PHY307F/407F - Computational Physics
Background Material for Expt. 1 - Fitting Techniques

David Harrison

INTRODUCTION

The use of computers to fit experimental data is probably the application that is used more than any other in computational physics. A whole course could easily be designed that would cover this topic alone.

This document and the accompanying experiment on fitting techniques only ‘scratch the surface’. One source for further information is the *Mathematica* notebooks that accompany the *Experimental Data Analyst (EDA)* package. These notebooks are available on-line in the `math‘EDA‘Notebooks` directory; David Harrison also has a hardcopy version. Of particular interest in the context of fitting of data are Chapters 4 through 7.

Usually when one is fitting data, a theoretical model is present and the analyst is attempting to determine either how well the data matches the model or what the values are for the parameters of the model. One of the simplest and most common of such cases is when one is fitting data to a straight line and attempting to determine the slope and intercept of the line.

There are also cases where one is fitting data in the absence of a theoretical model. This is often called *nonparametric fitting*, and will not be discussed here.

We concentrate below on fitting using least-squares regression. Although least-squares is the most commonly used algorithm it is not without some difficulties, particularly when the data is noisy.

As will be seen, a crucial distinction is between fitting to a **linear model** versus a **nonlinear model**.

These notes are organised as follows:

I. Linear fits.

   A. Least-squares regression.

   B. Fitting to data with experimental errors.

   C. Evaluating the goodness of a fit.

   D. Reweighting Data That Have No Explicit Errors

II. Nonlinear fits.

Note that the usual listing of the code you will be using in the experiment is missing from these notes. This is because their size is large (nearly 1600 lines for the linear fitter alone, not including subsidiary routines). Students who wish to take a look at the code may contact David Harrison.

I. LINEAR FITS

This chapter discusses fitting data to linear models. Calling the dependent variable *y* and the independent one *x*, a general representation of such a linear model *f(x)* can be given:

$$y = f(x) = \sum_{k=0}^{m} a_k \, X_k(x) \qquad \text{(I.1)}$$

Here the a_k are the parameters to be fit, and $X_k(x)$ are called the *basis functions*.

By far the most common choice of basis functions are polynomials. Imagine we are trying to fit *y* versus *x* to a straight line.

$$y = a_0 + a_1 x \qquad \text{(I.2)}$$

We are trying to determine the intercept a_0 and the slope a_1, and the two basis functions are *1* and *x*.

Imagine we are fitting the data to a second-order polynomial.

$$y = a_0 + a_1 x + a_2 x^2 \qquad \text{(I.3)}$$

This is a linear fit, and the techniques discussed in this section may be used. The fact that the basis functions are not linear has no relevance in this context.

Imagine we are fitting to an exponential.

$$y = a_1 e^{-a_2 x} \qquad \text{(I.4)}$$

This is not a linear fit, since the parameter a_2 is nonlinear. Note that, in this example, the relationship can be made linear by transforming.

$$ln(y) = ln(a_1) - a_2 x \tag{I.5}$$

Writing $a' \equiv ln(a_1)$ makes the relation a bit clearer.

$$ln(y) = a' - a_2 x \tag{I.6}$$

Thus, fitting the logarithm of $y$versus x to a straight line effectively fits to the original equation and is linear. A small point about this sort of transformation is that it introduces biases into the parameters but often those biases can be ignored; this is discussed in, for example, § 8.2 of the *EDA* manual.

Imagine we are fitting to a more complex exponential.

$$y = a_1 e^{-a_2 x} + a_3 x \tag{I.7}$$

There is no simple transformation that will linearize this form. The techniques discussed in the next section on nonlinear techniques are required.

I.A Least-Squares Regression

The standard technique for performing linear fitting is by least-squares, and this section discuss that algorithm.

However, as Emerson and Hoaglin[1] point out, the technique is not without problems.

