CS 3190: Foundations of Data Analysis **Due: Wednesday 10.30 at 2:45pm**

Homework 3: Regression and Gradient Descent

Instructions: Your answers are due at 2:45, 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/`) 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.

1. [**50 points**] Let $\texttt{x} \in \mathbb{R}^n$ hold the data for an explanatory variable, and $\texttt{y} \in \mathbb{R}^n$ be the data for the dependent variable. Here $n = 100$.

   (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 0.4 and 0.7.

   (b) [10 points] Split the data into a training set (the first 70 values) and the test set (the last 30 values). Run simple linear regression on the training set, and report the linear model. Again predict the y value at x value of 0.4 and of 0.7.

   (c) [15 points] Using the testing data, report the residual vector (it should be 30-dimensional, use the absolute value for each entry) for the model built on the full data, and another one using the model built just from the training data. Report the 2 norm of each vector.

   Also compute the 2-norm of the residual vector for the training data (a 70-dimensional vector) for the model build 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 $p = 5$. Report the first 5 rows of this matrix.

   Build and report the degree-5 polynomial model using this matrix on the training data.

   Report the 2 norm of the residual vector built for the testing data (from a 30-dimensional vector) and for the training data (from a 70-dimensional vector).

2. [**25 points**] Consider input data (X, y) where $X \in \mathbb{R}^{n \times d}$, and assume the rows are drawn iid from some fixed, and unknown distribution. Describe three ways to solve for a model $\alpha \in \mathbb{R}^{d+1}$ towards minimizing $\|X\alpha - y\|$; describe what the methods are – do not just list different commands in python. Explain potential advantages of each – these advantages may depend on the values of n and d.

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

$$f_1(x, y) = (x - 5)^2 + 2(y + 3)^2 + xy \qquad f_2(x, y) = (1 - (y - 3))^2 + 10((x + 4) - (y - 3)^2)^2$$

Starting with $(x, y) = (0, 2)$ run the gradient descent algorithm for each function. Run for T iterations, and report the function value at the end of each step.

(a) First, run with a fixed learning rate of $\gamma = 0.05$ for f_1 and $\gamma = 0.0015$ for f_2.

(b) Second, run with any variant of gradient descent you want. Try to get the smallest function value after T steps.

For f_1 you are allowed only $T = 10$ steps. For f_2 you are allowed $T = 100$ steps.

[+5 points] *If any students do significantly better than the rest of the class on f_2 in part (b), 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.*