# The Ways of Our Errors

## Optimal Data Analysis
## for Beginners and Experts

©Keith Horne
University of St.Andrews

DRAFT October 28, 2021

## Contents

# Part I
# Optimal Data Analysis

As an observational astronomer, you have survived a long airline flight, a terrifying taxi ride to the summit of a volcano, days of dodgey weather, hours coaxing flakey equipment back into an orderly lifestyle, exhaustion. At last, you attain that exhalted state of resonance with machine and sky. Your equipment is working in miraculous defiance of Murphy's Law. Everything that could go wrong did, but now you have emerged to savour a long clear night plucking data from the sky. Thus you succeed in acquiring an astronomical dataset. After such an ordeal, giving birth to the data, it seems shameful, even criminal, to analyze your data with anything less than optimal methods.

This book outlines a few basic principles, their translation into practical methods, and some detailed examples of optimal data analysis. My goal is to equip you with all the concepts and tools you will need to dig out the information from your data. I want you to understand what works and what doesn't, and why. My hope is in reading this book your understanding will build step by step from basic intuitive concepts to quite advanced techniques. Use the illustrations and examples to see what is going on, and study the equations to make the concepts sharp.

A lot of excellent software is available for analyzing data. Where appropriate, use it. Common user software is a very important resource as the pace of computing and data acquisition continues to accelerate. But you will need to understand what goes on behind the scenes, inside the black boxes, so that you can assess what comes out, relax when it looks right, perk up when it doesn't, and figure out what the ... went wrong and what to do about it.

In research we are always doing something new. "If you know what you are doing, it probably isn't research." So often you will find that the existing software doesn't do quite what you wanted it to do. You could give up ... You could go looking for an expert ... Or, you could use the techniques you've learned from this book to design your own optimal solution to the new data analysis problem.

New types of instrumentation generating new types of data are coming alive all the time. That technology enables the accelerating expansion in our knowledge of the Universe that we enjoy today. New generations of software will be required for calibration and analysis of datasets from those instruments. Creative scientists and programmers with a good understanding of the concepts and techniques developed in this book should be ready to meet these data analysis challenges of the future.

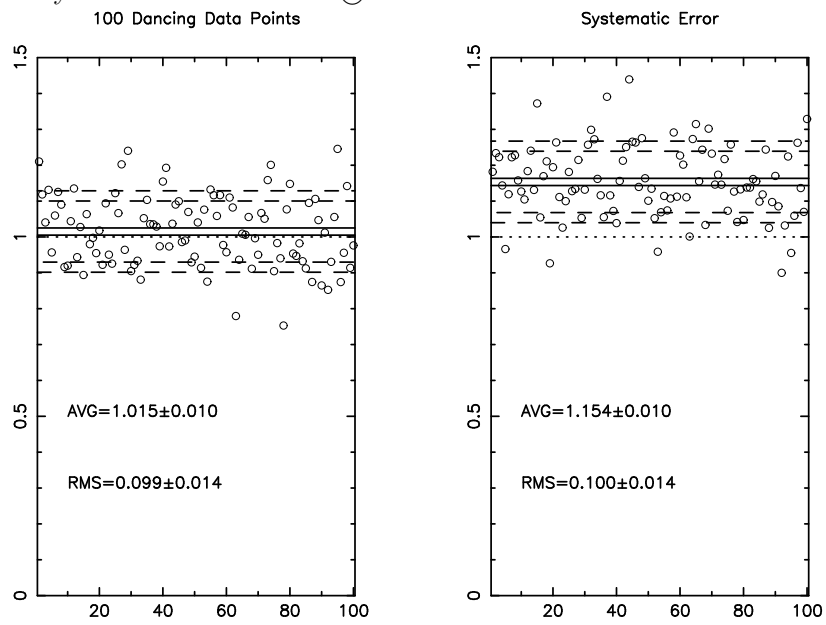


Figure 1: 100 dancing data points with 10% statistical errors. Note that (max-min)/5 is a good rough estimate of the standard deviation that is quick to read by eye from a plot of the data. The mean and standard deviation calculated from the data are given along with their respective error bars. With $N = 100$ data points the statistical error diminishes by $1/\sqrt{N}$, from 10% to 1% in this case. A systematic error with a +15% bias affects the data in the right hand panel, however.

# 1 Probability Concepts

Probability theory is a precise language. It allows us to think and talk clearly about the noise that affects our data, and how that noise affects the answers to the questions that we want to address with the data. Most of the concepts are fairly intuitive, but there is a bit of new jargon to pick up. The aim of this chapter is to present and illustrate many of the concepts, using pictures and examples as well as equations and words.

## 1.1 Fuzzy Numbers and Dancing Data Points

Numbers, as we know, have definite values like 1 or 5/7 or $\pi$. The data points you acquire when you do an experiment are in some sense different from numbers, because they have been affected by random noise. They are only approximate measurements. They are numbers that are a bit fuzzy. In probability theory, such numbers are called *random variables*. We will call them *fuzzy numbers*. Rather than having specific well defined values, you can think of your fuzzy data points as jumping or dancing around at random among all the possible values.

When you do an experiment, you get a specific set of data points, and these are just numbers with definite values. However, you can imagine doing the experiment again, and getting a second somewhat different set of numbers. Each time you repeat the experiment, the data you obtain will be slightly different.

I also find it helpful to visualize the effect of noise by imagining a movie showing a plot of the results from a long series of identical runs of the experiment. In each frame of the movie, you see the outcome of of one of the many identical but independent runs of the experiment. The results change from frame to frame because the random noise is different in each run of the experiment. The movie reveals how the data points dance and jitter about at random by various amounts as a result of noise processes that affect the results of the experiment.

In the usual case, each data point moves around at random, unaware of the ways that other points are moving. But it's also possible that patterns are present. One data point may go up every time its neighbor goes down. Such correlations may be present over a wide range of data points, whole sets of data points moving in concert. Many different patterns may be superimposed. How to describe the possibilities? Probability theory is the tool, the language, that lets us describe the precise character of that intricate dance.

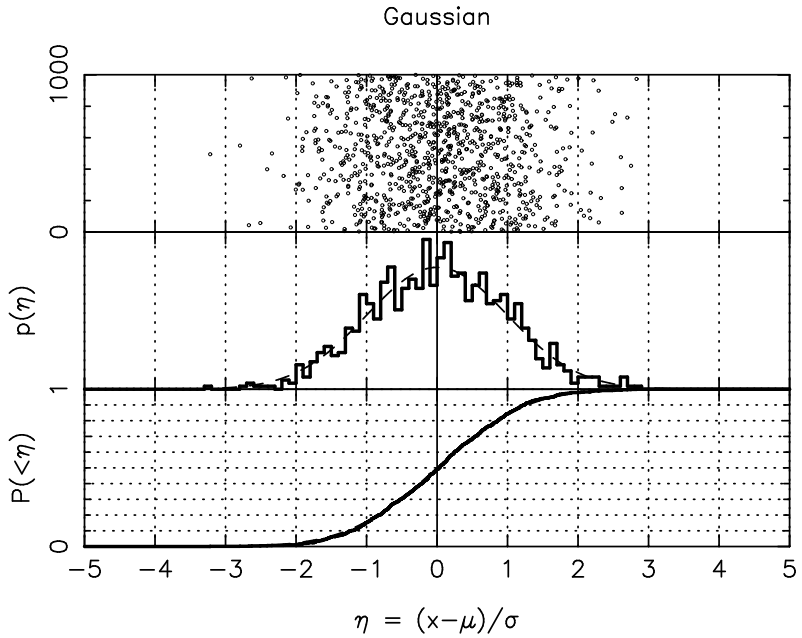


Figure 2: Data points with a Gaussian probability map with mean $\mu = 0$ and standard deviation $\sigma = 1$. The 1000 data points (top panel) are binned (middle panel) for comparison with the probability map $\mathrm{p}(\,\eta\,)$. This is then integrated (lower panel) for comparison with the cumulative probability $\mathrm{P}(<\eta\,)$.

## 1.2 Systematic and Statistical Errors – Bias and Variance

It is important to understand the difference between two types of errors that affect your data – statistical errors and systematic errors.

*Statistical errors* arise from the random noise that affects your data. In Fig. 1 the statistical noise is 10/If you can repeat the experiment over and over again, can reduce the impact of statistical errors. Each repetition gives an independent measurement, and by averaging $N$ such measurements the random statistical error in the average diminishes by a factor $1/\sqrt{N}$. In Fig. 1 averaging 100 measurements reduces the statistical noise to from 10% to 1%. With more repetitions the accuracy can improve still further. In principle you can increase $N$ to make the statistical errors as small as you like.

*Systematic errors* arise when the average value obtained from the measurements differs from the true value. That difference is called a *bias*. In Fig. 1b the data points have a bias of +0.15. While the statistical errors diminish as we average results from many repeat measurements, the bias remains unchanged. Thus it is best to design experiments and data analysis methods that avoid systematic errors completely. Such methods are called *unbiased* methods.

When a bias cannot be avoided, the next best alternative is to provide additional data in the form of calibrations to measure the bias so that it can be removed. A good overall strategy, then, is to use unbiased methods that eliminate systematic errors entirely, and then to optimize the methods to minimize the statistical errors.

## 1.3 Probability Maps

A 1-dimensional *probability map* $\mathrm{p}(\,x\,)$ is a *distribution* defined by its integrals over possible ranges of the variable $x$. If you integrate $\mathrm{p}(\,x\,)$ over some range, say $a \le x \le b$, the result is the *probability* that $x$ lies in that range:

$$\mathrm{P}(\,a \le x \le b\,) \equiv \int_a^b \mathrm{p}(\,x\,)\;dx\;. \tag{1}$$

To qualify as a probability map, the distribution $\mathrm{p}(\,x\,)$ must have two special properties. First, integrating $\mathrm{p}(\,x\,)$ over any range must always yield a probability between 0 and 1 inclusive. Second, when we are 100% certain that $x$ is somewhere in the range $-\infty \le x \le \infty$, the probability map is normalized accordingly:

$$\int_{-\infty}^{+\infty} \mathrm{p}(\,x\,)\;dx = 1\;. \tag{2}$$

Other names you may hear being used to refer to what we are calling a probability map are probability density, probability distribution, probability distribution function, pdf, and differential probabilities. We try to use the term probability map.

Since $\mathrm{p}(\,x\,)$ is a distribution, defined in terms of its integrals, the same information is contained in the *cumulative* probability map

$$\mathrm{P}(<x\,) \equiv \int_{-\infty}^{x} \mathrm{p}(\,u\,)\;du\;. \tag{3}$$

The cumulative probability $\mathrm{P}(<x\,)$ rises monotonically from 0 at $x=-\infty$ to 1 at $x=+\infty$, increasing or remaining constant, but never decreasing.

Mathematicians take care to make a distinction between *distributions* and *functions*. You are probably familiar with functions. A function $f$ has a definite value $f(x)$ at each specific value of $x$. A distribution, on the other hand, is defined only in terms of integrals over $x$. It may or may not have a well-defined value $\mathrm{p}(\,x\,)$. We write the probability map $\mathrm{p}(\,x\,)$ as if it were a function, but remember that only the probabilities, the integrals of $\mathrm{p}(\,x\,)$, are meaningful.

An important example of a distribution that is NOT a function is the *Dirac distribution* $\delta(x)$, defined by its integrals

$$\int_a^b \delta(x)\;dx \equiv \begin{cases} 1 & \text{if } a\le 0\le b \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

One way to visualize $\delta(x)$ is to think of a very tall spike at $x=0$ with height $H$ width $1/H$ so that the area is 1. The spike must then grow infinitely tall and narrow while keeping its area always 1.

If $\delta(x)$ were a function, it would have a well-defined value $\delta(0)$ at $x=0$. You might be tempted to say that this value is $\delta(0)=\infty$. But you would be wrong, because you can also equally well visualize a Dirac distribution as 2 tall spikes each of area 1/2 and displaced by tiny amounts on either side of $x=0$. In this alternative visualization, we have $\delta(0)=0$. Spooky, no?

A probability map $\mathrm{p}(\,x\,)$ is always a distribution, but it can also be a function IF it has a well defined value at every $x$. A very important probability map that is also a function is the *Gaussian distribution* :

$$\mathrm{p}(\,x|\mu,\sigma\,) = \frac{e^{-\eta^2/2}}{\left(2\pi\sigma^2\right)^{1/2}}\;, \tag{5}$$

where $\eta=(x-\mu)/\sigma$. The notation $\mathrm{p}(\,x|\mu,\sigma\,)$ means that the probability map for the fuzzy number $x$ has has 2 parameters: $\mu$, the mean (or centroid), and $\sigma$, the standard deviation (or dispersion) of $x$. The Gaussian has a symmetric bell-shaped probability peak centred at $x=\mu$. The half-width of the bell is approximately $\sigma$. The full-width at half-maximum is $(8\ln 2)^{1/2}\,\sigma$. In data analysis, the Gaussian is the most important probability map because in most cases it provides a satisfactory description of the effect of noise, causing a data value $x$ to jitter around its mean value $\mu$ by a typical amount $\sigma$. We'll discuss the precise meaning of this later.

The probability map $\mathrm{p}(\,x\,)$ gives the complete description of how noise affects the data value $x$. Integrals of $\mathrm{p}(\,x\,)$ give the probability that the noisy dancing data value falls in any range $a\le x\le b$ that you may wish to consider. Note that when $\mathrm{p}(\,x\,)$ is a function, the probability assigned to each specific value of $x$ is infinitessimally tiny. You must integrate over a finite range of $x$ to build up a finite probability.

Probability maps can also attach a finite probability to discrete values of $x$. For example, you may toss a coin and say the result is $x=-1$ if it comes up heads and $x=+1$ if it comes up tails. This random process is described by a discrete probability map

$$\mathrm{p}(\,x\,) = \tfrac{1}{2}\delta(x+1) + \tfrac{1}{2}\delta(x-1)\;. \tag{6}$$

Assuming that heads and tails are equally likely to occur, the coin toss probability map has 2 Dirac spikes, one at $x=-1$ and one at $x=+1$, each with probability 1/2. If the coin is not fair, then heads may come up more often or less often than tails. You can describe the outcome of tossing such a coin by

$$\mathrm{p}(\,x\,) = P_H\delta(x+1) + P_T\delta(x-1)\;, \tag{7}$$

where $P_H$ and $P_T$ are the probabilities that the coin comes up heads and tails respectively. Since negative probabilities are forbidden, and since the coin must come up something (presumably), then $0\le P_H\le 1$ and $P_T=1-P_H$.

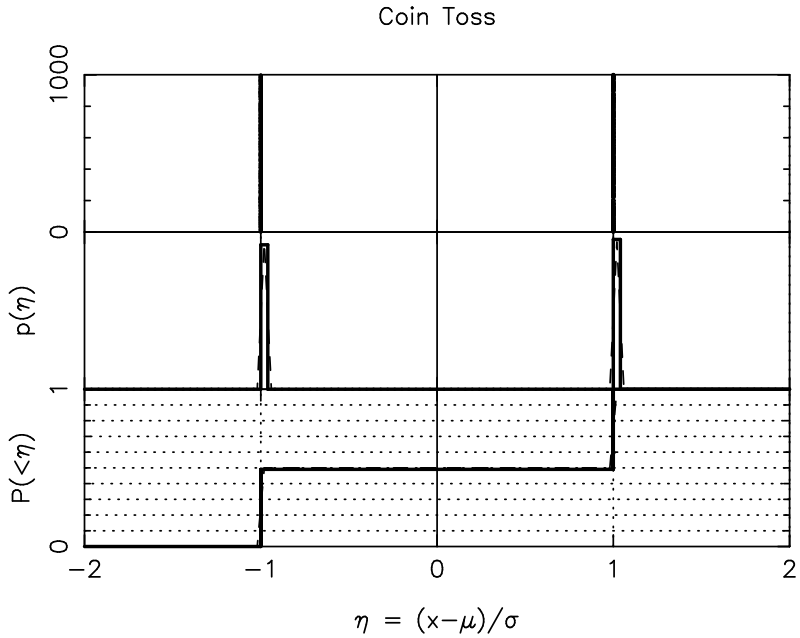


Figure 3: 1000 random coin tosses (top panel) are binned (middle panel) for comparison with the probability map $\mathrm{p}(\,\eta\,)$. This is then integrated (lower panel) for comparison with the cumulative probability $\mathrm{P}(<\eta\,)$.

## 1.4 Mean, Variance, and Standard Deviation

The *mean value* and *standard deviation* are precise and useful measures of the location and width of a probability map. The concepts are exactly analogous to the physical concepts of *centre of mass* and *moment of inertia*, which are useful in describing the way forces and torques affect the translation and rotation of a massive body.

We often use the mean and standard deviation as a convenient shorthand to describe the result of a measurement, and the accuracy of that measurement, i.e.

$$x \approx \mu \pm \sigma \tag{8}$$

means that the mean value of $x$ is $\mu$, and its standard deviation is $\sigma$. This shorthand omits information about the detailed shape of the probability map $\mathrm{p}(\,x\,)$, but gives us a good idea of the range of values of $x$ that are likely to occur.

The *mean value* of a probability map is analogous to the centre of mass. If the probability map $\mathrm{p}(\,x\,)$ is the distribution of mass along the length of a plank, then the mean value $\langle\, x\,\rangle$ corresponds to the pivot point where the plank will balance. The formal definition is

$$\langle\, x\,\rangle \equiv \int_{-\infty}^{+\infty} x\ \mathrm{p}(\,x\,)\ dx \tag{9}$$

Note here the angle bracket notation $\langle\,\cdot\,\rangle$, shorthand for a probability-weighted average.

The *variance* is analogous to the moment of inertia of a mass distribution:

$$\mathrm{Var}[\,x\,] \equiv \int_{-\infty}^{+\infty} (x - \langle\, x\,\rangle)^2\, \mathrm{p}(\,x\,)\ dx = \Big\langle\, (x - \langle\, x\,\rangle)^2 \,\Big\rangle\ . \tag{10}$$

Note again the use of $\langle\,\cdot\,\rangle$ in the last expression.

The *standard deviation* is the square root of the variance:

$$\sigma\,(x) \equiv \sqrt{\mathrm{Var}[\,x\,]}\ . \tag{11}$$

A variety of names and notations are in use to refer to the mean value and standard deviation. Other names you may encounter for the mean are the *centroid*, the *expected value*, $\mathrm{E}[x]$, $\mu(x)$, $\langle\, x\,\rangle$, $\bar{x}$. The standard deviation may be called the *dispersion*, the *rms*, the $\sigma$, or $\sigma(x)$. The variance may be called the $\sigma^2$, $\mathrm{Var}[\,x\,]$ or $\sigma^2(x)$. We will try to use $\langle\, x\,\rangle$ and $\mathrm{Var}[\,x\,]$ to refer to the mean value and variance, and $\sigma(x)$ for the standard deviation.

The mean and standard deviation are by no means the only possible ways to measure the location and width of a probability map.

## 1.5 Median and Quantiles

The mean and median are two different ways to define the center of a probability map. The *median* is the point that divides the probability into two equal halves:

$$\mathrm{P}(\, x \le \mathrm{Med}[\, x\,]\,) = \int_{-\infty}^{\mathrm{Med}[\, x\,]} \mathrm{p}(\, x\,)\; dx = \tfrac{1}{2}\;. \tag{12}$$

A *quantile*, $x_q$, splits the probability unequally:

$$\mathrm{P}(\, x \le x_q\,) = \int_{-\infty}^{x_q} \mathrm{p}(\, x\,)\; dx = q\;. \tag{13}$$

The probability is $q$ that $x \le x_q$, and $1-q$ that $x > x_q$. The median, $\mathrm{Med}[\, x\,] = x_{0.5}$, is the 50% quantile.

For a symmetric probability map, like the Gaussian, the mean and median are identical, $\langle\, x\,\rangle = x_{0.5}$. The mean and median differ, however, whenever the probability map is asymmetric. If $\mathrm{p}(\, x\,)$ has a large tail to high values, then the mean is larger than the median.

For a Gaussian, the points $x = \mu \pm \sigma$ are roughly 17% and 83% quantiles. The interval $\mu - \sigma \le x \le \mu + \sigma$ contains roughly 67% of the probability. For and $\mathrm{p}(\, x\,)$ you can define a $\pm\sigma$ interval using quantiles: the interval $x_{0.17} \le x \le x_{0.83}$ always encloses 67% of the probability.

The median and other quantiles offer interesting alternatives to the mean and standard deviation. They are sometimes called *robust* statistics because they are relatively insensitive to what happens way out in the wings of the probability map. For example, suppose that $\mathrm{p}(\, x\,)$ has a long tail on the high side. This actually happens, for example, when cosmic rays can hit your detector causing one or more data values to become very large. The cosmic ray hits may affect only a tiny fraction of your data, but when they strike their impact is huge. A probability map allowing for cosmic ray hits has a long tail to high values that has a small fraction of the probability. That tail has a strong effect on the mean and standard deviation. The tail can shift them to values so large that they no longer give a reasonable impression of the measurement. When cosmic ray hits are a factor, the median is often recommended instead of the mean to express the result of a measurement. We'll discuss the limitations of this practice, and alternatives that achieve better results, later on.

The median and other quantiles are also interesting because they are defined in a way that is in some sense coordinate-independent, and this makes it easier to do calculations with your measurements. When you plug your measurements into equations to compute other quantities of interest, the means and standard deviations of those quantities are not what you might at first expect.

Suppose you want to measure the mass of a star or black hole $M$ by measuring the orbital period $P$ and orbital velocity $V$ of some object that is orbiting around it. You can calculate $M$ using Kepler's law

$$M(V) = \frac{V^3 P}{2\pi G}\;, \tag{14}$$

where $G$ is the gravitational constant. Since $P$ is generally known to high accuracy, the uncertainty in $M$ stems mainly from the measurement error in $V$. You may be surprised to learn (see Fig. 4) that

$$\langle\, M\,\rangle > M(\langle\, V\,\rangle^3)\;. \tag{15}$$

You must cube the measurement of $V$ to estimate the mass $M$. The positive curvature of that cubic transformation skews the probability map $\mathrm{p}(\, M\,)$ to high values. When $V$ is high, the corresponding $M$ is higher still, and when $V$ is low, the corresponding $M$ is not so low.

The median and other quantiles don't have this problem. If the velocity $V$ exceeds $\mathrm{Med}[\, V\,]$ half of the time then the mass $M(V)$ will exceed $\mathrm{Med}[\, M\,]$ half of the time. For any velocity quantile $V_q$, the corresponding mass quantile $M_q$ is obtained by Kepler's law

$$M_q = M(V_q^3)\;. \tag{16}$$

The median and other quantiles are preserved through calculations.

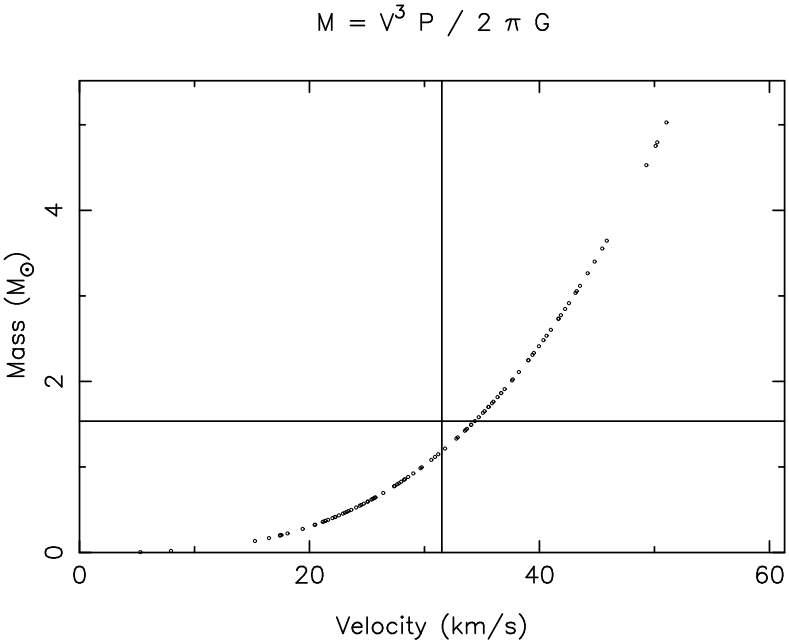


Figure 4: Mass estimates derived from velocity measurements. The average of masses calculated from the cube of the individual velocity measurements is higher than the mass calculated from the cube of the mean velocity. The median, and other quantiles, don't suffer from this problem.

## 1.6 Fuzzy Algebra

You know how to add, subtract, multiply and divide ordinary numbers. What happens if you do arithmetic with fuzzy data values? When you calculate something from one or more data points, the result is a new fuzzy number, characterized by its own centroid and dispersion.

How do the fuzzy data values used in a calculation carry through to affect the fuzzy result? With a couple of simple rules, and a bit of practice, you should become confident in your ability to perform all sorts of calculations with your fuzzy data values while keeping track of the fuzziness of the results. This technique is often called *propagation of errors*.

First, let's multiply a data value $X$ by a constant $a$. This produces a *scaled data value* $aX$. Its centroid and dispersion, along with the entire probability map, are simply stretched by a factor $a$:

$$\langle\, aX \,\rangle = a\langle\, X \,\rangle \;, \qquad\qquad \mathrm{Var}[\, aX \,] = a^2 \mathrm{Var}[\, X \,] \;. \tag{17}$$

Note that stretching increases the dispersion $\sigma$ by a factor $a$, and so the variance $\sigma^2$ scales by a factor $a^2$. The shape of the probability map is unaffected by scaling.

Now, let's add together two data points, $A$ and $B$. The mean and variance of the sum and difference are:

$$\langle\, A \pm B \,\rangle = \langle\, A \,\rangle \pm \langle\, B \,\rangle \;, \qquad\qquad \mathrm{Var}[\, A \pm B \,] = \mathrm{Var}[\, A \,] + \mathrm{Var}[\, B \,] \pm 2\mathrm{Cov}[\, A \,,\, B \,] \;. \tag{18}$$

The centroids add and subtract just like ordinary numbers. But the variances combine in a more complicated way.

If $A$ and $B$ are *independent*, this means that $\mathrm{Cov}[\, A \,,\, B \,] = 0$. In this case *variances add in quadrature*. Note that $\mathrm{Var}[\, A - B \,] = \mathrm{Var}[\, A \,] + \mathrm{Var}[\, B \,]$, not $\mathrm{Var}[\, A \,] - \mathrm{Var}[\, B \,]$. To consider a specific example, if $A \approx 1 \pm 1$ and $B \approx 2 \pm 2$, then $A + B \approx 3 \pm \sqrt{5}$ and $A - B \approx -1 \pm \sqrt{5}$. Try a few examples for yourself until you get the hang of it.

If $A$ and $B$ are *not independent*, then $\mathrm{Cov}[\, A \,,\, B \,] \neq 0$. In this case $\mathrm{Var}[\, A \pm B \,]$ may be either larger or smaller than in the independent case, since $\mathrm{Cov}[\, A \,,\, B \,]$ may be either positive or negative.

It is usually a good idea to consider simple examples or limiting cases to check that you understand any new result. This is also helpful in spotting errors in calculations. In this case, let's add a data point to itself. We know that since $A + A = 2A$, the result should be

$$\mathrm{Var}[\, A + A \,] = \mathrm{Var}[\, 2A \,] = 4\mathrm{Var}[\, A \,] = 4\sigma^2 \;. \tag{19}$$

This is a special case in which $A = B$, so that $\mathrm{Var}[\, A \,] = \mathrm{Var}[\, B \,] = \mathrm{Cov}[\, A \,,\, B \,] = \sigma^2$. We therefore find, for $A = B$, that

$$\mathrm{Var}[\, A + B \,] = \mathrm{Var}[\, A \,] + \mathrm{Var}[\, B \,] + 2\mathrm{Cov}[\, A \,,\, B \,] = \sigma^2 + \sigma^2 + 2\sigma^2 = 4\sigma^2 \;, \tag{20}$$

confirming that the general expression gives the correct result for this special case. Similarly, since $A - A = 0$ we are glad to find, for $A = B$, that

$$\mathrm{Var}[\,A - B\,] = \mathrm{Var}[\,A\,] + \mathrm{Var}[\,B\,] - 2\mathrm{Cov}[\,A\,,\,B\,] = \sigma^2 + \sigma^2 - 2\sigma^2 = 0\ . \tag{21}$$

You can use the above rules to work out the mean and variance of $aA + bB$, or any other linear combination of data values $X_i$ multiplied by coefficients $a_i$:

$$\left\langle \sum_i a_i\ X_i \right\rangle = \textstyle\sum_i a_i\ \langle\, X_i \,\rangle\ , \qquad\qquad \mathrm{Var}[\,\textstyle\sum_i a_i\ X_i\,] = \textstyle\sum_i \sum_j a_i\ a_j \mathrm{Cov}[\,X_i\,,\,X_j\,]\ . \tag{22}$$

## 1.7 Why Gaussians are Special

If a fuzzy number $X$ has mean $\mu$ and variance $\sigma^2$, a shorthand notation for this is $X \approx \mu \pm \sigma$. If $X$ has a Gaussian probability map, we write

$$X \sim G(\mu, \sigma^2) \approx \mu \pm \sigma\ . \tag{23}$$

Add up a large number $N$ of independent fuzzy numbers $X_i$. Regardless of the shapes of their individual probability maps, the sum, always tends to evolve toward a Gaussian probability map. We write this as

$$\sum_i X_i \approx \sum_i \mu_i \pm \sigma_i \to G\left(\sum_i \mu_i, \sum_i \sigma_i^2\right)\ . \tag{24}$$

This miracle is known as the *Central Limit Theorem*, or the *Law of Large Numbers*. Its proof involves using a Taylor series expansion of $\log \mathrm{p}(\sum X_i)$ near its peak value. The Gaussian probability map arises by truncation of the Taylor series to quadratic terms, arguing that the higher terms tend to zero as $N \to \infty$. The quadratic term survives because we know from fuzzy algebra that the first and second moments of the independent fuzzy numbers are conserved:

$$\left\langle \sum_i X_i \right\rangle = \textstyle\sum_i \langle\, X_i \,\rangle = \sum_i \mu_i\ , \qquad\qquad \mathrm{Var}[\,\textstyle\sum_i X_i\,] = \textstyle\sum_i \mathrm{Var}[\,X_i\,] = \sum_i \sigma_i^2\ . \tag{25}$$

The higher moments are not conserved in this way. Information about the higher moments is erased when large numbers of data points are added (or averaged) in this way.

Gaussians are special because errors often arise from accumulation of large numbers of small independent effects. For example, measurement errors caused by thermal noise in electronic measuring equipment arise as a result of a large number of random jostlings of molecules. A Gaussian probability map is therefore often a very good approximation to the true error distribution. Even when this isn't true for individual data points, it will become true if we average together a large number of independent data points.

## 1.8 Why $\chi^2$ is Special

$\chi^2$ is special because Gaussians are special. Start with a standard Gaussian fuzzy number $G(0,1)$, which has a Gaussian probability map with mean 0 and variance 1. Square it, and the result, by definition, is a $\chi^2_1$ fuzzy number, with 1 degree of freedom. This has mean $\langle\, \chi^2_1 \,\rangle = 1$ and variance $\mathrm{Var}[\,\chi^2_1\,] = 2$. We write

$$G(0,1)^2 \sim \chi^2_1 \approx 1 \pm \sqrt{2}\ . \tag{26}$$

Now add together $N$ independent squared standard Gaussians to obtain a $\chi^2_N$ fuzzy number with $N$ degrees of freedom, which has mean $\langle\, \chi^2_N \,\rangle = N$ and variance $\mathrm{Var}[\,\chi^2_N\,] = 2N$. Write this as

$$\sum^N G(0,1)^2 \sim \chi^2_N \approx N \pm \sqrt{2N}\ . \tag{27}$$

Finally, rescale the $\chi^2$ to a mean of 1, dividing the $\chi^2_N$ by $N$, to obtain a *reduced* $\chi^2$ with $N$ degrees of freedom, denoted $\chi^2/N$, which has mean 1 and variance $2/N$. We write

$$\frac{\sum^N G(0,1)^2}{N} \sim \chi^2/N \approx 1 \pm \sqrt{2/N}\ . \tag{28}$$

The above results summarize the relationship between Gaussian and $\chi^2$ probability maps. They are important for data analysis because the Central Limit Theorem makes Gaussians special, and thus $\chi^2$ is special too. Here's why:

The *errors*

$$\epsilon_i \equiv X_i - \langle\, X_i \,\rangle \tag{29}$$

represent the amounts by which the noisy data values $X_i$ depart from their true values $\langle\, X_i \,\rangle$. We usually don't know $\langle\, X_i \,\rangle$ – if we did then there would be nothing to learn from the noisy measurement! We formulate a *model* which predicts $\langle\, X_i \,\rangle = \mu_i(\alpha)$, depending on some set of parameters $\alpha$. For each data point we can then calculate *residuals*

$$\epsilon_i(\alpha) \equiv X_i - \langle\, \mu_i(\alpha) \,\rangle \ . \tag{30}$$

The errors $\epsilon_i$ and residuals $\epsilon_i(\alpha)$ are not exactly the same. For example, we can calculate the residuals, $\epsilon_i(\alpha)$, as functions of the parameters $\alpha$, but we never know the true errors. The residuals also jitter by a bit less than the errors. If we fit a model with $p$ parameters to $N$ data points, then the variance of the residuals tends to be a bit smaller, by a factor $(N-p)/N$, than the variance of the errors. This occurs because in the optimized fit the predicted data values $\mu_i(\alpha)$ tend to chase the noise in the data points. The $N$ degrees of freedom in the residuals is reduced to $N-p$ because fitting $p$ parameters should remove $p$ degrees of freedom. We won't worry too much about the distinction between errors and residuals, because we hope to be dealing with a good model, so that once the parameters are optimised the residuals are good approximations to the true errors.

Divide each error by its standard deviation $\sigma_i$ to obtain *normalized errors*

$$\eta_i \equiv \frac{\epsilon_i}{\sigma_i} = \frac{X_i - \langle\, X_i \,\rangle}{\sigma_i} \ . \tag{31}$$

The normalized errors $\eta_i$ have mean 0 and standard deviation 1. We write $\epsilon_i \approx 0 \pm \sigma_i$, and $\eta_i \approx 0 \pm 1$.

Divide each residual by its standard deviation $\sigma_i$, to obtain *normalized residuals*

$$\eta_i(\alpha) \equiv \frac{\epsilon_i(\alpha)}{\sigma_i(\alpha)} = \frac{X_i - \mu_i(\alpha)}{\sigma_i(\alpha)} \ . \tag{32}$$

We generally do not know the true $\sigma_i$, but our model includes a prediction for the *error bar* $\sigma_i(\alpha)$, which might or might not depend on the parameters $\alpha$.

