CS 170 HW 9

Due **2020-11-02, at 10:00 pm**

1 Study Group

List the names and SIDs of the members in your study group. If you have no collaborators, you must explicitly write none.

In addition, we would like to share correct student solutions that are well-written with the class after each homework. Are you okay with your correct solutions being used for this purpose? Answer "Yes", "Yes but anonymously", or "No"

2 How to Gamble With Little Regret

Suppose that you are gambling at a casino. Every day you play at a slot machine, and your goal is to minimize your losses. We model this as the experts problem. Every day you must take the advice of one of n experts (i.e. play at a slot machine). At the end of each day t, if you take advice from expert i, the advice costs you some c_i^t in $[0, 1]$. You want to minimize the regret R, defined as:

$$R = \frac{1}{T}\left(\sum_{t=1}^{T} c_{i(t)}^t - \min_{1\le i\le n} \sum_{t=1}^{T} c_i^t\right)$$

where $i(t)$ is the expert you choose on day t. Notice that in this definition, you are comparing your losses to the best expert, rather than the best overall strategy.

Your strategy will be probabilities where p_i^t denotes the probability with which you choose expert i on day t. Assume an all powerful adversary (i.e. the casino) can look at your strategy ahead of time and decide the costs associated with each expert on each day. Let C denote the set of costs for all experts and all days. Compute $\max_C(\mathbb{E}[R])$, or the maximum possible (expected) regret that the adversary can guarantee, for each of the following strategies.

(a) Any deterministic strategy, i.e. for each t, there exists some i such that $p_i^t = 1$.

(b) Always choose an expert according to some fixed probability distribution at every time step. That is, fix some $p_1, \ldots, p_n$, and for all t, set $p_i^t = p_i$.

What distribution minimizes the regret of this strategy? In other words, what is $\text{argmin}_{p_1,\ldots,p_n} \max_C(\mathbb{E}[R])$?

This analysis should conclude that a good strategy for the problem must necessarily be randomized and adaptive.

3 Variants on the Experts Problem

Consider the experts problem: We have n experts who predict it will either rain or shine tomorrow, one of which is always correct. Every day we guess based on the experts if it will rain or shine tomorrow. Our goal is to make as few mistakes in our predictions as possible.

(a) Suppose we always guess the prediction of the majority of experts who have been correct so far, breaking ties arbitrarily (Note that this set is always non-empty since one expert is always correct). Show that we make at most $\log n$ mistakes using this strategy.

(b) Suppose there are now k experts who are always correct instead of just one. Give a better upper bound on the number of mistakes the algorithm from the previous part makes makes.

(c) Suppose instead of one expert always being correct, we are only guaranteed that there is one expert who makes at most k mistakes. Consider the following algorithm: Let m be the least mistakes made by any expert so far. Use the prediction of the majority of experts who have made at most m mistakes so far.

Give an upper bound on the number of mistakes made by this algorithm.

4 (Optional MT2 Practice) Adding Many Edges At Once

Given an undirected, weighted graph $G(V, E)$, consider the following algorithm to find the minimum spanning tree. This algorithm is similar to Prim's, except rather than grow out a spanning tree from one vertex, it tries to grow out the spanning tree from every vertex at the same time.

procedure FINDMST($G(V, E)$)
  $T \leftarrow \emptyset$
  while T is not a spanning tree **do**
    Let $S_1, S_2 \dots S_k$ be the connected components of the graph with vertices V and edges T
    For each i, let e_i be the minimum-weight edge with exactly one endpoint in S_i
    $T \leftarrow T \cup \{e_1, e_2, \dots e_k\}$
  return T

For example, at the start of the first iteration, we'll have every vertex in its own S_i.

For simplicity, in the following parts you may assume that no two edges in G have the same weight.

(a) Show that this algorithm finds a minimum spanning tree.

(b) Give an exact upper bound (that is, an upper bound without using Big-O notation) on the worst-case number of iterations of the while loop in one run of the algorithm.

(c) Using your answer to the previous part, give an upper bound on the runtime of this algorithm.

5 (Optional MT2 Practice) Huffman Proofs

(a) Prove that in the Huffman coding scheme, if some symbol occurs with frequency more than $\frac{2}{5}$, then there is guaranteed to be a codeword of length 1. Also prove that if all symbols occur with frequency less than $\frac{1}{3}$, then there is guaranteed to be no codeword of length 1.

(b) Suppose that our alphabet consists of n symbols. What is the longest possible encoding of a single symbol under the Huffman code? What set of frequencies yields such an encoding?

6 (Optional MT2 Practice) Bimetallism

There is a blacksmith who can produce n different alloys, where alloy i sells for p_i dollars per unit. One unit of alloy i takes g_i grams of gold and s_i grams of silver to produce. The blacksmith has a total of G grams of gold and S grams of silver to work with, and can produce as many units of each type of alloy as they want within the material constraints. The blacksmith is allowed to produce and sell a non-integer number of units of each alloy.

1. Formulate the linear program to maximize the revenue of the blacksmith, and explain your decision variables, objective function, and constraints.

2. Formulate the dual of the linear program from part (a), and explain your decision variables, objective function, and constraints. The explanations provide economic intuition behind the dual. **Hint:** Formulate the dual first, then think about it from the perspective of the blacksmith when negotiating prices for buying G grams of gold and S grams of silver if they had already signed a contract for the prices for the output alloys p_i. Think about the breakeven point, from which the blacksmith's operations begin to become profitable for at least one alloy.

7 (Optional MT2 Practice) Max Flow Short Answer

(a) True or false: The following problem is equivalent to max-flow. Rather than try to push as much flow as possible from s to t, our goal is to find the largest c such that we can divide all the capacities by c and still be able to route 1 unit of flow from s to t. Justify your answer.

(b) We are given a graph where all edges have infinite capacity and a length ℓ_e. Our goal is to route one unit of flow from s to t such that $\sum_e \ell_e f_e$ is minimized. True or false: The smallest value of $\sum_e \ell_e f_e$ we can achieve is the length of the shortest path. Justify your answer.

(c) Suppose we didn't use a residual graph in max-flow, and instead just used the following algorithm: While there is a path from s to t in the original network such that every edge is not at capacity, push as much flow as possible through any such path.

Give a small example of a graph with unit capacities where this algorithm may only find a flow of size 1, but the max flow has size at least 2.

(d) Ford-Fulkerson needs at most F iterations to finish, where F is the size of the max-flow. Give an example of a graph where Ford-Fulkerson could take F iterations, but choosing the right path to push flow on causes it to finish in 2 iterations.

8 (Optional MT2 Practice) Zero-Sum Games Short Answer

(a) Suppose a zero-sum game has the following property: The payoff matrix M satisfies $M = -M^\top$. What is the expected payoff of the row player?

(b) True or False: If every entry in the payoff matrix is either 1 or -1 and the maximum number of 1s in any row is k, then for any row with less than k 1s, the row player's optimal strategy chooses this row with probability 0. Justify your answer.

(c) True or False: Let M_i denote the ith row of the payoff matrix. If $M_1 = \frac{M_2+M_3}{2}$, then there is an optimal strategy for the row player that chooses row 1 with probability 0.