ECEN 5712 (M. Varanasi) **Homework 1** **Due Jan. 23, 2022**

The problems of this homework require you to work with the data sets provided. For details, see the guide given here following the problems.

1. Consider the data set named Exp.txt that contains three dimensional inputs in $\mathcal{X} = [-1, 1]^3$ with a single integer label $y \in \mathcal{Y} = \{0, 1, 2, 3\}$. You are to solve the classification problem here using exploratory data analysis via data visualization.

   a. Make scatter plots of the data set in three dimensions and provide views that you think give some insight.

   b. Make pair-wise scatter plots and histograms for each attribute as in the case shown in the lecture notes for the Iris data set. Is there a pair of attributes more informative than others?

   c. Based on your visual data exploration specify mathematically a (possibly non-linear) classification rule that minimizes empirical risk that assigns a risk of 1 to incorrectly classified inputs and 0 to correctly classified inputs. The empirical risk here is hence just equal to the ratio of incorrectly classified points divided by the number of labeled data points.

   d. What is the minimum empirical risk (ERM)?

2. Consider data sets Classify-2D-wLabels-1.txt and Classify-2D-wLabels-2.txt for this problem. Each contains labeled two-dimensional inputs with input space $\mathcal{X} = [-1, 1]^2$ with the binary label set $\mathcal{Y} = \{0, 1\}$.

   a. Do a visual inspection via scatter plots of the two data sets. Are they linearly separable?

   b. Write a computer program that implements the general Perceptron algorithm from scratch (that works with d-dimensional binary labeled data). Your program should keep a count of the number of iterations it took to obtain the classifier and provide it if needed.

   c. Implement the Perceptron algorithm given in the lecture notes on the two data sets to determine the linear classifier and compare the number of iterations it took for the algorithm to get to its final answer for each data set. Plot the decision boundary at various stages of the algorithm (create an animation if you wish) to visualize how the separating hyperplane converges to one that perfectly separates the inputs with the two labels. Give the equation for your classifier for at least two instances of different initializations of the weight vector for each of the two data sets (but chosen to be the same across data sets).

3. Consider the data sets Classify-3DwLabels-1.txt and Classify-3DwLabels-2.txt for this problem. The input space for each is $\mathcal{X} = [-1, 1]^3$ and the output space is the binary label space $\mathcal{Y} = \{0, 1\}$.

    a. Using your code for the Perceptron algorithm find a linear classifier for the data set Classify-3DwLabels-1.txt. Mathematically, specify the classifier(s) returned by your program (and initialization(s)). Plot the input points in a three-dimensional scatter plot. In the same plot, include the plane separating the two sets of points and show a view of the plot that clearly shows that the classifier "works".

    b. Repeat the exercise of Part a of this problem for the data set Classify-3DwLabels-2.txt. Explore, experiment, play, and report your observations.

Programming Guide

All datasets in this homework are given as comma-separated values, where each row corresponds to a datapoint. The last value is the integer label of the class the point belongs to and the rest are the attributes of the datapoint.

For this homework, you can use any programming language and visualization library you would like. As an easy-to-use option, we recommend Python and matplotlib. Below are some links for this setup that might be helpful for this homework.

1. For reading the dataset, you can use the native csv library.

2. You can follow the following examples as guide for generating scatter plots for 2-dimensions and 3-dimensions.

3. For Question 1.b, you might want to consider generating a matrix of plots for visualizing the histograms and the pair-wise plots. Here is a quick guide on generating subplots to do so.

4. In Question 3.a, we expect you to implement the Perceptron algorithm yourself. In the process, you can use libraries like NumPy or SciPy for mathematical functions, but you should not use an off-the-shelf implementation of the algorithm.

Please make sure the code you submit runs stand-alone. If you are submitting multiple files that contain dependencies to each other, make sure the file/folder structure in your submission is the same as your setup. We should be able to run and evaluate your code without needing any reverse engineering. If you think it is not obvious, please include comments on how you expect your code to be executed or any specific versions of languages/libraries if that matters. For any concerns about submission format or any programming-related issues, feel free post your questions on Piazza.