Square each of the normalized residuals and add them up over $N$ data points to obtain the *"chi-square" statistic*,

$$\chi^2(\alpha) \equiv \sum_{i=1}^{N} \eta_i(\alpha) = \sum_{i=1}^{N} \left( \frac{X_i - \mu_i(\alpha)}{\sigma_i(\alpha)} \right)^2 \ . \tag{33}$$

This measures the *badness of fit*. Optimize the fit by adjusting the $p$ parameters $\alpha$ to minimize $\chi^2(\alpha)$. The minimum occurs at $\alpha = \hat{\alpha}$. The resulting minimum value,

$$\chi^2_{\rm min} \equiv \chi^2(\hat{\alpha}) \sim \chi^2_{N-p} \approx (N-p) \pm \sqrt{2(N-p)} \ , \tag{34}$$

is a fuzzy number that should closely resemble a $\chi^2$ with $N-p$ *degrees of freedom* – $N$ degrees of freedom from the errors in the $N$ data points, minus $p$ degrees of freedom from the $p$ parameters. Finally, divide the minimum $\chi^2$ by $N-p$ and you have the *reduced* $\chi^2$ *statistic*,

$$\frac{\chi^2_{\rm min}}{N-p} \approx 1 \pm \sqrt{2/(N-p)} \ . \tag{35}$$

This is a convenient statistic for testing the validity of the model because it should be close to 1 if the model is correct and may be too high if the model is wrong, or too low if the error bars are too large.

Because the errors affecting data points are often independent and well described by Gaussian probability maps, the badness-of-fit statistic $\chi^2_{\rm min}$, after fitting $p$ parameters to $N$ data points, is often well described by a $\chi^2$ distribution with $N-p$ degrees of freedom. This is why $\chi^2$ is special.

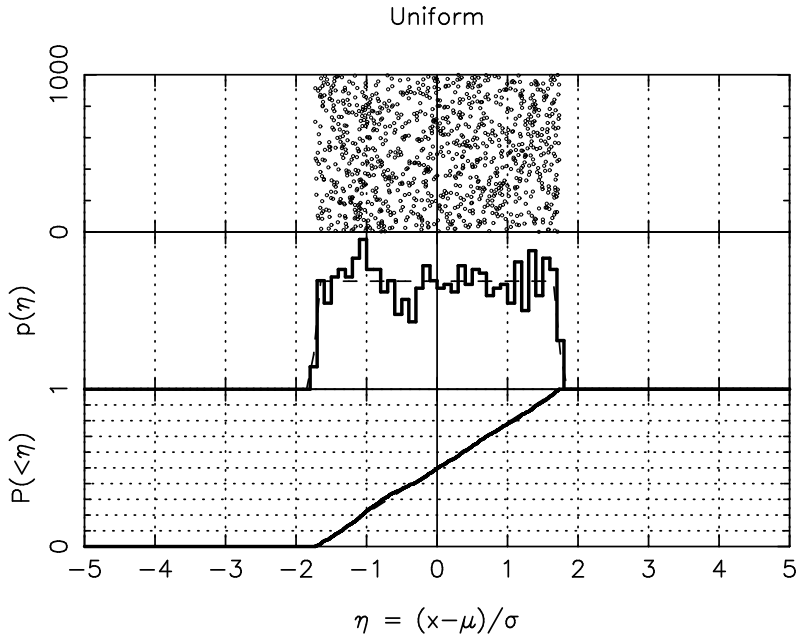


Figure 5: Uniform random numbers with mean $\mu = 0$ and standard deviation $\sigma = 1$. The 1000 random numbers (top panel) are binned (middle panel) for comparison with the probability map. This is then integrated (lower panel) for comparison with the cumulative probability.

# 2 A Menagerie of Probability Maps

You may enjoy becoming somewhat intimate with certain probability functions that frequently arise in practical applications. The selection here, by no means exhaustive, includes most of the functions that turned up in my data analysis career. I'd like to introduce you to some of their interesting quirks and secrets, which may help them to become your familiar friends, as they have for me.

I'll also arm you with code so that you can see how to generate your own pseudo-random number sequences for each case. These will be important later for Monte Carlo techniques, which use pseudo-random number sequences generated by the computer to simulate random noise processes in Nature. Be encouraged to play around with these.

## 2.1 Boxcar or Uniform (ranu.for)

The simplest is a uniform random variable, also known (to Americans) as a "boxcar". This has a flat-topped probability map (Fig. 5), with probability spread out uniformly between two limits $a$ and $b$,

$$\mathrm{p}(\,x|a,b\,) = \begin{cases} 1/|\,b-a\,| & \text{if} \quad a < x < b\ , \\ 0 & \text{otherwise} \quad . \end{cases} \tag{36}$$

Obviously, the mean value is half-way between the endpoints,

$$\langle\, x\,\rangle = \frac{(a+b)}{2}\ . \tag{37}$$

You can integrate $x^2$ to find that the standard deviation is $1/\sqrt{3} \approx 0.577$ of the way from the centre to the edge,

$$\sigma^2(x) = \left\langle\, (x - \langle\, x\,\rangle)^2\,\right\rangle = \frac{(b-a)^2}{12}\ . \tag{38}$$

Many computer programming languages helpfully include some means of generating sequences of **pseudo-random numbers**. The numbers are only pseudo-random because a completely deterministic algorithm is used to generate the sequence.

`FORTRAN` provides a function subroutine that you can call repeatedly to generate a sequence of **pseudo-random numbers** that are uniformly distributed between 0 and 1. You pass a **seed integer** `iseed` to the function subroutine `ran`, which returns the uniform pseudo-random number and a new value of `iseed` for use on the next call. Thus the code fragment

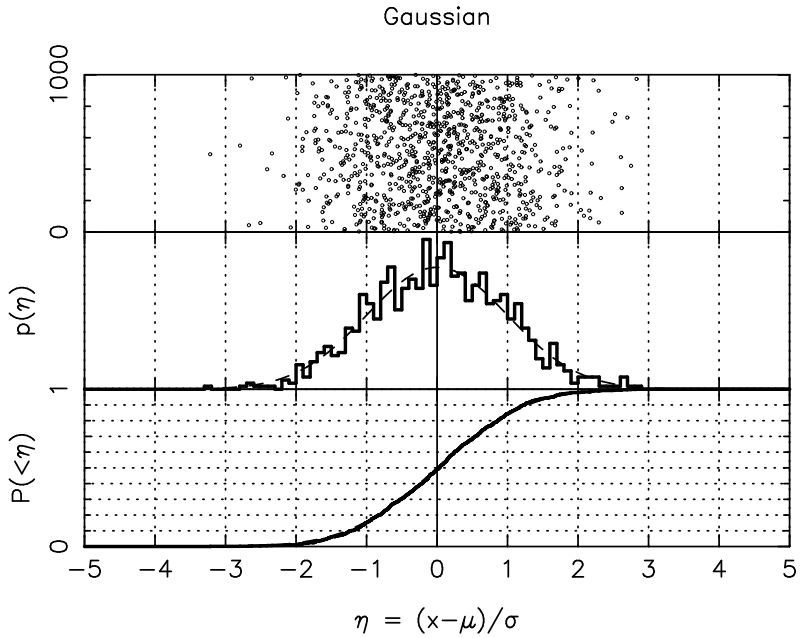


Figure 6: Gaussian random numbers with mean $\mu = 0$ and standard deviation $\sigma = 1$. The 1000 random numbers (top panel) are binned (middle panel) for comparison with the probability map. This is then integrated (lower panel) for comparison with the cumulative probability.

```
*****************************************************
        iseed = 436346
      do i = 1 , n
        x( i ) = ran( iseed )
      end do
*****************************************************
```

fills the array `x` with the sequence of pseudo-random numbers. The value of `iseed` changes each time around the loop. If you start with the same seed, you get the same sequence of random numbers. If you choose a different seed, you get a different sequence.

Because computers represent integers as 32-bit binary numbers, there are only a finite number, $2^{32} \sim 4.3 \times 10^9$, of different integers that can be used for the seed. The pseudo-random sequence must therefore eventually repeat, but not for a very long time.

You can easily shift and scale the random number to match any desired range $a$ to $b$,

```
*************************************************
        function ranu( a, b, iseed )
        ranu = a + ( b - a ) * ran( iseed )
        return
        end
*************************************************
```

The function `ranu` will then work just like `ran` except that the random numbers it returns are uniform over $a$ to $b$ rather than 0 to 1.

## 2.2 Gaussian or Normal (rang.for)

Gaussians are most important because measurement errors are usually Gaussian, or at least approximately Gaussian near their probability peak. Telescopic images of stars are blurred into 2-dimensional Gaussian profiles by the series of small perturbations produced in passing through the turbulent terrestrial atmosphere.

The Gaussian probability map (Fig. 6) is

$$\mathrm{p}(\,x|\mu,\sigma\,) = \frac{\exp\left\{\,-\eta^2/2\,\right\}}{\left(2\pi\sigma^2\right)^{1/2}}\ , \tag{39}$$

where

$$\eta = \frac{x-\mu}{\sigma}\ . \tag{40}$$

The two parameters are the mean $\mu$ and standard deviation $\sigma$. A standard Gaussian will have $\mu = 0$ and $\sigma = 1$.

Integrating the Gaussian probability map, we obtain its cumulative probability,

$$\mathrm{P}(<x|\mu,\sigma\,) = \int_{-\infty}^{x} \exp\left\{\,-\frac{1}{2}\left(\frac{u-\mu}{\sigma}\right)^2\right\}\frac{du}{\left(2\pi\sigma^2\right)^{1/2}} \equiv \mathrm{Erf}\left(\frac{x-\mu}{\sigma}\right)\ . \tag{41}$$

There is no analytic expression, but the integral is important enough to be given its own name, "The Error Function".

Derivatives of the Gaussian profile with respect to the parameters are

$$\frac{\partial P}{\partial \mu} = \frac{\eta\ P}{\sigma}\ , \qquad \frac{\partial P}{\partial \sigma} = \frac{(\eta^2-1)P}{\sigma}\ . \tag{42}$$

According to the Central Limit Theorem, when you add up a large number of independent random variables, regardless of their type, the result is close to a Gaussian. You could use this to generate approximate Gaussian noise by summing up a long sequence of uniform random numbers.

```
**************************************************
        function rang( avg, rms, iseed )
        real*8 sum
        n = 1000
        sum = 0.d0
      do i = 1, n
        sum = sum + ran( iseed )
      end do
        rang = sqrt( 12. ) * ( sum / n - 0.5d0 )
        rang = avg + rms * rang
        return
        end
**************************************************
```

Always use a double precision variable to accumulate a long sum, otherwise round-off errors can dominate the result. This is a very inefficient algorithm because you need to add up quite a large number of uniform random variables to make one that approximates a Gaussian. Fortunately, there is a faster way.

With a "Box-Muler" transformation, you can generate Gaussian random numbers accurately and quickly. First, generate two uniform random numbers that cover the square $-1 \le x \le +1$ and $-1 \le y \le +1$. Retain if inside the unit radius circle, $r^2 = x^2 + y^2 < 1$, otherwise try again. Then find two independent Gaussian random numbers using

$$g_1 = \frac{2x}{r}\,(-\ln r)^{1/2}\ , \qquad g_2 = \frac{2y}{r}\,(-\ln r)^{1/2}\ . \tag{43}$$

This is like magic. A function subroutine to accomplish this, shifting and scaling the result to a specified mean and rms, is:

```
**************************************************
        function rang( avg, rms, iseed )
* Gaussian random numbers
* input:
*       avg     r4      mean value
*       rms     r4      standard deviation
*       iseed   i4      seed integer to ran( iseed )
```

```
* output:
*       rang    r4 gaussian random number
*       iseed   i4      seed integer from ran( iseed )
        logical newpair /.true./
        rang = avg
        if( rms .le. 0. ) return
      if ( newpair ) then
        r2 = 10.
      do while ( r2 .ge. 1. )
        x = 2. * ran( iseed ) - 1.
        y = 2. * ran( iseed ) - 1.
        r2 = x * x + y * y
      end do
        u = sqrt( -2. * alog( r2 ) / r2 )
        rang = x * u
      else
        rang = y * u
      end if
        newpair = .not. newpair
        rang = avg + rms * rang
        return
        end
**************************************************
```

## 2.3 Lorentzian or Cauchy (ranl.for)

The Lorentzian probability map (Fig. 7) is

$$\mathrm{p}(\,x|\mu,\sigma\,) = \frac{1/\pi}{1+\eta^2}\ , \tag{44}$$

where

$$\eta = \frac{x-\mu}{\sigma}\ . \tag{45}$$

This function has a round-topped peak, like a Gaussian, but much wider wings that fall off as $1/x^2$. The peak is at $x=\mu$, and the probability density drops to half the peak value at $x=\mu\pm\sigma$.

Astronomers encounter Lorentzian probabilities in the broad damping wings of spectral lines. The broad wings arise from the finite lifetime $\Delta t$ of the quantum states in atoms. Photons are emitted and absorbed as atoms jump between two internal states with different energies. The mean photon energy is $E$, the energy difference between the two states, but the finite lifetime of the states makes this energy uncertain by $\Delta E$, where $\Delta E\ \Delta t \sim \hbar$. With each transition the photons emitted or absorbed have different energies, distributed around the mean energy with a Lorentzian probability.

Lorentz profiles also turn up in optics, specifically when the optics are dirty. Star images have Gaussian peaks caused by turbulence in the Earth's atmosphere, but dust particles on lenses and mirrors scatter light to wide angles, giving the star images broad Lorentzian wings that fall off as $1/x^2$.

If you look at a street lamp on a foggy night you will see that the lamp has a diffuse halo. This is caused by a fraction of the light entering water droplets suspended in the fog and emerging in random directions. The intensity of the halo is uniform near the light source but drops off as $1/\theta^2$ at larger angles, resembling our Lorentz profile.

Physicists may call this a Lorentzian distribution, after the physicist Lorentz, but mathematicians call it a Cauchy distribution, after the mathematician Cauchy. We all have our heroes.

Interestingly, the mean and standard deviation of the Lorentzian are not well defined. The wings are so broad that the integrals

$$\int_0^\infty \frac{x^b \mathrm{d}x}{1+x^2} \tag{46}$$

diverge for $b \geq 1$. Thus if we try to evaluate the mean, the integral with $b=1$ diverges logarithmically, and we find that we can get any answer we want depending on how we let the integration limits proceed to their respective infinities. When we try to evaluate the standard deviation, the integral with $b=2$ diverges linearly.

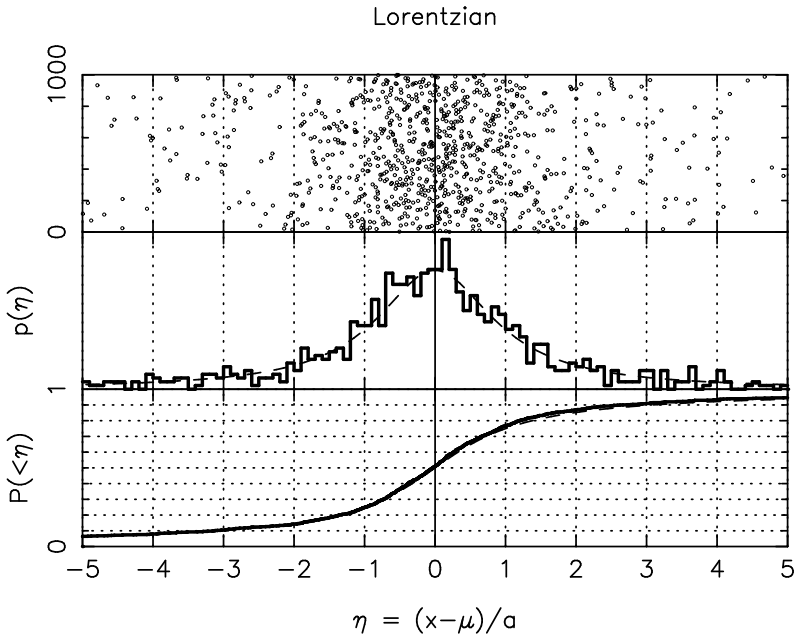


Figure 7: Lorentzian random numbers with mean $\mu = 0$ and half-width $a = 1$. The 1000 random numbers (top panel) are binned (middle panel) for comparison with the probability map. This is then integrated (lower panel) for comparison with the cumulative probability.

The ambiguous mean and infinite standard deviation are interesting quirks that may cause Mathematicians to smile or fret, depending on their disposition. They present no obstacle to the practical Physicists (including Astronomers) who just use the Lorentzian with $\mu$ and $\sigma$ as parameters.

The Lorentzian probability can be integrated to obtain an analytic expression for its cumulative probability,

$$\mathrm{P}(<x|\mu,\sigma) = \left(\frac{1}{2} + \frac{\arctan\eta}{\pi}\right) \ . \tag{47}$$

To generate Lorentzian random variables, use `ran` to generate a uniform variable on 0 to 1, and project them through the cumulative probability,

```
************************************************
        function ranl( iseed )
        logical firstcall / .true. /
        if( firstcall ) pi = 4. * atan2( 1., 1. )
        firstcall = .false.
        x( i ) = tan( pi * ( ran( iseed ) - 0.5 ) )
        return
        end
************************************************
```

Note that the first time this subroutine is called, the value of $\pi$ is obtained to machine precision using the `atan2` function. Its value is then remembered and used on subsequent calls.

## 2.4 Exponential (rane.for)

Exponential probabilities characterize the time intervals between consecutive events that are independent and have a uniform probability of occurring. Examples: time intervals between successive radioactive decays, between successive raindrops, between detection of successive photons from a constant light source.

The time for a lightbulb to fail has an exponential distribution. Interestingly, this doesn't depend on how long the lightbulb has already been working successfully. An old lightbulb that is still working has the same chance of dying as a new one.

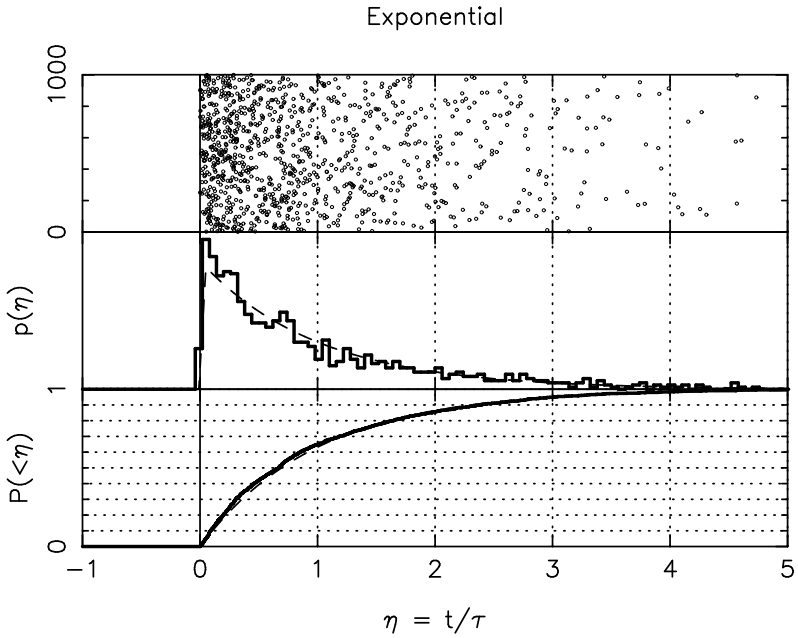


Figure 8: Exponential random numbers with an e-folding time $\tau = 1$. The 1000 random numbers (top panel) are binned (middle panel) for comparison with the probability map. This is then integrated (lower panel) for comparison with the cumulative probability.

The exponential's differential and cumulative probability maps (Fig. 8) are

$$\mathrm{p}(\,t|\tau\,) = \frac{\exp\{\,-t/\tau\,\}}{\tau}\ , \tag{48}$$

$$\mathrm{P}(<t|\tau\,) = 1 - \exp\{\,-t/\tau\,\}\ . \tag{49}$$

The mean and variance are

$$\langle\,t\,\rangle = \tfrac{1}{\tau}\int_0^\infty t\ e^{-t/\tau}\ \mathrm{d}t = \tau\int_0^\infty x\ e^{-x}\ \mathrm{d}x = \tau \tag{50}$$

$$\mathrm{Var}[\,t\,] = \tau^2\int_0^\infty (x-1)^2\ e^{-x}\ \mathrm{d}x = \tau^2 \tag{51}$$

You can generate exponential random numbers thus:

```
*************************************************
        function rane( tau, iseed )
        rane = - tau * alog( ran( iseed ) )
        return
        end
*************************************************
```

## 2.5 Power-Law (ranpl.for)

A **power-law distribution** (see Fig. 9) describes the distribution of sizes of earthquakes, landslides, avalanches, in which there are many small ones for every large one. The power-law distribution's probability map is

$$\mathrm{p}(\,x|\alpha\,) = f\ x^\alpha\ , \tag{52}$$

where the normalisation factor $f$, given by

$$f^{-1} = \int_{x_1}^{x_2} x^\alpha\,\mathrm{d}x = \begin{cases} \dfrac{x_2^{\alpha+1} - x_1^{\alpha+1}}{\alpha+1} & \text{if} \quad \alpha \neq -1 \\ \ln\,(x_2/x_1) & \text{if} \quad \alpha = -1 \end{cases}\ , \tag{53}$$

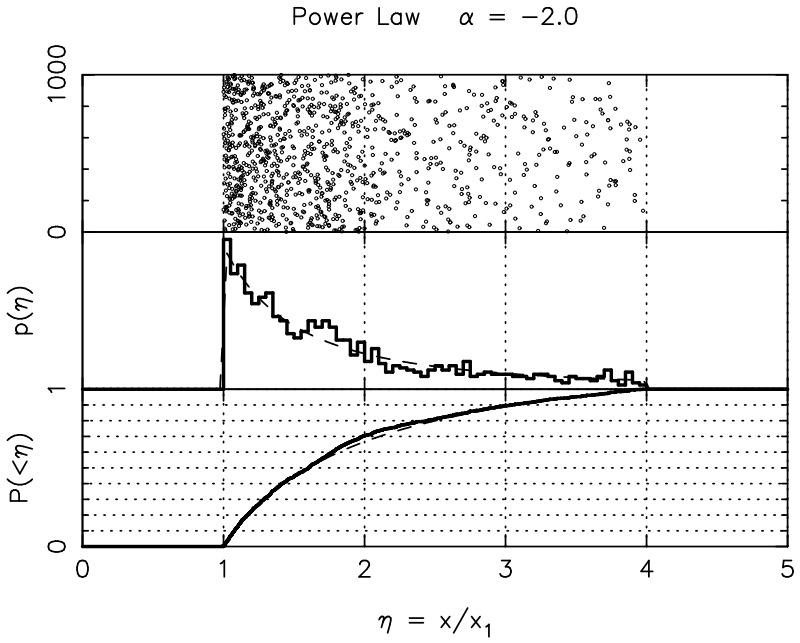


Figure 9: Power-law distribution. The 1000 random numbers (top panel) are binned (middle panel) for comparison with the probability map. This is then integrated (lower panel) for comparison with the cumulative probability.

ensures a total probability of 1 over the range $x_1 < x < x_2$.

The corresponding cumulative probability is

$$\mathrm{P}(<x|\alpha) = \begin{cases} \dfrac{(x/x_1)^{\alpha+1}-1}{(x_2/x_1)^{\alpha+1}-1} & \text{if} \quad \alpha \neq -1 \\ \dfrac{\ln(x/x_1)}{\ln(x_2/x_1)} & \text{if} \quad \alpha = -1 \end{cases} . \tag{54}$$

A subroutine to generate random numbers with a power-law distriubiton is:

```
*************************************************
        function ranpl( x1, x2, b, iseed )
* random numbers with power law probability
* input:
*       x1      r4 lower limit of range
*       x2      r4 upper limit of range
*       b       r4 power law index, i.e prob(x) = C x**b dx
*       iseed   i4 seed integer
* output:
*       ranpl   r4 random sample of power law
        r = ran(iseed)
      if( b.eq.-1. ) then
        a1 = alog( x1 )
        a2 = alog( x2 )
        ranpl = exp( a1 + ( a2 - a1 ) * r )
      else
        p = 1. + b
        a1 = x1 ** p
        a2 = x2 ** p
        ranpl = ( a1 + ( a2 - a1 ) * r ) ** (1./p)
      end if
        return
        end
*************************************************
```

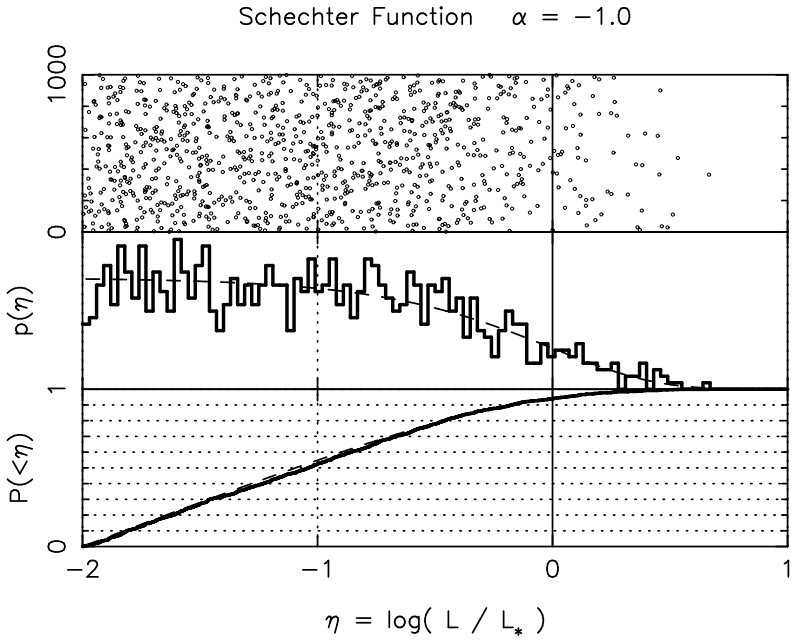


Figure 10: Schechter distribution for galaxy luminosities. The 1000 random numbers (top panel) are binned (middle panel) for comparison with the probability map. This is then integrated (lower panel) for comparison with the cumulative probability.

## 2.6 Schechter Distribution

The luminosity function of galaxies is often well approximated by a power-law distribution with slope $\alpha$ on the faint end and an exponential cutoff on the bright end at a characteristic luminosity $L_\star$.

$$\mathrm{p}(\,L|\alpha, L_\star\,) = fL^\alpha e^{-L/L_\star}\ . \tag{55}$$

This is known as the **Schechter distribution**, and is illustrated in Fig. 10 for the case $\alpha = -1$ in which faint galaxies are uniformly distributed in $\log L$.

The cumulative probability map is

$$\mathrm{P}(<L|\alpha, L_\star\,) = f\ \int_{L_F}^{L} L^\alpha\ e^{-L/L_\star}\ \mathrm{d}L\ , \tag{56}$$

and the normalisation factor $f$ is given by

$$f^{-1} = \int_{L_F}^{\infty} L^\alpha\ e^{-L/L_\star}\ \mathrm{d}L\ , \tag{57}$$

with $L_F$ the faint galaxy limit.

A subroutine to generate random numbers with a Schechter distriubiton is:

```
************************************************
       function ran_schechter( elf, elb, pow, iseed )
* Schechter galaxy luminosity function
* Input:
*       elf     r4 faint luminosity sharp cutoff
*       elb     r4 bright luminosity exponential cutoff
*       pow     r4 power-law luminosity function slope
*       iseed   i4 random number seed
* Output:
*       ran_schechter   r4 galaxy luminosity
*       iseed   i4 random number seed
```

```
        ran_schechter = 0.
        if( elf .le. 0. .or.  elb .le. elf ) return
        keep = 0
      do while ( keep .le. 0. )
        x = ranpl( elf / elb , 10., pow, iseed )
        if( ran( iseed ) .le. exp( - x ) ) keep = 1
      end do
        ran_schechter = elb * x
        return
        end
**************************************************
```

This employs `ranpl` to generate a power-law random number with slope $\alpha$ on the range from a faint limit $L_f$ up to $10 \times L_\star$. The exponential cutoff on the bright end is then applied by drawing a uniform random number to reject the power-law samples with probability $e^{-L/L_\star}$.

## 2.7 Chi-Square (ranchi2.for)

When measurement errors are Gaussian, or well approximated by Gaussians, the sums of squares of measurement errors have $\chi^2$ probabilities.

You can make a $\chi^2$ random variable by squaring a Gaussian random variable. The result is a $\chi^2$ random variable with 1 degree of freedom. Its expected value is 1, and its variance is 2.

If you take $\nu$ independent Gaussian random variables, square them and add up their squares, the result is a $\chi^2$ with $\nu$ degrees of freedom. Being a sum of squares, a $\chi^2$ random variable never takes on negative values. The expected value is $\langle\, \chi^2 \,\rangle = \nu$, and the variance is $\mathrm{Var}[\, \chi^2 \,] = 2\nu^2$. Thus loosely speaking, $\chi^2 \sim \nu \pm \sqrt{2\nu}$.

A **reduced** $\chi^2$ random variable, $\chi^2_\nu$, is obtained by dividing the $\chi^2$ random variable by its degrees of freedom. Its expected value is 1 and its variance is $2/\nu$. Thus a reduced $\chi^2$ is $\chi^2/\nu \sim 1 \pm \sqrt{2/\nu}$.

To generate $\chi^2$ random numbers,

```
**************************************************
        function ranchi2( nu, iseed )
        real*8 sum
      if( nu .lt. 100 ) then
        sum = 0.d0
      do i = 1, nu
        sum = sum + rang( 0., 1., iseed )**2
      end do
        ranchi2 = sum
      else
        dof = nu
        ranchi2 = rang( dof, sqrt( 2. * dof ), iseed )
      end if
        return
        end
**************************************************
```

## 2.8 Poisson

A photon count $n$ is a **Poisson random variable**. It is said to be subject to Poisson statistics, or photon counting statistics. It's probability map is

$$\mathrm{p}(\, n \,) = \frac{\Phi^n e^{-\Phi}}{n!} \ . \tag{58}$$

The photon count $n$ can take on only discrete non-negative integer values, 0,1,2,... The mean and variance of $n$ are

$$\langle\, n \,\rangle = \Phi \ , \qquad \mathrm{Var}[\, n \,] = \Phi \ . \tag{59}$$

You may hear prople say that "The uncertainty in $n$ counts is $\sqrt{n}$". Actually, that's not quite right. When the count rate $\Phi$ is low, a photon count $n = 0$ can occur. When the count rate is very low, the counts are in fact 0 most of the time, with a few 1's and occasional 2's. When $n = 0$, it would be silly to say that the uncertainty on $n$ is also 0. The uncertainty in $n$ counts is the square root of the *expected* number of counts, $\sigma(n) = \sqrt{\Phi}$, where $\Phi = \langle\, n \,\rangle$.

## 2.9 2-Dimensional Gaussian

A 2-dimensional Gaussian has 5 parameters, its centroid, width in two orthogonal directions, and orientation. The probability map is

$$\mathrm{p}(\, x, y | \mu_x, \mu_y, \sigma_x, \sigma_y, \theta \,) = \frac{\exp\left\{ -\eta^2/2 \right\}}{2\pi\sigma_x\sigma_y \cos\theta} \ , \tag{60}$$

where

$$\eta^2 = \frac{\eta_x^2 + \eta_y^2 - 2\eta_x\eta_y \sin\theta}{\cos^2\theta} \ , \tag{61}$$

$$\eta_x = \left( \frac{x - \mu_x}{\sigma_x} \right) \ , \quad \eta_y = \left( \frac{y - \mu_y}{\sigma_y} \right) \ . \tag{62}$$

Projections of Gaussian maps are Gaussian. The parameters $\mu_x$, $\mu_y$, $\sigma_x$, and $\sigma_y$ are the mean values and standard deviations of the Gaussian probability maps obtained by projecting onto the $x$ and $y$ axes. The first and second moments are

$$\begin{aligned} \langle\, x \,\rangle &= \mu_x \ , & \left\langle\, (x - \mu_x)^2 \,\right\rangle &= \sigma_x^2 \ , \\ \langle\, y \,\rangle &= \mu_y \ , & \left\langle\, (y - \mu_y)^2 \,\right\rangle &= \sigma_y^2 \ , \\ \mathrm{cov}(x,y) &= \langle\, (x - \mu_x)(y - \mu_y) \,\rangle = \sigma_x \ \sigma_y \ \cos\theta \quad . \end{aligned} \tag{63}$$

Slices across Gaussian maps are also Gaussian. If we hold $x$ fixed, then we find a Gaussian in $y$, and vice versa,

$$\begin{aligned} \langle\, \eta_x | y \,\rangle &= \eta_y \sin\theta \ , \quad \mathrm{Var}[\, \eta_x \,] = \cos\theta \ , \\ \langle\, \eta_y | x \,\rangle &= \eta_x \sin\theta \ , \quad \mathrm{Var}[\, \eta_y \,] = \cos\theta \ . \end{aligned} \tag{64}$$

The correlation coefficient between $x$ and $y$ is $\sin\theta$. Thus if $\sin\theta = 0$, the probability is just the product of two independent Gaussians in $x$ and $y$.

We can stretch the map in $x$ and $y$ to make $\sigma_x = \sigma_y = 1$. The resulting map will be circular if $\sin\theta = 0$, elliptical along the line $x = y$ if $\sin\theta > 0$, and elliptical along the line $x = -y$ if $\sin\theta < 0$.

# 3 Optimal Statistics

The goal of data analysis is to use measured data values to learn about the world by answering specific questions. It's up to you to decide what questions to ask. Data analysis techniques let you convert your data into relative probabilities of different possible answers to those questions.

A **statistic** is any number that you calculate from your data values, usually to provide a quantitative answer to some well-defined question. Your data points are random variables with noise properties quantified by probability maps, mean values, variances, and the other concepts we have developed and discussed in the previous chapter. Since a statistic is just a function of the data values, it is a random variable which "jitters" in response to the noise in the data.

Many people are skeptical of statistics. We hear it said that "You can prove anything with statistics", or "If you need statistics to show it, then I don't believe it". Such skepticism may be justified in some circumstances. It is possible, for example, to support any answer you want if you are dishonest enough to carefully select data subsets that support your prejudice while ignoring the rest of the evidence. Omitting relevant data is a dangerous operation that must always be approached with caution. It is also possible to dishonestly conceal the evidence against your viewpoint by doing sloppy data analysis so that you can conclude that "no contrary evidence was found in the data".

In fact, what is more true is that "You can't prove anything with statistics". You can never achieve 100% certainty no matter how strongly your present data support your present hypotheses. The next data points you receive may be the ones that rule out the theory. This is an important aspect of science – its conclusions are always falsifiable.

The best that you can do is to be honest in your data analysis. Use the best technqiues that you can to assess your hypotheses. Understand that your answers cannot be absolute, and be ready to discuss the relative probabilities of different answers.

Optimal statistics are designed to make the best possible use of all the information contained in the data points.

## 3.1 Optimal Averaging

We are going to start with very simple cases, and work our way up gradually to more and more complicated problems. In this way you should be able to build up your understanding and intuition as we go along.

Suppose you are trying to measure some quantity whose true value is $\langle X \rangle$. We will think of $\langle X \rangle$ as the brightness of a star, but remember that we could be talking about the wavelength or flux or centroid of a spectral line, or any quantity that you might want to measure from your data.

