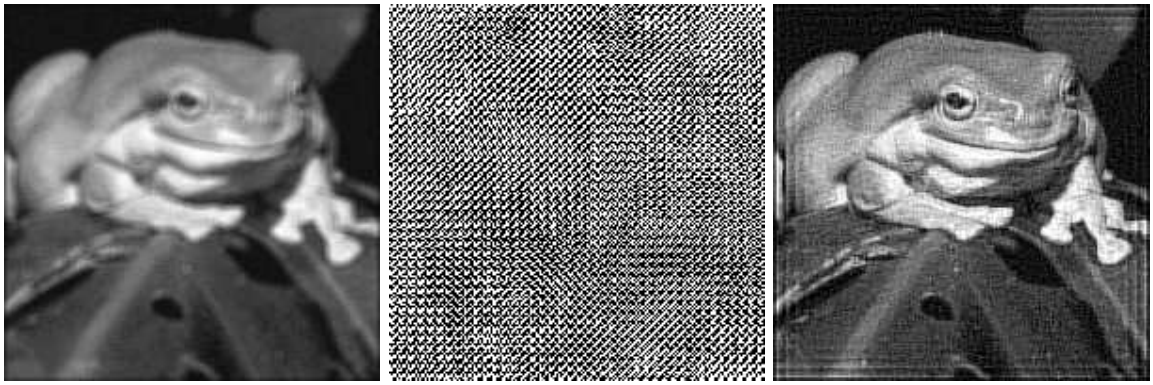
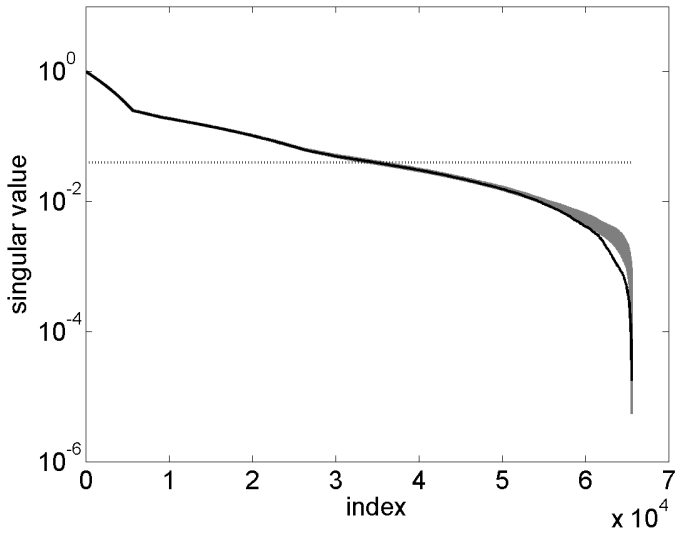
Inverse Problems

Course notes for ELEC 445 – Inverse Problems and Imaging



blurred direct inverse regularized inverse



Colin Fox, Geoff K. Nicholls, Sze M. Tan
2016 Edition

Contents

Preface

These notes accompany the lecture course ELEC 445 Inverse problems and Imaging taught as part of the Electronics fourth year in the Physics Department at the University of Otago.

Inverse problems occur whenever data is observed that depends on unknown quantities that we want to determine. That is just about every case where measurements are made, so the study of inverse problems is often described as the 'theory of experiment'. The term 'inverse problem' is usually reserved for cases where the deterministic mapping from unknowns to data is a complex physical relationship and where direct inversion presents analytic difficulties. In those cases, inverse problems are characterized by the solution being sensitive to errors in the data and the physical model. Examples of inverse problems include the various modalities of imaging from wave scattering used in non-invasive medical diagnostics, geophysical prospecting, and industrial process monitoring.

The course and these notes focus on general methods for understanding and solving inverse problems, and we will develop practical computational techniques for their solution. The sensitivity to errors means that direct inversion is seldom practical. A classical solution (in applied mathematics) is offered by regularization. Quantification of uncertainties can be provided by a probabilistic model for the measurement process, with inversion achieved via statistical inference.

We developed these notes around 1995 for the course 707 Inverse Problems in the Physics Department at the University of Auckland. At that time Geoff and Colin worked in the Mathematics Dept while Sze was in Physics. Since then, we have all left Auckland University; Sze moved to a high-tech company in the Silicon Valley, Geoff moved to the Stats Lab in Oxford, and Colin moved to Physics in Otago University to teach Electronics.

The online version of these notes seem to have been useful to a number of groups around the world who want to learn about pratical methods for solving inverse problems. In 2005 these notes ranked around 60 in Google's list of downloads of mathematical texts online, and it has been a big pleasure to subsequently meet people who have found the notes useful. We hope that you also find something of value here.

CF, GKN, SMT

Dunedin, Oxford, Sunnyvale
October 2016

1

Introduction to Inverse Problems

1.1 Examples of Inverse Problems

The aim of collecting data is to gain meaningful information about a physical system or phenomenon of interest. However, in many situations the quantities that we wish to determine are different from the ones which we are able to measure, or have measured. If the measured data depends, in some way, on the quantities we want, then the data at least contains some information about those quantities. Starting with the data that we have measured, the problem of trying to reconstruct the quantities that we really want is called an *inverse problem.* Loosely speaking, we often say an inverse problem is where we measure an *effect* and want to determine the *cause*.

Most science and engineering is data-driven in this way, though not always called an 'inverse problem'. Here we want to discuss the features and solution methods that are characteristic for the problems most typically treated under the umbrella of inverse problems. The quintessential setting is where the measurement process is a complex physical relationship, and inversion presents analytic difficulties.

Here are some examples of inverse problems:

- **Computer axial tomography.** Given a patient, we wish to obtain transverse slices through the body *in vivo,* and display pictures of these slices. It is known that the X-rays are partially transmitted through the body, and that the opacity of various internal structures to X-rays varies, so that a picture of the variation of the absorption coefficient in the body would give a good picture. However, the only measurements that one can make non-invasively is to shine X-rays through the patient and to measure the *total* absorption along lines through the body. Given a collection of such line integrals (the "data"), how do we reconstruct the absorption as a function of position in the body (the "image")?

- **Model fitting.** According to some theoretical model, the value of a quantity y depends on another quantity x via an equation such as

$$y = a + bx + cx^2 + dx^3. \tag{1.1}$$

  Given a set of measured points (x_i, y_i) (which are the "data" in this problem), how do we determine the values of a, b, c and d, and how confident are we of the result? In this case the "image" which we wish to determine is the set of numbers a through d. More generally, of course, the model can be more complicated and may depend on the

image in a non-linear way. Determining the half-life of a radioactive substance from measurements of the times at which decay products are detected is an example of model fitting.

- **Deconvolution.** Given a blurred photograph, or the result of passing a signal through a medium which acts as a filter, how can we reconstruct an unblurred version of the photograph, or the original signal before the filtering occurred? This type of problem is very important in designing computer modems, for example, because telephone lines will distort signals passing through them, and it is necessary to compensate for these distortions to recover the original signal. The problem of characterizing a linear, shift-invariant system by determining its impulse response is usually a problem in deconvolution.

- **Gridding or regridding.** Suppose that we wish to make a contour map of the height above sea-level of a region. This would be relatively easy if we had measurements of height on a regular grid of points so that the height between these points can be inferred by interpolation. In practice, we usually collect height data at irregularly spaced points with variable density of points in different locations. How do we use such data to reconstruct estimates of height on a regular grid? The problem of drawing isobars on a weather map from isolated barometer readings is essentially the same.

- **Radio-astronomical imaging.** When using a multi-element interferometer as a radio telescope, it turns out that the measured data is not the distribution of radio sources in the sky (called the "sky brightness" function) but is rather the Fourier transform of the sky brightness. It is not possible to measure the entire Fourier transform, but only to sample this transform on a collection of irregular curves in Fourier space. From such data, how is it possible to reconstruct the desired distribution of sky brightness?

- **Navigation.** When travelling in a boat or plane, it is useful to have an idea of the current location in close to real time. This is often done by making a variety of measurements, for example by using bearings to landmarks, stellar or satellite positions, and also by considering one's previous position and using information such as records of speed and heading. How should all of these separate pieces of information be combined together to give a coherent description of the vessel's motion?

- **Image analysis.** How does one automatically count the number of stars in a photograph of the sky, or the number of red blood cells in a microphotograph of a slide of a blood sample? The objects will generally overlap or may be of a variety of shapes. In these cases, the "data" is, typically, a picture of a scene containing the objects to be counted and the inverse problem is to find the number of objects. Closely related is the problem of image segmentation; A typical example is the problem of classifying regions of a satellite image of the earth's surface into regions of ocean, forest, agricultural land, etc.

- **Integral equations.** Many physical processes can be written as integral equations, including particle scattering and deformation of composite materials. Solution of integral equations such as the Fredholm equation of the first kind

$$\int_{s_0}^{s_1} k(x,s)\, f(s)\, \mathrm{d}s = d(x), \qquad x_0 \le x \le x_1, \tag{1.2}$$

where the kernel k and the function d are given, presents an inverse problem for the unknown function f. The special case where $k(x,s) = u(x-s)$ (u is the unit step) is the problem of numerical differentiation of the function d.

- **Geophysics.** Inverse problems have always played an important role in geophysics as the interior of the Earth is not directly observable yet the surface manifestation of waves that propagate through its interior are measurable. Using the measurements of seismic waves to determine the location of an earthquake's epicentre, or the density of the rock through which the waves propagate, are typical of inverse problems in which wave propagation is used to probe an object. Like many classes of inverse problems, "inverse eigenvalue problems" were first investigated in geophysics when, in 1959, the normal modes of vibration of the Earth were first recorded and the modal frequencies and shapes were used to learn about the structure of the Earth in the large.

From this very short and incomplete list, it is apparent that the scope of inverse problem theory is extensive and its applications can be found in many diverse fields. In this course, we shall be discussing various *general* methods for approaching such problems.

In this introductory chapter, we shall consider one of these problems – deblurring of a photograph – and highlight the ways in which it is representative of other inverse problems.

1.2 Image Space, Forward Map, Data Space, and Noise

In accordance with convention, the collection of values that we want to reconstruct is referred to as the *image*, even if those values do not represent a picture but are simply parameters that define a model. The set of all images is called *image space*. We usually denote the image by f .

The *forward problem* is the mapping from the image to the quantities that we are able to measure. In most of the examples that we consider, the details of the forward problem is given by some physical theory. For example, given the half-life of a radioactive substance, nuclear physicists can tell us how to calculate the time at which we will detect the decay products – at least in a statistical sense. The forward mapping may be linear or nonlinear and is denoted by A.

