LAB-3 MODEL FITTING

Madhumita Krishnan

9/25/2018

1 Introduction

This report addresses the problem of model fitting. Model fitting refers to fitting a mathematical function or curve that best describes the given data set. We use Pearson's Chi-squared error metric as a measure of goodness of fit, which is the sum of squared differences between the best fitting solution and the collective set of data values. Lower error value indicates good fit and a error value of zero indicates precise fit.

The main problem encountered in model fitting is the trade-off between conflicting demands of simplicity and accuracy. Increasing the number of parameters or increasing the degree of polynomials increases precision in model fitting but also increases the complexity and hence computational costs.

Iterative methods such as gradient descent can also be used if the number of data points(n) is too large. If n<10000 then normal equations provide the solution almost instantaneously.

The assignment consists of three parts. The first and second part requires us to fit a 2D line for the given data points. The third part provides the data of 3398 meals eaten by 83 different people. The objective is to propose a model which best describes the relationship between number of bites taken in a meal and the number of kilo-calories per bite in that meal. Here we first attempt at fitting a linear model and then move on to more complex models which would be more appropriate for the data in hand. Normal equations are used in all cases to derive the model coefficients.

2 Methods

In linear regression the normal equations are widely used to find the optimal parameters. Normal equations are got by minimizing the chi-squared error metric. Let $(x_1, y_1), (x_2, y_2) - - - (x_n, y_n)$ be a given set of data points .We wish to fit a function of the form

$$y = a_1 f_1(x) + a_2 f_2(x) + - - - - - - + a_n f_n(x) \tag{1}$$

where $f_1(x), f_2(x) - - - (f_n(x)$ are basic functions and $a_1, a_2 - - - a_n$ are the unknown parameters which have to be optimized to obtain a best fit for the data.The normal equation for determining the unknown parameters is

$$(A^T b) = A^T A X \tag{2}$$

where A and b are matrices that are formed from the given data and X contains the model coefficients.The solution for this normal equation is given by the equation

$$X = (A^T A)^{-1} A^T b \tag{3}$$

x	y
5	1
6	1
7	2
8	3
9	5

Table 1: Data used to fit a line for part-1.

2.1 2D Line fitting

The data given for part-1 of the lab is listed in Table 1 and the data for part-2 of the lab includes the point $(8, 14)$ in addition to the points given in Table 1 Here in both the parts of the assignment we desire to find a line that best fits the data given. The model to be fit is a special case of Equation 1, where $f_1(x) = x$ and $f_2(x) = 1$ and rest of the functions are equal to zero. So the model obtained is

$$y = mx + c \tag{4}$$

The values of unknown line parameters $'m'$ and $'c'$ can be got from Equation 3 by substituting

$$A = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix}, b = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \tag{5}$$

where $(x_1, y_1), (x_2, y_2), - - -(x_n, y_n)$ are the given data points. The code used for finding the model coefficients using normal equation is

```
A=[5 1;6 1;7 1;8 1;9 1];
b =[1;1;2;3;5];
X=inv(transpose(A)*A)*transpose(A)*b
x1=[5,6,7,8,9]; y1=[1,1,2,3,5];
g=X(1,1);
h=X(2,1);
plot(x1,y1,'o')
hold on
Y= g*x1+h
title('plot1:y=a*x+b model')
xlabel('x')
ylabel('y')
```

```
plot(x1,Y)
hold off
```

Similarly for part-2 of the lab we include the point $(8, 14)$ in our code.

```
A=[5 1;6 1;7 1;8 1;8 1;9 1];
b =[1;1;2;3;14;5];
X=inv(transpose(A)*A)*transpose(A)*b
x1=[5,6,7,8,8,9]; y1=[1,1,2,3,14,5];
g=X(1,1);
h=X(2,1);
plot(x1,y1,'o')
hold on
Y= g*x1+h
title('plot1:y=a*x+b model')
xlabel('x')
ylabel('y')
plot(x1,Y)
hold off
```

2.2 Curve fitting

For the third part of the lab we load the provided data-set of 3398 data points. We wish to fit a model that best describes the relationship between the number of bites taken in the meal and the number of kilo-calories per bite consumed. We first plot the data to analyze which model would fit the data best.

