SPRINGER BRIEFS IN PHYSICS

Peter Young

Everything You Wanted to Know About Data Analysis and Fitting but Were Afraid to Ask

Springer

SpringerBriefs in Physics

More information about this series at http://www.springer.com/series/8902

Peter Young

Everything You Wanted to Know About Data Analysis and Fitting but Were Afraid to Ask

Springer

Peter Young
Physics Department
University of California
Santa Cruz, CA
USA

ISSN 2191-5423 ISSN 2191-5431 (electronic)
SpringerBriefs in Physics
ISBN 978-3-319-19050-1 ISBN 978-3-319-19051-8 (eBook)
DOI 10.1007/978-3-319-19051-8

Library of Congress Control Number: 2015939824

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

These notes discuss, in a style intended for physicists, how to average data and fit it to some functional form. I try to make clear what is being calculated, what assumptions are being made, and to give a derivation of results rather than just quote them. The aim is put a lot of useful pedagogical material together in a convenient place.

Acknowledgments

I am grateful to Alexander Hartmann for inviting me to give a lecture at the Bad Honnef School on “Efficient Algorithms in Computational Physics” in September 2012, which provided the motivation to write up a first version of these notes, and also for several very helpful discussions.

In addition, I would like to thank Matt Wittmann for helpful discussions on fitting and data analysis using python and for permission to include his python codes. I am also grateful to Wittmann and Christoph Norrenbrock for helpful comments on an earlier version of the manuscript. This work is supported by the Humboldt Foundation, and by the NSF through grant DMR-1207036.

Contents

Chapter 1
Introduction

These notes describe how to average and fit numerical data that you have obtained, presumably by some simulation.

Typically you will generate a set of values x_i, $y_i, \ldots$, $i = 1, \ldots N$, where N is the number of measurements. The first thing you will want to do is to estimate various average values, and determine *error bars* on those estimates. As we shall see, this is straightforward if one wants to compute a single average, e.g. $\langle x \rangle$, but not quite so easy for more complicated averages such as fluctuations in a quantity, $\langle x^2 \rangle - \langle x \rangle^2$, or combinations of measured values such as $\langle y \rangle / \langle x \rangle^2$. Averaging of data will be discussed in Chap. 2.

Having obtained several good data points with error bars, you might want to fit this data to some model. Techniques for fitting data will be described in the second part of these notes in Chap. 3.

I find that the books on these topics usually fall into one of two camps. At one extreme, the books for physicists don't discuss all that is needed and rarely *prove* the results that they quote. At the other extreme, the books for mathematicians presumably prove everything but are written in a style of lemmas, proofs, ϵ's and δ's, and unfamiliar notation, which is intimidating to physicists. One exception, which finds a good middle ground, is Numerical Recipes [1] and the discussion of fitting given here is certainly influenced by Chap. 15 of that book. In these notes I aim to be fairly complete and also to derive the results I use, while the style is that of a physicist writing for physicists. I also include scripts in python, perl, and gnuplot to perform certain tasks in data analysis and fitting. For these reasons, these notes are perhaps rather lengthy. Nonetheless, I hope, that they will provide a useful reference.

Reference

1. W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical Recipes in C*, 2nd edn. (Cambridge University Press, Cambridge, 1992)

P. Young, *Everything You Wanted to Know About Data Analysis and Fitting but Were Afraid to Ask*, SpringerBriefs in Physics,
DOI 10.1007/978-3-319-19051-8_1

Chapter 2
Averages and Error Bars

2.1 Basic Analysis

A reference for the material in this subsection is the book by Taylor [1].

Suppose we have a set of data from a simulation, x_i, $(i = 1, \ldots, N)$, which we shall refer to as a *sample* of data. This data will have some random noise so the x_i are not all equal. Rather they are governed by a distribution $P(x)$, *which we don't know*.

The distribution is normalized,

$$\int_{-\infty}^{\infty} P(x)\, dx = 1, \tag{2.1}$$

and is usefully characterized by its moments, where the nth moment is defined by

$$\langle x^n \rangle = \int_{-\infty}^{\infty} x^n\, P(x)\, dx. \tag{2.2}$$

We will denote the average *over the exact distribution* by angular brackets. Of particular interest are the first and second moments from which one forms the mean μ and variance σ^2, by

$$\mu \equiv \langle x \rangle \tag{2.3a}$$

$$\sigma^2 \equiv \langle\, (x - \langle x \rangle)^2 \,\rangle = \langle x^2 \rangle - \langle x \rangle^2. \tag{2.3b}$$

The term "standard deviation" is used for σ, the square root of the variance.

In this section we will estimate the mean $\langle x \rangle$, and the uncertainty in our estimate, from the N data points x_i. The determination of more complicated averages and resulting error bars will be discussed in Sect. 2.2

P. Young, *Everything You Wanted to Know About Data Analysis and Fitting but Were Afraid to Ask*, SpringerBriefs in Physics,
DOI 10.1007/978-3-319-19051-8_2

In order to obtain error bars we need to assume that the data are uncorrelated with each other. This is a crucial assumption, without which it is very difficult to proceed. However, it is not always clear if the data points are truly independent of each other; some correlations may be present but not immediately obvious. Here, we take the usual approach of assuming that even if there are some correlations, they are sufficiently weak so as not to significantly perturb the results of the analysis. In Monte Carlo simulations, measurements which differ by a sufficiently large number of Monte Carlo sweeps will be uncorrelated. More precisely the difference in sweep numbers should be greater than a "relaxation time". This is exploited in the "binning" method in which the data used in the analysis is not the individual measurements, but rather an average over measurements during a range of Monte Carlo sweeps, called a "bin". If the bin size is greater than the relaxation time, results from adjacent bins will be (almost) uncorrelated. A pedagogical treatment of binning has been given by Ambegaokar and Troyer [2]. Alternatively, one can do independent Monte Carlo runs, requilibrating each time, and use, as individual data in the analysis, the average from each run.

The information *from the data* is usefully encoded in two parameters, the sample mean $\overline{x}$ and the sample standard deviation s which are defined by[1]

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i\,, \tag{2.4a}$$

$$s^2 = \frac{1}{N}\sum_{i=1}^{N} (x_i - \overline{x})^2\,. \tag{2.4b}$$

In statistics, notation is often confusing but crucial to understand. Here, an average indicated by an over-bar, $\overline{\cdots}$, is an average over the *sample of N data points*. This is to be distinguished from an exact average over the distribution $\langle \cdots \rangle$, as in Eqs. (2.3a) and (2.3b). The latter is, however, just a theoretical construct since we *don't know* the distribution $P(x)$, only the set of N data points x_i which have been sampled from it.

[1] The factor of N is often replaced by $N-1$ in the expression for the sample variance in Eq. (2.4b). We note, though, that the final answer for the error bar on the mean, Eq. (2.16), will be independent of how the intermediate quantity s^2 is defined. The rationale for $N-1$ is that the N terms in Eq. (2.4b) are not all independent since $\overline{x}$, which depends on all the x_i, is subtracted. Rather, as will be discussed more in the section on fitting, Chap. 3, there are really only $N-1$ independent variables (called the "number of degrees of freedom" in the fitting context) and so dividing by $N-1$ rather than N also has a rational basis. Here we prefer to use N.

Next we derive two simple results which will be useful later:

1. The mean of the sum of N independent variables *with the same distribution* is N times the mean of a single variable, and
2. The variance of the sum of N independent variables *with the same distribution* is N times the variance of a single variable.

The result for the mean is obvious since, defining $X = \sum_{i=1}^{N} x_i$,

$$\mu_X \equiv \langle X \rangle = \sum_{i=1}^{N} \langle x_i \rangle = N \langle x_i \rangle = N\mu. \tag{2.5}$$

The result for the standard deviation needs a little more work:

$$\begin{aligned}
\sigma_X^2 &\equiv \langle X^2 \rangle - \langle X \rangle^2 && \text{(2.6a)}\\
&= \sum_{i,j=1}^{N} \left(\langle x_i x_j \rangle - \langle x_i \rangle \langle x_j \rangle \right) && \text{(2.6b)}\\
&= \sum_{i=1}^{N} \left(\langle x_i^2 \rangle - \langle x_i \rangle^2 \right) && \text{(2.6c)}\\
&= N \left(\langle x^2 \rangle - \langle x \rangle^2 \right) && \text{(2.6d)}\\
&= N\sigma^2. && \text{(2.6e)}
\end{aligned}$$

To get from Eqs. (2.6b) to (2.6c) we note that, for $i \neq j$, $\langle x_i x_j \rangle = \langle x_i \rangle \langle x_j \rangle$ since x_i and x_j are assumed to be statistically independent. (This is where the statistical independence of the data is needed.) If the means and standard deviations are not all the same, then the above results generalize to

$$\mu_X = \sum_{i=1}^{N} \mu_i, \tag{2.7a}$$

$$\sigma_X^2 = \sum_{i=1}^{N} \sigma_i^2. \tag{2.7b}$$

Now we describe an important thought experiment. Let's *suppose* that we could repeat the set of N measurements *very many* many times, each time obtaining a value of the sample average $\overline{x}$. From these results we could construct a distribution, $\widetilde{P}(\overline{x})$, for the sample average as shown in Fig. 2.1.

If we do enough repetitions we are effectively averaging over the exact distribution. Hence the average of the sample mean, $\overline{x}$, over very many repetitions of the data, is given by

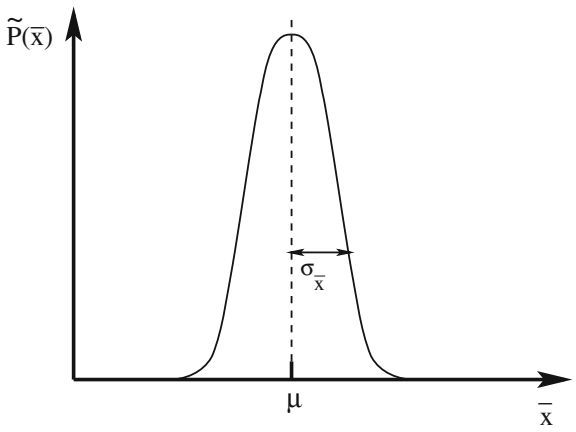

Fig. 2.1 The distribution of results for the sample mean $\overline{x}$ obtained by repeating the measurements of the N data points x_i many times. The average of this distribution is μ, the exact average value of x. The mean, $\overline{x}$, obtained from one sample of data typically differs from μ by an amount of order $\sigma_{\overline{x}}$, the standard deviation of the distribution $\widetilde{P}(\overline{x})$

$$\langle \overline{x} \rangle = \frac{1}{N} \sum_{i=1}^{N} \langle x_i \rangle = \langle x \rangle \equiv \mu, \tag{2.8}$$

i.e. it is the exact average over the distribution of x, as one would intuitively expect, see Fig. 2.1. Eq. (2.8) also follows from Eq. (2.5) by noting that $\overline{x} = X/N$.

In fact, though, we have only the *one* set of data, so we can not determine μ exactly. However, Eq. (2.8) shows that

$$\text{the best estimate of } \mu \text{ is } \overline{x}, \tag{2.9}$$

i.e. the sample mean, since averaging the sample mean over many repetitions of the N data points gives the true mean of the distribution, μ. An estimate like this, which gives the exact result if averaged over many repetitions of the experiment, is said to be unbiased.

We would also like an estimate of the uncertainty, or "error bar", in our estimate of $\overline{x}$ for the exact average μ. We take $\sigma_{\overline{x}}$, the standard deviation in $\overline{x}$ (obtained if one did many repetitions of the N measurements), to be the uncertainty, or error bar, in $\overline{x}$. The reason is that $\sigma_{\overline{x}}$ is the width of the distribution $\widetilde{P}(\overline{x})$, shown in Fig. 2.1, so a *single* estimate $\overline{x}$ typically differs from the exact result μ by an amount of this order. The variance $\sigma_{\overline{x}}^2$ is given by

$$\sigma_{\overline{x}}^2 \equiv \langle \overline{x}^2 \rangle - \langle \overline{x} \rangle^2 = \frac{\sigma^2}{N}, \tag{2.10}$$

which follows from Eq. (2.6e) with $\overline{x} = X/N$.

The problem with Eq. (2.10) is that **we don't know** σ^2 since it is a function of the exact distribution $P(x)$. We do, however, know the *sample* variance s^2, see

Eq. (2.4b), and the average of this over many repetitions of the N data points, is closely related to σ^2 since

$$\langle s^2 \rangle = \frac{1}{N} \sum_{i=1}^{N} \langle x_i^2 \rangle - \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \langle x_i x_j \rangle \tag{2.11a}$$

$$= \langle x^2 \rangle - \frac{1}{N^2} \left[N(N-1) \langle x \rangle^2 + N \langle x^2 \rangle \right] \tag{2.11b}$$

$$= \frac{N-1}{N} \left[\langle x^2 \rangle - \langle x \rangle^2 \right] \tag{2.11c}$$

$$= \frac{N-1}{N} \sigma^2. \tag{2.11d}$$

To get from Eqs. (2.11a) to (2.11b), we have separated the terms with $i = j$ in the last term of Eq. (2.11a) from those with $i \neq j$, and used the fact that each of the x_i is chosen from the same distribution and is statistically independent of the others. It follows from Eq. (2.11c) that

$$\text{the best estimate of } \sigma^2 \text{ is } \frac{N}{N-1} s^2, \tag{2.12}$$

since averaging s^2 over many repetitions of N data points gives σ^2. The estimate for σ^2 in Eq. (2.12) is therefore unbiased. Note that the expression for s^2 in Eq. (2.4a) is a sum of positive terms, so it is "self-averaging", which means that the deviation of the result for one sample of N data points from the average over many data sets (σ^2 in this case) tends to zero for $N \to \infty$.

Combining Eqs. (2.10) and (2.12) gives

$$\text{the best estimate of } \sigma_{\overline{x}}^2 \text{ is } \frac{s^2}{N-1}, \tag{2.13}$$

since this estimate is also unbiased. We have now obtained, using only information from the data, that the mean is given by

$$\mu = \overline{x} \pm \sigma_{\overline{x}}, \tag{2.14}$$

where

$$\sigma_{\overline{x}} = \frac{s}{\sqrt{N-1}}, \tag{2.15}$$

which we can write explicitly in terms of the data points as

$$\sigma_{\overline{x}} = \left[\frac{1}{N(N-1)} \sum_{i=1}^{N} (x_i - \overline{x})^2 \right]^{1/2}. \tag{2.16}$$

Remember that $\overline{x}$ and s are the mean and standard deviation of the (one set) of data that is available to us, see Eqs. (2.4a) and (2.4b).

As an example, suppose $N = 5$ and the data points are

$$x_i = 10, 11, 12, 13, 14, \tag{2.17}$$

(not very random looking data it must be admitted!). Then, from Eq. (2.4a) we have $\overline{x} = 12$, and from Eq. (2.4b)

$$s^2 = \frac{1}{5}\left[(-2)^2 + (-1)^2 + 0^2 + 1^2 + 2^2\right] = 2. \tag{2.18}$$

Hence, from Eq. (2.15),

$$\sigma_{\overline{x}} = \frac{1}{\sqrt{4}}\sqrt{2} = \frac{1}{\sqrt{2}}. \tag{2.19}$$

so

$$\mu = \overline{x} \pm \sigma_{\overline{x}} = 12 \pm \frac{1}{\sqrt{2}}. \tag{2.20}$$

How does the error bar decrease with the number of statistically independent data points N? Equation (2.11d) shows that s^2 does not vary systematically with N, at large N (where we neglect the factor of -1 compared with N) and so from Eq. (2.15) we see that

the error bar in the mean goes down like $1/\sqrt{N}$ for large N.

Hence, to reduce the error bar by a factor of 10 one needs 100 times as much data. This is discouraging, but is a fact of life when dealing with random noise.

For Eq. (2.15) to be really useful we need to know the probability that the true answer μ lies more than $\sigma_{\overline{x}}$ away from our estimate $\overline{x}$. Fortunately, for large N, the central limit theorem, derived in Appendix A, tells us (for distributions where the first two moments are finite) that the distribution of $\overline{x}$ is a Gaussian. For this distribution we know that the probability of finding a result more than one standard deviation away from the mean is 32 %, more than two standard deviations is 4.5 % and more than three standard deviations is 0.3 %. Hence we expect that most of the time $\overline{x}$ will be within $\sigma_{\overline{x}}$ of the correct result μ, and only occasionally will be more than two times $\sigma_{\overline{x}}$ from it. Even if N is not very large, so there are some deviations from the Gaussian form, the above numbers are often a reasonable guide.

However, as emphasized in Appendix A, distributions which occur in nature typically have much more weight in the tails than a Gaussian. As a result, the weight in the tails of the distribution *of the sum* can also be much larger than for a Gaussian even for quite large values of N, see Fig. A.1. It follows that the probability of an "outlier" can be much higher than that predicted for a Gaussian distribution, as anyone who has invested in the stock market knows well!

We conclude this subsection by discussing the situation when there are several random variables, x, y, z, ..., for which we generate a sample of data: $(x_i, y_i, z_i, \ldots)$ with $i = 1, 2, \ldots, N$. We indicate the means and standard deviations of the different variables by suffices, i.e.

$$\mu_x \equiv \langle x \rangle \tag{2.21a}$$

$$\sigma_x^2 \equiv \langle x^2 \rangle - \langle x \rangle^2, \tag{2.21b}$$

for averages over the exact distribution, and

$$s_x^2 \equiv \frac{1}{N} \sum_{i=1}^{N} (x_i - \overline{x})^2 \,. \tag{2.22}$$

for the sample variance. The main new feature is the appearance of cross-correlations between different variables. One defines the "covariance" of x and y by

$$\mathrm{Cov}(x, y) \equiv \langle xy \rangle - \langle x \rangle \langle y \rangle = \langle (x - \langle x \rangle)\ (y - \langle y \rangle) \rangle. \tag{2.23}$$

It is convenient to have a more compact notation for the covariance, analogous to that in Eq. (2.21b) for the variance. I use the notation σ_{xy}^2 for the covariance of x and y, i.e.

$$\sigma_{xy}^2 \equiv \langle (x - \langle x \rangle)\ (y - \langle y \rangle) \rangle. \tag{2.24}$$

This notation is not ideal since there is no guarantee that the covariance σ_{xy}^2 is positive.[2] The standard notation is to write the covariance of x and y as σ_{xy} (no square), but I find this even more confusing.

By analogy to Eq. (2.24) I write the sample covariance of x and y as

$$s_{xy}^2 \equiv \frac{1}{N} \sum_{i=1}^{N} (x_i - \overline{x})\ (y_i - \overline{y}) \,. \tag{2.25}$$

2.2 Advanced Analysis

In Sect. 2.1 we learned how to estimate a simple average, such as $\mu_x \equiv \langle x \rangle$, plus the error bar in that quantity, from a set of data x_i. Trivially this method also applies to a *linear* combination of different averages, μ_x, μ_y, ... etc. However, we often need

[2] One should therefore think of σ_{xy}^2 as a single quantity, rather than the square of something, just as χ^2, discussed extensively in the section on fitting below, is never regarded as the square of an object called χ. Admittedly, though, χ^2 can not be negative.

more complicated, *non-linear* functions of averages. One example is the fluctuations in a quantity, i.e. $\langle x^2 \rangle - \langle x \rangle^2$. Another example is a dimensionless combination of moments, which gives information about the *shape* of a distribution independent of its overall scale. Such quantities are very popular in finite-size scaling (FSS) analyses since the FSS form is simpler than for quantities with dimension. An popular example, first proposed by Binder, is $\langle x^4 \rangle / \langle x^2 \rangle^2$, which is known as the "kurtosis" (frequently a factor of 3 is subtracted to make it zero for a Gaussian).

Hence, in this section we consider how to determine *non-linear functions* of averages of one or more variables, $f(\mu_y, \mu_z, \ldots)$, where

$$\mu_y \equiv \langle y \rangle, \tag{2.26}$$

etc. For example, the two quantities mentioned in the previous paragraph correspond to

$$f(\mu_y, \mu_z) = \mu_y - \mu_z^2, \tag{2.27}$$

with $y = x^2$ and $z = x$ and

$$f(\mu_y, \mu_z) = \frac{\mu_y}{\mu_z^2}, \tag{2.28}$$

with $y = x^4$ and $z = x^2$.

The natural estimate of $f(\mu_y, \mu_z)$ from the sample data is clearly $f(\overline{y}, \overline{z})$. However, it will take some more thought to estimate the error bar in this quantity. The traditional way of doing this is called "error propagation", described in Sect. 2.2.1 and Chap. 3 of Ref. [1]. However, it is now more common to use either "jackknife" or "bootstrap" procedures, described in Sects. 2.2.2 and 2.2.3. At the price of some additional computation, which is no difficulty when done on a modern computer (though it would have been tedious in the old days when statistics calculations were done by hand), these methods automate the calculation of the error bar.

In addition, the estimate of $f(\mu_y, \mu_z)$ turns out to have some *bias* if f is a non-linear function. Usually this is small effect because it is order $1/N$, see for example Eq. (2.34), whereas the statistical error is of order $1/\sqrt{N}$. Since N is usually large, the bias is generally much less than the statistical error and so can generally be neglected. In any case, the jackknife and bootstrap methods also enable one to eliminate the leading ($\sim 1/N$) contribution to the bias in a automatic fashion.

2.2.1 Traditional Method

First we will discuss the traditional method, known as error propagation [1], to compute the error bar and bias. We expand $f(\overline{y}, \overline{z})$ about $f(\mu_y, \mu_z)$ up to second order in the deviations:

$$f(\overline{y},\overline{z}) = f(\mu_y,\mu_z) + (\partial_{\mu_y} f)\,\delta_{\overline{y}} + (\partial_{\mu_z} f)\,\delta_{\overline{z}} + \frac{1}{2}\,(\partial^2_{\mu_y\mu_y} f)\,\delta^2_{\overline{y}} + (\partial^2_{\mu_y\mu_z} f)\,\delta_{\overline{y}}\delta_{\overline{z}} + \frac{1}{2}\,(\partial^2_{\mu_z\mu_z} f)\,\delta^2_{\overline{z}} + \cdots, \tag{2.29}$$

where

$$\delta_{\overline{y}} = \overline{y} - \mu_y, \tag{2.30}$$

etc.

The terms of first order in the $\delta's$ in Eq. (2.29) give the leading contribution to the error, but would average to zero if the procedure were to be repeated many times. However, the terms of second order do not average to zero and so give the leading contribution to the bias. We now estimate that bias.