Please understand that you will never know the true star brightness $\langle X \rangle$. The best you can do is to obtain a series of measurements $X_i \pm \sigma_i$ for $i = 1...N$. Each measurement $X_i$ is a statistic. You have calculated $X_i$ by some algorithm from your data. In this case each $X_i$ stems from an image of the star taken with some telescope, camera, and detector. We will discuss later how best to measure star brightnesses from imaging data. Here we note that since the data are noisy, so too are your measurements of the star brightness. You will therefore need estimates $\sigma_i$ for the uncertainty in the individual measurements $X_i$. You will then want to combine your individual measurements in an optimal way, to obtain the most accurate estimate for the star's brightness. You will also want to know the accuracy of your result. Optimal averaging is the algorithm that accomplishes this.

We will assume that each measurement $X_i$ is independent of the others. This will be the case, for example, if each measurement arises from a different exposure of the star. If you have analyzed the same exposure in two different ways, then the errors affecting the two measurements will be correlated and the assumption will be violated.

We will also assume that each measurement gives an unbiased measurement of $\langle X \rangle$. This is not true for star brightnesses, since we observe the star through the Earth's atmosphere, where turbulent air blurs the image and scatters some of the starlight out of the beam that enters our telescope. We observe with a telescope and camera, where more light is lost to scattering and diffraction and vignetting. Finally the detector records only a fraction of the star's photons. However, you will be calibrating these systematic errors in various ways that we will discuss later. For now, assume that these calibrations are so good that we can consider each calibrated measurement of the star brightness to have no remaining systematic bias. We therefore have $\langle X_i \rangle = \langle X \rangle$ for all data points.

Sometimes you can assume that $\sigma_i$ are all the same, but usually they are all different. Even if you observed the star repeatedly using the same telescope with the same exposure time, the light losses vary because the Earth's rotation means each exposure observes the star on a different path through the atmosphere. Ideally, you will know the values of $\sigma_i$, or at least their relative values, with good accuracy, for you need this to construct the optimal average. You will have taken calibration data to understand the noise properties of each pixel of your image, and you will have extracted the star brightness $X_i$ in a way that delivers a good estimate of $\sigma_i$ for each exposure. Sometimes, you may

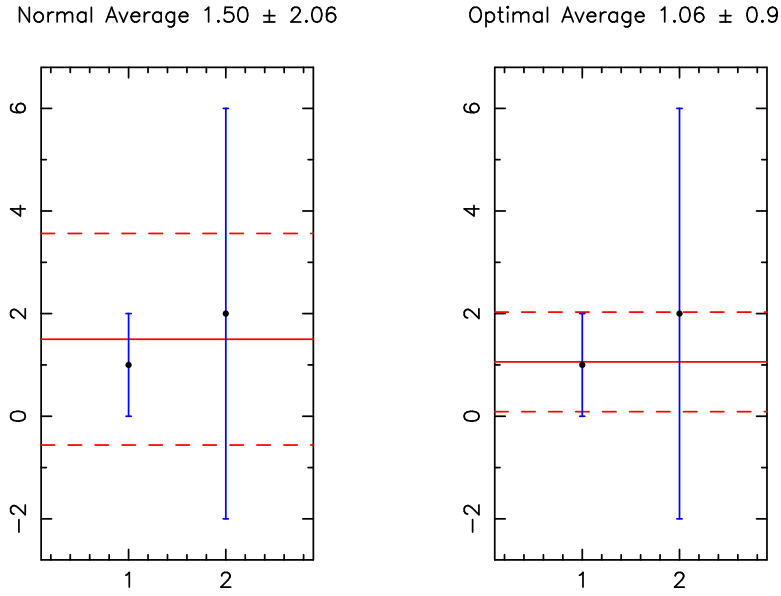


Figure 11: Normal and optimal averages of 2 data points. The normal average gives equal weight to all data points. In this example the normal average is less accurate than the first data point alone. The optimal average uses inverse-variance weights, $1/\sigma^2$, and is always more accurate than every individual data point.

receive data with no corresponding error bars. In that case you will be unable to construct an optimal average, and you will be forced to estimate $\sigma_i$ by fitting some model to the point-to-point scatter among the $X_i$.

Let's get down to business. What is the best way to combine the data values $X_i$ to obtain an optimal estimate for $\langle\, X \,\rangle$ ? If you don't know $\sigma_i$, or if you know that they are all equal, then the best you can do is to take a simple average of the data values

$$\bar{X} \equiv \frac{1}{N} \sum_{i=1}^{N} X_i \ . \tag{65}$$

This statistic has a variance

$$\mathrm{Var}\big[\, \bar{X} \,\big] = \frac{1}{N^2} \sum_i \sigma_i^2 \ . \tag{66}$$

If you do know the uncertainties $\sigma_i$, or at least have good estimates for them, then a simple average will not be the optimal way to combine the data values. The simple average gives the same weight to all data points. That's fine if the $\sigma_i$ are all equal, but if not then the more noisy data points will degrade the accuracy of the result you would get by using only the less noisy data points.

Let's look at a specific example (Figure 11). If we average a first data point $X_1 \pm \sigma_1 = 1 \pm 1$ with a second data point $X_2 \pm \sigma_2 = 2 \pm 4$, the result

$$\begin{aligned} \bar{X} \pm \sigma\left(\bar{X}\right) &= \left(\frac{X_1 + X_2}{N}\right) \pm \left(\frac{\sigma_1^2 + \sigma_2^2}{N^2}\right)^{1/2} \\ &= \left(\frac{1+2}{2}\right) \pm \left(\frac{1+16}{4}\right)^{1/2} = 1.5 \pm 2.06 \end{aligned} \tag{67}$$

is worse than that from the first data point alone. This leads to such practices as throwing out noisy data points. While improving the answer, that practice also throws away information. An optimal averaging method makes use of all of the information, so that even the noisier data points help to improve the accuracy of the result.

### 3.1.1 weighted average

It would clearly be sensible to consider giving more weight to the more accurate data points, and less weight to the noisier ones. A **weighted average**

$$\hat{X} \equiv \left(\sum_{i=1}^{N} w_i X_i\right) \Big/ \left(\sum_{i=1}^{N} w_i\right) \tag{68}$$

has a variance

$$\mathrm{Var}\Big[\,\hat{X}\,\Big] = \left(\sum_{i=1}^{N} w_i^2 \sigma_i^2\right) \Big/ \left(\sum_{i=1}^{N} w_i\right)^2 \ . \tag{69}$$

### 3.1.2 inverse-variance weights

How can you optimize the weights $w_i$ to minimize the variance? By setting the derivative to zero

$$0 = \frac{\partial}{\partial w_i}\left(\mathrm{Var}\Big[\,\hat{X}\,\Big]\right) = \frac{2 w_i \sigma_i^2}{\left(\sum_{k=1}^{N} w_k\right)^2} - \frac{2\sum_{k=1}^{N} w_k^2 \sigma_k^2}{\left(\sum_{k=1}^{N} w_k\right)^3} \ , \tag{70}$$

you will find that the optimal weights must satisfy

$$w_i \ \sigma_i^2 = \left(\sum_{i=1}^{N} w_k^2 \sigma_k^2\right) \Big/ \left(\sum_{i=1}^{N} w_k\right) \ . \tag{71}$$

You can then easily verify that the optimal weights are **inverse-variance weights**

$$w_i \propto 1/\sigma_i^2 \ . \tag{72}$$

The resulting **inverse-variance weighted average**

$$\hat{X} \equiv \left(\sum_{i=1}^{N} \frac{X_i}{\sigma_i^2}\right) \Big/ \left(\sum_{i=1}^{N} \frac{1}{\sigma_i^2}\right) \tag{73}$$

has a variance

$$\mathrm{Var}\Big[\,\hat{X}\,\Big] = 1 \Big/ \left(\sum_{i=1}^{N} \frac{1}{\sigma_i^2}\right) \ . \tag{74}$$

Let's revisit our toy problem of averaging two data points $1\pm1$ and $2\pm4$, this time using the optimal inverse-variance weights.

$$\begin{aligned} \hat{X} \pm \sigma\left(\hat{X}\right) &= \left(\frac{X_1/\sigma_1^2 + X_2/\sigma_2^2}{1/\sigma_1^2 + 1/\sigma_2^2}\right) \pm \left(\frac{1}{1/\sigma_1^2 + 1/\sigma_2^2}\right)^{1/2} \\ &= \left(\frac{1 + 2/16}{1 + 1/16}\right) \pm \left(\frac{1}{1 + 1/16}\right)^{1/2} = 1.06 \pm 0.97 \ . \end{aligned} \tag{75}$$

Notice how the noisier data point nevertheless improves the result. The optimal (inverse-variance weighted) average is more accurate than any individual data point. Indeed, it is more accurate than any subset of the data points. This is a happy feature of optimal methods: new data always improve the result. Throwing data away is never justified.

### 3.1.3 optavg.for

A subroutine for optimal averaging of data points is:

```
*************************************************
        subroutine optavg( n, dat, sig, avg, sigavg )
* optimal average of data
* input:
*       n       i4 number of data values
*       dat(n)  r4 data values
*       sig(n)  r4 error bars (<0 to ignore point)
* output:
*       avg     r4 optimal average
*       sigavg  r4 error bar on optimal average
*
        real*4 dat(*), sig(*)
        real*8 sum, sum1
        avg = 0.
        sigavg = -1.
        if( n .le. 0 ) return
        sum = 0.d0
        sum1 = 0.d0
      do i=1,n
      if( sig(i) .gt. 0. ) then
        wgt = 1. / sig(i)**2
        sum = sum + wgt * dat(i)
        sum1 = sum1 + wgt
      end if
      end do
        if( sum1 .le. 0.d0 ) return
        avg = sum / sum1
        var = 1. / sum1
        sigavg = sqrt( var )
        return
        end
*************************************************
```

Notice that the `do` loop through the data points skips any that have non-positive error bars. The two sums we require are accumulated in local variables `sum` and `sum1`, which are double precision to avoid round off errors. Always do this when a large number of values need to be added together.

## 3.2 Optimal Scaling

A spectrum is a set of measurements at many different wavelengths. A time series is a set of measurements at many different times. An image is a set of measurements at many different positions on the imaging detector, corresponding to directions on the sky. In these cases, and many others, your data points $X_i \pm \sigma_i$ are measured at different values of an **independent variable**. Data point $X_i$ may correspond to a specific wavelength $\lambda_i$ in the spectrum, or to a specific time $t_i$ for in a lightcurve, or to a specific pixel in an image.

A great many data analysis problems involve scaling a known partern to fit a set of data points. You may be trying to measure the strength of a spectral line whose central wavelength and width you know. You may be trying to measure the brightness of a faint star on an image, by scaling a **point-spread function** whose shape you have already defined from one or more brighter stars on the same image. In these cases, and many many others, you have in hand a pattern, $P_i$, describing the shape of a feature that you expect to see in the data. The problem you face is how best to scale that pattern $P_i$ to fit the data points $X_i \pm \sigma_i$. The model you need to fit to the data is

$$\mu_i = f P_i \ , \tag{76}$$

and the parameter you are interested in measuring is the **scale factor** $f$.

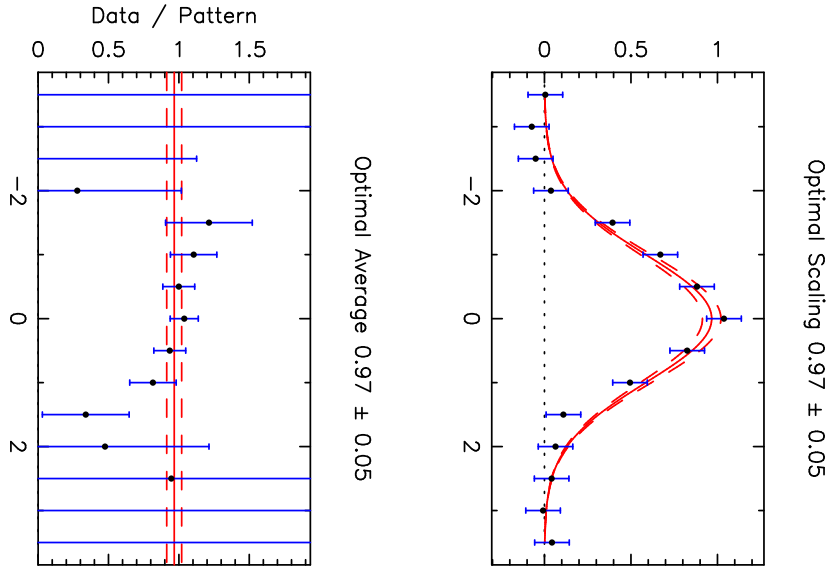


Figure 12: Optimal scaling of a Gaussian pattern to fit a set of 15 data points each with $\sigma = 0.1$. The centroid and width of the Gaussian pattern are assumed known. The optimal scale factor is the optimal average of the data points divided by the Gaussian pattern.

What is the best way to estimate the scale parameter $f$? You may be tempted to just "add up the counts", i.e. sum the data points. This would give you a biased estimator, however, since

$$\left\langle \sum_{i=1}^{N} X_i \right\rangle = \langle\, f \,\rangle \sum_{i=1}^{N} P_i \ . \tag{77}$$

Ok, no problem. You can fix this easily by dividing the sum of the data by the sum of the pattern.

$$\bar{f} \equiv \left( \sum_{i=1}^{N} X_i \right) \Bigg/ \left( \sum_{i=1}^{N} P_i \right) \ . \tag{78}$$

This nifty statistic $\bar{f}$ is now unbiased,

$$\langle\, \bar{f} \,\rangle = \langle\, f \,\rangle \ . \tag{79}$$

However, is $\bar{f}$ an optimal statistic? Sadly, no. The variance of $\bar{f}$ is

$$\mathrm{Var}\big[\, \bar{f} \,\big] = \left( \sum_{i=1}^{N} \sigma_i^2 \right) \Bigg/ \left( \sum_{i=1}^{N} P_i \right)^2 \ . \tag{80}$$

This variance is larger than the variance of the optimal statistic $\hat{f}$ that we are trying to construct here. We know this because $\bar{f}$ gives equal weight to each data point. It will clearly be beter to give more weight to data points that have low noise ($\sigma_i$ small) and strong signal ($P_i$ large). Notice that the variance of $\bar{f}$ can actually increase as more data are added. For example, if $P_i = 0$ for some data point, then that point increases the variance of $\bar{f}$ by contributing to the sum of $\sigma_i^2$ in the numerator but not to the sum of $P_i$ in the denominator. An optimal statistic would never allow new data to degrade the accuracy like this. How can we do better?

Let's try again. Remember how those $1/\sigma^2$ weights delivered an optimal average of data points? Why not do something similar here? Just weight the above sums with $1/\sigma_i^2$,

$$f_2 = \left( \sum_{i=1}^{N} \frac{X_i}{\sigma_i^2} \right) \Bigg/ \left( \sum_{i=1}^{N} \frac{P_i}{\sigma_i^2} \right) \ . \tag{81}$$

This new statistic $f_2$ is still unbiased, as you can easily verify. Its variance, as you can also easily verify, is

$$\mathrm{Var}[\, f_2\,] = \left(\sum_{i=1}^{N} \frac{1}{\sigma_i^2}\right) \Bigg/ \left(\sum_{i=1}^{N} \frac{P_i}{\sigma_i^2}\right)^2 \quad . \tag{82}$$

Is $f_2$ the optimal scale factor? No. How can we tell? Well, it's still possible for the variance of $f_2$ to degrade when a data point is added for which $P_i = 0$, increasing the numerator without affecting the denominator. Optimal statistics don't do this.

So our trial and error approach has not found the optimal statistic. How can we do it? Proceed as follows: First, try to construct an unbiased estimate for $f$ from a single data point. You'll then have $N$ independent unbiased estimates, one for each of the $N$ data points. Second, work out the variance of each of the unbiased estimators. Finally, with variances in hand, simply take the optimal (inverse-variance weighted) average.

Here we go. The expected values of the individual data points are

$$\langle\, X_i\,\rangle = \langle\, fP_i\,\rangle = \langle\, f\,\rangle P_i \ . \tag{83}$$

We can therefore construct a set of unbiased estimators $f_i$, for $i = 1...N$, by simply dividing each data point $X_i$ by the corresponding value $P_i$,

$$f_i \equiv X_i/P_i \ . \tag{84}$$

Using fuzzy algebra, we can verify that this is unbiased:

$$\langle\, f_i\,\rangle = \langle\, X_i/P_i\,\rangle = \langle\, X_i\,\rangle/P_i = \langle\, f\,\rangle \ . \tag{85}$$

We want to take an optimal average of $f_i$. In order to use inverse-variance weights, we need to know the variances of $f_i$. Using fuzzy algebra, we find

$$\mathrm{Var}[\, f_i\,] = \mathrm{Var}[\, X_i/P_i\,] = \mathrm{Var}[\, X_i\,]/P_i^2 = (\sigma_i/P_i)^2 \quad . \tag{86}$$

Notice here, and in Fig. 12, that when $P_i$ is small, the variance $\mathrm{Var}[\, f_i\,]$ is very large. As a result, low weight is given to data points where $P_i$ is small. What happens if $P_i = 0$ for some point? The variance of $\mathrm{Var}[\, f_i\,]$ is then infinite. This could be a disaster if we find we need to compute $\mathrm{Var}[\, f_i\,]$, for then we would need to divide by zero! Fortunately, this potential disaster is averted because we don't need to compute $\mathrm{Var}[\, f_i\,]$, rather we need the weights $1/\mathrm{Var}[\, f_i\,]$, and these approach zero as $P_i \to 0$.

### 3.2.1 The Golden Rule of Data Analysis

When scaling a known pattern $P_i$ to fit data points $X_i \pm \sigma_i$, the optimal estimate $\hat{f}$ of the scale factor $f$ is the optimal ($1/\sigma^2$-weighted) average of unbiased estimates $f_i \equiv X_i/P_i$ constructed from individual data points. The result, which we call *The Golden Rule*, is

$$\hat{f} \equiv \left(\sum_{i=1}^{N} \frac{X_i P_i}{\sigma_i^2}\right) \Bigg/ \left(\sum_{i=1}^{N} \frac{P_i^2}{\sigma_i^2}\right) \quad . \tag{87}$$

You can use fuzzy algebra to show that

$$\mathrm{Var}\Big[\, \hat{f}\,\Big] = 1 \Bigg/ \left(\sum_{i=1}^{N} \frac{P_i^2}{\sigma_i^2}\right) \quad . \tag{88}$$

**Please memorize these beautiful formulae!** They are the most important formulae in your data analysis your toolkit. Make them your familiar friends and they will help you out over and over and over again and again. The vast majority of data analysis problems you will encounter can be boiled down to scaling patterns to fit your data points.

### 3.2.2 optscl.for

A subroutine for optimal scaling is:

```
************************************************
        subroutine optscl( n, dat, sig, pat, f, sigf )
* optimal scaling of a pattern to fit data
* input:
*       n       i4 number of data points
*       dat(n)  r4 data values
*       sig(n)  r4 1-sigma error bars (<0 omits data point)
*       pat(n)  r4 pattern to be scaled to fit
* output:
*       f       r4 scale factor
*       sigf    r4 1-sigma error bar on f
        real*4 dat(*), sig(*), pat(*)
        real*8 sum, sum1
        f = 0.
        sigf = -1.
        if( n .le. 0 ) return
        sum = 0.d0
        sum1 = 0.d0
      do i=1,n
      if( sig(i) .gt. 0. ) then
        wgt = pat(i) / sig(i)**2
        sum = sum + dat(i) * wgt
        sum1 = sum1 + pat(i) * wgt
      end if
      end do
        if( sum1 .le. 0.d0 ) return
        f = sum / sum1
        var = 1. / sum1
        sigf = sqrt( var )
        return
        end
************************************************
```

## 3.3 Summary

We are beginning to build an understanding of how to estimate parameters by fitting models to data. We are considering *linear models* in which the parameters scale known patterns to fit the data points. We started with very simple cases, in which we can use our understanding of fuzzy numbers to construct *optimal algorithms* that deliver *unbiased* scale parameters with *minimum variance*.

We started with the simplest 1-parameter linear model, estimating the mean value of data points. We found that the *optimal average* is an *inverse-variance weighted* average of the data values. We found an intuitive way to generalize this result to the related problem of scaling a pattern to fit the data, and thereby derived the *optimal scaling* algorithm. In both cases the optimal scale parameter, and its uncertainty, are easily calculated with a single loop over the data points. These are the most important algorithms for data analysis. We will use them over and over again as we move on to more and more complicated problems.

## 3.4 Problems

1. Evaluate the simple and optimal averages, $\bar{X}$ and $\hat{X}$, of 3 data points $2 \pm 1$, $3 \pm 2$, and $1 \pm 3$. Evaluate the variances and standard deviations in both cases. Verify that the optimal average is more accurate than (a) the best single data point, (b) the average of any 2 data points, (c) the optimal average of any 2 data points.

2. Show that if the error bars are all equal, then the optimal average, $\hat{X}$ in Eqn (73), and its variance, Eqn (74), are the same as the simple average, $\bar{X}$ in Eqn (65), and its variance, Eqn (66).

3. Use fuzzy algebra to evaluate the variance of the optimal scale factor in Eqn (87) and thereby derive Eqn (88).

4. For counting statistics the $X_i$ are non-negative integers and it is often assumed incorrectly that $\sigma_i = \sqrt{X_i}$. Insert this into the Golden Rule to obtain formulae for the optimal average and its variance. What happens when one of the $X_i$ is zero?

5. Another method of optimizing a fit of the model $\mu_i = fP_i$ to $N$ data points $X_i \pm \sigma_i$ is to minimize the *badness-of-fit statistic* $\chi^2 \equiv \sum_{i=1}^{N} \eta_i^2$, where $\eta_i \equiv (X_i - \mu_i)/\sigma_i$ are *normalized residuals*. With $\sigma_i$ and $P_i$ known, accomplish the minimization by solving the equation $0 = \partial\chi^2/\partial f$, and compare the result with the Golden Rule.

6. Show by using a specific example that if you average 2 data points, and then average the result with a third data point, the result is not the same as the average of all three. Do the same using optimal averages and discover that the two calculations now give the same result. Generalize this to show that optimal averaging is *partition independent*, meaning that if you partition a dataset into $M$ subsets, optimally average the data in each subset, and then optimally average the $M$ subset averages, and the result is the same as the optimal average of the entire dataset. As a special case, you can combine a new data point with the optimal average of $N$ previous data points without having to remember the $N$ previous data points and repeat the long calculation. Is the same true for optimal scaling?

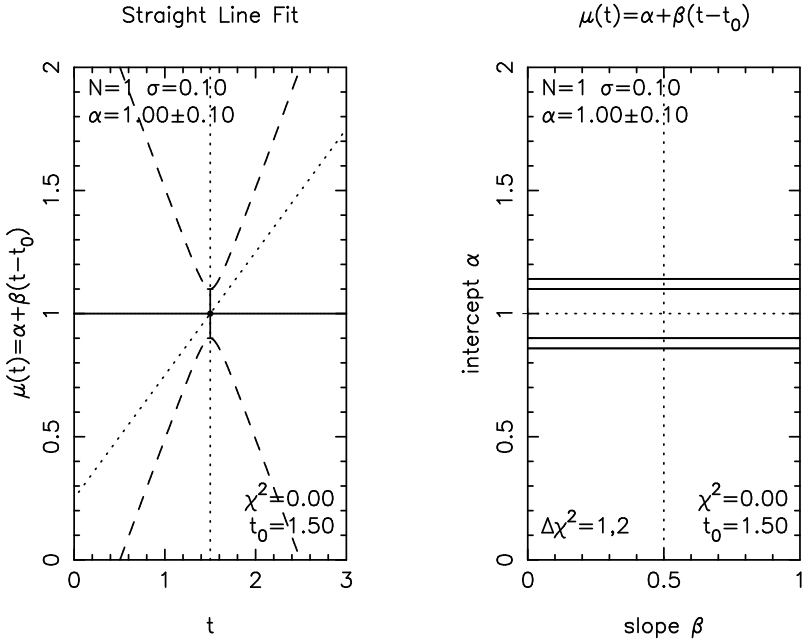


Figure 13: Straight line fit to 1 data point. The slope is degenerate.

# 4 Straight Line Fit

A straight line has 2 parameters, 2 degrees of freedom. Make a plot of your data points, say $X_i \pm \sigma_i$ at times $t_i$. Now, draw a line through the data with a ruler. You will first adjust the line 'by eye', holding the ruler against the plot, shifting it up and down, or left and right, and rotating it to change its slope until it 'runs through' the data points. You might think that there are 3 degrees of freedom here, up-down, left-right, and rotate, but in fact there are only 2 because shifting the ruler parallel to itself does not change the line you draw on the data. There are 2 degrees of freedom here, not 3.

In fitting a straight line to data points, you are using the computer to work out the best-fit rather than doing it by eye. Because this fit involves 2 parameters, instead of just 1, you will need to pick up a couple of new concepts, new mental pictures, that will help you to understand and visualize the nature of the connection between the data, the fitted line, and the line's parameters. Intuition gained from careful analysis of the straight-line will be helpful later, when you are trying to visualize higher-dimensional data sets and parameter spaces. I will lead you through a series of examples, to build up this intuition.

## 4.1 1 data point – degenerate parameters

It is clearly silly to try to fit a line through 1 data point. However, it can be helpful to examine trivial examples in some detail. If you have only 1 data point, $x_1 \pm \sigma_1$ at time $t_1$, you can draw many different lines with many different slopes that pass through the data point exactly. The slope of the line can be anything, provided the line pivots on the data point. The up-and-down shift of the line is constrained by the data point, but the slope of the line is not. This illustrates the concept of **degeneracy**. One degree of freedom, the up-down shift, is constrained by the data, but the other degree of freedom, the slope, is not.

The straight-line model has 2 parameters. To make the line pivot on the data point, write it as

$$\mu(t) = \alpha + \beta\,(t - t_1) \ , \tag{89}$$

where $\beta$ is the slope, and $\alpha$ is the predicted data at time $t = t_1$. If $\hat\alpha$ is the best-fit value of the parameter $\alpha$, it is clear that $\hat\alpha = x_1$, for this makes the line fit the data point exactly. As the data point jitters around, so will $\hat\alpha$. It is clear that $\langle\,\hat\alpha\,\rangle = x_1$ and $\mathrm{Var}[\,\hat\alpha\,] = \mathrm{Var}[\,x_1\,] = \sigma_1^2$. The slope $\beta$ is **degenerate** because as you change $\beta$ the line pivots on the data point, maintaining a perfect fit. The parameters $\alpha$ and $\beta$ are **orthogonal** or **un-correlated**, meaning that fixing the value of one has no effect on the probability map of the other.

Another way to parameterize the straight-line model is

$$\mu(t) = \alpha + \beta t \ , \tag{90}$$

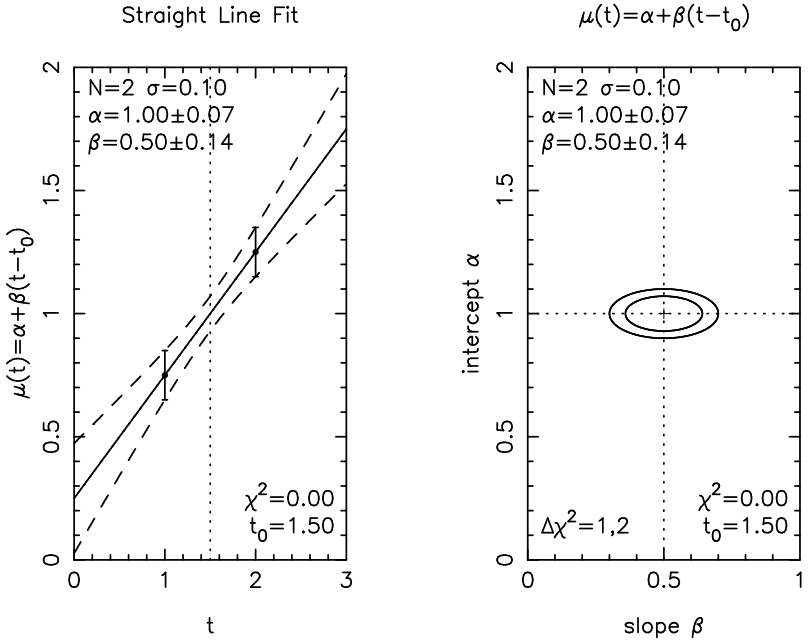


Figure 14: A straight line fits 2 data points exactly.

This time the parameter $\alpha$ is the predicted data at time $t = 0$. The best-fit line still fits the data point exactly, but now it imposes a constraint between the 2 parameters:

$$\hat{\alpha} = x_1 + \hat{\beta} t_1 \ . \tag{91}$$

The two parameters are thus **correlated**. If you change one parameter, the other must also change in order to maintain the fit. The fit is still degenerate, but now, because of the correlation between the parameters, the values of both parameters become essentially unconstrained. Whatever shift $\alpha$ you chose, you can find a slope $\beta$ that fits the data exactly. Whatever slope $\beta$ you chose, you can find a shift $\alpha$ to fits the data exactly.

## 4.2 2 data points – correlated vs orthogonal parameters

You have 2 data points $(t_1, x_1 \pm \sigma_1)$ and $(t_2, x_2 \pm \sigma_2)$. There is a unique line that passes through the 2 data points, fitting the data perfectly. As the 2 data points jitter up and down, you can imagine the best-fit line jittering around with them. This jittering of the line in response to the jittering data points defines the probabilty map p( $X(t)$ ) in the predicted data at any time.

To pivot the line on the first data point, write it as

$$\mu(t) = \alpha + \beta\,(t - t_1) \ . \tag{92}$$

The best fit is then

$$\hat{\alpha} = x_1 \ , \qquad \hat{\beta} = \frac{x_1 - x_2}{t_2 - t_1} \ . \tag{93}$$

These parameters are correlated.

There are 4 ways to fit a line through the ends of the error bars, missing both points by 1-$\sigma$. You can fit through the top of both error bars, shifting the line up by 1-$\sigma$, or through the bottom of both error bars, shifting it down by 1-$\sigma$. This degree of freedom shifts the centroid of the line but doesn't alter the slope. You can also fit through the top of the error bar on one point and the bottom of the error bar on the other, or vice-versa. This degree of freedom changes the slope by pivoting the line about the centroid of the data points.

To cleanly separate the two degrees of freedom, thereby obtaining an orthogonal parameterization of the line, move the pivot point to the centroid of the two data points, at the inverse variance weighted mean of the times

$$\hat{t} == \frac{t_1/\sigma_1^2 + t_2/\sigma_2^2}{1/\sigma_1^2 + 1/\sigma_2^2} \ . \tag{94}$$

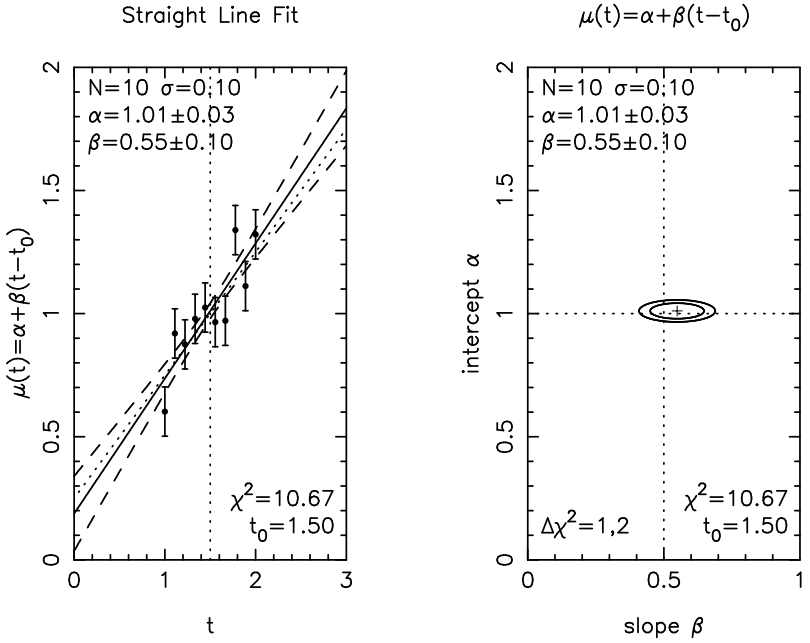


Figure 15: A straight line fit to 10 data points.

For this orthogonal parameterization, the model is

$$\mu(t) = \alpha + \beta\left(t - \hat{t}\right) \ , \tag{95}$$

and the best-fit parameters are

$$\hat{\alpha} = \frac{x_1/\sigma_1^2 + x_2/\sigma_2^2}{1/\sigma_1^2 + 1/\sigma_2^2} \ , \qquad \hat{\beta} = \frac{x_1 - x_2}{t_2 - t_1} \ . \tag{96}$$

## 4.3 $N$ data points

Now let's conquer the more general case. You have $N$ data points $x_i \pm \sigma_i$ at times $t_i$ Suppose there is no uncertainty in $t_i$, so that the error bars apply only to $x_i$. You want to fit these data points with a straight line, which you can parameterize as

$$\mu(t) = \alpha + \beta t \ . \tag{97}$$

You can easily construct a simple algorithm to fit this line to your data by using an optimal average to evaluate $\alpha$ and then optimal scaling to evaluate $\beta$. Start by setting the slope $\beta = 0$. Estimate $\alpha$ by computing the optimal average of the data:

$$\hat{\alpha} = \frac{\sum\limits_i x_i/\sigma_i^2}{\sum\limits_i 1/\sigma_i^2} \ , \qquad \mathrm{Var}[\,\hat{\alpha}\,] = \frac{1}{\sum\limits_i 1/\sigma_i^2} \ . \tag{98}$$

At this point $\hat{\alpha}$ is the best-fit line with zero slope. Now subtract the optimal average from the data, and estimate $\beta$ by optimal scaling of the pattern $t$ to fit the residuals:

$$\hat{\beta} = \frac{\sum\limits_i (x_i - \hat{\alpha})\, t/\sigma_i^2}{\sum\limits_i t^2/\sigma_i^2} \ , \qquad \mathrm{Var}\Big[\,\hat{\beta}\,\Big] = \frac{1}{\sum\limits_i t^2/\sigma_i^2} \ . \tag{99}$$

When you subtract off the scaled pattern $\hat{\beta}t$, you will find that the residuals no longer scatter about zero. You will therefore need to re-fit $\hat{\alpha}$ to residuals:

$$\hat{\alpha} = \frac{\sum\limits_i \left(x_i - \hat{\beta}t\right)/\sigma_i^2}{\sum\limits_i 1/\sigma_i^2} \ , \qquad \mathrm{Var}\Big[\,\hat{\beta}\,\Big] = \frac{1}{\sum\limits_i 1/\sigma_i^2} \ . \tag{100}$$

The formula for $\hat{\beta}$ involves $\hat{\alpha}$ and the formula for $\hat{\alpha}$ involves $\hat{\beta}$. You will therefore need to iterate the above 2 steps until the values cease to change significantly. Each time you adjust $\alpha$, the residuals have a slope and you need to re-adjust $\beta$. Each time you adjust $\alpha$, the residuals have an offset and you need to re-adjust $\beta$.

