Monte Carlo Methods

We have no knowledge of the environment.

- Monte Carlo methods require only experience.
- We solve problem based on averaging sample returns.
- Only for episodic tasks. only on completion of an episode values are updated.

[5.1] Monte Carlo Prediction:

learning state-value function for a given policy.

we estimate $v_\pi(s)$ i.e. value of state s under policy π.

$$v_\pi(s) = E[G_t \mid S_t = s]$$
$$v_\pi(s) = E[R_{t+1} + \gamma v_\pi(s') \mid S_t = s]$$

Each occurance of s in an episode is called visit to state s.

MC-methods

- first visit MC → avg. of returns following first visits to s
- every visit MC → avg. return following all visits to s.

First-visit MC prediction, for estimating $V \approx v_\pi$.

Input: a policy π to be evaluated

Initialise:

$V(S) \in \mathbb{R}$, arbitrarily $\forall s \in S$

Returns $(S) \leftarrow$ an empty list, $\forall s \in S$.

Loop forever (for each episode):

generate an episode following π: $S_0, A_0, R_1, S_1, A_1, R_2 \cdots$

$G \leftarrow 0$

Loop for each step of episode $t = T-1, T-2, \cdots 0$

$G \leftarrow \gamma G + R_{t+1}$

Unless S_t appears in $S_0, S_1 \cdots S_{t-1}$:

Append G to Returns (S_t)

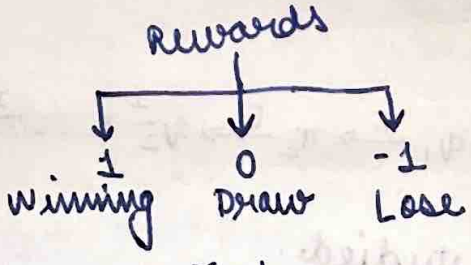
$V(S_t) \leftarrow$ avg. (Returns (S_t)).

Example 5.1 → Blackjack:

Object of the popular casino game is to obtain cards with sum close to 21 without exceeding it.

Face cards → 10, Ace → 1 or 11,

Sticks on any sum greater than 17 or equal to it.



$\gamma = 1$

[5.2] Monte Carlo estimation of action values:

If a model is not available, particularly useful to estimate action values, rather than state values.

Similar to state value estimation but now we don't just look at state s but we look at (s, a) pairs.

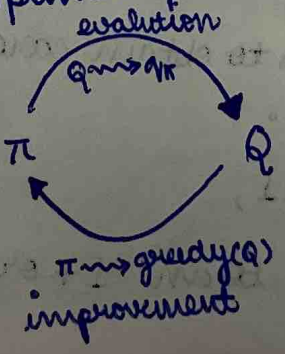
A visit to (s, a) means we are in state s and action a was taken in it.

The only complication is many state action pairs will never be visited.

To deal with this we mention in the start of the episodes state-action pair, and that every pair has a non-zero probability of being picked as the start.

[5.3] Monte Carlo Control:

We approximate Optimal policies



$$\pi_0 \xrightarrow{E} q_{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} q_{\pi_1} \xrightarrow{I} \pi_2 \xrightarrow{E} q_{\pi_2} \xrightarrow{I} \cdots \xrightarrow{I} \pi_* \xrightarrow{E} q_{\pi_*}$$

Policy evaluation is already studied.

$$\pi(s) \doteq \underset{a}{\operatorname{argmax}}\, q(s,a)$$

Policy improvement can be done by making each π_{k+1} greedy w.r.t q_{π_k}.

$$q_{\pi_k}(s, \pi_{k+1}(s)) = q_{\pi_k}(s, \underset{a}{\operatorname{argmax}}\, q_{\pi_k}(s,a))$$

$$= \max_a q_{\pi_k}(s,a)$$

$$\geq q_{\pi_k}(s, \pi_k(s))$$

$$\geq v_{\pi_k}(s)$$

We made 2 assumptions

→ The episodes have an exploring start

→ Policy evaluation can be done with infinite number of episodes.

Monte Carlo ES, for estimating $\pi \approx \pi_*$

Initialise:

$\pi(s) \in A(s)$ (arbitrarily) $\forall s \in S$

$Q(s,a) \in R$ (arbitrarily) $\forall s \in S, a \in A(s)$

Returns $(s,a) \leftarrow$ empty list, $\forall s \in S, a \in A(s)$

Loop forever (for each episode):

Choose $S_0 \in S, A_0 \in A(S_0)$ randomly such that all pairs have $p > 0$

generate an episode from S_0, A_0 following π: $S_0, A_0, R_1, \ldots S_{T-1}, A_{T-1}, R_T$

$G \leftarrow 0$

~~Unless the pair~~

Loop for each step, $t = T-1, T-2, \ldots, 0$:

$G \leftarrow \gamma G + R_{t+1}$

Unless the pair S_t, A_t appears in $S_0, A_0, \ldots S_{t-1}, A_{t-1}$:

Append G to Returns (S_t, A_t)

$Q(S_t, A_t) \leftarrow$ avg(Returns (S_t, A_t))

$\pi(S_t) \leftarrow \underset{a}{\operatorname{argmax}}\, Q(S,a)$.

