Exercise Solution for Chapter 2

Nanxi Zhang

July 22, 2020

1. Haven't figure out how to analyze the algorithm quantatively
2. The pseudocode is as follows:

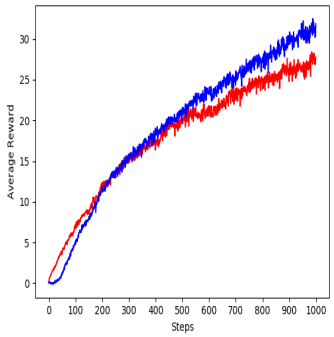
Algorithm 1 Greedy Algorithm

Initialization: $q = 0, q \in \mathrm{R}^m, N = 0, N \in \mathrm{R}^m$
for $t = 1, 2, ..., n$ **do**
    $a \leftarrow argmax_i q_i$(Ties are broken arbitrarily)
    Take action a, and recieve a reward R
    $N_a = N_a + 1$
    $q_a = q_a + \frac{1}{N_a}(R - q_a)$

3. Suppose the step size at step k is α_k. According to the updating rule, we have that

$$\begin{aligned} Q_{k+1} =& Q_k + \alpha_{k+1}(R_{k+1} - Q_k) \\ =& (1 - \alpha_{k+1})Q_k + \alpha_{k+1}R_{k+1} \\ =& (1 - \alpha_{k+1})(Q_{k-1} + \alpha_k(R_k - Q_{k-1})) + \alpha_{k+1}R_{k+1} \\ =& \Pi_{i=1}^{k+1}(1 - \alpha_i)Q_1 + \sum_{t=1}^{k}[\Pi_{j=t+1}^{k+1}(1 - \alpha_i)]\alpha_t R_t + \alpha_{k+1}R_{k+1} \end{aligned}$$

4. Code is in the file, The plot is as follows. The blue line is the average result of constant stepsize, the red line represent the changing step size. In this nonstationary enviroment, the average reward of constant stepsize is superior to the other.



5. Reason for spikes at the beginning:
   When the initial action value is set to be larger than its mean, agent might pull a good arm by chance and then update the estimates, the estimate will decrease with large probability. since it's the begining phase, some worse arms might haven't been played thus their action value haven't been updated, therefore larger than the updated good arms. Under the greedy frame, the agent will play these "worse" arms and update. Hence results in the spikes in the figure.
   Possible way to improve: If at the initialization, assign larger action values to better arms and smaller value to worse arm.
   Possible way to worsen the case: Action values assigned to worse arms are larger than arms with higher expected reward.

6. Setting 1:State unobservable.Suppose a mixed strategy is adopted by the agent.p_1 is the probability the agent take action 1. Thus the expected reward is

$$p_1[0.5 \times 0.1 + 0.5 \times 0.9] + (1 - p_1)[0.5 \times 0.2 + 0.5 \times 0.8] = 0.5$$

   which is irrelevant to p_1. Thus any strategy will result in the same expected reward
   Setting 2: When the situation is observable to the agent but the parameter of each action is unknown . Then he should adopt a learning strategy. The learning goal is the parameter for each action under both cases, which is to say, there are 2 sets of parameters to be learnt, one for case A and one for case B. Algorithms like UCB, Thompson sampling can be applied in this case .