Wouldn't it be nice to avoid having to iterate? Fortunately, you can do this rather simply by changing the parameterization of the line to

$$\mu(t) = \alpha + \beta\left(t - \hat{t}\right) \ , \tag{101}$$

where

$$\hat{t} = \frac{\sum\limits_i t/\sigma_i^2}{\sum\limits_i 1/\sigma_i^2} \ . \tag{102}$$

This simply shifts the origin of time from $t = 0$ over to $t = \hat{t}$, the inverse-variance weighted average of the times of observation. This small change in procedure brings a great reward. Instead of having to iterate many times to achieve a fit, you can now calculate $\hat{\alpha}$ as before, and this best-fit value of $\alpha$ applies for any value of $\alpha_1$. The pivot point $(t, x) = (\hat{t}, \hat{\alpha})$ is now at the inverse-variance weighted centroid of the data. Adjusting the slope $\alpha_1$ moves the right side up and the left side down, but the line still runs through the centroid of the data. The new parameters $\alpha$ and $\beta$ are said to be **orthogonal parameters**.

You can now calculate the best-fit value of $\beta$ by optimal scaling of the pattern $(t - \hat{t})$:

$$\hat{\beta} = \frac{\sum\limits_i x_i\left(t - \hat{t}\right)/\sigma_i^2}{\sum\limits_i \left(t - \hat{t}\right)^2/\sigma_i^2} \ , \qquad \mathrm{Var}\Big[\,\hat{\beta}\,\Big] = \frac{1}{\sum\limits_i \left(t - \hat{t}\right)^2/\sigma_i^2} \ . \tag{103}$$

Because $\hat{\alpha}$ and $\hat{\beta}$ are now orthogonal, there is no need to subtract $\hat{\alpha}$ from the data in the formula for $\hat{\beta}$, though it doesn't hurt to do so. Similarly, there is no need to re-solve for $\hat{\alpha}$.

## 4.4 fitline.for

A subroutine to fit a straight line to data points:

```
*****************************************************
        subroutine fitline( n, t, dat, sig, t0, a, b, siga, sigb )
* Use optavg and optscl to fit a straight line
*       y(t) = a + b * ( t - t0 )
* to n data points dat(i) +/- sig(i) at times t(i).
* The fit variance is
*       var( y(t) ) = siga**2 + ( ( t - t0 ) * sigb )**2
* Input:
*       n       i4 number of data points
*       t(n)    r4 time (independent variable)
*       dat(n)  r4 data values
*       sig(n)  r4 1-sigma error bars (<0 omits point)
* Output:
*       t0      r4 centroid of times
*       a       r4 centroid of data
*       b       r4 slope
*       siga    r4 1-sigma uncertainty in a (<0 if no data)
*       sigb    r4 1-sigma uncertainty in b (<0 if degenerate)
*
* 2002 May Keith Horne @ St.Andrews - use optavg and optscl
        real*4 t(*), dat(*), sig(*)
        t0 = 0.
        a = 0.
```

```
        b = 0.
        siga = -1.
        sigb = -1.
        if( n .le. 0 ) return
* shift origin to centroid of times and data
        call optavg( n, dat, sig, a, siga )
        if( siga .le. 0. ) return
        call optavg( n, t, sig, t0, sigt0 )
        if( sigt0 .le. 0. ) return
      do i=1,n
        t(i) = t(i) - t0
        dat(i) = dat(i) - a
      end do
* scale pattern t(i)-t0 to fit residuals dat(i)-a
        call optscl( n, dat, sig, t, b, sigb )
* restore times and data
      do i=1,n
        t(i) = t(i) + t0
        dat(i) = dat(i) + a
      end do
        return
        end
*************************************************
```

This code uses subroutines from our toolbox to accomplish a significant new task with a minimum of fuss. Passing the data values, $x_i \pm \sigma_i$ to `optavg` computes their optimal average $\hat{a}$ and the corresponding uncertainty $\sigma(\hat{a})$. This effectively scales a constant to fit the data. A second call to `optavg` yields $\hat{t}$, the $1/\sigma_i^2$-weighted centroid of the times $t_i$. The `do` loop subtracts the centroid from the data and the times. The time shift makes the intercept $\hat{a}$ and slope $\hat{b}$ orthogonal, and we can therefore calculate $\hat{b}$ and its uncertainty $\sigma\left(\hat{b}\right)$ by calling `optscl` to scale the orthogonal pattern $t - \hat{t}$ to fit the residuals. It was not strictly necessary to subtract $\hat{a}$, but we did it to reduce round-off errors when calculating $\hat{b}$. The final `do` loop restores the original centroids.

This example illustrates the general method of fitting a linear model by scaling a series of orthogonal patterns to fit the data. We will develop and generalize and apply this method in many examples to follow.

We are also illustrating here the good practice of using previously-tested subroutines to build solutions to your data analysis problems. This practice minimizes the opportunities coding effort. The names of the subroutines are chosen to make their action easy to remember. The inputs and outputs of each subroutine are briefly but carefully documented in comments at the top. Each subroutine does a specific job that is useful in the sense that you will need to do that job over and over again in many different contexts. The subroutine can be tested independently of any codes that may use it. Having the problem solved in one place, rather than duplicating the code over and over again in many different places, minimizes the opportunities for errors, and is easier to maintain because when you do find a bug there is only 1 place where you need to fix it. Once tested, the subroutine can be used as a black box with well defined inputs and outputs. You can forget about the details of the internal algorithm. The subroutines in your toolkit are building blocks. Accomplishing your task by assembling building blocks makes each new algorithm using relatively easy to understand by looking at the code.

However, the resulting code is inefficient. There are overheads associated with the 3 subroutine calls. The code uses at least 5 loops over the $N$ data values, when only 1 is required. If you have very large datasets, and you need to accomplish a large number of line fits, this version may be too slow. To produce a faster version, you can replace the subroutine calls with explicit code, eliminate redundant loops over the data, and keep the number of floating point operations to a minimum. Another unfortunate feature is that the input times and data values may be altered slightly due to round off errors associated with subtracting and adding back the centroid. This was necessary because we used `optscl` to evaluate $\hat{b}$. You can avoid this computing the slope with explicit code. Finally, if the input times, data values, or error bars happen to be extremely large or small numbers, the subroutine may crash with a floating point overflow or divide by zero. To avoid this, rescale the de-centroided times and data to span ranges of order unity, fit the model to the scaled data, and then correct the parameters for the scale factors.

## 4.5 Summary

The straight line is a simple 2-parameter linear model. It is simple enough that we can work out the solution exactly, using two applications of optimal scaling. In fitting to 1 data point, the solution is *degenerate* in the sense that there are an infinite set of parameters that achieve a perfect fit. With 2 data points the 2 parameters achieve an exact fit. In this case we encounter the problem of *correlated parameters*, which seems to require iteration to approach the solution. We then discover that a simple change of coordinates recasts the problem in terms of *orthogonal parameters*. With orthogonal parameters, each parameter can be optimized independently of the others. The optimal value for one parameter is independent of the adopted values, optimized or not, of all the other parameters.

We considered also some practical programming issues. We are beginning to assemble a toolkit of subroutines, well-tested black boxes with well-defined inputs and outputs that encapsulate the solutions to specific data analysis problems. These subroutines are building blocks that allow us to assemble solutions to a wide variety of data analysis problems while minimizing coding effort and opportunities for coding errors.

## 4.6 Problems

1.

2.

3.

4.

5.

# 5 Periodic Signals

We will now consider a problem with a mix of linear and non-linear parameters.

## 5.1 Sine Curve Fit

The same techniques of scaling orthogonal patterns to fit residuals can be used to fit a sine curve to data points. The sine curve model

$$\mu(t) = B + S\sin\omega t + C\cos\omega t \tag{104}$$

has 3 linear parameters, the background $B$, and the sine and cosine amplitudes $S$ and $C$. The period $P = 2\pi/\omega$ we assume to be known. You can also write the sine curve in terms of an amplitude $A$ and fiducial phase $\phi_0$:

$$\mu(t) = B + A\sin\omega t + \phi_0 \ . \tag{105}$$

The sine curve model is linear in $A$, $B$, $C$, and $S$, but non-linear in $\omega$ and $\phi_0$. For this reason it is best to estimate the parameters $B$, $S$, and $C$ and then calculate the amplitude $A$ and fiducial phase $\phi_0$ from

$$A = \left(S^2 + C^2\right)^{1/2} \ , \qquad \phi_0 = \arctan\left(-B/A\right) \ . \tag{106}$$

We will consider $\omega$ to be known.

Use optimal averaging to measure $B$, and optimal scaling to measure $S$ and $C$. If the data at times $t_i$ are $X_i \pm \sigma_i$, evaluate $s_i = \sin\omega t_i$ and $c_i = \cos\omega t_i$. The residuals of the fit are

$$\epsilon_i = X_i - (B + Ss_i + Cc_i) \tag{107}$$

You can update the parameters by optimal scaling of the appropriate patterns to fit the residuals

$$\Delta B = \frac{\sum\limits_i \epsilon_i/\sigma_i^2}{\sum\limits_i 1/\sigma_i^2} \ , \qquad \Delta S = \frac{\sum\limits_i \epsilon_i s_i/\sigma_i^2}{\sum\limits_i s_i^2/\sigma_i^2} \ , \qquad \Delta C = \frac{\sum\limits_i \epsilon_i c_i/\sigma_i^2}{\sum\limits_i c_i^2/\sigma_i^2} \ . \tag{108}$$

If the data populate the range of phases roughly equally, then the parameters will be almost orthogonal and the fit will converge with just a few iterations.

## 5.2 Periodogram

A *periodogram* employs a grid search technique to explore all values of the oscillation period $P$. At each trial period $P$, the background level and the sine and cosine amplitudes are calculated as above. A plot of $\chi^2$ vs $P$ then has high values at trial periods that give a poor fit to the data, and dips to low values at trial periods that give a good fit to the data.

It is important to take some care in designing the period grid used for the period scan. We want the period spacing to be fine enough that we do not skip over any dips in $\chi^2$, but not so fine that we waste computer time.

## 5.3 Fourier Frequencies for Equally Spaced Data

Suppose you have $N$ data points spaced in time by $\delta t$ at times

$$t_i = i\ \delta t \tag{109}$$

for $i = 0...N-1$. The total time spanned by the data is

$$T = (N-1)\ \delta t \ . \tag{110}$$

You will then be able to fit the data points exactly using a *Fourier series*

$$\mu(t) = \sum_{k=0}^{N_k} S_k \sin\left(2\pi t/P_k\right) + C_k \cos\left(2\pi t/P_k\right) \ . \tag{111}$$

Since there are $N$ data points, and the sine and cosine amplitudes $S_k$ and $C_k$ represent 2 degrees of freedom per perod, we expect there to be $N_k \approx N/2$ independent terms in the Fourier series. The *Fourier periods* are

$$P_k = \frac{1}{F_k} \ , \tag{112}$$

and the corresponding *Fourier frequencies* are

$$F_k = k \ \delta F \ , \tag{113}$$

for $k = 0, 1, ..., N_k$. We assume here, and justify below, that the Fourier frequencies are equally spaced.

The shortest and longest periods are special. The longest period, the $k = 0 term$, is $P_0 = \infty$. For this zero-frequency component, the sine term is zero and the cosine term is 1. We may therefore set $S_0 = 0$ in the Fourier series, and recognize that $C_0$ represents a constant component.

The shortest period, or highest frequency, on which the data provide information is 2 data points per cycle. This is the *Nyquist period*,

$$P_{\rm Nyq} = 2\delta t \ . \tag{114}$$

The corresponding *Nyquist frequency* is

$$F_{\rm Nyq} = \frac{1}{P_{\rm Nyq}} = \frac{1}{2\delta t} \ . \tag{115}$$

We now justify, heuristically, the assumption that the Fourier frequencies are equally spaced. The spacing between independent periods should be just sufficient to change the phase of the sine curve by 1 cycle over the duration $T$ of the observations, i.e.

$$\delta P \frac{T}{P} = P \ . \tag{116}$$

The spacing between independent periods is then

$$\delta P = \frac{P^2}{T} \ . \tag{117}$$

The corresponding frequency spacing is

$$\delta F = \delta P \left| \frac{dF}{dP} \right| = \frac{\delta P}{P^2} = \frac{1}{T} \ . \tag{118}$$

The independent Fourier frequencies, equally spaced by $\delta F = 1/T$, are then

$$F_k = k \ \delta F = \frac{k}{T} \ , \tag{119}$$

and the corresponding Fourier periods are

$$P_k = \frac{1}{F_k} = \frac{T}{k} \ , \tag{120}$$

for $k = 0, 1, ...k_{\rm max} \leq N/2$. Notice that the Fourier periods are not equally spaced, but the Fourier frequencies are.

If $N$ is odd, then $k_{\rm max} = (N-1)/2$ and the highest Fourier frequency is less than the Nyquist frequency. Let's count the parameters in the Fourier series. We have the constant term $C_0$, for $k = 0$. We also have the sine and cosine amplitudes, $S_k$ and $C_k$, for $k = 1, 2, ..., k_{\rm max}$. The total number of parameters is therefore $1 + 2k_{\rm max} = N$. Thus we have just enough parameters to expect to be able to fit the $N$ data points exactly.

If $N$ is even, then $k_{\rm max} = N/2$ and the highest Fourier frequency is equal to the Nyquist frequency. At the Nyquist frequency, notice that the sine term is exactly zero, while the cosine term alternates between +1 and -1. We can therefore set $S_{k_{\rm max}} = 0$, eliminating 1 parameter. Once again, let's count the number of parameters. First, we have the constant term $C_0$, for $k = 0$. Next, we have $S_k$ and $C_k$, for $k = 1, 2, ..., k_{\rm max} - 1$. Finally, we have $C_{k_{\rm max}}$, for $k = k_{\rm max} = N/2$. The total number of parameters is therfore $1 + 2(k_{\rm max} - 1) + 1 = N$. Once again the number of parameters equals the number of data points.

## 5.4 Periodic features of fixed width

Suppose that instead of fitting a sinusoid to the data, we are looking for eclipses of duration $W$ in the lightcurve that recur with period $P$, where $W/P < 1/2$ of a period. The same technique of scaling patterns to fit the lightcurve data can be used, but now the pattern is a lightcurve with an eclipse of appropriate width, rather a sine curve. We may pick the shape of the dip to be a Gaussian, or a boxcar, or some other shape that approximately represents the shape of the eclipse we seek to find. If the eclipse width is unknown, we need to search over an appropriate range of $W$ in

addition to searching over the period $P$ and epoch $t_0$ of the eclipse ephemeris. For each value of $P$ and $t_0$ and $W$, we scale the pattern to fit the observed lightcurve, and take note of the $\chi^2$ of the fit.

It is interesting to notice that in this case the period spacing needed for the search is different from that used in the case of a periodogram, where we fit sine curves to the data. In fitting a sine curve, the pattern is spread over the full period, while for an eclipse it is confined to a smaller fraction $W/P < 1/2$ of a period. We therefore need a closer period spacing if we are seaching for eclipses.

We want the period spacing $\delta P$ to be just enough to change the predicted eclipse time by some fraction of the eclipse duration. The number of cycles spanned by the data is $T/P$, so the change in the eclipse time from the beginning to the end of the experiment is

$$\delta P \ \frac{T}{P} < fW \ . \tag{121}$$

This means that the period spacing should be

$$\frac{\delta P}{P} = d(\ln(P)) < \frac{fW}{T} \ . \tag{122}$$

Thus when searching for periodic features of fixed width, the period grid used for the search should be equally spaced in $\ln(P)$. The spacing should be $\Delta(\ln(P)) = fW/T$, with $f$ being some constant, e.g. $f \approx 1/4$. If the min period is $P_0$, then the kth period is

$$P_k = P_0 \exp\left(kfW/T\right. \tag{123}$$

for $k = 0, 1, ...k_{\rm max}$. If the max period $P_{\rm max} < T$, then

$$k_{\rm max} = \frac{\ln(P_{\rm max}/P_0)}{\Delta(\ln(P))} \ . \tag{124}$$

This would be a smart way to construct the period grid.

If we make a plot of $chi^2$ vs $\log(P)$, we should see dips and peaks with a correlation length of order $\Delta(\ln(P)) = W/T$. Notice that we should use a different period grid for each eclipse width $W$ used in the search. When $W$ is large, we don't need to search over so many periods. We need to choose $f$ small enough so that the period scan does not miss any of the dips in $\chi^2$, but not so small that it wastes computer time in doing the search. In most cases $f \approx 1/4$ will be about right, but you should make a plot to verify that.

## 5.5 Summary

An oscillating signal is a model involving with a mix of linear and non-linear parameters. The period and phase of the oscillation are non-linear parameters, while the baseline and amplitude are linear parameters. By employing sine and cosine basis functions, the model has 3 linear parameters, and only the period is non-linear. Here we use optimal scaling to

It is possible to write explicit solutions for the 3-parameter linear model, a good exercise for the reader. We relied instead on *iteration* to refine an initial solution by *successive approximations*. Optimal scaling yields estimates for the 3 scale parameters. The optimal correction to each scale parameter is obtained by optimally scaling the corresponding pattern to fit residuals from the previous fit. The parameters are approximately orthogonal when the data provide a roughly uniform sampling of the phase of the sine curve. The iteration method is easy to implement, and converges in only a few iterations if the parameters are approximately orthogonal. It can be too slow, however, if the parameters are highly correlated.

In the next chapter we generalize to fitting $M$-parameter linear models to data points, a class of problems known by the obscure name *Linear Regression*.

## 5.6 Problems

1. Write an efficient subroutine to fit a 3-parameter linear model to data points.
2.
3.
4.
5.

# 6 Linear Regression

Having bolstered our courage a bit by conquering several special cases, let's now turn our attention onto the more general case of fitting an $M$-parameter linear model,

$$\mu_i(\alpha) = \sum_{j=1}^{M} \alpha_j P_{ij} \ . \tag{125}$$

As usual, the $N$ data points are $X_i \pm \sigma_i$. We have $M$ patterns, $P_{ij}$ for $j = 1...M$. Each pattern has a known shape, but is multiplied by an unknown scale parameter $\alpha_j$.

This so-called *linear regression* problem can be tackled using the iterative methods we discussed for the fit of a constant plus a sine curve to data points. In that problem there were 3 patterns and 3 scale parameters. We scaled each parameter in turn, optimizing its value while holding the other values fixed. If the patterns and parameters happen to be orthogonal to each other, the optimal solution is found after $M$ optimal scalings. More often, however, the parameters are not orthogonal, so that rescaling one parameter changes the optimal values of the others. The solution needs to be iterated. This can be very slow if there are strong correlations among the parameters. We will therefore need a more powerful method.

## 6.1 Normal Equations

The conventional approach is to write $\chi^2$ as a function of the $M$ scale parameters

$$\chi^2(\alpha) = \sum_{i=1}^{N} \left( \frac{X_i - \mu_i(\alpha)}{\sigma_i} \right)^2 \ . \tag{126}$$

To find where $\chi^2(\alpha)$ has a minimum, set to zero the derivatives with respect to each of the $M$ parameters:

$$0 = \frac{\partial \chi^2}{\partial \alpha_k} = -2 \sum_{i=1}^{N} \frac{X_i - \mu_i(\alpha)}{\sigma_i^2} \frac{\partial \mu_i}{\alpha_k} \ . \tag{127}$$

For the linear model, the derivatives with respect to the scale parameters are simply the patterns that they scale,

$$\frac{\partial \mu_i}{\partial \alpha_k} = P_{ik} \ . \tag{128}$$

The $M$ equations, for $k = 1...M$, are then

$$\sum_{i=1}^{N} \frac{X_i P_{ik}}{\sigma_i^2} = \sum_{i=1}^{N} \frac{\mu_i(\alpha) P_{ik}}{\sigma_i^2} = \sum_{i=1}^{N} \sum_{j=1}^{M} \frac{\alpha_j P_{ij} P_{ik}}{\sigma_i^2} \ . \tag{129}$$

Before proceding, take a moment to notice that for the case of a single parameter, $M = 1$ and $\mu_i(\alpha) = \alpha P_i$, the solution is obvious and recovers the optimal scaling algorithm

$$\hat{\alpha} = \sum_{i=1}^{N} \frac{X_i P_i}{\sigma_i^2} \Bigg/ \sum_{i=1}^{N} \left( \frac{P_i}{\sigma_i} \right)^2 \ . \tag{130}$$

It is customary to write the *normal equations* in matrix form

$$b_k = \sum_{j=1}^{M} H_{kj} \alpha_j \ . \tag{131}$$

The solution may then be written formally as

$$\hat{\alpha_i} = \sum_{k=1}^{M} \left( H^{-1} \right)_{ik} b_k \ . \tag{132}$$

In practice the coefficients of the elements of the vector $b_k$ and matrix $H_{kj}$ are straightforward to calculate, and a numerical matrix inversion code may then be used to obtain the solution. Here we will look conceptually at the solution to develop a more intuitive understanding of how the data points define a region of the $M$-dimensional parameter space that provides a good fit to the data.

The $M$-dimensional vector

$$b_k \equiv \sum_{i=1}^{N} \frac{X_i P_{ik}}{\sigma_i^2} \ , \tag{133}$$

is an inverse-variance weighted correlation between the $k$-th pattern and the data. If $b_k = 0$, this means that the corresponding pattern is orthogonal to the data. This pattern can be scaled by an arbitrary factor with no effect on the fit to the data points. The effect will be huge, however, in the gaps between data points. If this happens then that parameter is left unconstrained by the data, and you had better find some other way to set its value or else eliminate it from the problem.

## 6.2 The Hessian Matrix

The matrix which appears in the normal equations is the $M \times M$-dimensional *Hessian matrix*,

$$H_{kj} = \sum_{i=1}^{N} \frac{P_{ij} P_{ik}}{\sigma_i^2} \ . \tag{134}$$

This is a symmetric matrix that depends on the error bars and the patterns but not on the data values. The Hessian matrix plays a very important role in defining the size and shape of the region in parameter space that offers a good fit to the data. For this reason it is worthwhile to take some time to study it.

The Hessian matrix element $H_{kj}$ is an inverse-variance weighted correlation between the $k$-th and $j$-th patterns. Note that the diagonal elements of the Hessian matrix are always positive. If all the patterns are orthogonal, then the Hessian matrix is diagonal, and the diagonal elements are

$$H_{jj} = \sum_{i=1}^{N} \left( \frac{P_{ij}}{\sigma_i} \right)^2 = \frac{1}{\mathrm{Var}[\,\alpha_j\,]} \ . \tag{135}$$

The second equality holds whenever the $j$-th pattern is orthogonal to all the others, in which case the $j$-th row and column are entirely zero except for the positive diagnonal element giving the inverse variance of the scale parameter $\alpha_j$.

When two patterns are not orthogonal, the corresponding off-diagonal element of the Hessian matrix is non-zero, and may be either positive or negative. If $H_{jk} > 0$, then the $j$-th and $k$-th patterns are positively correlated, negative if inversely correlated. In general, the variances and co-variances of the parameters are obtained from the inverse of the Hessian matrix

$$\mathrm{Cov}[\,\hat{\alpha_j}\,,\,\hat{\alpha_k}\,] = \left(H^{-1}\right)_{jk} \ . \tag{136}$$

This assertion is plausible at this stage, and will become more obvious below as we consider the relationship between the Hessian matrix and the $\chi^2$ function.

## 6.3 $\chi^2$ bubbles

Imagine an $M$-dimensional bubble whose shape follows surfaces of constant $\chi^2$ in parameter space. The surface of the bubble is defined by

$$\Delta\chi^2 \equiv \chi^2 - \chi^2_{\rm min} = C \ , \tag{137}$$

where $\chi^2_{\rm min}$ is the lowest value of $\chi^2$ that occurs inside the bubble, and $C > 0$ defines the size of the bubble. Parameter values inside the bubble give better fits to the data than those outside. A small bubble defines a tight fit to the data, while a larger bubble includes parameters that give a looser fit to the data. The bubble may become greatly elongated in some directions of parameter space, when those particular combinations of parameters are poorly constrained by the data. At the same time the bubble remains narrow in directions that are tightly constrained. Thus the $\chi^2$ bubble defines a region of parameter space that provides a good fit to the data.

For a linear model with $M$ parameters, the $M$-dimensional $\chi^2$ bubble has an ellipsoidal shape, centred on and enclosing the optimal fit at $\hat{\alpha}$. For a mildly non-linear model, the bubble will be approximatly ellipsoidal in shape, at least for small bubbles in the vicinity of $\hat{\alpha}$. For larger bubbles, or a more strongly non-linear function, the shape can become distorted, curving to enclose C- or S-shaped volumes. For highly non-linear models the bubble may even pinch off to produce two or more disconnected bubbles around distinct local minima of $\chi^2$. Fitting the period of a periodic phenomenon is a good example where several solutions often occur with different multiples of the period spanning long gaps in the data.

The $\chi^2$ bubble is a geometric concept, but its size and shape depend on the coordinates we use in parameter space. If the bubble encloses is a single connected region, we can always stretch and rotate and bend the coordinate axes in parameter space so that in the new coordinates the bubble is spherical.

The Hessian matrix defines the *curvature* of the $\chi^2$ function with respect to the parameters, and hence the shape of the $\chi^2$ bubble in those particular coordinates.

$$H_{kj}(\alpha) \equiv \frac{1}{2}\frac{\partial^2\chi^2(\alpha)}{\partial\alpha_k\partial\alpha_j} = \sum_{i=1}^{N}\frac{1}{\sigma_i^2}\frac{\partial\mu_i}{\partial\alpha_k}\frac{\partial\mu_i}{\partial\alpha_j} - \sum_{i=1}^{N}\frac{X_i-\mu_i}{\sigma_i^2}\frac{\partial^2\mu_i}{\partial\alpha_k\alpha_j} \tag{138}$$

For a linear model, the second term vanishes, and the first term is independent of the parameters. This is sensible since for a linear model the $\chi^2$ is a quadratic function of the parameters, which has constant curvature.

In the vicinity of $\hat{\alpha}$, where $\chi^2(\alpha)$ takes on its minimum value, we can represent the $\chi^2$ function using a Taylor series expansion,

$$\chi^2(\alpha) = \chi^2_{\rm min} \; + \; \sum_{i=1}^{M}\sum_{j=1}^{M}\Delta\alpha_i \; H_{ij}(\hat{\alpha}) \; \Delta\alpha_j \; + \; \ldots \; , \tag{139}$$

where

$$\Delta\alpha_j \equiv \alpha_j - \hat{\alpha}_j \; . \tag{140}$$

Note that the linear term vanishes because $\hat{\alpha}$ is a local minimum. For a linear model, $\chi^2$ is a quadratic function of the scale parameters, and higher terms in the Taylor series vanish.

The symmetric positive-definite Hessian matrix has $M$ positive eigenvalues $\lambda_k$ and corresponding eigenvectors $B_k$. The eigenvectors of the Hessian matrix are orthogonal vectors that point along the principal axes of the $\chi^2$ bubble. Suppose we construct an new set of parameters $\beta_k$. Place the origin $\beta_k = 0$ at the minimum of $\chi^2$, and use the orthogonal coordinate axes defined by the principle axes of the $\chi^2$ bubble. In terms of the orthogonal parameters, the $\chi^2$ function is quadratic in each of the $\beta_k$, with no cross terms:

$$\Delta\chi^2 \equiv \chi^2 - \chi^2_{\rm min} \approx \sum_{i=1}^{M}\lambda_i \; \beta_i^2 = \sum_{i=1}^{M}\frac{\beta_i^2}{{\rm Var}\Big[\,\hat{\beta}_i\,\Big]} \; . \tag{141}$$

In these special coordinates, the Hessian matrix is diagonal, and the eigenvectors are the inverse-variances of the orthogonal scale parameters.

The new parameters $\beta_k$ are linear combinations of the original parameters $\alpha_k$,

$$\beta_k = \sum_{j=1}^{M} S_{kj} \; \Delta\alpha_j \; . \tag{142}$$

Similarly, the eigenvectors $B_k$ are linear combinations of the original patterns $P_k$.

Go on to show that co-variances of the $\alpha$ in terms of the the inverse of the Hessian matrix...

## 6.4 Summary

## 6.5 Problems

1.
2.
3.
4.
5.

# 7 Vector Space Perspectives

A very helpful way to think about the problem of fitting models to data, is to see the set of $N$ data points $X_i$ as the components of a vector $\vec{\boldsymbol{X}}$ in an $N$-dimensional *vector space*. The $M$ parameters $\alpha_j$ live in a different $M$-dimensional *parameter space*. The model $\vec{\boldsymbol{\mu}}(\alpha)$ defines a $M$-dimensional surface in the $N$-dimensional data space, since for each value of the parameters we have predicted data $\mu_i(\alpha)$, the i'th component of the vector $\vec{\boldsymbol{\mu}}(\vec{\boldsymbol{\alpha}})$.

This vector space perspective offers a very visual mental image of the data fitting and parameter estimation process. This is helpful in understanding intuitively what happens during a fit and also, as we see here, in desiging practical algorithms for finding solutions.

## 7.1 The Inner Product and Metric on Data Space

The inverse-variance weights we use in optimal data analysis define an inner-product on the data space:

$$\left\langle \vec{\boldsymbol{X}}, \vec{\boldsymbol{Y}} \right\rangle \equiv \sum_{i=1}^{N} \frac{X_i\, Y_i}{\sigma_i^2} \ . \tag{143}$$

The inner product in turn defines a metric on the data space. The length of a vector $\vec{\boldsymbol{X}}$ is

$$\left|\left| \vec{\boldsymbol{X}} \right|\right|^2 \equiv \left\langle \vec{\boldsymbol{X}}, \vec{\boldsymbol{X}} \right\rangle = \sum_{i=1}^{N} \left( \frac{X_i}{\sigma_i} \right)^2 \ , \tag{144}$$

and the angle $\theta$ between two vectors is

$$\cos\theta = \frac{\left\langle \vec{\boldsymbol{X}}, \vec{\boldsymbol{Y}} \right\rangle}{\left|\left| \vec{\boldsymbol{X}} \right|\right| \left|\left| \vec{\boldsymbol{Y}} \right|\right|} \ . \tag{145}$$

The $\chi^2(\alpha)$ that we minimize in accomplishing a fit is then the squared distance between the data vector $\vec{\boldsymbol{X}}$ and the $M$-dimensional surface $\vec{\boldsymbol{\mu}}(\alpha)$, accessible to the parameterised model,

$$\chi^2(\alpha) = \left|\left| \vec{\boldsymbol{X}} - \vec{\boldsymbol{\mu}}(\alpha) \right|\right|^2 = \sum_{i=1}^{N} \left( \frac{X_i - \mu_i(\alpha)}{\sigma_i} \right)^2 \ . \tag{146}$$

In effect, then, $\sigma_i$ is the natural unit of distance along the i'th axis of data space. The corresponding metric tensor is

$$g_{ij} = \frac{\delta_{ij}}{{\sigma_i}^2} \ . \tag{147}$$

We thus obtain a standard Euclidean geometry if we 'stretch' each axis by a factor $1/\sigma_i$, so that distances are in $\sigma$ units along all $N$ dimensions of the data space.

A pattern $\vec{\boldsymbol{P}}$ is also a vector in data space. Scaling the pattern $\vec{\boldsymbol{P}}$ by a factor $\alpha$ to fit the data $\vec{\boldsymbol{X}}$ means moving along a line in data space

$$\vec{\boldsymbol{\mu}}(\alpha) = \alpha\ \vec{\boldsymbol{P}} \tag{148}$$

until you reach the point closest to the data vector $\vec{\boldsymbol{X}}$. The closest point is found by using the inner product to project the data vector $\vec{\boldsymbol{X}}$ along the direction of the pattern vector $\vec{\boldsymbol{P}}$. This gives the closest approach at

$$\hat{\alpha} = \frac{\left\langle \vec{\boldsymbol{X}}, \vec{\boldsymbol{P}} \right\rangle}{\left\langle \vec{\boldsymbol{P}}, \vec{\boldsymbol{P}} \right\rangle} \ , \tag{149}$$

and thus we have re-written the optimal scaling result in vector notation.

Similarly, when fitting a 2-parameter model, e.g. fitting a straight line to data, or scaling 2 patterns to fit the data, we are roaming over a 2-dimensional plane in data space

$$\vec{\boldsymbol{\mu}}(\alpha_1, \alpha_2) = \alpha_1 \vec{\boldsymbol{P}}_1 + \alpha_2 \vec{\boldsymbol{P}}_2 \tag{150}$$

to locate the point of closest approach to the data vector $\vec{\boldsymbol{X}}$. The analogy continues to higher dimensions, where the $M$-parameter linear model

$$\vec{\boldsymbol{\mu}}(\alpha) = \sum_{i=1}^{M} \alpha_i \vec{\boldsymbol{P}}_i \tag{151}$$

spans an $M$-dimensional subspace of the $N$-dimensional data space.

## 7.2 Graham Schmidt Orthogonalisation

This vector space metaphor is helpful because with vectors we know how to use the inner product to decompose a vector $\vec{\boldsymbol{X}}$ into components $\vec{\boldsymbol{X}} = \vec{\boldsymbol{X}}_{\parallel} + \vec{\boldsymbol{X}}_{\perp}$ that are parallel and perpendicular to another vector $\vec{\boldsymbol{Y}}$,

$$\vec{\boldsymbol{X}}_{\parallel} = \frac{\left\langle \vec{\boldsymbol{X}}, \vec{\boldsymbol{Y}} \right\rangle}{\left\langle \vec{\boldsymbol{Y}}, \vec{\boldsymbol{Y}} \right\rangle} \vec{\boldsymbol{Y}} \ , \tag{152}$$

$$\vec{\boldsymbol{X}}_{\perp} = \vec{\boldsymbol{X}} - \vec{\boldsymbol{X}}_{\parallel} \ . \tag{153}$$

For a model with $M$ parameters, there are $M$ patterns $\vec{\boldsymbol{P}}_i$, each of which is a vector in the data space. For example, if you wish to optimise the parameters of the model, the derivatives of the predicted data with respect to each of the parameters defines a set of $M$ pattern vectors,

$$\vec{\boldsymbol{P}}_i = \frac{\partial \vec{\boldsymbol{\mu}}(\alpha)}{\partial \alpha_i} \ . \tag{154}$$

We will now use our vector-space perspective to construct an equivalent set of mutually orthogonal pattern vectors $\vec{\boldsymbol{B}}_i$. Since $\vec{\boldsymbol{B}}_i$ are orthogonal by construction, we will be able immediately to locate the best fit by optimally scaling each of these $M$ patterns in sequence.

