CS F320 Assignment 2

Abhinav Sukumar Rao[1], Pratyush Banerjee[2], and Rohan Daniel[3]

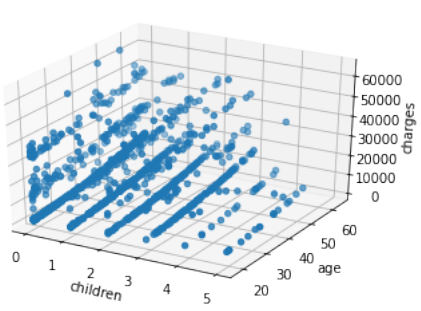
[1]2018A7PS0172H
[2]2018A7PS0312H
[3]2018A7PS0584H

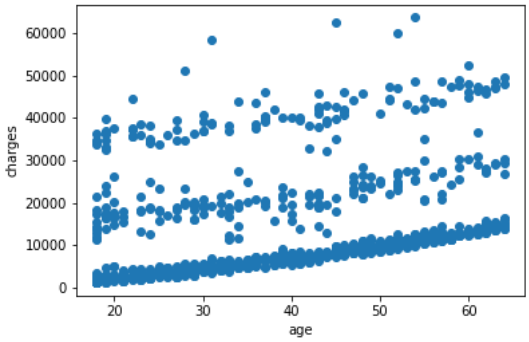
1 Introduction and Overview

This is our report on the implementation of Linear Regression using three methods i.e. a) Solving by normal equations b) Gradient descent c) Stochastic Gradient Descent

2 Pre-Processing

The independent variable consists of three features 'age', 'bmi' and 'children'. The target variable is 'charges'. We plotted 'charges' against the individual features to look for any correlation. It was observed that charges and bmi are not strongly correlated. We then made a 3D scatter plot for charges, age and children. We considered dropping the bmi feature but it did not significantly affect the model.





For feature scaling we standardized all the features and the the target variable by subtracting the mean and dividing by the variance. We decided to use this instead of normalization because the data is approximately normally distributed.

We then created a random 70-30 split to aid in training and testing. We created 20 such splits to build 20 regression models.

3 Model

3.1 Solving by Normal Equations

The regression equation in the expanded form is

$$\widehat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

In a vectorized form it can be written as follows

$$\widehat{y} = \theta^T \cdot x$$

The parameters (weights) can be obtained by solving the following algebraic equation

$$\widehat{\theta} = (X^T \cdot X)^{-1} \cdot X^T \cdot y$$

3.1.1 Implementation

```
def normal_equation(X, y):
  theta = np.dot(np.linalg.inv(np.dot(X.T, X)), np.dot(X.T, y))
  return theta
```

3.2 Gradient Descent

Gradient descent is an iterative optimization algorithm that minimizes a function by moving in the direction of negative of gradient. All the training examples are used at once to calculate the gradient.

3.2.1 Implementation

```
def grad_GD(W, X, y):
      return X.T.dot(X.dot(W) - y)

def GD(X, y, alpha = 0.01, epochs = 1000):
  W = np.random.rand(X.shape[1], 1) * 0.01
  for j in range(epochs):
      W = W - alpha/len(X) * grad_GD(W, X, y)
  return W
```

3.3 Stochastic Gradient Descent

Stochastic gradient descent is another iterative optimization algorithm like gradient descent. However, it uses one training example at a time to calculate the gradient. This greatly reduces the computational cost.

3.3.1 Implementation

```
def grad_SGD(W, X, y):
    return X * (X.dot(W) - y)

def SGD(X, y,alpha = 0.01, epochs = 1000):
    W = np.random.rand(X.shape[1], 1) * 0.01
    for j in range(epochs):
        i = randint(0,len(X)-1)
        W = W - alpha * grad_SGD(W, X[i], y[i]).reshape(-1,1)
    return W
```

4 Results

We calculated MSE for train and tests data for 20 random splits then found their mean and variance.

4.1 Normal Equations

Solving by normal equations gives the minimum train data error mean since it gives an exact solution by solving the algebra. Test error mean is the greatest because the model is fit in such a way.

4.2 Gradient Descent

Gradient Descent gives greater train data error than normal equations. It approximates the minima for the cost function. Test error mean is lower.

