

Linear Least Squares Curve Fitting of Straight Line Models

Jack Merrin

February 12, 2017

1 Summary

Curve fitting is a useful procedure for testing theoretical models and estimating model parameters and their uncertainties. Curve fitting is carried out by choosing the set of estimated parameters that minimize an error function. The conventional error function for analytic results with calculus is the sum of the squares of residuals. The residuals are the difference between the model prediction and the experimental observation. Weighted fitting takes into account the size of the error bars where measurements with smaller error bars count more heavily towards the minimization. Weighted fits are more robust to the effects of outlying points which generally have larger error bars. The weighted mean is a least squares procedure, fit to $y = A$, which is used to combine different experimental results of the same quantity. This is how CODATA and NIST calculate the accepted fundamental constants of nature from various experiments. Curve fitting is a tedious calculation and is best done with a software package like MATLAB. We illustrate how to fit the various linear models that are discussed with simulated data in MATLAB. One can either put the total data set into the curve fit or average repeated measurements first and fit those.

2 The least squares error formula for curve fitting

Suppose we have a set of measurements (x_i, y_i) and we assume the x_i span some range on the x axis, but there may be repetitions. This is in fact recommended. Always repeat measurements so you can calculate their uncertainties directly from the data. We will not factor in how to use the uncertainties the best way in curve fitting yet, but will consider it later. If we have a model $f = A + Bx$ we define a residual as the difference between the model and the measurement

$$r_i = y_i - A - Bx_i$$

or in general

$$r_i = y_i - f(x_i; A, B, \dots)$$

Now we can start defining some trial error functions. We could choose to minimize the residuals directly.

$$E = \sum_{i=1}^N (y_i - A - Bx_i)$$

The problem with this error function is that some of the errors will be positive and some will be negative. They basically cancel each other out and you can't find the best A and B . If we wanted to make the signs the same then we could try absolute values.

$$E = \sum_{i=1}^N |y_i - A - Bx_i|$$

The problem with this error function is that taking derivatives of absolute value functions is nasty. You can't get many nice results with calculus using this error function. The next simplest modification of the error function is to take

$$E = \sum_{i=1}^N (y_i - A - Bx_i)^2$$

Then we look for the least squares of the residuals. This is the convention that most people use, but it is not totally arbitrary. We will discuss in a more advanced lesson some justification for the least squares method.

3 Fitting $y = A + Bx$

Expanding the 9 terms in E for $f(x_i; A, B) = A + Bx$ gives

$$E = \sum_{i=1}^N (A^2 + 2ABx_i + B^2x_i^2 - 2Ay_i - 2Bx_iy_i + y_i^2)$$

Using the "S" notation we can write E as

$$E = A^2S_1 + 2ABS_x + B^2S_{xx} - 2AS_y - 2BS_{xy} + S_{yy}$$

where $S_1 = N$ and $S_{xy} = \sum_{i=1}^N x_iy_i$ etc. The way we find the minimum of E is to take the partial derivatives with respect to A and B and set them equal to zero.

$$\frac{\partial}{\partial A} \chi^2 = 0 \quad \frac{\partial}{\partial B} \chi^2 = 0$$

This gives two equations which can be solved for A and B .

$$\begin{aligned} AS_1 + BS_x &= S_y \\ AS_x + BS_{xx} &= S_{xy} \end{aligned}$$

This equation can be written as a matrix equation.

$$\begin{pmatrix} S_1 & S_x \\ S_x & S_{xx} \end{pmatrix} \begin{pmatrix} A \\ B \end{pmatrix} = \begin{pmatrix} S_y \\ S_{xy} \end{pmatrix}$$

As long as the determinant of the first matrix on the left is not zero we can solve these linear equations with Cramer's rule. The condition is basically that two of the x_i be different. You can't make any sense of the fit if all the measurements are just at one point. You need two different x_i to define a linear function.

$$A = \frac{\begin{vmatrix} S_y & S_x \\ S_{xy} & S_{xx} \end{vmatrix}}{\begin{vmatrix} S_1 & S_x \\ S_x & S_{xx} \end{vmatrix}} = \frac{S_y S_{xx} - S_x S_{xy}}{S_1 S_{xx} - S_x^2}$$

$$B = \frac{\begin{vmatrix} S_1 & S_y \\ S_x & S_{xy} \end{vmatrix}}{\begin{vmatrix} S_1 & S_x \\ S_x & S_{xx} \end{vmatrix}} = \frac{S_1 S_{xy} - S_x S_y}{S_1 S_{xx} - S_x^2}$$

A common expression is $\Delta = N S_{xx} - S_x^2$.