> *Various methods have been developed for fitting a straight line of the form:*
>
> *y = a + bx to the data xi,yi, i = 1,...,n.*
>
> *The best-known and most widely used method is least-squares regression, which involves algebraically simple calculations, fits neatly into the framework of inference built on the Gaussian distribution, and requires only a straightforward mathematical derivation. Unfortunately, the least-squares regression line offers no resistance. A wild data point can easily seize control of the fitted line and cause it to give a totally misleading summary of the relationship between y and x.*

[1] John D. Emerson and David C. Hoaglin, "Resistant Lines for y versus x", in David C. Hoaglin, Frederic Mosteller, and John W. Tukey, **Understanding Robust and Exploratory Data Analysis** (John Wiley, 1983, ISBN: 0-471-09777-2), pg.129.

The central idea of the algorithm is that we are seeking a function f(x) which comes as close as possible to the actual experimental data. We let the data consist of N {x,y} pairs.

$$data = (x_{1,}y_1),\ (x_{2,}y_2),\ \cdots\ (x_{N,}y_N) \tag{I.8}$$

Then for each data point the residual is defined as the difference between the experimental value of *y* and the value of *y* given by the function *f* evaluated at the corresponding value of *x*.

$$residual_i = y_i - f\,(x_i) \tag{I.9}$$

First, we define the sum of the squares of the residuals.

$$SumOfSquares = \sum_{i=1}^{N} residual_i^2 \tag{I.10}$$

Then the least-squares technique minimizes the value of *SumOfSquares*.

Here is a simple example. Imagine we have just a succession of x values, which are the result of repeated measurements.

$$(x_1, x_2, \cdots, x_N) \tag{I.11}$$

We wish to find an estimate of the expected value of x from this data. Call that estimated value $\bar{x}$. Then symbolically we may write the sum of the squares.

$$SumOfSquares = \sum_{i=1}^{N} (\bar{x} - x_i)^2 \tag{I.12}$$

For this to be a minimum, the derivative with respect to $\bar{x}$ must be zero:

$$\frac{dSumOfSquares}{d\bar{x}} = 2\sum_{i=1}^{N} (\bar{x} - x_i) = 0$$

$$\sum_{i=1}^{N} (\bar{x} - x_i) = 0$$

We write out the sum.

$$(\bar{x} - x_1) + (\bar{x} - x_2) + \ \cdots\ + (\bar{x} - x_N) = 0$$

We solve for *xbar*.

$$\bar{x} = (x_1 + x_2 + \ \cdots\ + x_N)/N \tag{I.13}$$

But this is just the mean (i.e., average) of the x_i . The mean has no resistance and a single contaminated data point can affect the mean to an arbitrary degree. For example, if x_1 goes to ∞, then so does $\bar{x}$. It is in exactly this sense that the least-squares technique in general offers no resistance.

Usually we are fitting data to a model for which there is more than one parameter.

$$y = a_0 X_0(x) + a_1 X_1(x) + \cdots + a_m X_m(x) \tag{I.14}$$

The least-squares technique then takes the derivative of the sum of the squares of the residuals with respect to each of the parameters to which we are fitting and sets each to zero.

$$\begin{gathered} \frac{\partial\, SumOfSquares}{\partial a_0} = 0 \\ \frac{\partial\, SumOfSquares}{\partial a_1} = 0 \\ . \\ . \\ \frac{\partial\, SumOfSquares}{\partial a_N} = 0 \end{gathered} \tag{I.15}$$

The analytic solution to this set of equations, then, is the result of the fit. Of course, one good way to solve these equations is by using the techniques of solving systems of linear equations that you studied in the Exercise.

If the fit were perfect, then the resulting *SumOfSquares* would be exactly zero. The larger the *SumOfSquares*, the less well the model fits the actual data.

I.B Fitting to Data with Experimental Errors

In an experimental context in the physical sciences almost all measured quantities have an error because a perfect experimental apparatus does not exist. Nonetheless, all too often real experimental data in the sciences and engineering do not have explicit errors associated with the values of the dependent or independent variables. In this case the least-squares fitting techniques discussed in the previous subsection are usually what are used.