In practice we are never able to make exact measurements and the *data* that we measure are a corrupted version of the measurement quantities. *Data space* is the set of all possible data. The corruption could be as small as the roundoff error produced by a computer representation of the measurements, it could be intrinsic in the measurement process such as the twinkle of star brightness produced by a turbulent atmosphere, or, more usually, the corruption is due to the inherent errors in the measurement process. So, strictly, the forward process is a mapping from the image to error-free data, $\bar{d}$, and the data we actually measure, d, is the corrupted form. The difference $\bar{d} - d$ is called the *noise* which we denote by n.

Thus the mapping from the image to actual data is given by the relation

$$d = A(f) + n.$$

The *inverse problem* is then the problem of finding the original image given the data and knowledge of the forward problem.

For example, in the case of deblurring photographs, the "image" is the sharp photograph, the "data" is the blurred photograph, and the forward problem is the blurring process. The inverse problem is to find the sharp photograph (image) from the blurred photograph (data) and knowledge of the blurring process.

1.3 Ill-Posed and Ill-Conditioned

There are several basic classifications of the forward problem. One classification that is useful when using low-level image representations, such as pixel images, depends on whether the image and data are functions of a continuous variable, or values at a finite set of discrete points, i.e., are infinite-dimensional or finite-dimensional. These could be classified as:

		data	
		∞-dim	finite-dim
image	∞-dim	data and image are functions (idealized case)	data discrete, reconstruct function image
	finite-dim	data is function, reconstruct discrete image	data and image discrete (computed case)

The Fredholm equation 1.2 is an example of a ∞-dim–∞-dim forward problem since both the functions $y(x)$ and $z(s)$ are defined on an interval (and require an infinite number of values to define them). The model-fitting problem in equation 1.1 is a case where the image is discrete (defined by the 4 values a, b, c, and d) and if there are a finite number of data values it gives a discrete-discrete problem

In practice, we can only ever measure a finite number of data values and so assuming ∞-dim data is always an idealization of an actual problem, though it is one that is commonly used in Physics as it can lead to a simplified analysis. Ultimately, computer implementation is necessarily finite dimensional, and for these purposes, each of idealized cases that have ∞-dim image or data is approximated on the computer by a problem that has finite-dim data and finite-dime image. This is not quite true for high-level representations; data is always finite-dimensional but it is possible to compute with countably-infinite image representations.

Whichever class the mapping A belongs to, the inverse problem of solving

$$A(f) = d \tag{1.3}$$

for f given d is called *well-posed* (by Hadamard in 1923) if:

(∃) a solution *exists* for any data d in data space,

(!) the solution is *unique* in image space, and

(C) the inverse mapping $d \mapsto f$ is *continuous*.

Condition (∃) holds when the range of the operator A is all f data space while condition (!) also requires that A is one-to-one. Together, conditions (∃) and (!) are equivalent to saying that the operator A has a well defined inverse A^{-1} with domain that is all of data space.

The requirement in condition (C) of continuous dependence of the solution image on the data is a necessary but not sufficient condition for the stability of the solution.

In the case of a well-posed problem, relative error propagation from the data to the solution is controlled by the *condition number*: if Δd is a variation of d and Δf the corresponding variation of f, then

$$\frac{||\Delta f||}{||f||} \leq \text{cond}\,(A)\,\frac{||\Delta d||}{||d||} \tag{1.4}$$

where (for linear forward problems)

$$\text{cond}\,(A) = ||A||\,\left|\left|A^{-1}\right|\right|.$$

If you have not met the concept of the norm of an linear operator (or transformation) before, the formal definition is

$$||A|| = \sup_{x \neq 0} \frac{||Ax||}{||x||}.$$

The quantity on the right hand side is in terms of (usual) vector norms and measures by how much the transformation "stretches" a vector x (in general, of course, A will also change the direction of the vector, so that Ax and x do not point in the same "direction"). The norm of the operator is found by considering the stretching factor for all non-zero vectors x, and finding the largest such factor. (*Technical note:* Mathematically, we use the supremum rather than the maximum operation in the definition in case the maximum is not achieved by any non-zero vector x. In such cases there is a sequence of nonzero x_n with the property that $||Ax_n|| \,/\, ||x_n||$ increases asymptotically to $||A||$).

Since the fractional error in f equals the condition number multiplied by the fractional error in d, smaller values of cond(A) are desirable. In the linear case, cond$(A) \geq 1$, and the case cond$(A) = 1$ occurs when A is similar to a multiple of the identity. If cond(A) is not too large, the problem 1.3 is said to be *well-conditioned* and the solution is stable with respect to small variations of the data. Otherwise the problem is said to be *ill-conditioned*. It is clear that the separation between well-conditioned and ill-conditioned problems is not very sharp and that the concept of well-conditioned problem is more vague than the concept of well-posed problem.

Hadamard went on to define a problem to be *ill-posed* if it does not satisfy all three conditions. So an ill-posed problem is one where an inverse does not exist because the data is outside the range of A, or the inverse is not unique because more than one image is mapped to the same data, or because an arbitrarily small change in the data can cause an arbitrarily large change in the image.

Hadamard believed that ill-posed problems were actually incorrectly-posed and "artificial" in that they would not describe physical systems. That is not the case. Most correctly stated inverse problems turn out to be ill-posed; In fact all of the examples in section 1.1 are examples of actual problems in which the inverse problem is ill-posed or at least ill-conditioned. The facts that CAT scans are performed successfully every day, or that oil reservoirs have been found by seismic investigation, is evidence that meaningful information can be gained from ill-posed inverse problems even though they cannot be strictly inverted.

The classical example of an ill-posed problem is a Fredholm integral equation of the first kind with a square integrable (Hilbert-Schmidt) kernel

$$\int_a^b k\,(x,s)\,f\,(s)\;\mathrm{d}s = d\,(x)\,, \qquad a \leq x \leq b. \tag{1.5}$$

If the solution f is perturbed by $\Delta f(s) = \epsilon \sin(2\pi ps)$, $p = 1, 2, \ldots$, ϵ = constant, then the corresponding perturbation of the right-hand side $d(x)$ is given by

$$\Delta d(x) = \epsilon \int_a^b k(x,s) \sin(2\pi ps)\, \mathrm{d}s, \qquad p = 1, 2, \ldots,$$

and due to the Riemann-Lebesgue lemma it follows that $\Delta d \to 0$ as $p \to \infty$. Hence, the ratio $\frac{||\Delta f||}{||\Delta d||}$ can become arbitrarily large by choosing the integer p large enough[1], thus showing that 1.5 is an ill-posed problem because it fails condition 3. In particular, this example illustrates that Fredholm integral equations of the first kind with square integrable kernels are extremely sensitive to high-frequency perturbations.

Strictly speaking, a problem that is ill-posed because it fails condition 3 must be infinite dimensional – otherwise the ratio $\frac{||\Delta f||}{||\Delta d||}$ stays bounded, although it may become very large. However, certain finite-dimensional discrete problems have properties very similar to these ill-posed problems, such as being highly sensitive to high-frequency perturbations and so we refer to them as (discrete) ill-posed problems.

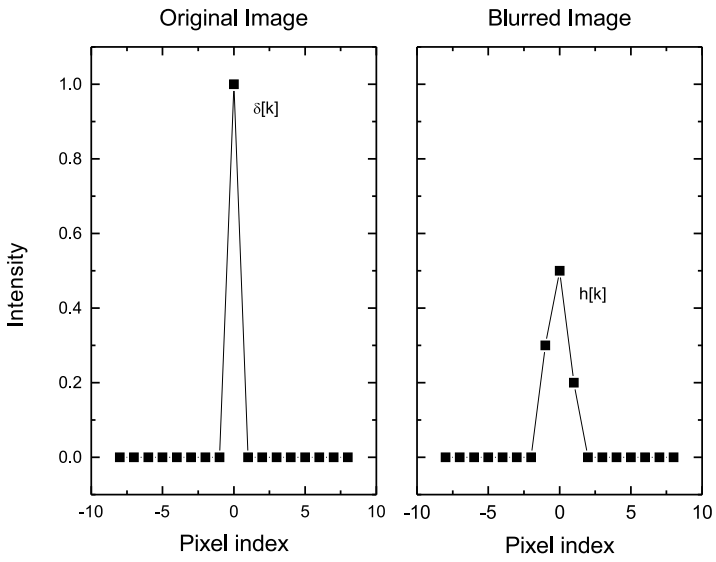
1.4 Case Study in Image Deblurring

Let us consider the deconvolution or deblurring problem. The desired quantity in this case is a sampled signal $x[k]$ evaluated on a grid of regularly-spaced times or an image, $x[k,l]$ represented its intensity values on a regular array of pixels. The quantity we can measure is a filtered version of this signal or a blurred version of the image. Given the measured data, how do we reconstruct the image?

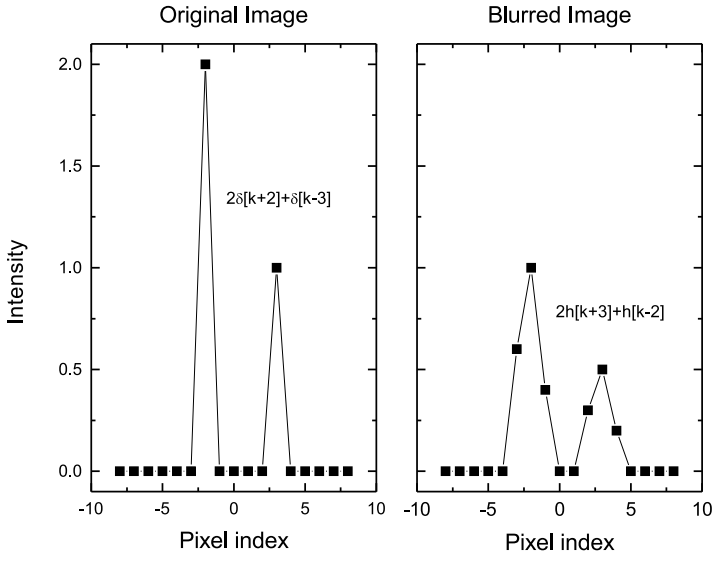
1.4.1 Modelling the forward problem

In this step we model how the image is turned into the measurements by the measurement process. For simplicity, we initially consider a discrete one-dimensional "image" which is simply a sequence of numbers representing intensity as a function of position. Suppose first that the image consists of a single bright point in the centre of a dark background. Mathematically, this is represented by a sequence $\delta[k]$ which is all zeros except at $k = 0$ where it takes on the value one. After the image is blurred, this single bright point becomes spread out into a region called the *point-spread function.* The graph below shows an example of a point spread function which is represented by the sequence $h[k]$.

[1]Students who have seen Hilbert-space methods for partial differential equations will recognize that the forward operator in this case is a Hilbert-Schmidt integral operator and hence is "compact". It follows that its inverse cannot be bounded.



