Multi-Armed Bandit(MABs)

What is Multi-Armed Bandits ?

Assume you are in a situation where you are proposed k jobs $\{J_1, J_2, ..., J_k\}$ with different salaries, where the salaries are not fixed, i.e. each salary corresponding to a specific job has a certain variance you don't know. Now you have to take an action which in our case is to choose one of the k jobs proposed to you. The goal at the end is to choose the optimal job which will bring you the maximum salary. Let say you are giving a possibility to work for a certain number of time or months $\{t_1, t_2, ..., t_n\}$ in order to make up your mind about the final job you will choose.

Terminology

1. Action A_t : choose one of the k jobs $\{J_1, J_2, ..., J_k\}$ at the time step $t \in \{t_1, t_2, ..., t_n\}$.

2. Action-value estimates $Q_t(a)$: it's the estimate salary called **Reward** **(R_t)** in Reinforcement Learning that we are going to use from now on, that you expect to get after working at a time $t \in \{t_1, t_2, ..., t_n\}$. Indeed, initially we don't know the exact reward we'll get after taking an action a called the true values $q_*(a)$ define by :

$$q_*(a) = \mathbb{E}[R_t | A_t = a], \text{ for } a \in \{J_1, J_2, ..., J_k\},$$

and

$$Q_t(a) \approx q_*(a), \text{ for all } a.$$

At the end the goal is to maximize the total reward. Maybe we should try to choose the actions that both learn their values (Exploration) and prefer those that seem best (Exploitation).

Exploration/Exploitation Dilemma

We define the greedy action A_t (action that maximize Q) at time t as :

$$A_t^* = \arg\max_a Q_t(a).$$

- If $A_t = A_t^*$, then we exploit.

- If $A_t \neq A_t^*$, then we explore.

Action-Value Methods

We want to learn the action-value estimates and nothing else. One solution could be to take the average reward, that is:

$$Q_t(a) = \frac{\sum_{i=1}^{t-1} R_i \times \mathbb{1}_{A_i}}{\sum_{i=1}^{t-1} \mathbb{1}_{A_i}},$$

where

$$\mathbb{1}_{A_i} = \begin{cases} 1 \text{ if } A_i = a, \\ 0 \text{ else.} \end{cases}$$

Also

$$\lim_{N_t(a) \to \infty} Q_t(a) = q_*(a),$$

where $N_t(a)$ is the number of times action a was taken by time t.

ϵ-greedy action selection

In greedy action selection, we always exploit. In ϵ-greedy, we are usually greedy but with a probability ϵ, we instead pick an action at random, that is:

$$A_t = \begin{cases} \arg\max_a Q(a), & \text{with probability } 1 - \epsilon, \\ \text{Random action}, & \text{with probability } \epsilon. \end{cases}$$

Turning averaging into a learning rule

Given the formula to learn the action-value estimates, let's look at an updating rule. Assume for simplicity we can write Q_n as:

$$Q_n = \frac{\sum_{i=1}^{n-1} R_i}{n-1}, \text{ for } n \neq 1,$$

then

$$Q_{n+1} = Q_n + \frac{1}{n}(R_n - Q_n).$$

Let us prove it. Assume $n \neq 1$, then

$$\begin{aligned}
Q_{n+1} &= \frac{\sum_{i=1}^{n} R_i}{n} \\
&= \frac{n-1}{n} \times \frac{\sum_{i=1}^{n} R_i}{n-1} \\
&= \frac{n-1}{n} \times \frac{\sum_{i=1}^{n-1} R_i + R_n}{n-1} \\
&= \frac{n-1}{n} \times \left(\frac{\sum_{i=1}^{n-1} R_i}{n-1} + \frac{R_n}{n-1} \right) \\
&= \frac{n-1}{n} \times \left(Q_n + \frac{R_n}{n-1} \right) \\
&= Q_n - \frac{Q_n}{n} + \frac{R_n}{n} \\
&= Q_n + \frac{1}{n}(R_n - Q_n).
\end{aligned}$$

Non-Stationarity Problem

The Non-Stationarity Problem says the the true action-values change slowly over time. Instead of using the the sample average, we will use the recency-weighted average:

$$\begin{aligned}
Q_{n+1} &= Q_n + \alpha(R_n - Q_n) \\
&= (1-\alpha)^n Q_1 + \sum_{k=1}^{n} \alpha(1-\alpha)^{n-k} R_k,
\end{aligned}$$

with $\alpha \in (0,1]$ is a constant step-size parameter.

Optimistic Initial Values

Assume we know that the action-value estimates is bounded by a certain value Q_b, that is for all a, $Q_t(a) \leq Q_b$. The optimistic initial value consists of initializing the action-value estimates by Q_b. We say that we initialize the action-value optimistically. This will encourage the exploration at the beginning.

Upper Confidence Bound (UCB)

UCB is another way to the choose an action by doing both exploration and exploitation. The action is chosen according to the following formula :

$$A_t = \arg\max_a \left[Q_t(a) + c\sqrt{\frac{\log(t)}{N_t(a)}} \right],$$

$Q_t(a)$ is the exploitation term, $\sqrt{\frac{\log(t)}{N_t(a)}}$ is the exploration term and c is the trade-off parameter between the two terms.

Next we will implement in python the ϵ-greedy, the optimistic initial value and the UCB.

Please feel free to reach me for any suggestion.