ECE 8540

LAB 3 - MODEL FITTING

SEPTEMBER 21, 2021

Tim Driscoll
Clemson University
Department of Electrical and Computer Engineering
tdrisco@clemson.edu

1 Introduction

This lab report concerns the problem of linear model fitting. Model fitting allows us to determine the parameters for equations that best fit an observed data set. These data sets can portray a range of information that are represented numerically for analysis. Model fitting is important because it will provide a means for future predictions given similar data sets or experiments. Previously, methods such as trend observation and estimation have been used to fit linear models to data sets. These methods do not provide the most accurate results and cause for inaccuracies in future predictions. This report focuses on an improved mathematical model, the normal equations. These equations allow for the parameters of linear models to be accurately determined.

2 Methods

The method used for fitting a line to a set of data points can be used on any function containing a linear combination of terms. Each term is composed of an unknown linear constant, a_i, and a basis function, $f_i(x)$, that does not need to be linear. This function can be expanded and observed by the following equation:

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

In this equation there are M unknowns paired with M basis functions. Then based on a the given data set solutions for the unknowns can be solved to produce a best fit. The best possible values of the unknowns are determined by minimizing the chi-squared error. The chi-squared error is defined as the difference between the best fitting model and the data set itself. Matrix notation can be used to simplify this equation, where the following three matrices are defined:

$$A = \begin{bmatrix} f_1(x_1) & f_2(x_1) & \cdots & f_M(x_1) \\ f_1(x_2) & f_2(x_2) & \cdots & f_M(x_2) \\ \vdots & & & \vdots \\ f_1(x_N) & f_2(x_N) & \cdots & f_M(x_N) \end{bmatrix} \tag{2}$$

$$x = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_M \end{bmatrix} \tag{3}$$

$$b = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \tag{4}$$

A is an NxM matrix that is composed of N rows of all the basis functions. Matrix x is size Mx1 and stores all the unknown values $a_1, ..., a_M$. Lastly matrix y is size Nx1 and stores all the outputs from the collected data. Using these three matrices equation 5 can be used to solve for all of the unknowns.

$$x = (A^T A)^{-1} A^T b \tag{5}$$

Equation 5 is referred to as the solution to the normal equations and given matrices A, and b the solutions for the unknowns in x can be solved for.

2.1 Straight Line Model

For this lab two different models were used to represent three different data sets. The first two data sets were given as a set of five and six points respectively. The points were in the form (x, y). Both of these data sets could also be modeled using the equation for a straight line. The equation for a straight line is a version of equation 1 that can be represented by the following:

$$y = a \cdot x + b \cdot 1 = ax + b \tag{6}$$

In this equation x and 1 represent the basis functions of the model and a and b represent the unknowns. The following five data points were from data set one: $(5, 1); (6, 1); (7, 2); (8, 3); (9, 5)$. Data set two was all the same points but included an extra point, $(8, 14)$. From these points the A, x, and b matrices could be constructed for each of the two data sets. For data set one the matrices are as follows:

$$A = \begin{bmatrix} 5 & 1 \\ 6 & 1 \\ 7 & 1 \\ 8 & 1 \\ 9 & 1 \end{bmatrix} \tag{7}$$

$$x = \begin{bmatrix} a \\ b \end{bmatrix} \tag{8}$$

$$b = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 3 \\ 5 \end{bmatrix} \tag{9}$$

These matrices could then be applied to equation 5 using the 2020a version of MATLAB to solve for x. Solving for x would give us the best values for a and b and could be applied to our straight line model. This same process was carried out for data set two using the following updated version of the matrices:

$$A = \begin{bmatrix} 5 & 1 \\ 6 & 1 \\ 7 & 1 \\ 8 & 1 \\ 8 & 1 \\ 9 & 1 \end{bmatrix} \tag{10}$$

$$x = \begin{bmatrix} a \\ b \end{bmatrix} \tag{11}$$

$$b = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 3 \\ 14 \\ 5 \end{bmatrix} \tag{12}$$

Again, using equation 5 the unknowns contained in matrix x were solved for and used in the straight line model.