If the blurring is *linear* and *spatially invariant,* an image which consists of two unequally bright points, one at $k = -2$ and the other at $k = 3$ which may be represented by the sequence $2\delta[k+2] + \delta[k-3]$ will be blurred into $2h[k+2] + h[k-3]$ as shown in the diagram below



We may readily generalize this blurring process to see that if the original image $x[k]$ is a linear combination of shifted δ sequences

$$x[k] = \sum_m c_m \delta[k-m], \tag{1.6}$$

then the blurred image $y[k]$ will be the *same* linear combination of shifted point-spread functions, i.e.,

$$y[k] = \sum_m c_m h[k-m]. \tag{1.7}$$

We now consider the problem of determining the coefficients c_m. This is easy once we realize that the δ sequence in (1.6) collapses the summation so that $c_k = x[k]$. Thus we may write the blurred image (1.7) as

$$y[k] = \sum_m x[m]\, h[k-m], \tag{1.8}$$

which we recognize simply as the *convolution* of the sequences x and h denoted $x * h$. Thus the process of convolution corresponds to the simple idea of spatially-invariant blurring. In Matlab, the function `conv` calculates the convolution of two sequences. For sequences of finite length, the convolution of a sequence of length M and a sequence of length N yields a sequence of length $M + N - 1$.

In two dimensions (e.g., for a photograph), images and point spread functions are represented by sequences with two indices. For example $x[m,n]$ denotes the intensity of the pixel in row m and column n of an image. The convolutional relationship readily generalizes in this situation to

$$y[k,l] = \sum_m \sum_n x[m,n]\, h[k-m, l-n]. \tag{1.9}$$

and again we write $y = x * h$.

1.4.2 Transformation into Fourier space

The process of convolution of two sequences is a moderately complicated process involving many additions and multiplications. Recall from the theory of Fourier transforms that we defined the convolution of two functions of t as

$$(x * h)(t) = \int_{-\infty}^{\infty} x(\tau)\, h(t-\tau)\, \mathrm{d}\tau \tag{1.10}$$

and we discovered that if we write the Fourier transforms of $x(t)$ and $h(t)$ as $X(\nu)$ and $H(\nu)$ respectively, where for example

$$X(\nu) = \int_{-\infty}^{\infty} x(t) \exp(-\mathrm{j}2\pi\nu t)\, \mathrm{d}t, \tag{1.11}$$

then the Fourier transform of $(x * h)$ is simply the product $X(\nu) H(\nu)$. Thus the operation of convolution in the t domain corresponds to multiplication of functions in the Fourier domain. The mapping from the function x to the data is then $d = Ax + n$, where A is the convolution $Ax \equiv x * h$, and n is an unknown function representing noise. The Fourier transform of this relation gives the relation between Fourier transforms:

$$D(\nu) = X(\nu) H(\nu) + N(\nu). \tag{1.12}$$

If the $H(\nu)$ is never zero, i.e., the forward problem is invertible, a straightforward inversion scheme is to calculate an estimate of X by

$$\hat{X}(\nu) = \frac{D(\nu)}{H(\nu)} \tag{1.13}$$

and an estimate of the function x is $\hat{x}$, the inverse Fourier transform of $\hat{X}$.

We wish to see if we can carry out a similar process using sequences. Since we will want to represent these sequences on a computer, we shall consider sequences of finite length N. Given a sequence $x[k]$ of length N, we shall assume that the index k ranges from 0 to $N-1$.

The *finite Fourier transform* of a sequence of length N is defined to be another sequence of length N. We shall use the convention of denoting the Fourier transform by the upper-case letter corresponding to the lower-case letter of the original sequence. For example, the finite Fourier transform $X[r]$ of $x[k]$ is defined by

$$X[r] = \sum_{k=0}^{N-1} x[k] \exp\left(-\frac{\mathrm{j}2\pi rk}{N}\right) \qquad \text{for } r = 0, 1, ..., N-1. \tag{1.14}$$

With this definition, we can recover $x[k]$ from $X[r]$ by the *inverse finite Fourier transform*

$$x[k] = \frac{1}{N} \sum_{r=0}^{N-1} X[r] \exp\left(\frac{\mathrm{j}2\pi rk}{N}\right) \qquad \text{for } k = 0, 1, ..., N-1. \tag{1.15}$$

You should be able to show that these relationships are indeed inverses of each other by using the fact that

$$\sum_{k=0}^{N-1} \exp\left(\frac{\mathrm{j}2\pi rk}{N}\right) = \begin{cases} N & \text{if } r \bmod N = 0 \\ 0 & \text{otherwise} \end{cases}, \tag{1.16}$$

which may simply be demonstrated by summing the geometric series.

Note that the normalizations in front of the forward and inverse transforms are different. The above conforms to the convention adopted in Matlab. An alternative convention which has the advantage that $X[0]$ is the average of the sequence $x[k]$ is to place the factor $1/N$ in front of the forward transform and have a factor of unity in the inverse transform. Yet another convention which leads to a more symmetrical forward and inverse transform is to place a factor of $1/\sqrt{N}$ in front of both transforms.

Consider now the product of two finite Fourier transforms:

$$\begin{aligned} X[r]\, H[r] &= \sum_{k=0}^{N-1} x[k] \exp\left(-\frac{\mathrm{j}2\pi rk}{N}\right) \sum_{l=0}^{N-1} h[l] \exp\left(-\frac{\mathrm{j}2\pi rl}{N}\right) \\ &= \sum_{k=0}^{N-1} \left\{ \begin{array}{c} \sum_{l=0}^{N-1-k} x[k]\, h[l] \exp\left(-\frac{\mathrm{j}2\pi r(k+l)}{N}\right) \\ + \sum_{l=N-k}^{N-1} x[k]\, h[l] \exp\left(-\frac{\mathrm{j}2\pi r(k+l)}{N}\right) \end{array} \right\} \\ &= \sum_{k=0}^{N-1} \left\{ \begin{array}{c} \sum_{m=k}^{N-1} x[k]\, h[m-k] \exp\left(-\frac{\mathrm{j}2\pi rm}{N}\right) \\ + \sum_{m=0}^{k-1} x[k]\, h[m-k+N] \exp\left(-\frac{\mathrm{j}2\pi rm}{N}\right) \end{array} \right\} \\ &= \sum_{m=0}^{N-1} \left(\sum_{k=0}^{N-1} x[k]\, h[(m-k) \bmod N] \right) \exp\left(-\frac{\mathrm{j}2\pi rm}{N}\right) \end{aligned}$$

where the substitutions $l = m - k$ and $l = m - k + N$ were used to go from the second to third line. The term

$$(x \circledast h)[m] = \sum_{k=0}^{N-1} x[k]\, h[(m-k) \bmod N]$$

is the *circular convolution*[2] of the sequences x and h. Hence the product of finite Fourier transforms is the finite Fourier transform of the circular convolution of the original sequences.

[2]The circular reference comes from the observation that the sum can be calculated as the sum of the products of the two sequences *wrapped* onto a circle with h shifted round by the amount m with respect to x.

Note that the wrap-around nature of the sum in the circular convolution can be understood in terms of the continuous Fourier transform. Since the finite Fourier transform corresponds to the continuous Fourier transform when both the original function and the transform are sampled and periodically repeated, the wrap-around comes from *spatial aliasing* resulting from the sampling in (spatial) frequency. The circular convolution can be used to calculate the usual convolution by first zero-padding the sequences x and h to twice their original length to avoid the aliasing. Setting

$$x_p[k] = \begin{cases} x[k], & k = 1, 2, \ldots, N-1 \\ 0, & k = N, N+1, \ldots, 2N-1 \end{cases}$$

and similarly for h_p we can see that

$$X_p[r] H_p[r] = \sum_{m=0}^{2N-1} (x * h)[m] \exp\left(-\frac{\mathrm{j}2\pi rm}{2N}\right)$$

is the finite Fourier transform (of length $2N$) of the desired convolution. The convolution may then be found by taking the inverse transform.

In summary, the mapping from the sequence x to the data is then $d = Ax + n$, where A is the convolution with the point-spread function $Ax \equiv x * h$, and n is the unknown noise sequence. Equivalently, we have the the relation between finite Fourier transforms of the zero-padded sequences:

$$D_p[r] = X_p[r] H_p[r] + N_p[r].$$

If the sequence $H_p[r]$ is never zero then the forward problem is invertible and an inversion scheme, analagous to the continuous case, is to calculate an estimate of X_p by

$$\hat{X}_p[r] = \frac{D_p[r]}{H_p[r]}$$

and an estimate of the sequence x is the first half of the sequence $\hat{x}_p$ calculated as the inverse finite Fourier transform of $\hat{X}_p$. So, as in the continuous case, the forward mapping may be inverted by division in the Fourier domain. This process is often called *deconvolution* as it undoes the operation of the convolution with the point-spread function.

1.4.3 A Pictorial Example

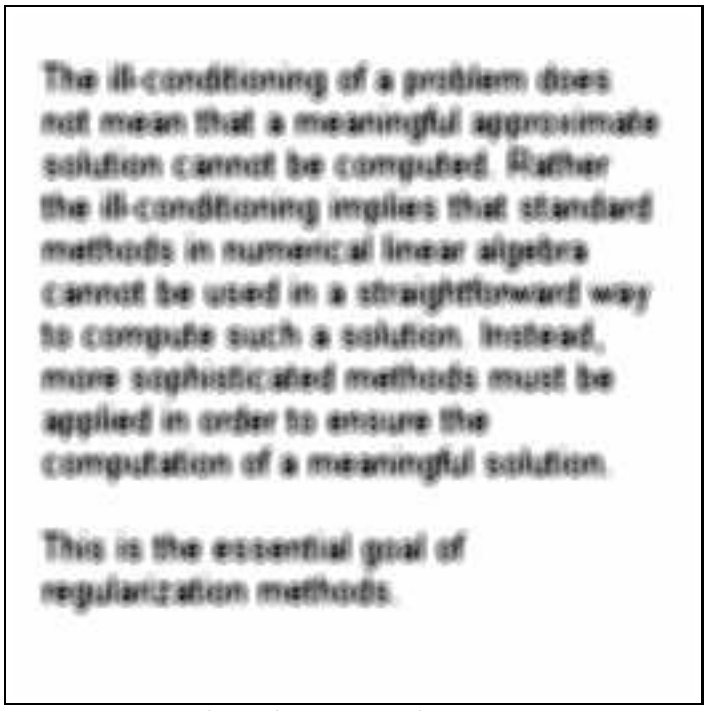
In the following example, an image of size 256×256 pixels is blurred with a point spread function which is a circle of radius $\sqrt{5}$ pixels. Random noise of r.m.s. amplitude about 1% of the maximum value is added to the blurred image and deconvolution is carried out by Fourier space division.

The original image has pixel values between 0 (background) and 255 (lettering). Usually the background in images is dark and the features in the image are brighter. However, in the following sequence of pictures, the value 0 is displayed as white so that the images are black on a white background; This has been done for ease of viewing and photocopying only.

