Machine Learning
Lab Assignment 1

DUE DATE: MARCH 22, 2019

OBJECT OF THE ASSIGNMENT:

To solve and understand linear regression problems by using gradient descent algorithm.

PROBLEM:

Implement linear regression with one or multiple independent variables to predict a dependent (target) variable.

INPUT OF THE PROBLEM:

Training dataset / testing examples

OUTPUT OF THE PROBLEM:

(a) Display the coefficients of the hyperplane, i.e., $\mathbf{w} = (w_0, w_1, \ldots, w_M)$.

(b) Predict values of the dependent variable for testing examples.

TESTING CASES:

1. The file `data1.txt` contains a training set of Auto Insurance in Sweden. The first column (= x) is the number of claims and the second column (= y) is the total payment for all the claims in thousands of Swedish Kronor.
   (a) Find the regression line $\hat{y} = w_0 + w_1 x$. Thus, use the gradient descent algorithm to find the weight $\mathbf{w} = (w_0, w_1)$.
   (b) Once you have found the regression equation, you can use the model to make predictions.
      (i) What is the predicted value of y when $x = 45$?
      (ii) What is the predicted value of y when $x = 25$?

2. The file `data2.txt` contains a training set of Basketball. Columns 1 – 4 are the feature variables x_1, x_2, x_3, x_4, and column 5 is the target variable y. The following data (x_1, x_2, x_3, x_4, y) are for each player:
   x_1 = height in feet
   x_2 = weight in pounds
   x_3 = percent of successful field goals (out of 100 attempted)
   x_4 = percent of successful free throws (out of 100 attempted)
   y = average points scored per game

(a) Find the hyperplane $\hat{y} = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4$. Thus, use the gradient descent algorithm to find the weight $\mathbf{w} = (w_0, w_1, w_2, w_3, w_4)$.

(b) Once you have found the hyperplane, you can use the model to make predictions.

(i) What is the predicted value of y when $(x_1, x_2, x_3, x_4) = (6.8, 210, 0.402, 0.739)$?

(ii) What is the predicted value of y when $(x_1, x_2, x_3, x_4) = (6.1, 180, 0.415, 0.713)$?

APPENDIX:

Stochastic Gradient Descent Algorithm

Step 1 Input training data set:

D = $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$ // N training examples

Step 2 Initialize $\mathbf{w}$, and choose a learning rate η.

// $\mathbf{w} = (w_0, w_1, \ldots, w_M)$; initialize all weights w_i to random values

Step 3 UNTIL a termination condition is met, DO

Step 4 FOR each training example $\mathbf{x}_n \in D$

$$\mathbf{w} \leftarrow \mathbf{w} + \eta\,(y_n - \mathbf{w}^T\mathbf{x}_n)\mathbf{x}_n$$

$$// \;\; \mathbf{w}^T\mathbf{x}_n = [w_0, w_1, \ldots, w_M]\begin{bmatrix} x_0 \equiv 1 \\ x_1 \\ x_2 \\ \vdots \\ x_M \end{bmatrix}$$

$$// \qquad = w_0\,x_0 + w_1x_1 + \ldots + w_Mx_M$$

REMARKS:

Stopping criteria of the **Gradient descent** algorithm usually includes:

(a) Stop when a maximum number of iterations has been exceeded.

(b) Stop when some error measure on the training set is small enough.

For example:

Let y_n = dependent variable, and $\widehat{y_n}$ = estimated value = $\mathbf{w}^T\mathbf{x}_n$. Then

(i) Mean Squared Error (MSE) = $\frac{1}{N}\sum_{n=1}^{N}(y_n - \widehat{y_n})^2$

(ii) Root Mean Squared Error (RMSE) = $\sqrt{\text{MSE}}$

(iii) Mean Absolute Error (MAE) = $\frac{1}{N}\sum_{n=1}^{N}|y_n - \widehat{y_n}|$