CS269: Machine Learning Theory
Problem Set 1

Due Wednesday, October 27, 2010

Ground Rules:

- This problem set is due at the beginning of class on October 27. Please bring a **hard copy** of your solutions with you to class. Late assignments may be submitted either to Jenn or to Edna Todd in 4532N Boelter, and will be penalized 25%. No assignments will be accepted more than 24 hours late.
- All solutions must be typed and printed out. Solutions that are not typed will be penalized 20%, and unreadable answers will not be graded.
- You will be graded on both correctness and clarity. Be concise and clear, especially when writing proofs! If you cannot solve a problem completely, you will get more partial credit if you identify the gaps in your argument rather than trying to cover them up.
- You are strongly encouraged to discuss the problem set with other students in the class. You may also consult other references, such as textbooks or online notes. However, you must write out the solutions to the problems **on your own**. Your answers should be in your own words, and you should understand the answers you are submitting. Additionally, you **must list all of your collaborators** on your assignment and **properly credit any sources** that you consult.

Problems:

1. **Learning n-dimensional boxes** *(60 points total)*

   In this problem, we consider the class of n-dimensional boxes. Each function c in this class is specified by a set of $2n$ values $\ell_1^c, \ell_2^c, \cdots, \ell_n^c$ and $u_1^c, u_2^c, \cdots, u_n^c$ which define an axis-aligned n-dimensional box. Given an n-dimensional input vector $\vec{x}$, $c(\vec{x})$ is defined to be 1 if for every $i \in \{1, \cdots, n\}$ the ith coordinate of $\vec{x}$ lies in $[\ell_i^c, u_i^c]$. Otherwise, $c(\vec{x})$ is defined to be 0.

   An example of a 2-dimensional box is shown in Figure 1.

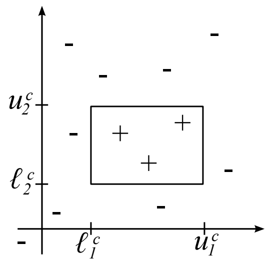

Figure 1: An example of a 2-dimensional box and some labeled points.

Hint: For each problem below, first think about the case when $n = 1$, then move on to the $n = 2$ case before trying to find a general solution.

(a) Let C be the class of n-dimensional boxes. State an efficient algorithm that takes as input an arbitrary set of points $\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_m$, each in $\mathbb{R}^n$, and their corresponding labels $c(\vec{x}_1), c(\vec{x}_2), \cdots, c(\vec{x}_m)$, and outputs a function $h \in C$ that is consistent with this data. You may assume a model of computation in which real numbers can be stored in constant memory and in which basic operations on real numbers (addition, comparisons, etc.) take constant time. Hint: Your algorithm should be simple enough to implement in a few lines of code. *(10 points)*

(b) Let D be an arbitrary, unknown distribution over points in $\mathbb{R}^n$, and let c be an arbitrary, unknown n-dimensional box. Suppose your algorithm from part a is given as input m points $\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_m$ drawn i.i.d. from D and their corresponding labels $c(\vec{x}_1), c(\vec{x}_2), \cdots, c(\vec{x}_m)$. Let h be the function output by your algorithm. Let δ be a fixed parameter in $(0, 1)$. Without referring to VC dimension, state and prove an upper bound on $\Pr_{\vec{x}\sim D}[h(\vec{x}) \neq c(\vec{x})]$ that holds with probability at least $1 - \delta$. Don't worry about getting the best possible constant factors in your bound, just focus on the important parameters (n, δ). *(20 points)*

(c) What is the VC dimension of the class of n-dimensional boxes? Provide a proof of your answer. *(20 points)*

(d) Use the VC dimension that you derived in part c to provide an alternate upper bound on $\Pr_{\vec{x}\sim D}[h(\vec{x}) \neq c(\vec{x})]$ that holds with probability at least $1 - \delta$. How does this bound compare to your answer from part b? Again, don't worry too much about constants. *(10 points)*

2. **VC dimension of linear threshold functions** *(40 points total)*

Consider the class of linear threshold functions which pass through the origin in n-dimensions. Each function c in this class can be represented as an n-dimensional weight vector $\vec{w}_c$ such that $c(\vec{x}) = 1$ if $\vec{w}_c \cdot \vec{x} \geq 0$ and $c(\vec{x}) = 0$ otherwise. In this problem, you will prove that the VC dimension of this class is n. Once again, you may find it useful to think about small values of n first.

(a) First prove that the VC dimension of the set of linear thresholds passing through the origin in n dimensions is *at least* n. That is, describe a set of n points that can be shattered by this class, and a give brief argument showing that they can be shattered. Your solution should hold for arbitrary values of n, not just one specific value. *(15 points)*

(b) Show that no set of $n + 1$ points can be shattered by this class. Hint: Recall that a set of k points $\vec{x}_1, \cdots, \vec{x}_k$ is linearly dependent if $\sum_{i=1}^{k} \lambda_i \vec{x}_i = 0$ for $\lambda_1, \cdots, \lambda_k$ such that at least one λ_i is non-zero. That is, there exists one point that can be written as a weighted sum of the others. Start by showing that no set of linearly dependent points can be shattered by this class. *(20 points)*

(c) In class we discussed a "bad" argument for why the number of examples needed learn n-dimensional thresholds should be linear in n, in which we considered the finite concept class containing only those linear threshold functions with parameters that can be stored in b bits of memory. (In class we did not limit ourselves to thresholds passing through the origin, but the argument is the same.) Briefly (in **one to two sentences**) comment on the relationship between the bounds that we achieved using this bad argument and the bounds we would achieve using VC dimension, paying particular attention to the dependence on n. *(5 points)*