2.2 Inverse Power Model

The third data set was composed of data for 3,398 meals eaten by 83 different people. Specifically, the relationship between kilocalories per bit versus number of bites taken. Prior to properly fitting a model to this data an appropriate model would need to be decided on. In order to determine which model would best represent the data the data was plotted and a series of unfitted models were compared to the plot. Some of the tested models included the following: $y = a\frac{1}{x} + b$, $y = ae^{-x} + b$, and $y = a\frac{1}{\sqrt{x}} + b$. After, testing variations of each one of these models it was decided that the data was best modeled by the equation 13 below.

$$y = a \cdot \frac{1}{\sqrt{x}} + b \cdot 1 = a\frac{1}{\sqrt{x}} + b \tag{13}$$

In equation 13 the $\frac{1}{\sqrt{x}}$ and the 1 represent the basis functions and a and b are again used to represent the two unknowns. The A, x, and b matrices constructed to fit this model are the following:

$$A = \begin{bmatrix} \frac{1}{\sqrt{x_1}} & 1 \\ \frac{1}{\sqrt{x_2}} & 1 \\ \vdots & \vdots \\ \frac{1}{\sqrt{x_N}} & 1 \end{bmatrix} \tag{14}$$

$$x = \begin{bmatrix} a \\ b \end{bmatrix} \tag{15}$$

$$b = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \tag{16}$$

In matrix 14 $x_1, ..., x_N$ represents the the x data from the set, or number of bites taken for 3398 meals eaten. In matrix 16 $y_1, ..., y_N$ represents the corresponding y data, or the kilocalories per bite for the 3398 meals eaten. The three matrices were constructed in MATLAB and equation 5 was used to solve for the unknowns in matrix x, a and b.

Table 1: Data sets one and two model parameter summary.

Data Set	a	b
1	1.0	-4.6
2	1.8	-8.7

2.3 MATLAB Implementation

Once it was determined how the matrices were setup for each data set equation 5 was implemented in software. The 2020a version of MATLAB on a Windows operating system was used to implement the code. All the matrices were constructed in the code and the solution to the normal equation was used to solve for the unknowns in each model. Once the unknowns were solved for the model could be plotted with the data series to review the results and fit of the model.

3 Results

3.1 Straight Line Models

After using MATLAB to solve for equation 5 using the different set of matrices for data set one and two the unknown parameters were solved for. The unknowns provided us with the values of the slope and y intercept for both of the models. Table 1 summarizes the values obtained for both data sets. It is important to highlight that in table 1 from data set one to data set two there was a change in the slope of 0.8 and a change in intercept 4.1. These changes in slope and intercept were almost a factor of 2 and were the result of the additional point in data set two, $(8, 14)$. The fitted models are plotted side by side in figure 1. This side by side allows for a clear comparison of the effects the outlier point $(8, 14)$. As observed the model for the second data set is rotated upwards and to the left towards the point $(8, 14)$. It can also be observed that this greatly increases the residual for each point in data set two when compared to data set one.

3.2 Inverse Power Model

The same procedure was followed in MATLAB to determine the unknowns for the inverse power model used to fit the data in set three. Table 2 displays the values obtained for the unknowns in the model for data set three. It can be observed that the value of a is relatively large to allow for the curve to get pulled from the origin and model the data. Figure 2 represents the plotted mode using the parameters from table 2 in equation 13. The plot does not include all 3398 data points to better highlight the model in the range of 0-50 bites. In the range from 0-50 bites there was a heavy cluster of data points. The data shows that there is an inverse relationship between the number of bites and the kilocalories per bite. As the number of bites increased the kilocalories would decrease. The model represents this general inverse relationship and fits the entire span of the data.

