Instructions

You are encouraged to discuss on course materials and homework. However, you must write the final solutions alone and **understand** them fully. You can consult books, notes or other resources, but not copy from them. Also, make sure the scan of your homework submission should be very clean/understandable. Your solutions to the problems should be submitted in a single word/pdf document (following a logical sequence) with your responses/arguments clearly presented. Supporting files such as the source codes should also be submitted if there's any programming assignment; and be well documented. You can either Matlab or Python to experiment with the PLA.

- DUE DATE: 19/09/2020

A) Problems from the Textbook

0.1 Learning Exercises - 20%

In the following exercises, you need to make your arguments convincing to get the points.

1. Do Exercise 1.1 of LFD.
2. Do Exercise 1.5 of LFD.

0.2 Perceptron Learning Algorithm -20%

In the following exercises, you need to make your arguments convincing to get the points.

1. Do Exercise 1.3-1 of LFD.
2. Do Exercise 1.3-2 of LFD.
3. Do Exercise 1.3-3 of LFD.

0.3 Experiments with Perceptron Learning Algorithm -30%

In the following exercises, you need to make your arguments convincing to get the points.

1. Generate a data set of size 20 as directed by Exercise 1.4 of LFD, and plot the examples $\{(x_n, y_n)\}$ 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.
2. Run Perceptron Learning Algorithm (PLA) on the data set above. Report the number of updates that PLA takes before converging. Plot the examples $\{(x_n, y_n)\}$, the target function f , and the final hypothesis g in the same figure. Comment on whether f is close to g.
3. Repeat everything in (2) with another randomly generated data set of size 20. Compare your results with (2).
4. Repeat everything in (2) with another randomly generated data set of size 100. Compare your results with (2).
5. Repeat everything in (2) with another randomly generated data set of size 1000. Compare your results with (2).

B) Probability Related Problems

Probability is, in many ways, the most fundamental mathematical technique for machine learning. This problem will review several basic notions from probability and make sure that you remember how to do some elementary proofs.

Recall that for a discrete random variable X whose values are integers, we frequently use the notation $P(X = x)$. If a random variable Y is continuous, we typically use a "density function" $f(Y = y)$. The conditions for $P(X = x)$ to be a valid probability distribution are that $\Sigma_{-\infty}^{\infty} P(X = x) = 1$ and $P(X = x) \geq 0$. Similarly for $f(Y = y)$ to be a valid continuous distribution, $\int_{-\infty}^{\infty} f(Y = y)dy = 1$ and $f(Y = y) \geq 0$.

Sometimes the underlying probability space has more than one variable (for example, the height and weight of a person). In this case, we may use notation like $f(X = x; Y = y)$ to denote the probability density function in several dimensions.

0.4 Independence -15%

Intuitively, two random variables X and Y are "independent" if knowledge of the value of one tells you nothing at all about the value of the other. Precisely, if X and Y are discrete, independence means that $P(X = x; Y = y) = P(X = x)P(Y = y)$, and if they are continuous, $f(X = x; Y = y) = f(X = x)f(Y = y)$. Show the following, for independent random variable X and Y:

$E[XY] = E[X]E[Y]$ for discrete and continous cases respectively.

0.5 I.I.D. assumption in spam filters -15%

Give at least 4 cases how "Independent and identically distributed random variables (i.i.d.)" assumption can be violated for spam filtering. Fore each case, please explain why i.i.d. is violated and how spammers might exploit the situation.

0.6 *Spam filtering equation- (optional bonus problem -10%)

Naive Bayes classifiers work by correlating the use of tokens (typically words, or sometimes other things), with spam and non-spam e-mails and then using Bayesian inference to calculate a probability that an email is or is not spam. Naive Bayes spam filtering is a baseline technique for dealing with spam. Let's suppose the suspected message contains the word "replica". Most people who are used to receiving e-mail know that this message is likely to be spam, more precisely a proposal to sell counterfeit copies of well-known brands of watches. The spam detection software, however, does not "know" such facts; all it can do is compute probabilities. Let $Pr(S|W)$ be the probability that a message is a spam given the word "replica" appears in it. Express $Pr(S|W)$ in terms of the following components we provide:

1. $Pr(S)$ is the overall probability that any given message is spam;
2. $Pr(W|S)$ is the probability that the word "replica" appears in spam messages;
3. $Pr(H)$ is the overall probability that any given message is not spam (is "ham");
4. $Pr(W|H)$ is the probability that the word "replica" appears in ham messages.