Our first step is to construct the $M$ orthogonal vectors $\vec{\boldsymbol{B}}_i$. To do this, apply the Graham-Schmidt orthogonalization algorithm to the $M$ pattern vectors $P_i$. The $\vec{\boldsymbol{B}}_i$ span the same subspace as the $\vec{\boldsymbol{P}}_i$. We can also (optionally) re-normalize each of the orthogonal basis vectors to obtain unit-length orthogonal vectors,

$$\vec{\boldsymbol{U}}_i \equiv \frac{\vec{\boldsymbol{B}}_i}{\left|\left| \vec{\boldsymbol{B}}_i \right|\right|} \tag{155}$$

The Graham-Schmidt orthogonalization unfolds as follows:

$$\vec{\boldsymbol{B}}_1 = \vec{\boldsymbol{P}}_1 \ , \qquad \vec{\boldsymbol{U}}_1 = \frac{\vec{\boldsymbol{B}}_1}{\left|\left| \vec{\boldsymbol{B}}_1 \right|\right|} \ , \tag{156}$$

$$\vec{\boldsymbol{B}}_2 = \vec{\boldsymbol{P}}_2 - \left\langle \vec{\boldsymbol{P}}_2, \vec{\boldsymbol{U}}_1 \right\rangle \vec{\boldsymbol{U}}_1 \ , \qquad \vec{\boldsymbol{U}}_2 = \frac{\vec{\boldsymbol{B}}_2}{\left|\left| \vec{\boldsymbol{B}}_2 \right|\right|} \ , \tag{157}$$

$$\vec{\boldsymbol{B}}_3 = \vec{\boldsymbol{P}}_3 - \left\langle \vec{\boldsymbol{P}}_3, \vec{\boldsymbol{U}}_1 \right\rangle \vec{\boldsymbol{U}}_1 - \left\langle \vec{\boldsymbol{P}}_3, \vec{\boldsymbol{U}}_2 \right\rangle \vec{\boldsymbol{U}}_2 \ , \qquad \vec{\boldsymbol{U}}_3 = \frac{\vec{\boldsymbol{B}}_3}{\left|\left| \vec{\boldsymbol{B}}_3 \right|\right|} \tag{158}$$

$$, ... \tag{159}$$

Each new pattern vector $\vec{\boldsymbol{P}}_i$ gives rise to a new basis vector $\vec{\boldsymbol{B}}_i$ that is by construction orthogonal to the previous ones. The $i$-th basis vector $\vec{\boldsymbol{B}}_i$ is therefore

$$\vec{\boldsymbol{B}}_i = \vec{\boldsymbol{P}}_i - \sum_{k=1}^{i-1} \left\langle \vec{\boldsymbol{P}}_i, \vec{\boldsymbol{U}}_k \right\rangle \vec{\boldsymbol{U}}_k = P_i - \sum_{k=1}^{i-1} \frac{\left\langle \vec{\boldsymbol{P}}_i, \vec{\boldsymbol{B}}_k \right\rangle}{\left\langle \vec{\boldsymbol{B}}_k, \vec{\boldsymbol{B}}_k \right\rangle} \vec{\boldsymbol{B}}_k \ . \tag{160}$$

In this way, we construct from the $M$ original patterns $\vec{\boldsymbol{P}}_i$ a set of $M$ orthogonal basis vectors, $\vec{\boldsymbol{B}}_i$. Each pattern vector is a linear combination of the orthogonal basis vectors,

$$\vec{\boldsymbol{P}}_i = \sum_{k=1}^{i} T_{ik} \vec{\boldsymbol{B}}_k \ , \qquad T_{ik} = \frac{\left\langle \vec{\boldsymbol{P}}_i, \vec{\boldsymbol{B}}_k \right\rangle}{\left\langle \vec{\boldsymbol{B}}_k, \vec{\boldsymbol{B}}_k \right\rangle} \ . \tag{161}$$

Similarly,

$$\vec{\boldsymbol{B}}_i = \sum_{k=1}^{i} S_{ik} \vec{\boldsymbol{P}}_k \ . \tag{162}$$

Note that with $T_{ii} = S_{ii} = 1$, and $T_{ij} = S_{ij} = 0$ zero for $j > i$, the matrices $S_{ij}$ and $T_{ij}$ are inverses.

The orthogonal vectors $B_i$ span the same vector space as the original patterns $P_i$. The orthogonal model,

$$\mu(\beta) = \sum_{k=1}^{M} \beta_k B_k \ , \tag{163}$$

is therefore an equivalent re-parameterisation of the original non-orthogonal model

$$\mu(\alpha) = \sum_{i=1}^{M} \alpha_i P_i \ . \tag{164}$$

In the orthogonal basis, the original patterns are

$$P_i = \sum_{k=1}^{M} \langle\, P_i, U_k \,\rangle\ U_k = \sum_{k=1}^{M} \frac{\langle\, P_i, B_k \,\rangle}{\langle\, B_k, B_k \,\rangle}\ B_k \ . \tag{165}$$

We therefore have

$$\beta_k = \sum_{i=1}^{M} \alpha_i \langle\, P_i, U_k \,\rangle = \sum_{i=1}^{M} \alpha_i \frac{\langle\, P_i, B_k \,\rangle}{\langle\, B_k, B_k \,\rangle} \ . \tag{166}$$

An optimal fit of the orthogonal model is found by optimal scaling of each of the orthogonal basis vectors to fit the data vector $X$. From the vector-space perspective, this means finding in the subspace spanned by the basis vectors the point that is closest to $X$. Because the basis vectors $B_i$ are orthogonal, each can be scaled to fit independently of the others. The best-fit model is

$$\hat{\mu} = \sum_{i=1}^{M} \langle\, X, U_i \,\rangle\ U_i = \sum_{i=1}^{M} \frac{\langle\, X, B_i \,\rangle}{\langle\, B_i, B_i \,\rangle}\ B_i = \sum_{i=1}^{M} \hat{\beta}_i B_i \ . \tag{167}$$

Thus the optimal scale parameters in the orthogonal basis are

$$\hat{\beta}_i = \frac{\langle\, X, B_i \,\rangle}{\langle\, B_i, B_i \,\rangle} \ . \tag{168}$$

For many purposes, the re-parameterized orthogonal model is sufficient. However, in some cases you may need to have one or more of the original scale parameters $\alpha_i$, which scale factors of the original patterns $P_i$. In such cases you may need to invert the transformation matrix $M$, where

$$\beta_k = \sum_{i=1}^{M} M_{ki} \alpha_i = \sum_{i=1}^{M} \frac{\langle\, P_i, B_k \,\rangle}{\langle\, B_k, B_k \,\rangle} \alpha_i \ , \tag{169}$$

$$\alpha_i = \sum_{k=1}^{M} \left(M^{-1}\right)_{ik} \beta_k \ . \tag{170}$$

# 8 Badness-of-Fit Statistics

Data analysis is the process by which we learn new things about the universe by fitting models to data. We must be creative in designing or selecting the model. Several alternatives may be on offer. There is no guarantee that we will make the right choice, or indeed that our menu of models will include a correct one. However, we are sometimes granted the opportunity to discover when a model is wrong, by quantitatively comparing its predictions against observational data. Thus we may expect one by one to reject false models, narrowing our focus to models that pass each successive test.

In fitting a model to data points, we must give the model a fair shot. This means adjusting some parameters of the model to let it achieve the best possible fit it can manage. Parameters optimized, the model can then be judged on the basis of whether or not that fit is good enough. The parameter values that optimize a fit are found in practice by minimizing some **badness-of-fit** statistic, which quantifies the mismatch between the observed data points and the corresponding predicted data points produced by the model.

Often we are interested in the values of some of the parameters, and in their uncertainties. In optimal fitting, we seek unbiased estimates of parameters that have minimum variance. Optimal methods try to make the best use of the data by eliminating systematic errors while keeping statistical errors to a minimum. As with the simple case of averaging data points, the accuracy of the parameters from an optimal fit should always improve when new data points are added.

## 8.1 Least Squares Fitting

Many choices are possible for the badness-of-fit statistic. A **Least-Squares fit** finds parameter values that minimize the **sum of squared residuals**

$$\mathrm{SSR} \equiv \sum_{i=1}^{N} \epsilon_i^2 \ , \tag{171}$$

where the **residuals**

$$\epsilon_i = X_i - \mu_i(p) \tag{172}$$

are the differences between the data values $X_i$ and the corresponding predicted values $\mu_i$, which depend on the parameters $p$.

Consider a simple example. Suppose we have a set of $N$ measurements $X_i \pm \sigma_i$. Our model is that these are measurements of some quantity $x$, whose value we would like to determine. The model has a single parameter, $x$. The badness-of-fit statistic is the sum of squared residuals,

$$\mathrm{SSR}(x) = \sum_{i=1}^{N} \left(X_i - x\right)^2 \ . \tag{173}$$

To minimize this, set to 0 its derivative with respect to $x$,

$$0 = \frac{\partial \mathrm{SSR}}{\partial x} = -2 \sum_{i=1}^{N} \left(X_i - x\right) \ . \tag{174}$$

The solution is a simple average of the data points

$$\bar{X} = \frac{1}{N} \sum_{i=1}^{N} X_i \ , \qquad\qquad \mathrm{Var}\left[\, \bar{X} \,\right] = \frac{1}{N^2} \sum_{i=1}^{N} \sigma_i^2 \ . \tag{175}$$

Least-squares fitting is not an optimal method unless all the data points have the same noise variance. The accuracy is degraded when the more noisy data points are combined with the less noisy ones.

## 8.2 Median Fitting

Another possible badness-of-fit statistic is the **sum of absolute values of residuals**

$$\mathrm{SAR} \equiv \sum_{i=1}^{N} \left|\, \epsilon_i \,\right| \ . \tag{176}$$

This also is not an optimal fitting method. In fact, it has an even larger variance than the least squares method. It does, however, have one redeming feature. By reducing the penalty for a bad fit from the square to the absolute value of the residual, the method pays less attention to outliers.

Such methods are referred to as **robust**. Robust methods are most useful when the highest accuracy is not required, and a small number of data points have highly discrepant values that tend to dominate an averaging of the data points. For example in CCD data a minority of pixels get hit by cosmic rays, launching their data values into outer space.

The method is called **median fitting** because in the particular problem of estimating the average value of data points, the result is the median of the data points. To see this, note that for this problem

$$\mathrm{SAR}(x) = \sum_{i=1}^{N} |\, X_i - x \,| = \sum_{X_i<x} (x - X_i) + \sum_{X_i>x} (X_i - x) \ . \tag{177}$$

To minimize $\mathrm{SAR}(x)$, set to 0 the derivative with respect to $x$,

$$0 = \frac{\partial \mathrm{SAR}}{\partial x} = \sum_{X_i<x} - \sum_{X_i>x} \ . \tag{178}$$

Thus SAR is minimized when equal numbers of data points are larger and smaller than $x$. That's the definition of the **median**.

## 8.3 $\chi^2$ Fitting

When data points have different noise variances $\sigma_i$, an optimal fit is achieved by minimizing the $\chi^2$ statistic

$$\chi^2 \equiv \sum_{i=1}^{N} \eta_i^2 \ , \tag{179}$$

i.e. the sum of squares of **normalized residuals**

$$\eta_i = \epsilon_i / \sigma_i \ . \tag{180}$$

If $M$ parameters are optimized to fit $N$ data points, the residuals of the best fit are said to have $\nu = N - M$ **degrees of freedom**.

Assuming that the model is unbiased, and that noise affecting the data points arises from independent Gaussian probability distributions,

$$\mathrm{p}(\,\eta_i\,) = \frac{\exp\left\{\,-\eta_i^2/2\,\right\}}{(2\pi)^{1/2}} \ , \tag{181}$$

then the probability distribution of the $\chi^2$ statistic is that obtained by summing the squares of $\nu$ independent Gaussian random variables. The expected value of the $\chi^2$ distribution with $\nu$ degrees of freedom is

$$\left\langle\, \chi^2 \,\right\rangle = \nu \ , \tag{182}$$

and its variance is

$$\mathrm{Var}\left[\, \chi^2 \,\right] = 2\nu \ . \tag{183}$$

A **reduced** $\chi^2$ statistic, denoted $\chi^2_\nu$ is obtained by dividing $\chi^2$ by the number of degrees of freedom so that its expected value is 1. Thus

$$\chi^2_\nu \equiv \frac{\chi^2}{\nu} \sim 1 \pm \left(\frac{2}{\nu}\right)^{1/2} \ . \tag{184}$$

The reduced $\chi^2$ statistic is used to decide whether or not a model has achieved an acceptable fit to the data.

## 8.4 Failure of $\chi^2$ in Estimating Error Bars

The results above assume that the error bars $\sigma_i$ are known. What if $\sigma_i$ are unknown? In this case it would be natural to introduce a new parameter $\sigma$ to represent the unknown measurement error bars, and then try to estimate $\sigma$ by minimizing $\chi^2$. However, this leads to a disaster. When we adjust parameters, including $\sigma$, to minimize $\chi^2$, we discover that $\chi^2$ is minimized by $\sigma \to \infty$, for then $\chi^2 \to 0$. This silly result indicates that $\chi^2$ fitting is incomplete. To escape from this disaster, we will need to develop a more powerful perspective, to which we now turn.

# 9 Surviving Outliers

Expect the unexpected. Gaussian probability maps may represent the majority of the data points, but what about those nasty outliers? One wicked data point way off the track can wreck your best effort at optimal data analysis.

In astronomical data, rogue data points are very common. CCD detectors are sensitive to light but also to cosmic rays. Cosmic rays hitting CCD detectors during an exposure can deposit a large number of electrons in one or a few pixels around the impact point, obliterating information about the light signal in that region.

## 9.1 Median filtering

A quick and dirty approach often used to protect results from outlaw data points, is to combine data values using a median, rather than averaging the data values. The median is a noisier statistic than the optimal average, when analysing a well behaved cluster of data points, but the median is far more robust to rogue data points.

The mean value of a pack of data points is dramatically affected if one data point is ejected to a large displacement. Suppose you have a cluster of $N$ data points each with mean value $\mu$ and standard deviation $\sigma$. The uncertainty in computing the centroid of these will be $\sigma/\sqrt{N}$. If one data point, selected at random, is moved a distance $L$ from the cluster, the mean value $\bar{X}$ shifts in the same direction by $\Delta\bar{X} = \frac{L}{N}$. If $L/N > \sigma/\sqrt{N}$, the bias caused by one rogue data point dominates the error budget.

The median is more robust. The median lies in the middle of the cluster of data points, with half of the data on one side and half on the other. A single point ejected at random has no effect on the median if the ejected point stays on the same side of the cluster. If it is ejected across to the other side of the cluster, the median line must move in the same direction enough to cross one member. The median doesn't need to move very far, however, because it lies in the midst of the crowded cluster of data points.

In the centre of a Gaussian cluster, the mean distance between data points is $\sqrt{2\pi}\sigma/N$. Thus the median is robust to large errors affecting a small random fraction of the data points. A rogue data point that moves away from the cluster has no effect on the median, while one that crosses the cluster shifts the median by typically $\sqrt{2\pi}\sigma/N$, which is less than the uncertainty in the median $??\sigma/\sqrt{N}$, provided $N$ is more than a few.

## 9.2 $\sigma$-clipping

Another approach is to identify exactly which data points are likely to be outliers, and reject them. This can retain some of the accuracy of the optimal average, provided the rogue data point can be identified with confidence.

# 10 Maximum Likelihood Fitting

A **maximum likelihood fit** adjusts the parameters of a model to maximize the probability of the data under the assumption that the model is correct. Your model provides for each data point $x_i$ a predicted value $\mu_i(\alpha)$ and an error bar $\sigma_i(\alpha)$, one or both of which depend on some or all of the model parameters $\alpha$. The conditional probability of the data, $\mathrm{p}(\, X \,|\, \alpha \,)$, is both a probability map over the $N$-dimensional data space $X$, and a function of the $M$ model parameters $\alpha$. The probability map assumes that the model is correct, and that the parameters $\alpha$ have their correct values. If the model is wrong, or if your parameters are not close to the right ones, then the probability map will miss most of the data points. If you have the right model, and your parameters are close, then the probability map will cover most of the data points. The general aim of maximum likelihood fitting is to make the probability map overlap most of the data points. This would seem to be quite a reasonable approach.

Let's look more closely at the conditional probability $\mathrm{p}(\, X \,|\, \alpha \,)$. Since this is a probability map on $X$, it must be normalized to unity when integrated over the full volume of the $N$-dimensional data space,

$$\int \mathrm{p}(\, X \,|\, \alpha \,)\ \mathrm{d}^N X = 1 \ . \tag{185}$$

Here $\mathrm{d}^N X$ is a differential volume of data space, $\mathrm{d}^N X = \prod_{i=1}^N \mathrm{d}X_i = \mathrm{d}X_1\ \mathrm{d}X_2\ ...\mathrm{d}X_N$. If the errors affecting different data points are independent, then $\mathrm{p}(\, X \,|\, \alpha \,)$ simply multiplies together the probabilities $\mathrm{p}(\, X_i \,|\, \alpha \,)$ associated with each data point, each axis of data space,

$$\mathrm{p}(\, X \,|\, \alpha \,) = \prod_{i=1}^{N} \mathrm{p}(\, X_i \,|\, \alpha \,) \ . \tag{186}$$

What we have at this stage is a prediction for what data values are likely and unlikely to arise if the model is correct and the parameters are $\alpha$. We don't yet have any constraint on the possible values of $\alpha$.

Now, conduct your experiment. The result is a specific dataset $x$ consisting of $N$ data points $x_i$, for $i = 1...N$. Plug those numbers into $\mathrm{p}(\, X \,|\, \alpha \,)$. This collapses what was a full data space of possibilities $X$ to a single point $x$, your specific dataset. What then remains is $\mathrm{p}(\, x \,|\, \alpha \,)$, no longer a distribution on $X$, but still a function of the parameters $\alpha$. In this way, your data points $x$ define the **likelihood function** $L(\alpha)$ on the parameter space of your model:

$$L(\alpha) \equiv \mathrm{p}(\, x \,|\, \alpha \,). \tag{187}$$

The maximum likelihood method assigns a relative probability on parameter space proportional to $L(\alpha)$. $L(\alpha)$ itself is not a probability map because it is not normalized to an integral of 1. If $\int L(\alpha)\, \mathrm{d}^M \alpha$ is finite, then you can normalize $L(\alpha)$ to make it a probability map. But often this integral diverges, and $L(\alpha)$ then gives only relative probabilities for different parameter values $\alpha$.

To perform a maximum likelihood fit, you must now hunt through the parameter space $\alpha$ to find the maximum likelihood parameter values $\alpha = \hat{\alpha}$ that maximize your likelihood function $L(\alpha)$. In practice, instead of maximising $L(\alpha)$ it is equivalent, and usually much easier, to maximise its natural logarithm. To maintain the closest possible connection between maximum likelihood fitting and $\chi^2$ fitting, note that maximising $L(\alpha)$ is equivalent to minimising the corresponding "**Badness of Fit**" statistic

$$B(\alpha) \equiv -2 \ln L(\alpha) \ . \tag{188}$$

This, as we will see below, is equivalent to $\chi^2$ in the case of Gaussian errors with known standard deviations.

## 10.1 Gaussian Errors: $\chi^2$ Fitting as a Special Case

Let's look in more detail at the special case of Gaussian errors. This case will help you to understand the relationship between the maximum likelihood fitting method and $\chi^2$ fitting. The Gaussian probability map associated with each data point is

$$\mathrm{p}(\, x_i \,|\, \alpha \,) = \frac{\exp\left\{ -\eta_i^2(\alpha)/2 \right\}}{(2\pi)^{1/2}\, \sigma_i(\alpha)} \ , \tag{189}$$

where as usual the normalized residuals are $\eta_i(\alpha) = (x_i - \mu_i(\alpha))\,/\sigma_i(\alpha)$. Since the errors on different data points are independent, multiply the Gaussians together to form the likelihood function:

$$L(\alpha) = \frac{\exp\left\{ -\frac{1}{2}\sum_{i=1}^{N} \eta_i^2(\alpha) \right\}}{(2\pi)^{N/2} \prod_{i=1}^{N} \sigma_i(\alpha)} = \frac{\exp\left\{ -\frac{1}{2}\,\chi^2(\alpha) \right\}}{Z_D(\alpha)} \ , \tag{190}$$

where

$$\chi^2(\alpha) = \sum_{i=1}^{N} \eta_i^2(\alpha) \ , \tag{191}$$

and the **partition function**,

$$Z_D(\alpha) = (2\pi)^{N/2} \prod_{i=1}^{N} \sigma_i(\alpha) \ , \tag{192}$$

represents the volume of data space that has significant probability. To maximize $L(\alpha)$, minimize

$$B(\alpha) \equiv -2\ln L(\alpha) = \chi^2(\alpha) + 2\ln Z_D(\alpha) = \sum_{i=1}^{N} \left( \frac{x_i - \mu_i(\alpha)}{\sigma_i(\alpha)} \right)^2 + 2\sum_{i=1}^{N} \ln \sigma_i(\alpha) + \frac{N}{2} \ln 2\pi \ . \tag{193}$$

Notice that $B(\alpha) = -2\ln L(\alpha)$ is $\chi^2(\alpha)$ plus the partition function term $2\ln Z_D(\alpha)$. When the error bars $\sigma_i$ are known, the partition function $Z_D$ is independent of $\alpha$. In this case maximizing the likelihood $L(\alpha)$ is exactly equivalent to minimizing $\chi^2(\alpha)$. You see, the $\chi^2$ fitting method is a special case of maximum likelihood fitting. When you have independent Gaussian errors with known error bars, minimising $\chi^2(\alpha)$ is equivalent to maximising $L(\alpha)$.

The analogy with $\chi^2$ fitting suggests that we can estimate maximum likelihood parameters and their confidence regions simply by substituting $B = -2\ln L$ in place of $\chi^2$. All of the results we have developed for $\chi^2$ fitting then carry over. For example, a 1-parameter 1-sigma confidence interval on the parameter $\alpha$ is the region around $\hat{\alpha}$ in which $B(\alpha)$ is not more than 1 unit above the minimum value $B\,(\hat{\alpha})$. Expanding $B(\alpha)$ in a Taylor series around the minimum at $\hat{(}\alpha)$, and keeping the quadratic terms, an estimate for the variance of $\hat{\alpha}$ from the curvature at the minimum is

$$\mathrm{Var}[\,\hat{\alpha}\,] = \left. \frac{2}{\partial^2 B/\partial\alpha^2} \right|_{\hat{\alpha}} \ . \tag{194}$$

When the error bars $\sigma_i(\alpha)$ depend on parameters $\alpha$, then the minima of $B = -2\ln L$ and $\chi^2$ do not coincide. In this case the partition function term $2\ln Z_D(\alpha)$ penalises models with large error bars. These models spread their probability over a larger volume of data space, and this lowers their chances of producing the observed data $x$. Thus maximum-likelihood fits emerge from a competition between the $\chi^2$ term and the $2\ln Z_D$ term. The $\chi^2$ term gives preference to models that fit the observed data well, while the $2\ln Z_D$ term penalises models that can also fit many other data sets equally well. This is an example of Ocham's razor – when two models fit the data equally well, prefer the one that is simplest. In this case the model with smaller $Z_D$, thus covering a relatively small volume of data space, is considered simpler than one with larger $Z_D$ that can reach a much broader range of data space.

## 10.2 Poisson Data

If maximum likelihood fitting is so closely related to $\chi^2$ fitting, why bother? Why not just keep it simple and stick with a $\chi^2$ fit? Indeed. Usually the error distributions are well approximated by Gaussians and it's perfectly fine to use $\chi^2$ fitting. But there are important and interesting exceptions when the errors cannot be assumed to be Gaussian.

Perhaps the most important case is Poisson data, in which objects are cast into bins, and the data are the number of objects collected by each bin. In astronomy the best example is photon counting data. A photon count $n$ has the probability map

$$\mathrm{p}(\,n\,|\,\mu\,) = \frac{\mu^n\, e^{-\mu}}{n!} \ , \tag{195}$$

where $n \geq 0$ is the integer number of photon counts, and the parameter $\mu$ provides both the expected value and the variance

$$\langle\, n \,\rangle = \mathrm{Var}[\, n \,] = \mu \ . \tag{196}$$

At high count rates, $\mu \gg 1$, the Poisson distribution can be approximated by a Gaussian with equivalent mean and variance. This is a consequence of the Central Limit Theorem, where the mean and variance of the underlying distribution is retained while other information about the shape of the distribution is lost as more and more photons are collected. At low count rates, $\mu \sim 1$, the Gaussian remains summetric and errs by assigning significant probability to $n < 0$. The correct Poisson distribution is restricted to $n \geq 0$ and maintains $\mu = \sigma$ by developing a long exponential tail to high values. At very low count rates, $\mu \ll 1$, most of the Poisson data values are 0, with an occasional 1 or 2. Maximum likelihood methods are required to correctly analyse low-count Poisson datasets.

Low Poisson count rates arise very often in X-ray astronomy. X-ray counts can be so low that most of the pixels in an X-ray image have 0 counts. Fortunately, the X-ray background count rate is also very low, so that a cluster of only 3 or 4 photons in nearby pixels may be sufficient to detect a new source. This is a case where $\chi^2$ fitting could be misleading, and the full power of maximum likelihood methods are needed to analyze the data.

## 10.3 Optimal Average of Poisson Data

Measure a star $N$ times to find $n_i$ photons counted during equal exposures $i = 1...N$. The $n_i$ are not all the same, due to photon-counting statistics. What is the brightness of the star, and how uncertain is it? Assuming the star is constant, and the exposure times are equal, the predicted count $\mu$ is the same for each time bin. The likelihood function is then

$$L(\mu) = \mathrm{p}(\, n_1, n_2, ..., n_N \,|\, \mu \,) = \prod_{i=1}^{N} \mathrm{p}(\, n_i \,|\, \mu \,) = \prod_{i=1}^{N} \frac{\mu^{n_i}\, e^{-\mu}}{n_i!} \ . \tag{197}$$

Maximizing the likelihood is equivalent to minimising

$$B(\mu) \equiv -2 \ln L(\mu) = 2 \sum_{i=1}^{N} [\, \mu - n_i \ln \mu + \ln (n_i!) \,] \ . \tag{198}$$

The derivatives with respect to $\mu$ are

$$\frac{\partial B}{\partial \mu} = 2 \sum_{i=1}^{N} \left( 1 - \frac{n_i}{\mu} \right) \ , \qquad\qquad \frac{\partial^2 B}{\partial \mu^2} = \frac{2}{\mu^2} \sum_{i=1}^{N} n_i \ . \tag{199}$$

Setting $\partial B / \partial \mu = 0$, to minimise $B$ and maximize $L$, gives

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} n_i \ . \tag{200}$$

Thus with Poisson data the maximum-likelihood estimator of $\mu$ is a simple average of the counts in the $N$ bins. The variance, from the second derivative, is

$$\mathrm{Var}[\, \hat{\mu} \,] = \frac{2}{\left. \frac{\partial^2 B}{\partial \mu^2} \right|_{\hat{\mu}}} = \frac{\hat{\mu}^2}{\sum_{i=1}^{N} n_i} = \frac{1}{N^2} \sum_{i=1}^{N} n_i = \frac{\hat{\mu}}{N} \ . \tag{201}$$

Note that for a single measurement the variance of the maximum likelihood estimate is $\mathrm{Var}[\, \hat{\mu} \,] = \hat{\mu}$, as expected for Poisson data, and that this improves as $1/\sqrt{N}$ when combining $N$ independent measurements.

## 10.4 Error Bars belong to the Model, not to the Data

You may at this stage be surprised at the result, that the maximum likelihood estimate is a simple average of the individual counts. Why is it not an optimal average? If you use the $\sqrt{n}$ rule, i.e. the data are $n_i \pm \sigma_i$, with $\sigma_i = \sqrt{n_i}$,

then optimal averaging appears gives a different result:

$$\hat{\mu} = \frac{1}{\sum\limits_{i=1}^{N} \frac{1}{\sigma_i^2}} = \frac{1}{\sum\limits_{i=1}^{N} \frac{1}{n_i}} \ . \tag{202}$$

Which one is right? Notice that your optimal average gives unequal weight to the data points because some are lower and some higher than the average, just by pure chance. This is the mistake. The low points should not have smaller error bars than the high ones. The optimal average using $\sqrt{n}$ for the error bars gives too much weight to data points that are low by chance, biasing the result to low values. If one of the $n_i$ happened to be zero, that point would get infinite weight, and the optimal average would be zero, an obviously silly result.

You can avoid this common mistake if you remember that **error bars belong to the model**, not to the data points. The model says the expected count rate is $\langle\, n \,\rangle = \mu$ and the uncertainty $\sigma\,[n] = \sqrt{\langle\, n \,\rangle} = \sqrt{\mu}$, i.e. the uncertainty is the same for every data point, not larger for higher counts and smaller for smaller counts. Since the error bars are the same for every point, the correct optimal average is an equally-weighted average.

## 10.5 Optimal Scaling with Poisson Data

The maximum likelihood analysis of averaging data points yielded a surprisingly simple result. What about optimal scaling to fit Poisson data? In this case the binned counts $n_i$ have expected values $\langle\, n_i \,\rangle = \mu_i = \alpha\, P_i$, a pattern $P_i$ times a scale factor $\alpha$. The likelihood is

$$L(\alpha) = \prod_{i=1}^{N} \mathrm{p}(\, n_i \,|\, \alpha\,) = \prod_{i=1}^{N} \frac{\mu_i^{n_i} e^{-\mu_i}}{n_i!} = \prod_{i=1}^{N} \frac{(\alpha\, P_i)^{n_i} \exp\{\, -\alpha\, P_i \,\}}{n_i!} \ , \tag{203}$$

with the corresponding "Badness of Fit"

$$B(\alpha) \equiv -2\, \ln L(\alpha) = 2 \sum_{i=1}^{N} [\, \mu_i - n_i\, \ln \mu_i + \ln\,(n_i!)\,] \,. \tag{204}$$

Derivatives with respect to the scale parameter $\alpha$ are

$$\frac{\partial B}{\partial \alpha} = \sum_{i=1}^{N} \frac{\partial B}{\partial \mu_i} \frac{\partial \mu_i}{\partial \alpha} = 2 \sum_{i=1}^{N} \left(1 - \frac{n_i}{\mu_i}\right) P_i = 2 \sum_{i=1}^{N} \left(P_i - \frac{n_i}{\alpha}\right) \ , \qquad \frac{\partial^2 B}{\partial \alpha^2} = \frac{2}{\alpha^2} \sum_{i=1}^{N} n_i \ . \tag{205}$$

Setting $\partial B/\partial \alpha = 0$, leads to

$$\hat{\alpha} = \frac{\sum\limits_{i=1}^{N} n_i}{\sum\limits_{i=1}^{N} P_i} \ . \tag{206}$$

Thus for Poisson data the maximum-likelihood estimator $\hat{\alpha}$ for the scale parameter $\alpha$ is found by dividing the total observed counts $\sum n_i$ by the total predicted counts $\sum P_i$. The variance, from the second derivative, is

$$\mathrm{Var}[\, \hat{\alpha}\,] = \frac{2}{\left.\frac{\partial^2 B}{\partial \alpha^2}\right|_{\hat{\alpha}}} = \frac{\hat{\alpha}^2}{\sum\limits_{i=1}^{N} n_i} = \frac{\sum\limits_{i=1}^{N} n_i}{\left(\sum\limits_{i=1}^{N} P_i\right)^2} \ . \tag{207}$$

Note that for $P_i = 1$ these formulae reduce to the results obtained above for optimal averaging.

## 10.6 Gaussian Approximation to Poisson Noise

At high count levels Poisson distributions resemble Gaussians with equal mean and variance, $\mu_i = \sigma_i^2$. We expect this "pseudo-Poisson" model to perform well at high count rates. Maximum likelihood for this model minimises

$$B \equiv -2\,\ln L = \sum_{i=1}^{N} \frac{(x_i-\mu_i)^2}{\mu_i} + \ln\mu_i + \text{const}\ . \tag{208}$$

The gradient with respect to a parameter $\alpha$ is then

$$\frac{\partial B}{\partial\alpha} = \sum_{i=1}^{N} \frac{\partial\mu_i}{\partial\alpha}\left[\frac{(x_i-\mu_i)^2}{{\mu_i}^2} + \frac{1-2\,(x_i-\mu_i)}{\mu_i}\right]\ , \tag{209}$$

and the curvature with respect to parameters $\alpha$ and $\beta$ is

$$\frac{\partial^2 B}{\partial\alpha\,\partial\beta} = \sum_{i=1}^{N} \frac{\partial^2\mu_i}{\partial\alpha\,\partial\beta}\left[\frac{(x_i-\mu_i)^2}{{\mu_i}^2} + \frac{1-2\,(x_i-\mu_i)}{\mu_i}\right] + \sum_{i=1}^{N} \frac{\partial\mu_i}{\partial\alpha}\frac{\partial\mu_i}{\partial\beta}\left[\frac{2\,(x_i-\mu_i)^2}{{\mu_i}^3} - \frac{1+4\,(x_i-\mu_i)}{{\mu_i}^2} + \frac{2}{\mu_i}\right]\ . \tag{210}$$

For **optimal scaling**, a known pattern $P_i$ scaled by a factor $\alpha$, the predicted data are $\mu_i = \sigma_i^2 = \alpha\,P_i$, and $\partial\mu_i/\partial\alpha = P_i$. The model is linear in the scale parameter $\alpha$. As $\alpha$ varies, the badness-of-fit varies with a slope and curvature

$$\frac{\partial B}{\partial\alpha} = \sum_{i=1}^{N} P_i + \frac{N}{\alpha} - \frac{1}{\alpha^2}\sum_{i=1}^{N}\frac{x_i^2}{P_i}\ , \qquad \frac{\partial^2 B}{\partial\alpha^2} = -\frac{N}{\alpha^2} + \frac{2}{\alpha^3}\sum_{i=1}^{N}\frac{x_i^2}{P_i}\ . \tag{211}$$

Optimising $\alpha$ via $\partial B/\partial\alpha = 0$ yields the quadratic equation

$$0 = \alpha^2\,\frac{\partial B}{\partial\alpha} = \alpha^2\,\sum_{i=1}^{N} P_i + \alpha\,N - \sum_{i=1}^{N}\frac{x_i^2}{P_i}\ . \tag{212}$$

The positive root gives the solution we want. We can write this in two ways

$$\hat\alpha = \frac{\left[1+\frac{4}{N^2}\left(\sum_{i=1}^{N}\frac{x_i^2}{P_i}\right)\left(\sum_{i=1}^{N}P_i\right)\right]^{1/2} - 1}{\frac{2}{N}\sum_{i=1}^{N}P_i} = \frac{\frac{2}{N}\sum_{i=1}^{N}\frac{x_i^2}{P_i}}{1+\left[1+\frac{4}{N}\left(\sum_{i=1}^{N}\frac{x_i^2}{P_i}\right)\left(\sum_{i=1}^{N}P_i\right)\right]^{1/2}}\ . \tag{213}$$

The corresponding variance is

$$\text{Var}[\,\hat\alpha\,] = \frac{2}{\left.\frac{\partial^2 B}{\partial\alpha^2}\right|_{\hat\alpha}} = \frac{\hat\alpha^2}{\frac{1}{\hat\alpha}\sum_{i=1}^{N}\frac{x_i^2}{P_i} - \frac{N}{2}} = \frac{\hat\alpha^2}{\frac{N}{2}\left[1+\frac{4}{N}\left(\sum_{i=1}^{N}\frac{x_i^2}{P_i}\right)\left(\sum_{i=1}^{N}P_i\right)\right]^{1/2}}\ . \tag{214}$$

For the **optimal average**, set $P_i = 1$ to find

$$\hat\alpha = \frac{\frac{2}{N}\sum_{i=1}^{N}x_i^2}{1+\left(1+\frac{4}{N}\sum_{i=1}^{N}x_i^2\right)^{1/2}} \qquad \text{Var}[\,\hat\alpha\,] = \frac{\hat\alpha^2}{\frac{N}{2}\left(1+4\sum_{i=1}^{N}x_i^2\right)^{1/2}} \tag{215}$$

This Gaussian approximation is actually more complicated than the full Poisson analysis.

