ECE 8540 Analysis of Tracking Systems

Lab 1 - Line and Curve Fitting

Rahil Modi
C14109603

September 1, 2020

1 Introduction

In this lab, we were asked to fit a line through a set of data points for first and second part, for the third part we were asked to fit a appropriate curve through some scattered data points. We were expected to use Normal Equations which is also known as Linear Regression. Generalized line fitting equation is given below:

$$y = a_1 f_1(x) + a_2 f_2(x) + ... + a_M f_M(x) \tag{1}$$

where $a_1 ... a_M$ are the unknowns terms. The terms $f_1(x), f_2(x), ..., f_M(x)$ are called basis functions. They are not linear. The unknowns must all be linear constants.

Using this fitting equation and using the standard matrices for these equations I have designed the best fit model for different data points

In this report I have explained the various steps in Linear Regression Model using MATLAB. The code snippets are attached at respective places and the entire code is attached in the appendix.

2 Methods

In this section, we derive the standard line fitting solution to normal equations and curve fitting solutions to incorporate different data sets.

2.1 Line Fitting for Data Sets 1 and 2

Derivation of the normal equations and their solution using basic Algebraic concepts. Now, the generalized equation of the line is of the form:

$$y = mx + c = a_1 f_1(x) + a_2 f_2(x) \tag{2}$$

where, a and b are constants. On comparing this equation with equation (1), we can see some similarities like,
$a_1 = m; a_2 = c; f_1(x) = x; f_2(x) = 1$. Since, this generalized line equation is of the form $y = a_1 f_1(x) + a_2 f_2(x)$ and is linear in unknowns we consider this as our model to fit the first two given data sets.

Now, for given data points (xi, yi) $\forall i = 1....N$, we define the residual e_i for each point as:

$$\sum_{i=1}^{M} a_i f_i(x_i) \tag{3}$$

The residual term is of the value by which we must minimize the for each point with respect to our model fit. The chi squared error is the distance between the best fitting solution and data points.

$$\chi^2(a_1, a_2, ..., a_M) = \sum_{i=1}^{N} [y_i - \sum_{j=1}^{M} a_j f_j(x_i)]^2 \tag{4}$$

To find the best possible values of the unknowns $a_1 ... a_M$, we need to minimize the chi-squared error matrix by taking the partial derivatives of χ^2 with respect to $a_1 ... a_M$ and equaling them to zero. The simplified equation is:

$$\forall k = 1....M \quad \sum_{i=1}^{N} f_k(x_i) y_i = \sum_{i=1}^{N} \sum_{j=1}^{M} f_k(x_i) f_j(x_i) a_j \tag{5}$$

Now the matrices are formed and they are as follows:

$$A = \begin{bmatrix} f_1(x_1) & f_2(x_1) & ... & f_M(x_1) \\ f_1(x_2) & f_2(x_2) & ... & f_M(x_2) \\ . & . & ... & . \\ . & . & ... & . \\ f_1(x_N) & f_2(x_N) & ... & f_M(x_N) \end{bmatrix} \tag{6}$$

$$x = \begin{bmatrix} a_1 \\ a_2 \\ . \\ . \\ a_M \end{bmatrix} \tag{7}$$

$$b = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_N \end{bmatrix} \tag{8}$$

We can write the equation(5) after rearranging them as:

$$A^T b = A^T A x \tag{9}$$

So equation (9) can be simplified for x as;

$$\boxed{x = (A^T A)^{-1} A^T b} \tag{10}$$

Equation (10) is the solution to the normal equations. Given properly con-structed matrices A, x and b, the solution to any problem in the form of equation (1) can be found using equation (10).But in our case the problem is reduced to the form of line fitting and hence we have $a_1 = m$; $a_2 = c$; $f_1(x) = x$; $f_2(x) = 1$. So we can rewrite the equations as follows:

$$A = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ . & . \\ . & . \\ x_n & 1 \end{bmatrix} \tag{11}$$

$$b = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_n \end{bmatrix} \tag{12}$$

And the matrix of unknowns $x = [m, c]^T$ is given by equation (10) as:

$$x[m \quad c]^T = (A^T A)^{-1} A^T b \tag{13}$$

2.1.1 Part I

For the first part of the lab, we need to fit a 2D line to the following five (x_i, y_i) data points: (5,1);(6,1);(7,2); (8,3); (9,5). Below is the MATLAB code used to find the unknown x and to plot a line fit graph for the given data.

