TTIC 31250 An Introduction to the Theory of Machine Learning

Homework # 5 **Due: May 27, 2020**

Groundrules: Same as before. You should work on the exercises by yourself but may work with others on the problems (just write down who you worked with). Also if you use material from outside sources, say where you got it.

Note: This is the last homework assignment. Your projects are due on June 3, the last day of class.

Exercises:

1. [**Zero-sum Games**] Consider the following zero-sum game. Player A (Alice) hides either a nickel (5 cents) or a quarter (25 cents) behind her back. Then, player B (Bob) guesses which it is. If Bob guesses correctly, he wins the coin. If Bob guesses incorrectly, he has to pay Alice 15 cents. In other words, the amount that Alice wins can be summarized by the following payoff matrix:

		Bob guesses	
		N	Q
Alice hides	N	-5	$+15$
	Q	$+15$	-25

This seems like a fair game since 15 cents is the average of 5 cents and 25 cents, but we will see that one of the players in fact has an advantage.

   (a) What is the value to Alice of the strategy "with probability 1/2 hide a nickel and with probability 1/2 hide a quarter"? (The value of a strategy is its value assuming that the opponent knows it and plays a best-response to it).

   (b) What is Alice's minimax optimal strategy, and what is its value?

   (c) What is Bob's minimax optimal strategy, and what is its value to Bob?

   (d) Is it better to be Alice or Bob in this game?

Problems:

2. [**On approximate Nash equilibria**] Consider a two-player general-sum game. Let us for concreteness focus on games where each player has n actions, and use R to denote the payoff matrix for the row player and C to denote the payoff matrix for the column player. (So if the row-player plays action i and the column-player plays action j, then the row-player gets R_{ij} and the column-player gets C_{ij}. Recall that a Nash Equilibrium is a pair of distributions p and q (one for each player) such that neither player has any incentive to deviate from its distribution assuming that the other player

doesn't deviate from its distribution either. Formally, a pair of distributions p (for the row player) and q (for the column player) is a Nash equilibrium if the following holds: assuming the column player plays at random from q, the expected payoff to the row player for each row i with $p_i > 0$ is equal to the maximum payoff out of all the rows ($e_i^T Rq = \max_{i'} e_{i'}^T Rq$); and, assuming the row player plays at random from p, the expected payoff to the column player for each column j with $q_j > 0$ is equal to the maximum payoff out of all the columns ($p^T Ce_j = \max_{j'} p^T Ce_{j'}$). (Here, e_i denotes the column-vector with a 1 in position i and 0 everywhere else).

Now, assume we have a game in which all payoffs are in the range $[0, 1]$. Define a pair of distributions p, q to be an "ϵ-Nash" equilibrium if each player has *at most* ϵ incentive to deviate. That is, the expected payoff to the row player for each row i with $p_i > 0$ is within ϵ of the maximum payoff out of all the rows, and vice-versa for the column player.

Using the fact that Nash equilibria must exist, show that there must exist an ϵ-Nash equilibrium in which each player has positive probability on at most $O(\frac{1}{\epsilon^2} \log n)$ actions (rows or columns).

Hint #1: what is a good randomized way to get a sparse approximation to a probability distribution p that was handed to you?

Hint #2: your solution will require using Hoeffding bounds and the union bound.

Note: this fact yields an $n^{O(\frac{1}{\epsilon^2} \log n)}$-time algorithm for finding an ϵ-Nash equilibrium. No PTAS (algorithm running in time polynomial in n for every fixed $\epsilon > 0$) is known, however.

3. **Compression bounds.** For some learning algorithms, the hypothesis produced by running the algorithm on a training set of size n can be uniquely described by giving k of the training examples. E.g., if you are learning an interval on the line using the simple algorithm "take the smallest interval that encloses all the positive examples," then the hypothesis can be reconstructed from just being told the outermost positive examples, so $k = 2$. For a conservative Mistake-Bound learning algorithm, you can reconstruct the hypothesis produced by the algorithm by just looking at the examples on which a mistake was made, so $k \leq M$, where M is the algorithm's mistake-bound. (In this case, you would also care about the *order* in which those examples arrived.)

   Your job in this problem is to prove a PAC generalization guarantee based on k (essentially, proving that if k is small, then this is a legitimate notion of a "simple" hypothesis; these are called *compression bounds*). Specifically, assume we fix a reconstruction procedure, so that for a given sequence of examples S' we have a well-defined hypothesis $h_{S'}$. You will show that

$$\Pr_{S \sim D^n} \left(\exists S' \subseteq S, |S'| = k, \text{ such that } h_{S'} \text{ has 0 error on } S - S' \text{ but true error } > \epsilon \right) \leq \delta,$$

   so long as

$$n \geq \frac{1}{\epsilon} \left(k \ln n + \epsilon k + \ln \frac{1}{\delta} \right).$$

(a) First, prove the following easier statement. Let's use $x_1, ..., x_n$ to denote the examples in S. Now suppose you are given a sequence of indices $i_1, ...i_k$. Define $A_{i_1,...,i_k}$ to be the event that $h_{(x_{i_1},...,x_{i_k})}$ has zero error on all examples $x_j \in S$ such that $j \notin \{i_1, ..., i_k\}$ and yet the true error of $h_{(x_{i_1},...,x_{i_k})}$ is more than ϵ. Prove that if $S \sim D^n$, the probability of event $A_{i_1,...,i_k}$ is at most $(1-\epsilon)^{n-k}$.

(b) Now use this to prove the guarantee in the displayed equation above.