

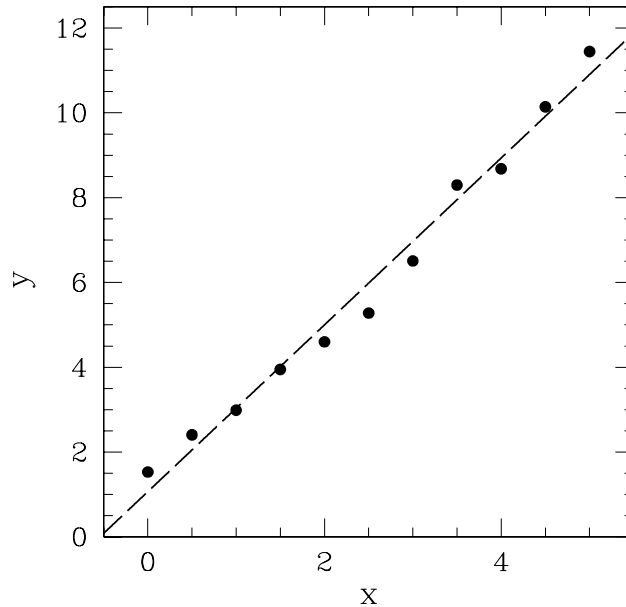
Physics 115/242
Least squares fitting.

Peter Young

(Dated: April 17, 2011)

I. INTRODUCTION

Frequently we are given a set of data points $(x_i, y_i), i = 1, 2, \dots, N$, through which we would like to fit to a smooth function. The function could be straight line (the simplest case), a higher order polynomial, or a more complicated function. As an example, the data in the figure below seems to follow a linear behavior and we may like to determine the “best” straight line through it. More generally, our fitting function, $y = f(x)$, will have some adjustable parameters and we would like to determine the “best” choice of those parameters.



The definition of “best” is not unique. However, the most useful choice, and the one nearly always taken, is the “least squares” fit, in which one minimizes the sum of the squares of the difference between the observed y -value, y_i , and the fitting function evaluated at x_i , i.e. one minimizes

$$\boxed{\sum_{i=1}^N [y_i - f(x_i)]^2}. \tag{1}$$

The simplest cases, and the only ones to be discussed in detail here, are where the fitting function is a *linear function of the parameters*. We shall call this a *linear model*. Examples are a straight line

$$y = a_0 + a_1x \quad (2)$$

and an m -th order polynomial

$$y = a_0 + a_1x + a_2x^2 + \cdots + a_mx^m = \sum_{\alpha=0}^m a_\alpha x^\alpha, \quad (3)$$

where the parameters to be adjusted are the a_α . (Note that we are *not* requiring that y is a linear function of x , only of the fit parameters a_α .)

An example where the fitting function depends *non-linearly* on the parameters is

$$y = a_0x^{a_1} + a_2. \quad (4)$$

Linear models are fairly simple because, as we shall see, the parameters are determined by *linear* equations, which always have a unique solution which can be found by straightforward methods. However, for fitting functions which are non-linear functions of the parameters, the resulting equations are *non-linear* which may have many solutions or none at all, and so are much less straightforward to solve.

Sometimes a non-linear model can be transformed into a linear model by a change of variables. For example if we want to fit to

$$y = a_0x^{a_1}, \quad (5)$$

which has a non-linear dependence on a_1 , then taking logs gives

$$\ln y = \ln a_0 + a_1 \ln x, \quad (6)$$

which is a *linear* function of the parameters $a'_0 = \ln a_0$ and a_1 . Fitting a straight line to a log-log plot is a very common procedure in science and engineering.

II. FITTING TO A STRAIGHT LINE

To see how least squares fitting works, consider the simplest case of a straight line fit, Eq. (2), for which we have to minimize

$$F(a_0, a_1) = \sum_{i=1}^N (y_i - a_0 - a_1x_i)^2, \quad (7)$$

with respect to a_0 and a_1 . Differentiating F with respect to these parameters and setting the results to zero gives

$$\sum_{i=1}^N (a_0 + a_1 x_i) = \sum_{i=1}^N y_i, \quad (8a)$$

$$\sum_{i=1}^N x_i (a_0 + a_1 x_i) = \sum_{i=1}^N x_i y_i. \quad (8b)$$

We write this as

$$U_{00} a_0 + U_{01} a_1 = v_0, \quad (9a)$$

$$U_{10} a_0 + U_{11} a_1 = v_1, \quad (9b)$$

where

$$\boxed{U_{\alpha\beta} = \sum_{i=1}^N x_i^{\alpha+\beta}}, \quad \text{and} \quad (10)$$

$$\boxed{v_\alpha = \sum_{i=1}^N y_i x_i^\alpha}. \quad (11)$$

The matrix notation, while perhaps an overkill here, will be convenient later when we do a general polynomial fit. Note that $U_{10} = U_{01}$. (More generally, later on, U will be a symmetric matrix). Equations (9) are two linear equations in two unknowns. These can be solved by eliminating one variable, which immediately gives an equation for the second one. The solution can also be determined from

