CS 170 Homework 14

Due **Saturday 12/6/2025, at 10:00 pm (grace period until 11:59pm)**

1 Study Group (1 point)

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

2 Monte Carlo Games (9 points)

Let's suppose we have a Monte Carlo algorithm (a randomized algorithm that has a deterministic bound on its runtime, but which only outputs the correct answer some of the time). Call this algorithm A; then $A(x, r)$ is the output of A on input x and random bits r. In this question, we will think of A as a distribution over many deterministic algorithms. Convince yourself that this makes sense: after all, if we fix a setting of the random bits r, we get $A_r(x)$, which is a deterministic algorithm (which may be wrong on some inputs). Let's fix a set of algorithms S (say, polynomial-time algorithms). Note that A has whatever property defines S if and only if it is a distribution over only algorithms in S (for example, we say a Monte Carlo algorithm is polynomial time if and only if it runs in polynomial time for all settings to the randomness, which is equivalent to all the deterministic algorithms in its distribution running in polynomial time).

We will define a function $c(a, x)$ which indicates whether the deterministic algorithm $a \in S$ is correct on input x; $c(a, x) = 1$ if a is correct on input x, and 0 if it is incorrect.

Let's use this function to define a zero-sum game; the row player will choose a, and the column player will choose x; then a payoff of $c(a, x)$ will go to the row player.

(a) (3 points) Describe the action and goal of the row and column players. Interpret these in the setting of 'correctness of the randomly chosen algorithm' that we constructed the game from.

Hint: Since $c(a, x)$ *is an indicator,* $\mathbb{E}[c(a, x)] = \Pr[c(a, x) = 1]$.

(b) (6 points) Using zero-sum game duality in conjunction with your interpretation above, what can we say about a problem if we know there exists a polynomial-time randomized algorithm that is correct with probability 2/3 on all inputs? What can we say if we know that there is a distribution of inputs on which no deterministic algorithm is correct with probability 2/3?

Hint: use the fact that a randomized algorithm induces a distribution over deterministic algorithms.

This content is protected and may not be shared, uploaded, or distributed.

3 Informed Predictions, Randomized! (11 points)

Define the *realizable* experts problem to be as follows: there are n experts who each make a prediction every day, and we need to use the experts' predictions to make our own daily prediction. Additionally, there is guaranteed to be at least one *perfect* expert that is always correct. In class, we found that the Halving Algorithm (which is deterministic) makes at most $\log_2 n$ mistakes.

Describe a randomized algorithm that, in expectation, makes at most $\log_e n = \ln n$ mistakes. Prove that your algorithm, indeed, makes $\ln n$ mistakes in expectation.

Hint: you may use the identity $\sum_{i=1}^{n} \frac{1}{i} \leq \ln n + 1$.

4 Variants on the Experts Problem (10 points)

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

(a) (3 points) 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.

Hint: if you're stuck, look at the Halving Algorithm from lecture.

(b) (3 points) 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.

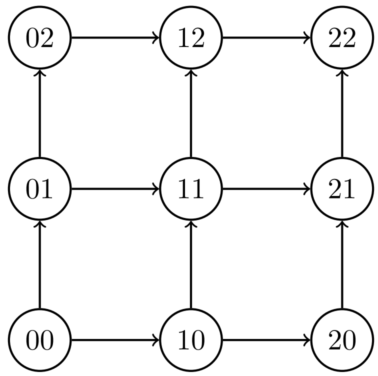
(c) (4 points) Suppose instead of one expert always being correct, we are only guaranteed that there is one expert who makes at most k mistakes. Consider the following algorithm: Let m be the least mistakes made by any expert so far. Use the prediction of the majority of experts who have made at most m mistakes so far.

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

This content is protected and may not be shared, uploaded, or distributed.

5 Multiplicative Weights (10 points)

Consider the following simplified map of Berkeley. Due to traffic, the time it takes to traverse a given path can change each day. Specifically, the length of each edge in the network is a number between $[0, 1]$ that changes each day. The travel time for a path on a given day is the sum of the edges along the path.



For T days, both Richard and Allison drive from node 00 to node 22.

To cope with the unpredictability of traffic, Allison builds a time machine and travels forward in time to determine the traffic on each edge on every day. Using this information, Allison picks the path with the smallest total travel time in the upcoming T days and uses the *same* path each day.

On the other hand, Richard wants to use the multiplicative weights update algorithm to pick a path each day. In particular, he wants to ensure that the difference between his expected total travel time over T days and Allison's total travel time is at most $T/10000$. Assume that Richard finds out the lengths of all the edges in the network, even those he did not drive on, at the end of each day.

(a) (2 points) How many experts should Richard use in the multiplicative weights algorithm?

(b) (1 point) What are the experts?

(c) (2 points) Given the weights maintained by the algorithm, how does Richard pick a route on any given day?

(d) (5 points) The regret bound for multiplicative weights is as follows:

Theorem. *Assuming that all losses for the n experts are in the range $[0, 4]$, the worst possible regret of the multiplicative weights algorithm run for T steps is $R_T \leq 8\sqrt{T \ln n}$.*

Use the regret bound to show that the expected total travel time of Richard is not more than $T/10000$ days worse than that of Allison for large enough T.

6 How to Gamble With Little Regret (10 points)

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 of their choosing). 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:

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

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, with justification.

(a) (2 points) Choose expert 1 at every step, i.e., $p_1^t = 1$ for all t.

(b) (4 points) Any deterministic strategy, i.e., for each t, there exists some i such that $p_i^t = 1$.

(c) (4 points) 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 $\operatorname{argmin}_{p_1,\ldots,p_n} \max_C(\mathbb{E}[R])$?

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

This content is protected and may not be shared, uploaded, or distributed.