Averaging Eq. (2.29) over many repetitions, and noting that

$$\langle\delta^2_{\overline{y}}\rangle = \langle\overline{y}^2\rangle - \langle\overline{y}\rangle^2 \equiv \sigma^2_{\overline{y}}, \quad \langle\delta^2_{\overline{z}}\rangle = \langle\overline{z}^2\rangle - \langle\overline{z}\rangle^2 \equiv \sigma^2_{\overline{z}}, \quad \langle\delta_{\overline{y}}\delta_{\overline{z}}\rangle = \langle\overline{y}\,\overline{z}\rangle - \langle\overline{y}\rangle\langle\overline{z}\rangle \equiv \sigma^2_{\overline{y}\,\overline{z}}, \tag{2.31}$$

we get

$$\langle f(\overline{y},\overline{z})\rangle - f(\mu_y,\mu_z) = \frac{1}{2}\,(\partial^2_{\mu_y\mu_y} f)\,\sigma^2_{\overline{y}} + (\partial^2_{\mu_y\mu_z} f)\,\sigma^2_{\overline{y}\,\overline{z}} + \frac{1}{2}\,(\partial^2_{\mu_z\mu_z} f)\,\sigma^2_{\overline{z}} + \cdots. \tag{2.32}$$

As shown in Eq. (2.13) $\sigma^2_{\overline{y}}$ is $(N-1)^{-1}$ times the average sample variance $\langle s^2_y\rangle$. Furthermore, as noted below Eq. (2.12), s^2_y is self averaging, which means that the difference between the value of s^2_y from one data set and the average over all data sets, σ^2_y, tends to zero for $N \to \infty$. Hence we can replace $\sigma^2_{\overline{y}}$ by $(N-1)^{-1}s^2_y$, and similarly replace $\sigma^2_{\overline{z}}$ by $(N-1)^{-1}s^2_z$. In the same way, we can replace $\sigma^2_{\overline{y}\,\overline{z}}$ by $(N-1)^{-1}$ times s^2_{yz}, the sample *covariance* of y and z, defined in Eq. (2.25). Hence, from Eq. (2.32), we have

$$f(\mu_y,\mu_z) = \langle f(\overline{y},\overline{z})\rangle - \frac{1}{(N-1)}\left[\frac{1}{2}\,(\partial^2_{\mu_y\mu_y} f)\,s^2_y + (\partial^2_{\mu_y\mu_z} f)\,s^2_{yz} + \frac{1}{2}\,(\partial^2_{\mu_z\mu_z} f)\,s^2_z\right] + \cdots. \tag{2.33}$$

The leading contribution to the bias is the $1/(N-1)$ term. It follows from Eq. (2.33) that if one wants to eliminate the leading contribution to the bias one should

$$\text{estimate } f(\mu_y,\mu_z) \text{ from } f(\overline{y},\overline{z}) - \frac{1}{(N-1)}\left[\frac{1}{2}\,(\partial^2_{\mu_y\mu_y} f)\,s^2_y + (\partial^2_{\mu_y\mu_z} f)\,s^2_{yz} + \frac{1}{2}\,(\partial^2_{\mu_z\mu_z} f)\,s^2_z\right]. \tag{2.34}$$

As claimed earlier, the bias correction is of order $1/N$. Note that it vanishes if f is a linear function, as shown in Sect. 2.1. The generalization to functions of more than two averages, $f(\mu_y, \mu_z, \mu_w, \ldots)$, is obvious.

Next we discuss the leading *error* in using $f(\overline{y}, \overline{z})$ as an estimate for $f(\mu_y, \mu_z)$. This comes from the terms linear in the δ's in Eq. (2.29). Just including these terms we have

$$\langle f(\overline{y}, \overline{z}) \rangle = f(\mu_y, \mu_z), \tag{2.35a}$$

$$\langle f^2(\overline{y}, \overline{z}) \rangle = f^2(\mu_y, \mu_z) + (\partial_{\mu_y} f)^2 \langle \delta_{\overline{y}}^2 \rangle + 2(\partial_{\mu_y} f)(\partial_{\mu_z} f) \langle \delta_{\overline{y}} \delta_{\overline{z}} \rangle + (\partial_{\mu_z} f)^2 \langle \delta_{\overline{z}}^2 \rangle. \tag{2.35b}$$

Hence

$$\begin{aligned} \sigma_f^2 &\equiv \langle f^2(\overline{y}, \overline{z}) \rangle - \langle f(\overline{y}, \overline{z}) \rangle^2 \\ &= (\partial_{\mu_y} f)^2 \langle \delta_{\overline{y}}^2 \rangle + 2(\partial_{\mu_y} f)(\partial_{\mu_z} f) \langle \delta_{\overline{y}} \delta_{\overline{z}} \rangle + (\partial_{\mu_z} f)^2 \langle \delta_{\overline{z}}^2 \rangle. \end{aligned} \tag{2.36}$$

As above, we use $s_{yy}^2/(N-1)$ as an estimate of $\langle \delta_{\overline{y}}^2 \rangle$ and similarly for the other terms. Hence

$$\text{the best estimate of } \sigma_f^2 \text{ is } \frac{1}{(N-1)} (\partial_{\mu_y} f)^2 s_y^2 + 2(\partial_{\mu_y} f)(\partial_{\mu_z} f) s_{yz}^2 + (\partial_{\mu_z} f)^2 s_z^2. \tag{2.37}$$

This estimate is unbiased to leading order in N. Note that we need to keep track not only of fluctuations in y and z, characterized by their variances s_y^2 and s_z^2, but also cross correlations between y and z, characterized by their covariance s_{yz}^2.

Hence, still to leading order in N, we get

$$f(\mu_y, \mu_z) = f(\overline{y}, \overline{z}) \pm \sigma_f, \tag{2.38}$$

where we estimate the error bar σ_f from Eq. (2.37) which shows that it is of order $1/\sqrt{N}$. The generalization to functions of more than two averages is obvious.

Note that in the simple case studied in Sect. 2.1 where $f(\mu_x)$ is a linear function, $f = \mu_x$, Eq. (2.33) tells us that there is no bias, which is correct, and Eq. (2.37) gives an expression for the error bar which agrees with Eq. (2.15).

In Eqs. (2.34) and (2.37) we need to keep track how errors in the individual quantities like $\overline{y}$ propagate to the estimate of the function f. This requires inputting by hand the various partial derivatives into the analysis program, and keeping track of all the variances and covariances. In the next two sections we see how *resampling* the data automatically takes account of error propagation without needing to input the partial derivatives and keep track of variances and covariances. There are two resampling approaches, called jackknife and bootstrap, and each provide a *fully automatic* method of determining error bars and bias.

2.2.2 Jackknife

We define the ith jackknife estimate, y_i^J $(i = 1, 2, \ldots, N)$ to be the average over all data in the sample *except the point* i, i.e.

$$y_i^J \equiv \frac{1}{N-1} \sum_{j \neq i} y_j \quad \left(= \overline{y} + \frac{1}{N-1} (\overline{y} - y_i) \right). \tag{2.39}$$

We also define corresponding jackknife estimates of the function f (again for concreteness we will assume that f is a function of just 2 averages but the generalization will be obvious):

$$f_i^J \equiv f(y_i^J, z_i^J). \tag{2.40}$$

In other words, we use the jackknife values, y_i^J, z_i^J, rather than the sample means, $\overline{y}, \overline{z}$, as the arguments of f. For example a jackknife estimate of the Binder ratio $\langle x^4 \rangle / \langle x^2 \rangle^2$ is

$$f_i^J = \frac{(N-1)^{-1} \sum_{j,(j \neq i)} x_j^4}{\left[(N-1)^{-1} \sum_{j,(j \neq i)} x_j^2 \right]^2} \tag{2.41}$$

The overall jackknife estimate of $f(\mu_y, \mu_z)$ is then the average over the N jackknife estimates f_i^J:

$$\overline{f^J} \equiv \frac{1}{N} \sum_{i=1}^{N} f_i^J. \tag{2.42}$$

It is straightforward to show that if f is a linear function of μ_y and μ_z then $\overline{f^J} = f(\overline{y}, \overline{z})$, i.e. the jackknife and standard averages are identical, see e.g. Eq. (2.39). However, when f is not a linear function, so there is bias, there *is* a difference, and we will now show the resampling carried out in the jackknife method can be used to determine bias and error bars in an automated way.

We proceed as for the derivation of Eq. (2.33), which we now write as

$$f(\mu_y, \mu_z) = \langle f(\overline{y}, \overline{z}) \rangle - \frac{A}{N} - \frac{B}{N^2} + \cdots, \tag{2.43}$$

where A is the term in rectangular brackets in Eq. (2.33), and we have added the next order correction. The jackknife data sets have $N-1$ points with the same distribution as the N points in the actual distribution, and so the bias in the jackknife average will be of the same form, with the same values of A and B, but with N replaced by $N-1$, i.e.

$$f(\mu_y, \mu_z) = \langle \overline{f^J} \rangle - \frac{A}{N-1} - \frac{B}{(N-1)^2} \cdots. \tag{2.44}$$

We can therefore eliminate the leading contribution to the bias by forming an appropriate linear combination of $f(\overline{y},\overline{z})$ and $\overline{f^J}$, namely

$$f(\mu_y,\mu_z) = N\langle f(\overline{y},\overline{z})\rangle - (N-1)\langle\overline{f^J}\rangle + O\left(\frac{1}{N^2}\right). \tag{2.45}$$

It follows that, to eliminate the leading bias without computing partial derivatives, one should

$$\text{estimate } f(\mu_y,\mu_z) \text{ from } Nf(\overline{y},\overline{z}) - (N-1)\overline{f^J}. \tag{2.46}$$

The bias is then of order $1/N^2$. However, as mentioned earlier, bias is usually not a big problem because, even without eliminating the leading contribution, the bias is of order $1/N$ whereas the statistical error is of order $1/\sqrt{N}$ which is much bigger if N is large. In most cases, therefore, N is sufficiently large that one can use *either* the usual average $f(\overline{y},\overline{z})$, or the jackknife average $\overline{f^J}$ in Eq. (2.42), to estimate $f(\mu_y,\mu_z)$, since the difference between them will be much smaller than the statistical error. In other words, elimination of the leading bias using Eq. (2.46) is usually not necessary.

Next we show that the jackknife method gives error bars, which agree with Eq. (2.37) but without the need to explicitly keep track of the partial derivatives and the variances and covariances.

We define the variance of the jackknife averages by

$$s^2_{f^J} \equiv \overline{\left(f^J\right)^2} - \left(\overline{f^J}\right)^2, \tag{2.47}$$

where

$$\overline{\left(f^J\right)^2} = \frac{1}{N}\sum_{i=1}^{N}\left(f_i^J\right)^2. \tag{2.48}$$

Using Eqs. (2.40) and (2.42), we expand $\overline{f^J}$ away from the exact result $f(\mu_y,\mu_z)$. Just including the leading contribution gives

$$\begin{aligned}
\overline{f^J} - f(\mu_y,\mu_z) &= \frac{1}{N}\sum_{i=1}^{N}\left[(\partial_{\mu_y} f)\,(y_i^J-\mu_y) + (\partial_{\mu_z} f)\,(z_i^J-\mu_z)\right] \\
&= \frac{1}{N(N-1)}\sum_{i=1}^{N}\Big[(\partial_{\mu_y} f)\,\left\{N(\overline{y}-\mu_y) - (y_i-\mu_y)\right\} \\
&\qquad\qquad + (\partial_{\mu_z} f)\,\{N(\overline{z}-\mu_z) - (z_i-\mu_z)\}\Big] \\
&= (\partial_{\mu_y} f)\,(\overline{y}-\mu_y) + (\partial_{\mu_z} f)\,(\overline{z}-\mu_z),
\end{aligned} \tag{2.49}$$

where we used Eq. (2.39). Similarly we find

$$\begin{aligned}\overline{(f^J)^2} &= \frac{1}{N}\sum_{i=1}^{N}\left[f(\mu_y,\mu_z)+(\partial_{\mu_y}f)\,(y_i^J-\mu_y)+(\partial_{\mu_z}f)\,(z_i^J-\mu_z)\right]^2\\ &= f^2(\mu_y,\mu_z)+2f(\mu_y,\mu_z)\left[(\partial_{\mu_y}f)\,(\overline{y}-\mu_y)+(\partial_{\mu_z}f)\,(\overline{z}-\mu_z)\right]\\ &\quad+(\partial_{\mu_y}f)^2\left[(\overline{y}-\mu_y)^2+\frac{s_y^2}{(N-1)^2}\right]+(\partial_{\mu_z}f)^2\left[(\overline{z}-\mu_z)^2+\frac{s_z^2}{(N-1)^2}\right]\\ &\quad+2(\partial_{\mu_y}f)(\partial_{\mu_z}f)\left[(\overline{y}-\mu_y)(\overline{z}-\mu_z)+\frac{s_{yz}^2}{(N-1)^2}\right].\end{aligned} \tag{2.50}$$

Hence, from Eqs. (2.47) to (2.49), the variance in the jackknife estimates is given by

$$s_{f^J}^2 = \frac{1}{(N-1)^2}\left[(\partial_{\mu_y}f)^2\,s_y^2+(\partial_{\mu_z}f)^2\,s_z^2+2(\partial_{\mu_y}f)(\partial_{\mu_z}f)s_{yz}^2\right], \tag{2.51}$$

which is just $1/(N-1)$ times σ_f^2, the estimate of the square of the error bar in $f(\overline{y},\overline{z})$ given in Eq. (2.37). Hence

$$\text{the jackknife estimate for } \sigma_f \text{ is } \sqrt{N-1}\,s_{f^J}. \tag{2.52}$$

Note that this is directly obtained from the jackknife estimates without having to put in the partial derivatives by hand. Note too that the $\sqrt{N-1}$ factor is in the *numerator* whereas the factor of $\sqrt{N-1}$ in Eq. (2.15) is in the *denominator*. Intuitively the reason for this difference is that the jackknife estimates are very close, much closer than the error in the means, since they would all be equal except that each one omits just one data point.

If N is very large, roundoff errors could become significant from having to subtract large, almost equal, numbers to get the error bar from the jackknife method. It is then advisable to group the N data points into N_{group} groups (or "bins") of data and take, as individual data points in the jackknife analysis, the average of the data in each group. The above results clearly go through with N replaced by N_{group}.

To summarize this subsection, to estimate $f(\mu_y,\mu_z)$ one can use either $f(\overline{y},\overline{z})$ or the jackknife average $\overline{f^J}$ in Eq. (2.42). The error bar in this estimate, σ_f, is related to the standard deviation in the jackknife estimates s_{f^J} by Eq. (2.52).

2.2.3 Bootstrap

The bootstrap, like the jackknife, is a resampling of the N data points. A brief discussion, in the context of data analysis is given in Ref. [3]. Whereas jackknife considers N new data sets, each of containing all the original data points minus

one, bootstrap uses N_{boot} data sets each containing N points obtained by random (Monte Carlo) sampling of the original set of N points. During the Monte Carlo sampling, the probability that a data point is picked is $1/N$ irrespective of whether it has been picked before. (In the statistics literature this is called picking from a set "with replacement".) Hence a given data point x_i will, *on average*, appear once in each Monte Carlo-generated data set, but may appear not at all, or twice, and so on. The probability that x_i appears n_i times is close to a Poisson distribution with mean unity. However, it is not exactly Poissonian because of the constraint in Eq. (2.53). It turns out that we shall need to include the deviation from the Poisson distribution even for large N. We shall use the term "bootstrap" to denote the Monte Carlo-generated data sets.

More precisely, let us suppose that the number of times x_i appears in a bootstrap data set is n_i. Since each bootstrap dataset contains exactly N data points, we have the constraint

$$\sum_{i=1}^{N} n_i = N. \tag{2.53}$$

Consider one of the N variables x_i. Each time we generate an element in a bootstrap dataset the probability that it is x_i is $1/N$, which we will denote by p. From standard probability theory, the probability that x_i occurs n_i times is given by a binomial distribution

$$P(n_i) = \frac{N!}{n_i!\,(N-n_i)!}\, p^{n_i}(1-p)^{N-n_i}. \tag{2.54}$$

The mean and standard deviation of a binomial distribution are given by

$$[n_i]_{\text{MC}} = Np = 1, \tag{2.55}$$

$$[n_i^2]_{\text{MC}} - [n_i]_{\text{MC}}^2 = Np(1-p) = 1 - \frac{1}{N}, \tag{2.56}$$

where $[\ldots]_{\text{MC}}$ denotes an exact average over bootstrap samples (for a fixed original data set x_i). For $N \to \infty$, the binomial distribution goes over to a Poisson distribution, for which the factor of $1/N$ in Eq. (2.56) does not appear. We assume that N_{boot} is sufficiently large that the bootstrap average we carry out reproduces this result with sufficient accuracy. Later, we will discuss what values for N_{boot} are sufficient in practice. Because of the constraint in Eq. (2.53), n_i and n_j (with $i \neq j$) are not independent and we find, by squaring Eq. (2.53) and using Eqs. (2.55) and (2.56), that

$$[n_i n_j]_{\text{MC}} - [n_i]_{\text{MC}}[n_j]_{\text{MC}} = -\frac{1}{N} \qquad (i \neq j). \tag{2.57}$$

First of all we just consider the simple average $\mu_x \equiv \langle x \rangle$, for which, of course, the standard methods in Sect. 2.1 suffice, so bootstrap is not necessary. However, this will show how to get averages and error bars in a simple case, which we will then generalize to non-linear functions of averages.

We denote the average of x for a given bootstrap data set by x_α^B, where α runs from 1 to N_{boot}, i.e

$$x_\alpha^B = \frac{1}{N}\sum_{i=1}^{N} n_{i,\alpha} x_i . \tag{2.58}$$

We then compute the bootstrap average of the mean of x and the bootstrap variance in the mean, by averaging over all the bootstrap data sets. We assume that N_{boot} is large enough for the bootstrap average to be exact, so we can use Eqs. (2.56) and (2.57). The result is

$$\overline{x^B} \equiv \frac{1}{N_{\text{boot}}}\sum_{\alpha=1}^{N_{\text{boot}}} x_\alpha^B = \frac{1}{N}\sum_{i=1}^{N}[n_i]_{\text{MC}} x_i = \frac{1}{N}\sum_{i=1}^{N} x_i = \overline{x} \tag{2.59}$$

$$s_{x^B}^2 \equiv \overline{\left(x^B\right)^2} - \left(\overline{x^B}\right)^2 = \frac{1}{N^2}\left(1-\frac{1}{N}\right)\sum_i x_i^2 - \frac{1}{N^3}\sum_{i\neq j} x_i x_j = \frac{1}{N}\left(\overline{x^2} - \overline{x}^2\right) = \frac{s^2}{N}, \tag{2.60}$$

where

$$\overline{\left(x^B\right)^2} \equiv \frac{1}{N_{\text{boot}}}\sum_{\alpha=1}^{N_{\text{boot}}}\left[\left(x_\alpha^B\right)^2\right]_{\text{MC}} . \tag{2.61}$$

We now average Eqs. (2.59) and (2.60) over many repetitions of the original data set x_i. Averaging Eq. (2.59) gives

$$\langle \overline{x^B} \rangle = \langle \overline{x} \rangle = \langle x \rangle \equiv \mu_x . \tag{2.62}$$

This shows that the bootstrap average $\overline{x^B}$ is an unbiased estimate of the exact average μ_x. Averaging Eq. (2.60) gives

$$\left\langle s_{x^B}^2 \right\rangle = \frac{1}{N}\left\langle s^2 \right\rangle = \frac{N-1}{N^2}\sigma^2 = \frac{N-1}{N}\sigma_{\overline{x}}^2, \tag{2.63}$$

where we used Eqs. (2.10) and (2.11c). Since $\sigma_{\overline{x}}$ is the uncertainty in the sample mean, we see that

$$\text{the bootstrap estimate of } \sigma_{\overline{x}} \text{ is } \sqrt{\frac{N}{N-1}}\, s_{x^B} . \tag{2.64}$$

Remember that s_{x^B} is the standard deviation of the bootstrap data sets. Usually N is sufficiently large that the square root in Eq. (2.64) can be replaced by unity.

As for the jackknife, these results can be generalized to finding the error bar in some possibly non-linear function, $f(\mu_y, \mu_z)$, rather than for μ_x. One computes the bootstrap estimates for $f(\mu_y, \mu_z)$, which are

$$f_\alpha^B = f(y_\alpha^B, z_\alpha^B). \tag{2.65}$$

In other words, we use the bootstrap values, y_α^B, f_α^B, rather than the sample means, $\overline{y}$, $\overline{z}$, as the arguments of f. The final bootstrap estimate for $f(\mu_y, \mu_z)$ is the average of these, i.e

$$\overline{f^B} = \frac{1}{N_{\text{boot}}} \sum_{\alpha=1}^{N_{\text{boot}}} f_\alpha^B. \tag{2.66}$$

Following the same methods in the jackknife section, one obtains the error bar, σ_f, in $f(\mu_y, \mu_z)$. The result is

$$\text{the bootstrap estimate for } \sigma_f \text{ is } \sqrt{\frac{N}{N-1}}\, s_{f^B}, \tag{2.67}$$

where

$$s_{f^B}^2 = \overline{\left(f^B\right)^2} - \left(\overline{f^B}\right)^2, \tag{2.68}$$

is the variance of the bootstrap estimates. Here

$$\overline{\left(f^B\right)^2} \equiv \frac{1}{N_{\text{boot}}} \sum_{\alpha=1}^{N_{\text{boot}}} \left(f_\alpha^B\right)^2. \tag{2.69}$$

Usually N is large enough that the factor of $\sqrt{N/(N-1)}$ is Eq. (2.67) can be replaced by unity. Equation (2.67) corresponds to the result Eq. (2.64) which we derived for the special case of $f = \mu_x$.

Again, following the same path as in the jackknife section, it is straightforward to show that the bias of the estimates in Eqs. (2.66) and (2.67) is of order $1/N$ and so vanishes for $N \to \infty$. However, if N is not too large it may be useful to eliminate the leading contribution to the bias in the mean, as we did for jackknife in Eq. (2.46). The result is that one should

$$\text{estimate } f(\mu_y, \mu_z) \text{ from } 2f(\overline{y}, \overline{z}) - \overline{f^B}. \tag{2.70}$$

The bias in Eq. (2.70) is of order $1/N^2$, whereas $f(\overline{y}, \overline{z})$ and $\overline{f^B}$ each have a bias of order $1/N$. However, it is not normally necessary to eliminate the bias since, if N is large, the bias is much smaller than the statistical error.

I have not systematically studied the values of N_{boot} that are needed in practice to get accurate estimates for the error. It seems that N_{boot} in the range 100–500 is typically chosen, and this seems to be adequate irrespective of how large N is.

It is sometimes stated, e.g. [3], that the bootstrap method can give error bars correctly even when there are correlations in the data. This is not so. If one applies bootstrap to the direct average of a set of data, it simply reproduces the results of

the standard analysis. Bootstrap is useful both to get error bars when one is looking at combination of averages of the data, and to get confidence limits when the noise on the data is not Gaussian, see Sect. 3.7. Unfortunately, bootstrap does not work miracles and cannot give correct error bars for correlated data.

To summarize this subsection, to estimate $f(\mu_y, \mu_z)$ one can either use $f(\overline{y}, \overline{z})$, or the bootstrap average in Eq. (2.66), and the error bar in this estimate, σ_f, is related to the standard deviation in the bootstrap estimates by Eq. (2.67).

2.2.4 Jackknife or Bootstrap?

The jackknife approach involves less calculation than bootstrap, and is fine for estimating combinations of moments of the measured quantities. Furthermore, identical results are obtained each time jackknife is run on the same set of data, which is not the case for bootstrap. However, the range of the jackknife estimates is very much smaller, by a factor of $\sqrt{N}$ for large N, than the scatter of averages which would be obtained from individual data sets, see Eq. (2.52). By contrast, for bootstrap, σ_{f^B}, which measures the deviation of the bootstrap estimates f_α^B from the result for the single actual data set $f(\overline{y}, \overline{z})$, *is equal to* σ_f, the deviation of the average of a single data set from the exact result $f(\mu_y, \mu_z)$ (if we replace the factor of $N/(N-1)$ by unity, see Eq. (2.67)). This is the main strength of the bootstrap approach; it samples the full range of the distribution of the sample distribution. Hence, if you want to generate data which covers the full range then should use bootstrap. This is useful in fitting, see for example, Sect. 3.7. However, if you just want to generate error bars on combinations of moments quickly and easily, then use jackknife.

References

1. J.R. Taylor, *The Study of Uncertainties in Physical Measurements* (University Science Books, Sausalito, California, 1997)
2. V. Ambegaokar, M. Troyer, Estimating errors reliably in Monte Carlo simulations of the Ehrenfest model. Am. J. Phys. **78**, 150 (2009)
3. M.E.J. Newman, G.T. Barkema, *Monte Carlo Methods in Statistical Physics* (Oxford University Press Inc., New York, USA, 1999)

Chapter 3
Fitting Data to a Model

A good reference for the material in this section is Chap. 15 of Numerical Recipes [1].