If there are assigned errors in the experimental data, say *erry*, then these errors are used to weight each term in the sum of the squares. If the errors are estimates of the standard deviation such a weighted sum is called the "chi squared" (χ^2) of the fit.

$$\chi^2 \equiv \sum_{i=1}^{N} \left(\frac{residual_i}{erry_i} \right)^2 \qquad \text{(I.16)}$$

The least-squares technique takes the derivatives of the χ^2 with respect to the parameters of the fit, sets each equation to zero, and solves the resulting set of equations. Thus, the only difference between this situation and the one discussed in the previous sub-section is that we weight each residual with the inverse of the error.

$$\begin{aligned} \frac{\partial \chi^2}{\partial a_0} &= 0 \\ \frac{\partial \chi^2}{\partial a_1} &= 0 \\ &. \qquad \text{(I.17)} \\ &. \\ \frac{\partial \chi^2}{\partial a_N} &= 0 \end{aligned}$$

Note that finding the solutions to both Equation I.15 of the previous sub-section and Equation I.17 are analytic, and can be solved using the techniques of the Exercise.

Some references refer to the weights w_i of a fit, while others call the errors *erry* the *standard deviation* σ.

$$w_i \equiv 1/erry^2 \equiv 1/\sigma^2$$

Also, some people refer to the "variance", which is the error or standard deviation squared.

If the data has errors in both the independent variable and the dependent one, say *errx* and *erry* respectively, then a common procedure is to use what is called an *effective variance technique*. The idea is fairly simple: the error in the dependent variable y is considered to have two components, one the explicit error *erry* and the other due to the error *errx* in the independent variable x. If *errx* is small, its contribution to the error in y is approximately:

$$\frac{df(x)}{dx}\, errx$$

In the usual case, it is reasonable to assume that these two contributions to the error in y are independent. Thus the effective error in y is found be combining these two terms in quadrature:

$$erry_{eff} \approx \left(erry^2 + (\frac{df(x)}{dx} errx)^2 \right)^{0.5} \tag{I.18}$$

There are subtleties in which value of x should be used in evaluating the slope $df(x)/dx$.[2]

In addition, it is fairly simple to show that unless the model *f(x)* is a straight line then Equations 1.17 are not linear. Thus, in general when the data have explicit errors in both coordinates the least-squares technique must iterate to a solution.

I.C Evaluating the Goodness of a Fit

As already mentioned, when the data have no explicit errors the *SumOfSquares* statistic measures how well the data fits to the model. Although a smaller *SumOfSquares* means a better fit, there is no apriori definition of what the word "small" means in this context. In many cases, analysts will use confidence intervals to try to characterize the goodness of fit for this case. There are many caveats to this approach, some of which are discussed in § 8.2.1 of the *EDA* manual. Nonetheless, the `Statistics‘ConfidenceIntervals‘` package, which is standard with Mathematica, can calculate these types of statistics.

When the data have errors, the ChiSquared statistic does provide information on what "small" means because the data is weighted with the experimenter’s estimate of the errors in the data.

The number of degrees of freedom of a fit is defined as the number of data points minus the number of parameters to which we are fitting. If we are doing, say, a straight line fit to two data points, the degrees of freedom are zero; in this case, the fit is also fairly uninteresting.

If we know the χ^2 and the *DegreesOfFreedom* for a fit, then the χ^2 probability *prob* can be defined:

$$prob = 100\frac{\Gamma(DegreesOfFreedom/2, \chi^2/2)}{\Gamma(DegreesOfFreedom/2)} \tag{I.19}$$

where:

[2] M. Lybanon, Amer. Jour. Phys. **52** (1984), 22 - 26.

$$\Gamma(a) \equiv \int_{t=0}^{\infty} t^{a-1} e^{-t} dt$$

$$\Gamma(a,z) \equiv \int_{t=z}^{\infty} t^{a-1} e^{-t} dt$$

The meaning of the χ^2 probability *prob* is that it is the probability in percent that a repeated experiment will return a chi-squared statistic greater than χ^2.

The interpretation of this statistic is a bit subtle. We assume that the experimental errors are random and statistical. Thus, if we repeated the experiment we would almost certainly get slightly different data, and would therefore get a slightly different result if we fit the new data to the same model as the old data. The χ^2 probability, then, is the chance that the fit to the new data would have a larger χ^2 than the fit we did to the old data.

If our fit returns a χ^2 of zero, then it is almost a certainty that any repeated measurement would yield a larger χ^2; in this case the probability as defined by Equation I.19 is 100%. Such a ‘perfect’ fit is essentially too good to be true.

