Reinforcement Learning

Epsilon-Greedy

Dr. Alireza Aghamohammadi

The Epsilon-Greedy Algorithm

- ❖ A popular and straightforward approach to MAB involves injecting randomness into the action-selection process.
- ❖ Most of the time, the agent exploits the best-known action, but occasionally it explores by selecting actions randomly.
- ❖ Let R_i denote the reward obtained after the i-th selection of an action, and let Q_n denote the estimated value of the action after it has been selected $n-1$ times. The estimate can be expressed as:

$$Q_n = \frac{R_1 + R_2 + \cdots + R_{n-1}}{n-1}$$

- ❖ Given Q_n and the n-th reward R_n, the updated estimate Q_{n+1} for all n rewards can be computed:

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

Epsilon-Greedy Algorithm

```
Input: Action set $\mathcal{A} = \{1, \dots, k\}$
Output: Action-value estimates $Q(a)$ for each action $a$
Initialization:
For each $a \in \mathcal{A}$, set $Q(a) \leftarrow 0$ and $N(a) \leftarrow 0$
while true do
    Select action $A$ according to:
        $$A \leftarrow \begin{cases} \arg\max_a Q(a) & \text{with probability } 1 - \epsilon \\ \text{a random action} & \text{with probability } \epsilon \end{cases}$$
    Obtain reward $R$ from the environment for action $A$
    Update the count: $N(A) \leftarrow N(A) + 1$
    Update the action-value estimate:
        $$Q(A) \leftarrow Q(A) + \frac{1}{N(A)} (R - Q(A))$$
end
```