CS 5350/6350, DS 4350: Machine Learning Spring 2024 Homework 2

Jordy A. Larrea Rodriguez

Handed out: February 1, 2024
Due date: February 15, 2024

General Instructions

Please read before you start

- You are welcome to talk to other members of the class about the homework. I am more concerned that you understand the underlying concepts. However, you should write down your own solution. Please keep the class collaboration policy in mind.
- Feel free to discuss the homework with the instructor or the TAs.
- Your written solutions should be brief and clear. You need to show your work, not just the final answer, but you do *not* need to write it in gory detail. Your assignment should be **no more than 10 pages**. Every extra page will cost a point.
- Handwritten solutions or photos of handwritten solutions will not be accepted.
- The homework is due by midnight of the due date. Please submit the homework on Canvas. You should upload two files: a report with answers to the questions below, and a compressed file (`.zip` or `.tar.gz`) containing your code.
- Some questions are marked **For 6350 students**. Students who are registered for CS 6350 should do these questions. Of course, if you are registered for CS 5350 or DS 4350, you are welcome to do the question too, but you will not get any credit for it.

Important Do not just put down an answer. We want explanations of your answers. No points will be given for just the final answer without an explanation.

1 Warmup: Boolean Functions

1. [3 points] Table 1 shows several data points (the x's) along with corresponding labels (y). (That is, each row is an example with a label.) Write down three different Boolean functions, all of which can produce the label y when given the inputs x.

y	$x1$	$x2$	$x3$	$x4$
0	0	1	1	0
0	1	1	1	0
1	0	1	1	1

Table 1: Initial data set

Response:

(a) $y_1 = (x_2 \veebar x_3) \ \vee x_4$

(b) $y_2 = (x_2 \vee x_3) \ \wedge x_4$

(c) $y_3 = x_4$

2. [5 points] Now the Table 1 is expanded to Table 2 by adding more data points. How many errors will each of your functions from the previous questions make on the expanded data set.

y	$x1$	$x2$	$x3$	$x4$
0	0	1	1	0
0	1	1	1	0
1	0	1	1	1
1	1	0	1	1
0	0	1	1	0
1	1	1	0	1

Table 2: Expanded data set

Response:

(a) The function y_1 has 0 errors given the expanded data set, since new features where the term $z = x2 \veebar x3$ is always true when the output is true. The same is true for $z \vee x_4$.

(b) The function y_2 has 0 errors given the expanded data set. It is clear that y is true whenever x_4 is true, so the $\wedge$ operator only outputs true if the following conditions are met, at least one of x_2 or x_3 are true and x_4 is true.

(c) The function y_3 offers the simplest solution since it reflects the value of x_4; furthermore, the function produces zero errors.

3. [5 points] Is the function in Table 2 linearly separable? If so, write down a linear threshold function that classifies the data. If not, prove that there is no linear threshold function that can classify the data.

   Response: The function that represents table 2 is linearly separable if the simplest solutions such as y_3 are considered. Replacing every 0 label with -1 produces a function where the threshold is $y : x_4 >= 0$.

2 Feature transformations

[10 points] Consider the concept class C consisting of functions f_r defined by a radius r as follows:

$$f_r(x_1, x_2) = \begin{cases} +1 & 24x_1^{2024} - 23x_2^{2023} \leq r \\ -1 & \text{otherwise} \end{cases}$$

Note that the hypothesis class is *not* linearly separable in $\mathbb{R}^2$.

Construct a function $\phi(x_1, x_2)$ that maps examples to a new *two-dimensional* space, such that the positive and negative examples are linearly separable in that space. The answer to this question should consist of two parts:

1. A function ϕ that maps examples to a new space.
2. A proof that in the new space, the positive and negative points are linearly separated. You can show this by producing such a hyperplane in the new space (i.e. find a weight vector $\mathbf{w}$ and a bias b such that $\mathbf{w}^T\phi(x_1, x_2) \geq b$ if, and only if, $f_r(x_1, x_2) = +1$.

Response:

Consider a feature transformation function $\phi(x1, x2)$ that when applied on $\mathbf{x} = [x_1 \;\; x_2]^T$ results in $\mathbf{z} = \phi(\mathbf{x}) = [x_1^{2024} \;\; x_2^{2023}]^T$. Allow the variable q to represent the result of the inequality $\mathbf{w}^T\mathbf{z} \geq b$. That is,

$$q = \begin{cases} True\ (+1) & \text{if } \mathbf{w}^T\mathbf{z} \geq b \\ False\ (-1) & \text{if } \mathbf{w}^T\mathbf{z} < b \end{cases}, \text{ where } \mathbf{w} = [-24 \;\; 23]^T \text{ and } b = -r$$

Furthermore, let the variable p be $True$ when $f_r(x_1, x_2) = +1$ and is parameterized by the inequality $24x_1^{2024} - 23x_2^{2023} \leq r$. By definition of linear separability, we must prove the double implication $q \iff p$ is true.