Further, for the particular image and point-spread function chosen, no spatial aliasing takes place when the circular convolution is used to calculate the convolution representing the blurring in the forward process. This is because the original image has the value 0 for a distance around the edge of the image which is wider than $\sqrt{5}$ pixels, i.e., the half-width of

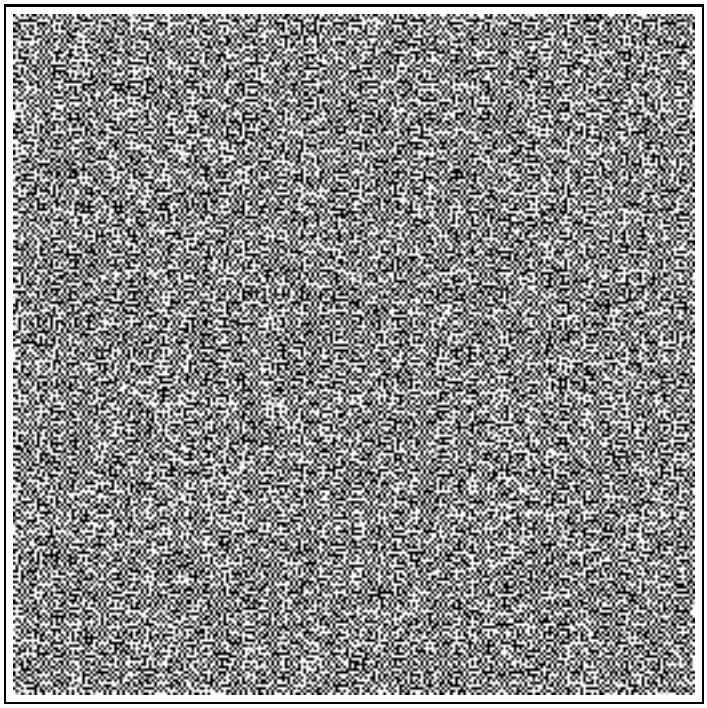
the point-spread function. Hence, again for ease of manipulation, the circular convolution is used to model the forward problem – without initial zero-padding.

The following picture shows the blurred image with noise added.



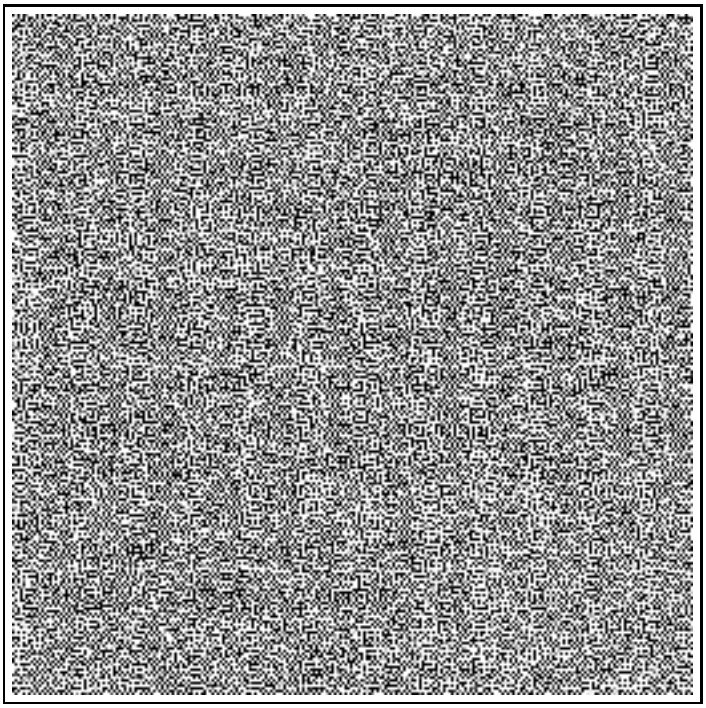
Blurred image with noise

It happens in this case that the forward mapping is invertible, so we can calculate the inverse image of the blurred image by division of the Fourier transforms. The result is shown in the following picture.



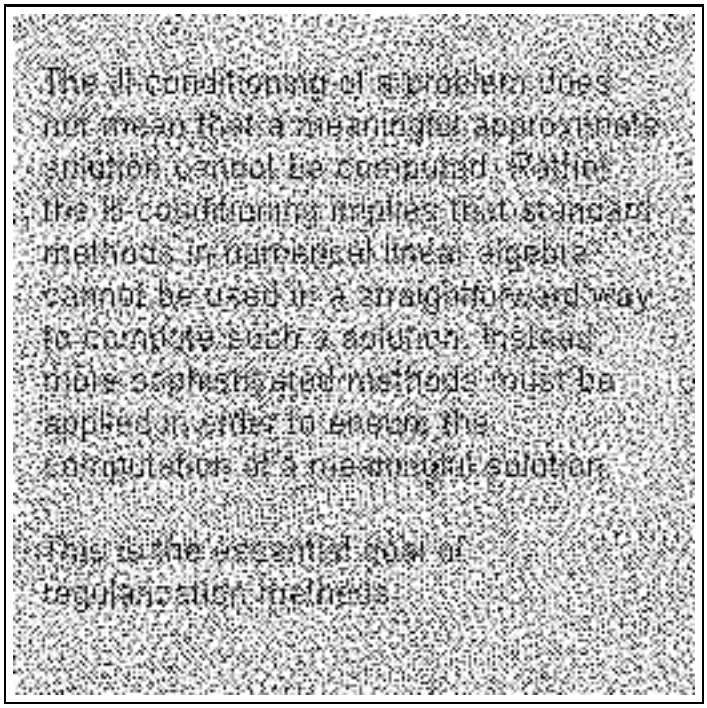
Inverse calculated by Fourier division, i.e.,
unregularized solution.

Note that the direct inverse does not give a useful reconstruction. One of the methods that we will investigate for reconstructing the image from noisey (and incomplete) data is 'regularization' in which we trade-off the accuracy of inverting the forward problem against a requirement that the reconstruction be 'smooth'. The details of that method are given later, but for the moment it is sufficient to know that the trade-off is controlled by the parameter λ; increasing λ makes the reconstruction smoother and the inversion more approximate. A small amount of regularization is achieved by setting $\lambda = 0.1$ and the result is the following picture.



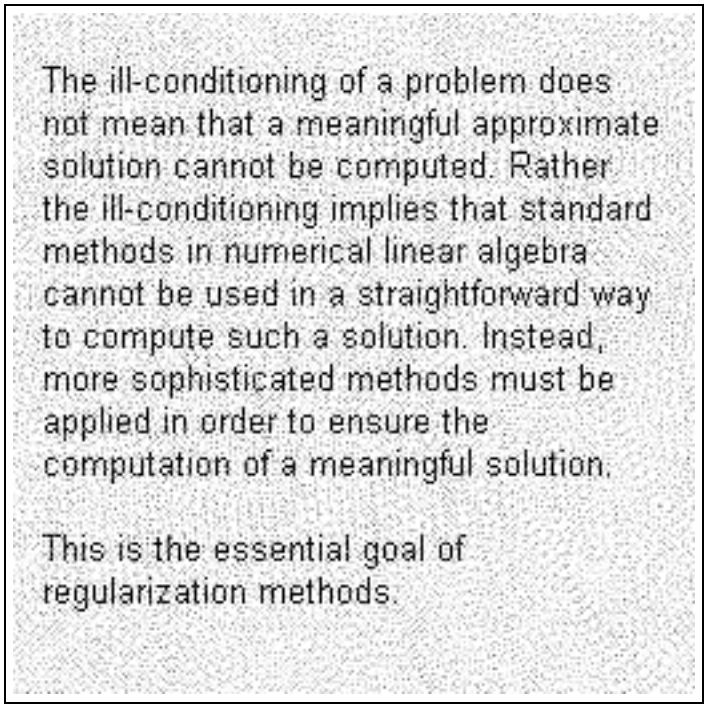
Regularized inverse image: $\lambda = 0.1$

That reconstruction is effectively the same as the first given by exact inversion, which is the case $\lambda = 0$ so there is no regularization. Greater regularizing is achieved by setting $\lambda = 1$,



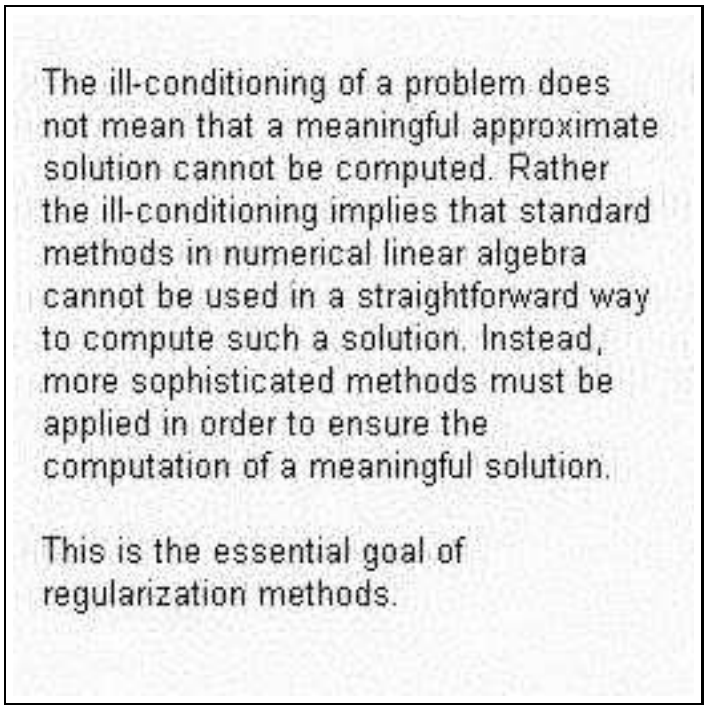
Regularized inverse image: $\lambda = 1$

and more still with $\lambda = 10$.



