CS 2416
Machine Learning Theory
ASSIGNMENT # 1
Due: October 20

The following homework is due at the beginning of class on October 20. Please note that late homework assignments will not be accepted. You may work with a partner from this class on the problems if you want. If you do work with a partner, please acknowledge this by writing down who you worked with on your assignment. Please try to avoid consulting the web or any papers for solutions. I expect you to write up your solutions in your own words, and to acknowlege any sources that you did use.

1. Give an efficient learning algorithm in the consistency model for linear separators. That is, the concept class C_n is the set of all functions of the form $a_1x_1 + \ldots + a_nx_n \geq 0$, where $a_1, \ldots, a_n$ are real-valued. The algorithm is given labelled examples consistent with some $c \in C_n$. (The examples are pairs (α, β) where α is a value for $\overline{x} = x_1, \ldots, x_n$ and $\beta \in \{0, 1\}$.)

2. Given an algorithm to learn 2-CNF formula over n boolean variables in the mistake-bound model. Your algorithm should run in polynomial-time per example (so the halving algorithm is not allowed). Give an upper bound on the number of mistakes it makes in the worst case.

3. An online mistake-bound algorithm is said to be *conservative* if it changes its hypothesis only when a mistake is made. Let $\mathcal{C}$ be a concept class and let $\mathcal{A}$ be an online learning algorithm (not necessarily conservative) which has a finite mistake bound M for $\mathcal{C}$. Prove that there is a conservative learning algorithm A' for $\mathcal{C}$ which also has mistake bound M.

4. In your excitement over Winnow, you decide that if the Winnow algorithm is good, then the following "Winnow2" algorithm might be even better. Winnow2 starts with all weights $w_1, \ldots w_n$ equal to 2, and the threshold θ is n as before. False positives are handled as in the usual Winnow algorithm by setting the appropriate weights to 0, but on a false negative example x, for all i such that $x_i = 1$, the corresponding weight w_i is set to w_i^2, i.e. the weight is *squared*. Does this modification work? Given an analysis of the Winnow2 algorithm for learning monotone disjunctions of size k. What mistake bound can you show for the algorithm? Is it asymptotically better, worse, or the same as the Winnow mistake bound?

5. (Exercise 1.5 from book) In Definition 2 in the book, a PAC learning algorithm can run in time polynomial in n and $size(c)$, where both n and $size(c)$ are given as part of the input. Prove that it is actually unnecessary to give $size(c)$ as part of the input. That is, show that if there is an efficient PAC learning algorithm for C that is given $size(c)$ as input, then there is an efficient PAC algorithm for C that is not given $size(c)$ as part of the input.