MATLAB CODE:

```
clc ;
close all ;
%Part 1
X = [5 6 7 8 9];
Y = [1 1 2 3 5];
A = [X' ones(5,1)];
b = Y';
x = (inv(A'*A))*A'*b;
i = 4:0.1:16;
y = x(1)*i + x(2);
figure
hold on
plot( i , y, 'r', X, Y,'bo')
axis ([4 15 0 15]);
xlabel('X-axis');
ylabel('Y-axis');
hold off
legend ('Fitted Line' , 'Data Points') ;
```

From the equation in line 8 we get the values for m and c in equation (13) which are $x[m, c]^T$ =[**1.0 , -4.6**].
After substituting these values in $y = mx + c$ we get the best model that will fit through all the data points. The slope of the line can be calculated by m = 1 and y intercept is at -4.6.The graph for given data points and the line fit model are shown in the Results section in figure (2).

2.1.2 Part II

For the second part of the lab, we need to add an extra point to our data set for line fitting. So now we fit a line to 6 data points which are given as, (x_i, y_i) data points: (5,1);(6,1);(7,2); (8,3); (9,5); (8,14). Below is the MATLAB code used to find the unknown x and to plot a line fit graph for the given data with additional data point as compared to part one.

MATLAB CODE:

```
19      %% Part 2
20 -    X = [5 6 7 8 9 8];
21 -    Y = [1 1 2 3 5 14];
22 -    A = [X' ones(6,1)];
23 -    b = Y';
24 -    x = (inv(A'*A))*A'*b;
25 -    i = 4:0.1:16;
26 -    y = x(1)*i + x(2);
27 -    figure
28 -    hold on
29 -    plot( i , y, 'r', X, Y,'bo')
30 -    axis ([4 15 0 15]);
31 -    xlabel('X-axis');
32 -    ylabel('Y-axis');
33 -    hold off
34 -    legend ('Fitted Line' , 'Data Points') ;
```

From the equation in line 24 we get the values for m and c in equation (13) which are x[m, c]T =[1.8 , -8.7].
After substituting these values in y = mx + c we get the best model that will fit through all the data points. The slope of the line can be calculated by m = 1.8 and y intercept is at -8.7.The graph for given data points and the line fit model are shown in the Results section in figure (3).

2.2 Curve Fitting for Data Set of people eating meals

For the third part of the lab, we were given a text file named 83peopleall-meals.txt. This file contained data for 3,398 meals eaten by 83 different people. The third column is the number of bites taken in the meal, and the fourth column is the number of kilocalories consumed. We need to decide and design a model that best fits these data points. So for the first step I plotted out all the points and see how they are plotted on a (X,Y) graph. The points are plotted as scatter with No of bites on X-axis and kilocalories per bite on Y-axis.

