ECEN 5712 (M. Varanasi) **Homework 2** **Due Feb. 3rd, 2022**

Problem 2 of this homework requires you to work with the data sets provided. For details, see the guide given here following the problems.

1. This problem is on linear MMSE estimation in statistical inference: the probabilistic model for inputs and the label is given to you and you are to estimate the label as an affine function of the inputs. Let Y be the label and let X_1 and X_2 be the two components of the two-dimensional input. Suppose

$$X_i = Y + Z_i \quad i \in \{1, 2\}$$

where Y is a zero-mean random variable with variance 4, the noise random variables Z_1 and Z_2 are both zero-mean with variances 1 and 4, respectively. Moreover, Y, Z_1 and Z_2 are mutually independent.

   a. Find an affine estimator $\hat{Y}$ of Y of the form

$$\hat{Y} = aX_1 + bX_2 + c$$

   that minimizes the mean square error $E[(Y - \hat{Y})^2]$.

   b. What is the mean square error of the optimum linear MMSE estimator?

   c. Compute

$$E[(Y - \hat{Y})X_i]$$

   for each $i \in \{1, 2\}$. Interpret your answer. Hint: Think of $E[XY]$ as an inner product between the random variables X and Y.

2. In this machine learning problem, the model used to generate the input vectors and the label is unknown to you and you must find it based on assumed hypotheses classes and training data. The training dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{250}$ named Regression.txt contains 250 two-dimensional inputs in $\mathcal{X} = [-2, 2]^2$ with a single real-valued label y for each input given in the last entry of each row. You are to train the linear regression algorithm to find the predictor $\hat{y}(\mathbf{x})$ for different hypotheses classes using this dataset. The criterion for finding $\hat{y}(\mathbf{x})$ will be to minimize empirical risk or training loss defined as the mean squared error $m^{-1} \sum_{i=1}^{m} (\hat{y}(\mathbf{x}) - y_i)^2$ in each case. The other dataset named Regression-Test.txt contains 50 input points and follows the same format as Regression.txt except that the labels are missing. You are going to submit the predictions of the label for the test data using each of the trained predictor.

   a. First split the dataset into training and validation data. You will use the first 200 samples as the training data and the last 50 as the validation data. Let Polynomial(n) denote the regression algorithm that considers two-dimensional polynomials of the attributes up to degree n.

   For instance, Polynomial(1) returns the predictor of the form

$$\hat{y}(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2.$$

whereas Polynomial(2) returns the predictor of the form

$$\hat{y}(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_1 x_2 + w_5 x_2^2,$$

and so on. Implement Polynomial(1), Polynomial(2), Polynomial(3) and Polynomial(4) regression algorithms. See programming guide for a hint.

b. Train the four algorithms over increments of 20 samples, i.e. measure the algorithms' training loss after training it with the first $m = 20, 40, 60, ..., 200$ samples. After training in each case, record the training and validation losses (validation loss is defined as $50^{-1} \sum_{i=201}^{250} (\hat{y}(\mathbf{x}) - y_i)^2$).

Plot the training and validation losses as a function of m for the four hypotheses classes on the same figure. Do you observe under-fitting and/or over-fitting? Which of the hypotheses classes you trained is the best choice for this problem?

c. Train your choice of the best hypotheses class with all 200 samples in Regression.txt. Submit its output for each of the 50 test inputs in Regression-Test.txt in a 50-line text file where line i is the output of your regression algorithm for the i^{th} point in Regression.txt. We will compute the generalization loss (defined as the sample mean of the squared errors over the 50 test inputs) using the labels for the inputs in Regression-Test.txt which were generated with the same mechanism used to generate the labels in Regression.txt. The accuracy of your algorithm will determine your grade.

Programming Guide

All datasets in this homework are given as comma-separated values, where each row corresponds to a datapoint. The last value in Regression.txt is the real number y you will use for regression. The test dataset only has the attributes and no y values listed.

For this homework, you can use any programming language and linear algebra, optimization, and visualization library you would like. As easy-to-use options, we recommend Python with NumPy, SciPy and matplotlib. Below are some links for this setup that might be helpful. You are not allowed to use scikit-learn or any equivalent high-level machine learning library. If you are unsure a library you want to use might fall in this category, please ask about the library and the function(s) you are planning to use on Piazza.

1. In Question 2.b, you don't have to implement each of the required regression algorithms from scratch. By doing some feature engineering, you can get away with just implementing linear regression and reusing it as a subroutine for higher-order polynomials. Here is how scikit-learn's Pipeline mechanism handles this. You can't use the scikit-learn functions directly, but you can use the idea presented here in your own implementation. This is a commonly used method in machine learning that is sometimes called 'kernel trick'.

2. The following libraries may or may not be useful for implementing your regression algorithms depending on your design.

   (a) NumPy is your go-to for linear algebra related functions.

   (b) SciPy.optimize has easy-to-use optimization functions. Alternatively, you can use PuLP which is slightly harder to use, but offers more flexibility and control.

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.