**CSCE 623 Spring 2019: Machine Learning. In-Class Work, Day 5 (09 Apr 2019)**

From Chapter 3: Linear Regression with “approximated” Gradient Descent (to be completed when instructed)

1. Least Squares Model Fitting Exercise
   1. Write pseudocode for a primitive method for determining the least-squares model fit in 1-variable linear regression (to find the 2 coefficients: *β*0 & *β*1 )
      * Your numerical observations are stored in matrix X. For each observation, assume you are given x1 and the corresponding y.
      * Think: One way to envision the process is as a learning algorithm which is using search to learn the best answer. If there was a cost function with only one minimum, then a greedy search could find it. Could you use a greedy search in two dimensions simultaneously? If so, how would you select the next search point?
      * Hint: RSS has only one global/local minimum, Thus, if you find that minimum, you’ve found the best values for the betas. When you are not at the minimum you could compute a “local gradient” near a value of *βi* by computing the RSS change occurring from an epsilon increase of the coefficient: RSS(*βi*+*ε*)-RSS(*βi’-ε*).   
        How would you use these local gradients to search for a best set of beta values? How would you know when to stop (what is your convergence criteria)?
   2. How would you extend your idea to a general multiple linear regression model fitting algorithm with more than 2 coefficients?

Linear Regression with poor-man’s Gradient Descent - Python Practice (to be completed when instructed):

* Implement your pseudocode in python.
* Then, using the dataset from a previous in-class example (repeated below), use your code to automatically hone in on *β*0 & *β*1 when you started with them both at 1.
* Improving your code: If your pseudocode used a separate computation for each beta, can you think of a way to vectorize the calculations such that all betas are updated simultaneously?
* Debugging your code -
  + Report (print out) the values of RSS as your system approaches the correct values. Are they shrinking? If not, you might be changing the betas in the wrong direction.
  + At some point the RSS won’t change much. When you get there you should stop.
* Were you able to get to *β0* and = 26.768, and *β1*= 0.644?
* Predict the value of someone who got an 80 on their aptitude test. Did you get about 78.288?

|  |  |  |
| --- | --- | --- |
| Student (i) | X1 | y |
| 1 | 95 | 85 |
| 2 | 85 | 95 |
| 3 | 80 | 70 |
| 4 | 70 | 65 |
| 5 | 60 | 70 |