4.3 Stochastic Gradient Descent

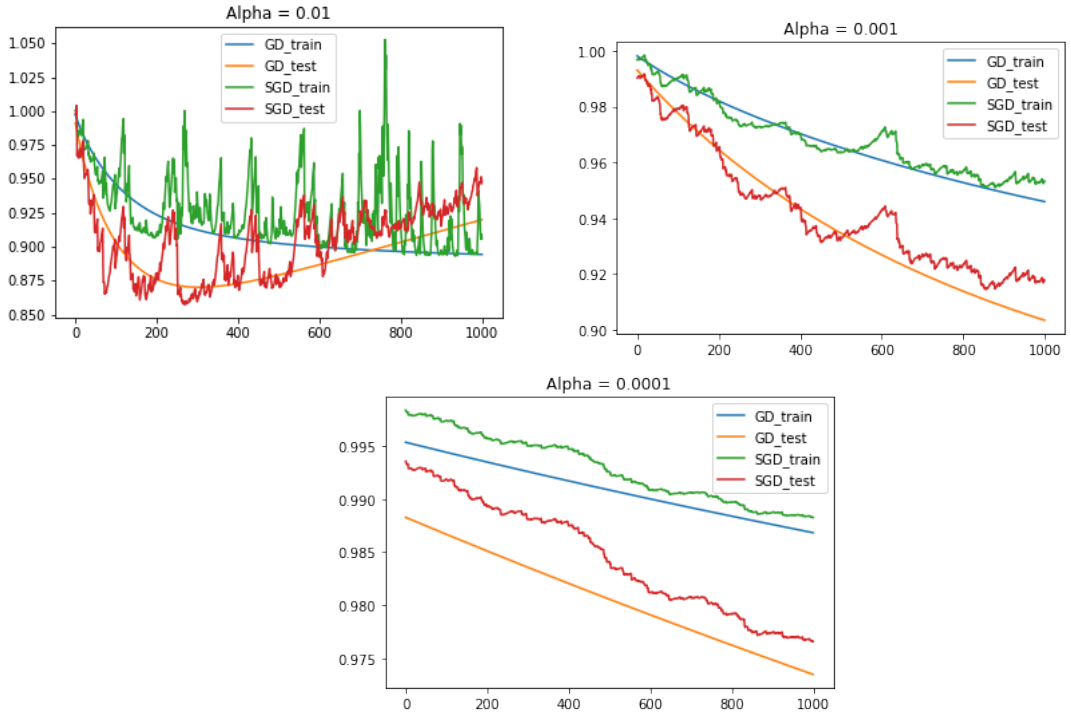
Stochastic Gradient Descent gives the greatest train data error. It uses one data point at a time randomly for each iteration. Test error mean is the lowest.

	Test Error Mean	Test Error Variance	Train Error Mean	Train Error Variance
Normal Equations	2.1035	0.4383	0.8791	6.1998e-05
Gradient Descent	0.9703	0.0018	0.8856	5.2149e-5
Stochastic Gradient Descent	0.9863	0.0019	0.8950	0.0003

Table 1: Errors observed from each method

5 Error vs. Epochs

We plotted graphs of error vs. epochs with various learning rates as below:



6 Question and Answers

i) GD and SGD are giving roughly the same results, normal equation method being a little better than the other two. In the real world scenario however, we would prefer using SGD or minibatch-GD.

ii) Normalization is about scaling using the maximum and minimum values of features as baselines so that the features lie between 0 and 1. Standardization is about scaling the features as if they were centered at the mean with a unit standard deviation. As aforementioned in the sections above, we have gone with standardization as the features are approximately normally distributed (in data that models the real world).

iii) Increasing the number of iterations does affect the loss. More iterations allows the training loss to be minimized further. Testing loss tends to decrease for some iteration and then increase. For a very large number of epochs, training loss will tend to zero and testing loss will be greater than the minimum due to the fact that we don't consider test data while training.

iv) The error function for Gradient descent was a smooth curve but SGD was spiky, filled with ups and downs. This is due to randomly sampling points, which are present at different locations.

v) The error will most likely not converge: It may either oscillate around the optimal value or may even diverge a bit for a few iterations depending on how much the weights are updated.

vi) Yes, the minimum error would have changed if the the bias term is not used. Since there is no bias term, the training examples would not be fit as closely as it would have been when using a bias term. Therefore, the error increases.

vii) The weight vector signifies the importance of the different features. It represents the coefficients of the regression equation. The higher the magnitude of the coefficient for the corresponding feature, the more influential it is on the model.
According to the coefficients obtained through the normal equations method, number of children has the maximum influence on the target value (insurance amount) and age has the minimum influence.
However, looking at further metrics such as standard error, and the Regression coefficient, we notice that BMI and number of children have high error, and low coefficients when plotted against charges individually. Additionally, through the plots obtained above, we see a linear relation between age and charges, which is not so evident for other features, especially BMI. As a result, we can say that age actually has the maximum influence while BMI, the minimum.