[5.4] Monte Carlo Control without exploring starts:

Two approaches to solve this → on-policy methods
→ off-policy methods

on policy methods attempt to evaluate or improve a policy that is used to make decisions, whereas off-policy methods evaluate or improve the policy different from that used to generate data.

Monte Carlo ES in 5.3 is example of on-policy.

In on policy, we generally have a soft policy i.e

$\pi(a|s) > 0 \quad \forall s \in S, a \in A(s)$, but gradually shifted closer to a deterministic policy.

we use ε-greedy i.e ~~with~~ most of the time we choose maximal estimated action value but with probability ε, we instead select an action at random.

All non greedy actions are given minimal policy of selection $\frac{\varepsilon}{|A(s)|}$, and the remaining bulk $1 - \varepsilon + \frac{\varepsilon}{|A(s)|}$.

ε-greedy policies are example of ε-soft policies for which

$\pi(a|s) \geq \frac{\varepsilon}{|A(s)|}$ $\forall$ states and actions for some $\varepsilon > 0$.

On-policy first-visit MC control (for ε-soft) estimates $\pi \approx \pi_*$:

Algorithm parameter: small $\varepsilon > 0$

Initialise:

$\pi \leftarrow$ an arbitrary ε-soft policy

$Q(s,a) \in \mathbb{R}$ (arbitrarily) $\forall s \in S, a \in A(s)$

Returns $(s,a) \leftarrow$ empty list, $\forall s \in S, \forall a \in A(s)$

Repeat forever (for each episode):

Generate an episode following π: $S_0, A_0, R_1, S_1, A_1, R_2 \dots$

$G \leftarrow 0$

Loop for each step of episode $t = T-1, T-2, \dots, 0$:

$G \leftarrow \gamma G + R_{t+1}$

Unless the pair S_t, A_t appears in $S_0, A_0, S_1, A_1 \dots$:

Append G to Returns (S_t, A_t)

$Q(S_t, A_t) \leftarrow$ average (Returns (S_t, A_t))

$A^* \leftarrow \underset{a}{\operatorname{argmax}}\, Q(S_t, a)$

For all $a \in A(S)$:

$$\pi(a|S_t) \leftarrow \begin{cases} 1 - \varepsilon + \frac{\varepsilon}{|A(S)|} & a = A^* \\ \frac{\varepsilon}{|A(S)|} & a \neq A^* \end{cases}$$

Any ε-greedy policy w.r.t q_π is an improvement over any ε-soft policy π is assured by policy improvements.

Let π' be an ε-greedy policy.

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

$$= \frac{\varepsilon}{|A(S)|} \sum_a q_\pi(s,a) + (1-\varepsilon) \max_a q_\pi(s,a)$$

$$\geq \frac{\varepsilon}{|A(s)|}\sum_a q_\pi(a|s) + (1-\varepsilon)\sum_a \frac{\pi(a|s) - \frac{\varepsilon}{|A(s)|}}{1-\varepsilon} q_\pi(s,a)$$

$$= \frac{\varepsilon}{|A(s)|}\sum_a q_\pi(s,a) - \frac{\varepsilon}{|A(s)|}\sum_a q_\pi(s,a) + \sum_a \pi(a|s)\, q_\pi(s,a)$$

$$= v_\pi(s)$$

If we consider a new environment that overrides the agents choice of action with probability ε with a randomly picked action.

Let $\tilde{v}_*$ and $\tilde{q}_*$ denote new environments optimal value functions

Then a policy π is optimal among policies iff $v_\pi = \tilde{v}_*$.

$$\tilde{v}_*(s) = (1-\varepsilon)\max_a \tilde{q}_*(s,a) + \frac{\varepsilon}{|A(s)|}\sum_a \tilde{q}_*(s,a)$$

$$= (1-\varepsilon)\max_a \sum_{s',r} p(s',r|s,a)\left[r + \gamma \tilde{v}_*(s')\right]$$

$$+ \frac{\varepsilon}{|A(s)|}\sum_a \sum_{s',r} p(s',r|s,a)\left[r + \gamma \tilde{v}_*(s')\right]$$

$$v_\pi(s) = (1-\varepsilon)\max_a q_\pi(s,a)\sum_{s',r} p(s',r|s,a)\left[r + \gamma v_\pi(s')\right]$$

$$+ \frac{\varepsilon}{|A(s)|}\sum_a \sum_{s',r} p(s',r|s,a)\left[r + \gamma v_\pi(s')\right]$$

If we compare both equations we get

$$v_\pi(s) = \tilde{v}_*(s).$$

[5.5] Off-Policy Prediction via Importance Sampling:

We use 2 policies → 1. that is learned about and becomes optimal
2. Exploratory policy used to generate behaviour.