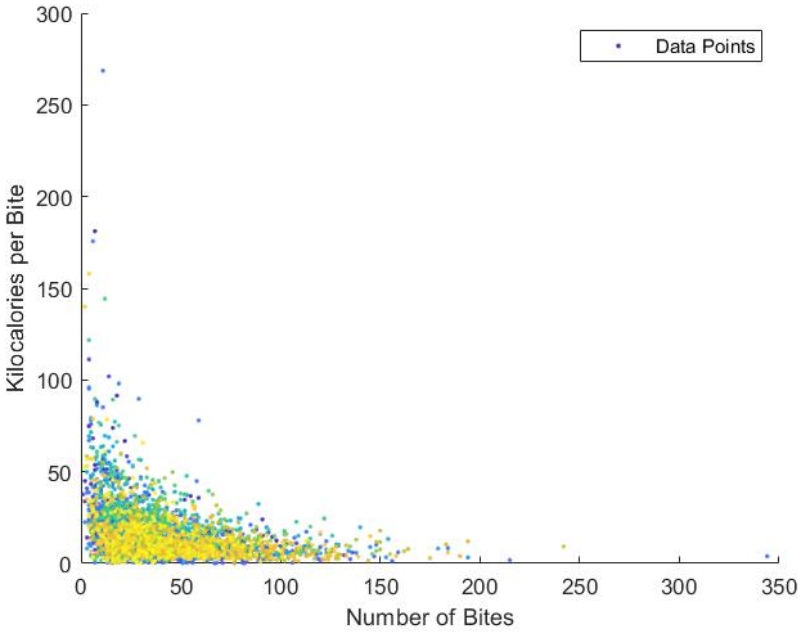


Figure 1: Scatter Plot of Data Points for Part 3

After looking at the points it looks like a logarithmic curve would fit the best among these points. I have kept the color of all the points in different RGB value so that we can better visualize the points that are overlapping. I have also tried exponential decay method but the decay was taking place too soon so, I went forward with logarithmic curve.

$$y = a + b * ln(x) \tag{14}$$

where $a_1 = a; f_1 = 1; a_2 = b; f_2 = ln(x)$.
This model equation is **linear** in unknowns and is of the generalized fitting equation type described in equation (1) and hence can be used for the normal equations.

The Matrix A and b are calculated as:

$$A = \begin{bmatrix} 1 & ln(x)_1 \\ 1 & ln(x)_2 \\ . & . \\ . & . \\ 1 & ln(x)_n \end{bmatrix} \tag{15}$$

$$b = \begin{bmatrix} y_1 \\ y_2 \\ . \\ \\ y_n \end{bmatrix}. \tag{16}$$

Now to calculate the unknowns $x = [a, b]^T$, MATLAB code is given below

```
35      %% Part 3
36 -    g = importdata('83people-all-meals.txt');
37 -    j = [g(:,3) (g(1:3398,4)./g(1:3398,3))];
38 -    figure
39 -    scatter (j(:,1), j(:,2), 25, linspace(1,10,length(j(:,1))), '.');
40 -    xlabel('Number of Bites');
41 -    ylabel('Kilocalories per Bite');
42 -    legend('Data Points');
43 -    X1 = j(:, 1);
44 -    Y1 = j(:, 2);
45 -    A1 = [ones(3398,1) log(X1)] ;
46 -    b1 = Y1;
47 -    h = 1:315;
48 -    x1 = (inv(A1'*A1))*A1'*b1;
49 -    y1 = x1(1) + x1(2)*log(h);          % y = a + b*log(x)
50 -    figure
51 -    hold on
52 -    axis ([0 250 0 100]);
53 -    k = plot(h, y1,'k');
54 -    l = plot(X1, Y1, 'g.');
55 -    uistack(k, 'top');
56 -    xlabel('Number of Bites');
57 -    ylabel('Kilo calories per Bite');
58 -    legend('Fitted Line', 'Data Points');
59 -    hold off
```

From the equation in line 48 we get the values for m and c in equation (13) which are x[m, c]T =[46.5 , -8.7].
After substituting these values in y = m + c* log(x) we get the best model that will fit through all the data points. The slope of the line can be calculated by m = 46.5 and y intercept is at -8.7.The graph for given data points and the line fit model are shown in the Results section in figure (5).

3 Results

3.1 For first data set

The model used to fit the data points given in part 1 is shown in the figure 2. As you can see in the graph itself that this is a good fit as the values of line are not much away from the data points. The points denoted with 'o' are the data points and line is the fitted line through these data points.

