My Favorite Least Squares Fitting Tricks (rough draft, not for distribution)

Charles Whitmer

August 25, 2020

Abstract

Over the years I have performed thousands of least squares fits, and written software in several languages to calculate those fits. I've got a simple way of looking at things that some people might find useful.

1 Linear Least Squares

Suppose we have a set of independent variables x_i and related observations y_i. The first fit we will look at will try to find a linear relationship between x and y, specifically, the best a and b for the model $y = ax + b$. To find these we will minimize the sum of squares S^2 over all possibilities for a and b:

$$S^2 = \sum_{i=1}^{N} (ax_i + b - y_i)^2 \tag{1}$$

The solution is found easily by taking derivatives of S^2 and setting them to 0:

$$\begin{aligned} \frac{\partial}{\partial a} S^2 &= 2 \sum_{i=1}^{N} x_i (ax_i + b - y_i) = 0 \\ \frac{\partial}{\partial b} S^2 &= 2 \sum_{i=1}^{N} (ax_i + b - y_i) = 0 \end{aligned} \tag{2}$$

We can rearrange this into two linear equations in two variables.

$$\begin{aligned} \sum_{i=1}^{N} ax_i^2 + bx_i &= \sum_{i=1}^{N} x_i y_i \\ \sum_{i=1}^{N} ax_i + b &= \sum_{i=1}^{N} y_i \end{aligned} \tag{3}$$

Since all the x_i and y_i are known, this consists only of a, b, and a bunch of constants, and so it is a simple problem in linear algebra.

Actually, the name "Linear Least Squares" does not refer to the linear model that we are fitting, instead it refers to the fact that the problem reduces to a set of linear equations in our unknown parameters, as in (3). We will expand on this later.

Also, especially looking forward to more complex models, I have found a particular notation useful in solving Linear Least Squares problems. Suppose I define "twiddle" parameters as follows:

$$\begin{aligned}
\widetilde{x} &= \sum_{i=1}^{N} x_i \\
\widetilde{xx} &= \sum_{i=1}^{N} x_i^2 \\
\widetilde{y} &= \sum_{i=1}^{N} y_i \\
\widetilde{xy} &= \sum_{i=1}^{N} x_i y_i \\
\widetilde{1} &= \sum_{i=1}^{N} 1 = N
\end{aligned} \tag{4}$$

It is important to keep in mind that $\widetilde{1}$ is not 1, but N in this case.

We can now rewrite (3) in matrix form:

$$\begin{pmatrix} \widetilde{xx} & \widetilde{x} \\ \widetilde{x} & \widetilde{1} \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \widetilde{xy} \\ \widetilde{y} \end{pmatrix} \tag{5}$$

And so the solution is:

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \widetilde{xx} & \widetilde{x} \\ \widetilde{x} & \widetilde{1} \end{pmatrix}^{-1} \begin{pmatrix} \widetilde{xy} \\ \widetilde{y} \end{pmatrix} \tag{6}$$

2 A Better Way to do Things

The above can be generalized to more complicated models, but first it is important to note that our simple definition of S^2 is not very useful. All of the data points in our fit have been given the same weight, when in real life that would usually not be the case. In most experiments, some data points are more accurate than others, and that should be reflected in our best fit.

The solution is to modify equation (1) to instead read:

$$\chi^2 = \sum_{i=1}^{N} \frac{(ax_i + b - y_i)^2}{\sigma_i^2} \tag{7}$$

where σ_i is the standard deviation (i.e. "error") of the measurement y_i.

Obviously, when a y_i value is more accurately determined, the σ_i is smaller and then the σ_i^2 in the denominator gives a larger contribution to the sum. In order to minimize χ^2 we are going to have to fit the points with small measurement errors more closely.

Note also that σ_i carries the same units as y_i, and that the sum will therefore be dimensionless. It turns out that if the model is a good one, and the σ_i values are reasonable, then the resulting minimal value of χ^2 will follow what is know as the chi-square distribution.

Each time you run your experiment and get new y_i values, your best fit will change slightly and you will get a new χ^2 value. You could repeat this process many times and histogram the χ^2 values. In this sense it is a distribution.

3 Degrees of Freedom

The chi-square distribution is well known, but is actually a set of distributions enumerated by k, the number of "degrees of freedom".