Regularized inverse image: $\lambda = 10$

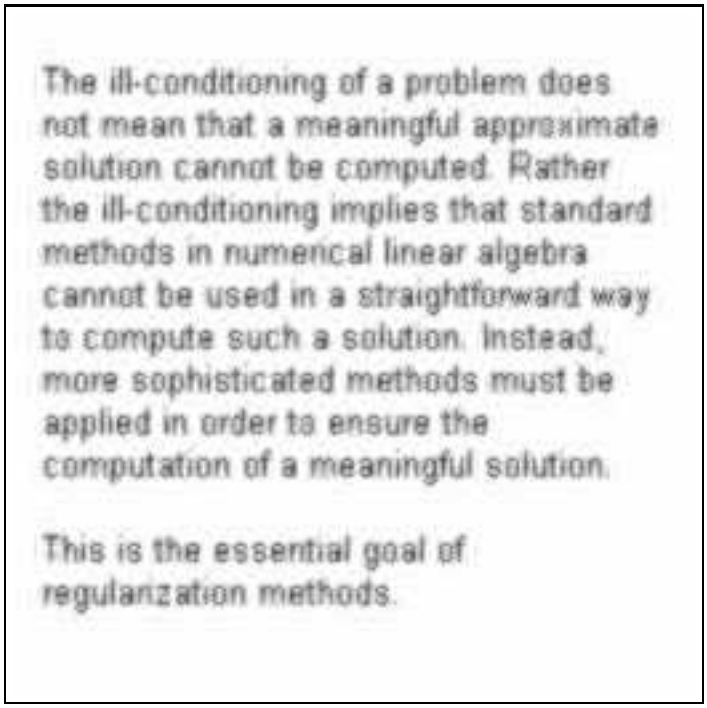
In terms of the L-curve, also covered later, $\lambda = 10$ is close to the optimum value. However, a bit clearer reconstruction is achieved using $\lambda = 100$.



Regularized inverse image: $\lambda = 100$

As you can see, the resulting reconstruction is perfectly readable. Note that this image has resulted from not inverting the forward mapping in the strict sense, but rather an approximation to the forward mapping. In doing so we have produced a result which is useful though not correct.

Regularizing further by setting $\lambda = 1000$ smooths more than is necessary and results in the following overly smooth reconstruction.



Regularized inverse image: $\lambda = 1000$

1.5 What Went Wrong with the Inverse?

By expressing the deblurring example of the previous section in the Fourier domain, the forward and inverse mapping were shown to have the simple structure of componentwise multiplication and division, respectively.

In the continuous image – continuous data case, the mapping from image to data was given in equation 1.12 as $D(\nu) = X(\nu) H(\nu) + N(\nu)$, so each Fourier component in the image is multiplied by a (complex) scalar and is corrupted by (complex) scalar valued noise to give the coefficient of a single Fourier component in the data. Similarly, the straightforward inverse of equation 1.13, $\hat{X}(\nu) = \frac{D(\nu)}{H(\nu)}$, is a scalar equation for each component. Combining these two equations we find that

$$\hat{X}(\nu) = \frac{X(\nu) H(\nu) + N(\nu)}{H(\nu)} = X(\nu) + \frac{N(\nu)}{H(\nu)},$$

i.e., the reconstruction is equal to the original image plus the term $\frac{N(\nu)}{H(\nu)}$. If $|H(\nu)|$ is small enough (or worse still zero) that term will be large, in fact larger than $X(\nu)$, and that component of the reconstruction will be principally determined by the unknown noise value $N(\nu)$. For the case of blurring, $H(\nu)$ necessarily has arbitrarily small (or zero) values whenever the point-spread function $h(t)$ is square integrable since then, by Parseval's identity, so is $H(\nu)$ and hence $H(\nu) \to 0$ as $\nu \to \infty$. Thus, high frequency components in the reconstruction will necessarily be determined by the details of the unknown noise. Note that the forward mapping for blurring of a finite-sized picture is a Fredholm integral equation and the observation of high-frequency difficulties in the inverse is exactly the same as reasoned in section 1.3.

Thus, the continuous image – continuous data deblurring problem is ill-posed because if the inverse exists it cannot be continuous. Of course, there will not be an inverse at all if $H(\nu)$ is zero for any values of ν (the division by $H(\nu)$ is undefined) in which case the deblurring problem is ill-posed because it fails the first condition for being well-posed. However, if $H(\nu)$ does take the value zero for some ν, and we are able to make *perfect noise-free measurements*, then the corresponding value of the data $D(\nu)$ is also zero and so we can find an inverse image of the data by choosing any value we please for $X(\nu)$. But then the inverse image is certainly not unique and the problem fails the second condition for being well-posed.

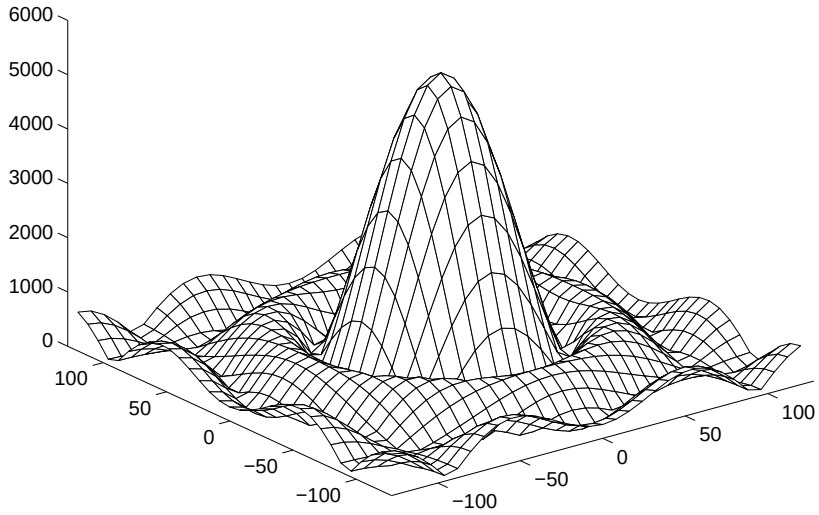
The discrete-discrete deblurring example has many of the same properties. Once again the inverse of the forward problem leads to a solution by Fourier division: $\hat{X}_p[r] = \frac{D_p[r]}{H_p[r]}$. In this case the Fourier transform $H_p[r]$ has a finite number of values and so, assuming the forward problem is invertible, $H_p[r]$ has a smallest non-zero value. However, if that value is small enough, the corresponding component in the reconstruction, given by

$$\hat{X}_p[r] = X_p[r] + \frac{N_p[r]}{H_p[r]},$$

is dominated by the noise, and the data effectively give no information about the corresponding component in the image.

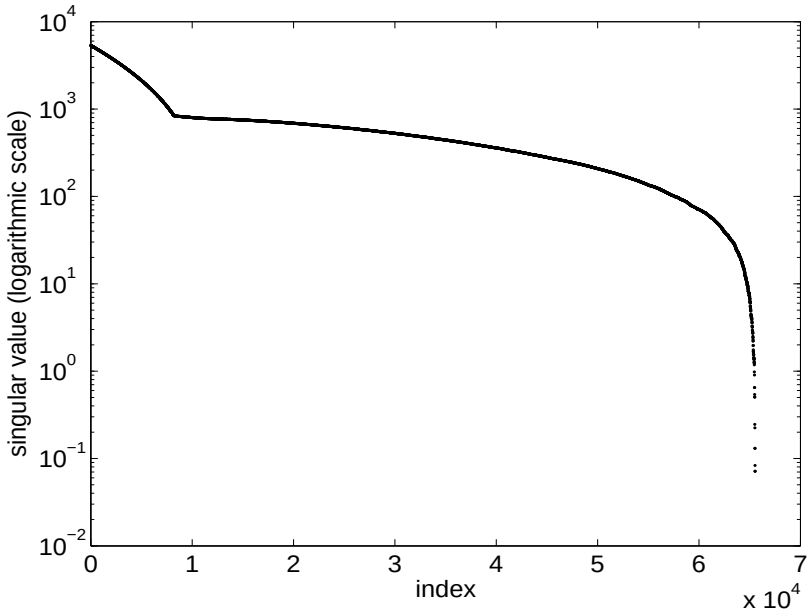
In the presence of noise, it makes no difference whether a value of $H_p[r]$, or $H(\nu)$, is very small or actually zero. In one case the forward mapping is invertible and in the other it is not, yet the data are effectively the same. We must conclude that the issue of invertibility of the forward problem, *per se*, is irrelevant when considering the practical inverse problem.

The blurring function used in the pictorial example was a circle of radius $\sqrt{5}$ pixels in a 256×256 pixels image. The magnitude of the finite Fourier transform is shown in the following figure.



Magnitude of finite Fourier transform of the point-spread function.

It is more usual to plot the singular values (see next chapter) of the forward map in descending order, on a logarithmic scale, as in the following figure.



Singular values of the forward map (which are the magnitude of finite Fourier transform of the point-spread function) plotted in descending order.

From the picture it can be seen the the Fourier transform does indeed have values that are close to zero and so we expect that inverse will show the symptoms of being ill-posed. The minimum and maximum magnitudes of the transform turn out to be 0.071 and 5355, respectively. So the forward problem is invertible, and strictly well-posed, but it is not well-conditioned with a condition number of 7.5×10^4 compared to the noise level of about 1%.

2

Linear Transformations

2.1 Introduction

We were able to understand the limitations of exact inversion in the deblurring example because, in the Fourier representation, the forward problem could be written as a set of uncoupled scalar equations and therefore consisted of componentwise multiplication. Each Fourier component in the image is multiplied by the corresponding value of the Fourier transform of the point-spread function and then has noise added to give the data. The 'inverse' of the forward problem is then componentwise division of the Fourier transform of the data by the appropriate scalar from the Fourier transform of the point-spread function; problems arose when the divisor had small magnitude. We would like to have a similar description of other linear inverse problems so that we can similarly predict when problems will arise..

In general, a linear forward problem will not necessarily be shift-invariant and a Fourier representation of image and data space will not lead to the forward mapping being a set of uncoupled scalar equations. Usually the size of image and data spaces are different (an extreme case is continuous-discrete problems) in which case the matrix representing the forward mapping is not even square. Yet, remarkably, in all these cases there is an analogue of Fourier space – given by the *singular value decomposition* (SVD) – in which the forward operator reduces to componentwise scalar multiplication and the straightforward inverse is by scalar division.

The simplest case to understand is when both image and data space are finite-dimensional, i.e., for discrete-discrete problems. We will develop the SVD for that case first and then go on to the case where image or data space is continuous.

2.2 A Linear Algebra Primer

Linear algebra forms the mathematical basis for the vector and matrix analysis that we use to analyze linear inverse problems where both image and data space is discrete. Furthermore, when viewed from a certain perspective to be explained in Section 2.11.1, it has a natural generalization to functional analysis, which is the basic tool in analysis of continuous inverse problems.

M-dimensional (real) vector space is denoted $\mathbb{R}^M$ and consists of the set of all ordered M-

tuples of real numbers, usually written as a column

$$\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_M \end{pmatrix}$$

so that the subscript denotes a *row* index. Addition of vectors is defined by $(\mathbf{u}+\mathbf{v})_k = u_k + v_k$ and scalar multiplication by $(c\mathbf{u})_k = cu_k$, $c \in \mathbb{R}$. We will sometimes write the scalar to the right of the vector: $\mathbf{u}c = c\mathbf{u}$. This will facilitate our "star" notation for resolutions of unity. In fact, we can view the expression $\mathbf{u}c$ as the matrix product of the column vector $\mathbf{u}$ ($M \times 1$ matrix) and the scalar c (1×1 matrix). This merely amounts to a change of point of view: Rather than thinking of c as multiplying $\mathbf{u}$ to obtain $c\mathbf{u}$, we can think of $\mathbf{u}$ as multiplying c to get $\mathbf{u}c$. The result is, of course, the same.