## 10.7 Schechter Fits to Binned Galaxy Luminosities

Let's look at a specific example, fitting a **Schechter distribution** to binned galaxy luminosities. The Schechter luminosity distribution,

$$\frac{\mathrm{d}n}{\mathrm{d}L}\,(L\,|\,n_\star, L_\star, \alpha) = \left(\frac{n_\star}{L_\star}\right)\left(\frac{L}{L_\star}\right)^\alpha e^{-L/L_\star}\ , \tag{216}$$

This model has two non-linear parameters, $\alpha$ and $L_\star$, and one linear parameter, $n_\star$. A power-law with slope $\alpha$ describes the luminosity distribution of faint galaxies. There is an exponential cutoff on the bright end at the characteristic bright-galaxy luminosity $L_\star$. The scale parameter $n_\star$ gives the number of galaxies per unit $\ln L$ predicted by the power-law when evaluated at $L_\star$.

Assuming that we have luminosities for a complete volume-limited sample of galaxies, we cast these into a set of luminosity bins. As galaxies cover a wide range of luminosity, it is convenient to employ luminosity bins equally spaced in $\log L$. Typically the luminosities are converted to magnitudes, $m = m_0 - 2.5\log_{10} L$, and the luminosity bins are set at 0.5 mag intervals. The $N$ data points $n_i$ are the number of galaxies with luminosity $L$ falling within the limits of the $i$-th luminosity bin, $L_i^- < L < L_i^+$, for $i = 1, ..., N$.

The Schechter model predicts the expected galaxy count in each luminosity bin, $\mu_i = n_\star P_i\,(L_\star, \alpha)$, where $n_\star$ is the scale parameter, and the $P_i$ are obtained by integrating the Schechter distribution over the luminosity bins,

$$\mu_i = n_\star\,P_i\,(L_\star, \alpha) = n_\star \int_{L_i^-}^{L_i^+} \frac{\mathrm{d}L}{L_\star}\left(\frac{L}{L_\star}\right)^\alpha e^{-L/L_\star}\ . \tag{217}$$

The binned galaxy counts $n_i$ are Poisson data, with probability distributions

$$\mathrm{p}(\,n_i\,|\,n_\star, L_\star, \alpha\,) = \frac{n_i^{\mu_i}\ e^{-\mu_i}}{n_i!}\ . \tag{218}$$

Thus when we estimate the scale parameter $n_\star$, we are scaling the pattern $P_i$ to fit the data $n_i$. The maximum likelihood estimator, as found above, is the ratio of total counts to predicted counts, i.e.

$$\hat{n}_\star = \frac{\sum_{i=1}^N n_i}{\sum_{i=1}^N P_i} \qquad\qquad \mathrm{Var}[\,\hat{n}_\star\,] = \frac{\hat{n}_\star}{N} = \frac{\sum_{i=1}^N n_i}{N\sum_{i=1}^N P_i}\ . \tag{219}$$

Fig. 16 shows the results of Schechter fits to the distribution of 100 galaxies when cast into $N = 17$ luminosity bins. Of the 17 luminosity bins, the 5 brightest are empty. The assignment of error bars to the empty bins is somewhat problematic. The top panel in Fig. 16 shows the correct maximum-likelihood fit allowing for the Poisson error distributions on the galaxy counts. The other three rows show results of different approximate treatments of the error bars, to be discussed below.

The galaxy luminosities in this example are random samples drawn from the Schechter distribution, obtained by using `ran_schechter.for`. For each $L_\star$ and $\alpha$, predicted counts $P_i$ are computed by numerically integrating the Schechter distribution over the 0.5-mag luminosity bins. The resulting pattern $P_i$ is then scaled to fit the observed counts, yielding the maximum likelihood estimate $\hat{n}_\star$ for the scale parameter $n_\star$. Constraints on the non-linear parameters $L_\star$ and $\alpha$ are then found by using a 2-dimensional grid search, optimising $n_\star$ in each case, and evaluating $B = -2\ln L(\hat{n}_\star, L_\star, \alpha)$, with $n_\star$ set to its optimal value $\hat{n}_\star$. The lowest value of $B$ gives the maximum-likelihood estimates $\hat{L}_\star$ and $\hat{\alpha}$, and the uncertainties (covariance matrix and confidence regions) are obtained by plotting contours of $B$ at appropriate levels above the minimum value.

Note that there is a correlation between $L_\star$ and $\alpha$ in the sense that a brighter cutoff luminosity $L_\star$ can be accommodated if the power-law slope $\alpha$ is reduced to favour fainter galaxies. The 1-$\sigma$ contour is well approximated by an ellipse, indicating that the parameters are well enough constrained that the non-linearity in the model is not very important over this range. The 2-$\sigma$ and 3-$\sigma$ contours deviate progressively from concentric ellipses, indicating that the non-linearity is becoming important in distorting the wings of the probability distribution.

In the second row of Fig. 16, the 5 empty bins are omitted from the fit, leaving 12 bins that collected 1 or more galaxies. This option effectively assigns an infinite error bar to the counts in the empty bins. This diminished constraint on the bright end allows the model to employ very large values of $L_\star$, provided $\alpha$ is lowered appropriately. The uncertainties are significantly larger than in the correct Poisson treatment.

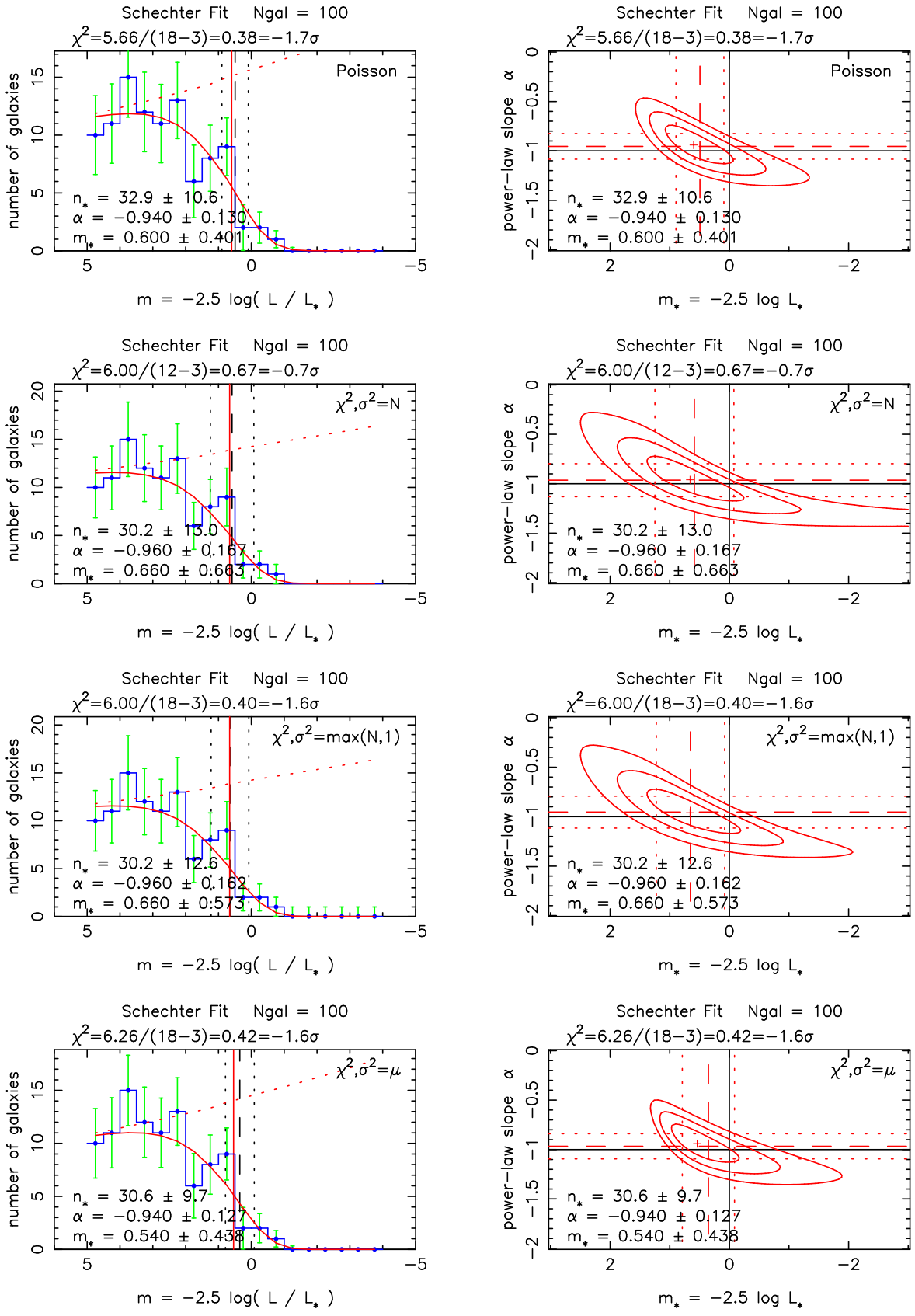


Figure 16: Schechter fits to binned galaxy luminosities, with 100 galaxies cast into 0.5 mag luminosity bins. The scale parameter $n_\star$ is found for each $L_\star$ and $\alpha$ by scaling the predicted counts to match the observed counts. Constraints on the non-linear parameters $L_\star$ and $\alpha$ are shown by the likelihood contours $\Delta\left(-2\ln L\right) = 2.3$, 6.17, and 11.8 correspond to 1, 2, and 3-$\sigma$ 2-parameter confidence regions. Four rows (top to bottom) compare results for different noise models: Poisson, Gaussian with $\sigma^2 = n$ but ignoring empty bins with $n = 0$, Gaussian including empty bins with $\sigma^2 = max(1, n)$, and Gaussian with $\sigma^2 = \mu$.

In the next row, the empty bins are assigned an error bar of 1 count, i.e. $n_i = 0 \pm 1$. This gives some constraint on the bright end, but that constraint is still too weak, and the result remains significantly worse than the correct Poisson treatment.

In the bottom row the Poisson distributions are approximated by Gaussians with $\sigma^2 = \mu$. The parameters are found by minimising $\chi^2$, but using $\sigma_i = n_\star P_i$ for the error bar on $n_i$. The solution for $\hat{n}_\star$ is found by solving a quadratic equation. The result here much closer, though not identical to, the correct Poisson treatment.

The effects illustrated by these Schechter fits may be considered typical of parameter estimation problems in which a model is fitted to Poisson data with low count levels. Minimising $\chi^2$ is adequate when the counts are high, because the Poisson error distributions are then well approximated by Gaussians. However, the fits are biased toward low counts if the error bars are calculated using observed counts, rather than the predicted counts.

At low counts the asymmetric Poisson error distributions – peaking at zero and with exponential tails – are not adequately approximated by Gaussians. When low counts occur, $\chi^2$ minimisation gives significantly degraded results compared with correct treatment of the Poisson error distributions.

## 10.8 Estimating Noise Parameters

The maximum likelihood approach also equips you to tackle the problem of estimating unknown error bars. When the error bars $\sigma_i$ are unknown, you may simply consider letting the error bars to be functions $\sigma_i(\alpha)$ of the model parameters $\alpha$, adding new parameters if necessary. A maximum likelihood fit then minimizes

$$-2\ln L(\alpha) = \chi^2(\alpha) + 2\ln Z_D(\alpha) = \sum_{i=1}^{N}\left(\frac{x_i - \mu_i(\alpha)}{\sigma_i(\alpha)}\right)^2 + 2\sum_{i=1}^{N}\ln\sigma_i(\alpha) + \frac{N}{2}\ln(2\pi)\ . \tag{220}$$

The constant term is unimportant and may omitted. We saw earlier that $\chi^2$ fitting fails in attempting to estimate unknown error bars. This failure is now cured by the additional term $-2\ln Z_D$ that emerges from the maximum likelihood analysis. The new term serves as a gentle penalty that prevents $\sigma_i \to \infty$. The fit is thus a compromise between minimizing $\chi^2$ and keeping $\sigma_i$ reasonably small. To see how this works in practice, let's consider a couple of specific examples.

### 10.8.1 equal but unknown error bars

You are trying to estimate a quantity $\mu$. You have $N$ unbiased measurements $X_i$, but you don't know how accurate these measurements are. Maybe your best guess is that the error bars are all the same, $\sigma_i = \sigma$. You can then adjust the parameters $\mu$ and $\sigma$ to minimize

$$-2\ln L(\mu,\sigma) = \sum_{i=1}^{N}\left(\frac{X_i - \mu}{\sigma}\right)^2 + 2N\ln\sigma + \frac{N}{2}\ln(2\pi)\ . \tag{221}$$

As $\sigma$ increases, the $\chi^2$ term decreases to 0, but the $2N\ln\sigma$ term increases to $\infty$. Thus there is a well defined minimum at $s = \hat{s}$.

To find this solution, set to zero the derivatives with respect to $\mu$

$$0 = \frac{\partial \ln L}{\partial \mu} = \frac{2}{\sigma^2}\sum_{i=1}^{N}(X_i - \mu)\ , \tag{222}$$

and with respect to $\sigma$

$$0 = \frac{\partial \ln L}{\partial \sigma} = -\frac{2}{\sigma^3}\sum_{i=1}^{N}(X_i - \mu)^2 + \frac{N}{\sigma}\ . \tag{223}$$

The sensible result that then arises is

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} X_i\ , \qquad \hat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N}(X_i - \hat{\mu})^2\ . \tag{224}$$

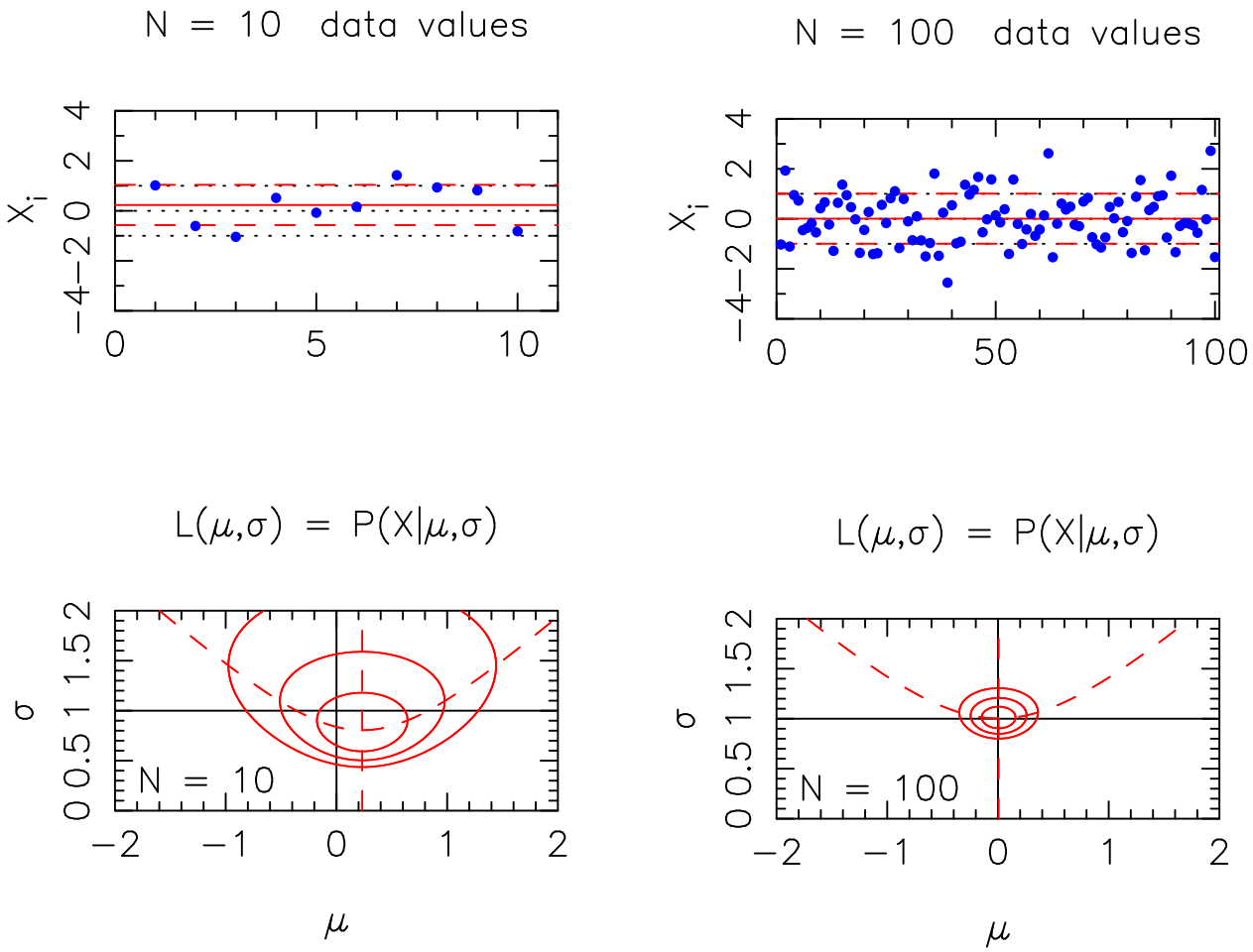


Figure 17: The likelihood function $L(\mu,\sigma)$ for estimation of the mean $\mu$ and standard deviation $\sigma$ for a sample of $N = 10$ and $N = 100$ data points. Likelihood contours $\Delta\,(-2\ln L) = 2.3$, 6.17, and 11.8 correspond to 1, 2, and 3-$\sigma$ 2-parameterconfidence regions.

Note that $\hat{\mu}$ is unbiased but $\hat{\sigma}^2$ is biased:

$$\langle\,\hat{\mu}\,\rangle = \langle\,X\,\rangle\;, \qquad \langle\,\hat{\sigma}^2\,\rangle = \frac{N-1}{N}\mathrm{Var}[\,X\,]. \tag{225}$$

The reason is that $\hat{\mu}$ "chases" the data points, reducing the scatter in the residuals by 1 degree of freedom when $\hat{\mu}$ is used to estimate the unknown true mean.

Fig. 17 shows the likelihood function $L(\mu,\sigma) = \mathrm{p}(\,X\,|\,\mu,\sigma\,)$ for two cases, with $N = 10$ and $N = 100$ data points. The 1, 2, and 3-$\sigma$ confidence regions are shown by the likelihood contours $\Delta\,(-2\ln L) = 2.3$, 6.17, and 11.8, respectively. Comparing the results for $N = 10$ and $N = 100$ we see that the confidence region shrinks by a factor of order $\sqrt{10}$, as expected. The quadratic dependence of $\hat{\sigma}^2$ on $\mu$ gives rise to an asymmetric likelihood function, with a skew toward higher $\sigma$ that is more pronounced for small $N$. Since $\hat{\mu}$ sits at the bottom of this parabola, there is a bias toward low values of $\sigma^2$ when $\hat{\mu}$ is used in the formula for $\hat{\sigma}$.

### 10.8.2 scaling error bars

You have error bar estimates $s_i$, but they need to be scaled by some factor $f$ to give larger or smaller errors $\sigma_i = f s_i$. The likelihood function is

$$L(\mu, f) = \frac{\exp\left\{ -\frac{1}{2f^2} \sum_{i=1}^{N} \left( \frac{X_i - \mu}{s_i} \right)^2 \right\}}{f^N \, (2\pi)^{N/2} \prod_{i=1}^{N} s_i} \; . \tag{226}$$

Let's adjust $\mu$ and $f$ to minimize the "badness of fit"

$$B(\mu, f) \equiv -2 \ln L(\mu, f) = \frac{1}{f^2} \sum_{i=1}^{N} \left( \frac{X_i - \mu}{s_i} \right)^2 + 2N \ln f + N \ln(2\pi) + 2 \sum_{i=1}^{N} \ln s_i \; . \tag{227}$$

As before, the $\chi^2$ term decreases to 0 as $f^{-2}$, while the $2N \ln f$ term increases to $\infty$, defining a minimum at $f = \hat{f}$.

To find this solution, evaluate the derivatives with respect to $\mu$ and $f$,

$$\frac{\partial B}{\partial \mu} = \frac{2}{f^2} \sum_{i=1}^{N} \frac{(X_i - \mu)}{s_i^2} \; , \qquad \frac{\partial B}{\partial f} = -\frac{2}{f^3} \sum_{i=1}^{N} \left( \frac{X_i - \mu}{s_i} \right)^2 + \frac{2\,N}{f} \; , \tag{228}$$

and set these to zero, thus minimising the "badness of fit", leading to

$$\hat{\mu} = \frac{\sum_{i=1}^{N} X_i / {s_i}^2}{\sum_{i=1}^{N} 1 / {s_i}^2} \; , \qquad \hat{f}^2 = \frac{1}{N} \sum_{i=1}^{N} \left( \frac{X_i - \hat{\mu}}{s_i} \right)^2 \; . \tag{229}$$

From the second derivatives,

$$\frac{\partial^2 B}{\partial \mu^2} = -\frac{2}{f^2} \sum_{i=1}^{N} \frac{1}{{s_i}^2} \; , \qquad \frac{\partial^2 B}{\partial f^2} = \frac{6}{f^3} \sum_{i=1}^{N} \left( \frac{X_i - \mu}{s_i} \right)^2 - \frac{2\,N}{f^2} \; , \tag{230}$$

the variances are

$$\mathrm{Var}[\,\hat{\mu}\,] = \left. \frac{2}{\partial^2 B / \partial \mu^2} \right|_{\hat{\mu}, \hat{f}} = \frac{\hat{f}^2}{\sum_{i=1}^{N} \frac{1}{{s_i}^2}} = \frac{\sum_{i=1}^{N} \left( \frac{X_i - \hat{\mu}}{s_i} \right)^2}{N \sum_{i=1}^{N} \frac{1}{{s_i}^2}} \; , \tag{231}$$

and

$$\mathrm{Var}\Big[\,\hat{f}\,\Big] = \left. \frac{2}{\partial^2 B / \partial f^2} \right|_{\hat{\mu}, \hat{f}} = \left[ \frac{3}{\hat{f}^3} \sum_{i=1}^{N} \left( \frac{X_i - \mu}{s_i} \right)^2 - \frac{N}{\hat{f}^2} \right]^{-1} = \frac{\hat{f}^2}{2\,N} \; . \tag{232}$$

## 10.9 additive systematic error

Suppose we have good estimates $\sigma_i$ for the measurement errors, but the data appear to have a larger scatter. We've just considered how to scale the error bars, in case they are under-estimated by some factor $f$. But suppose instead that we suspect the data to be affected by some independent additive noise component. We will assume that this extra noise has a gaussian distribution, and use the scatter in the data values to estimate the variance $\sigma_0^2$. Since we assume that the extra noise is independent of the measurement errors, the effective error bar on each data point is found by adding $\sigma_i$ and $\sigma_0$ in quadrature. Thus our model is a function of two parameters, $\mu$ and $\sigma_0^2$, and we find these by minimising the "Badness of Fit",

$$B(\mu, \sigma_0^2) \equiv -2 \ln L(\mu, \sigma_0^2) = \sum_{i=1}^{N} \frac{(X_i - \mu)^2}{{s_i}^2} + \sum_{i=1}^{N} \ln\left( {s_i}^2 \right) + \mathrm{const} \; . \tag{233}$$

where the augmented variances are

$$\mathrm{Var}[\,X_i\,] = {s_i}^2 \equiv {\sigma_0}^2 + {\sigma_i}^2 \ . \tag{234}$$

Derivatives of $B$ with respect to $\mu$ are

$$\frac{\partial B}{\partial \mu} = -2\sum_{i=1}^{N} \frac{X_i - \mu}{{s_i}^2} \ , \qquad \frac{\partial^2 B}{\partial \mu^2} = 2\sum_{i=1}^{N} \frac{1}{{s_i}^2} \ . \tag{235}$$

This leads to the usual optimal average,

$$\hat\mu = \frac{\displaystyle\sum_{i=1}^{N} X_i/\hat{s}_i^2}{\displaystyle\sum_{i=1}^{N} 1/\hat{s}_i^2} \ , \qquad \mathrm{Var}[\,\hat\mu\,] = \left[\sum_{i=1}^{N} 1/\hat{s}_i^2\right]^{-1} \ , \tag{236}$$

with

$$\hat{s}_i^2 = \hat\sigma_0^2 + {\sigma_i}^2 \ . \tag{237}$$

We still need an estimate $\hat\sigma_0^2$ for the variance $\sigma_0^2$ of the additional noise. For this, the relevant derivatives of $B$ are

$$\frac{\partial B}{\partial \sigma_0^2} = -\sum_{i=1}^{N} \frac{(X_i-\mu)^2}{{s_i}^4} + \sum_{i=1}^{N} \frac{1}{{s_i}^2} \ , \qquad \frac{\partial^2 B}{\partial (\sigma_0^2)^2} = 2\sum_{i=1}^{N} \frac{(X_i-\mu)^2}{{s_i}^6} - \sum_{i=1}^{N} \frac{1}{{s_i}^4} \ . \tag{238}$$

The maximum likelihood estimate $\hat\sigma_0^2$ is the solution of $\partial B/\partial {\sigma_0}^2 = 0$, or

$$\sum_{i=1}^{N} \frac{(X_i-\hat\mu)^2}{\hat{s}_i^4} = \sum_{i=1}^{N} \frac{1}{\hat{s}_i^2} \ . \tag{239}$$

This would appear to have no closed-form solution. To progress, define the "goodness" of data point $i$ as

$$g_i \equiv \left(\frac{\hat\sigma_0}{\hat{s}_i}\right)^2 = \frac{1}{1+(\sigma_i/\hat\sigma_0)^2} \ . \tag{240}$$

Note that $g_i \to 1$ for $\sigma_i << \hat\sigma_0$ and $g_i \to 0$ for $\sigma_i >> \hat\sigma_0$, so that $g_i$ measures in some sense of how "good" data point $i$ is for estimating $\hat\sigma_0$. In terms of $g_i$, since $\hat{s}_i^2 = {\sigma_0}^2/g_i$, the required equation becomes

$$\frac{1}{\hat\sigma_0^4}\sum_{i=1}^{N} g_i^2\,(X_i-\hat\mu)^2 = \frac{2}{\hat\sigma_0^2}\sum_{i=1}^{N} g_i \ . \tag{241}$$

A formal solution is thus

$$\hat\sigma_0^2 = \frac{\displaystyle\sum_{i=1}^{N} g_i^2\,(X_i-\hat\mu)^2}{\displaystyle\sum_{i=1}^{N} g_i} \ . \tag{242}$$

With $\hat\mu$ and $g_i$ depending on $\hat\sigma_0$, the solution must be found by iterating.

### 10.9.1 CCD readout noise and gain

You can tackle many other cases in a similar manner. For example, the noise that affects CCD data is usually modelled as two independent components, an additive readout noise from the electronic readout amplifier, and Poisson noise increasing as $\sqrt{\mu_i}$ where $\mu_i$ is the expected photon detection rate in pixel $i$ The CCD noise model,

$$\sigma_i^2 = \sigma_{\rm R}^2 + \frac{\mu_i}{G} \ , \tag{243}$$

has 2 parameters: the readout noise dispersion $\sigma_{\rm R}$, and the CCD gain parameter $G$. You can measure these by maximum likelihood fitting, provided you have data values with a range of exposure levels $\mu_i$. We discuss later this in more detail.

# 11 Error Bar Estimates

In this section we consider various methods to estimate the error bars of parameters estimated from data. A measurement is of no use unless we understand its uncertainties. Important not only are the values of our parameters, but equally their uncertainties.

## 11.1 Sample Variance

Consider first the simplest case, averaging of independent measurements. If you have error bars $\sigma_i$ on individual data points $x_i$, then you know that an inverse-variance weighted average of the data is optimal, and your analysis using fuzzy algebra provides the formula for the uncertainty in the optimal average,

$$\hat{x} \equiv \frac{\sum_{i=1}^{N} x_i/\sigma_i^2}{\sum_{i=1}^{N} 1/\sigma_i^2} \ , \qquad \mathrm{Var}[\,\hat{x}\,] = \frac{1}{\sum_{i=1}^{N} 1/\sigma_i^2} \ . \tag{244}$$

What can you do if your data points come with no individual error bar estimates? Assume you have $N$ independent measurements $x_i$ for $i = 1...N$, but no estimates of the error bars on these measurements. Suppose that each data point arises from the same equipment, so that it is reasonable to assume that they all have the same expected value $\mu$ and variance $\sigma^2$:

$$\langle\, x_i \,\rangle = \mu \ , \qquad \mathrm{Var}[\,x_i\,] = \sigma^2 \ . \tag{245}$$

Your job is to estimate the parameters $\mu$ and $\sigma^2$, and understand how uncertain their values are.

Estimating $\mu$ is easy. With no error bar estimates for individual data points, optimal averaging is not possible. But given the assumption is that all data points have the same error bar, the optimal average would be equally-weighted in any case.

$$\hat{\mu} = \bar{x} \equiv \sum_{i=1}^{N} x_i \ . \tag{246}$$

This is unbiased, with mean and variance

$$\langle\, \bar{x} \,\rangle = \mu \ , \qquad \mathrm{Var}[\,\bar{x}\,] = \sigma^2/N \ . \tag{247}$$

However, as we don't yet have an estimate for $\sigma^2$, we turn now to that.

If you knew $\mu$, then an unbiased estimate of $\sigma^2$ is the sample variance:

$$S^2 \equiv \sum_{i=1}^{N} (x_i - \mu)^2 \ . \tag{248}$$

To see this: note that $\eta_i = (x_i - \mu)/\sigma$ is a standard Gaussian, with mean 0 and variance 1. $\eta_i^2$ is therefore a $\chi^2$ with 1 degree of freedom, which has mean 1 and variance 2. Add up $N$ of these, one for each data point, and you get a $\chi^2$ with $N$ degrees of freedom, with mean $N$ and variance $2N$. This means that $S^2/sigma^2$ is a reduced $\chi^2$ with $N$ degrees of freedom, with mean 1 and variance $2/N$. This is exactly true for Gaussian noise, approximately so for non-Gaussian noise if $N$ is large (central limit theorem).

$S^2$ is therefore an unbiased estimator for $\sigma^2$:

$$\left\langle\, S^2 \,\right\rangle = \sigma^2 \ , \qquad \mathrm{Var}\left[\, S^2 \,\right] = \frac{2\sigma^4}{N} \ . \tag{249}$$

The above analysis assumes we know $\mu$. In fact, we don't. We have to estimate $\mu$ by $\hat{\mu} = \bar{x}$. We can still define the sample variance, using $\bar{x}$ rather than $\mu$:

$$S^2 \equiv \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{x})^2 \, . \tag{250}$$

Notice here that we divide by $N-1$ rather than $N$. This is because $\bar{x}$ chases the data points as they jitter around, and this eliminates 1 degree of freedom, leaving $N-1$ rather than $N$ degrees of freedom in the residuals. For example, if there is only 1 data point, then $\bar{x} = x1$, and the sum vanishes. Dividing by $N-1$, rather than $N$, keeps $S^2$ unbiased, and $S^2/\sigma^2$ is now a reduced $\chi^2$ with $N-1$ degrees of freedom, with mean 1 and variance $2/(N-1)$:

$$\left\langle\, S^2 \,\right\rangle = \sigma^2 \ . \qquad\qquad \mathrm{Var}\left[\, S^2 \,\right] = \tfrac{2\sigma^4}{N-1} \tag{251}$$

If for some reason you want to estimate $\sigma$, rather than $\sigma^2$, it is simple to take the square root $S \equiv \sqrt{S^2}$ . Because of the non-linear transformation, however, this is a biased estimate, $\left\langle \sqrt{S^2} \right\rangle \neq \sigma$. But the bias is small for large $N$, so $\left\langle\, S \,\right\rangle \approx \sigma$ . For large $N$, the fractional uncertainty in $S$ is 1/2 that of $S^2$:

$$\frac{\sigma(S)}{\left\langle\, S \,\right\rangle} \approx \frac{1}{2}\frac{\sigma(S^2)}{\left\langle\, S^2 \,\right\rangle} = \frac{\sqrt{2\sigma^4/(N-1)}}{2\sigma^2} = \frac{1}{\sqrt{2(N-1)}} \ . \tag{252}$$

So we have

$$\left\langle \sqrt{S^2} \right\rangle \approx \sigma \qquad\qquad \sigma\left(\sqrt{S^2}\right) \approx \frac{\sigma}{\sqrt{2(N-1)}} \ . \tag{253}$$

## 11.2 $\Delta\chi^2$ confidence intervals

Comparison of results from fuzzy algebra indicates that a $\Delta\chi^2 = 1$ criterion gives single-parameter 1-$\sigma$ error bars.

## 11.3 Bayesian parameter probability maps

Bayesian methods deliver not only optimal parameter values, but also their complete joint posterior probability distribution. The joint probability map encapsulates precisely our knowledge of the parameters, incorporating both our prior knowledge and the new information gleaned from the data.

When the parameters are well defined, the probability map has a single isolated probability peak. The location of the peak defines the optimal values, the width the uncertainties, and the shape the correlations among the model parameters.

### 11.3.1 1-parameter confidence intervals

Project the probability peak onto a particular parameter axis to obtain a probability map for that single parameter. In doing this you effectively integrate over the probability distribution of all other parameters. Those parameters are your **nuisance parameters**. You are not particularly interested in them, but you have to include them in your model in order to get a good fit to the data.

The posterior probability for the parameter $x$ is then

$$\mathrm{p}(\, x|D \,) = \frac{\exp\left\{ -\left(Q^2(x) - Q^2_{\mathrm{min}}\right)/2 \right\}}{Z_Q} \ , \tag{254}$$

where for each value of $x$ we have integrated the joint probability over all nuisance parameters.

The optimal value $\hat{x}$ at the probability peak minimizes the function $Q^2(x) - 2\ln Z_Q$. The second term $-2\ln Z_Q$ needs to be included if some of the data points have unknown error bars that depend on $x$, otherwise it can be omitted.

If the predicted data are linear functions of $x$, then $Q^2(x)$ will be a parabola. If the predicted data values are smooth functions of $x$, $Q^2(x)$ will also be smooth, and a parabola will be a good approximation in some region near the minimum. The quadratic approximation is

$$Q^2(x) \approx Q^2_{\mathrm{min}} + \left(\frac{x-\hat{x}}{\sigma(\hat{x})}\right)^2 \ . \tag{255}$$

The parabolic approximation to $Q^2(x)$ corresponds to a Gaussian approximation for the probability peak in $\mathrm{p}(\, x|D \,)$.

The uncertainty in $x$ is defined by the width of the peak in $\mathrm{p}(\, x|D \,)$, and hence by the curvature of $Q^2(x)$ near its minimum.

$$\mathrm{Var}[\, \hat{x} \,] = \left| \frac{2}{\partial^2 Q^2/\partial x^2} \right| \ . \tag{256}$$

The range $\hat{x} \pm \sigma(\hat{x})$ encloses 67% of the probability in the Gaussian approximation. This range is also defined by the criterion

$$\Delta Q^2(x) \equiv Q^2 - Q^2_{\rm min} < 1 \ . \tag{257}$$

### 11.3.2 2-parameter confidence regions

An elongation of the joint probability peak in the zone between two parameter axes signals that those parameters are correlated due to an ambiguity in the data. We may be unable to determine their individual values with much precision, but a combination of their values will be more tightly defined.

banana diagram – example: temp and area of blackbody spectrum.
elliptical $Q^2$ contours near peak.
higher $\Delta Q^2$ needed for 2-parameter region.
2-parameter region wider than 1-parameter intervals.