The policy being learned about is called target policy, policy used to generate behaviour, is called behaviour policy.

off-policy methods → greater variance
→ slower convergence
→ more powerful & general

We take a prediction problem, both target and behaviour policies are fixed. We need to estimate v_π or q_π, we have episodes following another policy b, where $b \neq \pi$, in this case $b \rightarrow$ behaviour policy, $\pi \rightarrow$ target policy.

We require that every action taken under π is also at least occasionally taken under b. i.e $\pi(a|s) > 0 \Rightarrow b(a|s) > 0$. this is called assumption of coverage.

b must be stochastic in states where it is not identical to π, but π may be deterministic.

In control → target policy → deterministically greed w.r.t action value functions.
→ behaviour policy → stochastic and exploratory.

Almost all off-policy methods use importance sampling.

importance sampling: a general technique for estimating expected values under one distribution given samples from other.

we weight returns acc. to the relative probability of their trajectories under the target and behaviour policies, called importance sampling ratio.

Given starting state $S_t \rightarrow A_t, S_{t+1}, A_{t+1}, S_{t+2} \cdots, S_T$

$$\Pr\{A_t, S_{t+1}, A_{t+1} \cdots, S_T\}$$

$$= \pi(A_t|S_t)\, p(S_{t+1}|S_t, A_t)\, \pi(A_{t+1}|S_{t+1}) \cdots p(S_T|S_{T-1}, A_{T-1})$$

$$= \prod_{k=t}^{T-1} \pi(A_k|S_k)\, p(S_{k+1}|S_k, A_k)$$

Thus the relative probability under target & behaviour is

$$\rho_{t:T-1} = \frac{\prod_{k=t}^{T-1} \pi(A_k|S_k)\, p(S_{k+1}|S_k, A_k)}{\prod_{k=t}^{T-1} b(A_k|S_k)\, p(S_{k+1}|S_k, A_k)}$$

$$\rho_{t:T-1} = \frac{\prod_{k=t}^{T-1} \pi(A_k|S_k)}{\prod_{k=t}^{T-1} b(A_k|S_k)}$$

we have G_t from the b policy

$$E[G_t \mid S_t = s] = v_b(s)$$

this cannot be averaged in terms of $v_\pi(s)$

But

$$\boxed{E[\rho_{t:T-1} G_t \mid S_t = s] = v_\pi(s)}$$

we define set of all timestamps in which state s is visited, denoted $\mathcal{T}(s)$. This is for an every-visit method, for a first-visit method $\mathcal{T}(s)$ would only include that were the first time stamps to visit s within their episodes. Let $T(t)$ denote first time of termination then $\{G_t\}_{t \in \mathcal{T}(s)}$ are the returns that pertain to state s and mean $\{\rho_{t:T(t)-1}\}_{t \in \mathcal{T}(s)}$

To estimate $v_\pi(s)$

$$\boxed{V(s) \doteq \frac{\sum_{t \in \mathcal{T}(s)} \rho_{t:T(t)-1} G_t}{|\mathcal{T}(s)|}}$$

↓

Ordinary importance sampling

$$\boxed{V(s) \doteq \frac{\sum_{t \in \mathcal{T}(s)} \rho_{t:T(t)-1} G_t}{\sum_{t \in \mathcal{T}(s)} \rho_{t:T(t)-1}}}$$

↓

Weighted importance sampling.

suppose we obtain K(s) returns for class s.

$$V_\pi(s) \approx \frac{1}{K(s)} \sum_{k=1}^{K(s)} \rho_{t(k):T(k)-1} G^{(k)}$$

↓ ordinary

$$V_\pi(s) \approx \frac{\sum_{k=1}^{K(s)} \rho_{t(k):T(k)-1} G^{(k)}}{\sum_{k=1}^{K(s)} \rho_{t(k):T(k)-1}}$$

↓ weighted

[5.6] Incremental Implementation:

In ordinary importance sampling, returns are scaled by the importance sampling ratio $\rho_{t:T(t)-1}$, then simply averaged, we can use incremental updates of averaging but use scaled returns instead.

For weighted importance sampling:

Suppose we have $G_1, G_2 \cdots, G_{n-1}$ all starting from the same state with a corresponding random weight W.

$$V_n \doteq \frac{\sum_1^{n-1} W_k G_k}{\sum_1^{n-1} W_k}, \quad n \geq 2$$

To update

$$V_{n+1} \doteq V_n + \frac{W_n}{C_n} [G_n - V_n], \quad n \geq 1$$

$$C_{n+1} \doteq C_n + W_{n+1}$$