CS 6140: Machine Learning	**Spring 2015**
College of Computer and Information Science	
Northeastern University	
Assignment 2	**February, 09**
Instructor: Bilal Ahmed	**Due Date: February, 21**

Linear Regression

1 Pre-requisites

- **Plotting:** In this assignment we will be generating a number of plots to visualize the results. Visualization plays a very important role in ML and most of the times you would be looking at different kinds of graphs/charts to get some key insights about your data and the performance of different ML algorithms. There are a number of third party plotting libraries and toolkits available for different platforms and you can select any of them you wish to work with.
- **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.

2 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.

2.1 Learning Regression Coefficients using Gradient Descent (*60 points*)

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 $\mathbf{x}$, 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 = 1000) so that the algorithm does not run forever.
5. For both datasets use ten-fold cross-validation to calculate the RMSE for each fold and the overall mean RMSE. Summarize your results for each dataset as a table where you report the SSE for each fold and also the average SSE and its standard deviation across the folds.
6. Select any fold, and plot the progress of the gradient descent algorithm for each dataset separately in two different plots. To this end plot the RMSE for each iteration.

2.2 Interpreting the results (*Extra-credit*)

For this part you can briefly discuss the impact of the different parameters on the convergence of gradient descent and the performance of the linear regression model. Please, provide specific parameter values and empirical evidence to support your claims. You can also include plots to further elucidate your claims/hypotheses.

1. How sensitive were the results to different starting weights? *(Hint: you can do multiple runs with the weights initialized randomly in* $[0, 1]$ *or* $[-1, 1]$*).*
2. Does the tolerance parameter highly affect the results?
3. How was the convergence of the gradient descent algorithm affected by different values of learning rates?

3 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, based on a ten-fold cross validation scheme.

4 Deriving Normal Equations for Univariate Regression

Consider the linear regression model for the case of a single input and output variable:

$$y = f_w(x) = w_0 + w_1 x$$

Assume that you have a training dataset consisting of N observations $(x_i), y_i)$ $\forall i \in \{1, 2, \ldots, N\}$. Find the values of w_0 and w_1 that have the minimum sum of squares error on the dataset.

5 Polynomial Regression

In this problem you will be working with the yacht dataset from the first problem and in addition we have a new dataset:

- **Sinusoid Dataset**: This is an artificial dataset created by randomly sampling the function $y = f(x) = 7.5\sin(2.25\pi x)$. You have a total of 100 samples of the input feature x and the corresponding noise corrupted output/function values y. In addition to this you are also given a validation set that has 50 samples.

5.1 Polynomial Regression using Normal Equations

You will be using your implementation of the normal equations from Problem-2. For ten fold cross-validation: calculate RMSE for each fold (for training data use the number of training instances (as N for calculating RMSE) and for test data use the number of test instances (as N for calculating RMSE)).

- **Sinusoid dataset:** Since, there is only one input feature, add higher powers of the input feature $p \in \{1, 2, 3, \ldots, 15\}$ and calculate the RMSE on the validation set. Remember that you are not required to do cross-validation, and you do not need to normalize the input feature values. Summarize your results by plotting the mean SSE on both the training and the validation set (on the same plot) versus $\max(p)$.
- **Yacht dataset:** In this problem you add higher powers of the six input features, $p \in \{1, 2, 3, \ldots, 7\}$ and calculate the mean RMSE using ten fold cross validation. For example, if $p = 2$ you will have twelve features corresponding to the original six input features and six new features obtained by squaring the original feature values. Remember, to employ the normalization scheme as outlined previously. Summarize your results by plotting the mean RMSE across folds on both the training and the validation set (on the same plot) versus $\max(p)$. **Note:** During every iteration of the cross validation you will calculate the RMSE for both the training and test data. The final quantity that you will plot will be the average of the RMSE across the ten folds (divide the sum of RMSE for each fold by 10).

NOTE: Do not duplicate the constant feature as you add new sets of features. Remember, that the final design matrix should only have a single columns of ones (preferably, the first column)

5.2 Interpreting the results (*Extra-credit*)

1. Does the addition of new features reduce the RMSE for both datasets? Is the impact of adding new features on the RMSE identical for both training data and test (validation) data?

2. How can we scale this approach to include higher order polynomials and also cross-terms between features? Do you think that this is an efficient approach?

6 The Hat Matrix

Using the Normal Equations we found that the least squares solution for linear regression is:

$$\mathbf{w} = (X^T X)^{-1} X^T \mathbf{y}$$

We can calculate the output values for our training data using the least squares solution as:

$$\hat{\mathbf{y}} = X\mathbf{w} = X(X^T X)^{-1} X^T \mathbf{y}$$

the matrix $H = X(X^T X)^{-1} X^T$ is also known as the "hat" matrix since it puts the hat on y. Show that the hat matrix H is:

1. Symmetric i.e, $H^T = H$, and

2. Idempotent i.e., $HH = H$