Suppose we perform an experiment and measure 10 values of y_i. The set of results can vary in a 10-dimensional space, and we would call that 10 degrees of freedom. When we fit this data to a linear model, i.e. one with two parameters, we would hope to take away two degrees of freedom from the residuals (the differences between the data and the best fit) and we would say that 8 degrees of freedom would be left over. The best fist found from minimizing equation (7) would give a χ^2 value that should follow a chi-square distribution with 8 degrees of freedom. We can look up the value in a table of the chi-square distribution for 8 degrees of freedom and read out how probable that result is. (At least in the old days I would do that. A good scientific calculator these days should have the function built in. On the HP-Prime you would use CHISQUARE_CDF; on the TI-84 the function is χ^2cdf.)

It happens that the mean value of the chi-square distribution for k degrees of freedom is exactly k. So when we do our experiment measuring our 10 data points and then fit them to a linear model, we would hope that the minimum χ^2 would be about 8. If it is far off from that, then either the model or our error estimates are bad.

To try to develop an intuition about the degrees of freedom, suppose we did an experiment taking two data points, and then fit a linear model to it. Well, we can perfectly fit a line to any two data points, so our χ^2 value would be zero. This is a poor experiment because we have no degrees of freedom left to judge how good the fit is. If we had taken 3 data points to test our linear model, then we would have one degree of freedom to test, and we would hope to get a χ^2 value around 1 for a successful fit.

4 A Linear Model Using χ^2

Let's revisit the fit of the linear model that we did in section 1. If you work through the math you will see that we simply need to redefine the twiddle variables like this:

$$
\begin{aligned}
\widetilde{x} &= \sum_{i=1}^{N} \frac{x_i}{\sigma_i^2} \\
\widetilde{xx} &= \sum_{i=1}^{N} \frac{x_i^2}{\sigma_i^2} \\
\widetilde{y} &= \sum_{i=1}^{N} \frac{y_i}{\sigma_i^2} \\
\widetilde{xy} &= \sum_{i=1}^{N} \frac{x_i y_i}{\sigma_i^2} \\
\widetilde{1} &= \sum_{i=1}^{N} \frac{1}{\sigma_i^2}
\end{aligned}
\tag{8}
$$

And now note that $\widetilde{1}$ is not even N, but something complicated, so beware when manipulating it.

Given the new twiddle definitions, the solution looks identical:

$$
\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \widetilde{xx} & \widetilde{x} \\ \widetilde{x} & \widetilde{1} \end{pmatrix}^{-1} \begin{pmatrix} \widetilde{xy} \\ \widetilde{y} \end{pmatrix} \tag{9}
$$

However, what we have now is a better result, and it required no more complexity.

5 More Interesting Models

What happens when we get more complicated than a linear model? Let's try a cubic model:

$$
y = ax^3 + bx^2 + cx + d \tag{10}
$$

You can work through the math, which is again easy, but here is the solution expressed in twiddle notation:

$$
\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} \widetilde{x^6} & \widetilde{x^5} & \widetilde{x^4} & \widetilde{x^3} \\ \widetilde{x^5} & \widetilde{x^4} & \widetilde{x^3} & \widetilde{x^2} \\ \widetilde{x^4} & \widetilde{x^3} & \widetilde{x^2} & \widetilde{x} \\ \widetilde{x^3} & \widetilde{x^2} & \widetilde{x} & \widetilde{1} \end{pmatrix}^{-1} \begin{pmatrix} \widetilde{x^3 y} \\ \widetilde{x^2 y} \\ \widetilde{xy} \\ \widetilde{y} \end{pmatrix} \tag{11}
$$

There is an obvious pattern to this, and it holds to all orders.

Once you have used this a few times, it becomes obvious and you can write it from memory. More than once I have written a polynomial fit routine in some computer language and not even needed notes. You simply sum up the twiddle parameters in a loop, arrange them in a matrix, and then call your favorite matrix inverter or linear equation solver.

