

EE 325: Probability and Random Processes

Programming Assignment 3

Due: 1159pm on Thursday, 10 October

A video recommendation system has classified the set of videos from which it recommends into K types. Consider the following version of the recommendation. A user comes to the system, receives a recommendation of one of the K types of videos, either clicks on it and watches it **or** just ignores the recommendation and goes away. All users are assumed homogenous and will click on a video of type k with probability p_k . Furthermore if a user clicks on a video, the system makes a random revenue of R_k which is uniformly distributed in the interval $[0, a_k]$. Assume that the users' propensity to click on the recommended video and the random revenue are independent of everything.

The system is allowed to recommend N users and wants to maximize its revenue from these users. If the p_k and the distribution of R_k were known, then the best thing to always recommend

$$k^* := \arg \max_{1 \leq k \leq K} E(p_k R_k)$$

i.e., recommend from the class of videos that maximizes the expected revenue.

Since the system does not know the p_k and the a_k , it considers two algorithms to use in maximizing its expected revenue from the N users. We will consider the following three systems for $N = 1000, 10,000$.

1. System 1: $K = 4, p_1 = 0.2, p_2 = 0.4, p_3 = 0.6, p_4 = 0.65, a_1 = a_2 = a_3 = a_4 = 2$.
2. System 2: $K = 4, p_1 = 0.2, p_2 = 0.4, p_3 = 0.6, p_4 = 0.65, a_1 = 2, a_2 = a_3 = 2.5, a_4 = 3$.
3. System 3: $K = 4, p_1 = 0.2, p_2 = 0.4, p_3 = 0.6, p_4 = 0.65, a_1 = 8, a_2 = a_3 = a_4 = 2$

1. **Algorithm A:** Assume System 1 for this algorithm. Fix $N_1 < N$ and a video from each class N_1/K times. Let R_k be the revenue earned from videos of type k . For the remaining $(N - N_1)$ users, recommend videos from the class that gave you the maximum revenue. The issue here is what is the best N_1 .

- (a) If N_1 is small then, intuitively, the wrong type may be chosen with higher probability. In fact, conditioned on N_1 and $\{p_k\}$ the probability of choosing the wrong type 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 from the first N_1 users. Let $R(N_1)$ be the expected revenue from the N users for a choice of N_1 . Of course, this also depends on the p_k and N .

Perform 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 system, compute the (theoretical) probability that the wrong type of video will be picked after N_1 users. Plot the theoretical and the empirical probabilities as a function of N_1 .
- (c) Submit all the programs.

2. **Algorithm B:** You can use this algorithm on all the three systems. We will use Hoeffding's inequality as follows. After s users have visited the system, let $n_k(s)$, be the number of times type k videos have been recommended, $m_k(s)$ be the number of times type k videos have been clicked, and $R_k(s)$ be the revenue from the type k videos.

First consider only System 1 and consider type k videos. Although we do not know p_k , we can use $n_k(s)$ and $m_k(s)$ in Hoeffding's inequality to determine for any s , upper bounds on p_k with a reasonable amount of confidence. Denote these upper bound by $UCB_k(s)$. Specifically, use Hoeffding's inequality to calculate $UCB_k(s) = \frac{m_k(s)}{n_k(s)} + X_k$ such that $\Pr(p_k \leq UCB_k(s) | n_k(s), m_k(s)) \geq (1 - \alpha)$. For the $(s + 1)$ -th user choose the type with the highest $UCB(s)$. If there is a tie, break it randomly. This is an elementary learning algorithm where you adaptively learn the best recommendation. 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 $p_k(s)$ and $R_k(s)$ as a function of k for $N = 10,000$.
- (b) Tabulate the sample average of the total reward from the N users and compare with the best expected value.
- (c) Submit the numerical results and a discussion on the effect of α , N , and the p .
- (d) Now adapt the preceding algorithm for System 2 and System 3 when the a_k are different and unknown and repeat the preceding set of plots.
- (e) A key assumption in this exercise is that the users do not change their preferences as the s increases from 1 to N . How realistic is that assumption? Suggest ways to capture the effect of the recommendation sequence on the change in p_k .

Using Hoeffding's Inequality

The following may be helpful for the second algorithm described.

- A coin has an unknown bias p .
- The coin has been tossed n , let Y_n be the number of times a head has been observed.
- Note that $E(Y_n) = np$ and $E(Y_n/n) = p$.
- Hoeffding's Inequality tells us that

$$\begin{aligned}\Pr(Y_n - E(Y_n) \geq y) &\leq e^{\frac{-2y^2}{n}} \\ \Pr(Y_n - np \geq ny) &\leq e^{-2ny^2} \\ \Pr\left(p - \frac{Y_n}{n} \geq ny\right) &\leq e^{-2ny^2} \\ \Pr\left(p - \frac{Y_n}{n} \geq y\right) &\leq e^{-2ny^2}\end{aligned}$$

- This bounds the probability.
- Let us now see how to use this result in a practical setting.
- For example, let $e^{\frac{-2y^2}{n}} = 0.01$. This corresponds to $y \approx 2.14n$
- What can we say about the true value of p based on this observation?