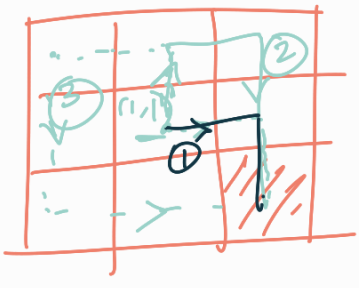
Monte Carlo

→ Agent learns from experience
→ Complete knowledge of env is not needed
→ Agent can learn from a model of the env., but does not need to know the probability distributions of all transitions. $p(s'|s,a)$

Ex.



① -5 g((1,1))
② -10 g
③ -20 g

-17.5

$V((1,1)) = -17.5$

$V(s) =$

3.2	13.2	-15.1
1.5	-17.5	-1.7
-4.5	-5.6	

① policy evaluation (Monte Carlo prediction)
② policy improvement (Monte Carlo control)

Monte Carlo Prediction

$$V(s) = \frac{1}{C(s)} \sum_{m=1}^{M} \sum_{t=0}^{T-1} \mathbb{I}(S_t^m = s)\, g_t^m$$

$$\mathbb{I}(S_t^m = s) = \begin{cases} 1, & S_t^m = s \\ 0, & S_t^m \neq s \end{cases}$$

Example :-

$S = \{s_0, s_1\}$ $\quad A = \{a_0, a_1\}$

T	1	2	3	4	5	6
S	s_0	s_1	s_0	s_0	s_1	s_0
A	a_0	a_0	a_1	a_0	a_1	-
R	-	5	10	15	-3	2

Given, $\pi(a|s)$

(1 episode)

$$V_\pi(s_0) = ?$$

① → $G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$

$= R_{t+1} + \gamma G_{t+1}$

② → $V_\pi(s) = E[G_t | S_t = s]$

$\gamma = 1$.

Sol :-

$g_6 = 0$

$g_5 = R_6 + \gamma G_6 = 2 + 1(0) = 2$

$g_4 = R_5 + \gamma G_5 = -3 + 2 = -1$

$g_3 = 15 - 1 = 14$

$$g_2 = 10 + 14 = 24$$

$$g_1 = 5 + 24 = 29$$

$$v_\pi(s_0) = \frac{g_1 + g_3 + g_4 + g_6}{4}$$

$$= \frac{29 + 14 + (-1) + 0}{4}$$

$$= \frac{42}{4} = 10.3$$

Iterative Calculation of mean

$$\frac{x_0 + x_1 + x_2 \cdots + x_n}{n} = \bar{x}_n$$

$$\bar{x}_n = \bar{x}_{n-1} + \frac{1}{n}\left(x_n - \bar{x}_{n-1}\right)$$

($\bar{x}_n$: new avg; $\bar{x}_{n-1}$: old avg; x_n: new val; $\bar{x}_{n-1}$: old avg)

$$= \frac{n\bar{x}_{n-1} + x_n - \bar{x}_{n-1}}{n}$$

$$= \frac{\bar{x}_{n-1}(n-1) + x_n}{n}$$

$$= \frac{\frac{x_0 + x_1 + x_2 \cdots + x_{n-1}}{(n-1)} \times (n-1) + x_n}{n}$$

$$= \bar{x}_n$$

Monte Carlo prediction (policy evaluation)

→ Given $\pi(a|s)$, $V(s) \approx V_\pi(s)$

Initialise

$V(s) =$ random

loop many times, (ideally loop forever)

→ choose the start state randomly.

① Simulate one episode under $\pi(a|s)$

→ $(s_0, a_0, r_1)(s_1, a_1, r_2), s_2 \ldots\ldots\ldots$

= history

② Update $V(s)$ based on the episode.

$G = 0$

for $t = T-1, T-2 \ldots\ldots 0$

$G_t = R_{t+1} + \gamma G_{t+1}$

$$\rightarrow V(s_t) = V(s_t) + \frac{1}{N(s_t)} \left(G_t - V(s_t)\right)$$

$V(s_t)$: new avg; $V(s_t)$: old avg; $N(s_t)$: # state s_t is visited; G_t: new value; $V(s_t)$: old avg

---

→ For 4 × 4 Gridworld, evaluate a random policy (actions are chosen randomly). → code the MC prediction algorithm.

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

Bellmans optimality equation for state value function

$$= \max_{a \in A} q_*(s,a)$$

Monte carlo Control

we maximise the state action value function.

$$S = \{ s_0, s_1, s_2 \}$$

$$A = \{ a_0, a_1 \}$$

$q_\pi(s,a) \rightarrow$ Q-table

	a_0	a_1	
s_0	1.25	1.75	a_1
s_1	1.3	1.4	a_1
s_2	1.3	10	a_0

ϵ-greedy

$0 \leq \epsilon \leq 1$

Say $\epsilon = 0.2$

20% of the times choose the action randomly, not according to the Q-table.

$\epsilon \uparrow \Rightarrow$ agent explores more.

On-policy Monte Carlo Control

Find $\pi \approx \pi^*$

initialise $Q(s,a) \rightarrow$ random values,

loop many times,

① → Simulate one episode using the Q-values in a ϵ-greedy method, $(s_0, a_1, r_3)(s_1, a_1, r_2)\ldots\ldots$

② → set $G = 0$

③ → update $Q(s,a)$ for the episode,

→ loop → $T-1, T-2 \ldots\ldots 0$

$$G_t = R_{t+1} + \gamma G_{t+1}$$

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha\left(G_t - Q(s_t, a_t)\right)$$

$Q(s_t, a_t)$: new avg; $Q(s_t, a_t)$: old avg; G_t: new value; $Q(s_t, a_t)$: old avg

---

$$q_\pi(s,a) = \mathbb{E}\left[G_t \mid S_t = s, A_t = a\right]$$