Frequently we are given a set of data points (x_i, y_i), $i = 1, 2, \cdots, N$, with corresponding error bars, σ_i, through which we would like to fit to a smooth function $f(x)$. The function could be straight line (the simplest case), a higher order polynomial, or a more complicated function. The fitting function will depend on M "fitting parameters", a_α and we would like the "best" fit obtained by adjusting these parameters. We emphasize that a fitting procedure should not only

1. give the values of the fit parameters, but also
2. provide error estimates on those parameters, and
3. provide a measure of how good the fit is.

If the result of part 3 is that the fit is very poor, the results of parts 1 and 2 are probably meaningless.

The definition of "best" is not unique. However, the most useful choice, and the one nearly always taken, is "least squares". For this case, one minimizes the weighted sum of the squares of the difference between the observed y-value, y_i, and the fitting function evaluated at x_i. The weight of each point depends on its error bar, since the fit should be more tightly bound to points with smaller error bars than to those with large error bars. The quantity to be minimized, called χ^2 ("chi-squared"),[1] is defined by

$$\chi^2 = \sum_{i=1}^{N} \left(\frac{y_i - f(x_i)}{\sigma_i} \right)^2 . \tag{3.1}$$

A big advantage of least squares over other definitions of "best" fit is that for a linear model (see below) the equations which determine the fit parameters are themselves

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

P. Young, *Everything You Wanted to Know About Data Analysis and Fitting but Were Afraid to Ask*, SpringerBriefs in Physics,
DOI 10.1007/978-3-319-19051-8_3

linear. We should mention that Eq. (3.1) implicitly assumes that the data points are uncorrelated. A generalization of the least squares method which is applicable, in principle, to correlated data is given later in Eqs. (G.5)–(G.7).

Often we assume that the distribution of the errors is Gaussian, since, according to the central limit theorem discussed in Appendix A, the sum of N independent random variables has a Gaussian distribution (under fairly general conditions) if N is large. However, distributions which occur in nature usually have more weight in the "tails" than a Gaussian, and as a result, even for moderately large values of N, the probability of an "outlier" might be much bigger than expected from a Gaussian, see Fig. A.1.

If the errors *are* distributed with a Gaussian distribution, and if $f(x)$ has the *exact* values of the fit parameters, then χ^2 in Eq. (3.1) is a sum of squares of N random variables with a Gaussian distribution with mean zero and standard deviation unity. However, when we have minimized the value of χ^2 with respect to the M fitting parameters a_α the terms are not all independent. It turns out, see Appendix B, that, at least for a linear model (which we define below), the distribution of χ^2 at the minimum is that of the sum of the squares of $N - M$ (not N) Gaussian random variables with zero mean and standard deviation unity.[2] We call $N - M$ the "number of degrees of freedom" ($N_{\rm DOF}$). The χ^2 distribution is discussed in Appendix C. The formula for it is Eq. (C.6).

The simplest problems are where the fitting function is a *linear function of the parameters*. We shall call this a *linear model*. Examples are a straight line ($M = 2$),

$$y = a_0 + a_1 x, \tag{3.2}$$

and an $(M - 1)$th order polynomial,

$$y = a_0 + a_1 x + a_2 x^2 + \cdots + a_{M-1} x^{M-1} = \sum_{\alpha=0}^{M-1} a_\alpha x^\alpha, \tag{3.3}$$

where the parameters to be adjusted are the a_α. (Note that we are *not* stating here that y has to be 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

$$y = a_0 x^{a_1} + a_2. \tag{3.4}$$

Linear models are fairly simply because, as we shall see below, the parameters are determined by *linear* equations, which, in general, have a unique solution that 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. We shall discuss fitting to both linear and non-linear models in these notes.

[2] Although this result is only valid if the fitting model is linear in the parameters, it is usually taken to be a reasonable approximation for non-linear models as well.

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

$$y = a_0 x^{a_1}, \tag{3.5}$$

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

$$\ln y = \ln a_0 + a_1 \ln x, \tag{3.6}$$

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. However, it should be noted that transforming the data does not exactly take Gaussian errors into Gaussian errors, though the difference will be small if the errors are "sufficiently small". For the above log transformation this means $\sigma_i / y_i \ll 1$, i.e. the *relative* error is much less than unity.

3.1 Fitting to a Straight Line

To see how least squares fitting works, consider the simplest case of a straight line fit, Eq. (3.2), for which we have to minimize

$$\chi^2(a_0, a_1) = \sum_{i=1}^{N} \left(\frac{y_i - a_0 - a_1 x_i}{\sigma_i} \right)^2, \tag{3.7}$$

with respect to a_0 and a_1. Differentiating χ^2 with respect to these parameters and setting the results to zero gives

$$a_0 \sum_{i=1}^{N} \frac{1}{\sigma_i^2} + a_1 \sum_{i=1}^{N} \frac{x_i}{\sigma_i^2} = \sum_{i=1}^{N} \frac{y_i}{\sigma_i^2}, \tag{3.8a}$$

$$a_0 \sum_{i=1}^{N} \frac{x_i}{\sigma_i^2} + a_1 \sum_{i=1}^{N} \frac{x_i^2}{\sigma_i^2} = \sum_{i=1}^{N} \frac{x_i y_i}{\sigma_i^2}. \tag{3.8b}$$

We write this as

$$U_{00}\, a_0 + U_{01}\, a_1 = v_0, \tag{3.9a}$$

$$U_{10}\, a_0 + U_{11}\, a_1 = v_1, \tag{3.9b}$$

where

$$U_{\alpha\beta} = \sum_{i=1}^{N} \frac{x_i^{\alpha+\beta}}{\sigma_i^2}, \quad \text{and} \tag{3.10}$$

$$v_\alpha = \sum_{i=1}^{N} \frac{y_i\, x_i^{\alpha}}{\sigma_i^2}. \tag{3.11}$$

The matrix notation, while 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 (3.9) 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

$$a_\alpha = \sum_{\beta=0}^{M-1} \left(U^{-1}\right)_{\alpha\beta} v_\beta, \tag{3.12}$$

(where we have temporarily generalized to a polynomial of order $M-1$). For the straight-line fit, the inverse of the 2×2 matrix U is given, according to standard rules, by

$$U^{-1} = \frac{1}{\Delta} \begin{pmatrix} U_{11} & -U_{01} \\ -U_{01} & U_{00} \end{pmatrix} \tag{3.13}$$

where

$$\Delta = U_{00}U_{11} - U_{01}^2, \tag{3.14}$$

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

$$a_0 = \frac{U_{11}\, v_0 - U_{01}\, v_1}{\Delta}, \tag{3.15a}$$

$$a_1 = \frac{-U_{01}\, v_0 + U_{00}\, v_1}{\Delta}. \tag{3.15b}$$

We see that it is straightforward to determine the slope, a_1, and the intercept, a_0, of the fit from Eqs. (3.10), (3.11), (3.14) and (3.15) using the N data points (x_i, y_i), and their error bars σ_i.

3.2 Fitting to a Polynomial

Frequently we need to fit to a higher order polynomial than a straight line, in which case we minimize

$$\chi^2(a_0, a_1, \cdots, a_{M-1}) = \sum_{i=1}^{N} \left(\frac{y_i - \sum_{\alpha=0}^{M-1} a_\alpha x_i^\alpha}{\sigma_i} \right)^2 \tag{3.16}$$

with respect to the M parameters a_α. Setting to zero the derivatives of χ^2 with respect to the a_α gives

$$\sum_{\beta=0}^{M-1} U_{\alpha\beta}\, a_\beta = v_\alpha, \tag{3.17}$$

where $U_{\alpha\beta}$ and v_α have been defined in Eqs. (3.10) and (3.11). Equation (3.17) represents M *linear* equations, one for each value of α. Their solution is again given by Eq. (3.12), i.e. it is expressed in terms of the inverse matrix U^{-1}, which is called the *covariance matrix*.

3.3 Error Bars

In addition to the best fit values of the parameters we also need to determine the error bars in those values. Interestingly, this information is *also* contained in the covariance matrix U^{-1}.

First of all, we explain the significance of error bars in fit parameters. We assume that the data is described by a model with a particular set of parameters $\vec{a}^{\,\text{true}}$ which, unfortunately, we don't know. If we were, somehow, to have many real data sets each one would give a different set of fit parameters $\vec{a}^{\,(i)}$, $i = 0, 1, 2, \ldots$, because of noise in the data, *clustered about the true set* $\vec{a}^{\,\text{true}}$. Projecting on to a single fit parameter, a_1 say, there will be a distribution of values $P(a_1)$ centered on a_1^{true} with standard deviation σ_1, see the top part of Fig. 3.1. Typically the value of a_1 obtained from our *one actual data set*, $a_1^{(0)}$, will lie within about σ_1 of a_1. Hence we define the error bar to be σ_1.

Unfortunately, we can't determine the error bar this way because we have only one actual data set, which we denote here by $y_i^{(0)}$ to distinguish it from other data sets that we will introduce. Our actual data set gives one set of fit parameters, which we call $\vec{a}^{\,(0)}$. Suppose, however, we were to generate many *simulated* data sets from of the one which is available to us, by generating random values (possibly with a Gaussian distribution though this won't be necessary yet) centered at the y_i with standard deviation σ_i. Fitting each simulated dataset would give different values for

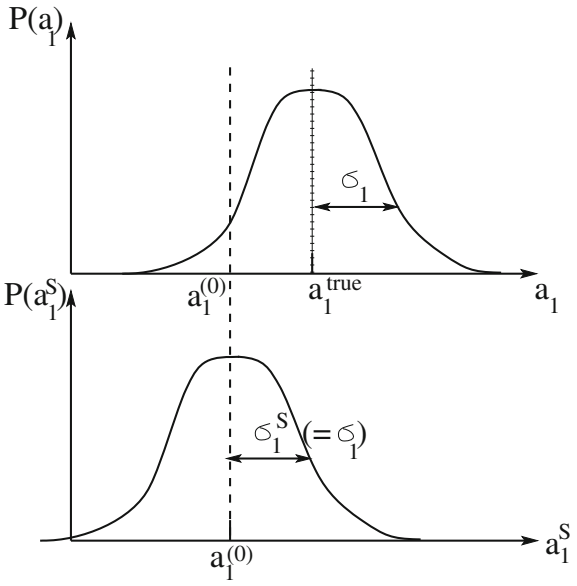

Fig. 3.1 The *top* figure shows the distribution of one of the fit parameters a_1 if one could obtain many real data sets. The distribution has standard deviation σ_1 about the true value a_1^{true} and is Gaussian if the noise on the data is Gaussian. In fact, however, we have only one actual data set which has fit parameter $a_1^{(0)}$, and this typically lies within about σ_1 of a_1^{true}. Hence we define the error bar on the estimate of a_1^{true} to be σ_1. However, we cannot calculate σ_1 directly from the distribution of a_1 because we have only one the one value, $a_1^{(0)}$. However, we can generate many *simulated* data sets from the one actual set and hence we can estimate the standard deviation, σ_1^S, of the distribution of the resulting fit parameter a_1^S, which is shown in the lower figure. This distribution is centered about the value from the actual data, $a_1^{(0)}$, and has standard deviation, σ_1^S. The important point is that if one assumes a linear model then one can show that $\sigma_1^S = \sigma_1$, see text. Even if the model is non linear, one usually assumes that the difference in the standard deviations is sufficiently small that one can still equate the true error bar with the standard deviation from the simulated data sets. We emphasize that the error bar quoted by fitting programs is actually σ_1^S, and this is assumed to equal σ_1. Furthermore, as shown in Appendices E and F, if the noise on the data is Gaussian (and the model is linear) both the distributions in this figure are also Gaussian

$\vec{a}$, *clustered now about* $\vec{a}^{(0)}$, see the bottom part of Fig. 3.1. We now come to an important, but rarely discussed, point:

> For a linear model the standard deviation of the fit parameters of these simulated data sets about $\vec{a}^{(0)}$, is equal to the standard deviation of the fit parameters of real data sets $\vec{a}$ about $\vec{a}^{\text{true}}$. The latter is what we *really* want to know (since it is our estimate of the error bar on $\vec{a}^{\text{true}}$) but can't determine directly. See Fig. 3.1 for an illustration. This result is also applicable to a non-linear model if it can be represented by an effective linear model for the needed range of parameters about the minimum of χ^2. Furthermore, we show in Appendices E and F that if the noise on the data is Gaussian (and the model is linear), the two distributions in Fig. 3.1 are also both Gaussian.

We shall now prove this result. As stated above, to derive the error bars in the fit parameters we take simulated values of the data points, y_i^S, which vary by some amount δy_i^S about $y_i^{(0)}$, i.e. $\delta y_i^S = y_i^S - y_i^{(0)}$, with a standard deviation given by the error bar σ_i. The fit parameters of this simulated data set, $\vec{a}^S$, then deviate from $\vec{a}^{(0)}$ by an amount $\delta\vec{a}^S$ where

$$\delta a_\alpha^S = \sum_{i=1}^{N} \frac{\partial a_\alpha}{\partial y_i}\, \delta y_i^S. \tag{3.18}$$

Averaging over fluctuations in the y_i^S we get the variance of a_α^S to be

$$\left(\sigma_\alpha^S\right)^2 \equiv \langle \left(\delta a_\alpha^S\right)^2 \rangle = \sum_{i=1}^{N} \sigma_i^2 \left(\frac{\partial a_\alpha}{\partial y_i}\right)^2, \tag{3.19}$$

since $\langle \left(\delta y_i^S\right)^2 \rangle = \sigma_i^2$, and the data points y_i are statistically independent. Writing Eq. (3.12) explicitly in terms of the data values,

$$a_\alpha = \sum_\beta \left(U^{-1}\right)_{\alpha\beta} \sum_{i=1}^{N} \frac{y_i\, x_i^\beta}{\sigma_i^2}, \tag{3.20}$$

and noting that U is independent of the y_i, we get

$$\frac{\partial a_\alpha}{\partial y_i} = \sum_\beta \left(U^{-1}\right)_{\alpha\beta} \frac{x_i^\beta}{\sigma_i^2}. \tag{3.21}$$

Substituting into Eq. (3.19) gives

$$\left(\sigma_\alpha^S\right)^2 = \sum_{\beta,\gamma} \left(U^{-1}\right)_{\alpha\beta} \left(U^{-1}\right)_{\alpha\gamma} \left[\sum_{i=1}^{N} \frac{x_i^{\beta+\gamma}}{\sigma_i^2}\right]. \tag{3.22}$$

The term in rectangular brackets is just $U_{\beta\gamma}$, and since U is given by Eq. (3.10) and is symmetric, the last equation reduces to

$$\left(\sigma_\alpha^S\right)^2 = \left(U^{-1}\right)_{\alpha\alpha}. \tag{3.23}$$

Recall that σ_α^S is the standard deviation of the fitted parameter values about the $\vec{a}^{(0)}$ when constructing simulated data sets from the one set of data that is available to us.

However, the error bar is defined to be the standard deviation the fitted parameter values would have relative to a_α^{true} if we could average over many actual data sets. To

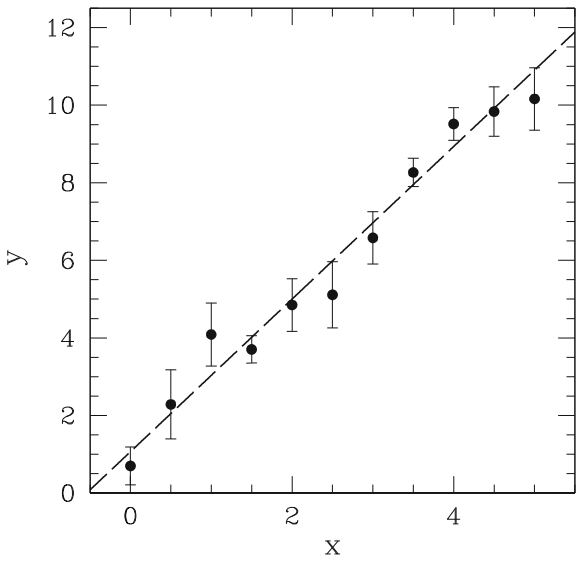

Fig. 3.2 An example of a straight-line fit to a set of data with error bars

determine this quantity we simply repeat the above calculation with $\delta y_i = y_i - y_i^{\text{true}}$ in which y_i is the value of the ith data point in one of the actual data sets. Since U is a constant (for a linear model) equations (3.18) to (3.23) go through unchanged simply omitting the superscript S's. The (unknown) values of y_i^{true} never enter. In other words

$$\sigma_\alpha^2 = \left(U^{-1}\right)_{\alpha\alpha}, \tag{3.24}$$

which shows that $\sigma_\alpha^s = \sigma_s$ for a linear model. However, this equality does not hold precisely for fitting to a non-linear model. We have therefore showed that the diagonal elements of the covariance matrix U^{-1} give the square of the errors bar in the fit parameters.

In addition to error bars, we also need a parameter to describe the quality of the fit. A useful quantity is the probability that, given the fit, the data could have occurred with a χ^2 greater than or equal to the value found. This is generally denoted by Q and, as shown in Appendix C, is given by

$$Q = \frac{1}{\Gamma(N_{\text{DOF}}/2)} \int_{\chi^2/2}^{\infty} y^{(N_{\text{DOF}}/2)-1}\, e^{-y}\, dy, \tag{3.25}$$

assuming the data have Gaussian noise. Here $N_{\text{DOF}} \equiv N - M$ is the number of degrees of freedom. Note that the effects of *non-Gaussian* statistics is to increase the probability of outliers, so fits with a fairly small value of Q, say around 0.01, may be considered acceptable. However, fits with a *very* small value of Q should not be trusted and the values of the fit parameters are probably meaningless in these cases.

For the case of a straight line fit, the inverse of U is given explicitly in Eq. (3.13). Using this information, and the values of (x_i, y_i, σ_i) for the data in Fig. 3.2, the fit parameters (assuming a straight line fit) are

$$a_0 = 0.84 \pm 0.32, \tag{3.26}$$

$$a_1 = 2.05 \pm 0.11, \tag{3.27}$$

in which the error bars on the fit parameters on a_0 and a_1, which are denoted by σ_0 and σ_1, are determined from Eq. (3.24). The data was generated by starting with $y = 1 + 2x$ and then adding some noise with zero mean. Hence the fit should be consistent with $y = 1 + 2x$ within the error bars, and it is. The value of χ^2 is 7.44 so $\chi^2/N_{\text{DOF}} = 7.44/9 = 0.866$ and the quality of fit parameter, given by Eq. (3.25), is $Q = 0.592$ which is good.

The off-diagonal elements of the covariance matrix U^{-1} are also useful since they contain information about correlations between the fitted parameters. More precisely, one can show, following the lines of the above derivation of σ_α^2, that the correlation of fit parameters α and β, known mathematically as their "covariance", is given by the appropriate off-diagonal element of the covariance matrix,

$$\text{Cov}(\alpha, \beta) \equiv \langle \delta a_\alpha \, \delta a_\beta \rangle = \left(U^{-1}\right)_{\alpha\beta}. \tag{3.28}$$

The correlation coefficient, $r_{\alpha\beta}$, which is a dimensionless measure of the correlation between δa_α and δa_β lying between -1 and 1, is given by

$$r_{\alpha\beta} = \frac{\text{Cov}(\alpha, \beta)}{\sigma_\alpha \sigma_\beta}. \tag{3.29}$$

A good fitting program should output the correlation coefficients as well as the fit parameters, their error bars, the value of χ^2/N_{DOF}, and the goodness of fit parameter Q.

So far we have considered a polynomial fit, which is a particular case of a linear model. If we fit to a *general* linear model, writing

$$f(x) = \sum_{\alpha=1}^{M} a_\alpha \, X_\alpha(x), \tag{3.30}$$

where $X_1(x), X_2(x), \cdots, X_M(x)$ are fixed functions of x called basis functions, χ^2 is given by

$$\chi^2 = \sum_{i=1}^{N} \left(\frac{y_i - \sum_{\alpha=1}^{M} a_\alpha X_\alpha(x_i)}{\sigma_i} \right)^2, \tag{3.31}$$

and the matrix U is given by

$$U_{\alpha\beta} = \sum_{i=1}^{N} \frac{X_\alpha(x_i)\, X_\beta(x_i)}{\sigma_i^2}. \tag{3.32}$$

Similarly, the quantities v_α in Eq. (3.11) become

$$v_\alpha = \sum_{i=1}^{N} \frac{y_i\, X_\alpha(x_i)}{\sigma_i^2}, \tag{3.33}$$

and, as before, the best fit parameters are given by the solution of the M linear equations

$$\sum_{\beta=1}^{M} U_{\alpha\beta}\, a_\beta = v_\alpha, \tag{3.34}$$

for $\alpha = 1, 2, \ldots, M$, namely by Eq. (3.12).

For a linear model, χ^2 is a quadratic function of the fit parameters and so the elements of the "*curvature matrix*",[3] $(1/2)\, \partial^2\chi^2/\partial a_\alpha \partial a_\beta$ are constants, independent of the values of the fit parameters. In fact, we see from Eqs. (3.1), (3.30) and (3.32) that

$$\frac{1}{2}\frac{\partial^2\chi^2}{\partial a_\alpha \partial a_\beta} = U_{\alpha\beta}, \tag{3.35}$$

so *the curvature matrix is equal to* U, given by Eq. (3.32) (Eq. (3.10) for a polynomial fit). Note that for a linear model the curvature matrix U is a constant, independent of the fit parameters and data values. However, U is not constant for a non-linear model.

3.4 Interpolating

In a physics context it usually the fitting parameters per se which are of interest. However, in other contexts where fitting is performed, for example machine learning [2], one is less interested in the fitting model and more interested in predicting the value of the function for a new value of x. We give a brief discussion of this here.

We assume a general linear model as in Eq. (3.30). Clearly the best estimate for y at some value x is the fitting function with the optimized fitting parameters evaluated at x, i.e.

[3] It is conventional to include the factor of $1/2$.

$$y(x) = \sum_{\alpha=1}^{M} a_\alpha \, X_\alpha(x), \tag{3.36}$$

and the error bar (squared) is given by

$$\begin{aligned} \sigma_y^2 &= \sum_{\alpha,\beta} \langle \delta a_\alpha \delta a_\beta \rangle \, X_\alpha(x) X_\beta(x) \\ &= \sum_{\alpha,\beta} \left(U^{-1} \right)_{\alpha\beta} X_\alpha(x) X_\beta(x), \end{aligned} \tag{3.37}$$

where we used Eqs. (3.24) and (3.28).

It is instructive to substitute for the optimized fitting parameters into Eq. (3.36), i.e.

$$\begin{aligned} y(x) &= \sum_{\alpha=1}^{M} X_\alpha(x) \sum_{\beta=1}^{M} \left(U^{-1} \right)_{\alpha\beta} \sum_{i=1}^{N} \frac{y_i X_\beta(x_i)}{\sigma_i^2} \\ &= \sum_{i=1}^{N} k(x, x_i) \, y_i, \end{aligned} \tag{3.38}$$

where the *kernel* $k(x, x_i)$ is given by

$$k(x, x_i) = \sum_{\alpha,\beta} \frac{X_\alpha(x) \left(U^{-1} \right)_{\alpha\beta} X_\beta(x_i)}{\sigma_i^2}. \tag{3.39}$$

The kernel is independent of the y_i and has the simple, intuitive property that

$$\sum_{i=1}^{N} k(x, x_i) = 1. \tag{3.40}$$

Clearly Eq. (3.40) is correct if the y_i in Eq. (3.38) are constant. In fact, Eq. (3.40) is correct quite generally as long as one of the basis functions is a constant (the usual situation). In this case $X_1(x)$, say, is equal to 1 so $\sum_i X_\beta(x_i)/\sigma_i^2 = U_{0\beta}$ according to Eq. (3.32). Hence

$$\sum_{i=1}^{N} k(x, x_i) = \sum_{\alpha,\beta} X_\alpha(x) \left(U^{-1} \right)_{\alpha\beta} U_{\beta 1} = \sum_{\alpha} X_\alpha(x) \, \delta_{\alpha 1} = 1, \tag{3.41}$$

which is Eq. (3.40).