As the χ^2 increases from zero, the probability decreases. It will be important for you to convince yourself that the ideal result of a fit to data is one in which the χ^2 probability is 50%.

If the probability is much greater than 50%, then the fit is too good; this can arise, for example, if the declared errors in the data are too large. If the probability is much less than 50%, then the fit is poor; this can indicate that the data does not really fit the model being used.

That said, say we have good data including good estimates of its errors, and we are fitting to a model which does match the data. If we repeat the experiment and the fit many times and form a histogram of the χ^2 probability for all the trials, it should be flat; we expect some trials to have very small or large probabilities even though nothing is wrong with the data or the model. Thus, if a single fit has the chi-squared probability that is very large or very small, perhaps the statistics "conspired" and there is nothing wrong with the data or the model. In this case, however, repeating the measurement is probably a good idea.

Despite these limitations, statistical analysis *is* useful. However, you will discover in the experiment that graphical analysis of fits is sometimes even more important. In a sense, this lesson leads to the motivation for the experiment on visualisation of data that you will be doing later in the term.

I.D Reweighting Data That Have No Explicit Errors

When the data have no explicit errors, the linear fitter you will use in the Experiment by default reweights the data.

If we assume that the scatter in the data points is random and statistical then it is reasonable to assume that the error in the dependent variable, *PseudoErrorY*, is given by:

$$PseudoErrorY = \sqrt{SumOfSquares\,/\,DegreesOfFreedom}$$

Thus, we reweight the data by dividing the residuals by this "fake" error when we form the sum of the squares.

The χ^2 calculated using this scheme is always equal to the degrees of freedom of the fit, and therefore is not of use in evaluating the goodness of the fit.

The value of *PseudoErrorY* is sometimes of interest, and is returned by the fitter you will use in the experiment.

For linear fits, the reweighting only changes the estimates in the errors in the parameters to which we are fitting: the values of those parameters is not effected.

Experience has shown that for most experimental data in the the sciences and engineering, reweighting of data is reasonable. Of course, it would be better if the experimentalist had estimated errors when the data were taken.

II. NONLINEAR FITS

The previous section introduced the distinction between linear and nonlinear models. To briefly review, the terms refer to the way in which the parameters to which we are fitting enter into the model. In this section we discuss nonlinear models.

A common use of nonlinear fitters is fitting, say, a nuclear spectrum to a Gaussian plus, say, a linear background. We have a number of *counts* from a multichannel analyzer as a function of the energy *E*.

$$counts = a_0 + a_1 E + a_2 e^{-\left((a_3 - E)^2/(2a_4^2)\right)} \tag{II.1}$$

Here the background has an intercept a_0 and slope a_1, the Gaussian peak has an amplitude of a_2, maximum at a_3 and standard deviation a_4.

Recall that if the data do not contain explicit errors, then we solve the simultaneous equations given by Equation I.15; if the data contain explicit errors we solve Equation I.17. This in general can be done analytically provided the model to which we are fitting is linear in the parameters.

For a nonlinear fit, no such analytic solutions are possible, so iteration is required to find the minimum in the sum of the squares or the chi-squared. linear fitting.

A subtle question is deciding when the iteration has gotten close enough to the answer.

If the data has noise, which is almost a certainty for real experimental data, then there is a further difficulty. We can take two sets of data from the same apparatus using the same sample, fit each dataset to a nonlinear model using identical initial values for the fit parameters, and get very different final fits. This situation leads to ambiguity about which fit results are "correct."

II.A The Levenberg-Marquardt Algorithm

If we imagine a plot of the value of the sum of the squares or the χ^2 as a function of the parameters to which we are fitting, in general for a nonlinear fit there may be many local minima instead of one big one, as is the case for a linear model.

The general technique for iteration, "steepest descent", is analogous to the following situation. It is "a dark and stormy night." Foggy too. You are on the side of a hill and want to find the valley. So you step in the direction in which the slope goes down, and continue moving in the direction of the local definition of "down" until you are in the valley. Of course, if you take *giant steps* you might step over the next hill and end up in the wrong valley. And when you get close to the bottom of the valley you will want to start taking *baby steps*.

