CMPE 257 Machine Learning Spring 2019

HW#1 Due February 22nd, 11:59 PM, on Canvas

0. Study and install Jupyter notebook (help: https://www.datacamp.com/community/tutorials/tutorial-jupyter-notebook)

2. (5 points) What types of Machine Learning, if any, best describe the following scenarios:

a. A coin classification system is created for a vending machine. The developers obtain exact coin specifications from the U.S. Mint and derive a statistical model of the size, weight, and denomination, which the vending machine then uses to classify coins.

b. A system that takes MRI images as input and identifies whether the patient has a tumor or not. A database of a large set of MRI images that have been labelled by doctors is available. An algorithm uses this data to infer decision boundaries and uses it to classify the tumor.

c. A system that mimics the sorting hat from the Harry Potter series needs to be designed. Students are sorted into houses based on their scores on various aptitude tests such as sports, language, chemistry, etc. A database of past students that includes their scores on different tests along with the houses they were sorted into is provided to you. You use this information and build an algorithm that can sort a new student based on their test scores.

d. A system that takes PET scans as input and identifies the different types of tissues in a 3-D image.

e. A computer develops a strategy for playing Tic-Tac-Toe by playing repeatedly and adjusting its strategy by penalizing moves that eventually lead to losing.

3. (5 points) There are two boxes with kittens. One box contains two white kittens and the other box contains one white and one brown kitten.

From one of the boxes, one white kitten randomly jumps out. What is the probability that the other kitten in the same box is also white? Explain your answers for partial credit.

4. (10 points) Consider a sample of 10 marbles drawn from a bin containing red and green marbles. The probability that any marble we draw is red is $\mu = 0.65$ (independently, with replacement).

We address the probability of getting no red marbles ($\nu = 0$) in the following cases:

a. We draw only one such sample. Compute the probability that $\nu = 0$.

b. We draw 1,000 independent samples. Compute the probability that (at least) one of the samples has $\nu = 0$.

Show your work for partial credit.

5. (20 points) Consider a Boolean target function over a 3-dimensional input space $X = \{0, 1\}^3$ (instead of our ±1 binary convention, we use 0,1 here since it is standard for Boolean functions). We are given a data set D of five examples represented in the table below, where $y_n = f(x_n)$ for n = 1, 2, 3, 4, 5.

x_n	y_n
0 0 0	0
0 0 1	1
0 1 0	1
0 1 1	0
1 0 0	1

Note that in this simple Boolean case, we can enumerate the entire input space (since there are only $2^3 = 8$ distinct input vectors), and we can enumerate the set of all possible target functions (there are only $2^{2^3} = 256$ distinct Boolean function on 3 Boolean inputs).

Let us look at the problem of learning f. Since f is unknown except inside D, any function that agrees with D could conceivably be f. Since there are only 3 points in X outside D, there are only $2^3 = 8$ such functions. The remaining points in X which are not in D are: 101, 110, and 111. We want to determine the hypothesis that agrees the most with the possible target functions. In order to quantify this, count how many of the 8 possible target functions agree with each hypothesis on all 3 points, how many agree on just 2 of the points, on just 1 point, and how many do not agree on any points. The final score for each hypothesis is computed as follows:

Score = (# of target functions agreeing with hypothesis on all 3 points) × 3 + (# of target functions agreeing with hypothesis on exactly 2 points) × 2 + (# of target functions agreeing with hypothesis on exactly 1 point) × 1 + (# of target functions agreeing with hypothesis on 0 points) × 0.

For each of the hypothesis g below, compute the score as defined above:

a. g returns 1 for all three points.

b. g returns 0 for all three points.

c. g is the XOR function applied to x, i.e., if the number of 1s in x is odd, g returns 1; if it is even, g returns 0.

d. g returns the opposite of the XOR function: if the number of 1s is odd, it returns 0, otherwise returns 1.

6. (10 points) Exercise 1.3 from the textbook:

Exercise 1.3

The weight update rule in (1.3) has the nice interpretation that it moves in the direction of classifying $\mathbf{x}(t)$ correctly.

(a) Show that $y(t)\mathbf{w}^{\mathrm{T}}(t)\mathbf{x}(t) < 0$. *[Hint: $\mathbf{x}(t)$ is misclassified by $\mathbf{w}(t)$.]*

(b) Show that $y(t)\mathbf{w}^{\mathrm{T}}(t+1)\mathbf{x}(t) > y(t)\mathbf{w}^{\mathrm{T}}(t)\mathbf{x}(t)$. *[Hint: Use (1.3).]*

(c) As far as classifying $\mathbf{x}(t)$ is concerned, argue that the move from $\mathbf{w}(t)$ to $\mathbf{w}(t+1)$ is a move 'in the right direction'.