We fit a simple linear model to observe the model fit.For the linear model we use the equation 4.We develop matrices A and b from the given data. The code developed for the linear model is

```
y =  peopleallmeals{:,4}./peopleallmeals{:,3}
peopleallmeals{:,5} = y
x=peopleallmeals{:,3}
plot(x,y,'o')
hold on
A=[x x.^0]
b=y;
X=inv(transpose(A)*A)*transpose(A)*b
g=(X(1,1))
h=(X(2,1))
```

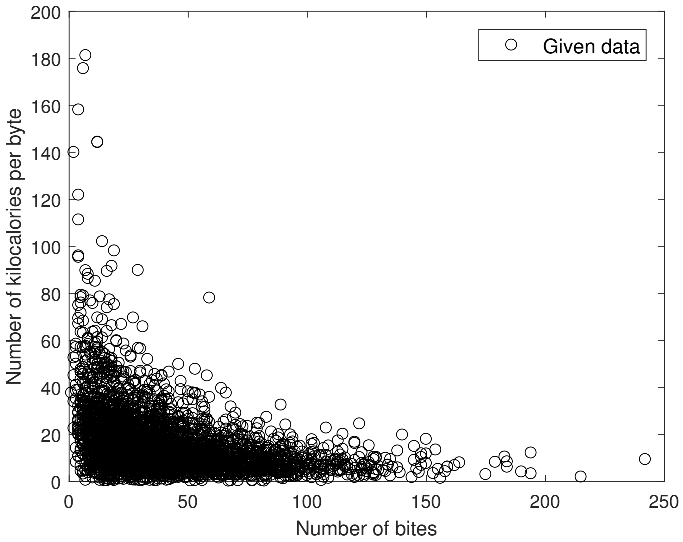

Figure 1: The data points for part-3

```
x1=sort(x)
Y1=(g.*x1)+h
xlabel('number of bites')
ylabel('number of kilocalories per byte')
plot(x1,Y1)
hold off
```

We then try fitting a quadratic linear model to observe the best fit for the data. We use the equation for fitting the model

$$y = gx^2 + hx^1 + i \tag{6}$$

In Equation 6 the model coefficients are g, h, i and we use normal equations to find the value of these coefficients. To build matrices A and b for this model we use the form

$$A = \begin{bmatrix} x_1^2 & x_1 & 1 & \\ x_2^2 & x_2 & 1 & \\ \vdots & \vdots & \vdots & \vdots \\ x_n^2 & x_n & 1 & \end{bmatrix}, b = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \tag{7}$$

The code developed for the above model in equation 6 is

```
y =  peopleallmeals{:,4}./peopleallmeals{:,3}
peopleallmeals{:,5} = y
x=peopleallmeals{:,3}
plot(x,y,'o')
hold on
A=[x.^2 x.^1 x.^0 ]
b=y;
X=inv(transpose(A)*A)*transpose(A)*b
g=(X(1,1));
h=X(2,1);
i=X(3,1);
x1=sort(x);
Y1=(g.*(x1.^2))+(h.*(x1))+i;
xlabel('number of bites')
ylabel('number of kilocalories per byte')
plot(x1,Y1)
hold off
```

As the Figure 1 of the set of data points shows many points descending down near the X axis. We then try to fit a curve of the form

$$xy = a \tag{8}$$

which is a rectangular hyperbola .The slope of this function gets flatter as x increases.For Equation 8 we develop matrices A and b of the form

$$A = \begin{bmatrix} 1/x_1 \\ 1/x_2 \\ \vdots \\ 1/x_n \end{bmatrix}, b = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \tag{9}$$

where x is the number of bites and y is the number of kilo-calories per bite.The unknown parameter a is found using normal equation.

The code developed for the model in equation 8 is

```
y =  peopleallmeals{:,4}./peopleallmeals{:,3}
peopleallmeals{:,5} = y
x=peopleallmeals{:,3}
plot(x,y,'o')
hold on
A=[1./x]
b=y;
```