### 11.3.3 $M$-parameter confidence regions

$Q^2$ ellipsoid.
Hessian matrix $\partial^2 Q^2/\partial x \partial y$.
relation to covariance matrix.
eigenvectors as principal axes of ellipsoid.
eigenvalues.

### 11.3.4 influence of prior information

## 11.4 Monte-Carlo Error Bars

Jiggle the data points, using Gaussian random numbers, to form a sequence of "fake" datasets. Re-fit the model to each fake dataset. The result is a set of fitted models with parameters that jiggle around in response to the jiggling data values. You can compute mean or median values of any single parameter, and sample variances to define the uncertainty in that parameter. You can also examine joint 2-d distributions of any 2 parameters, and consider higher-dimensional confidence regions as required.

Monte-Carlo methods neatly define the joint probability distribution of the parameters. The Monte-Carlo methods assume Gaussian errors.

## 11.5 Bootstrap Error Bars

What can you do if you don't know the error properties well enough to assume Gaussian errors? In the Bootstrap method, you again form a sequence of "fake" datasets, but this time you don't perturb any of the data values, but rather you decide at random which points to keep and which ones to omit. Bootstrap method see how the parameters jiggle in response to changes in the weights assigned to each data point.

In fact, you select $N$ points at random, with replacement. Thus some points will be omitted altogether, most points will be selected once, some will be selected twice, a few three times, and so forth.

Bootstrap methods work well only when there are a large number of degrees of freedom – more data points than parameters – and no parameters determined primarily by a tiny number of data points. For example, if you were fitting a line to 2 data points, the bootstrap would clearly fail whenever one of the data points was omitted, because then the fit would be degenerate.

# 12 Bayesian Methods

## 12.1 Bayes Theorem

The easiest way to remember Bayes Theroem is to derive it from the relationship between a joint probability map $\mathrm{p}(\,X,Y\,)$ and the two conditional probability maps $\mathrm{p}(\,X|Y\,)$ and $\mathrm{p}(\,Y|X\,)$. The conditional probability map $\mathrm{p}(\,X|Y\,)$ is the probability map for $X$ when $Y$ is held fixed at a specific value. This is proportional to the joint probability $\mathrm{p}(\,X,Y\,)$, but with a different normalization. The joint probability is normalized over the joint domain

$$\int \mathrm{p}(\,X,Y\,)\;dX\;dY = 1\;, \tag{258}$$

and the contitional probability is normalized over the more restricted domain that remains when the conditioned variable is fixed

$$\int \mathrm{p}(\,X|Y\,)\;dX = 1\;. \tag{259}$$

You can therefore write the joint probability as

$$\mathrm{p}(\,X,Y\,) = \mathrm{p}(\,X|Y\,)\;\mathrm{p}(\,Y\,)\;, \tag{260}$$

where

$$\mathrm{p}(\,Y\,) = \int \mathrm{p}(\,X,Y\,)\;dX\;. \tag{261}$$

You can also hold fixed $X$ rather than $Y$, leading to

$$\mathrm{p}(\,X,Y\,) = \mathrm{p}(\,Y|X\,)\;\mathrm{p}(\,X\,)\;. \tag{262}$$

Equating the the two expressions for $\mathrm{p}(\,X,Y\,)$ leads to **Bayes theorem**

$$\mathrm{p}(\,Y|X\,)\;\mathrm{p}(\,X\,) = \mathrm{p}(\,X|Y\,)\;\mathrm{p}(\,Y\,)\;. \tag{263}$$

## 12.2 Bayesian Data Analysis

This seemingly trivial statement about probabilities becomes profound when applied to data analysis. If we take $X =$ Data $D$ and $Y =$ Model $\mu$, then Bayes theorem is

$$\mathrm{p}(\,\mu|D\,) = \frac{\mathrm{p}(\,D|\mu\,)\;\mathrm{p}(\,\mu\,)}{\mathrm{p}(\,D\,)}\;. \tag{264}$$

Now, interpret this as follows. The **inference** that we make about the world is $\mathrm{p}(\,\mu|D\,)$. This is the **posterior probability** that we assign to the model $\mu$ after we have obtained the data $D$. This inference is proportional to the now familiar **likelihood** $L(\mu) = \mathrm{p}(\,D|\mu\,)$, which is the probability that the data $D$ will arise if the model $\mu$ is correct.

But the inference is also proportional to the **prior** probability $\mathrm{p}(\,\mu\,)$. We appear to be free to assign this prior in any way we wish. This lets us express a prior knowledge or a prejudice about the relative likelihood of various models or of different parameter values of the chosen model $\mu$.

Notice that the inference we make about the world from a given dataset is **not unique**. The inference depends not only on the data, but also on our prior. Two people who use different priors will reach different conclusions from their correct analyses of the same data. This interesting and somewhat subtle aspect of data analysis and the way we acquire information about the world is not always appreciated. In particular, maximum likelihood fitting does not allow for different priors, or rather, it assigns a specific prior with equal probability given to each model.

Finally, the factor $\mathrm{p}(\,D\,)$ in the denominator is just a normalization factor

$$\mathrm{p}(\,D\,) = \int \mathrm{p}(\,D|\mu\,)\;\mathrm{p}(\,\mu\,)\;d\mu \tag{265}$$

to ensure that our inference has probability 1 when summed over all possible models.

## 12.3 Blending Data with Prior Knowledge

The Bayesian formalism allows for expressions of prejudice, preconception, and prior knowledge in the interpretation of data. It makes sense that this should be a component of any faithful representation of our relationship with data as we struggle for understanding.

Let's look at the influence of prior information by considering a specific example. Suppose you are interested in some quantity $x$. Let's say that, based on your prior experience, you suspect that $x$ may be near $d$. Perhaps you can be even more specific, placing $x$ in the region of $d \pm s$. If you want to make use of this prior knowledge, you could adopt a Gaussian prior

$$\mathrm{p}(\,x\,) = \frac{\exp\left\{\,-\frac{1}{2}\left(\frac{x-d}{s}\right)^2\,\right\}}{\left(2\pi s^2\right)^{1/2}} \tag{266}$$

with expected value $d$ and standard deviation $s$.

Now, as a scientist, your prejudices are not enough to keep you content. They need to be tested. So, out you go to do a series of experiments, from which you obtain $N$ measurements of $x$, whose values are $X_i \pm \sigma_i$. By obtaining the new data, your knowledge of $x$ may or may not change significantly from your preconception. It will depend on how good the data are, compared with the strength of your pre-conception.

Your inference about $x$ after considering the data $D$ is

$$\mathrm{p}(\,x|D\,) = \frac{\exp\left\{\,-Q^2/2\,\right\}}{\displaystyle\int \exp\left\{\,-Q^2/2\,\right\}dx} = \frac{\exp\left\{\,-\Delta Q^2/2\,\right\}}{Z_Q}, \tag{267}$$

where

$$Q^2 = \chi^2 + \left(\frac{x-d}{s}\right)^2 \ , \tag{268}$$

$$\Delta Q^2 = Q^2 - Q^2_{\rm min} \ , \tag{269}$$

and

$$Z_Q = \left(2\pi s^2\right)^{1/2} Z_D = \left(2\pi s^2\right)^{1/2} \prod_{i=1}^{N} \left(2\pi\sigma_i^2\right)^{1/2} \ . \tag{270}$$

Bayesian estimation looks rather like $\chi^2$ fitting, except that we now use $Q^2$ rather than $\chi^2$ to measure the badness-of-fit. Note that the prior acts just like a data point at $d \pm s$. This makes $Q^2$ essentially the same as $\chi^2$, but now calculated over both the data $D$ and the prior data point $d \pm s$.

A quadratic approximation to $Q^2$ is

$$Q^2(x) = \chi^2_{\rm min} + \left(\frac{x - x_{\rm ML}}{\sigma_{\rm ML}}\right)^2 + \left(\frac{x-d}{s}\right)^2 \ . \tag{271}$$

The maximum likelihood estimate $x_{\rm ML} \pm \sigma_{\rm ML}$ occurs where $\chi^2(x)$ reaches its minimum value $\chi^2_{\rm min}$.

The Bayesian estimate, found by setting to 0 the derivative of $Q^2(x)$ with respect to $x$, is

$$x_{\rm B} = \left(\frac{x_{\rm ML}/\sigma^2_{\rm ML} + d/s^2}{1/\sigma^2_{\rm ML} + 1/s^2}\right) \ , \tag{272}$$

with variance

$$\mathrm{Var}[\,x_{\rm B}\,] = \left(1/\sigma^2_{\rm ML} + 1/s^2\right)^{-1} \ . \tag{273}$$

If you really had no clue beforehand what the value of $x$ might be, then you will have adopted a very wide prior, $s \to \infty$. The Bayesian fit then reverts to a maximum likelihood fit. In this case your preconception has had virtually no effect, and you accept completely the result delivered by the new data. A completely open mind accepts all data.

On the other hand, if your prejudice is rather stronger, you will have adopted a narrower prior, $s << \sigma_{\rm ML}$. The data then have only a little influence, moving your prior value a little bit toward the value favored by the data. If you are armed (or lumbered) with very strong preconceptions, it requires correspondingly accurate data to over-ride the prejudice and bend your prior conclusion.

If you are absolutely certain that $x = d$, then your prior is a Dirac delta function. You opinion about $x$ cannot to respond to the data. If your prejudice is sufficiently strong, you will be unmoved by any data.

## 12.4 Model vs Model

The evaluation of relative probabilities for 2 models with different parameter spaces quite subtle. For the relative probability of 2 models, use a Bayesian analysis. First, we re-derive Bayes theorem. The joint probability of two events $A$ and $D$ is

$$\mathrm{p}(\,A,D\,) = \mathrm{p}(\,D|A\,)\;\mathrm{p}(\,A\,) = \mathrm{p}(\,A|D\,)\;\mathrm{p}(\,D\,)\;, \tag{274}$$

and thus Bayes theorem is

$$\mathrm{p}(\,A|D\,) = \frac{\mathrm{p}(\,D|A\,)\;\mathrm{p}(\,A\,)}{\mathrm{p}(\,D\,)}\;. \tag{275}$$

Similarly, for events $B$ and $D$:

$$\mathrm{p}(\,B|D\,) = \frac{\mathrm{p}(\,D|B\,)\;\mathrm{p}(\,B\,)}{\mathrm{p}(\,D\,)}\;. \tag{276}$$

If $D$ is true, then the relative probability of $A$ to that of $B$ is

$$\frac{\mathrm{p}(\,A|D\,)}{\mathrm{p}(\,B|D\,)} = \frac{\mathrm{p}(\,A\,)}{\mathrm{p}(\,B\,)}\frac{\mathrm{p}(\,D|A\,)}{\mathrm{p}(\,D|B\,)}\;. \tag{277}$$

In our data analysis problem, the symbols are interpreted as $A =$ "Model A is true", $B =$ "Model B is true", and $D =$ "The dataset $D$ was observed".

The prior probabilities of the two models are $\mathrm{p}(\,A\,)$ and $\mathrm{p}(\,B\,)$, and the first factor on the right-hand side is the ratio of these prior probabilities. The two models with equal prior probabilities, or one may have a higher prior probability than the other. Thus there is no unique answer to the question about the relative probability of the two models. Different people can correctly analyse the same dataset, yet reach radically different conclusions if they subscribe to radically different priors for the two models.

The second factor is the likelihood ratio of the two models. This expresses the effect of the data, which is to shift the prior probability ratio of the two models by the ratio of the likelihood that these models will produce the dataset that was observed. The two models may have equivalent or different parameter spaces. Let Model A have $N_A$ parameters $\alpha$, and Model B have $N_B$ parameters $\beta$. Since we are interested in the models, rather than specific values of their parameters, the Bayesian analysis considers all possible values of these parameters, weighted by the probabilities that those particular parameter values are correct. The likelihood ratio is thus

$$\frac{\mathrm{p}(\,D|A\,)}{\mathrm{p}(\,D|B\,)} = \frac{\displaystyle\int \mathrm{p}(\,D|A,a\,)\;\mathrm{p}(\,a|A\,)\;da}{\displaystyle\int \mathrm{p}(\,D|B,b\,)\;\mathrm{p}(\,b|B\,)\;db}\;. \tag{278}$$

Note here the integrations over parameter space, weighted by both the prior and the likelihood of the parameters. This integration is referred to as *marginalising over the nuisance parameters of the problem.*

For specific parameter values $a$ in Model A, assume that the $N$ data values $X_i$ have expected values $\mu_i(A,a)$ and standard deviations $\sigma_i(A,a)$. The likelihood (assuming Gaussian statistics) is then

$$L_A\,(a) \equiv \mathrm{p}(\,D|A,a\,) = \frac{\exp\Big\{\,-\frac{1}{2}\chi^2(D,A,a)\,\Big\}}{Z_A\,(a)}\;, \tag{279}$$

where

$$\chi^2\,(D,A,a) = \sum_{i=1}^{N}\left(\frac{X_i - \mu_i\,(A,a)}{\sigma_i\,(A,a)}\right)^2 \tag{280}$$

and the *partition function* for Model A is

$$Z_A\,(a) \equiv \int \exp\Big\{\,-\tfrac{1}{2}\chi^2(D,A,a)\,\Big\} d^N D = (2\pi)^{N/2}\prod_{i=1}^{N}\sigma_i\,(A,a)\;. \tag{281}$$

We can of course also write analogous expressions for Model B. The likelihood ratio is

$$\frac{L_A\,(a)}{L_B\,(b)} \equiv \frac{\mathrm{p}(\,D|A,a\,)}{\mathrm{p}(\,D|B,b\,)} = \frac{Z_B\,(b)}{Z_A\,(a)}\exp\Big\{\,-\tfrac{1}{2}\left[\chi^2(D,A,a) - \chi^2(D,B,b)\right]\,\Big\} \tag{282}$$

The quantity analogous to $\Delta\chi^2$ is

$$-2\ln\left(\frac{L_A(a)}{L_B(b)}\right) = \chi^2\left(D,A,a\right) - \chi^2\left(D,B,b\right) + 2\sum_{i=1}^{N}\ln\left(\frac{\sigma_i\left(A,a\right)}{\sigma_i\left(B,b\right)}\right) . \tag{283}$$

The likelihood ratio depends on the difference in $\chi^2$ between the two models, and also on the relative sizes of the error bars for the two models. The $\Delta\chi^2$ rewards models and parameters that give a good fit to the data. The partition function ratio (the Occam factor) punishes models and parameters that place large errors on the data points.

Finally, let's tie the above results together. The relative probabilities of the two models, marginalising over the parameters of each model, and including the possibility of different parameter spaces, is

$$\frac{\mathrm{p}(\,A|D\,)}{\mathrm{p}(\,B|D\,)} = \frac{\mathrm{p}(\,A\,)}{\mathrm{p}(\,B\,)}\frac{\mathrm{p}(\,D|A\,)}{\mathrm{p}(\,D|B\,)} = \frac{\mathrm{p}(\,A\,)}{\mathrm{p}(\,B\,)}\frac{\displaystyle\int \mathrm{p}(\,D|A,a\,)\;\mathrm{p}(\,a|A\,)\;da}{\displaystyle\int \mathrm{p}(\,D|B,b\,)\;\mathrm{p}(\,b|B\,)\;db} . \tag{284}$$

The parameter space integrals can be evaluated numerically in any specific cases. Analytic integration is possible if we can approximate the integrand by a product of Gaussian functions. While the analysis below is not rigorous, it captures and illustrates the main result. Let's assume that $\mathrm{p}(\,D|A,a\,)$ resembles a product of $N_a$ Gaussian functions, centred at $a_k = \hat{a}_k$ with standard deviations $\sigma_k$, for each of the $N_a$ parameters of Model A:

$$\mathrm{p}(\,D|A,a\,) = \frac{\exp\left\{\,-\frac{1}{2}\left[\,\chi^2\left(\hat{a}\right) + \displaystyle\sum_{k=1}^{N_a}\left(\frac{a_k-\hat{a}_k}{\sigma_k}\right)^2\right]\right\}}{(2\pi)^{N/2}\displaystyle\prod_{i=1}^{N}\sigma_i} . \tag{285}$$

This will be the case when the parameters of Model A are independent. If they are not, then you can rotate the parameter axes, thus re-parameterising the model, so that the new parameters are independent.

Similarly, let the priors be approximated by Gaussian functions of $a_k$ centred at $\bar{a}_k$ of standard deviation $\Delta_k$:

$$\mathrm{p}(\,D|A,a\,) = \frac{e^{-\chi^2(a)/2}}{\displaystyle\prod_{i=1}^{N}\left(2\pi\sigma_i^2\right)^{1/2}} = \frac{\exp\left\{\,-\frac{1}{2}\displaystyle\sum_{k=1}^{N_a}\left(\frac{a_k-\bar{a}_k}{\Delta_k}\right)^2\right\}}{(2\pi)^{N_a/2}\displaystyle\prod_{k=1}^{N_a}\Delta_k} . \tag{286}$$

Combining those gives

$$\mathrm{p}(\,D|A,a\,)\;\mathrm{p}(\,a|A\,) = \frac{\exp\left\{\,-\frac{1}{2}\left[\,\chi^2\left(\hat{a}\right) + \displaystyle\sum_{k=1}^{N_a}\left(\frac{a_k-\hat{a}_k}{\sigma_k}\right)^2 + \sum_{k=1}^{N_a}\left(\frac{a_k-\bar{a}_k}{\Delta_k}\right)^2\right]\right\}}{(2\pi)^{(N+N_a)/2}\displaystyle\prod_{i=1}^{N}\sigma_i\prod_{k=1}^{N_a}\Delta_k} . \tag{287}$$

The numerator is a product of Gaussian functions for each parameter. Complete the squares for each parameter to find a product of Gaussian functions peaking at $\tilde{a}_k$ with standard deviations $S_k$, where

$$\tilde{a}_k = S_k^2\left(\frac{\hat{a}_k}{\sigma_k^2} + \frac{\bar{a}_k}{\Delta_k^2}\right) , \tag{288}$$

and

$$\frac{1}{S_k^2} = \frac{1}{\sigma_k^2} + \frac{1}{\Delta_k^2} . \tag{289}$$

Thus the peak probability for parameter $a_k$ occurs at the optimal average of the maximum likelihood estimate $\hat{a}_k$ and the prior $\bar{a}_k$. For good parameters well determined by the data, $\tilde{a} \to \hat{a}$ and $S_k \to \sigma_k$. For bad parameters poorly determined by the data, $\tilde{a} \to \bar{a}$ and $S_k \to \Delta_k$.

This gives

$$\mathrm{p}(\,D|A,a\,)\;\mathrm{p}(\,a|A\,) = \frac{\exp\left\{\,-\frac{1}{2}\left[\chi^2\,(\hat{a}) + \sum_{k=1}^{N_a}\left(\frac{\tilde{a}_k-\hat{a}_k}{\sigma_k}\right)^2 + \sum_{k=1}^{N_a}\left(\frac{\tilde{a}_k-\bar{a}_k}{\Delta_k}\right)^2 + \sum_{k=1}^{N_a}\left(\frac{a_k-\tilde{a}_k}{S_k}\right)^2\right]\right\}}{(2\pi)^{(N+N_a)/2}\prod_{i=1}^{N}\sigma_i\prod_{k=1}^{N_a}\Delta_k}\;. \tag{290}$$

Now, integrate over the $N_a$ dimensions of the parameter space, picking up a factor $\sqrt{2\pi}S_k$ for the integral over $a_k$, to find

$$\int \mathrm{p}(\,D|A,a\,)\;\mathrm{p}(\,a|A\,)\;da = \frac{\exp\left\{\,-\frac{1}{2}\left[\chi^2\,(\hat{a}) + \sum_{k=1}^{N_a}\left(\frac{\tilde{a}_k-\hat{a}_k}{\sigma_k}\right)^2 + \sum_{k=1}^{N_a}\left(\frac{\tilde{a}_k-\bar{a}_k}{\Delta_k}\right)^2\right]\right\}}{(2\pi)^{N/2}\prod_{i=1}^{N}\sigma_i}\prod_{k=1}^{N_a}\frac{S_k}{\Delta_k}\;. \tag{291}$$

Note that

$$\frac{S_k}{\Delta_k} = \frac{\sigma_k}{\left(\sigma_k^2+\Delta_k^2\right)^{1/2}}\;, \tag{292}$$

so that this factor is 1 for each poorly-determined "bad" parameter, with $\sigma_k \gg \Delta_k$, and $\sigma_k/\Delta_k$ for each well-determined "good" parameter.

Finally, make similar approximations for Model B. Then

$$\frac{\mathrm{p}(\,A|D\,)}{\mathrm{p}(\,B|D\,)} = \frac{\mathrm{p}(\,A\,)}{\mathrm{p}(\,B\,)}\,\frac{L_A\,(\hat{a})}{L_B\left(\hat{b}\right)}\,\frac{\prod_{k=1}^{N_a}\frac{\sigma\,(\hat{a}_k)}{\Delta\,(a_k)}}{\prod_{k=1}^{N_b}\frac{\sigma\left(\hat{b}_k\right)}{\Delta\,(b_k)}}\;. \tag{293}$$

In addition to the ratio of prior probabilities, and the ratio of likelihoods, the probability rises by a factor $f = \Delta/\sigma$ whenever the data confine a parameter to a range $\sigma$ that is a factor $f$ times smaller than its prior range $\Delta$.

## 12.5 Assessing Prior Knowledge

A license to over-ride any data with a sufficiently rigid prejudice seems unreasonable. Surely if data arrive that disagree with our prior, we should seriously consider abandoning those convictions and becoming more open minded. An overly narrow-minded viewpoint should lose credibility.

On the other end of the scale, a very wide prior has no predictive power. All things are possible. If we find fossilized dinosaur bones, they could have been set there in 4004 BC to make it look like dinosaurs were here 100 million years ago. If we can remember our childhood, the memories could have been planted 10 minutes ago. While we may enjoy wild speculations, we should not take them too seriously. If a very wide prior were reasonable, the data would sprinkle in all over the place rather than clustering in a relatively narrow range.

Not everyone held the same prior, and so a range of opinions exists after considering the data. The degree of overlap between the prior and the data should let us assess how reasonable it was to hold that prior before we received the data. We should weigh the opinions based on the success of the prior.

Thus we are lead to contemplate a community of people with a range of priors. The distribution of priors within the community is in effect a probability distribution on the parameters of the prior, $\mathrm{p}(\,d,s\,)$ in our simple example. The inference made by the community as a whole would be

$$\mathrm{p}(\,x|D\,) = \int \mathrm{p}(\,D|x,d,s\,)\;\mathrm{p}(\,x|d,s\,)\;\mathrm{p}(\,d,s\,)\;dd\;ds\;\;. \tag{294}$$

Where is this going? After the data have arrived, we should re-assess the priors. Assign probabilities to the priors in the light of the data. Downgrade highly dogmatic viewpoints that cannot be moved by any data. Downgrade also overly open viewpoints that are convinced by anything. Pay attention to priors that have a fair degree of overlap with the data, not necessarily perfect.

This is very much like what happens in a scientific community, and perhaps also like what happens as we assess data inside a human mind. We tend to ignore dogmatic voices calling attention to all the data that support a pet model while conveniently forgetting the rest. We tend also to ignore voices that are so gullible as to believe anything. The voices we listen to are those whose ability to judge seems sound because their prejudices tend to be reasonably compatible with the data and yet they are reasonably open to new results.

$$\mathrm{p}(\,d,s|D\,) = \int \mathrm{p}(\,x|D\,)\;\mathrm{p}(\,x|d,s\,)\;\mathrm{p}(\,d,s\,)\;dx \tag{295}$$

# Part II
# Astronomical Data Analysis

Here we use the techniques of optimal data analysis to develop applications for several types of astronomical data.

# 13 Photon Counting Data

Let's observe a star whose spectral energy distribution is $f_\nu(\lambda)$. Use a telescope with area $A$, and expose for a time $t$. The number of photons you expect to detect is

$$\Phi = At \int \frac{f_\nu(\lambda)\, d\nu}{h\nu} Q(\lambda) e^{-\tau(\lambda)} \tag{296}$$

The product $f_\nu d\nu$ is the energy per unit area and time. Dividing by $h\nu$ converts the energy to photons. The factor $e^{-\tau}$ is the fraction of photons that survive passage through the interstellar medium, and the Earth's atmosphere, to enter the telescope. The factor $Q(\lambda)$, the detector quantum efficiency, is the fraction of the incident photons with wavelength $\lambda$ that make it through the optics and are actually recorded by the detector.

A very handy number to remember is that a star of magnitude 0 (e.g. Vega) produces a flux of about $10^3$ photons $\mathrm{cm}^{-2}\mathrm{s}^{-1}\mathrm{\AA}^{-1}$. This will let you very quickly quickly calculate the number of photons to expect when observing a star of magnitude $m$

$$\Phi \sim 10^{3-0.4m} \left(\frac{A}{\mathrm{cm}^2}\right) \left(\frac{t}{\mathrm{s}}\right) \left(\frac{\Delta\lambda}{\mathrm{\AA}}\right) \ . \tag{297}$$

For example, a 1 metre scope ($A = 100^2\pi/4 = 7.8 \times 10^3 \mathrm{cm}^2$), observing a 20th magnitude star ($10^{-0.4m} = 10^{-8}$) for 1 minute ($t = 60s$) produces 4.7 photons per Å.

The photons actually detected will always be a whole integer – you can't detect half a photon! Of course the *expected* number of photons, $\Phi$, is generally not an integer. When you observe the star, you will sometimes count more photons, and sometimes less. The individual counts are always integers, but if you average a large number of measurements, the average will be $\Phi$.

The photon count $N$ is a **Poisson random variable**. It is said to be subject to Poisson statistics, or photon counting statistics. It's probability map is

$$\mathrm{p}(\, N\,) = \frac{\Phi^N e^{-\Phi}}{N!}. \tag{298}$$

The photon count can take on only discrete non-negative integer values, 0,1,2,... The mean and variance of $N$ are

$$\langle\, N\,\rangle = \Phi \qquad \sigma^2(N) = \Phi \tag{299}$$

You may hear that "The uncertainty in $N$ counts is $\sqrt{N}$", but remember that this would be silly if $N = 0$. The uncertainty in $N$ counts is the square root of the *expected* number of counts, $\sigma N = \sqrt{\Phi}$, where $\Phi = \langle\, N\,\rangle$.

# 14 CCD Imaging Detectors

Most astronomical imaging and spectroscopy is done with CCD detectors. A CCD, or Charge Coupled Device, is a solid state integrated circuit fabricated from a silicon chip. You find CCDs in consumer electronics, in in digital cameras and camcorders, but the ones used as astronomical detectors are rather more expensive because they must be very linear in their response to light and they must work at very faint light levels.

The CCD chip has a rectangular grid of potential wells formed by applying voltages to a grid of crossed electrodes. Exposing the CCD to light promotes electrons into the conduction band. These photo-electrons are almost immediately trapped inside the nearest potential well. The light image is thereby stored on the chip in the form of a pattern of charges in the rectangular grid of potential wells.

The CCD is read out by applying special sequences of voltages to transfer the charge from one pixel to the next. If charge transfer efficiency is good, the transfer is almost perfect. If not then some charge gets left behind, leading to a tail of charge lagging behind any sharp feature in the image – a star image for example. The charge transfer efficiency was troublesome in the early days, but is seldom noticeable with modern CCDs.

The CCD readout occurs one row at a time. The charges stowed in the final row of pixels are shifted toward the corner pixel that is connected to the readout amplifier. This is repeated to read out the entire row of pixels. Then the image is then shifted down one row to load the next row into position for readout. This sequence is repeated until all rows have been read out.

The charge from the corner pixel is presented to the input of the readout amplifier. The amplified output voltage is presented to an ADC (analogue to digital converter), which generates a digital number proportional to the voltage that is stored in the appropriate slot in a computer memory buffer. Thus the image is translated from charges on the chip to numbers in the computer. The numbers are referred to as 'data numbers' (DN) or 'analog data units' (ADU). The ADC typically generates 16-bit numbers, from 0 to $2^{16} = 65536$. When the voltage is too high, the ADC saturates and the output remains stuck at 65536.

The readout amplifier has a bias voltage $B$ and a gain $G$, determined by the amplifier's output and feedback resistors. The gain determines the number of ADUs that correspond to 1 electron, and is typically given in units of $e^-$/ADU, or alternatively photons/ADU.

## 14.1 Comparison of CCDs and Photon Counting Detectors

Photon counting detectors have essentially no readout noise. The dead time between individual photon counts is typically a microsecond, so that photomultiplier tubes can be highly linear up to $10^6$ counts/s. Two-dimensional photon counters are slower, however, because it takes time to determine the spatial position of each photon. They typically lose linearity at count rates of a few counts/pixel/s, limiting their use at high light levels.

| CCD advantages: | CCD disadvantages: |
|---|---|
| High quantum efficiency. | Readout noise. |
| Linear response. | Slow readout, hence dead time between exposures. |
| Large dynamic range. | Cosmic ray hits. |
| Fixed rectangular pixel format. | |
| | Saturation of the Analog-to-Digital converter. |
| | Hot pixels. |
| | Blocked columns. |
| | Incomplete charge transfer. |
| | Bleeding of charge from saturated pixels. |

The primary advantage of CCD detectors is their high 80% quantum efficiency compared to typically 20% for photo-cathodes. This 5-fold quantum efficiency advantage makes CCDs the best choice for most applications. The exception is when photon rates are so low that the CCD readout noise is larger than the photon counting noise. Two examples are extremely high spectral or time resolution, where photon counting detectors can still win out.

CCD readout noise has decreased steadily over the years. A modern CCD with 4 $e^-$/pixel rms requires a signal of only 16 photons to beat the readout noise. The readout noise can be decreased by reading out the chip more slowly – because the thermal noise on the readout amplifier is then averaged over a longer time. This implies a trade-off between readout noise and readout time.

For high-speed observing, the CCD readout time becomes a critical factor. As chips sizes grow, readout times increase in step with the number of pixels. The $2048 \times 4096$ pixel chips in use today can take several minutes to read out. For high-speed applications, frame transfer CCDs offer a solution. Only half of the chip is exposed to light, the other half being kept in the dark behind a mask. At the end of each exposure, the charge image is rapidly shifted to the masked half of the chip, and then it is read out while the original half of the chip is taking the next exposure. This effectively eliminates the dead time.

## 14.2 Bias Level and Readout Noise

The CCD readout amplifier requires a bias voltage which the ADC maps to a positive number even when no light reaches the detector. The bias level $B$ might typically be several 100 ADU.

Thermal noise in the amplifier causes the output voltage to fluctuate in time, which in turn causes the ADC output to fluctuate. The bias level therefore has a Gaussian distrubution with mean value $B$ and standard deviation $\sigma_B$.

The bias $B$ and rms readout noise $\sigma_B$ in ADU can be measured from a *bias exposure* obtained by reading out the chip without opening the shutter. Even better is to use the *overscan* and/or *underscan* regions. These are obtained during CCD readout by letting the ADC take several extra samples of the readout amplifier's output voltage just before and after reading out each row of pixels.

Because all pixels are read out through the same amplifier and ADC, the bias level and readout noise should be uniform over the entire image. If this is the case, then we may use a sample mean and variance over a set of $N$ pixels to estimate the bias and readout noise:

$$\bar{B} = \frac{1}{N}\sum_{i=1}^{N} B_i \tag{300}$$

$$\begin{aligned} \sigma_B^2 &= \frac{1}{N-1}\sum_{i=1}^{N}(B_i - \bar{B})^2 \\ &= \frac{1}{N-1}\left(\sum_{i=1}^{N} B_i^2 - N\bar{B}^2\right) \end{aligned} \tag{301}$$

Because cosmic rays can hit while the chip is being read out, they do appear on bias frames and in overscan and underscan regions even though the shutter was never opened. A $\sigma$-clip is effective in excluding from the sums any pixels zapped by cosmic rays.

The bias and readout noise you measure from a bias frame are in ADU. You can convert this to $e^-$ using the gain. A good CCD might have a readout noise of 4 $e^-$ rms or less. To beat this you would need to detect 16 or more photons/pixel. If your count rate is less, then you are losing information, and you should consider increasing the exposure time until the photon noise beats the readout noise. The two noise terms add in quadrature, so the effect of readout noise drops off rapidly as your exposure time increases. For example, if the rms readout noise is half the rms photon noise, the total noise is increased by only 20% rms.

## 14.3 Flat Fields

The CCD pixels are not identical. Some are more sensitive to light than others. Each has its own spectral sensitivity pattern $Q_i(\lambda)$. Write this as

$$Q_i(\lambda) = F_i(\lambda)Q(\lambda)\ , \tag{302}$$

where $Q(\lambda)$ is some mean quantum efficiency curve for the CCD, and $F_i(\lambda)$ is a correction factor, of order 1, to account for the individual pixel sensitivities. The wavelength dependence of $F_i$ is usually ignored, but it can could be investigated using flat fields taken through different filters.

To measure $F_i$, we take a **flat field** exposure, exposing the chip to a uniform light distribution. This may be bright twilight sky, or an illuminated panel inside the dome, or some internal lamp. If the illumination delivers an exposure of $\Phi$ (ADU/pixel) for the mean quantum efficiency $Q(\lambda)$, the response allowing for the individual pixel sensitivities will be $F_i\Phi$. The raw data we expect in a flat field exposure is then

$$\langle\, D_i \,\rangle = B_i + F_i\Phi/G. \tag{303}$$

The flat field factors $F_i$ may then be estimated by dividing the bias-subtracted flat field data by some suitable average over $N$ pixels

$$\hat{F}_i = \frac{D_i - B_i}{\frac{1}{N}\sum_{j=1}(D_j - B)} . \tag{304}$$

This normalization ensures that $\hat{F}_i \sim 1$, so that these estimates are small corrections.

Take a lot of flat field frames so that you can combine them to form a very high accuracy **master flat**. In combining the individual flat field exposures, take care to to allow for changes in the flat field illumination from one exposure to the next, and to omit corrupted data from any pixels affected by cosmic ray hits.

## 14.4 CCD Noise Model

The main sources of noise in CCD data are readout noise and photon counting noise. There are cosmic ray hits as well, affecting a small number of CCD pixels.

The **noise model** for raw CCD data is

$$\mathrm{Var}[\, D_i \,] = \sigma_B^2 + \frac{F_i\, \Phi_i}{G} \ . \tag{305}$$

The first term is the CCD **readout noise**, characterized by $\sigma_B$ ( rms in DN ), which is measured from bias images, as described below.

The second term is the photon counting noise, which depends on the CCD **gain** $G$ ( counts/ DN ). The gain can be measured from flat field images by quantifying how the photon counting noise increases with the exposure level, as described below.

