EE 325: Probability and Random Processes

Assignment 3

Due: 1159pm on Monday, 03 October

There are three biased coins, named A, B, and C, with unknown biases designated, respectively, p_A, p_B, and p_C. You are allowed N tosses and you have to maximise the total number of heads in the N tosses. If you you knew the p, then the best thing to do would be to toss the coin with the highest p, say p^*, and obtain an expected reward of Np^* heads. Since you do not know the p, let us explore two algorithms to use in maximising the expected reward. Evaluate for the following choices of the ps. (i) $p_A = 0.2, p_B = 0.4, p_C = 0.7$ (ii) $p_A = 0.45, p_B = 0.5, p_C = 0.58$. We will use $N = 20, 100, 1000, 5000$.

1. **Algorithm A:** Fix $N_1 < N$ and toss each coin $N_1/3$ times. Let n_A, n_B, and n_C be the number of heads obtained for coins A, B, and C. For the remaining $N - N_1$ toss the one with the highest n. The issue here is what is the best N_1.

   (a) If N_1 is small then, intuitively, the wrong coin may be chosen with higher probability. In fact, conditioned on N_1 and (p_A, p_B, p_C), the probability of choosing the wrong coin after N_1 samples can be determined. Determine that expression.

   (b) If N_1 is large then there is not enough time to use the more reliable information collected in the first N_1 times. Let $R(N_1)$ be the expected number of heads in the N tosses for a choice of N_1. Of course, this also depends on the p and N.

   Peform the following computation experiment. For each of the eight cases, simulate the above algorithm for every $N_1 < N$ 1000 times and find the sample average for $R(N_1)$.

   (a) Plot the sample average $R(N_1)$ vs N_1 for the 8 cases and point out the optimum N_1.

   (b) For each of the cases, compute the (theoretical) probability that the wrong coin will be picked after N_1 tosses. Plot the theoretical and the empirical probabilities as a function of N_1.

   (c) Submit all the programs.

2. **Algorithm B:** We will use Hoeffding's inequality as follows. After k tosses let $n_A(k)$, $n_B(k)$, and $n_C(k)$ be the number of times coins A, B, and C were used and let $k_A(k)$, $k_B(k)$, and $k_C(k)$ be the number of times the corresponding coins tossed heads; $n_A(k) + n_B(k) + n_C(k) = k$.

Consider coin A. Although we do not know p_A, we can use $n_A(k)$ and $k_A(k)$ in Hoeffding's inequality to determine at any time k, we can obtain an upper bound on p_A with a reasonable amount of confidence. Denote this upper bound by $UCB_A(k)$. Specifically, use Hoeffding's inequality to calculate $UCB_A(k) = \frac{k_A(k)}{n_A(k)} + X_A$ such that $\mathsf{Pr}(p_A \leq UCB_A | n_A, k_A) \geq (1 - \alpha)$. Similarly, calculate $UCB_B(k)$ and $UCB_C(k)$. For the $(k+1)$-th toss choose the coin with the highest $UCB(k)$. If there is a tie, break it randomly. This is an elementary learning algorithm where you adaptively learn to use the best coin. This algorithm also has many nice properties that a more full fledged course will explore in detail.

Write a program to implement this algorithm. For $\alpha = 0.1, 0.05$ and $\alpha = 0.01$ and the values p and N as in the previous algorithm, submit the following plots.

(a) Plot the sample average of $k_A(k)/k$, $k_B(k)/k$ and $k_C(k)/$ as a function of k for $N = 5000$ for each combination of the parameter values.

(b) For each combination of the parameter values, tabulate the sample average of the total reward over the N trials and compare with the best expected value.

(c) Submit the numerical results and a discussion on the effect of α, N, and the p.