15-859(B) Machine Learning Theory

Homework # 6 **Due: April 20, 2009**

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.

Exercises:

1. **DFAs.** A *distinguishing sequence* for a DFA is a sequence of actions such that the observations produced from these actions uniquely determine the starting state. I.e., a sequence h such that if $q \neq q'$ then $\text{obs}(q, h) \neq \text{obs}(q', h)$. (Here, "$\text{obs}(q, h)$" is the sequence of observations produced by executing h from q.)

   A *homing sequence* for a DFA is a sequence of actions such that the observations produced from these actions uniquely determine the *ending* state. I.e., a sequence h such that if $qh \neq q'h$ then $\text{obs}(q, h) \neq \text{obs}(q', h)$. (Here, "$qh$" is the ending state produced by executing h from q.)

   (a) Describe a strongly-connected DFA that has no distinguishing sequence. Note that the definition of "$q \neq q'$" is that there must exist a sequence $h_{qq'}$ such that $\text{obs}(q, h_{qq'}) \neq \text{obs}(q', h_{qq'})$, it's just that no single h works for all pairs.

   (b) Give a homing sequence for your DFA.

2. **Online resource sharing.** Consider a system with n users and m resources. User i has permissions for some subset M_i of the m resources (if we construct a bipartite graph with users on the left and resources on the right, then these are the neighbors of user i). However, user i can only use $k_i \leq |M_i|$ of the M_i resources at a time. Finally, each resource j has a size s_j, and if several users are using a given resource, they have to split it equally. The goal of a user is to maximize total resource usage.

   Formally, the game proceeds as follows. Each user i simultaneously chooses some subset S_i of k_i out of their $|M_i|$ neighbors. Let n_j be the total number of users who choose resource j. Then, user i gets payoff $\sum_{j \in S_i} s_j / n_j$. (This is equivalent to the market-sharing game of Goemans, Li, Mirrokni and Thottan.)

   Suppose we (user i) repeatedly play this game each day. We could place this in the framework of "combining expert advice", except the number of experts $\binom{|M_i|}{k_i}$ is exponential. Show how you could instead model this in the Kalai-Vempala framework to get a polynomial-time regret-minimizing algorithm. Make sure to argue how you solve the offline problem.

Problems:

3. **Policy iteration.** The goal of this problem is to prove that policy iteration will eventually reach the optimal policy. Recall that in policy iteration, given some policy π_i, you solve the linear system to compute the state values under that policy:

$$V^{\pi_i}(s) = R(s, \pi_i(s)) + \gamma \sum_{s'} \Pr_{s,\pi_i(s)}(s') V^{\pi_i}(s').$$

(Here, "$R(s, a)$" is the expected reward of executing action a from state s.) Then, we define policy π_{i+1} to be the greedy policy with respect to those values. That is,

$$\pi_{i+1}(s) = \arg\max_a \left[R(s, a) + \gamma \sum_{s'} \Pr_{s,a}(s') V^{\pi_i}(s') \right],$$

and so on.

(a) As an easy first step, argue that if $\pi_{i+1} = \pi_i$ (i.e., $\pi_{i+1}(s) = \pi_i(s)$ for all states s), then π_i is optimal.

(b) As the harder second step, argue that the values never decrease (i.e., for all s, $V^{\pi_{i+1}}(s) \geq V^{\pi_i}(s)$). This completes the argument because there are only a finite number of different policies.

Hint: what about a hybrid policy that uses π_{i+1} for one step and then π_i from then on? How about π_{i+1} for two steps?

4. **Sample complexity bounds.** For some learning algorithms, the hypothesis produced can be uniquely described by a small subset of 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 the outermost positive examples, so $k = 2$. For a conservative Mistake-Bound learning algorithm, you can reconstruct the hypothesis 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 may also care about the *order* in which those examples arrived.)

Prove a PAC guarantee based on k. Specifically, fixing a description language (reconstruction procedure), so for a given set S' of examples we have a well-defined hypothesis $h_{S'}$, 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

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

Hint: Think of S' as a subset of indices, and imagine drawing points in S by drawing those in S' first. Also, be clear in your analysis what the events are that you are taking the union bound over.

Note the similarity of the form of this bound to VC-dimension and other bounds we have seen. These are often called "compression bounds".