Project 9

Developing a Vectorized Solution for Linear Regression

The goal of linear regression is to minimize the cost function:

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

where the hypothesis is:

$$h_\theta(x^i) = \theta_0 x_0^i + \theta_1 x_1^i + \cdots + \theta_n x_n^i$$

where $x_0^i = 1$, x_j^i is the jth feature or the ^{ith} sample and theta is a vector of weights (or parameters) to be adjusted such that $J(\theta)$ is minimized.

Minimization involves taking the partial derivative of $J(\theta)$ with respect to theta and multiplying times a learning rate, α, and subtracting this from each element in the vector, Θ. We called this term, $slope_j$.

$$slope_j = \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^i$$

During each step of the gradient descent loop, the values of theta come closer to values that will minimize the cost function.

A vectorized representation, informally, is a collection of symbols represented as matrices and manipulated by operators symbolizing matrix multiplication, matrix transposition, matrix subtraction, as described in class. The first three parts of this assignment ask you to vectorize important elements of linear regression. This involves a mathematical argument of the sort I made in class for the much more complex vectorization of logistic regression. You must define your matrices and vectors, expand them when appropriate to indicate what is being argued, and do size analysis before each matrix multiplication. Further, your solutions must be clearly and carefully written on lined paper. If you can't write clearly and carefully, you must use a word processor. Just providing me with some vector operations is necessary but not sufficient. You must show how you moved from an iterative to a vectorized solution.

A. Vectorize the hypothesis used in linear regression as defined in class and above
B. Vectorize $slope_j$ as defined in class and above
C. Vectorize gradient descent as defined in class. The answer will be a single for loop (do until convergence) with one line of matrix operations within it. Hint: in B you vectorized $slope_j$.

Now, produce an expression for a vector of slopes, one for each feature x_j. Finally, use the vector of slopes to produce the single line of matrix operations necessary for gradient descent.

D. Write a on octave program, asgn14.m, that:

- Reads in a csv data file (data.csv) found in my github repository
- Stores the data into vectors X and Y, adjusting X to include a column of 1s since $x_0 = 1$
- Does a scatter plot of the X,Y vectors, with markers of shape and color of your choosing.
- Leaves the plot on the screen for further plots.
- Invokes a gradient descent function that will return the new and improved theta vector:
  function [theta] = gradientDescent(X, Y, theta, alpha, num_iterations)
  This function uses the vectorized representations you developed in C
  This function is saved as gradientDescent.m and is invoked by asgn14.m
- Plots $h_\theta(X)$ as a function of X where *X* is the vector extracted from data.csv, on the same figure as the scatter plot, above. The vector resulting from the action of the hypothesis on *X* must be vectorized.

To do Part D you will need initial theta values, a learning parameter, and a stopping condition.

- Set the theta values to 0.0
- Set the learning rate to 0.0001
- Set the stopping condition to a fixed number of iterations, 1000

You'll find a very nice explanation of linear regression along with an iterative solution at:

https://spin.atomicobject.com/2014/06/24/gradient-descent-linear-regression/

It will give you some appreciation for the vectorized solution that you are developing.

Matt Nedrich is a software developer at Atomic Object. The data used for this problem was downloaded from his github repository, a link to which you'll find at the link above.

We've done some octave in class. I've added a dozen or so examples to octaveExamples.txt that will round out what you need to know to do this project.

Finally, on the class website there is a solution for the vectorization of logistic regression. If you can follow this, you're on your way to produce a vectorized solution to logistic regression.

Submit:

- Your mathematical argument for vectorized solutions
- Hard copies of gradientDescent.m and asgn14.m
- Zipped, electronic submission of asgn14.m and gradientDescent.m

Have fun. This project might seem hard at first. Once you understand what's going on, it will be as if the clouds have parted and the land is bathed in golden light. To get to this happy place might take a trip or two to the gym or my office, whichever you prefer.

Finally, you might be tempted to scour the internet for a solution. Don't. It is not only unethical to do so (because I've told you not to), you'll also miss the parting of the clouds and "golden apples of the sun."