Presumably the data close to x have the larger weight in Eq. (3.40). It would be interesting to examine this. I expect that Eq. (3.40) is approximately true even if one of the basis functions is not a constant.

3.5 Fitting to a Non-linear Model

As for linear models, one minimizes χ^2 in Eq. (3.1). The difference is that the resulting equations are non-linear so there might be many solutions or non at all. Techniques for solving the coupled non-linear equations invariably require specifying an initial value for the variables a_α. The most common method for fitting to non-linear models is the Levenberg-Marquardt (LM) method, see e.g. Numerical Recipes [1]. Implementing the Numerical Recipes code for LM is a little complicated because it requires the user to provide a routine for the derivatives of χ^2 with respect to the fit parameters as well as for χ^2 itself, and to check for convergence. Alternatively, one can use the fitting routines in the `scipy` package of `python` or use `gnuplot`. But see the comments in Appendix D about getting the error bars in the parameters correct, which apply when fitting to linear as well as non-linear models. Gnuplot and scipy scripts for fitting to a non-linear model are given in Appendix H.

One difference from fitting to a linear model is that the curvature matrix, defined by the LHS of Eq. (3.35), is not constant but is a function of the fit parameters. Hence it is no longer true that the standard deviations of the two distributions in Fig. 3.1 are equal. However, it still generally assumed that the difference is small enough to be unimportant and hence that the covariance matrix, which is now defined to be the inverse of the curvature matrix *at the minimum of* χ^2, still gives information about error bars on the fit parameters. This is discussed more in the next two subsections, in which we point out, however, that a more detailed analysis is needed if the model is non-linear and the spread of fitted parameters is sufficiently large that it *cannot* be represented by an effective linear model, i.e. χ^2 is not well fitted by a parabola over the needed range of parameter values.

As a reminder:

- The *curvature matrix* is defined in general by the LHS of Eq. (3.35), which, for a linear model, is equivalent to Eq. (3.32) (Eq. (3.10) for a polynomial fit.)
- The *covariance matrix* is the inverse of the curvature matrix. For a linear model this matrix is constant, independent of the fit parameters or data values. However, for a non-linear model this is no longer true and we are then interested in the covariance matrix at the minimum of χ^2. Its diagonal elements give error bars on the fit parameters according to Eq. (3.24) (but see the caveat in the previous paragraph for non-linear models) and its off-diagonal elements give correlations between fit parameters according to Eqs. (3.28) and (3.29).

3.6 Confidence Limits

In the last two subsections we showed that the diagonal elements of the covariance matrix give an error bar on the fit parameters. In this section we extend the notion of error bar to embrace the concept of a "confidence limit".

There is a theorem [1] which states that, for a linear model, if we take simulated data sets assuming Gaussian noise in the data about the actual data points, and compute the fit parameters $\vec{a}^{S(i)}$, $i = 1, 2, \ldots$ for each data set, then the probability distribution of the $\vec{a}^S$ is given by the multi-variable Gaussian distribution

$$P(\vec{a}^S) \propto \exp\left(-\frac{1}{2}\sum_{\alpha,\beta} \delta a_\alpha^S \, U_{\alpha\beta} \, \delta a_\beta^S\right), \tag{3.42}$$

where $\delta\vec{a}^S \equiv \vec{a}^{S(i)} - \vec{a}^{(0)}$, U, given by Eq. (3.32), is the curvature matrix which can also be defined in terms of the second derivative of χ^2 according to Eq. (3.35), and $\vec{a}^{(0)}$ is the fit to the actual data set. A proof of this result is given in Appendix E. It applies for a linear model with Gaussian noise, and also for a non-linear model if the uncertainties in the parameters do not extend outside a region where an effective linear model could be used. (In the latter case one still needs a non-linear routine to *find* the best parameters). Note that for a non-linear model, U is not a constant and it is the curvature *at the minimum* of χ^2 which has to be put into Eq. (3.42).

From Eq. (3.35) the change in χ^2 as the parameters are varied away from the minimum is given by

$$\Delta\chi^2 \equiv \chi^2(\vec{a}^S) - \chi^2(\vec{a}^{(0)}) = \sum_{\alpha,\beta} \delta a_\alpha^S \, U_{\alpha\beta} \, \delta a_\beta^S, \tag{3.43}$$

in which the χ^2 are all evaluated from the single (actual) data set $y_i^{(0)}$. Equation (3.42) can therefore be written as

$$P(\vec{a}^S) \propto \exp\left(-\frac{1}{2}\Delta\chi^2\right). \tag{3.44}$$

We remind the reader that in deriving Eq. (3.44) we have assumed the noise in the data is Gaussian and that either the model is linear or, if non-linear, the uncertainties in the parameters do not extend outside a region where an effective linear model could be used.

Hence the probability of a particular deviation, $\delta\vec{a}^S$, of the fit parameters in a simulated data set away from the parameters in the *actual* data set, depends on how much this change increases χ^2 (evaluated from the actual data set) away from the minimum. In general a "confidence limit" is the range of fit parameter values such that $\Delta\chi^2$ is less than some specified value. The simplest case, and the only one we discuss here, is the variation of *one* variable at a time, though multi-variate confidence limits can also be defined, see Numerical Recipes [1].

We therefore consider the change in χ^2 when one variable, a_1^S say, is held at a specified value, and all the others ($\beta = 2, 3, \cdots, M$) are varied in order to minimize χ^2. Minimizing $\Delta\chi^2$ in Eq. (3.43) with respect to a_β^S gives

$$\sum_{\gamma=1}^{M} U_{\beta\gamma}\,\delta a_\gamma^S = 0, \qquad (\beta = 2, 3, \ldots, M). \tag{3.45}$$

The corresponding sum for $\beta = 1$, namely $\sum_{\gamma=1}^{M} U_{1\gamma}\,\delta a_\gamma^S$, is not zero because δa_1 is fixed. It will be some number, c say. Hence we can write

$$\sum_{\gamma=1}^{M} U_{\alpha\gamma}\,\delta a_\gamma^S = c_\alpha, \qquad (\alpha = 1, 2, \ldots, M), \tag{3.46}$$

where $c_1 = c$ and $c_\beta = 0\,(\beta \neq 1)$. The solution is

$$\delta a_\alpha^S = \sum_{\beta=1}^{M} \left(U^{-1}\right)_{\alpha\beta} c_\beta = \left(U^{-1}\right)_{\alpha 1} c. \tag{3.47}$$

For $\alpha = 1$ this gives

$$c = \delta a_1^S / \left(U^{-1}\right)_{11}. \tag{3.48}$$

Substituting Eq. (3.47) into Eq. (3.43), and using Eq. (3.48) we find that $\Delta\chi^2$ is related to $\left(\delta a_1^S\right)^2$ by

$$\Delta\chi^2 = \frac{(\delta a_1^S)^2}{\left(U^{-1}\right)_{11}}. \tag{3.49}$$

(Curiously, the coefficient of $(\delta a_1)^2$ is one over the 11 element of the inverse of U, rather than U_{11} which is how it appears in Eq. (3.43) in which the $\beta \neq 1$ parameters are free rather than adjusted to minimize χ^2.)

From Eq. (3.44) we finally get

$$P(a_1^S) \propto \exp\left(-\frac{1}{2}\,\frac{(\delta a_1^S)^2}{\sigma_1^2}\right), \tag{3.50}$$

where

$$\sigma_1^2 = \left(U^{-1}\right)_{11}. \tag{3.51}$$

As shown in Appendices E and F, Eqs. (3.42), (3.44) and (3.50) also apply, under the same conditions (linear model and Gaussian noise on the data) to the probability for $\delta a_1 \equiv a_1^{\text{true}} - a_1^{(0)}$, where we remind the reader that $a_1^{(0)}$ is the fit parameter obtained from the actual data, and a_1^{true} is the exact value. In other words

the probability of the true value is given by

$$P(\vec{a}^{\,\text{true}}) \propto \exp\left(-\frac{1}{2}\Delta\chi^2\right), \tag{3.52}$$

where

$$\Delta\chi^2 \equiv \chi^2(\vec{a}^{\,\text{true}}) - \chi^2(\vec{a}^{\,(0)}), \tag{3.53}$$

in which we remind the reader that both values of χ^2 are evaluated from the single set of data available to us, $y_i^{(0)}$. Projecting onto a single parameter, as above, gives

$$P(a_1^{\text{true}}) \propto \exp\left(-\frac{1}{2}\,\frac{(\delta a_1)^2}{\sigma_1^2}\right), \tag{3.54}$$

so $\langle(\delta a_1)^2\rangle = \sigma_1^2 = \left(U^{-1}\right)_{11}$, in agreement with what we found earlier in Eq. (3.24). We emphasize that Eqs. (3.52) and (3.54) assume Gaussian noise on the data points, and that either the model is linear or, if non-linear, the range of uncertainty in the parameters is small enough that a description in terms of an effective linear model is satisfactory.

However we have done more than recover our earlier result, Eq. (3.24), by more complicated means since we have gained *additional* information. From the properties of a Gaussian distribution we now know that, from Eq. (3.54), the probability that a_α lies within one standard deviation σ_α of the value which minimizes χ^2 is 68 %, the probability of its being within two standard deviations is 95.5 %, and so on. Furthermore, from Eq. (3.52), we see that

> if a single fit parameter is one standard deviation away from its value at the minimum of χ^2 (the other fit parameters being varied to minimize χ^2), then $\Delta\chi^2 = 1$.

This last sentence, and the corresponding equations Eqs. (3.52) and (3.54), are not valid for a non-linear model if the uncertainties of the parameters extends outside the range where an effective linear model can be used. In this situation, to get confidence limits, one should do a bootstrap resampling of the data, as discussed in the next subsection. Even for a linear model one needs bootstrap resampling to get confidence limits if the noise on the data is non-Gaussian.

However, if one is not able to resample the data we argue that it is better to take the range where $\Delta\chi^2 \leq 1$ as an error bar for each parameter rather than the error bar determined from the curvature of χ^2 at the minimum, see Fig. 3.3. The left hand plot is for a linear model, for which the curve of $\Delta\chi^2$ against δa_1 is exactly a parabola, and the right hand plot is a sketch for a non-linear model, for which it is not a parabola though it has a quadratic variation about the minimum shown by the dashed curve. For the linear case, the values of δa_1 where $\Delta\chi^2 = 1$ are the *same* as the values $\pm\sigma_1$, where σ_1 is the standard error bar obtained from the *local* curvature in the vicinity of the minimum. However, for the non-linear case, the values of δa_1 where

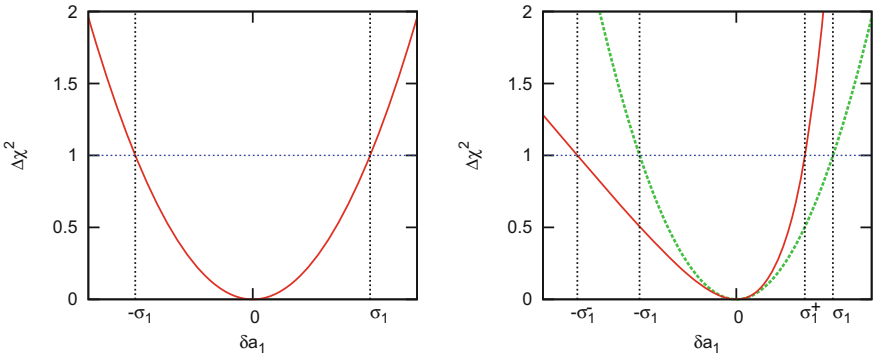

Fig. 3.3 *Left* The change in χ^2 as a fit parameter a_1 is varied away from the value that minimizes χ^2 for a *linear* model. The shape is a *parabola* for which $\Delta\chi^2 = 1$ when $\delta a = \pm\sigma_1$, where σ_1 is the error bar. *Right* The *solid curve* is a sketch of the change in χ^2 for a *non-linear* model. The curve is no longer a parabola and is not even symmetric. The *dashed curve* is a parabola which fits the solid curve at the minimum. The fitting program only has information about the *local* behavior at the minimum and so gives an error range $\pm\sigma_1$ at which the value of the parabola is 1. However, the parameter a_1 is clearly more tightly constrained on the plus side than on the minus side, so a better way to determine the error range is to look *globally* and locate the values of δa_1 where $\Delta\chi^2 = 1$. This gives an error bar σ_1^+ on the plus side, and a different error bar, σ_1^-, on the minus side, both of which are different from σ_1

$\Delta\chi^2 = 1$ are *different* from $\pm\sigma_1$, and indeed the values on the positive and negative sides, σ_1^+ and σ_1^-, are not equal. For the data Fig. 3.3, it is clear that the value of a_1 is more tightly constrained on the positive side than the negative side, and so it is better to give the error bars as $+\sigma_1^+$ and $-\sigma_1^-$, obtained from the range where $\Delta\chi^2 \leq 1$, rather the symmetric range $\pm\sigma_1$. While it is plausible that the range where $\Delta\chi^2 \leq 1$ is a reasonable estimate of the uncertainty in the fit parameter, one can not assign a precise confidence limit to it. If possible, error bars and a confidence limit should be obtained from an alternative approach, a bootstrap resampling of the data as discussed in the next section.

3.7 Confidence Limits by Resampling the Data

Each data point (x_i, y_i) with its error bar σ_i, comes from averaging over N_i values of "raw data" whose mean is y_i and whose standard deviation gives σ_i according to Eq. 2.15. If one has access to this raw data, one can do a bootstrap resampling of it to obtain:

- Confidence limits for a linear model if the noise is non-Gaussian.
- Both error bars and confidence limits for a non-linear model in which the range of parameter uncertainty extends outside the region where an effective linear model is applicable.

As discussed in Sect. 2.2.3, one generates bootstrap data sets in which the data points have values $y^B_{i,\alpha}$, where α runs from 1 to $N_{\rm boot}$, the number of bootstrap data sets. The distribution of the $y^B_{i,\alpha}$ for a given i has a standard deviation equal to the estimate of standard deviation on the mean of the actual data set, i.e. σ_i, see Eq. 2.67 (replacing the factor of $\sqrt{N/(N-1)}$ by unity which is valid since N is large in practice). Hence, if we fit each of the $N_{\rm boot}$ bootstrap data sets, the scatter of the fitted parameter values will be a measure of the uncertainty in the values from the single *actual* dataset. Forming a histogram of the values of a single parameter we can obtain a confidence interval within which 68 %, say, of the bootstrap datasets lie (16 % missing on either side) and interpret this range as a 68 % confidence limit for the actual parameter value. The justification for this interpretation has been discussed in the statistics literature, see e.g. the references in Ref. [1], but I'm not able to go into the details here. The method can be applied to both linear and non-linear models, and does not assume Gaussian noise.

Unfortunately, use of the bootstrap procedure to get error bars in fits to non-linear models does not yet seem to be a standard procedure in the statistical physics community.

If one does not have access to the "raw" data, but is confident that the noise is close to Gaussian, another possibility, which is useful for non-linear models, is is to generate *simulated* data sets, assuming Gaussian noise on the y_i values with standard deviation given by the error bars σ_i. Each simulated dataset is fitted and the distribution of fitted parameters is determined. This corresponds to the analytical approach in Appendix E but without the assumption that the model can be represented by an effective linear one over of the needed parameter range.

3.8 A Tale of Two Probabilities. When Can One Rule Out a Fit?

In this section we assume that the noise on the data is Gaussian. We have, so far, considered two different probabilities. Firstly, as discussed in Appendix C, the value of χ^2 is typically in the range $N_{\rm DOF} \pm \sqrt{2N_{\rm DOF}}$. The quality of fit parameter Q is the probability that, *given the fit*, the data could have this value of χ^2 or greater, and is given mathematically by Eq. (3.25). It varies from unity when $\chi^2 \ll N_{\rm DOF} - \sqrt{2N_{\rm DOF}}$ to zero when $\chi^2 \gg N_{\rm DOF} + \sqrt{2N_{\rm DOF}}$. In other words Q only changes substantially if χ^2 changes by an amount of order $\sqrt{N_{\rm DOF}}$. We emphasize that Q is the probability of the data given the fit.

Secondly, in the context of error bars and confidence limits, we have discussed, in Eqs. (3.52) and (3.54), the probability that a fit parameter, a_1 say, takes a certain value relative to the optimal one. Equation (3.52) tells us that the relative probability of two fits changes substantially when χ^2 varies by unity. Note that Eqs. (3.52) and (3.54) refer to the relative probabilities of two fits, given the data.

At first it seems curious that Q needs a much bigger change in χ^2 to change significantly than does the relative probability of two fits ($\sqrt{N_{\mathrm{DOF}}}$ rather than 1). While there is no mathematical inconsistency, since the two probabilities refer to different situations (one is the probability of the data given the fit and the other is the relative probability of two fits given the data), it is useful to understand this difference intuitively.

We take, as an example, a problem where we want to know whether the data can be modeled by a straight line, or whether a quadratic term needs to be included as well. A set of data is shown in Fig. 3.4.

Looking at the left panel one sees that the data more or less agrees with the straight-line fit ($Q = 0.124$). However, one also sees systematic trends: the data is too high for small x and for high x, and too low for intermediate x. The probability that this trend would occur by chance is very low. Chi-squared just sums up the contributions from each data point and is insensitive to any systematic trend in the deviation of the data from the fit. Hence the value of χ^2, in itself, does not tell us that this data is unlikely to be represented by a straight line. It is only when we add another parameter in the fit which corresponds to those correlations, that we realize the straight-line model is relatively very unlikely. In this case, the extra parameter is the coefficient of x^2, and the resulting parabolic fit is shown in the right figure.

The qualitative comments in the last paragraph are made more precise by the parameters of the fits. The straight-line fit gives $a_0 = 0.59 \pm 0.26$, $a_1 = 2.003 \pm 0.022$ with $\chi^2 = 48.2$, $Q = 0.124$, whereas the parabolic fit gives $a_0 = 2.04 \pm 0.40$, $a_1 = 1.588 \pm 0.090$, $a_2 = 0.0203 \pm 0.00425$ with $\chi^2 = 25.5$, $Q = 0.924$. The actual parameters which were used to generate the data are $a_0 = 2$, $a_1 = 1.6$, $a_2 = 0.02$,

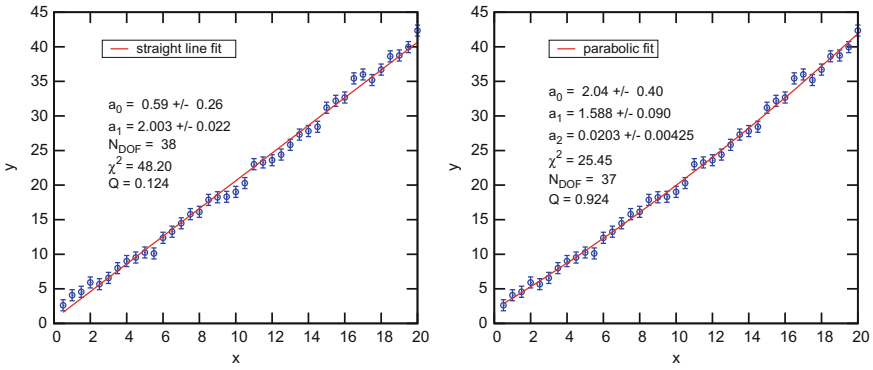

Fig. 3.4 *Left* A straight-line fit to a data set. The value of Q is reasonable. However, one notices that the data is systematically above the fit for small x and for large x while it is below the fit for intermediate x. This is unlikely to happen by random chance. This remark is made more precise in the right figure. *Right* A parabolic fit to the same data set. The value of Q is larger than for the straight-line fit, but the main result is that the coefficient of the quadratic term is about 5 σ away from zero, showing that the straight-line fit in the *left panel* is much less likely than the parabolic fit

and there is Gaussian noise with standard deviation equal to 0.8. Hence the parameters of the quadratic fit are correct, but those of the linear fit are not. Although the quality of fit factor Q for the straight-line fit is reasonable, the quadratic fit strongly excludes having the fit parameter a_2 equal to zero, since zero is 4.78 standard deviations away from the best value. As shown in Eq. (A3), the probability of a 4.78-sigma deviation or greater for a Gaussian distribution is $\mathrm{erfc}(4.78/\sqrt{2}) \simeq 1.78 \times 10^{-6}$, which is tiny. Thus a careful analysis correctly concludes that the straight-line fit is unlikely to be correct.

From the figures we see that difference in χ^2 between the quadratic fit and the straight-line fit (in which we force $a_2 = 0$) is 22.8, which should equal $(0.0203/0.00425)^2$ according to Eqs. (3.49) and (3.24), and indeed it does. This provides a useful check on the parameters computed by the fit program (gnuplot in this case).

The moral of this tale is that a reasonable value of Q does not, in itself, ensure that you have the *right* model. Another model might be more probable. To quote from Ch. 14 of Numerical Recipes [1],[4]

> If a statistic falls in a *reasonable* part of the distribution, you must not make the mistake of concluding that the (null) hypothesis is "verified" or "proved". That is the curse of statistics. It can never prove things, only disprove them!

We will discuss more the question of determining the right model (model selection) in the next section.

3.9 Model Selection (i.e. How to Avoid Over-Fitting): Maximum Likelihood Versus Bayes

Apart from the last subsection we have *assumed* that the model is given, and our goal is to obtained the best fit parameters of that model. In the last subsection we started a discussion of how to compare *different* models, which is necessary if the correct model is not known.

For the data shown in Fig. 3.4 we showed that the value of χ^2 for the parabolic fit ($M = 3$) is much smaller than that of the straight-line fit ($M = 2$), and we argued that it is therefore preferred. This is correct since the data was indeed obtained from a parabolic function (plus noise). However, suppose we consider higher order fits. Since the fit is obtained by minimizing χ^2, it is clear that χ^2 can only decrease with the number of fit parameters M. (Here we consider polynomial fits for which the order of the polynomial is $M - 1$.) This is illustrated by the left panel line in Fig. 3.5, which shows χ^2 against the number of fit parameters M for $2 \leq M \leq 12$ for the data in Fig. 3.4.

Based on the plot of χ^2 in the left panel Fig. 3.5, should we say that a 9th order polynomial ($M = 10$), say, is to be preferred to a quadratic fit because it has a

[4] I find the use of the word "null" in the quote to be confusing. It is, however, common usage in the statistics literature. The brackets round it are mine.

