CS 4780/5780: Machine Learning for Intelligent Systems **(Due: 09/29/20)**

Assignment #2: Hypothesis Testing, Linear Classifiers, Perceptrons

Instructor: Nika Haghtalab, Thorsten Joachims *Name:* Student name(s), *Netid:* NetId(s)

Course Policy: Read all the instructions below carefully before you start working on the assignment, and before you make a submission.

- Please include your name and NetIDs on the first page. We recommend typesetting your submission in LaTeX, and an Overleaf template is linked on the Canvas module. When submitting, remember to mark which page has which response.
- Assignments are due by 5pm on the due date in PDF form on Gradescope.
- Late assignments can be submitted on Gradescope up to Sunday, Oct 4 at 5pm. This is also when the solutions will be released.
- You can do this assignment in groups of 1-2. Please submit no more than one submission per group. Collaboration across groups is not permitted.
- All sources of material outside the course must be cited. The University Academic Code of Conduct will be strictly enforced.

Submission Instructions: All group members must be added to the Gradescope submission. If you're the one submitting, add your group members on Gradescope. Otherwise, make sure you are added to the submission. Put your names on the PDF, this helps us track the groups in case there are errors on Gradescope.

Problem 1: Hypothesis Testing (16 + 16 = 32 points)

In this exercise you will evaluate the performance of two binary hypotheses h_A and h_B with significance tests. The file `preds.csv` contains the predictions of h_A and h_B of whether an image depicts a cancerous node or a healthy one. The first column contains a human doctor's diagnosis, i.e. the ground truth we aim to predict, and the second and third columns are the predictions of h_A and h_B, respectively.

Let $X_1, \ldots, X_n$ be independent and identically distributed (i.i.d.) random variables.
As a refresher, Hoeffding's inequality states

$$\alpha = P(|\overline{X} - \mathbb{E}[\overline{X}]| > \epsilon) \leq 2\exp(-2n\epsilon^2) \tag{0.1}$$

where the *sample mean* is denoted $\overline{X} := \frac{1}{n}\sum_{i=1}^{n} X_i$.

(a)

i. Choose a distribution for the random variables $\{X_i\}_{i\in[n]}$ that adequately describes the diagnosis data and accompanying classifier output. What does the value of X_i correspond to? How about $\overline{X}$? For each parameter of the associated probability law, explain what the parameter models.

ii. Use the statistics of this distribution and Eq. 0.1 to derive the $(1-\alpha)$ confidence interval for the (0-1 loss) prediction error as was shown in class.

iii. Compute the $(1-\alpha) = 97.5\%$ confidence interval for each classifier using the sample results with the equation you derived in part (ii).

(b) Use the exact binomial McNemar test to decide whether, with 80% confidence, the error rate of hypothesis h_A is significantly different than that of h_B. Show each step of the calculation - the definition of the null hypothesis, the contingency table, the test calculation, and the final conclusion. Show whether or not the null hypothesis can be rejected with 80% confidence. **Note:** Multiple definitions of the McNemar test appear on the internet (i.e. Wikipedia) and on the slides. Use the one on the slides.

Problem 2: Linear Classifiers (5 + 5 + 5 + 10 + 10 = 35 points)

We will be using linear classifiers to recreate certain functions, which we will call $f(\cdot)$. Recall that a linear classifier is written as

$$h_{\overrightarrow{w},b}(\overrightarrow{x}) = \begin{cases} +1, & \text{if } \overrightarrow{w} \cdot \overrightarrow{x} + b > 0 \\ -1, & \text{otherwise} \end{cases}$$

with parameter vectors $\overrightarrow{w}$ and term b. A homogeneous linear classifier has $b = 0$. For the following questions, construct the linear classifier by choosing $\overrightarrow{w}$ and b.

(a) First create a linear classifier which recreates the logical **NOT** function. The instance space is $X = \{+1, -1\}$. Here $(+1)$ corresponds to TRUE and (-1) corresponds to FALSE, for both the instance space and output of the linear classifier. The function f_{NOT} which we want to recreate is:

$$f_{NOT}(+1) = -1$$
$$f_{NOT}(-1) = +1$$

Give the parameters w, b which create a linear classifier which recreates the logical NOT function f_{NOT}. Note that w is a number, not a vector.

(b) Consider the instance space $X = \mathbb{R}^d$ and let $\overrightarrow{x} = (x_1, \ldots, x_d)$ be a vector of length d. Consider the function

$$f_M(\overrightarrow{x}) = \begin{cases} +1, & \text{if } \sum_{i=1}^{d} x_i > M \\ -1, & \text{otherwise} \end{cases}$$

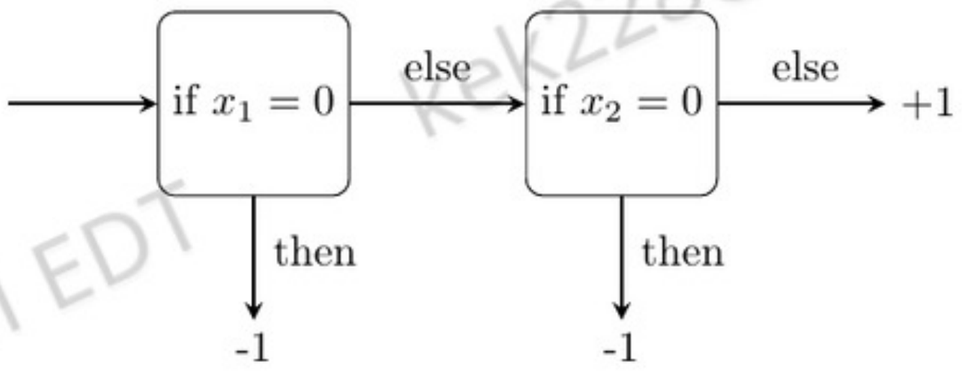
Giver parameters $\overrightarrow{w}, b$. Note here that $\overrightarrow{w}$ is a vector.

