Assignment 2

This assignment is due back in class on Thursday March 4. Please submit before the class begins. 

Note: Make sure you write and justify all your statements in a precise, formal way.

Problem 1

Solve problem 1.1 on page 26 of the text by Kearns and Vazirani.

Problem 2

Solve problem 1.5 on page 27 of the text by Kearns and Vazirani.

Guidelines: The algorithm that learns without prior knowledge of the parameter size(c) (call it $B(n, \epsilon, \delta)$) uses the algorithm that requires size(c) in its input (call it $A(n, size(c), \epsilon, \delta)$) as a subroutine, testing the hypothesis output by $A()$ in each run. This can be done as follows:

Algorithm $B(n, \epsilon, \delta)$

1. $i \leftarrow 0$
2. Repeat
3. $i \leftarrow i + 1$
4. $h_i \leftarrow$ hypothesis output by $A(n, \sqrt{i}, \delta, \delta)$
5. Test $h_i$: $e_i \leftarrow$ estimated error of $h_i$ on a sample of size $m_i$
6. Until $e_i \leq \sqrt{\delta}$
7. Output $h = h_i$

Notice that algorithm $B$ may run forever. We therefore relax the definition of efficient PAC learnability by allowing this to happen with low probability, as detailed in the next paragraph.

Your Task: Set appropriate values to the parameters in boxes to get a concrete version of algorithm $B$. Show that there is a polynomial $p(n, size(c), \epsilon, \delta)$ such that with probability at least $1 - \delta$, algorithm $B$ halts in time $p(n, size(c), \epsilon, \delta)$ and outputs a hypothesis $h$ such that $error(h) \leq \epsilon$. You must supply detailed arguments showing that these conditions hold.
Problem 3

In class we saw that if we get an on-line learning algorithm for a concept class $C$ with mistake bound $M$ we can transform it to a PAC-learning algorithm for the same class with sample complexity $\frac{M}{\epsilon} \ln \frac{M}{\delta}$.

Now suppose you are given an on-line learning algorithm for a concept class $C$ but you do not know its mistake bound. Show how to transform it to a PAC-learning algorithm for the same class. You have to give the algorithm, analyze it and give the sample size.

**Hint:** As in class we are allowed failure probability of $\delta$. The problem is that if we employ a similar strategy to the one in class we do not know how many rounds we will have ($M$) and therefore we cannot give a portion of $\frac{\delta}{M}$ to each round.

Problem 4

Solve problem 1.2 on page 27 of the text by Kearns and Vazirani.