Multi-armed Bandits: Reinforcement Learning

Sanyam Singhal

April 5, 2022

1 Summary

1. k-armed bandits is an evaluative feedback problem. In this, we choose one out of k possible actions at each time-step over a course of multiple time-steps and for each action selection the environment generates a numerical reward sampled from the reward distribution associated with that action.

2. The mean of the reward distribution of an action is called its value. The action selected at the time step t is denoted as A_t and the reward obtained is denoted as R_t, then the value of an action a is defined as:

$$q_*(a) = \mathbb{E}[R_t|A_t = a] \tag{1}$$

3. Empirical estimate of $q_*(a)$ is denoted as $Q_t(a)$. When we select the action for which this empirical estimate is highest at any given step, we are said to be exploiting our current knowledge and such actions are called greedy actions. If we perform non-greedy actions, we are said to exploring. Since estimates are based on finite samples, it is essential that we sample each infinitely often to get accurate mean reward for each action. So, this means we have to suitably balance exploration (to sample each action many times) and exploitation (to maximize cumulative reward).

4. Law of large numbers suggests the following method (the sample-average method) to estimate $q_*(a)$.

$$Q_t(a) = \frac{\sum_{i=1}^{t-1} R_i \mathbb{I}_{A_i=a}}{\sum_{i=1}^{t-1} \mathbb{I}_{A_i=a}} \tag{2}$$

   We can have greedy action selection scheme where at every time step we choose action $A_t = arg\max_a Q_t(a)$ but to ensure some exploration we can extend this to what are called ϵ-greedy methods where at every time step, we explore with a probability ϵ and exploit with a probability $1 - \epsilon$. The ϵ in general, can vary with time.

5. For stationary problems, i.e. those where the reward distribution for each action is fixed wrt time, ϵ-greedy methods guarantee convergence to actual means due to law of large numbers.

6. **Basic Bandit Algorithm**

   (a) **Initialize**:
   For all $a = 1$ to k:
   $Q(a) \leftarrow 0$
   $N(a) \leftarrow 0$

   (b) **Loop indefinitely**: //(practically a large number of times)
   $A \leftarrow arg\max_a Q(a)$ with probability $1 - \epsilon$.
   $A \leftarrow$a random action with probability ϵ
   $R \leftarrow$bandit(A) //(environment generates a reward)
   $N(A) \leftarrow N(A) + 1$
   $Q(A) \leftarrow Q(A) + \frac{1}{N(A)}[R - Q(A)]$.

The value of Q for each action is our evaluated action value.

7. We can optimize our estimation method to consume lesser memory and computation by slight algebraic manipulation and converting it into a recursive update. Let Q_n be the estimate of an action after it has been selected $n-1$ times and let R_i be the reward received after the i^{th} selection of this action. Then:

$$Q_{n+1} = \frac{1}{n}\sum_{i=1}^{n} R_i = Q_n + \frac{1}{n}[R_n - Q_n] \tag{3}$$

Note that this brings the time and space complexity for updation from asymptotically linear to asymptotically constant.

8. In general, we can value estimation methods of the kind

$$\text{New Estimate} = \text{Old Estimate} + \text{Step Size}[\text{Target} - \text{Old Estimate}] \tag{4}$$

Step size at a time step t is often denoted as α_t.

9. Conditions for the step size sequence for the convergence of the value estimates are:

$$\sum_{n=1}^{\infty} \alpha_n(a) = \infty \tag{5}$$

$$\sum_{n=1}^{\infty} \alpha_n^2(a) < \infty \tag{6}$$

Non-converging sequences can be used to tackle non-stationary problems.

10. Since our value estimation methods depend on our initial guess of the action value estimates, we can use this to incorporate prior knowledge or to encourage exploration. This method is called optimistic values initialization. It is only suited for stationary problems as initial knowledge is inconsequential in a non-stationary problem.

11. In ϵ-greedy methods, the non-greedy methods are selected randomly during exploration. Moreover, the greedy actions need not be the actually optimal action. The uncertainties in the estimates of the action-values depend on the number of times that action has been selected (law of large numbers). So, we can make a score that involves current estimate of action-value as well as the sampling frequency dependent uncertainty. One such examples is the UCB (Upper Confidence Bound) score based action selection. Action taken at a time step t is given by:

$$A_t = arg\max_a \left[Q_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}} \right] \tag{7}$$

Here, $N_t(a)$ is the number of times action a was selected prior to time t. If $N_t(a) = 0$ then action a is considered to be the maximizing action.

The quantity being maxed over is like an upper bound on the possible true value of the action a.

12. We can also consider a learning that is probabilistic at every time steps but the probabilities are assigned based on a numerical preference for each state and these numerical preferences are some function of the rewards sampled after those actions. The numerical preferences are updated as stochastic gradient ascent so this class of methods is called the Gradient Bandit algorithms. Let $H_t(a)$ be the numerical preference of action a then we follow soft-max distribution to determine the probability of choosing the action a.

$$\mathbb{P}(A_t = a) = \frac{e^{H_t(a}}{\sum_{b=1}^{k} e^{H_t(b}} = \pi_t(a) \tag{8}$$

Let the average reward accumulated until (including) time t be $\bar{R}$. This serves as a baseline because actions that give reward greater than this should get higher preference and likewise a lower preference if the reward is lesser than the baseline. Other actions must move in the opposite direction.

$$H_{t+1}(A_t) = H_t(A_t) + \alpha(R_t - \bar{R}_t)(1 - \pi_t(A_t)) \tag{9}$$

$$H_{t+1}(a) = H_t(a) - \alpha(R_t - \bar{R}_t)\pi_t(A_t) \text{ for all } a \neq A_t \tag{10}$$