Algorithms Under Uncertainty : Quiz 3
Full Marks : 15

11/1/2023

Write solutions in the space provided. NO extra pages will be provided. Write brief and precise solutions. Meaningless rambles fetch negative credits.

Problem 1. (10 points) Recall the ski-rental problem. You can rent a ski for 1 unit of money per day or buy it for B units of money and use it forever. The algorithm does not know how long the ski season is going to last and hence makes a decision each day (of course, once you buy, there is no decision to be made). Use Yao's principle to whow that any *randomized* algorithm must have a competitive ratio of at least 4/3.

- a) (3 points) Clearly state the input distribution you will choose as an oblivious adversary. Remember you cannot assume that you know the moves of the deterministic algorithm. (Hint: The distribution needs to be bi-modal , that is you need two deterministic instances, each occurring with a certain probability)

  Solution. There are several ways to show this. I am showing one of them. So recall that after you fix a B (the buying cost), the only other parameter that uniquely defines a problem instance is the number of days that it will snow - suppose we denote it by D. Now, we pick the following input distribution - $\mathbb{P}[D = 1] = 1/2$ and $\mathbb{P}[D = B] = 1/2$.

- b) (3 points) What is an upper bound on the expected cost of the optimal solution ?

  Solution. It is very easy to see that if $D = 1$, then optimal solution is 1. If $D = B$, the optimal solution is B. Hence, the expected value of the optimal solution is $(B + 1)/2$.

- c) (4 points) What is a lower bound on the expected cost of any deterministic online algorithm and hence what is the lower bound on the competitive ratio ?

  Solution. Any deterministic algorithm can be parameterized by $1 \leq k \leq B$ such that the algorithm rents for $k - 1$ days and then buys. Let us figure out the expected cost of this algorithm.

  - If $k = 1$: The algorithm always pay B irrespective of the instance
  - If $k > 1$: The algorithm pays 1 if $D = 1$ and $k - 1 + B$ if $D = B$. Hence, the expected cost is $1/2 + (k - 1 + B)/2 = (k + B)/2$

  Remember, Yao's lemma requires us to compare the ratio of the expected cost of the *best possible* deterministic algorithm on the chosen input distribution to the expected cost of the optimal solution. It is easy to see from the above calculations that the best deterministic expected cost is achieved by the algorithm with $k = 2$. Hence, the lower bound on any randomized algorithm is $(B + 2)/(B + 1)$ which can be as large as 4/3.

Problem 2. (5 points) Show that randomization is necessary for obtaining a no-regret algorithm for the $\{0, 1\}$-experts problem (the one with 'mistakes'). As usual, let n be the number of experts and T be the number of rounds.

- a) (4 points) Design a family of sequences such that, given any T, any deterministic algorithm suffers T many mistakes but the best expert suffers at most T/n many mistakes. (family means for each deterministic algorithm, there will be at least one bad sequence in the family).

  Solution. We design the following adversarial sequence. Whichever expert the algorithm chooses, we make it commit a mistake and all the others predict correctly (in other words, the loss vector will be all 0's except the coordinate corresponding to the expert selected by the algorithm). Clearly, the algorithm makes T mistakes. Now, think of a matrix with the loss vectors on each day as columns and each row denoting an expert. Then, this matrix has T-many 1s. By PHP, one row must have at most T/n many 1s. Hence, the best expert commits at most T/n mistakes.

- b) (1 point) Why does the above example imply the required result ?

  Solution. Choosing $n = 2$ shows that regrest is T/2 which is linear. Hence, randomization is necessary for sub-linear regret.