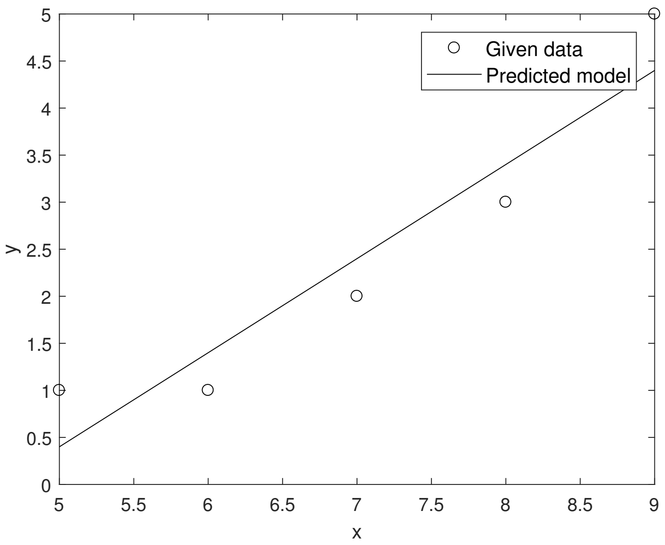

Figure 2: 2D line plot plot for part-1

```
X=inv(transpose(A)*A)*transpose(A)*b
g=(X(1,1))
x1=sort(x)
Y1=(g./x1)
xlabel('number of bites')
ylabel('number of kilocalories per byte')
plot(x1,Y1)
hold off
```

3 Results

For part-1 of the lab we use the simple linear model to fit a 2D line to the given data points.The line equation is given by $y = 1.0x - 4.6$.The code-1 gives us the plot It can be seen in Figure 2 that a linear model has been been fit to the data points in table 1.

Now for part-2 we use the code and fit the same type of model into the data which has a point in addition to part-1.The line equation is given by $y = 1.8154x - 8.6769$.

Figure 3 gives us the output model for part-2. We obtain a linear model with axes x and y plotted from the data points that are given. We see the point $(8, 14)$ acts as an outlier in the data provided.

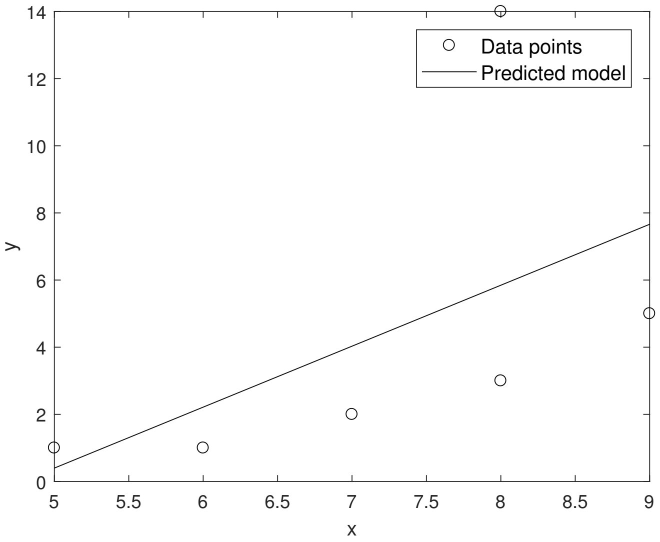

Figure 3: 2D line plot for part-2

For part-3 of the lab we have the meals data-set. We eliminate the outliers in this data-set and we obtain first a simple 2D line model fit. The plot for the 2D line model is given

The figures in part-3 of the lab has been plotted with the x axis as Number of bites and Y axis as Number of kilo-calories per bite. Looking at the figure 4 we can clearly conclude that it is not an appropriate fit for the given data.The chi-squared error which is equal to $7.5983 * 10^5$ is too large.

We notice that the plot of data points shows one bend so we try to fit a quadratic linear model into the data.The plot for the quadratic linear model is given. From the Figure 5 the chi-squared error shows a decrease when compared to 2D line plot but the coefficient of x^2 is small and hence this term is not very significant.

We now come to our final model which is a rectangular hyperbola. The plot for the model is given This is our final model.Figure 6 shows the output plot and we see that the rectangular hyperbola $xy = 242.0896$,where the coefficient a=242.0896 obtained from the code, fits the given points fairly well.We find out that of the three models we observed this model has the best fit.

