Gradient Descent for Linear Regression

In this problem you will be working with three datasets for regression:

- **Housing:** This is a regression dataset where the task is to predict the value of houses in the suburbs of Boston based on thirteen features that describe different aspects that are relevant to determining the value of a house, such as the number of rooms, levels of pollution in the area, etc.
- **Yacht:** This is a regression dataset where the task is to predict the resistance of a sailing yacht's structure based on six different features that describe structural and buoyancy properties.
- **Concrete:** This is a regression dataset where the task is to predict the compressive strength of concrete on eight different features. There are a total of 1030 instances and all the features are numeric.

Recall, the model for linear regression:

$$y = f_w(x) = \sum_{i=1}^{m} w_i x_i$$

and the task is to estimate the least squares error (LSE) regression coefficients w i for predicting the output y based on the observed data x, using gradient descent.

1. Normalize the features in the training data using z-score normalization.

2. Initialize the weights for the gradient descent algorithm to all zeros i.e., $w = [0, 0, \ldots, 0]^T$.

3. Use the following set of parameters:

(a) Housing: learning rate = 0.4×10^{-3}, tolerance = 0.5×10^{-2}

(b) Yacht: learning rate = 0.1×10^{-2}, tolerance = 0.1×10^{-2}

(c) Concrete: learning rate = 0.7×10^{-3}, tolerance = 0.1×10^{-3}

NOTE: Here tolerance is defined based on the difference in root mean squared error (RMSE)

measured on the training set between successive iterations. Where, RMSE is defined as:

$$RMSE = \sqrt{\frac{SSE}{N}}$$

N is the number of training instances.

4. Be sure to set a maximum number of iterations (recommended maximum iterations = 50000) so that the algorithm does not run forever.

• **Normalization**: In the previous assignment we normalized the entire dataset prior to learning, and then used ten-fold cross validation to evaluate the performance of the learning algorithm. However, the correct way of normalizing data would be to normalize the training data and record the normalization parameters i.e., mean and standard deviation for z-score normalization and min and max feature values for feature re-scaling. This minimizes the chances of introducing any bias during the performance evaluation of the learning algorithm. In a real-world scenario you would not have access to test data during the training phase.

To **summarize**: only normalize your training data, and then use the normalization parameters to normalize your test data and then estimate the accuracy/error.

• **Adding the Constant Feature**: For every regression problem remember to add a column of ones to your dataset.

Least Squares Regression using Normal Equations

Use the housing and yacht dataset to estimate the regression weights using normal equations. Contrast the performance (measured through RMSE) to the results obtained using the gradient descent algorithm. In this problem you will calculate the analytical solution that we obtained through Normal equations to learn your weight vector, and contrast the performance (training and test RMSE) for your gradient-descent based implementation for problem-1.