Now we find the uncertainties in A and B by using the propagation of error formula.

$$\sigma_A^2 = \sum_{i=1}^N \left(\frac{\partial A}{\partial y_i} \right)^2 \sigma_i^2 \quad \sigma_B^2 = \sum_{i=1}^N \left(\frac{\partial B}{\partial y_i} \right)^2 \sigma_i^2$$

If we have not repeated the measurements at each x_i several times then we don't know what the σ_i are. This is a good justification for repeating the measurements. If we used all different x_i then we can just assume that $\sigma_i = \sigma_y$ is the same for all the points. If we use some arguments from statistics we can say

$$E \approx N - 2 \approx \frac{\sum_{i=1}^n (y_i - A - Bx_i)^2}{\sigma_y^2}$$

There are N points, but two degrees of freedom less because we have two fit parameters. The expression in each parenthesis is roughly equal to σ_y^2 . So we can write

$$\sigma_y^2 = \frac{1}{N - 2} \sum_{i=1}^n (y_i - A - Bx_i)^2$$

Now we can do the calculation of the uncertainty. We have

$$\sum_{i=1}^N y_i = y_1 + y_2 + \dots + y_n$$

I define $1_i = \frac{\partial y}{\partial y_i}$. It has the properties that $1_i = 1$ for each i . $\sum_{i=1}^N 1_i = N$, $(1_i)^2 = 1_i$, $\sum_{i=1}^N (1_i)x_i = \sum_{i=1}^N x_i$

$$\left(\frac{\partial A}{\partial y_i}\right) = (1_i S_{xx} - S_x x_i)/\Delta$$

$$\left(\frac{\partial A}{\partial y_i}\right)^2 = (1_i)^2 S_{xx}^2 + S_x^2 x_i^2 - 2(1_i) S_{xx} S_x x_i / \Delta^2$$

$$\sum_{i=1}^N \left(\frac{\partial A}{\partial y_i}\right)^2 = (N S_{xx}^2 + S_x^2 S_{xx} - 2 S_{xx} S_x^2) / \Delta^2$$

$$\sigma_A^2 = (S_{xx}/\Delta) \sigma_y^2$$

$$\sigma_A = \sigma_y \sqrt{\frac{S_{xx}}{\Delta}}$$

A similar calculation for σ_B gives

$$\sigma_B = \sigma_y \sqrt{\frac{N}{\Delta}}$$

If we know each σ_i then it makes sense to just do the weighted calculation which we will describe later in this lesson.

4 Fitting $y = Bx$

For a model like Ohm's law, $V = IR$, we want to fit a straight line forced to pass through the origin, $y = Bx$. The calculus is similar.

$$E = \sum_{i=1}^N (y_i - Bx_i)^2 = S_{wy} - 2BS_{wx} + B^2 S_{xx}$$

Minimizing for B using $\frac{d\chi^2}{dB} = 0$ we find

$$-2S_{wy} + 2\bar{B}S_{wx} = 0$$

$$B = \frac{S_{wy}}{S_{wx}}$$

The propagation of error formula gives the uncertainty of B .

$$\sigma_B^2 = \sum_{i=1}^N \left(\frac{\partial B}{\partial y_i}\right)^2 \sigma_i^2$$

Again if we don't know the uncertainties we can use

$$\begin{aligned}\sigma_y^2 &= \frac{1}{N-1} \sum_{i=1}^N (y_i - Bx_i)^2 \\ \left(\frac{\partial B}{\partial y_i} \right) &= x_i / S_{xx} \\ \left(\frac{\partial B}{\partial y_i} \right)^2 &= x_i^2 / S_{xx}^2 \\ \sigma_B^2 &= \sum_{i=1}^N \left(\frac{\partial B}{\partial y_i} \right)^2 \sigma_y^2 = \frac{\sigma_y^2}{S_{xx}} \\ \sigma_B &= \frac{\sigma_y}{S_{xx}}\end{aligned}$$