1. Prove $q \implies p$.
   Assume q to be true, then $q \coloneqq \mathbf{w}^T\mathbf{z} \geq b = +1$.

   Substituting $\mathbf{z}$, $\mathbf{w}$, and b.

   $$q \coloneqq [-24 \;\; 23] \begin{bmatrix} x_1^{2024} \\ x_2^{2023} \end{bmatrix} \geq -r$$

   By definition of the inner product,

   $$q \coloneqq -24x_1^{2024} + 23x_2^{2023} \geq -r$$

   Dividing both sides of inequality by -1 and by flip rule of inequalities,

   $$q \coloneqq 24x_1^{2024} - 23x_2^{2023} \leq r$$

   Since $q \coloneqq \mathbf{w}^T\mathbf{z} \geq b = 24x_1^{2024} - 23x_2^{2023} \leq r = f_r(x_1, x_2)$ when q is $+1$, then p must also be true.

2. Prove $p \implies q$
   Assume p to be true, then $p \coloneqq 24x_1^{2024} - 23x_2^{2023} \leq r = +1$. Dividing both sides of inequality by -1 and by flip rule of inequalities,

$$p \coloneqq 24x_1^{2024} - 23x_2^{2023} \leq r$$

   Thus, from a posteriori established in part 1, if $q \coloneqq \mathbf{w}^T\mathbf{z} \geq b = 24x_1^{2024} - 23x_2^{2023} \leq r$, then $q = +1$. So, q must be true.

We have proved that $q \implies p$ and $p \implies q$, therefore, $q \iff p$ must be true by definition of double implication.

3 Mistake Bound Model of Learning

In both the questions below, we will consider functions defined over n Boolean features. That is, each example in our learning problem is a n-dimensional vector from $\{0,1\}^n$. We will use the symbol $\mathbf{x}$ to denote an example and $\mathbf{x}_i$ denotes its i^{th} element. (We will assume that there is no noise involved.)

For all questions below, it is not enough to just state the answer. You need to justify your answer with a short proof.

1. Consider the concept class $\mathcal{C}_1$ defined as follows: Each element of $\mathcal{C}_1$ is defined using a fixed instance $\mathbf{z} \in \{0,1\}^n$ as follows:

$$f_{\mathbf{z}}(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} = \mathbf{z} \\ 0 & \mathbf{x} \neq \mathbf{z}. \end{cases}$$

   That is, the function $f_{\mathbf{z}}$ predicts 1 if, and only if, the input to the function is $\mathbf{z}$.

   Our goal is to come up with a mistake bound algorithm that will learn any function $f \in \mathcal{C}_1$.

   (a) [5 points] Determine $|\mathcal{C}_1|$, the size of concept class.
   Response:
   The size of the concept class is equal to the number of n binary strings or rather $|\mathcal{C}_1| = 2^n$. This is true due to there being n elements being used to enumerate a binary process.

   (b) [15 points] Write a mistake bound learning algorithm for this concept class that makes no more than *one* mistake on any sequence of examples presented to it. Please write the algorithm concisely in the form of pseudocode.

   Prove the mistake bound for this algorithm.

   Algorithm CON (Consistent On Negatives) P: Set of all positive examples, N: Set of all negative examples

$$h \leftarrow \begin{cases} +1 & \text{positive example} \\ -1 & \text{negative example} \end{cases}$$

2. Suppose we have a concept class $\mathcal{C}_2$ that consists of exactly n functions $\{f_1, f_2, \cdots, f_n\}$, where each function f_i is defined as follows:

$$f_i(\mathbf{x}) = \mathbf{x}_i.$$

That is, the function f_i returns the value of the i^{th} feature.

(a) [5 points] How many mistakes will the algorithm **CON** from class make on any function from this concept class? Response: Since the function is a binary decision, then the hypothesis will make at most one mistake. The mistake is made on positive examples at most once because the algorithm corrects any mistake made on positive examples during iterations.

(b) [5 points] How many mistakes will the Halving algorithm make on any function from this concept class? Response: Once again, the outputs are $\in$0,1, so the halving algorithm will make $\leq\ lg(|C|)$ mistakes.