Similarly, the set of all column vectors with M complex entries is denoted by $\mathbb{C}^M$. $\mathbb{C}^M$ is a complex vector space, with vector addition and scalar multiplication defined exactly as for $\mathbb{R}^M$, the only difference being that now the scalar c may be complex. A subspace of a vector space V is a subset $S \subset V$ that is "closed" under vector addition and scalar multiplication. That is, the sum of any two vectors in S also belongs S, as does any scalar multiple of any vector in S. For example, any plane through the origin is a subspace of $\mathbb{R}^3$, whereas the unit sphere is not. $\mathbb{R}^M$ is a subspace of $\mathbb{C}^M$, provided we use only real scalars in the scalar multiplication. We therefore say that $\mathbb{R}^M$ is a real subspace of $\mathbb{C}^M$. It often turns out that even when analyzing real objects, complex methods are simpler than real methods. (For example, the complex exponential form of Fourier series is formally simpler than the real form using sines and cosines.)

A basis for $\mathbb{C}^M$ is a collection of vectors $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_N\}$ such that

1. *any* vector $\mathbf{u} \in \mathbb{C}^M$ can be written as a linear combination of the $\mathbf{b}_n$'s, i.e., $\mathbf{u} = \sum_n c_n \mathbf{b}_n$, and
2. there is just one set of coefficients $\{c_n\}$ for which this can be done.

The scalars c_n are called the *components* of $\mathbf{u}$ *with respect to the basis* $\{\mathbf{b}_n\}$, and they necessarily depend on the choice of basis. It can be shown that any basis for $\mathbb{C}^M$ has exactly M vectors, i.e., we must have $N = M$ above. If $N < M$, then some vectors in $\mathbb{C}^M$ cannot be expressed as linear combinations of the $\mathbf{b}_n$'s and we say that the collection $\{\mathbf{b}_n\}$ is *incomplete*. If $N > M$, then every vector can be expressed in an infinite number of different ways, hence the c_n's are not unique. We then say that the collection $\{\mathbf{b}_n\}$ is *overcomplete*. Condition (1) and (2), above, are equivalent to demanding that the M vectors in a basis are *complete*. Of course, not every collection of M vectors in $\mathbb{C}^M$ is a basis! In order to form a basis, a set of M vectors must be linearly independent, which means that no vector in it can be expressed as a linear combination of all the other vectors.

The standard basis $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_M\}$ in $\mathbb{C}^M$ (as well as in $\mathbb{R}^M$) is defined by

$$(\mathbf{e}_m)_k = \delta_{mk} \equiv \begin{cases} 1, & k = m \\ 0, & k \neq m \end{cases} .$$

in which δ_{mk} is the *Kronecker delta*.

2.2.1 Systems of linear equations

A function A from $\mathbb{C}^N$ to $\mathbb{C}^M$ is said to be linear if

$$A(c\mathbf{u}+\mathbf{v}) = cA(\mathbf{u}) + A(\mathbf{v}) \text{ for all } \mathbf{u},\mathbf{v} \in \mathbb{C}^N \text{ and all } c \in \mathbb{C}.$$

Thus A preserves the vector space structure of $\mathbb{C}^N$. When A is linear we write $A\mathbf{u}$ instead of $A(\mathbf{u})$ because linear operators act in a way similar to multiplication.

Now let $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_N\}$ be a basis for $\mathbb{C}^N$ and $\{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_N\}$ be a basis for $\mathbb{C}^M$. Any vector $\mathbf{u} \in \mathbb{C}^N$ can be written as $\mathbf{u} = \sum_{n=0}^{N} u_n \mathbf{b}_n$ ($\{u_n\}$ are the components of $\mathbf{u}$ in this basis) and so

$$A\mathbf{u} = A\sum_{n=0}^{N} u_n \mathbf{b}_n = \sum_{n=0}^{N} u_n (A\mathbf{b}_n).$$

Since each $A\mathbf{b}_n \in \mathbb{C}^M$ it can be written as $A\mathbf{b}_n = \sum_{m=0}^{M} a_{mn}\mathbf{d}_m$, for some complex numbers a_{mn}, and therefore the components of $A\mathbf{u}$ with respect to the basis $\{\mathbf{d}_m\}$ are

$$(A\mathbf{u})_m = \sum_{n=0}^{N} a_{mn} u_n.$$

Hence the linear function A, given a particular choice of bases, can be represented by the $M \times N$ matrix $[a_{mn}]$. Conversely, any $M \times N$ matrix defines a linear operator from $\mathbb{C}^N$ to $\mathbb{C}^M$ (or possibly $\mathbb{R}^N$ to $\mathbb{R}^M$ if the matrix is real). Note that the operator is basis independent whereas matrices are basis dependent[1].

The matrix is applied to a vector $\mathbf{f}$ in $\mathbb{R}^N$ by the process of matrix multiplication, e.g.,

$$\mathbf{d} = \mathbf{A}\mathbf{f}. \tag{2.1}$$

There are two common ways of interpreting this matrix product, each is useful in different circumstances. The one which we shall most commonly think about is where the matrix $\mathbf{A}$ is considered as a collection of n column vectors $\mathbf{c}_1$, $\mathbf{c}_2, ..., \mathbf{c}_n$ written next to each other as shown

$$\mathbf{A} = \begin{pmatrix} \vdots & \vdots & & \vdots \\ \mathbf{c}_1 & \mathbf{c}_2 & \cdots & \mathbf{c}_n \\ \vdots & \vdots & & \vdots \end{pmatrix}. \tag{2.2}$$

The product $\mathbf{Af}$ may then be seen as the calculation of a linear combination of the vectors $\mathbf{c}_1$, $\mathbf{c}_2, ..., \mathbf{c}_n$ with the components of $\mathbf{f}$ giving the coefficients.

$$\mathbf{A}\mathbf{f} = \begin{pmatrix} \vdots & \vdots & & \vdots \\ \mathbf{c}_1 & \mathbf{c}_2 & \cdots & \mathbf{c}_n \\ \vdots & \vdots & & \vdots \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix} \tag{2.3}$$

$$= f_1\mathbf{c}_1 + f_2\mathbf{c}_2 + \cdots + f_n\mathbf{c}_n. \tag{2.4}$$

[1] A simple example of a basis-independent object is a vector which we think of as a geometric arrow. The components of the vector depend on the basis we use to represent it, but the geometric object exists independent of our choice of basis. Operations on the vector such as 'double its length' or 'rotate about some axis by 45 degrees' are linear; their martix representation depends on the basis we choose to describe the space.

In this interpretation of the matrix vector product, we highlight the fact that the image of the linear transformation is spanned by the columns of the matrix $\mathbf{A}$. When we now consider the system of equations

$$\mathbf{Af} = \mathbf{d} \tag{2.5}$$

we are asking how we can synthesize the vector $\mathbf{d}$ by taking the appropriate linear combination (specified by the solution $\mathbf{f}$) of the columns of the matrix $\mathbf{A}$.

An alternative way of viewing the system of equations (2.5) is as a set of M simultaneous equations in N unknowns. From this point of view, we focus on the *rows* rather than the columns of $\mathbf{A}$ and write the matrix product as

$$\begin{pmatrix} \cdots & \mathbf{r}_1^{\mathrm{T}} & \cdots \\ \cdots & \mathbf{r}_2^{\mathrm{T}} & \cdots \\ & \vdots & \\ \cdots & \mathbf{r}_m^{\mathrm{T}} & \cdots \end{pmatrix} \mathbf{f} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} \tag{2.6}$$

where $\mathbf{r}_i^{\mathrm{T}}$ is the i^{th} row of the matrix $\mathbf{A}$. The i^{th} simultaneous equation that is to be satisfied is the inner product relationship $\mathbf{r}_i^{\mathrm{T}}\mathbf{x} = d_i$ which defines in hyperplane in $\mathbb{R}^n$. The solution process may be thought of as finding the intersection of all of these hyperplanes.

Traditionally, a *system* of equations $\mathbf{Af} = \mathbf{d}$ is said to be *overdetermined* if there is no solution and *underdetermined* if there are infinitely many solutions. In the theory of inverse problems, we are trying to reconstruct $\mathbf{f}$ from a measurement of $\mathbf{d}$. As we shall see below, however, we need to be concerned with more than just the system of equations for the *specific* measurement result $\mathbf{d}$. Due to measurement errors, the measured value of $\mathbf{d}$ is necessarily different from $\mathbf{Af}$, and it is necessary to obtain good reconstructions even in the presence of noise. These considerations make it necessary to study the properties of the *operator* $\mathbf{A}$ in more detail.

2.2.2 Inner Products

The standard inner product in $\mathbb{C}^N$ generalizes the dot product in $\mathbb{R}^N$ and is defined as

$$\langle \mathbf{u}, \mathbf{v} \rangle = \sum_{n=1}^{N} \bar{u}_n v_n, \qquad \mathbf{u}, \mathbf{v} \in \mathbb{C}^N$$

where u_n and v_n are the components of $\mathbf{u}$ and $\mathbf{v}$ with respect to the standard basis and $\bar{u}_n$ denotes the complex conjugate[2] of u_n. When $\mathbf{u}$ and $\mathbf{v}$ are real vectors, this reduces to the usual inner product in $\mathbb{R}^N$. The standard inner product in $\mathbb{C}^N$ between $\mathbf{u}$ and $\mathbf{v}$ can also be written in matrix form as $\mathbf{u}^{\mathrm{H}}\mathbf{v}$ where the superscript H denotes the Hermitian conjugate of the matrix. This reduces to $\mathbf{u}^{\mathrm{T}}\mathbf{v}$ for $\mathbb{R}^N$. Using this definition, the *norm* of $\mathbf{u}$ defined by

$$||\mathbf{u}|| = \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle} = \left(\sum_{n=1}^{N} |u_n|^2 \right)^{\frac{1}{2}} \tag{2.7}$$

is real and nonnegative, which is sensible. Thus the inner-product and the norm it induces satisfy the following important properties:

[2]We use the convention of conjugating the first variable which is common in physics. In mathematics the convention is to conjugate the second variable. Our choice has the benefit of making the "star" operator linear.

(P) *Positivity:* $||\mathbf{u}|| > 0$ for all $\mathbf{u} \neq \mathbf{0}$ in $\mathbb{C}^N$, and $||\mathbf{0}|| = 0$.

(H) *Hermiticity:* $\overline{\langle \mathbf{u}, \mathbf{v} \rangle} = \langle \mathbf{v}, \mathbf{u} \rangle$ for all $\mathbf{u}, \mathbf{v}$ in $\mathbb{C}^N$.