7. (10 points) Problem 1.2 from textbook:

Problem 1.2 Consider the perceptron in two dimensions: $h(\mathbf{x}) = \mathrm{sign}(\mathbf{w}^{\mathrm{T}}\mathbf{x})$ where $\mathbf{w} = [w_0, w_1, w_2]^{\mathrm{T}}$ and $\mathbf{x} = [1, x_1, x_2]^{\mathrm{T}}$. Technically, $\mathbf{x}$ has three coordinates, but we call this perceptron two-dimensional because the first coordinate is fixed at 1.

(a) Show that the regions on the plane where $h(\mathbf{x}) = +1$ and $h(\mathbf{x}) = -1$ are separated by a line. If we express this line by the equation $x_2 = ax_1 + b$, what are the slope a and intercept b in terms of w_0, w_1, w_2?

(b) Draw a picture for the cases $\mathbf{w} = [1, 2, 3]^{\mathrm{T}}$ and $\mathbf{w} = -[1, 2, 3]^{\mathrm{T}}$.

In more than two dimensions, the $+1$ and -1 regions are separated by a *hyperplane*, the generalization of a line.

8. (20 points) Problem 1.4 (a) – (e) from textbook:

Problem 1.4 In Exercise 1.4, we use an artificial data set to study the perceptron learning algorithm. This problem leads you to explore the algorithm further with data sets of different sizes and dimensions.

(a) Generate a linearly separable data set of size 20 as indicated in Exercise 1.4. Plot the examples $\{(\mathbf{x}_n, y_n)\}$ as well as the target function f on a plane. Be sure to mark the examples from different classes differently, and add labels to the axes of the plot.

(b) Run the perceptron learning algorithm on the data set above. Report the number of updates that the algorithm takes before converging. Plot the examples $\{(\mathbf{x}_n, y_n)\}$, the target function f, and the final hypothesis g in the same figure. Comment on whether f is close to g.

(c) Repeat everything in (b) with another randomly generated data set of size 20. Compare your results with (b).

(d) Repeat everything in (b) with another randomly generated data set of size 100. Compare your results with (b).

(e) Repeat everything in (b) with another randomly generated data set of size 1,000. Compare your results with (b).

Submit your .ipynb file.

9. (20 points) Problem 1.5 from textbook. Submit your .ipynb file

Problem 1.5 The perceptron learning algorithm works like this: In each iteration t, pick a random $(\mathbf{x}(t), y(t))$ and compute the 'signal' $s(t) = \mathbf{w}^{\mathrm{T}}(t)\mathbf{x}(t)$. If $y(t) \cdot s(t) \leq 0$, update $\mathbf{w}$ by

$$\mathbf{w}(t+1) \longleftarrow \mathbf{w}(t) + y(t) \cdot \mathbf{x}(t) \ ;$$

One may argue that this algorithm does not take the 'closeness' between $s(t)$ and $y(t)$ into consideration. Let's look at another perceptron learning algorithm: In each iteration, pick a random $(\mathbf{x}(t), y(t))$ and compute $s(t)$. If $y(t) \cdot s(t) \leq 1$, update $\mathbf{w}$ by

$$\mathbf{w}(t+1) \longleftarrow \mathbf{w}(t) + \eta \cdot (y(t) \quad s(t)) \cdot \mathbf{x}(t) \ ,$$

where η is a constant. That is, if $s(t)$ agrees with $y(t)$ well (their product is > 1), the algorithm does nothing. On the other hand, if $s(t)$ is further from $y(t)$, the algorithm changes $\mathbf{w}(t)$ more. In this problem, you are asked to implement this algorithm and study its performance.

(a) Generate a training data set of size 100 similar to that used in Exercise 1.4. Generate a test data set of size $10,000$ from the same process. To get g, run the algorithm above with $\eta = 100$ on the training data set, until a maximum of $1,000$ updates has been reached. Plot the training data set, the target function f, and the final hypothesis g on the same figure. Report the error on the test set.

(b) Use the data set in (a) and redo everything with $\eta = 1$.

(c) Use the data set in (a) and redo everything with $\eta = 0.01$.

(d) Use the data set in (a) and redo everything with $\eta = 0.0001$.

(e) Compare the results that you get from (a) to (d).

10. (10 points) Problem 1.11 from textbook

Problem 1.11 The matrix which tabulates the cost of various errors for the CIA and Supermarket applications in Example 1.1 is called a *risk* or *loss matrix*.

For the two risk matrices in Example 1.1, explicitly write down the in sample error E_{in} that one should minimize to obtain g. This in-sample error should weight the different types of errors based on the risk matrix. *[Hint: Consider $y_n = +1$ and $y_n = -1$ separately.]*

Example 1.1 (Fingerprint verification)**.** Consider the problem of verifying that a fingerprint belongs to a particular person. What is the appropriate error measure?