15-859(B) Machine Learning Theory

Homework # 4 **Due: March 20, 2006**

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. In class, we argued that if we had an algorithm $\mathcal{A}$ with a noticeable δ' probability of producing a hypothesis of error at most $\epsilon/2$, we could convert it into an algorithm $\mathcal{B}$ that has a high $1-\delta$ probability of producing a hypothesis of error at most ϵ. The reduction was that we run $\mathcal{A}$ for $N = \frac{1}{\delta'}\log\frac{2}{\delta}$ times, and then test the N hypotheses produced on a new test set, choosing the one that performs best. However, we didn't do the Chernoff argument for the last step in detail in lecture. For this exercise, use Chernoff bounds to finish the argument. That is, assuming that at least one of these N hypotheses has error at most $\epsilon/2$, give an explicit bound (without O notation) on the size of the test set needed so that with probability at least $1-\delta/2$ the hypothesis that performs best on the test set has error at most ϵ.

Problems:

2. 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. In addition, we are given parameters $\alpha > 1$ and $\beta < 1$. 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 multiply $w_{iy'}$ by β.

   Using $\alpha = 3/2$ and $\beta = 1/2$, 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.

3. Show for $\alpha = 1+\epsilon$ and $\beta = 1-\epsilon$, that Balanced-Winnow approximates the constraints in the maxent algorithm. You may assume for simplicity that there are only two labels, positive and negative, and $\epsilon \leq 1/4$. Specifically,

   (a) If M_p is the number of mistakes made on positive examples, and M_n is the number of mistakes on negative examples, then

   $$M_p \leq M_n(1 + O(\epsilon)) + O(\frac{1}{\epsilon}\log n)$$

and vice-versa. Hint: think about the fake variable x_0.

(b) Show this implies that if N_p is the true number of positive examples seen, and $\hat{N}_p$ is the number of times the algorithm has predicted positive, then

$$\hat{N}_p \leq N_p(1 + O(\epsilon)) + O(\frac{1}{\epsilon}\log n),$$

and similarly for negative examples.

(c) Show that the same statements hold if we only consider the subset of examples having $x_i = 1$. That is, the number of mistakes on positives having $x_i = 1$ is approximately the number of mistakes on negatives having $x_i = 1$, implying the same statement about the number of times the algorithm *predicts* positive given that $x_i = 1$ versus the number of true positives with $x_i = 1$.