8803 Machine Learning Theory

Homework # 1 **Due: February 2nd 2010**

This homework is due by the start of class on February 2nd. You can either submit the homework via the course page on T-Square or hand it in at the beginning of the class on February 2nd. Start early!

Groundrules:

- Your work will be graded on correctness, clarity, and conciseness. You should only submit work that you believe to be correct; if you cannot solve a problem completely, you will get significantly more partial credit if you clearly identify the gap(s) in your solution. It is good practice to start any long solution with an informal (but accurate) proof summary that describes the main idea.
- You may collaborate with others on this problem set and consult external sources. However, you must *write your own solutions* and *list your collaborators/sources* for each problem.

Problems:

1. **Due Jan. 26th.** Send an email to Nina (`ninamf@cc.gatech.edu`) with subject `8803MLT student` containing a few sentences about your background and your research interests (please include your department and graduate program).

2. **Expressivity of LTFs.**
   Assume each example x is given by n boolean features (variables). A *decision list* is a function of the form: "if ℓ_1 then b_1, else if ℓ_2 then b_2, else if ℓ_3 then b_3, ..., else b_m," where each ℓ_i is a literal (either a variable or its negation) and each $b_i \in \{0, 1\}$. For instance, one possible decision list is the rule: "if $\overline{x}_1$ then positive, else if x_5 then negative, else positive." Decision lists are a natural representation language in many settings and have also been shown to have a collection of useful theoretical properties.
   (a) Show that conjunctions (like $x_1 \wedge \bar{x}_2 \wedge x_3$) and disjunctions (like $x_1 \vee \bar{x}_2 \vee x_3$) are special cases of decisions lists.
   (b) Show that decisions lists are a special case of linear threshold functions. That is, any function that can be expressed as a decision list can also be expressed as a linear threshold function "$f(x) = +$ iff $w_1 x_1 + \ldots w_n x_n \geq w_0$", for some values w_0, w_1, ..., w_n.

3. **Mistake-bound model.** k-CNF is the class of Conjunctive Normal Form formulas in which each clause has size at most k. E.g., $x_4 \wedge (x_1 \vee x_2) \wedge (x_2 \vee \bar{x}_3 \vee x_5)$ is a 3-CNF. Give an algorithm to learn 3-CNF formulas over n boolean features in the mistake-bound model. Your algorithm should run in polynomial-time per example (so the "halving algorithm" is not allowed). How many mistakes does it make at most?

Extra Credit:

4. **Perceptron.** Describe a set S of $O(n)$ examples over $\{0,1\}^n$ that are linearly separable by a hyperplane through the origin, but where the Perceptron algorithm takes exponential time for learning (i.e., time $2^{\Omega(n)}$). Specifically, we are imagining we repeatedly cycle the perceptron algorithm through the examples until we have $w \cdot x > 0$ for every positive example $x \in S$ and we have $w \cdot x < 0$ for every negative $x \in S$. For simplicity, use the version of the Perceptron algorithm that does not normalize examples (i.e., when a mistake is made on a positive example x, it sets $w = w + x$, rather than $w = w + x/|x|$, and similarly for mistakes on negatives). This won't really matter since $|x| \leq \sqrt{n}$ for $x \in \{0,1\}^n$, but it is easier to think about since the w_i will always be integral. Explain why your set of examples has the desired property.