Machine Learning Theory **CSCI599, Spring 2018**

Problem Set 1

Due: Monday, February 19, 3 p.m. by email

Please title your email "CSCI599_PS1".

---

1. Assume each example x is given by n boolean variables. A decision list is a function of the form: "if ℓ_1 then b_1, else if ℓ_2 then b_2, else if ℓ_3 then b_3, ..., else b_m," where each ℓ_i is a literal (either a variable or its negation) and each $b_i \in \{0,1\}$.

   (a) Show that conjunctions (like $x_1 \wedge \overline{x_2} \wedge x_3$) and disjunctions (like $x_1 \vee \overline{x_2} \vee x_3$) are special cases of decisions lists.

   (b) Show that decisions lists are a special case of linear threshold functions. That is, any function that can be expressed as a decision list can also be expressed as a linear threshold function "$f(x) = +$ iff $w_1 x_1 + \ldots w_n x_n \geq w_0$", for some values $w_0, w_1, \ldots, w_n$.

2. An online mistake-bound algorithm is said to be conservative if it changes its hypothesis only when a mistake is made. All of the online mistake-bound algorithms we have seen in class (the elimination algorithm, the decision list algorithm, Winnow, etc.) are conservative. In this problem, you will show that this is not a coincidence: Let C be any concept class, and let A be any online learning algorithm (not necessarily conservative) which has a finite mistake bound M for C. Prove that there exists a conservative online learning algorithm A' for C which also has mistake bound M.

3. (a) Recall the Winnow1 algorithm explained in class for learning the class of Boolean disjunctions. Consider the following modification to the algorithm: The modified algorithm doubles its weights on positive examples even when it did not make a mistake. What is the mistake bound of this modified algorithm? Justify your answer.

   (b) Fix any value $1 \leq k \leq n$. Suppose that we run the Winnow1 algorithm on a k-sparse monotone disjunction over $\{0,1\}^n$. Is it possible for the algorithm to make $\Omega(k \log n)$ mistakes? Justify your answer.

4. (a) Let $\mathcal{H}^n_{\text{dis}}$ be the class of Boolean disjunctions over the variables $x_1, \ldots, x_n$. What is the VC-dimension of the class $\mathcal{H}^n_{\text{dis}}$?

   (b) For a set $I \subseteq \{1, 2, \ldots, n\}$, we define a parity function $h_I : \{0,1\}^n \to \{0,1\}$ as follows: On a binary vector $x = (x_1, \ldots, x_n) \in \{0,1\}^n$, we have:

   $$h_I(x) = \left(\sum_{i \in I} x_i \right) \bmod 2 \ .$$

   (That is, h_I computes the parity of bits in I.) What is the VC-dimension of the class of all such parity functions, $\mathcal{H}^n_{\text{parity}} = \{h_I : I \subseteq \{1, 2, \ldots, n\}\}$?

   (c) Find an example of a concept class $\mathcal{C}$ such that $\mathcal{C}$ is infinite while the VC-dimension of $\mathcal{C}$ is 1. Find an example of a concept class whose VC-dimension is infinity.

5. The Perceptron Convergence Theorem shows that the Perceptron algorithm will not make too many mistakes as long as every example is "far" from the separating hyperplane of the target halfspace. In this problem you will explore a variant of the Perceptron algorithm and show that it performs well (given a little help in the form of a good initial hypothesis) as long as every example is "far" (in terms of angle) from the separating hyperplane of the *current hypothesis*.

   Consider the following variant of Perceptron:

   - Start with an initial hypothesis vector $w = w^{\text{init}}$.
   - Given example $x \in \mathbb{R}^n$, predict according to the linear threshold function $w \cdot x \geq 0$.
   - Given the true label of x, update hypothesis vector w as follows:
     - If the prediction is correct, leave w unchanged.
     - If the prediction is incorrect, set $w \leftarrow w - (w \cdot x)x$.

   So the update step differs from that of Perceptron shown in class in that $(w \cdot x)x$ (rather than x) is added or subtracted to w. (Note that if $\|x\|_2 = 1$, then this update causes vector w to become orthogonal to x, i.e., we add or subtract the multiple of x that shrinks w as much as possible.)

   Suppose that we run this algorithm on a sequence of examples that are labeled according to some linear threshold function $v \cdot x \geq 0$ for which $\|v\|_2 = 1$. Suppose moreover that

   - each example vector x has $\|x\|_2 = 1$;
   - the initial hypothesis vector w^{init} satisfies $\|w^{\text{init}}\|_2 = 1$ and $w^{\text{init}} \cdot v \geq \gamma$ for some fixed $\gamma > 0$;
   - each example vector x satisfies $\frac{|w \cdot x|}{\|w\|_2} \geq \delta$, where w is the current hypothesis vector when x is received. (Note that for a unit vector x, this quantity $\frac{|w \cdot x|}{\|w\|_2}$ is the cosine of the angle between vectors w and x.)

   Show that under these assumptions the algorithm described above will make at most $\frac{2}{\delta^2} \ln(1/\gamma)$ many mistakes.