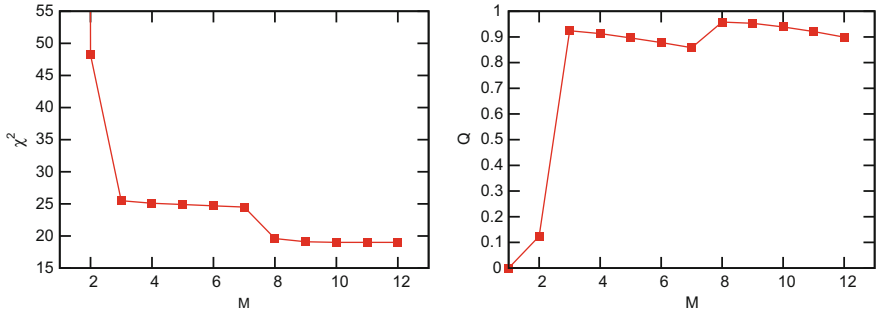

Fig. 3.5 *Left* A plot of χ^2 against the number of fit parameters for polynomial fits to the data in Fig. 3.4. (*Note* the order of the polynomial is $M-1$.) The values of χ^2 decrease monotonically with M, as expected. There is a rapid drop in going from $M=2$ to $M=3$, but then a much more gradual decrease. This is consistent with the fact that the data was generated from a parabolic function plus Gaussian noise. The reason for the subsidiary drop in going from $M=7$ to 8 is unclear. Perhaps, by coincidence, the noise in the data has generated a noticeable x^7 component. *Right* The goodness of fit parameter Q, see Ref. [1] and Eq. (3.25), for different values of M. Although the value of χ^2 monotonically decreases with increasing M as shown in the *left panel*, the goodness of fit parameter involves χ^2 *per degree of freedom*, where $N_{\rm DOF}=N-M$, and so there is a penalty be paid as the number of fit parameters increases, since this decreases the number of degrees of freedom. As a result, Q has a peak at $M=3$, a parabola, (which indeeds corresponds to the function which generated the data) and then (slowly) decreases. Based on this data, one would come to the (correct) conclusion the data should be fitted to a parabola. For $M>3$, which is the region of over-fitting, the variation of Q with M is non-monotonic, at least in this case

smaller χ^2? Intuitively we would say "no" because we feel that a fit with a smaller number of parameters is more likely to be correct than a fit with a larger number, if the quality of the fits is very similar. Clearly if the number of fit parameters is equal to the number of data points, 40 here, then the fit will go perfectly through the points. However, in practice this can lead to large oscillations between the points, in order to force the curve to exactly fit the data. This unphysical result is called "*over-fitting*". In addition, the fit parameters become very large when over-fitting.

The problem of determining the correct model (the order of polynomial in the present example) is called "*model selection*".

We shall apply different approaches to the model selection problem to two sets of data: that in Fig. 3.4 and also the smaller data set with just ten points shown in Fig. 3.9. Table 3.1 shows the parameters for all possible polynomial fits to the data in Fig. 3.6.

We will discuss two approaches to the problem of model selection and over-fitting. These correspond to the the two basic approaches to statistics, the one more familiar to most physicists is called "frequentist", and the other is called "Bayesian". The frequentist approach is called "maximum likelihood" in the context of fitting, and is just the least-square method described in the earlier part of these notes. We explain in Sect. 3.9.1 below why last-squares is called maximum likelihood and how one can approach model selection within this approach. However, a fully system-

Table 3.1 Fit parameters a_α for polynomial fits to the data shown in Fig. 3.9 for different numbers of fit parameters M

	$M = 2$	$M = 3$	$M = 4$	$M = 7$	$M = 10$	exact
a_0	−0.826	0.650	0.148	−2.206	−31.02	1
a_1	−1.303	−9.42	−4.52	44.14	740.3	−10
a_2		8.12	−3.57	−364.9	−6329.6	8
a_3			7.79	1304.1	24157.5	
a_4				−2407.2	−35884.6	
a_5				2210.4	−29162.7	
a_6				−789.6	185757	
a_7					−275351	
a_8					183538	
a_9					−47449.3	

For the larger values of M the fit parameters are very large indicating severe over-fitting
Note The order of the polynomial is $M - 1$

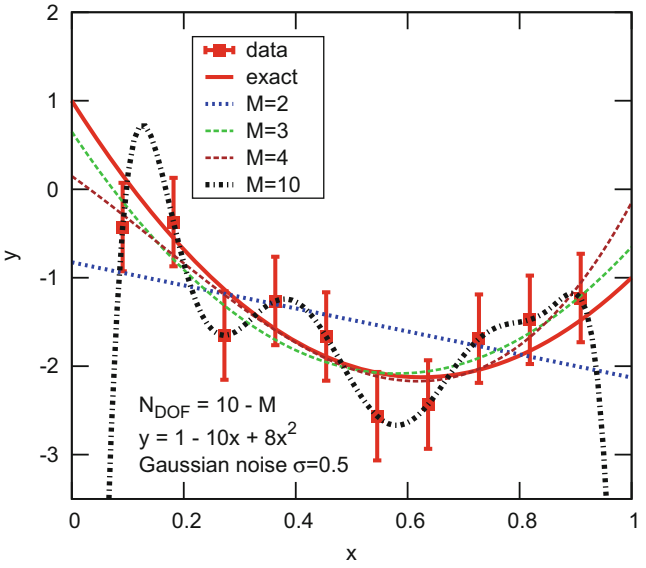

Fig. 3.6 The data points are obtained by adding Gaussian noise with standard deviation 0.5 to the parabolic function $y = 1 - 10x + 8x^2$ (shown by the solid line). Also shown are polynomial fits with $M = 2, 3, 4$ and 10 parameters. The values of χ^2 for all values of M from 2 through 10 are shown in the left panel of Fig. 3.7. For $M = 10$ $(= N)$ the fit goes perfectly through the points. However, unphysical oscillations are seen in the fitting function for $M = 10$, clear evidence of over fitting. Furthermore the fit parameters get very large when M increases, as shown in Table 3.1

atic, maximum likelihood method for model selection does not appear to have been developed. The alternative Bayesian approach is described in Sect. 3.9.2. It has been argued elsewhere, e.g. [2], to provide provides a systematic, robust method of model selection which avoids over-fitting. However, we shall find some reservations about this approach when we apply it to actual data.

3.9.1 Maximum Likelihood

First we will show that the "frequentist" approach to statistics, called maximum likelihood in the context of fitting, corresponds to the least-squares approach discussed up to now in these notes. In the frequentist approach to statistics, we determine the probability of a particular event in a random process by repeating the process many times and dividing the number of times the specified event occurred by the total number of events. In the limit when the number of events tends to infinity this ratio tends to the actual probability. This is to be distinguished from the "Bayesian" approach, discussed in the next subsection, in which, in addition to the data, we include *extra* information in the form of a "prior" distribution for some parameters.

In curve fitting, we only have one set of data, but, as we have repeatedly emphasized in these notes, we obtain unbiased estimates of fit parameters and their uncertainties by a thought experiment in which we consider the results that *would* be obtained if we could obtain many data sets. In particular, the error bars in the fit parameters come precisely from the scatter that would be obtained by repeating the fit on many data sets, *assuming that the fit from the one set of data that we actually have is correct*. If, as we shall do in the rest of this section, we assume Gaussian noise on the data, the error bars determine the whole probability distribution of the data.

Thus the frequentist approach gives the probability of the data given the fit. This seems a bit strange. We would really like to know what is the probability that the set of fitted parameters in correct. However, as stated by Numerical Recipes [1] (implicity assuming the frequentist approach)

> . . . there is no statistical universe of models from which the parameters are drawn. There is just one model, the correct one, and a statistical universe of data sets that are drawn from it!

Thus, as mentioned earlier in these notes, in the frequentist approach, we take the probability of the data given the fit, as a measure of whether the fit parameters are likely to be right. If we assume that the fit is correct, and the noise on the data is Gaussian with variance σ_i^2, then this probability is

$$P(\{y\}) = \frac{1}{(2\pi)^{N/2}\left(\prod_{i=1}^{N}\sigma_i\right)} \exp\left[-\frac{1}{2}\sum_{i=1}^{N}\left(\frac{y_i - \sum_{\alpha} a_\alpha X_\alpha(x_i)}{\sigma_i}\right)^2\right]. \qquad (3.55)$$

For ease of notation, we have written this for a linear model, but the generalization to a non-linear model is obvious.

A sensible approach, then, is to maximize this probability, which is known as "maximum likelihood" method. It is equivalent to maximizing the expression in the exponential in Eq. (3.55), which is just $(-1/2)$ times χ^2, and hence to minimizing χ^2. *Thus, in the context of fitting, maximum likelihood is equivalent to the standard least-squares approach.*

We can use the result that least-squares is equivalent to maximum likelihood to formulate the least squares method for correlated data. This is described in Appendix G.

Now we now that least-squares corresponds to maximum likelihood (frequentist) approach to statistics, we can use it to tackle the problem of overfitting.

The intuition that the simplest model which fits the data is to be preferred can be inferred from the results for χ^2 in in Fig. 3.5 since χ^2 decreases *considerably* in going from $M = 2$ (a straight line) to 3 (a parabola), but then decreases only *slightly* for larger M. We might therefore conclude "*by eye*" that underlying model is a parabola. But how can we select the right model in a systematic manner?

Intuitively, we would like to add a penalty to χ^2 which increases with M and then look at the minimum of the resulting quantity as a function of M. One way to do this is to consider the quality of fit factor Q [1], given in Eq. (3.25) since this involves the value of χ^2 *per degree of freedom*, so if χ^2 does not decrease by much on increasing M by one, the value of χ^2 per degree of freedom will increase, so Q can decrease. This is shown in the right part of Fig. 3.5 where a peak in Q is seen at $M = 3$ (the correct value). At larger values of M, in the region of over-fitting, the variation of Q with M is non-monotonic, at least in this case.

We have also analyzed the data in Fig. 3.9 which has just 10 data points. The data are from a quadratic function, $f(x) = 1 - 10x + 8x^2$, plus Gaussian noises with standard deviation 0.5. The value of χ^2, shown in the left part of Fig. 3.7, decreases

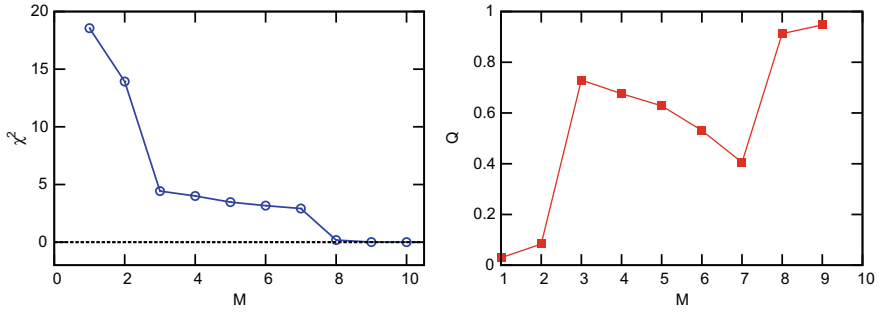

Fig. 3.7 *Left* A plot of χ^2 versus M for the data in Fig. 3.9. This shows a rapid drop in going from $M = 2$ to 3 but then continues to decrease on further increasing M. However, the data is being over-fitted in this region. *Right* A plot of the confidence of fit factor Q for the data shown in Fig. 3.9. This shows a peak at $M = 3$, followed by a gradual decrease. (For still larger values of M, Q increases again, presumably connected with the fact that the fit is perfect for $M = N$)

considerably when M in increases from 2 and 3, but decreases much more slowly for the next few values of M. Of course, $\chi^2 = 0$ for $M = N$ $(= 10)$. Fit parameters for different values of M are shown in Table 3.1 and the corresponding values of Q are shown in the right part of Fig. 3.7. A clear peak is seen for $M = 3$, the parabolic fit, indicating, again correctly, that a parabola is the right function to model the data.

3.9.2 A Bayesian Approach

The Bayesian approach heavily uses "conditional" probability distributions, so we start by explaining these. Consider two random variables x and y. We write $P(X, Y)$ as the probability that x has the value X and y has the value Y, and also write $P(X)$ as the probability that x has value X, irrespective of the value of y. Clearly $P(X)$ can be obtained by *summing* $P(X, Y)$ over Y, i.e.

$$P(X) = \sum_Y P(X, Y). \tag{3.56}$$

We also need *conditional* probabilities, such as $P(X|Y)$, which is the probability that x has value X *given* that y has value Y, and $P(Y|X)$ which is the probability that y has value Y given that x has value X.

We can relate $P(X, Y)$ to conditional probabilities in two different ways. Firstly, we can determine the probability that x has value X, and multiply this by the conditional probability of Y given X. Alternatively, we can do the same thing with X and Y interchanged. Thus we have

$$P(X, Y) = P(Y|X)\, P(X) = P(X|Y) P(Y). \tag{3.57}$$

If we divide by $P(X)$ we get

$$P(Y|X) = \frac{P(X|Y)\, P(Y)}{P(X)}, \tag{3.58}$$

which is known as *Bayes' theorem* and plays a central role in the Bayesian approach to statistics. From Eqs. (3.56) and (3.57) we have

$$P(X) = \sum_Y P(X|Y)\, P(Y), \tag{3.59}$$

so the denominator in Eq. (3.58) is the normalizing constant which ensures that the sum of the conditional probabilities on the left hand side over all values of Y is equal to one. We can therefore also write Bayes' theorem in the form

$$P(Y|X) = \frac{P(X|Y)\,P(Y)}{\sum_{Y'} P(X|Y')\,P(Y')}. \tag{3.60}$$

In the Bayesian approach to statistics, Eq. (3.58) is interpreted in the following way. $P(Y)$ is taken to be some knowledge about Y which has been acquired before the statistical analysis is performed, and is called the "*prior*". Data is then used to determine $P(X|Y)$, the probability of X given this value of Y. Bayes' theorem then gives $P(Y|X)$, the probability of Y now that the statistical information about X has been included, and is called the "*posterior*" distribution. If, later on, more information is available, say that a third variable z has value Z, then $P(Y|X)$ can be taken as the prior, and the posterior distribution for y, including information about *both* x and z is given according to Bayes' theorem by

$$P(Y|X, Z) = \frac{P(Z|Y, X)\,P(Y|X)}{P(Z|X)}, \tag{3.61}$$

where $P(Y|X, Z)$ is the conditional probability for Y given both that $x = X$ and $z = Z$, and the denominator is equal to $\sum_Y P(Z|Y, X)\,P(Y|X)$. If Eq. (3.61) looks confusing, note that X is fixed throughout and could be omitted from the notation, in which case Eq. (3.61) just Bayes' theorem, Eq. (3.58), for Y and Z.

Bayesian statistics is sometimes regarded with suspicion on the grounds that the prior distributions used seem either to be subjective or to be determined by mathematical convenience rather than physical intuition. Nonetheless, Bayesian methods can be very useful in many situations, and the problem of model selection in fitting is claimed [2] to be one of them. Recall from Fig. 3.9 and Table 3.1 that the problem with maximum likelihood methods is that they prefer complicated models which over-fit the data, give oscillatory behavior between the data points, and produce unphysically large values for the fit parameters.

> Thus, we have the notion that simpler models are better, and this is precisely the sort of additional information that is included in Bayesian analysis in the form of a prior. Hence a Bayesian analysis is natural for the model selection problem.

Our goal is to determine the best "model", $\mathcal{M}$, to fit the data. Specifying a model requires specifying a functional form and the number of fit parameters M. Since we stick to polynomials here we can indicate a model simply by specifying the number of parameters M. We want to compute the relative probabilities of different values of M, given the data, i.e. $P(M|D)$, where the symbol D indicates our set of data.

To do this we need some additional information to prevent over-fitting. We therefore introduce a prior for the fit parameters, and a mathematically convenient choice is a Gaussian,

$$P(F|\gamma) = \left(\frac{\gamma}{2\pi}\right)^{M/2} \exp\left[-\frac{\gamma}{2}\sum_{\alpha=1}^{M} a_\alpha^2\right], \tag{3.62}$$

in which we write F to symbolically indicate the fitting parameters. The quantity γ is called a "*hyperparameter*" because it controls the parameters of the fit. Equation (3.62) gives the probability of the fit parameters given a particular value for the hyperparameter. For simplicity we have taken the same value of γ for each of the fit parameters. Clearly Eq. (3.62) can be criticized for the reason mentioned above, namely that it is chosen for mathematical convenience rather than any real prior information. Nonetheless, we shall see that it serves the purpose of penalizing over-fitting.

The probability of M given the data is obtained by summing over all possible values of γ so we have

$$P(M|D) = \sum_{\gamma} P(\gamma|D) = \sum_{\gamma} \frac{P(D|\gamma)P(\gamma)}{P(D)}, \tag{3.63}$$

where we used Bayes' equation to get the last equality. The denominator is a constant independent of the model or fit parameters and will be ignored from now on. What do we take for $P(\gamma)$? Since we have no information on it, one might think of setting it to a constant. However, a constant distribution between 0 and ∞ is not normalizable, so we have to put in bounds. This is tricky because we have no idea of γ is very large or very small or in between. One does better by taking $P(\log \gamma)$ to be constant because, even though one still needs bounds, one can cover a huge range of magnitudes. Hence we take $P(\gamma) \propto 1/\gamma$. We expect that $P(D|\gamma)$ will be fairly sharply peaked at some value $\hat{\gamma}$ and that this value will dominate the sum in Eq. (3.63). Hence we have

$$P(M|D) \propto \frac{P(D|\hat{\gamma})}{\hat{\gamma}}, \tag{3.64}$$

and so the ratio of probabilities for two different values of M is

$$\frac{P(M_1|D)}{P(M_2|D)} = \frac{P_{M_1}(D|\hat{\gamma}_{M_1})}{P_{M_2}(D|\hat{\gamma}_{M_2})} \frac{\hat{\gamma}_{M_2}}{\hat{\gamma}_{M_1}}, \tag{3.65}$$

where we temporarily indicate that the distribution $P(D|\gamma)$ depends on the number of fit parameters M. (The ratio of the γ values in Eq. (3.65) comes from our choice of a uniform distribution for $P(\ln \gamma)$ and is often ignored.)

Hence our goal is to determine $P(D|\gamma)$ and maximize it with respect to γ. Now $P(D|\gamma)$ is obtained by *summing* over all possible values for the fit parameters, i.e.

$$P(D|\gamma) = \sum_{F} P(D|F)\, P(F|\gamma). \tag{3.66}$$

The probability of the data given the fit, $P(D|F)$, is given by the maximum likelihood result in Eq. (3.55), namely

$$P(D|F) = \frac{1}{(2\pi)^{N/2}\left(\prod_{i=1}^{N}\sigma_i\right)} \exp\left[-\frac{1}{2}\sum_{i=1}^{N}\left(\frac{y_i - \sum_\alpha a_\alpha X_\alpha(x_i)}{\sigma_i}\right)^2\right] \quad (3.67)$$

Multiplying this by Eq. (3.62) and substituting into Eq. (3.66) we get

$$P(D|\gamma) = \left(\frac{\gamma}{2\pi}\right)^{M/2} \frac{1}{(2\pi)^{N/2}\left(\prod_{i=1}^{N}\sigma_i\right)} \prod_{\alpha=1}^{M}\left(\int_{-\infty}^{\infty} da_\alpha\right) \exp\left[-\frac{1}{2}E_0(\{a_\alpha\})\right], \quad (3.68)$$

where the cost function, $E_0(\{a_\alpha\})$, is given by

$$E_0(\{a_\alpha\}) = \sum_{i=1}^{N}\left(\frac{y_i - \sum_{\alpha=1}^{M} a_\alpha X_\alpha(x_i)}{\sigma_i}\right)^2 + \gamma\sum_\alpha a_\alpha^2. \quad (3.69)$$

The first term in $E_0(\{a_\alpha\})$ is just χ^2 and the second term has the effect of a "*regularizer*" which penalizes fits with large parameter values. Over-fitting leads to very large parameter values, see Table 3.1 for an example, so the second term in Eq. (3.69) acts as to suppress over-fitting, as desired.

We need to find a minimum of Eq. (3.69) with respect to the fit parameters (equivalent to the maximum of the exponential in Eq. (3.68)), which is straightforward because it is quadratic function of the parameters (a mathematical advantage of the Gaussian prior). The solution for the parameters is still given by Eq. (3.12), with the v_α still given by Eq. (3.33), and the only change is that $U_{\alpha\beta}$ now has an extra term involving γ,

$$U_{\alpha\beta} = \sum_{i=1}^{N}\frac{X_\alpha(x_i)\,X_\beta(x_i)}{\sigma_i^2} + \gamma\,\delta_{\alpha\beta}. \quad (3.70)$$

We will write the values of fit parameters at the minimum of E_0 as $\hat{a}_\alpha$.

From Eq. (3.12) these are given by

$$\hat{a}_\alpha = \sum_{\beta=1}^{M}\left(U^{-1}\right)_{\alpha\beta} v_\beta, \quad (3.71)$$

where now U is given by Eq. (3.70) and v by Eq. (3.33). These maximize the exponential in Eq. (3.68) and are called the *maximum posterior* (or MAP) values of the fit parameters.

The expectation value of a_α using the Gaussian weight in Eq. (3.68) is just the optimal value $\hat{a}_\alpha$. To perform the integrals in Eq. (3.68) we expand the a_α about $\hat{a}_\alpha$ and perform the resulting Gaussian integrals by completing the square. The result is

$$P(D|\gamma) = \frac{\gamma^{M/2}}{(\det U)^{1/2}} \frac{1}{(2\pi)^{N/2}\left(\prod_{i=1}^{N}\sigma_i\right)} \exp\left[-\frac{1}{2}E_0(\{\hat{a}_\alpha\})\right], \tag{3.72}$$

see the discussion below Eq. (G.3) for an explanation of where the determinant comes from. Because γ in Eq. (3.70) only appears proportional to the identity matrix, the eigenvectors of U are independent of γ and so the eigenvalues can be written as $\lambda_I = \lambda_I^{(0)} + \gamma$, where $\lambda_I^{(0)}$ is the eigenvalue in the unregularized case with $\gamma = 0$. We use an uppercase Roman letter to label one of the M eigenvalues. Hence, from Eq. (3.72), the log of the probability of the data given the hyperparameter γ is given by

$$\ln P(D|\gamma) = -\frac{1}{2}\overline{E}(\{\hat{a}_\alpha\}, \gamma), \tag{3.73}$$

where the cost function, $\overline{E}(\{\hat{a}_\alpha\}, \gamma)$, which we have to minimize with respect to γ, is given by

$$\overline{E}(\{\hat{a}_\alpha\}, \gamma) = E_0(\{\hat{a}_\alpha\}) + \sum_{I=1}^{M} \ln\left(\frac{\lambda_I^{(0)} + \gamma}{\gamma}\right) + \sum_{i=1}^{N} \ln(2\pi\sigma_i^2) \tag{3.74}$$

$$= \sum_{i=1}^{N}\left(\frac{y_i - \sum_{\alpha=1}^{M}\hat{a}_\alpha X_\alpha(x_i)}{\sigma_i}\right)^2 + \gamma\sum_{\alpha=1}^{M}\hat{a}_\alpha^2 + \sum_{I=1}^{M}\ln\left(\frac{\lambda_I^{(0)} + \gamma}{\gamma}\right) + \sum_{i=1}^{N}\ln(2\pi\sigma_i^2), \tag{3.75}$$

where we have written $\det U = \prod_I(\lambda_I^{(0)} + \gamma)$. We should mention that the eigenvalues $\lambda_I^{(0)}$ are *independent* of both the parameters $\hat{a}_\alpha$ and also γ since they come from diagonalizing the matrix U in Eq. (3.70) without the γ term. In the machine learning literature $P(D|\gamma) \equiv \exp(-\overline{E}(\{\hat{a}_\alpha\}, \gamma)/2)$ is called the *evidence function* [2].

We now discuss each of the terms in the cost function E in Eq. (3.75).