5 Fitting $y = A$

The last model we will consider is a horizontal line.

$$E = \sum_{i=1}^N (y_i - A)^2 = S_{yy} - 2AS_y + A^2S_1$$

. So we take the first derivative $E'(A) = 0$ and find

$$0 = -2S_y + 2AS_1$$

or

$$A = S_y / S_1 = S_y / N = \frac{y_1 + y_2 + \dots + y_N}{N} = \bar{y}$$

That is just the arithmetic mean for A . We can find the standard deviation in the mean by the same procedure we used before.

$$\begin{aligned}\left(\frac{\partial A}{\partial y_i} \right) &= (1_i) / N \\ \sum_{i=1}^N \left(\frac{\partial A}{\partial y_i} \right)^2 \sigma_y^2 &= \sigma_y^2 \sum_{i=1}^N (1_i)^2 / N^2 = \sigma_y^2 / N\end{aligned}$$

We use for σ_y where $A = \bar{y}$.

$$\sigma_y^2 = \frac{1}{N-1} \sum_{i=1}^n (y_i - \bar{y})^2$$

So the standard deviation of the mean is given by

$$\bar{\sigma} = \sigma_y / \sqrt{N}$$

6 Weighted fits and using chi squared

It is more useful to do curve fitting when you have repeated the measurements. Then you can know what the different standard deviations are. If the standard deviation is large then you don't want to count those points as much as the ones for which the standard deviation is small. Another method of doing least squares curve fitting is to weight the residuals by $w_i = 1/\sigma_i^2$. This sum is called chi squared, χ^2 .

$$\chi^2 = \sum_{i=1}^N \frac{(y - f(x_i; A, B))^2}{\sigma_i^2}$$

This does not change the math so much that we cannot solve for A and B . You just have to replace each $S_?$ formula with $S_{w?}$. For example, $S_1 \rightarrow S_w$, $S_x \rightarrow S_{wx}$, and so on.

There are slight differences in how you interpret chi squared as a statistic if you use each pair of data points in the equation or first average the points and calculate their standard deviation of the mean. This boils down to using (x_i, y_i, σ_i) or $(x_i, y_i = \bar{y}_i, \sigma_i = \bar{\sigma}_i)$. Often you don't have access to the individual data points if you look at someone else's data, so you have to use the average values and the standard deviations of the mean instead of (y_i, σ_i) . We can then summarize the results from before with the new weighting scheme.

For $y = A + Bx$.

$$A = \frac{S_{wy}S_{wxx} - S_{wx}S_{wxy}}{S_w S_{wxx} - S_{wx}^2} \quad \sigma_A = \sqrt{\frac{S_{wxx}}{S_w S_{wxx} - S_{wx}^2}}$$

$$B = \frac{S_w S_{wxy} - S_{wx} S_{wy}}{S_w S_{wxx} - S_{wx}^2} \quad \sigma_B = \sqrt{\frac{S_w}{S_w S_{wxx} - S_{wx}^2}}$$

For $y = Bx$. We have $B = \frac{S_{wxy}}{S_{wxx}}$. So we can calculate the uncertainty.

$$\left(\frac{\partial B}{\partial y_i}\right) = w_i x_i / S_{wxx}$$

$$\left(\frac{\partial B}{\partial y_i}\right)^2 w_i^{-1} = w_i x_i^2 / S_{wxx}^2$$

$$\sigma_B^2 = \sum_{i=1}^N \left(\frac{\partial B}{\partial y_i}\right)^2 w_i^{-1} = \frac{1}{S_{wxx}}$$

$$\sigma_B = \frac{1}{\sqrt{S_{wxx}}}$$

Probably the most useful weighted formula is for the weighted mean when $y = A = y_w$. This allows you to average together measurements.

$$\chi^2 = \sum_{i=1}^N \frac{(y_i - y_w)^2}{\sigma_i^2} = \sum_{i=1}^N w_i (y_i - y_w)^2$$

$$\chi^2 = S_{wyy} - 2S_{wy}y_w + S_w y_w^2$$

Taking $\frac{d\chi^2}{dy_w} = 0$ we find

$$y_w = \frac{S_{wy}}{S_w} = \frac{w_1 y_1 + w_2 y_2 + \dots + w_N y_N}{w_1 + w_2 + \dots + w_N}$$

This is referred to as the weighted mean. We can find the uncertainty in the weighted mean

$$\sigma_{y_w}^2 = \sum_{i=1}^N (w_i/S_w)^2 w_i^{-1} = 1/S_w$$

$$\sigma_{y_w} = \frac{1}{\sqrt{w_1 + w_2 + \dots + w_N}}$$

In the limit, that all w_i are the same we recover the arithmetic mean and the standard deviation of the mean for the uncertainty.