(L) *Linearity:* $\langle \mathbf{u}, c\mathbf{v} + \mathbf{w} \rangle = c \langle \mathbf{u}, \mathbf{v} \rangle + \langle \mathbf{u}, \mathbf{w} \rangle$ for all $\mathbf{u}, \mathbf{v}, \mathbf{w}$ in $\mathbb{C}^N$ and $c \in \mathbb{C}$.

Note that linearity only holds for the second argument. From (L) and (H) it follows that the inner product is *anti-linear* in the first argument, i.e.

$$\langle c\mathbf{u} + \mathbf{w}, \mathbf{v} \rangle = \bar{c} \langle \mathbf{u}, \mathbf{v} \rangle + \langle \mathbf{w}, \mathbf{v} \rangle .$$

Inner-products other than the standard one are often used, particularly when a non-standard inner-product is more natural. Such a circumstance occurs when the different coordinates correspond to quantities with different units or scales and we end up with the inner-product

$$\langle \mathbf{u}, \mathbf{v} \rangle = \sum_{n=1}^{N} \mu_n \bar{u}_n v_n,$$

where the μ_n are positive weights. This inner-product, like all others, also satisfies properties (P), (H), and (L).

While the norm is defined in terms of the inner-product, it is possible to recover the inner-product from the norm using the *polarization identity*:

$$\langle \mathbf{u}, \mathbf{v} \rangle = \frac{1}{4} \left(||\mathbf{u} + \mathbf{v}||^2 - ||\mathbf{u} - \mathbf{v}||^2 \right) + \frac{1}{4\mathrm{i}} \left(||\mathbf{u} + \mathrm{i}\mathbf{v}||^2 - ||\mathbf{u} - \mathrm{i}\mathbf{v}||^2 \right) . \tag{2.8}$$

For real vectors, the second term is zero and the identity reduces to $\langle \mathbf{u}, \mathbf{v} \rangle = \frac{1}{4} \left(||\mathbf{u} + \mathbf{v}||^2 - ||\mathbf{u} - \mathbf{v}||^2 \right)$.

Vectors $\mathbf{u}, \mathbf{v}$ in $\mathbb{C}^N$ are said to be *orthogonal* (w.r.t. the inner-product) if $\langle \mathbf{u}, \mathbf{v} \rangle = 0$. A basis $\{\mathbf{b}_n\}$ is *orthonormal* (w.r.t. the inner-product) if $\langle \mathbf{b}_k, \mathbf{b}_l \rangle = \delta_{kl}$. Hence the concept of orthogonality is relative to the choice of inner-product, as is the concept of length. The standard inner-product has the useful property that the standard basis is orthonormal with respect to that inner-product.

2.2.3 Resolution of the Identity

Given an **orthonormal** basis $\{\mathbf{b}_k\}_{k=1}^N$ with respect to some inner product, the **projection** of a vector $\mathbf{v}$ along the basis vector $\mathbf{b}_k$ is given by:

$$\mathbf{b}_k \, \langle \mathbf{b}_k, \mathbf{v} \rangle \tag{2.9}$$

This is a vector of length $\langle \mathbf{b}_k, \mathbf{v} \rangle$ in the direction of the unit vector $\mathbf{b}_k$.

If the projections of $\mathbf{v}$ along all of the basis vectors are added together, the result must be equal to $\mathbf{v}$, since the basis vectors span the entire space, i.e.,

$$\mathbf{v} = \sum_{k=1}^{N} \mathbf{b}_k \, \langle \mathbf{b}_k, \mathbf{v} \rangle \tag{2.10}$$

For a basis $\{\mathbf{b}_k\}_{k=1}^N$ which is not necessarily orthonormal, it is still possible to write any vector $\mathbf{v}$ as a linear combination of the basis vectors $\mathbf{b}_k$, but the coefficients are not simply given by inner products of the form $\langle \mathbf{b}_k, \mathbf{v} \rangle$.

If we now consider the standard inner product on $\mathbb{C}^N$, the projection of $\mathbf{v}$ on $\mathbf{b}_k$ may be written as $\mathbf{b}_k \left(\mathbf{b}_k^{\mathrm{H}} \mathbf{v} \right)$, and

$$\mathbf{v} = \sum_{k=1}^{N} \mathbf{b}_k \left(\mathbf{b}_k^{\mathrm{H}} \mathbf{v} \right) = \left(\sum_{k=1}^{N} \mathbf{b}_k \, \mathbf{b}_k^{\mathrm{H}} \right) \mathbf{v}. \tag{2.11}$$

Since this is true for all vectors $\mathbf{v}$,

$$\mathbf{I} = \sum_{k=1}^{N} \mathbf{b}_k \, \mathbf{b}_k^{\mathrm{H}} \tag{2.12}$$

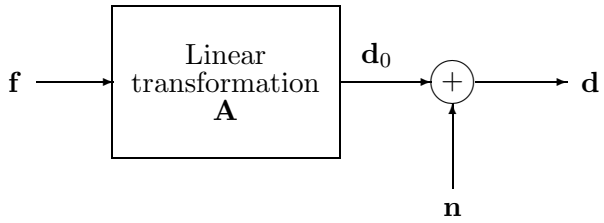
where $\mathbf{I}$ is the $N \times N$ identity matrix. Note that since $\mathbf{b}_k$ is a column N-vector, $\mathbf{b}_k \, \mathbf{b}_k^{\mathrm{H}}$ is an $N \times N$ matrix for each k, called the outer product of $\mathbf{b}_k$ with itself. The representation of $\mathbf{I}$ in terms of the outer products of vectors in an orthonormal basis is called a *resolution of the identity*. The matrix $\mathbf{b}_k \, \mathbf{b}_k^{\mathrm{H}}$ is called a *projection operator* along $\mathbf{b}_k$.

2.3 The Linear Inverse Problem

We consider several specific examples which may be regarded as prototypical of the linear inverse problems which are encountered in practice. These should be kept in mind as we discuss linear transformations more abstractly in the subsequent sections. Schematically, all linear inverse problems may be represented by the block diagram shown. The input (or "image") $\mathbf{f}$ is the collection of quantities we wish to reconstruct and the output $\mathbf{d}$ are the data we measure. In a linear inverse problem, the relationship between $\mathbf{f}$ and $\mathbf{d}$ is

$$\mathbf{d} = \mathbf{A}\mathbf{f} + \mathbf{n} \tag{2.13}$$

where $\mathbf{A}$ is a *linear* transformation, and $\mathbf{n}$ represents an *additive* noise process which prevents us from knowing the noise-free data $\mathbf{y} = \mathbf{A}\mathbf{f}$ precisely. Different applications give rise to image and data spaces of different sizes, but we shall wish to present a unified treatment of all of these.



2.3.1 Model Fitting to Data

In the simplest case, we measure a collection of data $\{(x_i, y_i)\}_{i=1}^m$ and attempt to fit a straight line $y = f_0 + f_1 x$ to the data, where f_1 is the gradient and f_0 is the intercept. The dimension

of the "image" space is only two, but the data space is m dimensional. In terms of the general picture, we have

$$\underbrace{\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}}_{\mathbf{d}} = \underbrace{\begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_m \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} f_0 \\ f_1 \end{pmatrix}}_{\mathbf{f}} + \underbrace{\begin{pmatrix} n_1 \\ n_2 \\ \vdots \\ n_m \end{pmatrix}}_{\mathbf{n}} \tag{2.14}$$

The noise $\mathbf{n}$ here represents the uncertainty in our measurements of y_i. We normally would make $m \gg 2$ measurements, and find (unless we are very lucky) that the points do not lie exactly on any straight line. Nevertheless, we usually "solve" for f_0 and f_1 by finding the line which is best in the least-squares sense. i.e., we find $\hat{\mathbf{f}}$ so as to minimize the value of $\left\|\mathbf{d} - \mathbf{A}\hat{\mathbf{f}}\right\|^2$. So long that $m \geq 2$, and that $x_1 \neq x_2$ there is a unique solution for $\hat{\mathbf{f}}$.

Instead of a straight line, we may fit a polynomial of the form $y = f_0 + f_1 x + \cdots + f_n x^n$ to the data. Even more generally, we need not use powers of x, but may be given a collection of n functions $g_1(x), g_2(x), \ldots, g_n(x)$ and be required to fit $y = f_1 g_1(x) + f_2 g_2(x) + \cdots + f_n g_n(x)$. In this case, we write

$$\underbrace{\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}}_{\mathbf{d}} = \underbrace{\begin{pmatrix} g_1(x_1) & g_2(x_1) & \cdots & g_n(x_1) \\ g_1(x_2) & g_2(x_2) & \cdots & g_n(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ g_1(x_m) & g_2(x_m) & \cdots & g_n(x_m) \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} f_1 \\ f_1 \\ \vdots \\ f_n \end{pmatrix}}_{\mathbf{f}} + \underbrace{\begin{pmatrix} n_1 \\ n_2 \\ \vdots \\ n_m \end{pmatrix}}_{\mathbf{n}} \tag{2.15}$$

Once again, we would normally make $m \gg n$ measurements in order to find the values of $f_1, \ldots, f_n$. The situation in which there are much more data than "image" parameters to fit is generally called "model-fitting". It is usually the case in model fitting that the model is not a perfect fit to the data for any choice of the parameters, and we have to use some criterion such as least squares to obtain a "best" solution. It is also usually the case that there is a unique best solution in the least squares sense. One can see how things can go wrong, if for example the functions $g_1(x), \ldots, g_n(x)$ are poorly chosen. For example, if the functions are not linearly independent, there may be an infinite number of choices of $\mathbf{f}$, all of which give the same value of $\mathbf{Af}$. In such a situation, solutions will not be unique. In the following we shall consider more precisely the conditions under which solutions exist and to what extent they are unique.

2.3.2 Indirect Imaging

In model fitting, the size of the "image" is taken to be much smaller than the size of the data. If this is not the case, the parameters of the model are not well-determined by the data. For example, if we are fitting a parabola to data which consist only of two points, there are an infinite number of parabolas which pass exactly through the data. There are however problems in which the size of the image space is comparable to or larger than the size of data space. This situation often occurs in practice because images are often functions of continuous variables such as time or (physical) space, so that the dimensionality of image space is infinite. On the other hand, only a finite number of data can be measured, so that data space is necessarily finite dimensional. An example of a problem of this type is that of

image deconvolution discussed in the previous chapter. The true image is actually a function defined on a plane and hence lies in an infinite dimensional space, but we made the space finite dimensional by discretizing the image into pixels. As discussed previously, the data consist of linear combinations of the values in the image. In the example we chose to discretize the blurred picture (the data) so that it has the same number of points as the image. This makes it convenient from the point of view of using discrete Fourier transforms, but there is no fundamental reason for doing this. As a second example, consider again the problem of X-ray tomography, in which line integrals through the patient are recorded at various angles in order to reconstruct the distribution of absorption within the body. The dimension of the image space is again infinite, but can be made finite by suitable discretizing the cross sectional area of the patient. The dimension of the data space depends on the number of angles used, and the number of points at which the X-ray intensity is measured.

