15-859(B) Machine Learning Theory

Homework # 7 **Due: 4/18, 2002**

Fun with Winnow. You may wish to review the lecture notes for 1/24.

Note: This is the last homework assignment. Projects are due on the last day of class. More info on the take-home final later.

Problems:

1. Here is a variation on the Winnow algorithm, called *Balanced Winnow*. First of all, we introduce a fake variable x_0 which is set to 1 in every example. For each variable x_i ($0 \leq i \leq n$), and each output value y ($y \in \{-, +\}$, but you can also use this for multi-valued outputs) we have a weight w_{iy}. All weights are initialized to 1. The algorithm proceeds as follows:

   (a) Given example x, predict the label y such that $\sum_i x_i w_{iy}$ is largest.

   (b) If the algorithm makes a mistake, predicting y' when then correct answer is y, then for each $x_i = 1$, multiply the weight w_{iy} by α, and divide the weight $w_{iy'}$ by β.

   For simplicity of analysis, let's use $\alpha = 3/2$ and $\beta = 2$ (but if you prefer, you can use $\alpha = \beta = 2$).

   Prove that as with the standard Winnow algorithm, this algorithm makes at most $O(r \log n)$ mistakes on any disjunction (OR-function) of r variables.

2. A generalization of OR-functions (and AND-function) is the class of functions of the form: $x_{i_1} + x_{i_2} + \ldots + x_{i_r} \geq k$. (E.g., an OR-function uses $k = 1$, and an AND-function uses $k = r$.) Let's call these "k-of-r functions". Pick either the standard Winnow algorithm or the Balanced Winnow algorithm, as well as your choice of multiplier and divider, and prove a mistake bound of $O(rk \log n)$. Suggestion: use $\alpha = \beta = 1 + \frac{1}{2k}$.

   (You can peek at the proof in Littlestone's paper if you get stuck)