This formula is quite useful. If you have a series of accurate measurements some of which are precise and others not so much you can combine them to get a more accurate and precise average based on the weighting. Most people are probably already familiar with this in terms of calculating their grades. Say 25 percent for homework, 25 percent for the midterm and 50 percent for the final. Then an average score might look like

$$\frac{87 \times 25 + 95 \times 25 + 88 \times 50}{25 + 25 + 50} = 89.5$$

Pretty close, let's give them an A minus anyways.

7 MATLAB functions for straight line curve fitting

7.1 $y = A + Bx$

Example 7.1. Write a Matlab function that fits a straight line $y = A + Bx$ through a dataset.

Solution 7.1. We can easily compute all the sums in Matlab. Computing the sums by hand would be quite tedious.

```
function [ A sigA B sigB chi2 ] = lineABx( xi, yi, sigi )
%[A sigA B sigB chi2] = lineBx(xi, yi, sigi)
% Least squares fit to a line y = A + Bx
wi = sigi.^(-2);
Sw = sum(wi);
Swx = sum(wi.*xi);
Swxx = sum(wi.*xi.*xi);
```

```

Swy = sum(wi.*yi);
Swxy = sum(wi.*xi.*yi);
Delta = Sw*Swxx - Swx^2;
A = (Swy*Swxx - Swx*Swxy)/Delta;
B = (Sw*Swxy - Swx*Swy)/Delta;
sigA = sqrt(Swxx/Delta);
sigB = sqrt(Sw/Delta);
chi2 = sum( wi.*(yi - A- B.*xi).^2)
end

```

Example 7.2. Write a Matlab function that fits an unweighted straight line $y = A + Bx$ through a dataset.

Solution 7.2. We have

```

function [ A sigA B sigB] = lineABxU( xi, yi )
%[A sigA B sigB] = lineBx(xi, yi)
%Unweighted least squares fit to a line y = A + Bx
S = length(xi);
Sx = sum(xi);
Sxx = sum(xi.*xi);
Sy = sum(yi);
Sxy = sum(xi.*yi);
Delta = S*Sxx - Sx^2;
A = (Sy*Sxx - Sx*Sxy)/Delta;
B = (S*Sxy - Sx*Sy)/Delta;
sigma = sqrt(1/(length(xi)-2)*sum((yi - A - B.*xi).^2));
sigA = sigma*sqrt(Sxx/Delta);
sigB = sigma*sqrt(S/Delta);
end

```

7.2 $y = Bx$

Example 7.3. Write a Matlab function to fit a straight line through the origin

Solution 7.3. We can evaluate the sums easily in Matlab.

```

function [ B sigB ] = lineBx( xi, yi, sigi )
%[B sigB chi2] = lineBx(xi, yi, sigi)
%Weighted least squares fit to a line y = Bx
wi = sigi.^(-2);
Swxy = sum(wi.*xi.*yi);
Swxx = sum(wi.*xi.*xi);
B = Swxy/Swxx;

```



```

sigB = 1/sqrt(Swxx);
chi2 = sum( wi.*(yi - B.*xi).^2)
end

```

Example 7.4. Write a Matlab function to fit an unweighted straight line through the origin

Solution 7.4. The code for the unweighted fit is as follows

```

function [ BU sigBU ] = lineBxU( xi, yi )
%[B sigB] = lineBx(xi, yi )
%Weighted least squared fit to a line y = Bx
Sxy = sum(xi.*yi);
Sxx = sum(xi.*xi);
BU = Sxy/Sxx;
variance = 1/length(yi)* sum( (yi - BU).^2 );
sigma = sqrt(variance);
sigBU = sigma/sqrt(Sxx);
end

```

7.3 $y = A$

Example 7.5. Write a Matlab function to calculate the weighted mean and its uncertainty.

