ESE 545: Data Mining **Fall 2018**

Project 4 Writeup

Name: *Joseph Lauigan*

1. **Problem 1**

   The problem formulation is such that, at each time t, we are to recommend one category A_i to the user and observe the reward for that category ($r_{i,t} \in [0, 1]$). We want to maximize the reward (the total number of clicks through all rounds), and such this is a multi-bandit problem. Some algorithms are based on the loss in each round, which will be denoted by $l_{i,t} = 1 - r_{i,t}$, and will minimize the loss over all rounds. As we are only allowed to present the user one category during each round, this is not a semi-bandit problem, as we are not presenting the user with multiple categories in each round and building statistics on sets of categories. We are to build the statistics of each category individually, and in an online fashion.

   The metric of "correctness" is the regret of the algorithm - how well that algorithm fared in a particular round t compared to the optimal arm, summed over all rounds.

2. **Problem 2**

   For the partial feedback setting, the EXP3 algorithm was employed, with the parameter $\eta = \frac{1}{\sqrt{T}}$, where T is the total number of rounds. The EXP3 algorithm only updates the arm that incurred the loss in a given round t, thus this falls within a partial feedback setting for the multi-armed bandit problem. The regret was averaged over 100 repetitions. **The total average regret is 1262.60**

3. **Problem 3**

   For the full feedback setting, the multiplicative-weight update algorithm was implemented. The regret was averaged over 100 repetitions. **The average regret is 675.91**

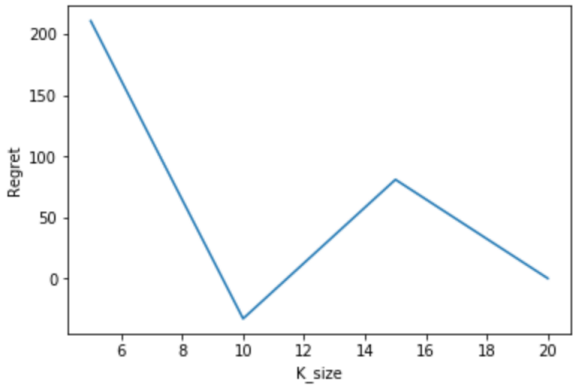
4. **Problem 4**

   To achieve smallest regret in the partial feedback setting, the problem will be reformulated as a semi-bandit problem rather than a true multi-armed bandit problem. Now, the algorithm will still only put forward one category to serve to the user in each round, but we can choose a subset k of the categories to get feedback on. The following algorithm is a variant of the EXP4 algorithm - which is known as the contextual bandit problem. The main difference between this algorithm and EXP4 is that the algorithm itself holds a probability distribution over all the arms, and weights the experts probability distributions by this and then picks an arm. The algorithm updates this distribution based on the EXP3 update.

   The basic idea for this algorithm is as follows: we will have k-experts, which will be their own EXP3 multi-armed bandit algorithms that hold their own probability distributions over the arms in each

round - these distributions will be accessible to the algorithm. In each round t, the algorithm will sum the distributions over all the experts and weight it by it's own probability distribution over the arms. The distribution is then normalized and then sampled from to choose an arm/ad for that round. A loss is incurred for the algorithm and it's distribution is updated according to the EXP3 update step. Then each expert's distribution will be sampled from to produce an arm/ad recommendation. These recommendation's will incur a loss, and each expert's distribution will be updated according to the EXP3 update. For any round t, the most recommendations that will have a loss revealed to the algorithm is k+1 - so this is still a partial feedback setting.

The EXP3 update step has the η parameter, which is set to $\frac{1}{\sqrt{T}}$ where T is the number of rounds. The other free parameter is k, the number of experts. A sweep was done plotting Regret vs Num. of Experts



From this graph, the optimal number of experts was set to $k = 10$. Since the regret is stochastic, the regret was averaged over 100 repetitions. **The average regret achieved is 1.96**