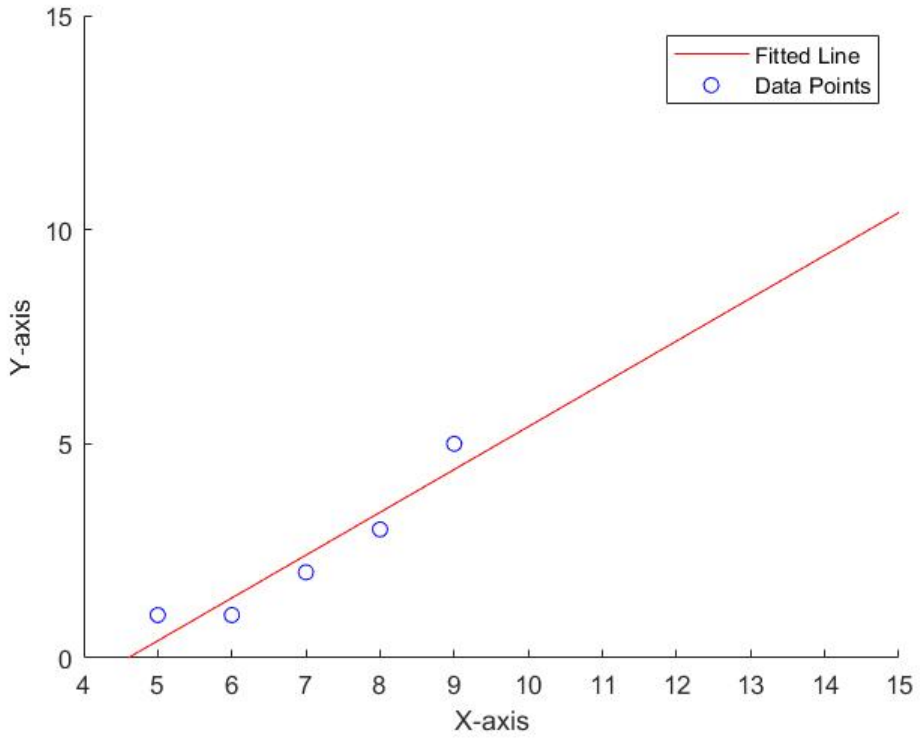


Figure 2: Data Points for Part 1 with Line Fit

3.2 For second data set

As you can clearly see in figure 3 that the now this model is not the best model as due to the new additional point the line which is fitted is no longer proper as the slope of the line has changed drastically and now the variation in the values of the line from the data points is very high and this is not at all desirable. Below you can see the plot with the new data point and below that you will see the comparison between the Part 1 and Part 2.