Solution 7.5. Implementing the theory we have

```

function [ A sigA chi2] = lineA( xi, yi, sigi )
%[A sigA chi2] = lineA(xi, yi, sigi)
%Least squares fit to a constant y = A
wi = sigi.^(-2);
Sw = sum(wi);
Swy = sum(wi.*yi);
A = Swy/Sw;
sigA = 1/sqrt(Sw);
chi2 = sum( wi.*(yi - A).^2 );
end

```

8 MATLAB curve fitting with simulated data

Example 8.1. Illustrate a weighted linear fit to $y = A+Bx$. For the purpose of simulating, data we will assume a linear model plus random gaussian noise. The gaussian distribution

noise can be generated with the command $\text{normrnd}(\mu, \sigma) = \text{normrnd}(0, 3)$. We will take $N_i = 5$ measurements at each point for $x = 1, 2, 3, \dots, 10$. With the model $y = 10 + 3x$. The goal of this simulation is to compare fitting the measurements and their standard deviations with the means at each x and their standard deviation of the mean. It will turn out fitting either way gives identical estimates.

```

Solution 8.1. clear; figure(1); clf;
format long;

A = 10;
B = 3;
mu = 0;
sig = 3;
Ni = 5;

x = 1:1:10
allx = [];
ally = [];
allysig = [];
avgy = [];
for i = 1:length(x)
    xtemp = [];
    temp = [];
    tempsig = [];
    for j = 1:Ni
        err = normrnd(mu, sig);
        temp(j) = A + B*x(i) + err;
        xtemp = [xtemp x(i)];
    end
    sigy1 = std(temp);
    allx = [allx xtemp];
    ally = [ally temp];
    allysig = [allysig ones(1, Ni)*sigy1 ];
    avgy(i) = [mean(temp)];
    avgysig(i) = [sigy1/sqrt(Ni)];
end

[A1 sigA1 B1 sigB1 chi1] = lineABx(allx, ally, allysig);
[A2 sigA2 B2 sigB2 chi2] = lineABx(x, avgy, avgysig);

figure(1);
plot(allx, ally, 'k', 'markersize', 20);
hold on;
plot(x, avgy, 'xr', 'markersize', 20);
axis( [ 0 11 0 1.1*max(ally)]);

```

```

xlabel('x','fontsize',20);
ylabel('y','fontsize',20);
x1 = [0 11]
y1 = [ A1 (A1 +B1*11)];
plot(x1,y1);

```

You can check the fit parameters $A1$, $A2$, $B1$, and $B2$ to see that they are numerically the same. Using the measurements with the standard deviations is equivalent to using the means with the standard deviations of the mean. Whichever you choose to display is up to you. In physics usually one shows the standard error of the mean because the samples are assumed to be identical and there are often a surplus of points. It might make more sense in biology to show the actual variation of the samples where they can be different and the number of measurements is less. You can modify the code by increasing N_i to show that the parameter estimates become more accurate and precise as N_i is increased.

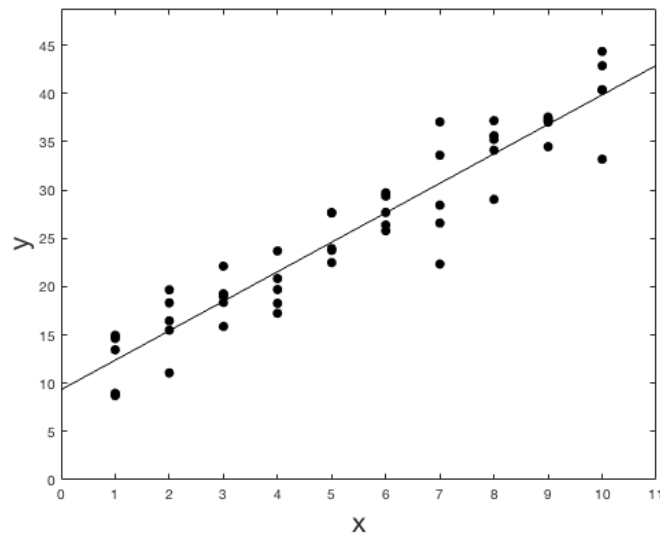


Figure 1: Model $y = 10 + 3x + \text{normrnd}(0,3)$, with five samples using standard deviation and measurements