1. The first term is just χ^2.
2. Intuitively we want to add to χ a term which increases with M to penalize overfitting. This role is played by the the second and third terms in Eq. (3.75), which are each the sum of M factors. The second term is the regularizer which was discussed after Eq. (3.69).
3. The third term is the most interesting. From Eq. (3.62) we see that $1/\gamma$ is the variance of the prior distribution for each of the a_α. Further, $1/(\lambda_I^{(0)} + \gamma)$, being the Ith eigenvalue of the covariance matrix, is the variance of the posterior distribution of the linear combination of fit parameters corresponding to the Ith eigenvector. We'll call this σ_I^2. Since we chose the same value of γ for each of the fit parameters, the variance of the prior distribution for eigenvector I is also $1/\gamma$ for all I. We'll denote this by $\left(\sigma_I^{(P)}\right)^2$. We therefore use the following definitions,

$$\sigma_I^{(P)} = \frac{1}{\gamma^{1/2}}, \qquad \sigma_I = \frac{1}{\left(\gamma + \lambda_I^{(0)}\right)^{1/2}}, \qquad \sigma_I^{(0)} = \frac{1}{\left(\lambda_I^{(0)}\right)^{1/2}}, \quad (I = 1, 2, \ldots, M), \tag{3.76}$$

in which we also define $\sigma_I^{(0)}$ to be the standard deviation in the estimate of the Ith eigenvector for $\gamma = 0$, see Fig. 3.8 for an illustration. Hence the third term in Eq. (3.75) can be written as

$$2\sum_{I=1}^{M} \ln\left(\frac{\sigma_I^{(P)}}{\sigma_I}\right). \tag{3.77}$$

Equivalently, from Eqs. (3.73) and (3.75) the (multiplicative) contribution of this term to $P(D|\gamma)$ is

$$P(D|\gamma) \propto \prod_{I=1}^{M}\left(\frac{\sigma_I}{\sigma_I^{(P)}}\right). \tag{3.78}$$

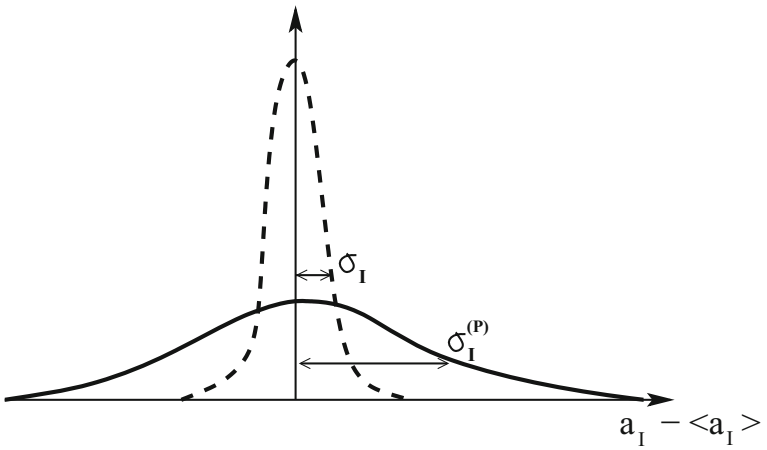

Fig. 3.8 The posterior (*dashed line*) and prior (*solid line*) distributions for a combination of fit parameters corresponding to the Ith eigenvector, a_I, of the covariance matrix. The width of the prior distribution, $\sigma_I^{(P)}$ is equal to $1/\gamma^{1/2}$ (for all I). The width of the posterior distribution, σ_I, is equal to $1/(\gamma + \lambda_I^{(0)})^{1/2}$ where $\lambda_I^{(0)}$ is the Ith eigenvalue of the matrix U (the inverse of the covariance matrix) in the absence of the hyperparameter γ. Parameters which are not much affected by the regularization due to γ have $\lambda_I^{(0)} \gg \gamma$, so $\sigma_I \ll \sigma_I^{(P)}$, while parameters which are affected by the regularization have $\lambda_I^{(0)} \ll \gamma$, which gives $\sigma_I \simeq \sigma_I^{(P)}$. The third term in Eq. (3.75) gives a penalty of $2\ln\left(\sigma_I^{(P)}/\sigma_I\right)$, see Eq. (3.77), to the cost function E for each parameter in the fit which is not significantly altered by regularization

As a reminder, the factor of $\sigma_I^{(P)}$ $(= \gamma^{-1/2})$ in Eq. (3.77) or (3.78) comes from the normalization of the prior distribution in Eq. (3.62), while the factor of σ_I, the width of the distribution of posterior distribution in fit parameter I, comes from integrating over the fit parameters in Eq. (3.68) (see also Eq. (3.66)). We see that the third term provides a penalty given by Eq. (3.77) for each parameter I which is not much affected by the regularization, i.e. $\gamma \ll \lambda_I^{(0)}$. In this way, we prevent the minimum of the cost function being at $\gamma = 0$, which would just give the maximum likelihood result.

4. The fourth and last term in Eq. (3.75) depends on the data but is independent of the fit parameters or the hyperparameter γ and so will be omitted in our subsequent discussion.

We emphasize that the maximum posterior (MAP) values of the fitting parameters, $\hat{a}_\alpha$, depend on the hyperparameter γ. However, when we minimize the cost function $\overline{E}(\{\hat{a}_\alpha\}, \gamma)$ in Eq. (3.75) with respect to γ we can neglect this dependence because the $\hat{a}_\alpha$ are precisely those values where $\partial E_0/\partial a_\alpha$ equal zero (and the difference between $\overline{E}$ and E_0 in Eq. (3.74) does not depend on the $\hat{a}_\alpha$). Hence minimizing Eq. (3.75) with respect to γ we find that the optimal choice for γ is given by the self-consistent solution of

$$\gamma \sum_{\alpha=1}^{M} \hat{a}_\alpha^2 = \sum_{I=1}^{M} \frac{\lambda_I^{(0)}}{\gamma + \lambda_I^{(0)}}. \tag{3.79}$$

We remind the reader that the fit parameters $\hat{a}_\alpha$ depend on γ, but the eigenvalues $\lambda_I^{(0)}$ are independent of the $\hat{a}_\alpha$ and γ. We denote by $\hat{\gamma}$ the value of γ which satisfies Eq. (3.79).

Note that we really want the probability for the model, and in Eq. (3.64) this is not given exactly by $P(\hat{\gamma}|D)$ but has an extra factor of $\hat{\gamma}^{-1}$. Furthermore, we will neglect the last term in Eq. (3.75) since it is independent of fit parameters or γ. Hence, for different values of M we will compare the values of

$$P(M|D) \propto \exp\left(-\frac{1}{2} E(\{\hat{a}_\alpha\}, \hat{\gamma})\right) \tag{3.80}$$

where

$$E(\{\hat{a}_\alpha\}, \hat{\gamma}) = \sum_{i=1}^{N} \left(\frac{y_i - \sum_{\alpha=1}^{M} \hat{a}_\alpha X_\alpha(x_i)}{\sigma_i} \right)^2 + \hat{\gamma} \sum_{\alpha=1}^{M} \hat{a}_\alpha^2 + \sum_{I=1}^{M} \ln\left(\frac{\lambda_I^{(0)} + \hat{\gamma}}{\hat{\gamma}} \right) + 2 \ln \hat{\gamma}. \tag{3.81}$$

Our Bayesian procedure to avoid overfitting is therefore as follows:

1. Choose a value for M and initial value for γ.
2. Determine the optimal fitting parameters $\hat{a}_\alpha$ from Eq. (3.71), and hence also the eigenvalues $\lambda_I^{(0)}$.

3. Substitute these values into the RHS of Eq. (3.79) to get a new value for γ. Go to 3.9.2 and iterate to convergence to determine the optimal value $\hat{\gamma}$.
4. Repeat for different values of M and compare the values of the cost function $E(\{\hat{a}_\alpha\}, \hat{\gamma})$ in Eq. (3.81). The optimal choice for M is at the minimum of E.

The Bayesian procedure we have described has some undesirable features. In the maximum likelihood approach, the value of χ^2 correctly remains the same if a translation or scaling of either the x or y axes takes place, and the fit itself is the same apart from the trivial change of variables. However, the cost function in the Bayesian analysis does not have these invariances. For example, if we just add a constant to all the data the fit should remain the same except that the constant is added to fit parameter a_0. However, the coupling of the regularization parameter γ to a_0 in the second term in Eq. (3.75) means that the fit is changed in a non-trivial way in the Bayesian analysis. Because of this lack of invariance, it may be useful to transform the data so that the x and y variables lie between -1 and 1, say, before doing the Bayesian analysis, and, indeed, we shall find it necessary to do this.

Even if all the axes are transformed in this way, we still find some strange features in the self-constency condition, Eq. (3.79), used to determine $\hat{\gamma}$. The $\lambda_I^{(0)}$ are the eigenvalues of the matrix U in Eq. (3.32). The elements of U get very large if the error bars on the data are small. We find that in this case the optimal value of γ does not get correspondingly large, with the result that $\lambda_I^{(0)} \gg \hat{\gamma}$ even for several parameters I which are not well determined by the data. This disagrees with the interpretation in Ref. [2] that those fit parameters I with $\lambda_I^{(0)} \gg \hat{\gamma}$ *are* those which *are* well determined by the data.

In Fig. 3.9 we plot the optimized cost function $E(\{\hat{a}_\alpha\}, \hat{\gamma})$ given by Eq. (3.81), for the two different data sets, as a function of M. Recall that both sets were determined from parabolas plus noise, and so the optimal choice of M should be 3. This is correctly reproduced for the data in Fig. 3.4 but not for the data in Fig. 3.9.

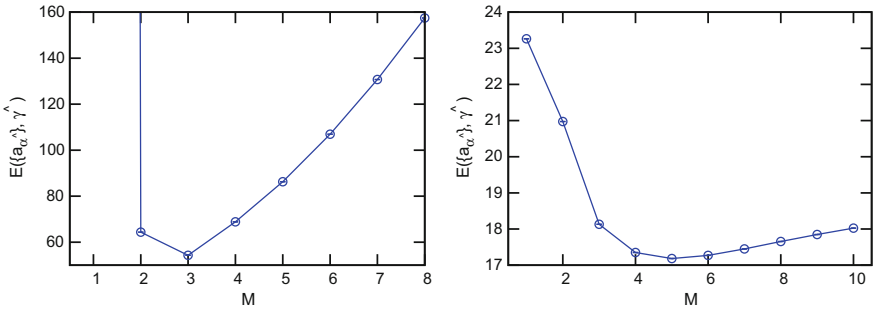

Fig. 3.9 *Left* The optimized cost function $E(\{\hat{a}_\alpha\}, \hat{\gamma})$ in Eq. (3.81) plotted against M for the data in Fig. 3.4. *Right* The same quantity but for the data in Fig. 3.9. For both data sets the optimal choice for M (where E is a minimum) should equal three. Hence this method works for the data in Fig. 3.4 but not for that in Fig. 3.9

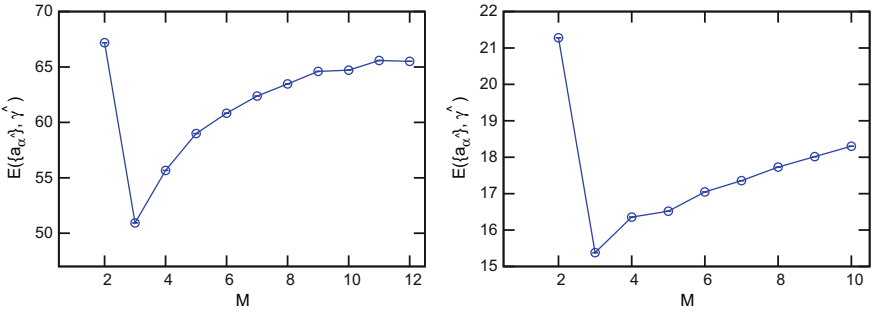

Fig. 3.10 *Left* The optimized cost function $E(\{\hat{a}_\alpha\}, \hat{\gamma})$ in Eq. (3.81) is plotted versus M when the data in Fig. 3.4 is scaled so that the x and y variables lie between -1 and 1. *Right* The same quantity but for the data in Fig. 3.9. Unlike the unscaled analysis in Fig. 3.9, this correctly gives a minimum at $M = 3$ for both data sets

From the above discussion we suspect that the problem with the data in Fig. 3.9 is due to lack of invariance of the method to shift and scaling of the axes. We have therefore performed the same analysis but with the x and y variables scaled to lie between -1 and 1 and show the results in Fig. 3.10. Now, the minimum is correctly found at $M = 3$ for both data sets.

3.9.3 Conclusions for Model Selection

In the maximum likelihood approach one computes the goodness of fit factor Q and looks for a peak as a number of fit parameters, M, see the right hand panel in Figs. 3.5 and 3.7. Although χ^2 monotonically deceases with increasing M, Q depends on χ^2 per degree of freedom and the latter decreases with increasing M. Hence there is a penalty on increasing M which can only be compensated for if there is a *substantial* decrease in χ^2. If M is too large, which is the over-fitting regime, χ^2 only decreases slightly with M and so Q decreases. (Eventually Q may increase again since the function fits the data perfectly for $M = N$.)

In the Bayesian approach, one applies a regularization parameter γ and determines its optimal value. From this a cost function function is found whose minimum value is the optimal choice for M. This worked "off the shelf" for one set of data studied, see Fig. 3.9, but to have it work for both sets of data it was necessary to scale the x and y coordinates to the range -1 to 1, see Fig. 3.10. The Bayesian approach is more complicated than maximum likelihood method and evidently has to be applied with care. It can be generalized, for example, by allowing different hyperparameters for each fit variable, but at a substantial *additional* complexity.

References

1. W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical Recipes in C*, 2nd edn. (Cambridge University Press, Cambridge, 1992)
2. C.M. Bishop, *Pattern Recognition and Machine Learning* (Springer, New York, 2006)

Appendix A
Central Limit Theorem

In this appendix we give a proof of the central limit theorem.

We assume a distribution that falls off sufficiently fast at $\pm\infty$ that the mean and variance are finite. This *excludes*, for example, the Lorentzian distribution:

$$P_{\text{Lor}} = \frac{1}{\pi}\frac{1}{1+x^2}. \tag{A.1}$$

A common distribution which *does* have a finite mean and variance is the Gaussian distribution,

$$P_{\text{Gauss}} = \frac{1}{\sqrt{2\pi}\,\sigma}\exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]. \tag{A.2}$$

Using standard results for Gaussian integrals you should be able to show that the distribution is normalized and that the mean and standard deviation are μ and σ respectively. We note that the probability that that a Gaussian random variable is more than $c\,\sigma$, where c is a constant, away from the mean is given by

$$\begin{aligned} P(|x-\mu| > c\,\sigma) &= \frac{2}{\sqrt{2\pi}\,\sigma}\int_{\mu+c\sigma}^{\infty}\exp\left[-(x-\mu)^2/(2\sigma^2)\right], \\ &= \frac{2}{\sqrt{\pi}}\int_{c/\sqrt{2}}^{\infty} e^{-t^2}\,dt \\ &= \text{erfc}(c/\sqrt{2}), \end{aligned} \tag{A.3}$$

where erfc is the complementary error function [1].

Consider a distribution, *not necessarily Gaussian*, with a finite mean and distribution. We pick N independent and identically distributed random numbers x_i from such a distribution and form the sum

P. Young, *Everything You Wanted to Know About Data Analysis and Fitting but Were Afraid to Ask*, SpringerBriefs in Physics,
DOI 10.1007/978-3-319-19051-8

$$X = \sum_{i=1}^{N} x_i .$$

distribution.

The determination of the distribution of X, which we call $P_N(X)$, uses the Fourier transform of $P(x)$, called the "characteristic function" in the context of probability theory. This is defined by

$$Q(k) = \int_{-\infty}^{\infty} P(x) e^{ikx}\, dx.$$

Expanding out the exponential we can write $Q(k)$ in terms of the moments of $P(x)$

$$Q(k) = 1 + ik\langle x\rangle + \frac{(ik)^2}{2!}\langle x^2\rangle + \frac{(ik)^3}{3!}\langle x^3\rangle + \cdots .$$

It will be convenient in what follows to write $Q(k)$ as an exponential, i.e.

$$\begin{aligned} Q(k) &= \exp\left[\ln\left(1 + ik\langle x\rangle + \frac{(ik)^2}{2!}\langle x^2\rangle + \frac{(ik)^3}{3!}\langle x^3\rangle + \cdots\right)\right] \\ &= \exp\left[ik\mu + \frac{(ik)^2\sigma^2}{2!} + \frac{c_3(ik)^3}{3!} + \frac{c_4(ik)^4}{4!} + \cdots\right], \end{aligned} \tag{A.4}$$

where c_3 involves third and lower moments, c_4 involves fourth and lower moments, and so on. The c_n are called *cumulant* averages.

For the important case of a Gaussian, the Fourier transform is obtained by "completing the square". The result is that the Fourier transform of a Gaussian is also a Gaussian, namely,

$$Q_{\text{Gauss}}(k) = \exp\left[ik\mu - \frac{k^2\sigma^2}{2}\right], \tag{A.5}$$

showing that the higher order cumulants, c_3, c_4, etc. in Eq. (A.4) *all vanish* for a Gaussian.

The distribution $P_N(x)$ can be expressed as

$$P_N(x) = \int_{-\infty}^{\infty} P(x_1)dx_1 \int_{-\infty}^{\infty} P(x_2)dx_2 \cdots \int_{-\infty}^{\infty} P(x_N)dx_N\, \delta(X - \sum_{i=1}^{N} x_i). \tag{A.6}$$

We evaluate this by using the integral representation of the delta function

$$\delta(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{ikx}\, dk, \tag{A.7}$$

so

$$P_N(X) = \int_{-\infty}^{\infty} \frac{dk}{2\pi} \int_{-\infty}^{\infty} P(x_1)dx_1 \int_{-\infty}^{\infty} P(x_2)dx_2 \cdots$$
$$\int_{-\infty}^{\infty} P(x_N)dx_N \exp[ik(x_1 + x_2 + \cdots x_N - X)] \tag{A.8}$$
$$= \int_{-\infty}^{\infty} \frac{dk}{2\pi} Q(k)^N e^{-ikX}, \tag{A.9}$$

showing that the Fourier transform of $P_N(x)$, which we call $Q_N(k)$, is given by

$$Q_N(k) = Q(k)^N. \tag{A.10}$$

Consequently

$$Q_N(k) = \exp\left[ikN\mu - \frac{Nk^2\sigma^2}{2} + \frac{Nc_3(ik)^3}{4!} + \frac{Nc_4(ik)^4}{4!} + \cdots\right]. \tag{A.11}$$

Comparing with Eq. (A.4) we see that

> the mean of the distribution of the sum of N independent and identically distributed random variables (the coefficient of $-ik$ in the exponential) is N times the mean of the distribution of one variable, and the variance of the distribution of the sum (the coefficient of $-k^2/2!$) is N times the variance of the distribution of one variable.

These are general statements applicable for *any* N and have already been derived in Sect. 2.1.

However, if N is *large* we can now go further. The distribution $P_N(X)$ is the inverse transform of $Q_N(k)$, see Eq. (A.9), so

$$P_N(X) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \exp\left[-ikX' - \frac{Nk^2\sigma^2}{2!} + N\frac{c_3(ik)^3}{3!} + \frac{Nc_4(ik)^4}{4!} + \cdots\right] dk, \tag{A.12}$$

where

$$X' = X - N\mu. \tag{A.13}$$

Looking at the $-Nk^2/2$ term in the exponential in Eq. (A.12), we see that the integrand is significant for $k < k^\star$, where $N\sigma^2(k^\star)^2 = 1$, and negligibly small for $k \gg k^\star$. However, for $k \sim k^\star$ the higher order terms in Eq. (A.12), (i.e. those of order k^3, k^4 etc.) are very small since $N(k^\star)^3 \sim N^{-1/2}$, $N(k^\star)^4 \sim N^{-1}$ and so on.

Hence the terms of higher order than k^2 in Eq. (A.12), do not contribute for large N and so

$$\lim_{N\to\infty} P_N(X) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \exp\left[-ikX' - \frac{Nk^2\sigma^2}{2}\right] dk. \tag{A.14}$$

In other words, for large N the distribution is the Fourier transform of a Gaussian, which, as we know, is also a Gaussian. Completing the square in Eq. (A.14) gives

$$\begin{aligned}\lim_{N\to\infty} P_N(X) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} \exp\left[-\frac{N\sigma^2}{2}\left(k - \frac{iX'}{N\sigma^2}\right)^2\right] dk \ \exp\left[-\frac{(X')^2}{2N\sigma^2}\right] \\ &= \frac{1}{\sqrt{2\pi N}\,\sigma} \exp\left[-\frac{(X - N\mu)^2}{2N\sigma^2}\right],\end{aligned} \tag{A.15}$$

where, in the last line, we used Eq. (A.13). This is a Gaussian with mean $N\mu$ and variance $N\sigma^2$. Equation (A.15) is the central limit theorem in statistics. It tells us that,

> for $N \to \infty$, the distribution of the sum of N independent and identically distributed random variables is a *Gaussian* whose mean is N times the mean, μ, of the distribution of one variable, and whose variance is N times the variance of the distribution of one variable, σ^2, *independent of the form of the distribution of one variable*, $P(x)$, provided only that μ and σ are finite.

The central limit theorem is of such generality that it is extremely important. It is the reason why the Gaussian distribution has such a preeminent place in the theory of statistics.

Note that if the distribution of the individual x_i is Gaussian, then the distribution of the sum of N variables is *always* Gaussian, even for N small. This follows from Eq. (A.10) and the fact that the Fourier transform of a Gaussian is a Gaussian.

In practice, distributions that we meet in nature, have a much broader tail than that of the Gaussian distribution, which falls off very fast at large $|x - \mu|/\sigma$. As a result, even if the distribution of the sum approximates well a Gaussian in the central region for only modest values of N, it might take a much larger value of N to beat down the weight in the tail to the value of the Gaussian. Hence, even for moderate values of N, the probability of a deviation greater than σ can be significantly larger than that of the Gaussian distribution which is 32 %. This caution will be important in Chap. 3 when we discuss the quality of fits.

We will illustrate the slow convergence of the distribution of the sum to a Gaussian in Fig. A.1, in which the distribution of the individual variables x_i is

$$P(x) = \frac{3}{2}\,\frac{1}{(1 + |x|)^4}. \tag{A.16}$$

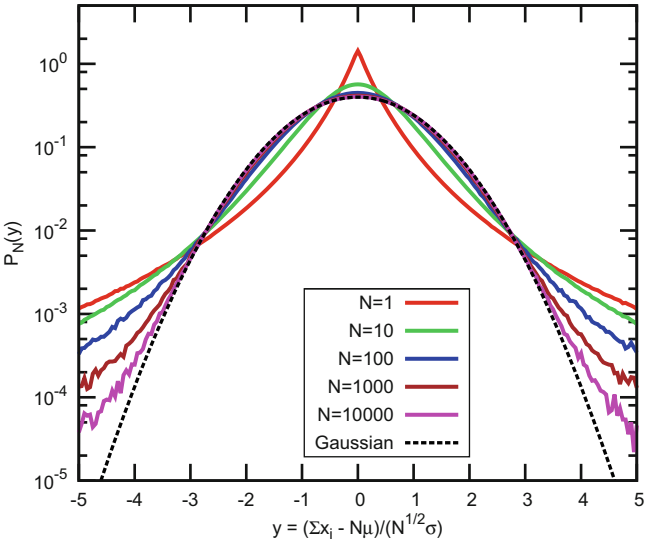

Fig. A.1 Figure showing the approach to the central limit theorem for the distribution in Eq. (A.16), which has mean, μ, equal to 0, and standard deviation, σ, equal to 1. The horizontal axis is the sum of N random variables divided by $\sqrt{N}$ which, for all N, has zero mean and standard deviation unity. For large N the distribution approaches a Gaussian. However, convergence is non-uniform, and is extremely slow in the tails. Note the log vertical scale which is necessary to display the weight in the tails

This has mean 0 and standard deviation 1, but moments higher than the second do not exist because the integrals diverge. For large N the distribution approaches a Gaussian, as expected, but convergence is very slow in the tails.