$$\boxed{a_\alpha = \sum_{\beta=0}^m (U^{-1})_{\alpha\beta} v_\beta}, \quad (12)$$

where the inverse of the 2×2 matrix U is given, according to standard rules, by

$$U^{-1} = \frac{1}{\Delta} \begin{pmatrix} U_{11} & -U_{01} \\ -U_{01} & U_{00} \end{pmatrix} \quad (13)$$

where

$$\boxed{\Delta = U_{00}U_{11} - U_{01}^2}, \quad (14)$$

and we have noted that U is symmetric so $U_{01} = U_{10}$. The solution for a_0 and a_1 is therefore given by

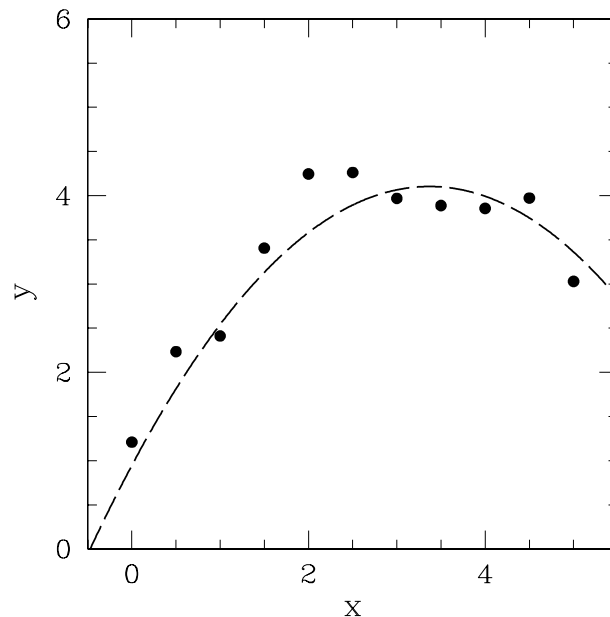
$$\boxed{a_0 = \frac{U_{11} v_0 - U_{01} v_1}{\Delta}}, \quad (15a)$$

$$\boxed{a_1 = \frac{-U_{01} v_0 + U_{00} v_1}{\Delta}}. \quad (15b)$$

We see that it is straightforward to determine the slope, a_1 , and the intercept, a_0 , of the fit from Eqs. (10), (11), (14) and (15) using the N data points (x_i, y_i) .

III. FITTING TO A POLYNOMIAL

Frequently it may be better to fit to a higher order polynomial than a straight line, as for example in the plot below where the fit is a parabola.



Using the notation for an m -th order polynomial in Eq. (3), we need to minimize

$$F(a_0, a_1, \dots, a_m) = \sum_{i=1}^N \left(y_i - \sum_{\alpha=0}^m a_{\alpha} x_i^{\alpha} \right)^2 \quad (16)$$

with respect to the $(m + 1)$ parameters a_{α} . Setting to zero the derivative of F with respect to a_{α} gives

$$\sum_{i=1}^N x_i^{\alpha} \left(y_i - \sum_{\beta=0}^m a_{\beta} x_i^{\beta} \right) = 0, \quad (17)$$

which we write as

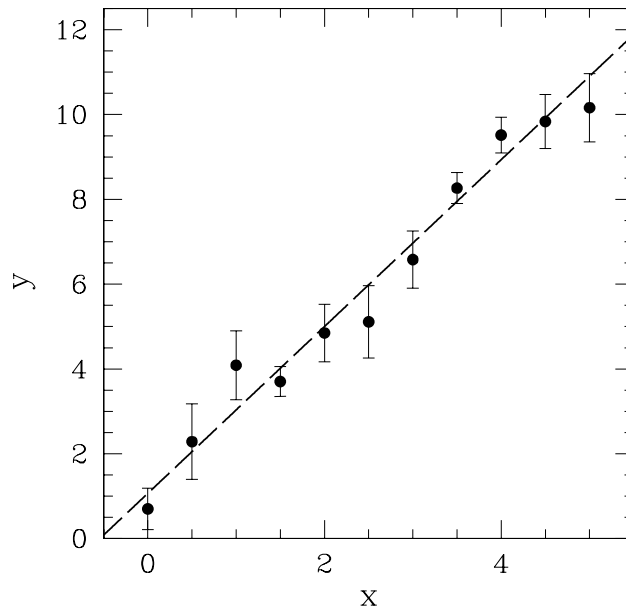
$$\boxed{\sum_{\beta=0}^m U_{\alpha\beta} a_{\beta} = v_{\alpha}}, \quad (18)$$

where $U_{\alpha\beta}$ and v_{α} have been defined in Eqs. (10) and (11). Eq. (18) represents $m + 1$ *linear* equations, one for each value of α . Their solution is given formally by Eq. (12).