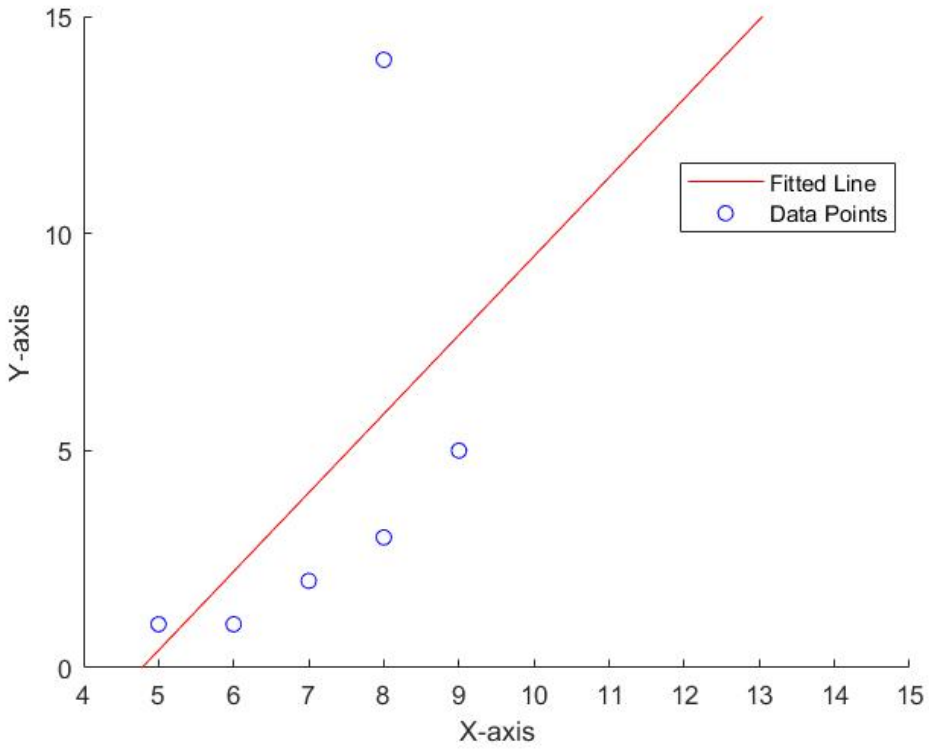
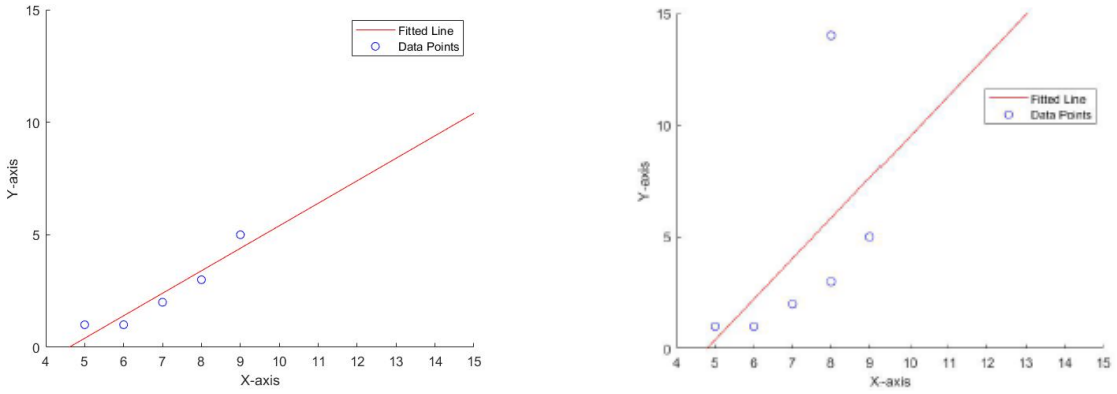


Figure 3: Data Points for Part 2 with Line Fit



(a) Data Points from Part 1

(b) Data Points from Part2

Figure 4: A comparison of line fitting with additional (8,14) point included

3.3 For third data set

It can be seen in the below image that the logarithmic curve is fitting properly through the scatter points and the fit looks proper. As I have tried to fit a linear regression model in a non linear model it cannot be said as the best fit which we can achieve but for representation purposes this is a good fit.

