Data mining for networks

Reinforcement Learning : The multi-armed bandit

Multi-armed bandit

1. In ε-greedy action selection, for the case of two actions and ε = 0.5, what is the probability that the greedy action is selected?

2. Consider a k-armed bandit problem with k = 4 actions, denoted 1, 2, 3, and 4. Consider applying to this problem a bandit algorithm using Ɛ -greedy action selection, sample-average action-value estimates, and initial estimates of Q1(a) = 0, for all a. Suppose the initial sequence of actions and rewards is A1 = 1, R1 = 1, A2 = 2, R2 = 1, A3 = 2, R3 = 2, A4 = 2, R4 = 2, A5 = 3, R5 = 0. On some of these time steps the Ɛ case may have occurred, causing an action to be selected at random.
   a. On which time steps did this definitely occur?
   b. On which time steps could this possibly have occurred?

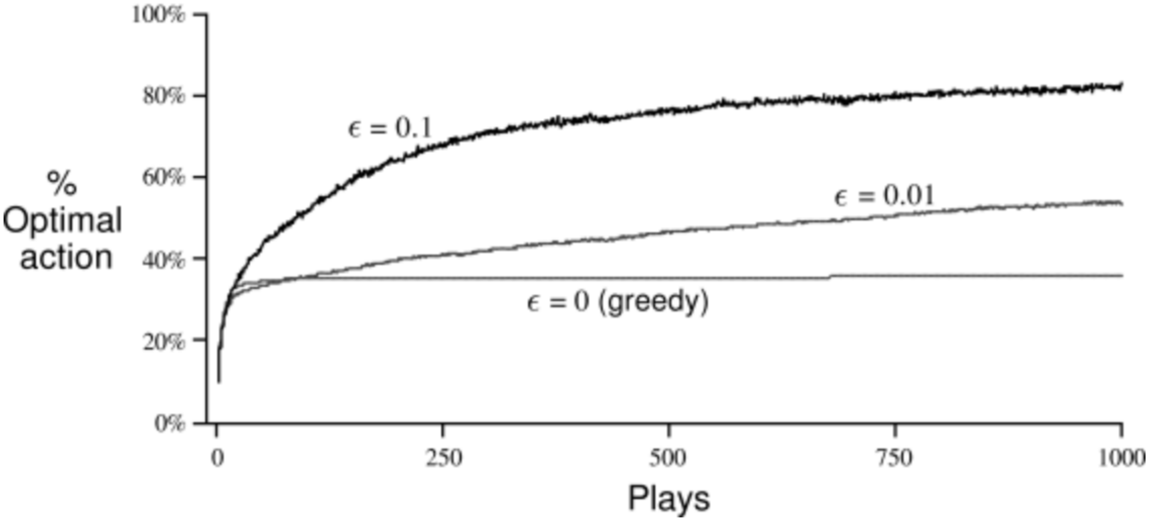
3. Implement the following two bandit algorithms:
   a. **Incremental Uniform.** This algorithm repeatedly loops through the arms pulling each arm once each time through the loop. The number of pulls for any two arms will never differ by more than 1. The average rewards for each arm are tracked and for the simple regret objective, the arm with the best average reward is returned as the best arm.
   b. **Ɛ-Greedy.** This is the Ɛ-Greedy algorithm from the course notes, where 0 < Ɛ < 1 is a parameter to the algorithm. The arm that currently looks best is selected with probability 1- Ɛ and otherwise a random arm is selected. Note that if there are k arms, then the (1/k)-Greedy algorithm will behave very much like Incremental Uniform, it is just a randomized version of that approach.

   Apply these algorithms to the following cases:
   i. Create a bandit with 10 arms. Nine of the arms should have parameters (0.05; 1) meaning they always return 0.05 as the reward. The remaining arm should have parameters (1; 0.1) meaning that 10% of the time it returns a reward of 1. This later arm has twice the expected reward of the others.
   i. Create a bandit with 20 arms. The i'th arm (for i = 1; … ; 20) should have parameters (i/20; 0.1). This models a situation where all arms give infrequent rewards that range the entire spectrum of magnitudes.

   Make the number of loops and the value of Ɛ vary and see to what action value it converges.

4. (Non-stationary Problem) If the step-size parameters α_n are not constant, then the estimate Q_n is a weighted average of previously received rewards with a weighting different from that given by $Q_{n+1} = Q_n + \alpha[R_n - Q_n]$. What is the weighting on each prior reward for the general case, analogous to $Q_{n+1} = Q_n + \alpha_n[R_n - Q_n]$, in terms of the sequence of step-size parameters?

5. In the comparison shown in the Figure below, which method will perform best in the long run in terms of cumulative reward and probability of selecting the best action? How much better will it be? Express your answer quantitatively.



6. We have used sample averages to estimate action values because sample averages do not produce the initial bias that constant step sizes do. However, sample averages are not a completely satisfactory solution because they may perform poorly on nonstationary problems. Is it possible to avoid the bias of constant step sizes while retaining their advantages on nonstationary problems? One way is to use a step size of $\gamma_t = \frac{\alpha}{\omega_t}$, where $\gamma > 0$ is a conventional constant step size, and ω_t is a trace of one that starts at 0:

$$\omega_{t+1} = \omega_t + \alpha\,(1 - \omega_t) \text{ for } t > 0 \text{ with } \omega_1 = \alpha$$

Carry out an analysis to show that γ_t is an exponential recency-weighted average without initial bias.