CS 3190: Foundations of Data Analysis **Due: Tuesday 10.25 at 11:50pm**

Homework 3: Regression and Gradient Descent

Instructions: Your answers are due at 11:50, before the beginning of class on the due date. You must turn in a pdf through canvas. I recommend using latex (`http://www.cs.utah.edu/~jeffp/teaching/latex/`, see also `http://overleaf.com`) for producing the assignment answers. If the answers are too hard to read you will lose points, entire questions may be given a 0 (e.g. **sloppy pictures with your phone's camera are not ok, but very careful ones are**)

Please make sure your name appears at the top of the page.

You may discuss the concepts with your classmates, but write up the answers entirely on your own. **Be sure to show all the work involved in deriving your answers! If you just give a final answer without explanation, you may not receive credit for that question.**

We will use two datasets found here:
`http://www.cs.utah.edu/~jeffp/teaching/FoDA/x.csv`
`http://www.cs.utah.edu/~jeffp/teaching/FoDA/y.csv`
There are many ways to import data in python, the `genfromtext` command in numpy provides an easy solution. Here is a short tutorial from this course that should be helpful:
`https://www.youtube.com/watch?v=CUVX1-tsJ2Q`
`https://colab.research.google.com/drive/1PIIC6X9pMj4wBTP954F9E1XOsKCiMzRs`

1. [**50 points**] Let $\mathtt{x} \in \mathbb{R}^n$ hold the data for an explanatory variable, and $\mathtt{y} \in \mathbb{R}^n$ be the data for the dependent variable. Here $n = 120$.

   (a) [10 points] Run simple linear regression to predict `y` from `x`. Report the linear model you find. Predict the value of y for the new x values of 1 and 10.

   (b) [10 points] Split the data into a training set (the first 100 values) and the test set (the last 20 values). Run simple linear regression on the training set, and report the linear model. Again predict the y value at x values of 1 and 10.

   (c) [15 points] Using the testing data, report the residual vector (it should be 20-dimensional) for the model built on the full data, and another one using the model built just from the training data. For each residual vector, report the corresponding **root mean squared error**

   Also compute the **root mean squared error** of the residual vector for the training data (a 100-dimensional vector) for the model built on the full data, and also for the model built on the training data.

   (d) [15 points] Expand data set `x` into a $n \times (p+1)$ matrix $\tilde{X}_p$ using standard polynomial expansion for $p = 3$. Report the first 3 rows of this matrix.

   Build and report the degree-3 polynomial model using this matrix on the training data.

   Report the **root mean squared error** of the residual vector built for the testing data (from a 20-dimensional vector) and for the training data (from a 100-dimensional vector).

2. **[25 points]** Consider a data set (X, y) where $X \in \mathbb{R}^{n \times 4}$; and its decomposition into a test dataset (X_{test}, y_{test}) and a training data set (X_{train}, y_{train}). Assume that X_{train} is not just a subset of X, but also prepends a column of all 1s. We build a linear model

$$\alpha = (X_{train}^T X_{train})^{-1} X_{train}^T y_{train}$$

where $\alpha \in \mathbb{R}^5$. The test data (X_{test}, y_{test}) consists of 3 data points (x_1, y_1), (x_2, y_2), and (x_3, y_3) where $x_1, x_2, x_3 \in \mathbb{R}^4$. Explain how to use (write a mathematical expression) this test data to estimate the generalization error. That is, if one new data point arrives x, how much squared error would we expect the model α to have compared to the true unknown value y?

3. **[25 points]** Consider two functions

$$f_1(x, y) = (x - 3)^2 + 3(y + 1)^2 \qquad f_2(x, y) = (1 - (y - 3))^2 + 10((x + 4) - (y - 3)^2)^2$$

For (a) - (c) below, you also need to report the function value, gradient, and 2-norm of the gradient at the end of each iteration – one set of values per line. This is a common way to see if your results make sense, and is not hard to quickly read through.

   (a) [8 points] Run gradient descent on f_1 with starting point $(x, y) = (0, 0)$, $T = 30$ steps and $\gamma = .01$

   (b) [8 points] Run gradient descent on f_1 with starting point $(x, y) = (10, 10)$, $T = 100$ steps and $\gamma = .03$

   (c) [9 points] Run any variant of gradient descent you want for f_2 with starting point $(x, y) = (0, 2)$. Try to get the smallest function value after $T = 100$ steps. Also explain in a couple of sentences the procedure you used.

[+5 points] *If any students do significantly better than the rest of the class on f_2 in part (c), we will award up to 5 extra credit points. To obtain extra points, a detailed description of how the gradient descent is performed is required.*