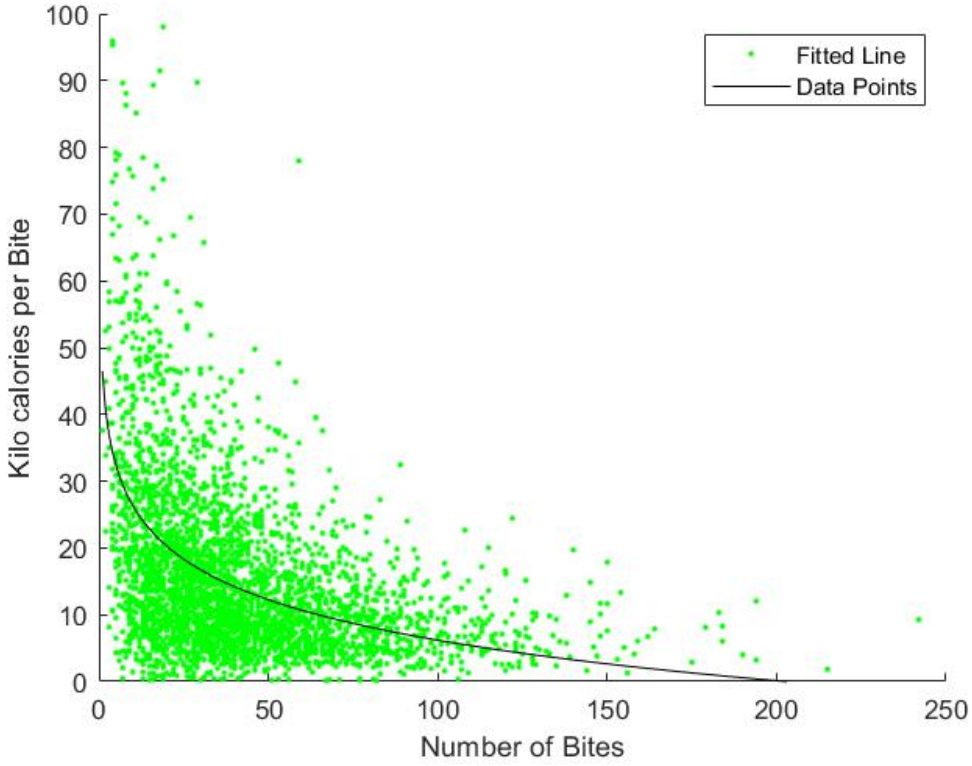


Figure 5: Data Points for Part 3 with Curve Fit

4.1 Appendix

Below is code set for the entire lab implemented on MATLAB.

```
clc ;
close all ;
%Part 1
X = [5 6 7 8 9];
Y = [1 1 2 3 5];
A = [X' ones(5,1)];
b = Y';
x = (inv(A'*A))*A'*b;
i = 4:0.1:16;
y = x(1)*i + x(2);                          % y = mx + c
figure
hold on
plot( i , y, 'r', X, Y,'bo')
axis ([4 15 0 15]);
xlabel('X-axis');
ylabel('Y-axis');
hold off
legend ('Fitted Line' , 'Data Points') ;
%% Part 2
X = [5 6 7 8 9 8];
Y = [1 1 2 3 5 14];
A = [X' ones(6,1)];
b = Y';
x = (inv(A'*A))*A'*b;
i = 4:0.1:16;
y = x(1)*i + x(2);                          % y = mx + c
figure
hold on
plot( i , y, 'r', X, Y,'bo')
axis ([4 15 0 15]);
xlabel('X-axis');
ylabel('Y-axis');
hold off
legend ('Fitted Line' , 'Data Points') ;
%% Part 3
g = importdata('83people-all-meals.txt');
j = [g(:,3) (g(1:3398,4)./g(1:3398,3))];
figure
```

```
39 -    scatter (j(:,1), j(:,2), 25, linspace(1,10,length(j(:,1))), '.');
40 -    xlabel('Number of Bites');
41 -    ylabel('Kilocalories per Bite');
42 -    legend('Data Points');
43 -    X1 = j(:, 1);
44 -    Y1 = j(:, 2);
45 -    A1 = [ones(3398,1) log(X1)] ;
46 -    b1 = Y1;
47 -    h = 1:315;
48 -    x1 = (inv(A1'*A1))*A1'*b1;
49 -    y1 = x1(1) + x1(2)*log(h);          % y = a + b*log(x)
50 -    figure
51 -    hold on
52 -    axis ([0 250 0 100]);
53 -    k = plot(h, y1,'k');
54 -    l = plot(X1, Y1, 'g.');
55 -    uistack(k, 'top');
56 -    xlabel('Number of Bites');
57 -    ylabel('Kilo calories per Bite');
58 -    legend('Fitted Line', 'Data Points');
59 -    hold off
```