# 15-859(B) Machine Learning Theory

**Homework # 2** **Due: February 13, 2006**

---

**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. **Weighted-majority.** For this problem you may use either the deterministic or randomized weighted-majority algorithm.

   (a) Suppose we have some initial belief about which expert is likely to be the best one. In that case, a natural modification to the Weighted-Majority algorithm is that instead of initializing all the weights to 1, we instead initialize $w_i = p_i$, where $p_i$ is our initial belief that expert $i$ is going to be best ($\sum_{i=1}^{n} p_i = 1$). Show how this results in a bound where the $\ln n$ term is replaced with $\ln(1/p_i)$. For example, if you pick the randomized algorithm, you should get a statement that for all $i$,

   $$M \leq \frac{1}{\epsilon}\left[m_i \ln(1/(1-\epsilon)) + \ln(1/p_i)\right],$$

   where $m_i$ is the number of mistakes of expert $i$. So, this bound is better if our prior beliefs turn out to be reasonable.[1]

   (b) What if we have (countably) infinitely many experts? Use your answer to part (a) to show how you can replace $\ln n$ with $O(\log i)$ in comparing our performance to that of the $i$th expert.

2. **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.

**Problems:**

3. **Perceptron for approximately maximizing margins.** In class we saw that the perceptron algorithm makes at most $1/\gamma^2$ mistakes on any sequence of examples that is linearly-separable by margin $\gamma$ (i.e., any sequence for which there exists a unit-length vector $w^*$ such that all examples $x$ satisfy $\ell(x)(w^* \cdot x)/||x|| \geq \gamma$, where $\ell(x) \in \{-1, 1\}$ is the label of $x$).

---

[1]Notice that in this analysis we are *not* assuming that the best expert is actually picked from our prior. We simply are producing a bound that depends on what our beliefs were.

Suppose you are handed a set of examples $S$ and you want to actually find a large-margin separator for them. One approach is to directly solve for the maximum-margin separator using convex programming (which is what is done in the SVM algorithm). However, if you only need to *approximately* maximize the margin, then another approach is to use Perceptron. In particular, suppose you cycle through the data using the Perceptron algorithm, updating not only on mistakes, but also on examples $x$ that your current hypothesis gets correct by margin less than $\gamma/2$. Assuming your data is separable by margin $\gamma$, show that this is guaranteed to halt in a number of rounds that is polynomial in $1/\gamma$. (In fact, you can replace $\gamma/2$ with $(1-\epsilon)\gamma$ and have bounds that are polynomial in $1/(\epsilon\gamma)$.)

4. **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.