Models that look even more complex can be solved, i.e. fit, in this way. For example, last year I needed a fit to this model:

$$y = a\ sin(\omega t) + b\ cos(\omega t) + c\ t + d \tag{12}$$

I knew the value of ω already, so this model is just linear in four unknowns, and therefore I could use linear least squares. By defining more twiddle variables, the solution can be quickly written:

$$\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} \widetilde{sin\ sin} & \widetilde{sin\ cos} & \widetilde{t\ sin} & \widetilde{sin} \\ \widetilde{sin\ cos} & \widetilde{cos\ cos} & \widetilde{t\ cos} & \widetilde{cos} \\ \widetilde{t\ sin} & \widetilde{t\ cos} & \widetilde{t^2} & \widetilde{t} \\ \widetilde{sin} & \widetilde{cos} & \widetilde{t} & \widetilde{1} \end{pmatrix}^{-1} \begin{pmatrix} \widetilde{y\ sin} \\ \widetilde{y\ cos} \\ \widetilde{y\ t} \\ \widetilde{y} \end{pmatrix} \tag{13}$$

I can be as casual as I like with the notation, as long as it comes back to a rigorous definition:

$$\begin{aligned} \widetilde{sin\ sin} &= \sum_{i=1}^{N} \frac{sin^2(\omega t_i)}{\sigma_i^2} \\ \widetilde{sin\ cos} &= \sum_{i=1}^{N} \frac{sin(\omega t_i)cos(\omega t_i)}{\sigma_i^2} \\ \widetilde{t\ sin} &= \sum_{i=1}^{N} \frac{t_i sin(\omega t_i)}{\sigma_i^2} \\ \widetilde{y\ sin} &= \sum_{i=1}^{N} \frac{y_i sin(\omega t_i)}{\sigma_i^2} \end{aligned} \tag{14}$$

and so on.

Take a moment to look at the pattern of the matrix in equation (13). The parameters a, b, c, d multiply the functions $sin(\omega t)$, $cos(\omega t)$ t, and 1 in the model (12). Since a multiplies the $sin(\omega t)$, the first line of the matrix multiplies the *sin* by each of the four functions. In the second line, the *cos* multiplies each of the functions, and so on. This is obvious if you think of the derivatives that lead to this.

Also note that the matrix is always symmetric. This is a good thing to check.

6 Bonus: The Covariance of the Fit

I call this a bonus, because while any statistics course will teach you how to solve linear least squares problems, they almost never cover this part.

Suppose you have fit some data to a cubic according to equation (11). Now suppose somebody asks you what the error bars are on your resultant parameter a. How would you know?

One way to get the error on a is through the usual method of error propagation. Specifically, if $a = f(y_1, y_2, ..., y_N)$ then

$$\sigma_a^2 = \sum_{i=1}^{N} \sigma_i^2 (\frac{\partial f}{\partial y_i})^2 \tag{15}$$

as long as the errors on each y_i are independent.

We have such a dependence of a on the set of y_i, and so we can do the calculation. But wait! If we have already calculated the matrix inverse in equation (11), then just call it $\boldsymbol{C}$,

$$\boldsymbol{C} = \begin{pmatrix} \widetilde{x^6} & \widetilde{x^5} & \widetilde{x^4} & \widetilde{x^3} \\ \widetilde{x^5} & \widetilde{x^4} & \widetilde{x^3} & \widetilde{x^2} \\ \widetilde{x^4} & \widetilde{x^3} & \widetilde{x^2} & \widetilde{x} \\ \widetilde{x^3} & \widetilde{x^2} & \widetilde{x} & \widetilde{1} \end{pmatrix}^{-1} \tag{16}$$

and this $\boldsymbol{C}$ just happens to be the covariance matrix of the best fit parameters. (The proof involves only a bit of linear algebra.)

And now it is easy to pick out:

$$\begin{aligned} \sigma_a &= \sqrt{\boldsymbol{C}_{11}} \\ \sigma_b &= \sqrt{\boldsymbol{C}_{22}} \end{aligned} \tag{17}$$

and so on.