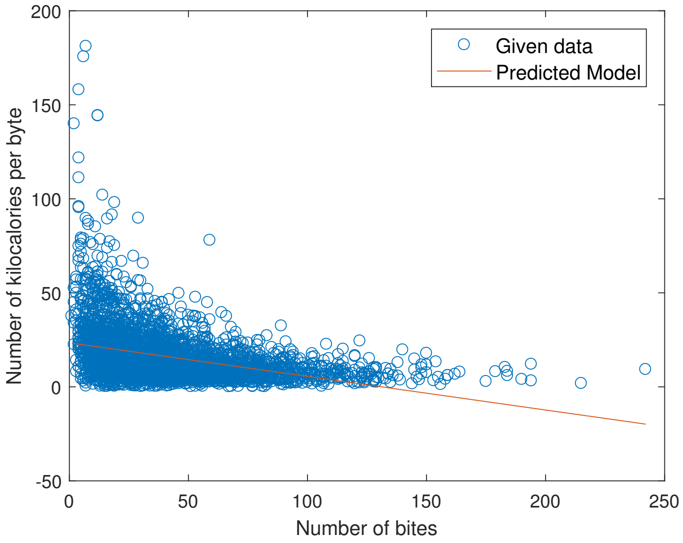

Figure 4: 2D Line model $y = mx + c$ for part-3

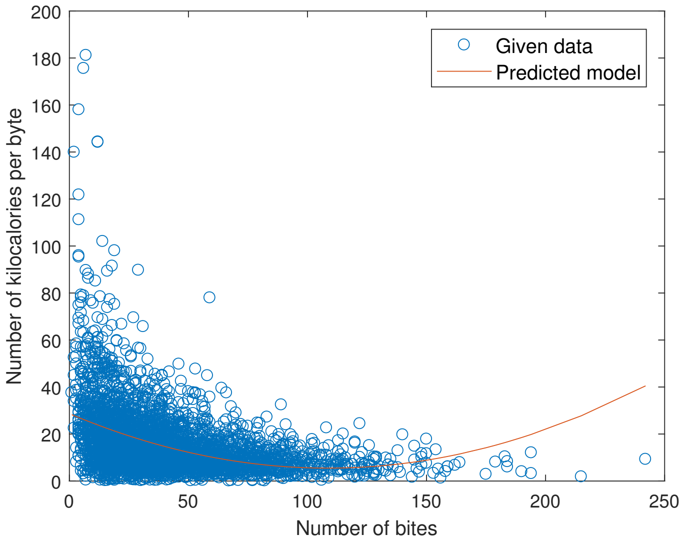

Figure 5: Quadratic linear model for part-3

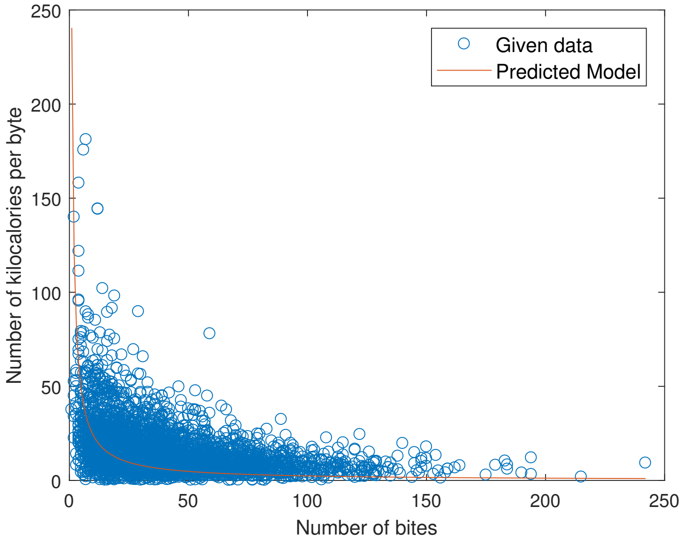

Figure 6: Rectangular Hyperbola model for part-3

4 Conclusion

From the first and second parts of the lab assignments we see that linear regression is sensitive to outliers.Vertical deviations of outliers from the original line of best fit are large,squared deviations are larger.Hence these outliers tend to pull the line towards itself away from the other data points.These outliers may exist due to experimental errors or variability in measurement.It will be beneficial to filter such data from the data set.

In the case of the third part normal equation provides instantaneous solutions for a small data set.If the number of data points exceed 10,000 normal equation which involves inverting of matrices leads to high computational costs.Sometimes A^TA is non-invertible. This can happen when we have redundant values. We then resort to iterative methods like Gradient Descent to find the unknown parameters .Gradient Descent is a first-order iterative optimization algorithm which is used not only in linear regression but all over Machine Learning.