Converting policy iteration to be model-free

Two expressions of action value:

- Expression 1 requires the model

$$q_{\pi_k}(s,a) = \sum_r p(r|s,a)r + \gamma \sum_{s'} p(s'|s,a) v_{\pi_k}(s')$$

- Expression 2 does not requires the model

$$q_{\pi_k}(s,a) = \mathbb{E}[G_t|S_t = s, A_t = a]$$

> Idea to achieve model-free RL: Use expression 2 to calculate $q_{\pi_k}(s,a)$ based on data.

Procedure of Monte Carlo (MC) estimation of action values:

- Starting from (s,a), following policy π_k, generate an episode.
- The return of episode is $g(s,a)$.
- $g(s,a)$ is a sample of G_t.
- Then

$$q_{\pi_k}(s,a) = \mathbb{E}[G_t|S_t = s, A_t = a] \approx \frac{1}{N} \sum_{i=1}^{n} g^{(i)}(s,a)$$

Pseudocode: MC Basic algorithm

Initialization: Initial guess π_0.
Goal: Search for an optimal policy.

For the kth iteration $(k = 0, 1, 2, \dots)$, do
    For every state $s \in \mathcal{S}$, do
        For every action $a \in \mathcal{A}(s)$, do
            Collect sufficiently many episodes starting from (s,a) by following π_k
            Policy evaluation:
            $q_{\pi_k}(s,a) \approx q_k(s,a) =$ the average return of all the episodes starting from (s,a)
        Policy improvement:
        $a_k^*(s) = \arg\max_a q_k(s,a)$
        $\pi_{k+1}(a|s) = 1$ if $a = a_k^*$, and $\pi_{k+1}(a|s) = 0$ otherwise

> - MC Basic reveals the core idea of MC-based model-free RL, but not practical due to **low efficiency**.
> - MC Basic is a variant of the policy iteration algorithm.

MC Exploring Starts

Utilizing samples more efficiently

> Visit: every time a state-action appears in the episode, it is called a **visit** of that state-action pair.
>
> e.g.
>
> $$s_1 \xrightarrow{a_2} s_2 \rightarrow \ldots$$
>
> (s_1, a_2) is a pair.

Consider a episode, starting from (s_1, a_1):

$s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} s_3 \xrightarrow{a_3} \ldots$

The subepisode be viewed as a new episode. Like

$$\begin{aligned} s_2 \xrightarrow{a_2} s_3 \xrightarrow{a_3} \ldots \quad & [\text{subepisode starting from}(s_2, a_2)] \\ s_3 \xrightarrow{a_3} \ldots \quad & [\text{subepisode starting from}(s_3, a_3)] \end{aligned}$$

These new episodes can be used to estimate more action values.

Updating policies more efficiently

The strategy is to use the return of a single episode to approximate the corresponding action value.

Pseudocode:

Initialization: Initial policy $\pi_0(a|s)$ and initial value $q(s,a)$ for all (s,a). Returns$(s,a) = 0$ and Num$(s,a) = 0$ for all (s,a).
Goal: Search for an optimal policy.

For each episode, do
  Episode generation: Select a starting state-action pair (s_0, a_0) and ensure that all pairs can be possibly selected (this is the exploring-starts condition). Following the current policy, generate an episode of length T: $s_0, a_0, r_1, \ldots, s_{T-1}, a_{T-1}, r_T$.
  Initialization for each episode: $g \leftarrow 0$
  For each step of the episode, $t = T-1, T-2, \ldots, 0$, do
    $g \leftarrow \gamma g + r_{t+1}$
    Returns$(s_t, a_t) \leftarrow$ Returns$(s_t, a_t) + g$
    Num$(s_t, a_t) \leftarrow$ Num$(s_t, a_t) + 1$
    Policy evaluation:
    $q(s_t, a_t) \leftarrow$ Returns$(s_t, a_t)/$Num(s_t, a_t)
    Policy improvement:
    $\pi(a|s_t) = 1$ if $a = \arg\max_a q(s_t, a)$ and $\pi(a|s_t) = 0$ otherwise

MC ϵ-Greedy: Learning without exploring starts

An ϵ-greedy policy is a stochastic policy that has a higher chance of choosing the greedy action and the same nonzero probability of taking any other action.

$$\pi(a|s) = \begin{cases} 1 - \frac{\epsilon}{|\mathcal{A}(s)|}(|\mathcal{A}(s)| - 1), & \text{for the greedy action,} \\ \frac{\epsilon}{|\mathcal{A}(s)|}, & \text{for the other } |\mathcal{A}(s)| - 1 \text{ actions,} \end{cases}$$

where $|\mathcal{A}(s)|$ denotes the number of actions associated with s.

Pseudocode:

Initialization: Initial policy $\pi_0(a|s)$ and initial value $q(s,a)$ for all (s,a). Returns$(s,a) = 0$ and Num$(s,a) = 0$ for all (s,a). $\epsilon \in (0,1]$
Goal: Search for an optimal policy.

For each episode, do
Episode generation: Select a starting state-action pair (s_0, a_0) (the exploring starts condition is not required). Following the current policy, generate an episode of length T: $s_0, a_0, r_1, \ldots, s_{T-1}, a_{T-1}, r_T$.
Initialization for each episode: $g \leftarrow 0$
For each step of the episode, $t = T-1, T-2, \ldots, 0$, do
$g \leftarrow \gamma g + r_{t+1}$
Returns$(s_t, a_t) \leftarrow$ Returns$(s_t, a_t) + g$
Num$(s_t, a_t) \leftarrow$ Num$(s_t, a_t) + 1$
Policy evaluation:
$q(s_t, a_t) \leftarrow$ Returns$(s_t, a_t)/$Num(s_t, a_t)
Policy improvement:
Let $a^* = \arg\max_a q(s_t, a)$ and

$$\pi(a|s_t) = \begin{cases} 1 - \frac{|\mathcal{A}(s_t)|-1}{|\mathcal{A}(s_t)|}\epsilon, & a = a^* \\ \frac{1}{|\mathcal{A}(s_t)|}\epsilon, & a \neq a^* \end{cases}$$