But having the covariance matrix is even more powerful than that. Suppose that, having done the fit, you need to report some function $g(a, b, c, d)$. What is the error on that calculated value? The answer comes from the covariance matrix:

$$\sigma_g^2 = \begin{pmatrix} \frac{\partial g}{\partial a} & \frac{\partial g}{\partial b} & \frac{\partial g}{\partial c} & \frac{\partial g}{\partial d} \end{pmatrix} \boldsymbol{C} \begin{pmatrix} \frac{\partial g}{\partial a} \\ \frac{\partial g}{\partial b} \\ \frac{\partial g}{\partial c} \\ \frac{\partial g}{\partial d} \end{pmatrix} \tag{18}$$

To make this more concrete, suppose we have fit our data to the linear model $y = ax + b$ and what is important to us is not simply a or b, but the x-intercept. The x-intercept is obviously:

$$x_0 = -\frac{b}{a} \tag{19}$$

Assume we have calculated:

$$\boldsymbol{C}_{linear} = \begin{pmatrix} \widetilde{xx} & \widetilde{x} \\ \widetilde{x} & \widetilde{1} \end{pmatrix}^{-1} \tag{20}$$

then we would have:

$$\sigma_{x_0}^2 = \begin{pmatrix} \frac{b}{a^2} & -\frac{1}{a} \end{pmatrix} \boldsymbol{C}_{linear} \begin{pmatrix} \frac{b}{a^2} \\ -\frac{1}{a} \end{pmatrix} \tag{21}$$

where the two vectors dotted into the matrix are the appropriate partial derivatives.

7 Note: Experiment Design

Go back and take a look at the solutions for three different problems: equations (9), (11), and (13). Note that y only appears in the vector on the right in all cases. The matrix, which determines the covariance of the fit parameters, depends only on the form of the model, the locations of the independent variables x_i or t_i, and how accurately you will be able to measure, i.e. σ_i.

This means that we can understand the accuracy of an experiment before we have even performed it! This is powerful, and permits you to strategically choose where, x_i, or when, t_i, you make your observations, in order to get the best results.

8 A Generalized χ^2

The usual definition of χ^2 for fitting some model $y = f(x)$ is this:

$$\chi^2 = \sum_{i=1}^{N} \frac{(f(x_i) - y_i)^2}{\sigma_i^2} \tag{22}$$

which makes the assumption that the measurement errors on y_i are independent. What if they are not?

Suppose we define $\boldsymbol{M}$ as the diagonal matrix:

$$\boldsymbol{M} = \begin{pmatrix} \frac{1}{\sigma_1^2} & & \\ & \ddots & \\ & & \frac{1}{\sigma_N^2} \end{pmatrix} \tag{23}$$

then we could rewrite the definition of χ^2 with the weighting matrix $\boldsymbol{M}$ as:

$$\chi^2 = \sum_{i=1}^{N} \sum_{j=1}^{N} \boldsymbol{M}_{ij} (f(x_i) - y_i)(f(x_j) - y_j) \tag{24}$$

Another way to think of $\boldsymbol{M}$ is as the inverse of the covariance matrix of all the y_i:

$$\boldsymbol{M} = \begin{pmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_N^2 \end{pmatrix}^{-1} \tag{25}$$

which is diagonal in this case because all the y_i are assumed to be independent.

And that is the key. When the y_i errors are correlated, we should define χ^2 by equation (24), with $\boldsymbol{M}$ being the inverse of the covariance matrix of the y_i.

9 Comparing Fluence Histograms

Consider the histograms in figure 1, which show neutron fluence as computed by two different programs. We would like to know if the histograms match as well as they should, given the

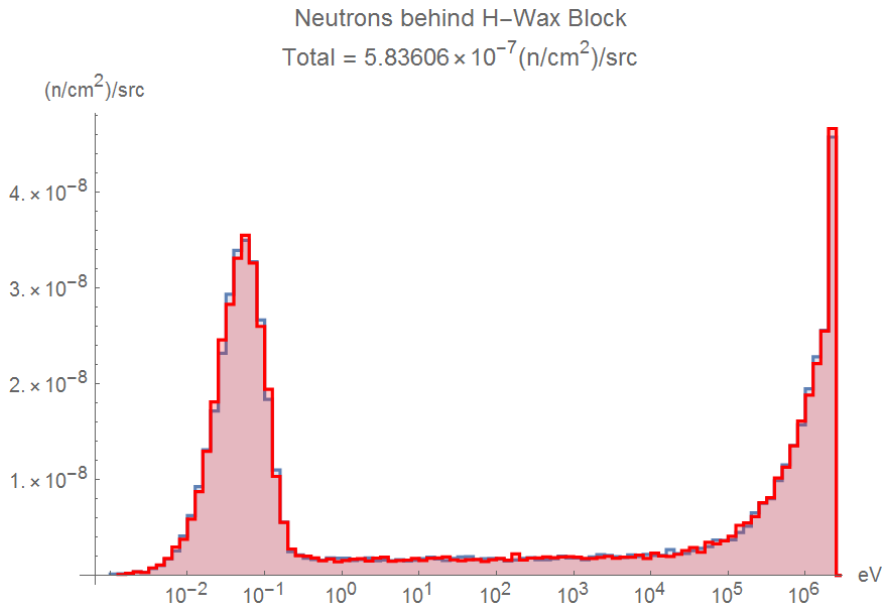

Figure 1: Two fluence histograms that we would like to compare.