(c) Consider the instance space $X = \mathbb{R}^{100}$ and let $\overrightarrow{x} = (x_1, \ldots, x_{100})$. Consider the function

$$f_{100}(\overrightarrow{x}) = \begin{cases} +1, & \text{if } \sum_{i=1}^{10} x_i > \sum_{j=91}^{100} x_j + 5 \\ -1, & \text{otherwise} \end{cases}$$

Give the parameters $\overrightarrow{w}$ and b.

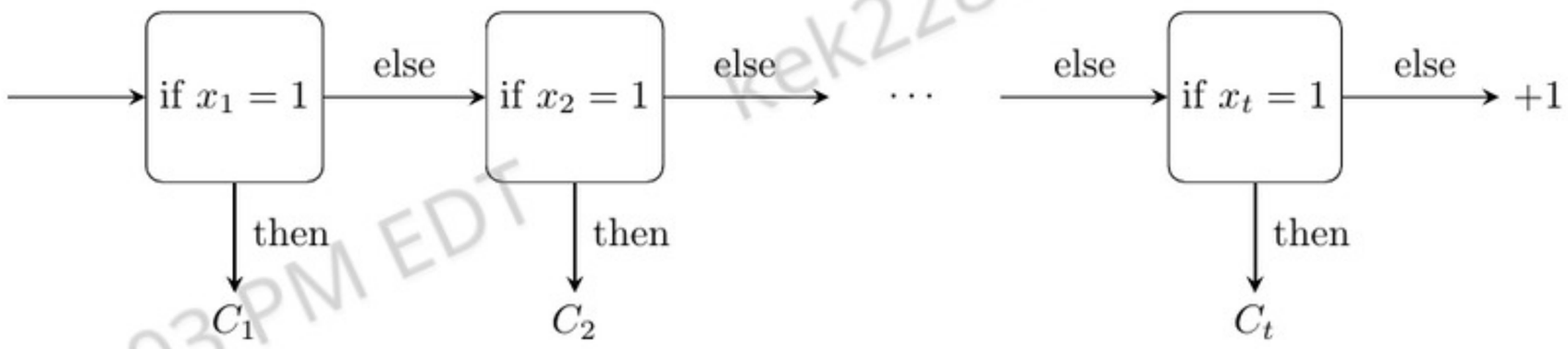
(d) Consider a decision list with two segments as shown.



Create a linear classifier which recreates this 2-segment decision list. Here x_1 refers to the first element and x_2 to the second element of our data vector. The instance space will be $X = \{0, 1\}^2$.

Give the parameters $\overrightarrow{w}_{Cond2}, b_{Cond2}$ for the linear classifier that recreates the 2-segment decison list. For example, we want the linear classifier to classify $x = (1, 0)$ to -1.

(e) Next consider a generalization of the 2-segment decision list. Now there are t segments, and the output of each segment is a variable $C_i = \{-1, +1\}$. However the instance space will be $X = \{0, 1\}^d$, where $d \geq t$.



Give explicit expressions for the parameters $\overrightarrow{w}_{Cond}, b_{Cond}$ which create a linear classifier that recreates this t-segment decision list using input data with number of features d.

Problem 3: Perceptrons (15 + 18 = 33 points)

(a) Consider the three-point 2D dataset $\{(\vec{x}_i, y_i)|i \in [3]\} = \{((1,3),+1),((-1,3),-1),((0,1),+1)\}$.

Starting with $\vec{w}^{(0)} = (0,0)$, how many updates will you have to perform to $\vec{w}$ until convergence for a homogeneous batch perceptron (i.e. bias term $b = 0$)? Write down the sequence of updates $\vec{w}^{(t)}$ (starting from $t = 1$ until convergence), assuming that we iterate through the data points in the following order: $\{(1,3),(-1,3),(0,1)\}$. Provide the correct list of weight updates and show your calculations in order to receive full credit.

(b) Your friend comes to you, desperate for your Perceptron expertise. Their dataset is massive (with more than 10 trillion training examples), and after hours of training their perceptron (until convergence), their code malfunctioned and did not save the final weight vector.

Thankfully, at every training iteration, the code saved which example was used for the update step. Surprisingly, only five of the 10 trillion+ training examples were ever misclassified. They are listed below, along with the number of times they were used in an update step.

Training Example	Number of Times Used in an Update Step
(1, 3, 3, 7, 0), +1	1
(0, 0, 0, 0, 1), -1	3
(0, 0, 0, 1, 0), +1	1
(9, 1, 2, 0, 0), +1	1
(2, 8, 4, 0, -2), -1	1

What is the final weight vector of this perceptron (i.e. the weight vector that would have been saved if the code had not malfunctioned)? Note that this perceptron is also homogeneous (bias term $b = 0$). Assume that $\vec{w}^{(0)} = (0,0,0,0,0)$. Include a *one-sentence* explanation for how the final weight vector was computed.