Hence polynomial least squares fits, being linear in the parameters, are also quite straightforward. We just have to solve a set of linear equations, Eq. (18), to determine the fit parameters.

IV. FITTING TO DATA WITH ERROR BARS

Frequently we have an estimate of the uncertainty in the data points, the “error bar”. A fit would be considered satisfactory if it goes through the points “within the error bars” (we will discuss at the end of this handout more precisely what this means). An example of data with error bars and a straight line fit is shown in the figure below.



If some points have smaller error bars than other we would like to force the fit to be closer to those points. A suitable quantity to minimize, therefore, is

$$\chi^2 = \sum_{i=1}^N \left(\frac{y_i - f(x_i)}{\sigma_i} \right)^2, \quad (19)$$

called the “chi-squared” function¹, in which σ_i is the error bar for point i . Assuming a polynomial fit, we proceed exactly as before, and again find that the best parameters are given by the solution of Eq. (18), i.e.

$$\sum_{\beta=0}^m U_{\alpha\beta} a_{\beta} = v_{\alpha}, \quad (20)$$

¹ χ^2 is to be thought of as a single variable rather than the square of something called χ . This notation is standard.

but where now $U_{\alpha\beta}$ and v_α are given by

$$U_{\alpha\beta} = \sum_{i=1}^N \frac{x_i^{\alpha+\beta}}{\sigma_i^2}, \quad \text{and} \quad (21)$$

$$v_\alpha = \sum_{i=1}^N \frac{y_i x_i^\alpha}{\sigma_i^2}. \quad (22)$$

The solution of Eqs. (20) can be obtained from the inverse of the matrix U , as in Eq. (12).

Interestingly the matrix U^{-1} contains additional information. We would like to know the *range* of values of the a_α which provide a suitable fit. It turns out that the square of the uncertainty in a_α is just the corresponding diagonal element of U^{-1} , so

$$\delta a_\alpha^2 = (U^{-1})_{\alpha\alpha}. \quad (23)$$

For the case of a straight line fit, the inverse of U is given explicitly in Eq. (13). Using this information, and the values of (x_i, y_i, σ_i) for the data in the above figure, I find that the fit parameters (assuming a straight line fit) are

$$a_0 = 0.840 \pm 0.321, \quad (24)$$

$$a_1 = 2.054 \pm 0.109, \quad (25)$$

in which the error bars on the fit parameters, δa_0 and δa_1 , are determined from Eq. (23). I had generated the data by starting with $y = 1 + 2x$ and adding some noise with zero mean. Hence the fit should be consistent with $y = 1 + 2x$ within the error bars, and it is.

It is all very well to get some fit parameters and error bars but they don't mean much unless the fit really describes the data, which means, roughly speaking, that it goes through the data within the error bars. To see if this is the case, we look at the value χ^2 , Eq. (19), with the optimal parameters. If the fit goes through the data within about one error bar, then χ^2 should be about N . To be more precise, we have to take into account that we *adjusted several parameters* to get the *best* fit. If the number of fit parameters, N_{fit} is equal to the number of data points, then we can clearly get the fit to go *exactly* through the data, so χ^2 would be zero. Hence the relevant quantity is not N but rather, it turns out, *the difference* $N_{\text{DOF}} = N - N_{\text{fit}}$, which is called the number of degrees of freedom (DOF). (When fitting to an m -th order polynomial, $N_{\text{fit}} = m + 1$ and $N_{\text{DOF}} = N - m - 1$.) For a good fit, we expect that χ^2/N_{DOF} should be about 1.

There is a detailed theory, which we do not have time to go into, which converts a value of χ^2 for a given number of degrees of freedom to a ‘‘goodness of fit’’ parameter, Q , which is the

probability that this value of χ^2 , or greater, could occur by chance, assuming that the data points are distributed with a Gaussian² distribution. A very small value indicates that the fit is very unlikely, and one should then look for another model to fit the data. (Another possibility is that the error bars have been underestimated.) The interested student is referred to the books, such as Numerical Recipes, Secs. 15.1 and 15.2, for more details.

² Why Gaussian? From the central limit theorem, discussed a bit later in the course, one can show that, for large N , the distribution of χ^2 in Eq. (19) does not depend on the form of the distribution of the y_i , and is the same as if the distribution of the y_i were Gaussian. Hence, if N is quite large, Q is a good estimate of the probability that the specified value of χ^2 (or larger) could occur by chance, even if the individual data points *don't* have a Gaussian distribution. Note, though, that if N is not large enough for the central limit theorem to apply, the probability of getting a large deviation from the mean is invariably larger than would be expected from a Gaussian distribution. The reason is that the Gaussian falls off very fast at large deviations, and distributions which occur in nature, generically seem to fall off less fast.