statistics, or is there some difference between them, as might be seen in the lower (thermal) peak?

We would take the results of the program that we declare to be the "gold standard", in this case MCNP, and use it as our model, and then ask what is the χ^2 when compared to the results of another program.

Our measurements in this case are the fluence values in the histogram bins. Are they independent or correlated? The answer is that they are highly correlated. A single neutron path will contribute fluence to many bins during the simulation. We have seen in detail how neutrons scatter, and we know that when the neutron has low energy it is very likely to scatter into a nearby energy bin. And so counts and fluence in one bin make it likely that neighboring bins also have counts.

Therefore, we need to use equation (24) to compute the χ^2. The number of degrees of freedom is the number of bins, and we would use the chi^2 distribution to determine how well the histograms match.

In order to do this, we need to know the covariance matrix for the measurements in each bin. Unfortunately, this is a matter of current software development as we cannot calculate the covariance from first principles, and so need to collect it during simulation.

10 Comparing Distribution Histograms

We also have another kind of histogram to compare that is fundamentally different. In figure 2 for example, we have scattered neutrons off of a target and then count them in bins according to the angle of the scattering. This has the nice property that each neutron can contribute only once to the histogram. Each neutron is independent of the next, so does this mean that the bin counts are independent? Interestingly, they are not.

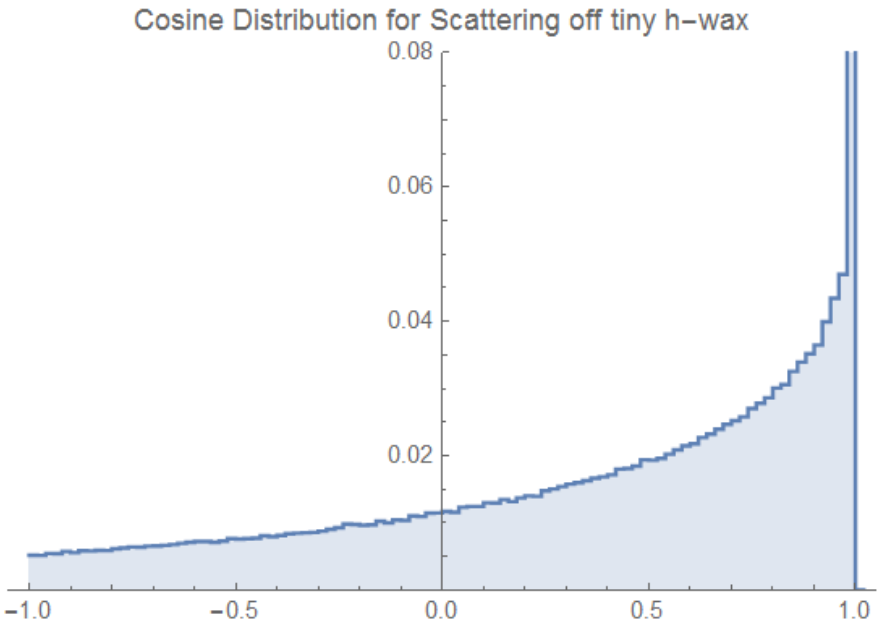

Figure 2: A histogram showing a probability distribution.

When a neutron is counted in one bin, it is *not* counted in another! This actually means that there is a negative correlation between bin counts. It may be easier to understand when you think of it in terms of a random fluctuation away from the average. Suppose one bin, by pure luck, happens to get more hits than its fair share. Because the total number of particles is fixed, some other bin or bins must be lower than expected.

This differs in a fortunate way from the fluence histogram in that we can calculate the covariance matrix for the bins from first principles. We will not need to collect any extra data to compute the χ^2 for the comparison of two of these histograms.