**Exercises:**

1. **Intervals.** Consider the class $C$ of intervals on the real line $\mathbb{R}$. That is, a legal target function is specified by an interval $[a, b]$ and classifies an example $x$ as positive if $x \in [a, b]$ and as negative otherwise. Give an algorithm to learn this class in the PAC model, and a proof that a training sample of size $O\left(\frac{1}{\epsilon} \log \frac{1}{\delta}\right)$ is sufficient for your algorithm to achieve error $\leq \epsilon$ with probability $\geq 1 - \delta$.

2. **More on margins.** We didn’t prove a uniform convergence result for large margin separators in class (e.g., a sample complexity bound for SVM). However, to get some intuition, prove that it is not possible to have a set $S$ of $\frac{1}{\gamma^2} + 1$ points in the unit ball such that every labeling of $S$ is achievable by a linear separator through the origin of margin at least $\gamma$. Hint: this has a one-sentence proof.

An aside: the “ghost sample” proof we gave in class does not immediately give a uniform convergence result from this “VC-dimension-ish” statement, because, when the double-sample $S''$ is partitioned into $S$ and $S'$, there may be some large-margin labelings of $S$ that do not extend to any large-margin labeling of $S''$. So a more involved proof is needed.

3. **Models of learning.** Consider the following four learning models. In each of these we are given a class $C$, and the goal of the learning algorithm is to exactly recover the target concept $c \in C$.

**Equivalence Query:** In this model the algorithm can propose a hypothesis $h$ and is told either that $h$ is correct, or else is given a counterexample $x$ such that $h(x) \neq c(x)$. This is really the same as the Mistake-Bound model.

**Restricted Equivalence Query:** Same as above except $h$ must be from class $C$.

**Membership Query Only:** In this model the algorithm can propose examples $x$ and is told their labels. But the algorithm is not allowed any mistakes / equivalence queries: it must exactly recover the target given only the membership queries.
Teacher-directed: Like the membership query model, but now a teacher who knows the target function proposes the examples to be queried. The examples must uniquely identify the target concept, in the sense that no other \( c' \in C \) is consistent with this set of data. I.e., think of yourself as the teacher, trying to teach an adversarial (but consistent and proper) learning algorithm.

For each class of functions below, state in which of the above models it can or cannot be learned/taught using a number of queries polynomial in \( n \), along with a brief explanation.

(a) The class of all functions over \( \{0, 1\}^n \) having exactly one positive example.
(b) The class of all functions over \( \{0, 1\}^n \) having at most one positive example.
(c) The class of monotone conjunctions over \( \{0, 1\}^n \) (including the conjunction of nothing, which is always positive).
(d) The class of decision lists over \( \{0, 1\}^n \).

4. XOR of Conjunctions. Consider the class \( C \) of “XORs of two conjunctions”. This is the class of functions \( f \) that can be described as \( f(x) = T_1(x) \oplus T_2(x) \) where \( T_1 \) and \( T_2 \) are conjunctions, and “\( \oplus \)” is XOR; so, \( f(x) \) is positive if \( x \) satisfies exactly one of \( T_1 \) or \( T_2 \). For instance, the following is an XOR of two conjunctions:

\[
x_1 x_3 \bar{x}_5 \oplus x_2 x_4.
\]

This function is positive on examples 11100 and 11011, and is negative on examples 00000 and 11110.

Give an algorithm that learns this class \( C \) over \( \{0, 1\}^n \) in the mistake-bound model. Your algorithm should have a mistake bound polynomial in \( n \) and should be efficient (running in polynomial time per example) too. Your algorithm need not use \( C \) as its hypothesis representation. If you get stuck, then for partial credit give an inefficient algorithm.

5. [Course evaluation] Please fill out the online course evaluation and write down “Done”. (These are totally anonymous, so you get full credit for writing “Done” without actually doing the evaluation, but please do the evaluation – it’s quite useful for us.) You may give yourself an extra 15 minutes in addition to the allotted 24 hours to complete this question.