15-859(B) Machine Learning Theory

Homework # 5 **Due: March 30, 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. [**SQ model**] Give an algorithm to learn the class of *decision lists* in the SQ model (and argue correctness for your algorithm). Be clear about what specifically the queries χ are and the tolerances τ. Remember, you are not allowed to ask for conditional probabilities like $\Pr[A|B]$ but you can ask for $\Pr[A \wedge B]$.

   So, combined with your results from Homework 1, this gives an algorithm for learning decision trees of size s in the SQ model with queries and τ^{-1} on the order of $n^{O(logs)}$, matching our SQ-dimension lower bounds.

Problems:

In the rest of this homework you will show that the class of polynomial-size boolean formulas is equivalent to the class NC^1. NC^1 is the class of $O(\log n)$-depth {AND, OR, NOT} circuits where each gate has at most 2 inputs. Boolean formulas are just the generalization of DNF in which we allow ANDs and ORs to appear in any order, e.g., $x_1 \vee (x_2 \wedge (\bar{x}_1 \vee x_3))$. You can think of a Boolean formula as a circuit that looks like a tree, except you are allowed to have multiple copies of each input (or equivalently, the inputs are allowed to have out-degree greater than 1). On the other hand, general circuits can be more compact than formulas since intermediate nodes may also have fanout greater than 1. This is also exercise 6.2 (p.141) in the book.

2. [**circuits and formulas part 1**] Show that for any circuit of depth d of {AND, OR, NOT} gates with fanin ≤ 2, there is an equivalent boolean formula of size $O(2^d)$. Thus, NC^1 is contained in the class of polynomial-size boolean formulas.

3. [**circuits and formulas part 2**] Show that for any boolean formula of size s, there exists an equivalent circuit of depth $O(\log s)$. Thus polynomial-size boolean formulas are contained in the class NC^1.

   Thus, this implies our hardness results for learning NC^1 (from class on Feb 23) carry over to general Boolean formulas (can't even weak-learn in poly-time over the uniform distribution, even with membership queries, under cryptographic assumptions). Note: Problem 2 should be pretty straightforward. Problem 3 is trickier.