Appendix B
The Number of Degrees of Freedom

We assume Gaussian noise on the data and consider first a straight line fit, so we have to determine the values of a_0 and a_1 which minimize Eq. (3.7). The N terms in Eq. (3.7) are not statistically independent at the minimum because the values of a_0 and a_1, given by Eq. (3.8), depend on the data points (x_i, y_i, σ_i).

Consider the "residuals" defined by

$$\epsilon_i = \frac{y_i - a_0 - a_1 x_i}{\sigma_i}. \tag{B.1}$$

If the model were exact and we use the exact values of the parameters a_0 and a_1 the ϵ_i would be independent and each have a Gaussian distribution with zero mean and standard deviation unity. However, choosing the *best-fit* values of a_0 and a_1 *from the data* according to Eq. (3.8) implies that

$$\sum_{i=1}^{N} \frac{1}{\sigma_i} \epsilon_i = 0, \tag{B.2a}$$

$$\sum_{i=1}^{N} \frac{x_i}{\sigma_i} \epsilon_i = 0, \tag{B.2b}$$

which are are two *linear constraints* on the ϵ_i. This means that we only need to specify $N - 2$ of them to know them all. In the N dimensional space of the ϵ_i we have eliminated two directions, so there can be no Gaussian fluctuations along them. However the other $N - 2$ dimensions are unchanged, and will have the same Gaussian fluctuations as before. Thus χ^2 has the distribution of a sum of squares of $N - 2$ Gaussian random variables. We can intuitively understand why there are $N - 2$ degrees of freedom rather than N by considering the case of $N = 2$. The fit goes perfectly through the two points so one has $\chi^2 = 0$ exactly. This implies that there are zero degrees of freedom since, on average, each degree of freedom adds 1 to χ^2.

P. Young, *Everything You Wanted to Know About Data Analysis and Fitting but Were Afraid to Ask*, SpringerBriefs in Physics,
DOI 10.1007/978-3-319-19051-8

Clearly this argument can be generalized to any fitting function which depends *linearly* on M fitting parameters, assuming Gaussian noise on the data. The result is that χ^2 has the distribution of a sum of squares of $N_{\mathrm{DOF}} = N - M$ Gaussian random variables, in which the quantity N_{DOF} is called the "number of degrees of freedom".

Even if the fitting function depends non-linearly on the parameters, this last result is often taken as a reasonable approximation.

Appendix C
The Chi-squared Distribution and the Goodness of Fit Parameter Q

The χ^2 distribution for m degrees of freedom is the distribution of the sum of m independent random variables with a Gaussian distribution with zero mean and standard deviation unity. To determine this we write the distribution of the m variables x_i as

$$P(x_1, x_2, \cdots, x_m)\, dx_1 dx_2 \cdots dx_m = \frac{1}{(2\pi)^{m/2}}\, e^{-x_1^2/2}\, e^{-x_2^2/2} \cdots e^{-x_m^2/2}\, dx_1 dx_2 \cdots dx_m. \tag{C.1}$$

Converting to polar coordinates, and integrating over directions, we find the distribution of the radial variable to be

$$\widetilde{P}(r)\, dr = \frac{S_m}{(2\pi)^{m/2}}\, r^{m-1}\, e^{-r^2/2}\, dr, \tag{C.2}$$

where S_m is the surface area of a unit m-dimensional sphere. To determine S_m we integrate Eq. (C.2) over r, noting that $\widetilde{P}(r)$ is normalized, which gives

$$S_m = \frac{2\pi^{m/2}}{\Gamma(m/2)}, \tag{C.3}$$

where $\Gamma(x)$ is the Euler gamma function defined by

$$\Gamma(x) = \int_0^\infty t^{x-1}\, e^{-t}\, dt. \tag{C.4}$$

From Eqs. (C.2) and (C.3) we have

$$\widetilde{P}(r) = \frac{1}{2^{m/2-1}\Gamma(m/2)}\, r^{m-1} e^{-r^2/2}. \tag{C.5}$$

P. Young, *Everything You Wanted to Know About Data Analysis and Fitting but Were Afraid to Ask*, SpringerBriefs in Physics,
DOI 10.1007/978-3-319-19051-8

This is the distribution of r but we want the distribution of $\chi^2 \equiv \sum_i x_i^2 = r^2$. To avoid confusion of notation we write X for χ^2, and define the χ^2 distribution for m variables as $P^{(m)}(X)$. We have $P^{(m)}(X)\,dX = \widetilde{P}(r)\,dr$ so the χ^2 distribution for m degrees of freedom is

$$P^{(m)}(X) = \frac{\widetilde{P}(r)}{dX/dr} = \frac{1}{2^{m/2}\Gamma(m/2)}\, X^{(m/2)-1}\, e^{-X/2} \qquad (X > 0). \tag{C.6}$$

The χ^2 distribution is zero for $X < 0$. Using Eq. (C.4) and the property of the gamma function that $\Gamma(n+1) = n\Gamma(n)$ one can show that

$$\int_0^\infty P^{(m)}(X)\,dX = 1, \tag{C.7a}$$

$$\langle X \rangle \equiv \int_0^\infty X\, P^{(m)}(X)\,dX = m, \tag{C.7b}$$

$$\langle X^2 \rangle \equiv \int_0^\infty X^2\, P^{(m)}(X)\,dX = m^2 + 2m, \quad \text{so} \tag{C.7c}$$

$$\langle X^2 \rangle - \langle X \rangle^2 = 2m. \tag{C.7d}$$

Furthermore, the peak of the distribution is at $X = m - 2$ (for $m > 2$).

From Eqs. (C.7b) and (C.7d) we see that typically χ^2 lies in the range $m - \sqrt{2m}$ to $m + \sqrt{2m}$. For large m the distribution approaches a Gaussian according to the central limit theory discussed in Appendix A. Typically one focuses on the value of χ^2 per degree freedom since this should be around unity for all m.

The goodness of fit parameter is the probability that the specified value of χ^2, or greater, could occur by random chance. From Eq. (C.6) it is given by

$$Q = \frac{1}{2^{m/2}\Gamma(m/2)} \int_{\chi^2}^\infty X^{(m/2)-1}\, e^{-X/2}\, dX, \tag{C.8}$$

$$= \frac{1}{\Gamma(m/2)} \int_{\chi^2/2}^\infty y^{(m/2)-1}\, e^{-y}\, dy, \tag{C.9}$$

which is known as an incomplete gamma function. Code to generate the incomplete gamma function is given in Numerical Recipes [1]. There is also a built-in function to generate the goodness of fit parameter in the `scipy` package of `python` and in the graphics program `gnuplot`, see the scripts in Appendix H.

The χ^2 distribution for several value of $m \equiv N_{\rm DOF}$ is plotted in Fig. C.1. The mean and variance are given by Eqs. (C.7b) and (C.7d). For large m, according to the central limit theorem, the χ^2 distribution becomes a Gaussian.

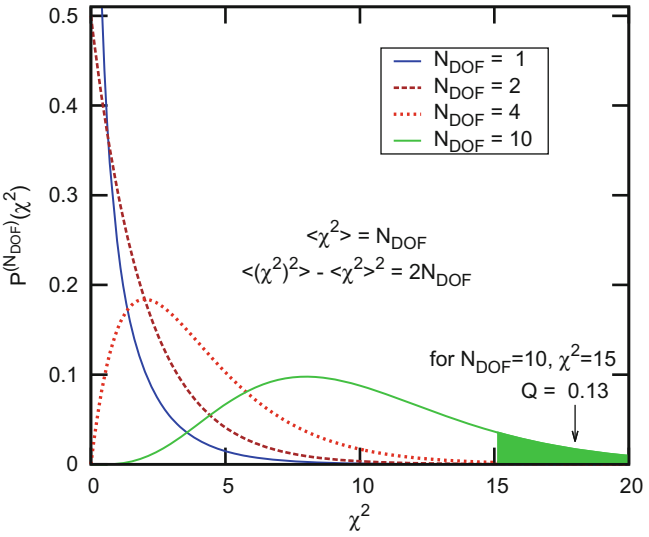

Fig. C.1 The χ^2 distribution for several values of N_{DOF} the number of degrees of freedom. The mean and standard deviation depend on N_{DOF} in the way specified. The goodness of fit parameter Q, defined in Eq. (3.25), depends on the values of N_{DOF} and χ^2, and is the probability that χ^2 could have the specified value or larger by random chance. The area of the shaded region in the figure is the value of Q for $N_{\mathrm{DOF}} = 10$, $\chi^2 = 15$. Note that the *total* area under each of the curves is unity because they represent probability distributions

Note that $Q = 1$ for $\chi^2 = 0$ and $Q \to 0$ for $\chi^2 \to \infty$. Remember that m is the number of degrees of freedom, called N_{DOF} elsewhere in these notes, and that $N_{\mathrm{DOF}} = N - M$, where N is the number of data points and M is the number of fit parameters.

Appendix D
Asymptotic Standard Error and How to Get Correct Error Bars from Gnuplot

Sometimes one does not have error bars on the data. Nonetheless, one can still use χ^2 fitting to get an *estimate* of those errors (assuming that they are all equal) and thereby also get an error bar on the fit parameters. The latter is called the "asymptotic standard error". Assuming the same error bar $\sigma_{\rm ass}$ for all points, we determine $\sigma_{\rm ass}$ from the requirement that χ^2 per degree of freedom is precisely one, i.e. its mean value according to Eq. (C.7b). This gives

$$1 = \frac{\chi^2}{N_{\rm DOF}} = \frac{1}{N_{\rm DOF}} \sum_{i=1}^{N} \left(\frac{y_i - f(x_i)}{\sigma_{\rm ass}} \right)^2 , \tag{D.1}$$

or, equivalently,

$$\sigma_{\rm ass}^2 = \frac{1}{N_{\rm DOF}} \sum_{i=1}^{N} (y_i - f(x_i))^2 . \tag{D.2}$$

The error bars on the fit parameters are then obtained from Eq. (3.24), with the elements of U given by Eq. (3.10) in which σ_i is replaced by $\sigma_{\rm ass}$. Equivalently, one can set the σ_i to unity in determining U from Eq. (3.10), and estimate the error on the fit parameters from

$$\sigma_\alpha^2 = (U)_{\alpha\alpha}^{-1} \, \sigma_{\rm ass}^2, \quad \text{(asymptotic standard error)}. \tag{D.3}$$

A simple example of the use of the asymptotic standard error in a situation where we don't know the error on the data points, is fitting to a constant, i.e. *determining the average of a set of data*, which we already discussed in detail in Chap. 2. In this case we have

$$U_{00} = N, \qquad v_0 = \sum_{i=1}^{N} y_i , \tag{D.4}$$

P. Young, *Everything You Wanted to Know About Data Analysis and Fitting but Were Afraid to Ask*, SpringerBriefs in Physics,
DOI 10.1007/978-3-319-19051-8

so the only fit parameter is

$$a_0 = \frac{v_0}{U_{00}} = \frac{1}{N}\sum_{i=1}^{N} y_i = \overline{y}, \tag{D.5}$$

which gives, naturally enough, the average of the data points, $\overline{y}$. The number of degrees of freedom is $N-1$, since there is one fit parameter, so

$$\sigma_{\text{ass}}^2 = \frac{1}{N-1}\sum_{i=1}^{N} (y_i - \overline{y})^2\,, \tag{D.6}$$

and hence the square of the error on a_0 is given, from Eq. (D.3), by

$$\sigma_0^2 = \frac{1}{U_{00}}\,\sigma_{\text{ass}}^2 = \frac{1}{N(N-1)}\sum_{i=1}^{N} (y_i - \overline{y})^2\,, \tag{D.7}$$

which is precisely the expression for the error in the mean of a set of data given in Eq. (2.16).

I now mention that a popular plotting program, `gnuplot`, which also does fits but unfortunately presents error bars on the fit parameters incorrectly if there are error bars on the data. Whether or not there are error bars on the points, `gnuplot` gives the "asymptotic standard error" on the fit parameters. `Gnuplot` calculates the elements of U correctly from Eq. (3.10) including the error bars, but then apparently also determines an "assumed error" from an expression like Eq. (D.2) but including the error bars, i.e.

$$\sigma_{\text{ass}}^2 = \frac{1}{N_{\text{DOF}}}\sum_{i=1}^{N}\left(\frac{y_i - f(x_i)}{\sigma_i}\right)^2 = \frac{\chi^2}{N_{\text{DOF}}}, \qquad \texttt{(gnuplot)}. \tag{D.8}$$

Hence `gnuplot`'s σ_{ass}^2 is just the chi-squared per degree of freedom. The error bar (squared) quoted by `gnuplot` is $(U)_{\alpha\alpha}^{-1}\,\sigma_{\text{ass}}^2$, as in Eq. (D.3). However, this is wrong since the error bars on the data points have *already* been included in calculating the elements of U, so the error on the fit parameter α should really be $(U)_{\alpha\alpha}^{-1}$. Hence,

> to get correct error bars on fit parameters from `gnuplot` when there are error bars on the points, you have to divide `gnuplot`'s asymptotic standard errors by the square root of the chi-squared per degree of freedom (which gnuplot calls `FIT_STDFIT` and, fortunately, computes correctly).

I have checked this statement by comparing with results from Numerical Recipes routines, and also, for straight-line fits, by my own implementation of the formulae. It is curious that I found no hits on this topic when Googling the internet. Can no one else have come across this problem? Correction of `gnuplot` error bars is implemented in the `gnuplot` scripts in Appendix H.

The need to correct `gnuplot`'s error bars applies to linear as well as non-linear models.

I recently learned that error bars on fit parameters given by the routine `curve_fit` of `python` also have to be corrected in the same way. This is shown in two of the python scripts in Appendix H. Curiously, a different python fitting routine, `leastsq`, gives the error bars correctly.

Appendix E
The Distribution of Fitted Parameters Determined from Simulated Datasets

In this section we derive the equation for the distribution of fitted parameters determined from simulated datasets, Eq. (3.42), assuming an arbitrary linear model, see Eq. (3.30). Projecting on to a single fitting parameter, as above, this corresponds to the lower figure in Fig. 3.2.

We have *one* set of y-values, $y_i^{(0)}$, for which the fit parameters are $\vec{a}^{(0)}$. We then generate an *ensemble* of simulated data sets, y_i^S, assuming the data has Gaussian noise with standard deviation σ_i centered on the actual data values $y_i^{(0)}$. We ask for the probability that the fit to one of the simulated data sets has parameters $\vec{a}^S$.

This probability distribution is given by

$$P(\vec{a}^S) = \prod_{i=1}^{N} \left\{ \frac{1}{\sqrt{2\pi}\sigma_i} \int_{-\infty}^{\infty} dy_i^S \exp\left[-\frac{\left(y_i^S - y_i^{(0)}\right)^2}{2\sigma_i^2} \right] \right\} \prod_{\alpha=1}^{M} \delta\left(\sum_{\beta} U_{\alpha\beta} a_\beta^S - v_\alpha^S \right) \det U, \tag{E.1}$$

where the factor in curly brackets is (an integral over) the probability distribution of the data points y_i^S, and the delta functions project out those sets of data points which have a particular set of fitted parameters, see Eq. (3.34). The factor of $\det U$ is a Jacobian to normalize the distribution. Using the integral representation of the delta function, and writing explicitly the expression for v_α from Eq. (3.33), one has

$$P(\vec{a}^S) = \prod_{i=1}^{N} \left\{ \frac{1}{\sqrt{2\pi}\sigma_i} \int_{-\infty}^{\infty} dy_i^S \exp\left[-\frac{\left(y_i^S - y_i^{(0)}\right)^2}{2\sigma_i^2} \right] \right\} \times \tag{E.2}$$

$$\prod_{\alpha=1}^{M} \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} dk_\alpha \exp\left[ik_\alpha \left(\sum_{\beta} U_{\alpha\beta} a_\beta^S - \sum_{i=1}^{N} \frac{y_i^S X_\alpha(x_i)}{\sigma_i^2} \right) \right] \right) \det U\,. \tag{E.3}$$

P. Young, *Everything You Wanted to Know About Data Analysis and Fitting but Were Afraid to Ask*, SpringerBriefs in Physics,
DOI 10.1007/978-3-319-19051-8

We carry out the y integrals by "completing the square",

$$P(\vec{a}^S) = \prod_{\alpha=1}^{M} \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} dk_\alpha \right) \prod_{i=1}^{N} \left\{ \frac{1}{\sqrt{2\pi}\sigma_i} \int_{-\infty}^{\infty} dy_i^S \exp\left[-\frac{\left(y_i^S - y_i^{(0)} + i\vec{k} \cdot \vec{X}(x_i) \right)^2}{2\sigma_i^2} \right] \right\} \tag{E.4}$$

$$\times \exp\left[-\frac{1}{2\sigma_i^2} \left(\left(\vec{k} \cdot \vec{X}(i) \right)^2 + 2i \left(\vec{k} \cdot \vec{X}(x_i) \right) y_i^{(0)} \right) \right] \times \exp\left[i \sum_{\alpha,\beta} k_\alpha \, U_{\alpha\beta} \, a_\beta^S \right] \det U. \tag{E.5}$$

Doing the y^S-integrals, the factors in curly brackets are equal to unity. Using Eqs. (3.32) and (3.33) and the fact that the $U_{\alpha\beta}$ are independent of the y_i^S, we then get

$$P(\vec{a}^S) = \prod_{\alpha=1}^{M} \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} dk_\alpha \right) \exp\left[-\frac{1}{2} \sum_{\alpha,\beta} k_\alpha \, U_{\alpha\beta} \, k_\beta + i \sum_{\alpha,\beta} k_\alpha \, \delta v_\alpha^S \right] \det U, \tag{E.6}$$

where

$$\delta v_\beta^S \equiv v_\beta^S - v_\beta^{(0)}, \tag{E.7}$$

with

$$v_\alpha^{(0)} = \sum_{i=1}^{N} \frac{y_i^{(0)} \, X_\alpha(x_i)}{\sigma_i^2}. \tag{E.8}$$

We do the k-integrals by working in the basis in which U is diagonal. The result is

$$P(\vec{a}^S) = \frac{(\det U)^{1/2}}{(2\pi)^{M/2}} \exp\left[-\frac{1}{2} \sum_{\alpha,\beta} \delta v_\alpha^S \left(U^{-1} \right)_{\alpha\beta} \delta v_\beta^S \right]. \tag{E.9}$$

Using Eq. (3.34) and the fact that U is symmetric we get our final result

$$P(\vec{a}^S) = \frac{(\det U)^{1/2}}{(2\pi)^{M/2}} \exp\left[-\frac{1}{2} \sum_{\alpha,\beta} \delta a_\alpha^S \, U_{\alpha\beta} \, \delta a_\beta^S \right], \tag{E.10}$$

which is Eq. (3.42), including the normalization constant in front of the exponential.

Appendix F
The Distribution of Fitted Parameters from Repeated Sets of Measurements

In this section we derive the equation for the distribution of fitted parameters determined in the hypothetical situation that one has many actual data sets. Projecting on to a single fitted parameter, this corresponds to the upper panel in Fig. 3.2.

The exact value of the data is $y_i^{\text{true}} = \vec{a}^{\text{true}} \cdot \vec{X}(x_i)$, see Eq. (3.30), and the distribution of the y_i in an actual data set, which differs from y_i^{true} because of noise, has a distribution, assumed Gaussian here, centered on y_i^{true} with standard deviation σ_i. Fitting each of these real data sets, the probability distribution for the fitted parameters is given by

$$P(\vec{a}) = \prod_{i=1}^{N} \left\{ \frac{1}{\sqrt{2\pi}\sigma_i} \int_{-\infty}^{\infty} dy_i \, \exp\left[-\frac{\left(y_i - \vec{a}^{\text{true}} \cdot \vec{X}(x_i)\right)^2}{2\sigma_i^2} \right] \right\} \prod_{\alpha=1}^{M} \delta\left(\sum_{\beta} U_{\alpha\beta} a_\beta - v_\alpha \right) \det U, \tag{F.1}$$

see Eq. (E.1) for an explanation of the various factors. Proceeding as in Appendix E we have

$$P(\vec{a}) = \prod_{i=1}^{N} \left\{ \frac{1}{\sqrt{2\pi}\sigma_i} \int_{-\infty}^{\infty} dy_i \, \exp\left[-\frac{\left(y_i - \vec{a}^{\text{true}} \cdot \vec{X}(x_i)\right)^2}{2\sigma_i^2} \right] \right\} \times \tag{F.2}$$

$$\prod_{\alpha=1}^{M} \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} dk_\alpha \exp\left[ik_\alpha \left(\sum_{\beta} U_{\alpha\beta} a_\beta - \sum_{i=1}^{N} \frac{y_i \, X_\alpha(x_i)}{\sigma_i^2} \right) \right] \right) \det U, \tag{F.3}$$

P. Young, *Everything You Wanted to Know About Data Analysis and Fitting but Were Afraid to Ask*, SpringerBriefs in Physics,
DOI 10.1007/978-3-319-19051-8

and doing the y-integrals by completing the square gives

$$P(\vec{a}) = \prod_{\alpha=1}^{M} \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} dk_\alpha \right) \tag{F.4}$$

$$\times \exp\left[-\frac{1}{2\sigma_i^2} \left(\left(\vec{k} \cdot \vec{X}(i) \right)^2 + 2i \left(\vec{k} \cdot \vec{X}(x_i) \right) \left(\vec{a}^{\,\text{true}} \cdot \vec{X}(x_i) \right) \right) \right]$$

$$\times \exp\left[i \sum_{\alpha,\beta} k_\alpha \, U_{\alpha\beta} \, a_\beta \right] \det U. \tag{F.5}$$

Using Eq. (3.32) we then get

$$P(\vec{a}) = \prod_{\alpha=1}^{M} \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} dk_\alpha \right) \exp\left[-\frac{1}{2} \sum_{\alpha,\beta} k_\alpha \, U_{\alpha\beta} \, k_\beta + i \sum_{\alpha,\beta} k_\alpha \, U_{\alpha\beta} \, \delta a_\beta \right] \det U, \tag{F.6}$$

where

$$\delta a_\beta \equiv a_\beta - a_\beta^{\text{true}}, \tag{F.7}$$

and we used Eq. (3.32). The k-integrals are done by working in the basis in which U is diagonal. The result is

$$P(\vec{a}) = \frac{(\det U)^{1/2}}{(2\pi)^{M/2}} \exp\left[-\frac{1}{2} \sum_{\alpha,\beta} \delta a_\alpha \, U_{\alpha\beta} \, \delta a_\beta \right]. \tag{F.8}$$

In other words, the distribution of the fitted parameters obtained from many sets of actual data, about the *true* value $\vec{a}^{\,\text{true}}$ is a Gaussian. Since we are assuming a linear model, the matrix of coefficients $U_{\alpha\beta}$ is a constant, and so the distribution in Eq. (F.8) is the *same* as in Eq. (E.10). Hence

> For a linear model with Gaussian noise, the distribution of fitted parameters, obtained from simulated data sets, relative to *value from the one actual data set*, is the same as the distribution of parameters from many actual data sets relative to *the true value*, see Fig. 3.2.

This result is also valid for a non-linear model if the range of parameter values needed is sufficiently small that the model can be represented by an effective one. It is usually assumed to be a reasonable approximation even if this condition is not fulfilled.

