15-859(B) Machine Learning Theory

Homework # 3 **Due: February 22, 2012**

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. [**VC-dimension of MAJ**(H)] Show that if hypothesis class H has VC-dimension d, then the class $\text{MAJ}_k(H)$ has VC-dimension $O(kd \log kd)$. Recall that $\text{MAJ}_k(H)$ is the class of functions achievable by taking majority votes over k functions in H. Note that we are only asking for an upper bound here, not a lower bound.

Problems:

In problems 2-4, you will prove that the VC-dimension of the class H_n of halfspaces in n dimensions is $n+1$. (H_n is the set of functions $a_1x_1 + \ldots + a_nx_n \geq a_0$, where $a_0, \ldots, a_n$ are real-valued.) We will use the following definition: The *convex hull* of a set of points S is the set of all convex combinations of points in S; this is the set of all points that can be written as $\sum_{x_i \in S} \lambda_i x_i$, where each $\lambda_i \geq 0$, and $\sum_i \lambda_i = 1$. It is not hard to see that if a halfspace has all points from a set S on one side, then it must have the entire convex hull of S on that side as well. [Then problem 5 will relate this to the Perceptron bound]

2. [**lower bound**] Prove that VC-dim$(H_n) \geq n+1$ by presenting a set of $n+1$ points in n-dimensional space such that one can partition that set with halfspaces in all possible ways. (And, show how one can partition the set in any desired way.)

3. [**upper bound part 1**] The following is "Radon's Theorem," from the 1920's.

   Theorem. *Let S be a set of $n+2$ points in n dimensions. Then S can be partitioned into two (disjoint) subsets S_1 and S_2 whose convex hulls intersect.*

   Show that Radon's Theorem implies that the VC-dimension of halfspaces is *at most* $n+1$. Conclude that VC-dim$(H_n) = n+1$.

4. [**upper bound part 2**] Now we prove Radon's Theorem. We will need the following standard fact from linear algebra. If $x_1, \ldots, x_{n+1}$ are $n+1$ points in n-dimensional space, then they are linearly dependent. That is, there exist real values $\lambda_1, \ldots, \lambda_{n+1}$ *not all zero* such that $\lambda_1 x_1 + \ldots + \lambda_{n+1} x_{n+1} = 0$.

   You may now prove Radon's Theorem however you wish. However, as a suggested first step, prove the following. For any set of $n+2$ points $x_1, \ldots, x_{n+2}$ in n-dimensional space, there exist $\lambda_1, \ldots, \lambda_{n+2}$ *not all zero* such that $\sum_i \lambda_i x_i = 0$ and $\sum_i \lambda_i = 0$. (This is called *affine dependence*.) Now, think about the lambdas...

5. [**More on margins**] Algorithms such as Perceptron and SVMs do especially well when data is linearly separable by a large L_2 margin γ.[1] For example, the Perceptron algorithm makes at most $O(1/\gamma^2)$ mistakes; so, if the margin γ is large compared to $1/\sqrt{n}$, then the number of mistakes is small compared to the VC-dimension bound. On the other hand, it is also possible for the margin bound to be much worse than the VC-dimension bound. Give an example of $O(n)$ points in $\{0,1\}^n$ that *are* linearly separable but where the Perceptron algorithm would make an exponential number of updates if you cycled through the data until you have $w \cdot x > 0$ for every positive example in your set S and $w \cdot x < 0$ for every negative example in your set S. For concreteness, let us consider a version of the Perceptron algorithm that does not normalize the examples to all have Euclidean length 1: it just adds or subtracts the given positive/negative example from the weight vector on a mistake (this will make things conceptually easier). In particular, with this version the weights are always integral. *So, it is sufficient to come up with a set of $O(n)$ linearly-separable examples in $\{0,1\}^n$ such that the only integral-weight linear separator has exponential-sized weights.*[2]

   Hint: your example will also prove that the Perceptron algorithm is not a legal solution to problem 4 on hwk 1.

[1] As in Lecture 4, defining margin as the minimum distance of any example to the separator when examples have been normalized to unit Euclidean length.

[2] This will also imply that the margin of separation for these examples is exponentially small. In particular, since examples are in $\{0,1\}^n$, the mistake bound of the non-normalizing Perceptron algorithm becomes n/γ^2, and if this is exponentially large, then γ must be exponentially small.