It is customary to work with calibrated CCD data, i.e. after subtracting the bias and dividing by the flat field:

$$C_i = \frac{D_i - B}{F_i} \ . \tag{306}$$

The noise model for calibrated data is

$$\mathrm{Var}[\, C_i \,] = \frac{\sigma_B^2}{F_i^2} + \frac{\Phi_i}{F_i\, G} \ . \tag{307}$$

Calibrated data are often analyzed incorrectly by using a noise model appropriate for raw data – i.e. by setting the flat field factors to 1 when calculating the noise variances. This common but easily avoided error results in sub-optimal measurements with increased noise. The flat field $F_i$ is often less than 1, for example in vignetted parts of the image, or for pixels with lower sensitivity. These pixels are noisier, and if they are not given appropriately reduced weight they will degrade the quality of your measurements.

## 14.5 Measuring the CCD Gain

How can we measure the CCD Gain? The basic idea is to determine how the Poisson noise increases with the light level. This can be done using flat fields at different exposure levels. In low exposures readout noise contributes, so we need to make a correction for that.

### 14.5.1 from two identical flats

A quick way to measure the gain of a CCD is to look at the noise in flat field exposures. Suppose you have taken two identical flats, $X_1$ and $X_2$, with $\langle\, X_1 \,\rangle = \langle\, X_2 \,\rangle = \langle\, X \,\rangle$ and $\mathrm{Var}[\, X_1 \,] = \mathrm{Var}[\, X_2 \,] = \mathrm{Var}[\, X \,]$. Since the CCD noise model for raw data $X$ is

$$\mathrm{Var}[\, X \,] = \sigma_B^2 + \langle\, X \,\rangle / G \ , \tag{308}$$

then from the variance of the difference

$$\mathrm{Var}[\, X_1 - X_2 \,] = 2(\sigma_B^2 + \langle\, X \,\rangle / G) \ , \tag{309}$$

and the mean of the sum

$$\langle\, X_1 + X_2 \,\rangle = 2\langle\, X \,\rangle + 2\langle\, B \,\rangle \ , \tag{310}$$

you can calculate the gain

$$G = \frac{\langle\, X_1 + X_2 \,\rangle - 2\langle\, B \,\rangle}{\mathrm{Var}[\, X_1 - X_2 \,] - 2\sigma_B^2} \ . \tag{311}$$

Since the above applies for every pixel in the image, this approach yields a large number of independent gain estimates $G_i$, one for each pixel $i$ of the image,

$$G_i \equiv \frac{X_{1i} + X_{2i} - B_{1i} - B_{1j}}{(X_{1i} - X_{2i})^2 - 2\sigma_B^2} \ . \tag{312}$$

If the gain is the same for every pixel, then you can average these independent gain estimates,

$$\bar{G} = \frac{1}{N_i} \sum_{i=1} N_i G_i \ . \tag{313}$$

### 14.5.2 from many flats

You may have flat fields with different exposure levels, or exposure times. This is good for calibrating the gain since then you can verify whether or not the noise increases with the exposure level in the manner predicted by the CCD noise model.

Let's devise an optimal method for measuring the CCD gain from a flat field frame. Consider the noise model for calibrated CCD data. Rearrange the equation to obtain an expression for $1/G$:

$$\frac{1}{G} = \frac{F_i}{\Phi} \left( \left\langle\, (C_i - \Phi)^2 \,\right\rangle - \frac{\sigma_B^2}{F_i^2} \right) \ . \tag{314}$$

We have already established $F_i$ and $\sigma_B$ from the master flat field and analysis of noise in bias frames, respectively. Our task here is to estimate $G$ and $\Phi$. We opt to estimate $1/G$, rather than $G$, to avoid dividing by the noisy quantity $C_i - \Phi$, which may sometimes be 0.

On the flat field frame, or at least in some sub-region, $\Phi$ should be roughly constant – that's what a flat field frame means! We can therefore measure $\Phi$ to high accuracy using an optimal average over the pixels

$$\hat{\Phi} = \sum_i \frac{C_i F_i^2}{\sigma_B^2 + F_i\ \hat{\Phi}/\hat{G}} \Bigg/ \sum_i \frac{F_i^2}{\sigma_B^2 + F_i\ \hat{\Phi}/\hat{G}} \ . \tag{315}$$

The variances used for the optimal weights depend on $\Phi/G$, and for these we have inserted estimates $\hat{\Phi}$ and $\hat{G}$. These may initially be rough estimates, but we will improve them by iterating the equations. With care taken to omit cosmic ray hits, the optimal average over many pixels should deliver an essentially noise-free estimate of $\Phi$.

Next, we identify an unbiased single-pixel estimate for $1/G$:

$$\frac{1}{\hat{G}_i} = \frac{F_i}{\hat{\Phi}} \left( \left( C_i - \hat{\Phi} \right)^2 - \frac{\sigma_B^2}{F_i^2} \right) \ , \tag{316}$$

along with its variance

$$\mathrm{Var}\left[ \frac{1}{\hat{G}_i} \right] = \left( \frac{F_i}{\hat{\Phi}} \right)^2 \left( \frac{\sigma_B^2}{F_i^2} + \frac{\hat{\phi}}{F_i\ \hat{G}} \right) \ . \tag{317}$$

An optimal average over the pixels then updates our estimate for $1/G$:

$$\frac{1}{\hat{G}_i} = \sum_i \frac{\hat{\phi} F_i (C_i - \hat{\Phi})^2 - \sigma_B^2/F_i}{\sigma_B^2/\hat{\Phi} + F_i/\hat{G}} \Bigg/ \sum_i \frac{\hat{\phi}^2}{\sigma_B^2/\hat{\Phi} + F_i/\hat{G}} \ . \tag{318}$$

Because the error bars depend on $\hat{G}$ and $\hat{\Phi}$, it is necessary to iterate the equations to convergence.

# 15 High-Precision Variable Star Photometry

Here we develop optimal data analysis algorithms for high-precision photometry of variable stars in crowded fields.

Our approach is to develop a model $E(i,t)$ for the calibrated CCD data $C(i,t)$. We will use rough methods at first and optimal methods at the end to estimate parameters of the model. The model will include a sky background $S(i,t)$ plus a sum of $N_s$ discrete sources:

$$E(i,t) = S(i,t) + \sum_{s=1}^{N_s} f(s,t) P(i,s,t) \ . \tag{319}$$

The point-spread function $P(i,s,t)$ will be defined from the images of point sources.

## 15.1 Sky Background

The sky background may be uniform or may include spatial structure. The sky model may be intended to fit a large number of overlapping unresolved sources.

Start by dividing the image at time $t$ into a coarse grid of sub-images. Obtain initial sky estimates for each sub-image. Do this in a way that rejects stars and cosmic rays, e.g. sigma-clipping or median filtering.

You need a sky level $S(i,t)$ for every pixel. You can do this simply by interpolating the coarse grid of sky values. You can apply some smoothing to reduce noise. You can model the spatial structure by some smooth function, splines or low-order polynomials. A quadratic model would be

$$\begin{array}{rcccccc} S(i,t) & = & S_0(t) & + & S_x(t)\ \eta_x(i) & + & S_y(t)\ \eta_y(i) \\ & + & S_{xx}(t)\ \eta_{xx}(i) & + & S_{yy}(t)\ \eta_{yy}(i) & + & S_{xy}(t)\ \eta_{xy}(i) \ . \end{array} \tag{320}$$

Find the coefficients by linear regression. The spatial basis functions might be

$$\begin{array}{rcl} \eta_x(i) & = & (x(i) - x_0)/N_x \\ \eta_y(i) & = & (y(i) - y_0)/N_y \\ \eta_{xx}(i) & = & \eta_x^2(i) \\ \eta_{yy}(i) & = & \eta_y^2(i) \\ \eta_{xy}(i) & = & \eta_x(i)\ \eta_y(i) \ , \end{array} \tag{321}$$

where $(x_0, y_0)$ is a pixel near the centre of the frame, and $(N_x, N_y)$ are the number of pixels. A more nearly orthogonal basis may be constructed from these, e.g. by Graham-Schmidt orthogonalization, but it may not be worth the effort.

## 15.2 Source Detection

Once you have a sky level for each pixel, you can hunt for discrete sources. Base your source detection on the normalized residuals

$$\eta(i,t) = (C(i,t) - E(i,t))/\sigma(i,t) \ , \tag{322}$$

$$\sigma^2(i,t) = \frac{\sigma_B^2}{F^2(i)} + \frac{E(i,t)}{F(i)G} \ , \tag{323}$$

$$E(i,t) = S(i,t) + \sum_s f(s,t) P(i,s,t) \ . \tag{324}$$

Your initial model for $E(i,t)$ will include sky only, but as your source list develops you can include photon counting noise from discrete sources as well.

Be aware that your estimates for the sky levels, star fluxes, star positions, and point-spread functions are initially very rough, so be cautious. Add stars gradually.

Find the largest value of $\eta(i,t)$, and set a series of detection thresholds decreasing by some factor, say 2.

As soon as you have a PSF model, you can do a $\Delta\chi^2$ test comparing a star-plus-sky model relative to a sky-only model.

## 15.3 Differential Corrections

Once you have rough measurements of the sky $S(i,t)$, the point spread function $P(i,s,t)$, and the source fluxes $f(s,t)$ and centroids $x_0(s,t)$, $y_0(s,t)$, subtract these from the calibrated data to get residuals

$$\epsilon(i,t) = C(i,t) - E(i,t) \ . \tag{325}$$

The residuals will exhibit patterns that indicate the ways in which each of the fit parameters need to be improved. Examine residuals in the vicinity of each star, revising the parameters of those stars.

You can use a **differential correction model** to improve the the local sky model, the star flux and centroid.

$$\begin{aligned} \Delta\epsilon(i,t) &= \Delta S_0(i,t) \\ &+ \Delta S_x(t)\ \eta_x(i,s,t) \\ &+ \Delta S_y(t)\ \eta_y(i,s,t) \\ &+ \Delta f(s,t) P(i,s,t) \\ &- \Delta x(s,t)\ f(s,t)\ \frac{\partial P(i,s,t)}{\partial x} \\ &- \Delta y(s,t)\ f(s,t)\ \frac{\partial P(i,s,t)}{\partial y} \ . \end{aligned} \tag{326}$$

Fit this to the residuals in the vicinity of each star. The linearized model has 6 parameters that scale 6 basis functions, three for the sky, and 3 for the star. The parameters are revised by means of a linear regression fit. As the parameterization is designed to be approximately orthogonal, the fit should be well behaved, and can be solved by numerically inverting the $6 \times 6$ matrix, or by iteration.

To model the sky background in the vicinity of the star, a constant may suffice, or a linear function of $x$ and $y$ may be needed to follow sky gradients and nearby stars not yet included in the discrete source list. Basis functions $\eta_x(i,s,t) \propto x(i) - x_0(s,t)$ and $\eta_y(i,s,t) \propto y(i) - y_0(s,t)$, render the sky gradients approximately orthogonal to the other parameters.

The star flux and centroid and the sky gradients are roughly orthogonal parameters, but there is an unavoidable anti-correlation between the sky level and star flux

To update a star brightness:

$$f(s,t) \to f(s,t) + \sum_i \frac{\epsilon(i,t)\ P(i,s,t)}{\sigma^2(i,t)} \ \Bigg/ \sum_i \frac{P^2(i,s,t)}{\sigma^2(i,t)} \tag{327}$$

$$\mathrm{Var}[\, f(s,t)\,] = 1 \ \Bigg/ \sum_i \frac{P^2(i,s,t)}{\sigma^2(i,t)} \tag{328}$$

To update a star centroid:

$$x_0(s,t) \to x_0(s,t) - \sum_i \frac{\epsilon(i,t)\ f(s,t)\ \partial P(i,s,t)/\partial x}{\sigma^2(i,t)} \ \Bigg/ \sum_i \frac{f^2(s,t)\,(\partial P(i,s,t)/\partial x)^2}{\sigma^2(i,t)} \tag{329}$$

$$\mathrm{Var}[\, x_0(s,t)\,] = 1 \ \Bigg/ \sum_i \frac{f^2(s,t)\,(\partial P(i,s,t)/\partial x)^2}{\sigma^2(i,t)} \tag{330}$$

Similar results apply for updating $y_0(s,t)$, as well as for the local sky value $S_0(s,t)$ and the local sky gradients $S_x(s,t)$ and $S_x(s,t)$.

## 15.4 Accuracy of Stellar Brightnesses

How accurately may we expect to measure the brightness of a star? Suppose we know everything except the star brightness $f$. We can then subtract the sky and other nearby stars, and estimate the star brightness by scaling the PSF to match residual data.

The residuals after dividing by the exposure time $t$ and subtracting the sky and nearby stars are

$$\epsilon_i \ = \ C_i - S_i/t - \sum_{j \neq s}^{N_s} \Phi_j P_{ji}, \tag{331}$$

for which the corresponding model is

$$\begin{aligned} \langle\, R_i \,\rangle &= \Phi_s P_{si} \\ \mathrm{Var}[\, R_i \,] &= \left(\frac{\sigma_B}{tF_i}\right)^2 + \frac{S_i + \sum_{s=1}^{N_s} \Phi_s P_{si}}{tF_i G}. \end{aligned} \tag{332}$$

The familiar result for scaling a pattern to fit data is then

$$\hat{\Phi_s} = \frac{\sum_{i=1}^{N} P_{si} R_i / \mathrm{Var}[\, R_i \,]}{\sum_{i=1}^{N} P_{si}^2 / \mathrm{Var}[\, R_i \,]} \qquad \mathrm{Var}\left[\, \hat{\Phi}_s \,\right] = \frac{1}{\sum_{i=1}^{N} P_{si}^2 / \mathrm{Var}[\, R_i \,]} \tag{333}$$

It will be instructive to work out results for an isolated star. The noise variance on the star brightness is

$$\mathrm{Var}\left[\, \hat{\Phi} \,\right] = \left( \sum_{i=1}^{N} \frac{P_i^2}{\left(\frac{\sigma_B}{tF_i}\right)^2 + \frac{S_i + \Phi P_i}{tF_i G}} \right)^{-1} . \tag{334}$$

The squared signal-to-noise ratio of the star brightness measurement is

$$\frac{\Phi^2}{\mathrm{Var}\left[\, \hat{\Phi} \,\right]} = \sum_{i=1}^{N} \frac{\Phi^2 P_i^2}{\left(\frac{\sigma_B}{tF_i}\right)^2 + \frac{S_i + \Phi P_i}{tF_i G}} \tag{335}$$

To gain some intuition here, let's consider several individual sources of noise. If the dominant noise is spatially uniform, for example readout noise or sky noise.

## 15.5 Point-Spread Functions

A delicate step required for accurate stellar photometry is defining the point-spread function (PSF). We are fortunate that astronomical images are often generously endowed with star images providing high-quality information about the PSF.

The light from a star is recorded over a range of detector pixels. The starlight distribution defines the point-spread function $P_{s,i}$. The PSF is normalized to a sum of 1 over the $N$ pixels

$$\sum_{i=1}^{N} P_{s,i} = 1. \tag{336}$$

A number of effects ensure that the PSF is at least somewhat different for every star image. The PSF changes in time due to variable seeing, pointing, guiding. The PSF changes across the image due to image rotation and optical aberrations of the imaging system. The PSF depends on the spectrum of the star due to chromatic aberration and differential refraction and seeing effects. It is usually impractical to model these effects in detail, and so an empirical calibration of the PSF is usually adopted.

For many applications it will suffice to adopt a single PSF for each image

$$P_{s,i} = P(u, v, t) \ , \tag{337}$$

where $u = x_i - x_s$ and $v = y_i - y_s$ are the $X$ and $Y$ offsets of the CCD pixel at $(X_i, Y_i)$ measured from the star centroid at pixel coordinates $(x_s, y_s)$.

One may in principle derive an independent PSF for each star by simply normalizing each star image. This needs to be done with care so that nearby star images are subtracted and do not contribute.

$$P_{s,i} \to P_{s,i} + (C_i - S_i) / \Phi_s \ , \tag{338}$$

$$\mathrm{Var}[\,P_{s,i}\,] = \mathrm{Var}[\,C_i\,]\,/\Phi_s^2 \quad . \tag{339}$$

This PSF estimate is rather noisy, however, resulting in photometric errors that can be large particularly for faint stars. We can do better by optimally averaging over a number of star images.

Often an elite subset of bright isolated unsaturated star images is designated, the "psf stars", to define the psf on each frame. An optimal estimate will make use of information from a large number of stars images, employing appropriate inverse-variance weights to ensure that the noisier star images don't degrade the accuracy of the psf.

The psf may be estimated by taking inverse-variance weighted averages of "nearby" normalized star images.

$$P(x-x_0,y-y_0,t) \rightarrow P(x-x_0,y-y_0,t) + \sum_s \frac{\epsilon(x,y,t)\ f(s,t)}{\sigma^2(x,y,t)} \Bigg/ \sum_s \frac{f^2(s,t)}{\sigma^2(x,y,t)} \quad , \tag{340}$$

$$\mathrm{Var}[\,P(x-x_0,y-y_0,t)\,] = 1 \Bigg/ \sum_s \frac{f^2(s,t)}{\sigma^2(x,y,t)} \quad . \tag{341}$$

This defines a psf relative to the centroid of the star image.

### 15.5.1 PSF interpolation

As star images are seldom exactly centred on a pixel, some blurring of the psf is inevitable when averaging the normalized star images. Efforts to reduce the blurring may employ an interpolation scheme to shift each star image to the nearest pixel centre before averaging with other similarly-shifted star images to form the psf. Interpolating noisy data is a dubious procedure, however. One must also interpolate the pixel-centred psf model back to the actual centroid of each star.

An alternative to interpolation is to construct an array of psfs intended to apply to stars with centroids falling close to each node of an array of positions within the pixel. This may be accomplished with a weighted average selecting star images whose centroids are close to the desired sub-pixel position.

A third approach fits normalized star images without interpolation with a psf model that is a low-order polynomial function of the sub-pixel centroid position.

A more sophisticated treatment allows the PSF shape to change with position on the detector, and with the wavelength of the light

$$P_{s,i} = P(u,v,t,x_s,y_s,\lambda) \ . \tag{342}$$

Spatial gradients arise from aberrations in the telescope and camera optics, and because of the airmass gradient across the field of view. The spatial gradients in the PSF may be modelled with low-order polynomials. For example, a quadratic model would be

$$\begin{array}{rcccccc} P_{s,i} & = & P_0(u,v) & + & P_x(u,v)\ \eta_x & + & P_y(u,v)\ \eta_y \\ & + & P_{xx}(u,v)\ \eta_x^2 & + & P_{yy}(u,v)\ \eta_y^2 & + & P_{xy}(u,v)\ \eta_x\ \eta_y \ . \end{array} \tag{343}$$

The coefficients are found by linear regression fits to a normalized star images. The spatial basis functions may be taken to be

$$\begin{array}{rcl} \eta_x & \equiv & (x-x_0)/N_x \\ \eta_y & \equiv & (y-y_0)/N_y \\ \eta_{xx} & \equiv & \eta_x^2 \qquad\qquad rcl \\ \eta_{yy} & \equiv & \eta_y^2 \\ \eta_{xy} & \equiv & \eta_x\ \eta_y \ , \end{array} \tag{344}$$

where $(x_0,y_0)$ is a pixel near the image centre, and $(N_x,N_y)$ are the number of pixels. A more nearly orthogonal polynomials may be constructed from these, but this may not be worth the effort since the required inverse-variance weights at the star positions are unknown and changing as more stars are added to the fit.

The PSF shape also depends on wavelength because the atmosphere is like a prism, bending blue light more than red light. The blue light from a star appears slightly higher in the sky than the red light. This effect is what causes the 'green flash' sometimes seen at sunset when the red image of the Sun sets a few seconds before its blue image. The image of each star is dispersed vertically into a low-resolution spectrum. A filter is generally used to limit the range of wavelengths reaching the detector, but even so stars with different spectra have slightly different PSFs.

It is difficult to include colour-dependent effects in the PSF without some knowledge of the spectra of the stars being imaged. A simple approach would be to allow the PSF to depend linearly on a colour index $c_s$. The colour index is the magnitude difference of the star for two filters with different wavelengths, which may adequately characterize the spectral differences by distinguishing the redder stars from the bluer ones. This is usually not done, and the result of using the same PSF for stars of different colour is a small systematic error that depends on the colour of the star. Such errors are usually calibrated out later by fitting the magnitude residuals by linear functions of colour indices.

The PSF shape may be modelled using analytic functions, e.g. a Gaussian

$$P(u,v) = \frac{1}{2\pi\Delta^2} e^{-\frac{1}{2}\left(\frac{r}{\Delta}\right)^2} \tag{345}$$

or a modified Lorentzian

$$P(u,v) \propto \frac{1}{1+(r/\Delta)^p}, \tag{346}$$

where $r^2 = u^2 + v^2$ is the distance from the centre of the PSF and $\Delta$ controls the PSF width. The rms width $\Delta$ may be estimated from the second moments of normalized star images.

Ellipticity may also be needed to allow for elongated star images due to differential refraction or imperfect guiding during the exposures. The ellipticity is introduced by writing

$$r^2 = (u/\Delta_u)^2 + (v/\Delta_v)^2 + u/v/\Delta_{uv} \tag{347}$$

$$\Delta^2 = \Delta_u^2 + \Delta_v^2 + \Delta_{uv} \ . \tag{348}$$

Fitting the non-linear parameters of analytic PSF models may be slow. The three PSF parameters $\Delta_u$, $\Delta_v$, $\Delta_{uv}$ may be found initially from moments of bright star images, and then improved by weighted averages of differential corrections.

The analytic models may suffice for fainter stars, but brighter stars require a more accurate model. These may be included as a PSF map added to the the analytic model.

### 15.5.2 PSF modelling

### 15.5.3 PSF gradients

Adopting a single psf for the entire image, while admittedly a rough approximation, is nevertheless satisfactory for many purposes. Provided the psf model is accurately normalized, errors in the star centroid and psf model should affect the photometry only in second order. The photometric errors may also be to some extent correctable in differential photometry where each star is calibrated relative to its neighbors.

More sophisticated treatments allow the psf to vary across the image. One approach is to construct an array of psfs by averaging normalized star images from sub-regions of the image. Interpolation then yeilds a representation of the psf centred at any desired position within the frame.

A second approach is to represent the psf as a low-order polynomial function of position on the image. The psf models is then calibrated by a linear regression fit to the normalized star images.

## 15.6 Astrometry

More accurate star brightness measurements are likely to result if the star centroids are accurately known rather than having to be re-determined from each star image. A bias may arise in star brightness measurements if the star centroid is not fixed, because the fit is then free to centre up on the nearest noise spike.

When multiple images of the same field are available, star positions may be determined from a simultaneous fit to all images rather than independent determinations from each image. Care must be taken, however, because the star brightness measurements will be severely biased if the star centroids are not accurate.

## 15.7 Differential Photometry

Systematic errors may dominate.

Calibrate systematic errors using stars that have similar pattern of residuals in time.

# 16 Absolute Photometry and Spectrophotometry

## 16.1 magnitudes

Ancient Greek astronomers assigned stars to magnitude classes 1 throuth 6 from the brightest to faintest stars visible in the night sky. Today, we can predict and measure brightnesses very precisely, yet magnitudes survive in the language used by astronomers to express the brightness of a source.

Magnitudes are a precise logarithmic brightness scale.

$$m_1 - m_2 - -2.5 \log\left(f_1/f_2\right) \ , \tag{349}$$

where $f_1/f_2$ is the ratio of the two stellar fluxes. The factor 2.5 stretches the magnitude scale so that 5 magnitudes is a factor 100, roughly the range of brightness covered by the ancient Greek classification.

Magnitudes anoy some physicists and radio astronomers, who would much prefer to use fluxes, or logarithms of fluxes, without the extra factor 2.5. If you learn and practice a few simple rules, you will soon become fluent with the astronomers magnitudes.

5 magnitudes is a factor of 100, exactly.
1 magnitude is a factor of $100^{1/5} \approx 2.512$.
2 magnitude is a factor of $10^{0.4} \approx 2.512^2 \approx 6.3$.
3 magnitude is a factor of $10^{0.6} \approx 2.512^3 \approx 15.8$.

It is a happy accident that
0.1 magnitude is a brightness change of about 10%.
0.01 magnitude is a brightness change of about 1%.

A beautiful and charming aspect of magnitudes is that they embrace in a convenient way both large changes in brightness (5 magnitudes per factor 100) and small changes in brightness (0.1 mag is about 10%) while maintaining contact with the ancient system used by the early observers (stars visible to the human eye in a dark sky are of the 1st through 6th magnitude.)

## 16.2 standard photometric systems

In the early days, when astronomers had just begun to measure brightnesses of stars, the bright star Vega was adopted as a zero-magnitude standard star. The brightness of any star measured by any detector system could always be defined relative to Vega. The magnitude is then

$$m = -2.5 \log\left(f/f(\mathrm{Vega})\right) \tag{350}$$

Using colour filters, fluxes and magnitudes of stars can be measured in different parts of the spectrum. The Johnson $UBV$ system based on filtered photometry with photon-counting photomultiplier tubes became the standard. The wide-band filters were chosen with $U$ and $B$ on opposite sides of the Balmer continuum edge at 3640Å, and $V$ near 5500Å corresponding roughly to the peak of the human eye's sensitivity. As red-sensitive detectors became available, the $UBV$ system was extended to include Cousins $R$ and $I$.

A hot star is brighter in blue light and fainter in red light. A colour index is the difference of the magnitudes measured with two different filters

$$B - V = -2.5 \ \log\left(f(B)/f(V)\right) \tag{351}$$

where here the flux units are understood to be those of Vega measured in the same $B$ and $V$ bandpasses, so that Vega has color index of 0. A star hotter than Vega has $B - V < 0$.

As spectrographs were developed, improving knowledge of the spectral energy distributions of stars made it feasible to define magnitude systems based on physical fluxes rather than dividing by the spectrum of Vega. Two systems in use are the $\mathrm{AB}_\nu$ magnitude system, based on fluxes in $f_\nu$ units, and the $\mathrm{ST}_\lambda$ system, based on fluxes in $f_\lambda$ units.

$$\begin{aligned} \mathrm{AB}_\nu &\equiv -48.6 - 2.5 \ \log\left(\frac{f_\nu}{\mathrm{erg\ cm^{-2}s^{-1}Hz^{-1}}}\right) \\ &= 16.4 - 2.5 \ \log\left(\frac{f_\nu}{\mathrm{mJy}}\right) \\ \mathrm{ST}_\lambda &\equiv 21.1 - 2.5 \ \log\left(\frac{f_\lambda}{\mathrm{erg\ cm^{-2}s^{-1}\AA^{-1}}}\right) \end{aligned} \tag{352}$$

The zero points are chosen so that Vega is magnitude zero in the $V$ bandpass, which is a green wavelength region close to the peak sensitivity of the human eye. The mJy is $10^{-26}$ erg cm$^-$2s$^-$1Hz$^-$1, a flux unit used by radio astronomers. Note that 1 mJy is AB magnitude 16.4.

## 16.3 broadband fluxes

It is possible to make a clean definition of fluxes and wavelengths and magnitudes in such a way that photometry and spectrophotometry are united. This allows photometry and spectra of a source to be plotted in a consistent way on the same diagram. More importantly, models that predict the spectrum of a source can then be fitted simultaneously to the broad-band photometric measurements as well as to the higher resolution spectrophotometric measurements.

In photometry, the spectral bandpass $P(\lambda)$ gives the fraction of photons at wavelength $\lambda$ that are detected. For a spectrograph, each pixel $i$ has its own narrow passband $P_i(\lambda)$, centred at wavelength $\lambda_i$ with a bandwidth $\Delta\lambda_i$, not necessarily equal to the pixel spacing.

For a wide passband, an observation does not measure the flux at some wavelength, but rather a weighted average of the flux over a range of wavelengths. Observing a star of with spectrum $f_\nu(\lambda)$, using a detector with bandpass $P(\lambda)$, the count rate (photon cm$^{-2}$s$^{-1}$) is

$$\Phi(P) = \int P(\lambda) f_\nu(\lambda) d\nu/h\nu \ . \tag{353}$$

Division by $h\nu$ converts the energy units to photons, and $P(\lambda)$ is the fraction of those that are detected.

What mean flux and wavelength should we assign to this measurement? What source with a spectrum that is constant in $f_\nu$ would give the same count rate?

$$f_\nu(P) \equiv \frac{\int f_\nu(\lambda)\ P(\lambda)\ d\lambda/\lambda}{\int P(\lambda)\ d\lambda/\lambda} \ . \tag{354}$$

If you prefer $f_\lambda = cf_\nu/\lambda^2$, the analogous result is

$$f_\lambda(P) \equiv \frac{\int f_\lambda(\lambda)\ P(\lambda)\ \lambda d\lambda}{\int P(\lambda)\ \lambda d\lambda} \ . \tag{355}$$

Thus the broadband fluxes $f_\nu(P)$ and $f_\lambda(P)$ are weighted averages of the source spectrum, $f_\nu(\lambda)$ or $f_\lambda(P)$ over the bandpass $P(\lambda)$. The weights should be exactly as given above, otherwise the broadband flux is not proportional to the observed count rate. Watch out for other definitions, averaging the source spectrum over the bandpass without including the extra factors of $\lambda$ for $\langle\, f_\lambda \,\rangle$ and $1/\lambda$ for $\langle\, f_\nu \,\rangle$.

Broadband magnitudes $\mathrm{AB}_\nu(P)$ and $\mathrm{ST}_\lambda(P)$ are then defined in the usual way from these definitions of the broadband fluxes.

## 16.4 pivot wavelength

What wavelength do we associate with the broad-band measurement? The mean wavelength

$$\langle\, \lambda \,\rangle = \frac{\int P(\lambda)\ \lambda d\lambda}{\int P(\lambda)\ d\lambda} \ , \tag{356}$$

does not reflect the photons actually detected, which will have shorter wavelengths when a blue source is observed and longer wavelengths when a red star is observed. We could include the source photon distribution in the weighting,

$$\langle\, \lambda \,\rangle \equiv \frac{\int P(\lambda)\ f_\nu(\lambda)\ d\lambda}{\int P(\lambda)\ f_\nu(\lambda)\ d\lambda/\lambda} \ , \tag{357}$$

but then the wavelength depends in detail on the source spectrum, which may not be known.

For narrow bands, we convert fluxes using $f_\lambda = cf_\nu/\lambda^2$. With a broad bandpass, it is not so obvious what wavelength, if any, will preserve this relationship for broad-band fluxes. The wavelength that does the trick is the *pivot wavelength*

$$\lambda_p(P) \equiv \left(\frac{cf_\nu(P)}{f_\lambda(P)}\right)^{1/2} = \left(\frac{\int P(\lambda)\ \lambda d\lambda}{\int P(\lambda)\ d\lambda/\lambda}\right)^{1/2} \quad . \tag{358}$$

The dependence on the stellar spectrum cancels, so that the pivot wavelength for each passband is the same no matter what type of source spectrum is being investigated.

## 16.5 zero point calibration

Let's efine the *instrumental magnitude* to be

$$m = -2.5 \log \mathrm{ADU/s} \ . \tag{359}$$

Then the aim of a photometric transformation is to be able to transform our measured instrumental magnitude into a magnitude on some standard system. To be specific, let's assume we want to transform instrumental magnitudes $m$ to standard Johnson $V$ magnitudes. The simplest model is a constant offset:

$$V = m - m_1 \ . \tag{360}$$

Here the parameter $m_1$ is the $V$ magnitude of a star that produces a signal of 1 ADU/s, and is therefore a measure of the sensitivity of your system. $m_1$ is sometimes called the *zero point* of the instrumental magnitude system.

Consider a specific example. Suppose one of the stars you have imaged has magnitude $V = 6.5$ in the standard system, in this case Johnson $V$. When you subtract the local sky level, and then add up the star counts over all pixels of the psf, you find that this star produces a signal of 12345 ADU in a 10 second exposure. Then the $V = 6.5$ mag star produced a signal of 1234.5 ADU/s. $m_1$ must therefore be fainter than 6.5 by a factor $2.5 \log 1234.5$.

$$m_1 = 6.5 + 2.5 \log 1234.5 = 14.23 \tag{361}$$

A star of magnitude $V = m_1 = 14.23$ will produce a signal of 1 ADU/s.

Once you have measured $m_1$, you can then predict that a star of magnitude $V$, observed with your system, will produce a signal of $10^{0.4(m_1 - V)}$ ADU/s.

## 16.6 atmospheric extinction

In practice, of course, you observed the sky through the Earth's atmosphere, and this diminishes the starlight. To account for that, the usual model is

$$m = V + m_1 + m_a(a - a_0) \tag{362}$$

where $a \approx \sec(z)$ is the *airmass* at which you observed the star, and $a_0$ is some standard airmass – usually chosen to be 0 or 1.

The *zenith distance* $z$ is the angle between the star and the zenith at the time of observation. The airmass is 1 for a star observed at the zenith, and it increases as $\sec(z)$ (approximately) as the zenith distance $z$ increases.

To determine *the extinction coefficient* $m_a$, which has units of magnitudes per airmass, you need to observe your star at several times during the night, catching it at several different airmasses. Plot the instrumental magnitude $m(a)$ vs airmass $a$, and fit a straight line to determine $m_1$ and $m_a$. With this model, $m_1$ is the $V$ magnitude of a star that produces 1 ADU/s when observed at the standard airmass $a_0$.

## 16.7 colour terms

You can calculate $m_a$ independently for any star that has been observed at several different airmasses, and $m_1$ from any star whose magnitude is known in the standard system. The answeres should be similar for each star you try. In practice, however, there will be differences from star to star, because stars have different spectra.

The zero point $m_1$ will be different for red and blue stars. For example, an unfiltered CCD passband generally has more response in the red than the standard $V$ passband, so that red stars appear brighter to the CCD than blue stars with the same $V$ magnitude. Similarly, extinction is higher in the blue than the red, and so $m_a$ also depends on the star colour.

To calibrate these colour effects, you can plot the derived values of $m_1$ and $m_a$ vs some colour index $c$, say $c = B-V$. A linear model may be a good fit:

$$m - V = m_1 + m_a * (a - a_0) + m_c * (c - c_0) + m_{ca}(a - a_0)(c - c_0) \ . \tag{363}$$

Here $c_0$ is set arbitrarily, and the parameters $m_1$, $m_a$, $m_c$, and $m_{ca}$ are determined from a fit to the data.

Once the 4 parameters are determined, tour model can predict the instrumental magnitude, and hence the count rate, of a star given its $V$ magnitude and colour index $c = B - V$ when observed at any airmass.

# 17 Spectroscopy

## 17.1 wavelength calibration

## 17.2 optimal spectrum extraction

## 17.3 atmospheric corrections

## 17.4 slit loss corrections

## 17.5 flux calibration

## 17.6 cross-correlation velocities

# 18 Astro-Tomography

## 18.1 eclipse mapping

## 18.2 doppler tomography

## 18.3 echo mapping

## 18.4 physical parameter mapping

# 19 Acknowledgements

Thanks to Alison Campbell for suggesting the title.

Thanks to the following people for corrections and helpful comments on early versions of the manuscript: Alison Campbell, Andrew Collier-Cameron, Steve Kane, Rachel Street.

All errors remaining are the sole responsibility of the author.