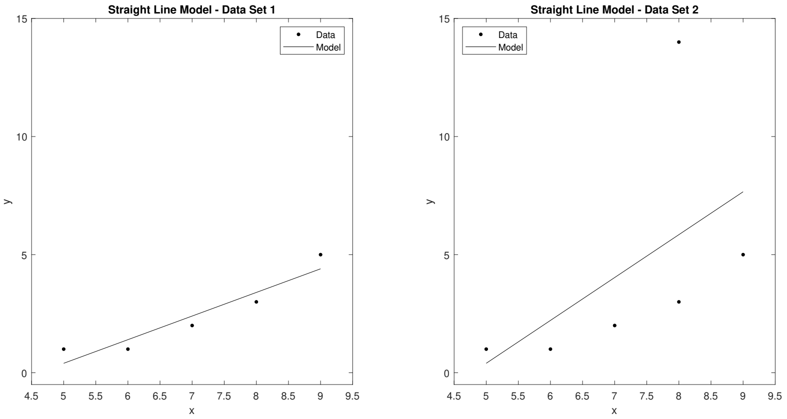

Figure 1: Comparison of model fitting for data sets one and two.

Table 2: Data set three model parameter summary.

Data Set	a	b
3	82.6	0.4

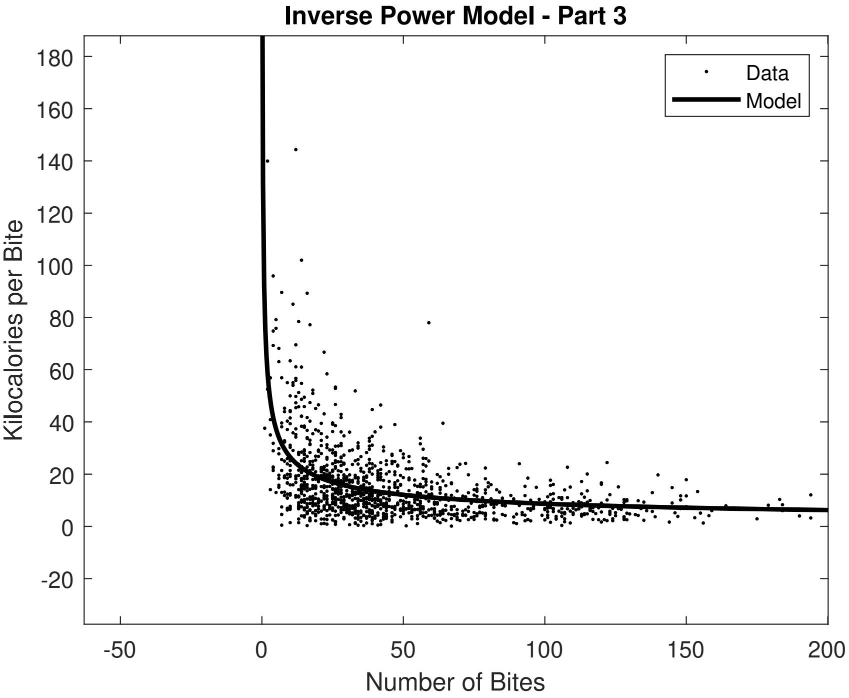

Figure 2: Inverse power model over reduced data set three.

4 Conclusion

Using the solution to the normal equations is a very powerful for determining the parameters to linear models. This tool allows for models to be accurately fitted to data sets. From these models future predictions about the data can be reliably made. Through this lab it was also apparent how outliers in data sets can affect the model. The effects of outliers are especially apparent on data sets with limited entries. When observing figure 1 the effects of a single point are drastic on the accuracy of the model. The increase in the residual of all the points is large when moving from data set one to two. In this lab I also learned that finding the best value for the unknowns is only important if the correct model is chosen for the data set. If a improper model is chosen finding the best parameters will not help the model produce accurate predictions. When fitting a model to data set three it was critical to take time and observe the data so the best model could be selected. Looking at figure 2 I used the apparent inverse relationship between number of bites and kilocalories per bite to choose a model. Ultimately, using the normal equations to fit models to different data sets proved to be very effective in my results. In figure 1 and 2 the models are accurately fit to the data sets and without outliers the residuals of each point is at a minimum.