Similarly, to do a nonlinear fit we must find the "valley" in the plot of the *SumOfSquares* or χ^2 versus the parameters to which we are fitting. The Levenberg-Marquardt algorithm used by many nonlinear fitters is essentially some clever heuristics to define giant steps and baby steps in the steepest descent method. Further details can be found in the references.

II.B Initial Values of the Parameters

As we stated the previous sub-section, there can be many local minima in the *SumOfSquares* or χ^2. Thus, it is easy for a nonlinear fit to fall into the wrong minimum, giving a totally wrong answer.

The solution is to provide initial estimates of the parameters to a nonlinear fitter, and in general you will discover this to be necessary. If these estimates are poor, one possibility is that the fitter will fall into the wrong minimum. Another possibility is that the fitter will iterate in a direction away from any minimum and "wander in the wilderness" until it is stopped. Thus, a well designed nonlinear fitter will set a maximum to the number of iterations it will perform before aborting.

For the case of, say, nuclear spectra with multiple overlapping peaks, this estimation can become difficult. Despite the claims sometimes seen in glossy advertisements, there is no known software that can find and estimate peaks for data such as this as well as a human expert. This is in part because of the great ability of the human visual system to be an intuitive integrator.

II.C Errors in Both Data Coordinates

Recall from the chapter on linear fitting that if the data have explicit errors in both coordinates, the effective variance technique makes the fit essentially nonlinear unless the model is a straight line. Thus, in this case the fitter you will be using in the experiment iterates until it finds a minimum in the χ^2. When there are errors in both coordinates, the nonlinear fitter you will be using in the experiment also calculates the error in the dependent variable based on the effective variance. However, although there is a fairly comprehensive literature on using this technique in linear fits, I know of no literature about using the technique in nonlinear fits.

The main justification of using effective variances in nonlinear fits is based only on a series of experiments done by David Harrison and former U of T Physics undergraduate Zorawar Singh Bassi in 1996. It was found that most often the algorithm would produce reasonable results.

II.D Reweighting

For linear fits, the default behavior of the fitter is to reweight data without explicit errors by using a statistical assumption about the scatter in the data; this was discussed in § I.D.

For nonlinear fits, reweighting can cause the value of the parameters to which we are fitting to be changed. Thus, by default the nonlinear fitter you will be using does not reweight such data. This behavior is controllable by an option.

REFERENCES

Philip R. Bevington, **Data Reduction and Error Analysis** (McGraw-Hill, 1969). Pages 134 ff and 204 ff introduce least squares techniques. Chapter 11 has a classic introduction to nonlinear fitting techniques.

Gene H. Golub and James M. Ortega, **Scientific Computing and Differential Equations** (Academic Press, 1992).
Pages 89ff and 139ff contains another good introduction to fitting that also

discusses using QR factorization.

Matthew Lybanon, Amer. Jour. Physics **51** (1984), p. 22.
Discusses a modified and improved effective variance technique.

Jay Orear, Amer. Jour. Physics **50** (1982), p 912.
Introduces the effective variance technique.

Xiang Ouyang and Philip L. Varghese, Appl. Optics **28** (1989), p. 1538.
A discussion of a popular Fourier transform-based algorithm for fitting spectra to Galatry and Voigt profiles.

William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling, **Numerical Recipes: The Art of Scientific Computing or Numerical Recipes in C: The Art of Scientific Computing** (Cambridge Univ. Press).
Chapter 14 introduces least square fitting, and includes a valuable section on singular value decomposition. Section 14.4 has good brief introduction to nonlinear fitting and the Levenberg-Marquardt algorithm.

William H. Press and Saul. A. Teukolsky, Computers in Physics **6** (1992) p. 274.
A discussion of fitting when the data has errors in both coordinates, with an example of the Brent method the can be used when the model is linear.

J.R. Taylor, **An Introduction to Error Analysis** (University Science Books, 1982), p. 158ff.
A good discussion of least-squares techniques, this also discusses the "statistical assumption" used by the fitters you will use in this experiment when the data has no explicit errors.

A.P. De Weljer, C.B. Lucasius, L. Buydens, G. Kateman, H.M. Heuvel, and H. Mannee, Anal. Chem. **66** (1994), p 23.
An illuminating discussion of the research into fitting to curves using neural networks, this article also has a good review of some of the problems with conventional techniques.