Instructor: Rocco A. Servedio

Computer Science 4252: Introduction to Computational Learning Theory
Problem Set #1 Fall 2025

Due 11:59pm Tuesday, September 23, 2025

See the course Web page for instructions on how to submit homework.
Important: To make life easier for the TAs, **please start each problem on a new page.**

Remember to strive for both clarity and concision in your solutions;
solutions which are excessively long may be penalized.

Problem 1 In this problem you'll play around with some of the concept classes over $\{0,1\}^n$ that we discussed in class (and some related concept classes).

(i) Show that any decision list on variables $x_1, \ldots, x_n$ can be expressed as an n-term DNF formula.

(ii) True or false: there is a linear threshold function over $\{0,1\}^n$ such that any DNF that expresses it must have $2^{\Omega(n)}$ terms. Justify your answer.

(iii) A *k-junta* is a Boolean function $f : \{0,1\}^n \to \{0,1\}$ that only depends on k of the n input variables, i.e.

$$f(x_1, \ldots, x_n) = g(x_{i_1}, \ldots, x_{i_k})$$

for some $g : \{0,1\}^k \to \{0,1\}$ and some $1 \leq i_1 < \cdots < i_k \leq n$.

True or false: If f is a $(\log n)$-junta, then f is computed by a poly(n)-term DNF. (Justify your answer.)

Problem 2

(i) Suppose that you were to use the online algorithm for learning decision lists described in class to learn a target function which is in fact a conjunction of r literals over variables $x_1, \ldots, x_n$. Prove that the algorithm would make at most $O(n)$ mistakes irrespective of the value of r.

(ii) A decision list has ℓ *alternations* if the ordered sequence of output bits it contains changes value ℓ times. For instance, the decision list

$$(\overline{x}_3, 1), (x_4, 1), (\overline{x}_7, 0), (x_5, 1), (x_1, 0), (x_8, 0), (1)$$

(which means "if $\overline{x}_3$ then output 1, else ... else if x_8 then output 0, else output 1") has $\ell = 4$ alternations.

What is the best asymptotic mistake bound you can prove for the algorithm from class for learning a length-r decision list with ℓ alternations over $x_1, \ldots, x_n$?

Problem 3 In a fit of enthusiasm, you decide that if the Winnow1 algorithm is good, the following "HyperWinnow1" algorithm must be even better. HyperWinnow1 starts with all weights $w_1, \ldots, w_n$ equal to 2 (the threshold is $\theta = n$ as before). False positives are handled as in the usual Winnow1 algorithm by setting the appropriate weights to 0, but on a false negative example x, for all i such that $x_i = 1$ the corresponding weight w_i is set to w_i^2, i.e. the weight is **squared**.

Is your enthusiasm misguided? Give an analysis of the HyperWinnow1 algorithm for learning monotone disjunctions of size k. (Hint: It's a good idea to follow the general approach of the analysis of the Winnow1 algorithm.) What mistake bound can you show for the algorithm? Try to give the best bound that you can come up with. How does this compare with Winnow1's mistake bound if k is small compared with n?

Problem 4 In class we were a little casual about what exactly it means for an online learning algorithm to "maintain a hypothesis" or "update its hypothesis" between examples. In this problem we'll make this a bit more precise by introducing a notion of the *state* of a learning algorithm, and we'll prove a simple lower bound on the number of states that a learning algorithm needs to learn a finite concept class $\mathcal{C}$.

Here goes. Let X be a (finite) domain, and let $\mathcal{C}$ be a (finite) concept class over X. A *finite-state online learning algorithm* A is given by

- a finite set Ω (the "state space" of the learner),
- an initial state $\omega_1 \in \Omega$ (the state the learner is in before receiving any examples),
- a *prediction function* $\text{pred} : \Omega \times X \to \{0, 1\}$, and
- a *next-state function* $\text{next} : \Omega \times X \times \{0, 1\} \to \Omega$.

Aligning this with our discussion from class: at every stage in the execution of the learning algorithm the algorithm is some current internal state, which is an element of Ω. If the current internal state is $\omega \in \Omega$, then the current hypothesis is $h_\omega = \text{pred}(\omega, \cdot)$ (note that this is a function that maps $X \to \{0, 1\}$). Once the true label $c(x) \in \{0, 1\}$ for an example x is received, the algorithm updates its hypothesis by transitioning to state $\text{next}(\omega, x, c(x))$, which is some element ω' of Ω.

Finally, let's write Hyp to denote the set of all hypotheses $\{h_\omega : \omega \in \Omega\}$.

Show that if A is a finite-state online learning algorithm for concept class $\mathcal{C}$ with a finite mistake bound, then the number of states $|\Omega|$ must be at least $|\mathcal{C}|$.