CSE514 – Fall 2021 Programming Assignment 1

This assignment is to enhance your understanding of objective functions, regression models, and the gradient descent algorithm for optimization. It consists of a programming assignment (with two optional extensions for bonus points) and a report.

Topic

Design and implement a gradient descent algorithm or algorithms for multi-variate regression.

Programming work

A) Uni-variate linear regression

In the lecture, we discussed uni-variate linear regression $y = f(x) = mx+b$, where there is only a single independent variable x.
Your program must specify the objective function of least squares and be able to apply the gradient descent algorithm for optimizing a uni-variate linear regression model.

B) Multi-variate linear regression

In practice, we typically have multi-dimensional (or multi-variate) data, i.e., the input **x** is a vector of features with length p. Assigning a parameter to each of these features, plus the b parameter, results in p+1 model parameters. Multi-variate linear models can be succinctly represented as:

$y = f(\mathbf{x}) = (\mathbf{a} \cdot \mathbf{x})$ (i.e., dot product between **a** and **x**),

where $\mathbf{a} = (a_0, a_1, \ldots, a_p)^T$ and $\mathbf{x} = (1, x_1, \ldots, x_p)^T$, with a_0 in place of b in model.
Your program must be able to apply the gradient descent algorithm for optimizing a multi-variate linear regression model.

C) Optional extension 1 – Multi-variate polynomial regression

For bonus points, extend to a multi-variate quadratic regression model.

D) Optional extension 2 – Sparse multi-variate regression

For bonus points, extend the model by adding a penalty term to learn a sparse model.
As I briefly mentioned in my lecture, sparseness can be introduced to a regression model by adding the L1 regularization term to the objective function as a penalty. You may choose a different penalty term if you wish, as long as you describe it clearly and it successfully learns a sparse model.

IMPORTANT: Regression is basic, so there are many implementations available. But you MUST implement your method yourself. This means that you cannot use a software package and call an embedded function for regression (or Lasso regression) within the package. You may use the other basic functions provided by the package, but the gradient descent and regression algorithm must be implemented by yourself.

Data to be used

We will use the Concrete Compressive Strength dataset in the UCI repository at

UCI Machine Learning Repository: Concrete Compressive Strength Data Set

(https://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength)

Note that the last column of the dataset is the response variable (i.e., y).

There are 1030 instances in this dataset.

Use the first 900 instances for training and the last 130 instances for testing. This means that you should learn parameter values for your regression models using the training data and then test the model using the testing data.

What to submit – follow the instructions here to earn full points

- (80 pts total) The report
  - Introduction
    - (10 pts) Your description/formulation of the problem,
    - (10 pts) the details of your algorithm (e.g., parameter updating rules),
    - (5 pts) and pseudo-code
  - Results
    - (10 pts) MSEs of your models on the training dataset when using only one of the predictor variables (uni-variate regression) and when using all eight (multi-variate regression). You should have a total of nine MSE results.
    - (10 pts) MSEs of your model on the testing data points. Again, you should have a total of nine MSE results.
    - (10 pts) Plots of your trained uni-variate models on top of scatterplots of the training data used (please plot the data using the x-axis for the predictor variable and the y-axis for the response variable).
  - Discussion
    - (10 pts) Describe how the different models compared in performance on the training data. Did the same models that performed well on the training data do well on the testing data?

        - (10 pts) Describe how the performance of the uni-variate models predicted or failed to predict which features were “more important” in the multi-variate model(s).
        - (5 pts) Draw some conclusions about what factors predict concrete conpressive strength.
    - Note: as a guideline, I expect the report to be about 10 pages, mostly to display results. Please label your plots and tables with enough information that a reader can understand them without referring back to the text.

- (20 pts total) Your program (in a language you choose) including
    - (15 pts) The code itself
    - (5 pts) Brief instructions on how to run your program (input/output plus execution environment and compilation if needed) – in a separate file
    - Note: We won’t grade your program’s code for good coding practices or documentation. However, if we find your code difficult to understand or run, we may ask you to run your program to show it works on a new dataset.

- (10 bonus points) Include in your report the details of your algorithm extensions that enable quadratic regression, and the results from fitting a quadratic model to the data.

- (10 bonus points) Include in your report the details of your algorithm extensions that enable sparse regression, and the results from fitting a sparse model to the data.

Due date

Wednesday, October 13 (midnight, STL time). Submission to Gradescope via course Canvas.