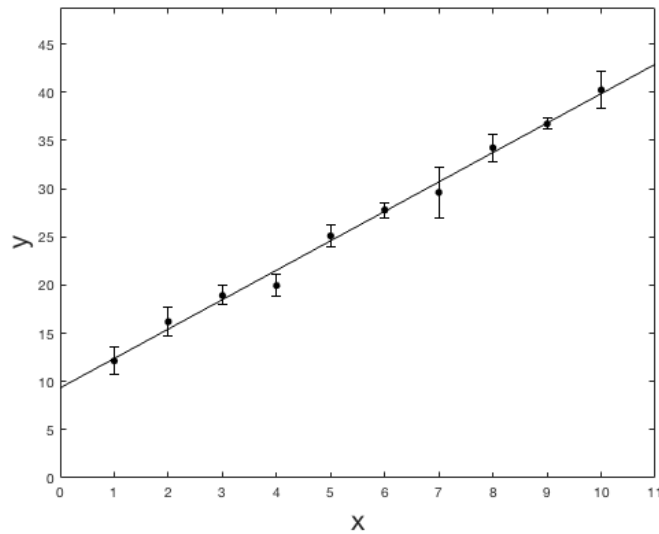


Figure 2: Model $y = 10 + 3x + \text{normrnd}(0,3)$, with five samples using means and standard deviation of the means. We obtain the same fit as before

Example 8.2. *Show the effect of an outlying point when a linear fit is weighted or unweighted. We will use simulated data like before but where the last point is an outlier. We will show the effect in an unweighted fit and a weighted fit which is more robust and fits the data better.*

Solution 8.2. `clear;`

`format long;`

`x = 1:10`

`y = [13.69 14.9 17.1 24.4 25.4 26.6 29.48 32.0 37.36 48.0]`

`sigy = [0.58 1.0 1.0 1.2 1.6 1.4 0.86 1.8 0.84 8.1]`

`figure(1);clf;`

`errorbar(x,y,sigy,'.k','markersize',10);`

`hold on;`

`axis([0 11 0 1.3*max(y)]);`

`xlabel('x','fontsize',20);`

`ylabel('y','fontsize',20);`

`title('Effect of an outlying point on unweighted fit','fontsize',18);`

`[A1 sigA1 B1 sigB1 chi1] = lineABx(x,y, sigy);`

`[A2 sigA2 B2 sigB2] = lineABxU(x, y);`

`x1 = [0 11]`

`y1 = [A1 (A1 +B1*11)];`

`plot(x1,y1,'k-');`

`x2 = [0 11]`

```
y2 = [ A2 (A2 +B2*11)];  
plot(x2,y2,'k-.');
```

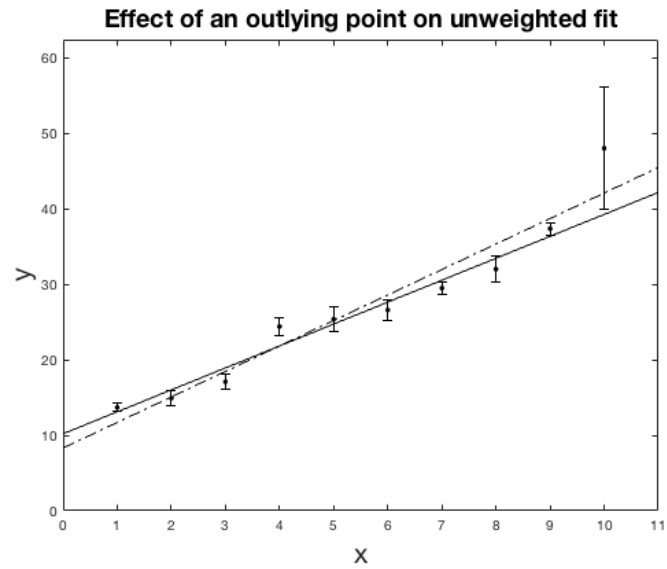


Figure 3: Model $y = 10 + 3x + \text{normrnd}(0,3)$, with five samples, standard deviation of the mean used and an outlying point. The outlying point shifts the slope of the unweighted fit, but does not effect the weighted fit too much.