

15-859(B) Machine Learning Theory

Homework # 2

Due: February 8, 2007

Groundrules: Same as before. You should work on the exercises by yourself but may work with a partner 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. **A bad modification to Winnow.** Suppose that we modify Winnow so that it doubles its weights on positive examples even when it did *not* make a mistake. Show how this can cause the algorithm to make an unbounded number of mistakes, even if all examples *are* consistent with some disjunction.
2. **Balanced Winnow.** 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 (as usual, $y \in \{-, +\}$, but you can also use this algorithm 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.

Problems:

3. **Tracking a moving target.** Here is a variation on the deterministic Weighted-Majority algorithm, designed to make it more adaptive.
 - (a) Each expert begins with weight 1 (as before).
 - (b) We predict the result of a weighted-majority vote of the experts (as before).
 - (c) If an expert makes a mistake, we penalize it by dividing its weight by 2, but *only* if its weight was at least $1/4$ of the average weight of experts.

Prove that in any contiguous block of trials (e.g., the 51st example through the 77th example), the number of mistakes made by the algorithm is at most $O(m + \log n)$, where m is the number of mistakes made by the best expert *in that block*, and n is the total number of experts.

4. **Balanced winnow revisited.** 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

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

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

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

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 $x_i = 1$.