Appendix G
Fitting Correlated Data

Consider N data points (x_i, y_i, σ_i), $i = 1, 2, \cdots N$. Correlations among the y-values are described by a matrix C where

$$C_{ij} \equiv \langle \delta y_i \delta y_j \rangle = \langle y_i y_j \rangle - \langle y_i \rangle \langle y_j \rangle. \tag{G.1}$$

In this section we assume a linear model for ease of notation, but the generalization to a non-linear model is straightforward. Assuming Gaussian noise, the probability distribution for the data which gives these correlations is

$$P(\{y\}) = \frac{1}{(2\pi)^{N/2}(\det C)^{1/2}} \exp\left[-\frac{1}{2} \sum_{i,j} \left(y_i - \sum_\alpha a_\alpha X_\alpha(x_i) \right) \left(C^{-1} \right)_{ij} \left(y_j - \sum_\beta a_\beta X_\beta(x_j) \right) \right], \tag{G.2}$$

where we have used the following properties of Gaussian integrals:

$$\int_{-\infty}^{\infty} dy_1 \int_{-\infty}^{\infty} dy_2 \cdots \int_{-\infty}^{\infty} dy_N \exp\left[-\frac{1}{2} \sum_{i,j} y_i A_{ij} y_j \right] = \frac{(2\pi)^{N/2}}{(\det A)^{1/2}}, \tag{G.3}$$

$$\frac{\displaystyle\int_{-\infty}^{\infty} dy_1 \int_{-\infty}^{\infty} dy_2 \cdots \int_{-\infty}^{\infty} dy_N \, y_k y_l \exp\left[-\frac{1}{2} \sum_{i,j} y_i A_{ij} y_j \right]}{\displaystyle\int_{-\infty}^{\infty} dy_1 \int_{-\infty}^{\infty} dy_2 \cdots \int_{-\infty}^{\infty} dy_N \exp\left[-\frac{1}{2} \sum_{i,j} y_i A_{ij} y_j \right]} = \left(A^{-1} \right)_{kl}, \tag{G.4}$$

P. Young, *Everything You Wanted to Know About Data Analysis and Fitting but Were Afraid to Ask*, SpringerBriefs in Physics,
DOI 10.1007/978-3-319-19051-8

in which we assumed that the matrix A is positive definite, i.e. all its eigenvectors are positive. Equations (G.3) and (G.4) are obtained by doing a change of variables in the N-dimensional space of the y-s to new variables which are in the direction of the eigenvectors of A. The Jacobian of the transformation is unity because it is the determinant of the (orthogonal) matrix, U, which the diagonalizes the real, symmetric matrix A. The integrals are now independent and can be easily performed. In Eq. (G.3) the product of the eigenvalues has been written as the determinant. In Eq. (G.4) we have noted that if $A = UDU^{-1}$, where D is the diagonal matrix with eigenvalues on the diagonal, then $A^{-1} = UD^{-1}U^{-1}$ (and, since U is orthogonal, $U^{-1} = U^T$, the transpose).

Since least-squares is equivalent to maximum likelihood we have to maximize the probability in Eq. (G.2). This is equivalent to minimizing a "*cost function*" which is minus (two times) the exponent in Eq. (G.2), i.e.

$$\sum_{i,j}\left(y_i - \sum_{\alpha} a_\alpha X_\alpha(x_i)\right)\left(C^{-1}\right)_{ij}\left(y_j - \sum_{\beta} a_\beta X_\beta(x_j)\right). \tag{G.5}$$

In the absence of correlations $C_{ij} = \sigma_i^2\,\delta_{ij}$, $\left(C^{-1}\right)_{ij} = \delta_{ij}/\sigma_i^2$ and we recover the earlier expression for χ^2 in Eq. (3.31). Minimizing Eq. (G.5) with respect to the a_α we get equations of the same form as before, namely

$$\sum_{\beta=1}^{M} U_{\alpha\beta}\,a_\alpha = v_\beta, \tag{G.6}$$

but with different expressions for U and v, namely

$$U_{\alpha\beta} = \sum_{i,j} X_\alpha(x_i)\left(C^{-1}\right)_{ij} X_\beta(x_j), \tag{G.7a}$$

$$v_\alpha = \sum_{i,j} X_\alpha(x_i)\left(C^{-1}\right)_{ij} y_j. \tag{G.7b}$$

rather than Eqs. (3.32) and (3.33). The covariance matrix of the parameters is still given by Eq. (3.28).

In practice, though, we rarely have enough information on the correlations between data points for Eqs. (G.5–G.7) to be useful.

Appendix H
Scripts for Some Data Analysis and Fitting Tasks

In this Appendix I give sample scripts using perl, python and gnuplot for some basic data analysis and fitting tasks. I include output from the scripts when acting on certain datasets which are available on the web.

Note "`this_file_name`" refers to the name of the script being displayed (whatever you choose to call it.)

H.1 Scripts for a Jackknife Analysis

The script reads in values of x on successive lines of the input file and computes $\langle x^4\rangle/\langle x^2\rangle^2$, including an error bar computed using the jackknife method.

H.1.1 Perl

```
#!/usr/bin/perl
#
#  Usage: "this_file_name data_file"
# (make the script executable; otherwise you have to preface the command with "perl")
# $n = 0;
$x2_tot = 0; $x4_tot = 0;
#
# read in the data
#
while(<>)   # Note this very convenient perl command which reads each line of
               # of each input file in the command line
{
    @line = split;
    $x2[$n] = $line[0]**2;
    $x4[$n] = $x2[$n]**2;
    $x2_tot += $x2[$n];
    $x4_tot+= $x4[$n];
    $n++;
}
#
```


P. Young, *Everything You Wanted to Know About Data Analysis and Fitting but Were Afraid to Ask*, SpringerBriefs in Physics,
DOI 10.1007/978-3-319-19051-8

```
# Do the jackknife estimates
#
for ($i = 0; $i < $n; $i++)
{
    $x2_jack[$i] = ($x2_tot - $x2[$i]) / ($n - 1);
    $x4_jack[$i] = ($x4_tot - $x4[$i]) / ($n - 1);
}
$x2_av = $x2_tot / $n;    # Do the overall averages
$x4_av = $x4_tot / $n;
$g_av = $x4_av / $x2_av**2;

$g_jack_av = 0; $g_jack_err = 0; # Do the final jackknife estimate
for ($i = 0; $i < $n; $i++)
{
    $dg = $x4_jack[$i] / $x2_jack[$i]**2;
    $g_jack_av += $dg;
    $g_jack_err += $dg**2;
} $g_jack_av /= $n;
$g_jack_err /= $n;
$g_jack_err = sqrt(($n - 1) * abs($g_jack_err - $g_jack_av**2));

printf " Overall average is   %8.4f\n", $g_av;
printf " Jackknife average is %8.4f +/- %6.4f \n", $g_jack_av, $g_jack_err;
```

Executing this file on the data in http://young.physics.ucsc.edu/bad-honnef/data.HW2 gives

```
 Overall average is     1.8215
 Jackknife average is   1.8215 +/- 0.0368
```

H.1.2 Python

```
#
# Program written by Matt Wittmann
#
# Usage: "python this_file_name data_file"
# import fileinput
from math import *

x2 = []; x2_tot = 0.
x4 = []; x4_tot = 0.
for line in fileinput.input():  # read in each line in each input file.
                                # similar to perl's while(<>)
    line = line.split()
    x2_i = float(line[0])**2
    x4_i = x2_i**2
    x2.append(x2_i)            # put x2_i as the i-th element in an array x2
    x4.append(x4_i)
    x2_tot += x2_i
    x4_tot += x4_i
n = len(x2)                       # the number of lines read in
#
# Do the jackknife estimates
#
x2_jack = []
x4_jack = []
for i in xrange(n):
    x2_jack.append((x2_tot - x2[i]) / (n - 1))
    x4_jack.append((x4_tot - x4[i]) / (n - 1))

x2_av = x2_tot / n            # do the overall averages
x4_av = x4_tot / n
```

```
g_av = x4_av / x2_av**2
g_jack_av = 0.; g_jack_err = 0.

for i in xrange(n):              # do the final jackknife averages
    dg = x4_jack[i] / x2_jack[i]**2
    g_jack_av += dg
    g_jack_err += dg**2

g_jack_av /= n
g_jack_err /= n
g_jack_err = sqrt((n - 1) * abs(g_jack_err - g_jack_av**2))

print " Overall average is   %8.4f" % g_av
print " Jackknife average is %8.4f +/- %6.4f" % (g_jack_av, g_jack_err)
```

The output is the same as for the perl script.

H.2 Scripts for a Straight-Line Fit

H.2.1 Perl, Writing Out the Formulae by Hand

```
#!/usr/bin/perl
#
#  Usage: "this_file_name data_file"
#  (make the script executable; otherwise preface the command with "perl")
#
# Does a straight line fit to data in "data_file" each line of which contains
#  data for one point, x_i, y_i, sigma_i
# $n = 0;
while(<>)    # read in the lines of data
{
    @line = split;  # split the line to get x_i, y_i, sigma_i
    $x[$n] = $line[0];
    $y[$n] = $line[1];
    $err[$n] = $line[2];
    $err2 = $err[$n]**2;    # compute the necessary sums over the data
    $s += 1 / $err2;
    $sumx  += $x[$n] / $err2 ;
    $sumy  += $y[$n] / $err2 ;
    $sumxx += $x[$n]*$x[$n] / $err2 ;
    $sumxy += $x[$n]*$y[$n] / $err2 ;
    $n++;
}

$delta = $s * $sumxx - $sumx * $sumx ;  # compute the slope and intercept
$c = ($sumy * $sumxx - $sumx * $sumxy) / $delta ;
$m = ($s *  $sumxy - $sumx * $sumy) / $delta ;
$errm = sqrt($s / $delta) ;
$errc = sqrt($sumxx / $delta) ;

printf ("slope     = %10.4f +/- %7.4f \n", $m, $errm); # print the results
printf ("intercept = %10.4f +/- %7.4f \n\n", $c, $errc);

$NDF = $n - 2; # the no. of degrees of freedom is n - no. of fit params
$chisq = 0;    # compute the chi-squared
for ($i = 0; $i < $n; $i++)
{
    $chisq += (($y[$i] - $m*$x[$i] - $c)/$err[$i])**2;
}
$chisq /= $NDF;
printf ("chi squared / NDF = %7.4lf \n", $chisq);
```

Acting with this script on the data in http://young.physics.ucsc.edu/bad-honnef/data.HW3 gives

```
slope     =     5.0022 +/-  0.0024
intercept =     0.9046 +/- 0.2839

chi squared / NDF =  1.0400
```

H.2.2 Python, Writing Out the Formulae by Hand

```
#
# Program written by Matt Wittmann
#
# Usage: "python this_file_name data_file"
#
# Does a straight-line fit to data in "data_file", each line of which contains
# the data for one point, x_i, y_i, sigma_i
#
import fileinput
from math import *

x = []
y = []
err = [] s = sumx = sumy = sumxx = sumxy = 0.

for line in fileinput.input():   # read in the data, one line at a time
    line = line.split()          # split the line
    x_i = float(line[0]);   x.append(x_i)
    y_i = float(line[1]);   y.append(y_i)
    err_i = float(line[2]); err.append(err_i)
    err2 = err_i**2
    s += 1 / err2               # do the necessary sums over data points
    sumx    += x_i / err2
    sumy    += y_i / err2
    sumxx   += x_i*x_i / err2
    sumxy   += x_i*y_i / err2

n = len(x)                        # n is the number of data points
delta = s * sumxx - sumx * sumx # compute the slope and intercept
c = (sumy * sumxx - sumx * sumxy) / delta
m = (s * sumxy - sumx * sumy) / delta
errm = sqrt(s / delta)
errc = sqrt(sumxx / delta)

print "slope     = %10.4f +/- %7.4f " % (m, errm)
print "intercept = %10.4f +/- %7.4f \n" % (c, errc)

NDF = n - 2                     # the number of degrees of freedom is n - 2
chisq = 0.

for i in xrange(n):             # compute chi-squared
    chisq += ((y[i] - m*x[i] - c)/err[i])**2;

chisq /= NDF
print "chi squared / NDF = %7.4lf " % chisq
```

The results are identical to those from the perl script.

H.2.3 Python, Using a Built-In Routine from Scipy

```
#
# Python program written by Matt Wittmann
#
# Usage: "python this_file_name data_file"
#
# Does a straight-line fit to data in "data_file", each line of which contains
# the data for one point, x_i, y_i, sigma_i.
#
# Uses the built-in routine "curve_fit" in the scipy package. Note that this
# requires the error bars to be corrected, as with gnuplot
# from pylab import *
from scipy.optimize import curve_fit

fname = sys.argv[1] if len(sys.argv) > 1 else 'data.txt'
x, y, err = np.loadtxt(fname, unpack=True)  # read in the data
n = len(x)

p0 = [5., 0.1]  # initial values of parameters
f = lambda x, c, m: c + m*x                 # define the function to be fitted
                                                    # note python's lambda notation
p, covm = curve_fit(f, x, y, p0, err)      # do the fit
c, m = p
chisq = sum(((f(x, c, m) - y)/err)**2)   # compute the chi-squared
chisq /= n - 2                                    # divide by no.of DOF
errc, errm = sqrt(diag(covm)/chisq)      # correct the error bars

print "slope     = %10.4f +/- %7.4f " % (m, errm)
print "intercept = %10.4f +/- %7.4f \n" % (c, errc)
print "chi squared / NDF = %7.4lf " % chisq
```

The results are identical to those from the above scripts.

H.2.4 Gnuplot

```
#
# Gnuplot script to plot points, do a straight-line fit, and display the
# points, fit, fit parameters, error bars, chi-squared per degree of freedom,
# and goodness of fit parameter on the plot.
#
# Usage: "gnuplot this_file_name" #
#
The data is assumed to be a file "data.HW3", each line containing
# information for one point (x_i, y_i, sigma_i). The script produces a
# postscript file, called here "HW3b.eps".
#
set size 1.0, 0.6
set terminal postscript portrait enhanced font 'Helvetica,16'
set output "HW3b.eps"
set fit errorvariables  # needed to be able to print error bars
f(x) = a + b * x        # the fitting function
fit f(x) "data.HW3" using 1:2:3 via a, b # do the fit
set xlabel "x"
set ylabel "y"
ndf = FIT_NDF            # Number of degrees of freedom
chisq = FIT_STDFIT**2 * ndf # chi-squared
Q = 1 - igamma(0.5 * ndf, 0.5 * chisq) # the quality of fit parameter Q
#
#  Below note how the error bars are (a) corrected by dividing by
#  FIT_STDFIT, and (b) are displayed on the plot, in addition to the fit
#  parameters, neatly formatted using sprintf.
```

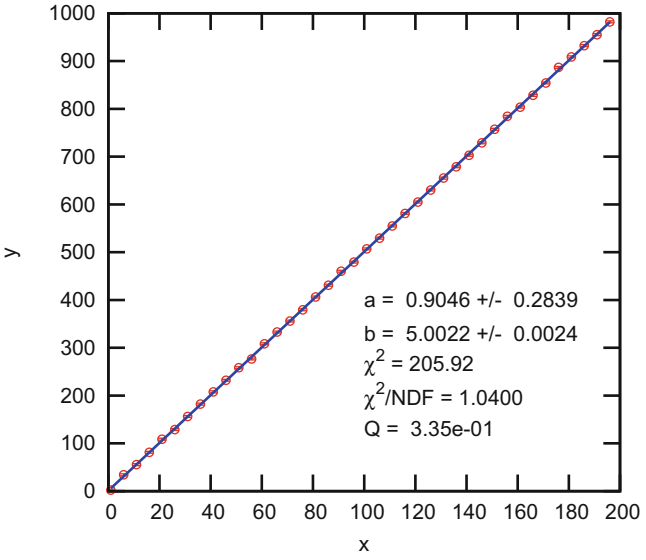

Fig. H.1 Plot showing the data used and the resulting fit to a linear model discussed in Sect. H.2

```
#
set label sprintf("a = %7.4f +/- %7.4f", a, a_err/FIT_STDFIT) at 100, 400
set label sprintf("b = %7.4f +/- %7.4f", b, b_err/FIT_STDFIT) at 100, 330
set label sprintf("{/Symbol c}^2 = %6.2f", chisq) at 100, 270
set label sprintf("{/Symbol c}^2/NDF = %6.4f", FIT_STDFIT**2) at 100, 200
set label sprintf("Q = %9.2e", Q) at 100, 130
plot \                          # Plot the data and fit
"data.HW3" using 1:2:3 every 5 with errorbars notitle pt 6 lc rgb "red" lw 2, \
f(x) notitle lc rgb "blue" lw 4 lt 1
```

The plot shows the result of acting with this gnuplot script on the data in http://young.physics.ucsc.edu/bad-honnef/data.HW3. The results agree with those of the other scripts.

H.3 Scripts For a Fit to a Non-linear Model

We read in lines of data each of which contains three entries x_i, y_i and σ_i. These are fitted to the form

$$y = T_c + A/x^{\omega}, \tag{G.1}$$

to determine the best values of T_c, A and ω.

H.3.1 Python

```
#
# Python program written by Matt Wittmann
#
# Usage: "python this_file_name data_file"
#
# Does a fit to the non-linear model
#
# y = Tc + A / x**w
#
# to the data in "data_file", each line of which contains the data for one point,
# x_i, y_i, sigma_i.
#
# Uses the built-in routine "curve_fit" in the scipy package. Note that this
# requires the error bars to be corrected, as with gnuplot
#
from pylab import *
from scipy.optimize import curve_fit
from scipy.stats import chi2

fname = sys.argv[1] if len(sys.argv) > 1 else 'data.txt'
x, y, err = np.loadtxt(fname, unpack=True)  # read in the data
n = len(x)                                  # the number of data points

p0 = [-0.25, 0.2, 2.8]                      # initial values of parameters
f = lambda x, Tc, w, A: Tc + A/x**w         # define the function to be fitted
                                            # note python's lambda notation
p, covm = curve_fit(f, x, y, p0, err)       # do the fit
Tc, w, A = p
chisq = sum(((f(x, Tc, w, A) - y)/err)**2)  # compute the chi-squared
ndf = n -len(p)                             # no. of degrees of freedom
Q = 1. - chi2.cdf(chisq, ndf)               # compute the quality of fit parameter Q
chisq = chisq / ndf                         # compute chi-squared per DOF
Tcerr, werr, Aerr = sqrt(diag(covm)/chisq)  # correct the error bars

print 'Tc = %10.4f +/- %7.4f' % (Tc, Tcerr)
print 'A  = %10.4f +/- %7.4f' % (A, Aerr)
print 'w  = %10.4f +/- %7.4f' % (w, werr)
print 'chi squared / NDF = %7.4lf' % chisq
print 'Q  = %10.4f' % Q
```

When applied to the data in http://young.physics.ucsc.edu/bad-honnef/data.HW4 the output is

```
Tc =    -0.2570 +-  1.4775
A  =     2.7878 +-  0.8250
w  =     0.2060 +-  0.3508
chi squared / NDF =  0.2541
Q  =     0.9073
```

H.3.2 Gnuplot

```
#
# Gnuplot script to plot points, do a fit to a non-linear model
#
# y = Tc + A / x**w
#
# with respect to Tc, A and w, and display the points, fit, fit parameters,
```

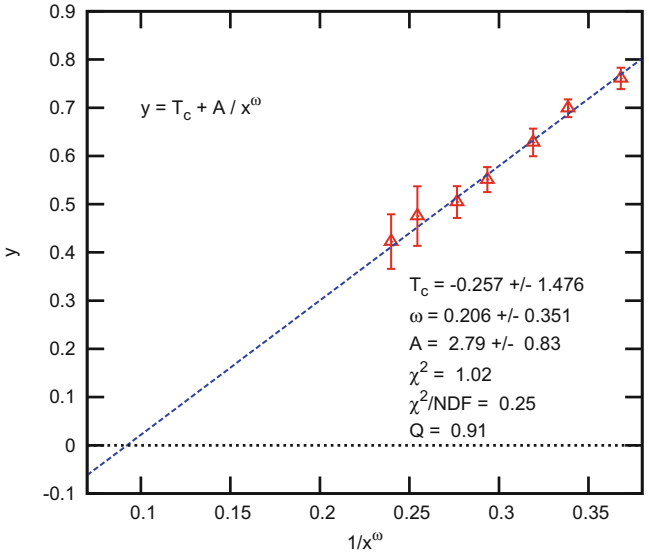

Fig. H.2 Plot showing the data used and the resulting fit to a non-linear model discussed in Sect. H.3

```
# error bars, chi-squared per degree of freedom, and goodness of fit parameter
# on the plot.
#
# Here the data is assumed to be a file "data.HW4", each line containing
# information for one point (x_i, y_i, sigma_i). The script produces a
# postscript file, called here "HW4a.eps".
#
set size 1.0, 0.6
set terminal postscript portrait enhanced
set output "HW4a.eps"
set fit errorvariables       # needed to be able to print error bars
f(x) = Tc + A / x**w              # the fitting function
set xlabel  "1/x^{/Symbol w}"
set ylabel "y"
set label "y = T_c + A / x^{/Symbol w}" at 0.1, 0.7
Tc = 0.3                          # need to specify initial values
A = 1
w = 0.2
fit f(x) "data.HW4" using 1:2:3 via Tc, A, w # do the fit
set xrange [0.07:0.38]
g(x) = Tc + A * x
h(x) = 0 + 0 * x
ndf = FIT_NDF                     # Number of degrees of freedom
chisq = FIT_STDFIT**2 * ndf  # chi-squared
Q = 1 - igamma(0.5 * ndf, 0.5 * chisq) # the quality of fit parameter Q
#
#  Below note how the error bars are (a) corrected by dividing by
#  FIT_STDFIT, and (b) are displayed on the plot, in addition to the fit
#  parameters,  neatly formatted using sprintf.
#
set label sprintf("T_c = %5.3f +/- %5.3f",Tc, Tc_err/FIT_STDFIT) at 0.25, 0.33
set label sprintf("{/Symbol w} = %5.3f +/- %5.3f",w, w_err/FIT_STDFIT) at 0.25, 0.27
set label sprintf("A = %5.2f +/- %5.2f",A, A_err/FIT_STDFIT) at 0.25, 0.21
set label sprintf("{/Symbol c}^2 = %5.2f", chisq) at 0.25, 0.15
set label sprintf("{/Symbol c}^2/NDF = %5.2f", FIT_STDFIT**2) at 0.25, 0.09
set label sprintf("Q = %5.2f", Q) at 0.25, 0.03
```

```
#
# Plot the data and the fit
# plot "data.HW4" using (1/$1**w):2:3 with errorbars notitle lc rgb "red" lw 3 pt 8 ps 1.5, \
g(x) notitle lc rgb "blue" lw 3 lt 2 , \
h(x) notitle lt 3 lw 4
```

The plot shows the result of acting with this gnuplot script on the data at http://young.physics.ucsc.edu/bad-honnef/data.HW4 The results agree with those of the python script above.

The quoted error bars in T_c are clearly ridiculous and arise because the code gives symmetric error bars whereas the variation of χ^2 about the minimum is very asymmetric, as sketched in the right panel of Fig. 3.4. It would be better to get asymmetric error bars for T_c by determining χ^2 as a function of T_c, while optimizing with respect to the other parameters, and then estimating the values of T_c where $\Delta\chi^2 = 1$, see Fig. 3.4 and the discussion in Sect. 3.6. The interested student is invited to do this. Even better would be to do the bootstrap analysis discussed in Sect. 3.7 but this requires the raw data, that is to say the N_i y-values for each data point i which, when averaged, give the results for y_i and σ_i used in the fit. Unfortunately the raw data is not available in this case.

Reference

1. W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical Recipes in C*, 2nd edn. (Cambridge University Press, Cambridge, 1992)