The fineness of the discretization of the image is to some extent arbitrary, so long as it is fine enough that details can be represented adequately. As we refine the discretization, the number of image points increases, and at some stage, there will be an infinite number of images which will all map to exactly the same data under the linear transformation. (This will certainly happen once the number of image points exceeds the number of data points, but it can also happen well before that). Choosing a good reconstruction from among the infinity of possibilities then becomes an issue, since we would not like the appearance of the solution to change markedly as the discretization is refined beyond a certain point.

We shall refer to problems in which the number of image points is comparable to or greater than the number of data points as "indirect imaging." The distinction between model fitting and indirect imaging is qualitative rather than quantitative and we shall thus study them as special cases of the linear inverse problem.

2.4 Anatomy of a linear transformation

One way to treat the system of equations

$$\mathbf{d} = \mathbf{A}\mathbf{f} \tag{2.16}$$

where $\mathbf{A}$ is rectangular (and hence certainly not invertible) is to consider the operator $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ (or $\mathbf{A}\mathbf{A}^{\mathrm{T}}$) which is square and potentially invertible. This idea of analyzing the properties of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ (or $\mathbf{A}\mathbf{A}^{\mathrm{T}}$) in order to give information about $\mathbf{A}$ leads to a very important way of characterizing the behaviour of any finite dimensional linear transformation called the *singular value decomposition.* We shall investigate this in greater detail after a preliminary review of the eigenspace properties of real symmetric matrices. The advantage of first considering symmetric (and thus square) matrices is that the linear transformation defined by such a matrix maps a space into itself, whereas the domain and range spaces of the transformation defined by a rectangular matrix are different.

2.4.1 Eigenvalues and eigenvectors of real symmetric matrices

In elementary courses on linear algebra, it is shown that a real, symmetric (and hence square) $m \times m$ matrix $\mathbf{M}$ always has *real* eigenvalues and the eigenvectors of such a matrix may always be chosen to form an *orthonormal basis* of $\mathbb{R}^m$. If the eigenvalues are denoted μ_i and the corresponding eigenvectors are denoted $\mathbf{u}_i$, the statement that $\mathbf{u}_i$ is an eigenvector

associated with eigenvalue μ_i may be written

$$\mathbf{M}\mathbf{u}_i = \mathbf{u}_i \mu_i. \tag{2.17}$$

If we now write the column vectors $\mathbf{u}_1, ..., \mathbf{u}_m$ next to each other to form the square matrix

$$\mathbf{U} = \begin{pmatrix} \vdots & \vdots & & \vdots \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_m \\ \vdots & \vdots & & \vdots \end{pmatrix}, \tag{2.18}$$

the relations (2.17) for $i = 1, ..., m$ may be written as

$$\begin{aligned} \mathbf{M}\mathbf{U} &= \begin{pmatrix} \vdots & \vdots & & \vdots \\ \mathbf{M}\mathbf{u}_1 & \mathbf{M}\mathbf{u}_2 & \cdots & \mathbf{M}\mathbf{u}_m \\ \vdots & \vdots & & \vdots \end{pmatrix} = \begin{pmatrix} \vdots & \vdots & & \vdots \\ \mu_1\mathbf{u}_1 & \mu_2\mathbf{u}_2 & \cdots & \mu_m\mathbf{u}_m \\ \vdots & \vdots & & \vdots \end{pmatrix} \\ &= \begin{pmatrix} \vdots & \vdots & & \vdots \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_m \\ \vdots & \vdots & & \vdots \end{pmatrix} \begin{pmatrix} \mu_1 & & & \\ & \mu_2 & & \\ & & \ddots & \\ & & & \mu_m \end{pmatrix} = \mathbf{U}\mathbf{D}, \end{aligned} \tag{2.19}$$

where $\mathbf{D}$ is the diagonal matrix with the eigenvalues on the diagonal. Since $\mathbf{U}$ is an orthogonal matrix, it is invertible and so

$$\mathbf{M} = \mathbf{U}\mathbf{D}\mathbf{U}^{-1} = \mathbf{U}\mathbf{D}\mathbf{U}^{\mathrm{T}}. \tag{2.20}$$

Note that the identity $\mathbf{U}^{-1} = \mathbf{U}^{\mathrm{T}}$ follows from the columns of $\mathbf{U}$ being orthonormal; such a matrix is called *unitary*. The decomposition, in equation 2.20, of the original matrix $\mathbf{M}$ in terms of its eigenvectors may also be written as

$$\mathbf{M} = \sum_{k=1}^{m} \mu_k \mathbf{u}_k \mathbf{u}_k^{\mathrm{T}} \tag{2.21}$$

in which we are highlighting the representation of $\mathbf{M}$ as the weighted sum of outer-products of its eigenvectors. We can verify that the right hand side is indeed equal to $\mathbf{M}$ by considering its action on the eigenvectors $\mathbf{u}_i$. Since these eigenvectors form a basis, a verification which shows that they are transformed correctly ensures (by linearity) that all vectors are also transformed correctly.

The decomposition (2.21) means that the action of the real symmetric matrix $\mathbf{M}$ on an input vector $\mathbf{x} \in \mathbb{R}^m$ may be understood in terms of three steps:

1. It resolves the input vector along each of the eigenvectors $\mathbf{u}_k$, the component of the input vector along the i^{th} eigenvector being given by $\mathbf{u}_k^{\mathrm{T}}\mathbf{x}$,
2. The amount along the k^{th} eigenvector is multiplied by the eigenvalue μ_k,
3. The product tells us how much of the k^{th} eigenvector $\mathbf{u}_k$ is present in the product $\mathbf{M}\mathbf{x}$.

Thus, the eigenvectors of $\mathbf{M}$ define a "privileged" basis in which the action of $\mathbf{M}$ is particularly simple. Each of the components of $\mathbf{x}$ along the m eigenvectors is stretched *independently* by an amount given by the eigenvalue. Figure 2.1 is a schematic representation of the process. Only two of the m orthogonal eigenvectors are shown.

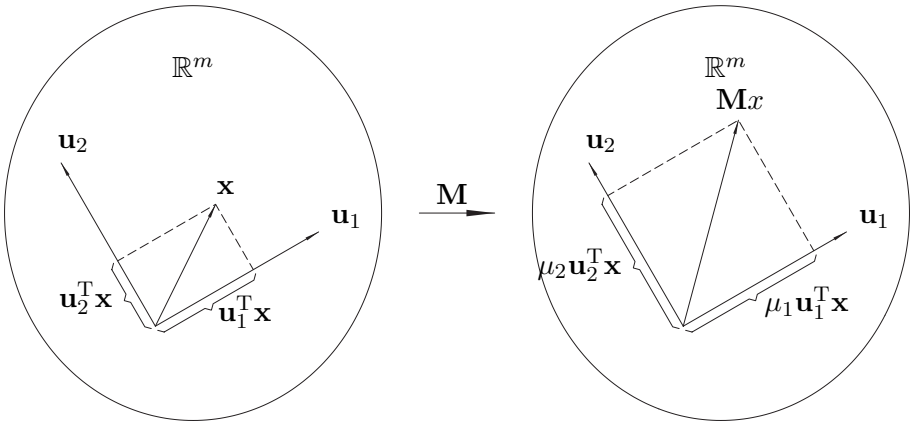


Figure 2.1: Effect of a real symmetric matrix $\mathbf{M}$ on a vector $\mathbf{x}$.

2.4.2 Functions of a real symmetric matrix

Since a real symmetric matrix $\mathbf{M}$ may be written as in (2.20), it is easy to compute any power of $\mathbf{M}$

$$\mathbf{M}^n = \left(\mathbf{U}\mathbf{D}\mathbf{U}^{\mathrm{T}}\right)^n = \mathbf{U}\mathbf{D}^n\mathbf{U}^{\mathrm{T}}, \tag{2.22}$$

since $\mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{U}\mathbf{U}^{\mathrm{T}} = \mathbf{I}$ as $\mathbf{U}$ is unitary. Since $\mathbf{D}$ is diagonal, raising $\mathbf{D}$ to the n^{th} power simply raises each of its (diagonal) elements to the n^{th} power. If we define arbitrary functions of a matrix in terms of the associated power series, we see that

$$f\left(\mathbf{M}\right) = \mathbf{U} f\left(\mathbf{D}\right) \mathbf{U}^{\mathrm{T}} \tag{2.23}$$

$$= \sum_{k=1}^{m} f\left(\mu_k\right) \mathbf{u}_k \mathbf{u}_k^{\mathrm{T}}. \tag{2.24}$$

In particular, the inverse of the matrix $\mathbf{M}$ is

$$\mathbf{M}^{-1} = \sum_{k=1}^{m} \frac{1}{\mu_k} \mathbf{u}_k \mathbf{u}_k^{\mathrm{T}}. \tag{2.25}$$

The matrix is invertible provided that no eigenvalue is equal to zero. The eigenvectors of $\mathbf{M}^{-1}$ are the same as those of $\mathbf{M}$, only the eigenvalues are reciprocated. Each direction which is stretched when $\mathbf{M}$ is applied contracted by $\mathbf{M}^{-1}$, and vice versa.

2.4.3 Singular value decomposition of a real rectangular matrix

Let us suppose that $\mathbf{A} \in \mathbb{R}^{m \times n}$ is a real rectangular matrix which maps vectors in $\mathbb{R}^n$ to vectors in $\mathbb{R}^m$. We may consider the two square symmetric matrices $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ and $\mathbf{A}\mathbf{A}^{\mathrm{T}}$ which may be formed from $\mathbf{A}$ and which are of size $n \times n$ and $m \times m$ respectively.

Since each of these matrices are square and symmetric, we may obtain the eigenvectors and eigenvalues of each. The eigenvectors may be chosen to form orthonormal bases of the respective spaces. We note that each of the matrices is *positive semidefinite*, which means that all their eigenvalues are non-negative. This is easy to see for if $\mathbf{v}$ is an eigenvector of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$, belonging to eigenvalue λ, then

$$\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{v} = \lambda \mathbf{v}. \tag{2.26}$$

Multiplying on the left by $\mathbf{v}^{\mathrm{T}}$ and grouping the terms,

$$\left(\mathbf{v}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\right)(\mathbf{A}\mathbf{v}) = \lambda\left(\mathbf{v}^{\mathrm{T}}\mathbf{v}\right). \tag{2.27}$$

On the left hand side we have a non-negative quantity, the square of the norm of $\mathbf{A}\mathbf{v}$. On the right, $\mathbf{v}^{\mathrm{T}}\mathbf{v}$ is positive and so λ must be non-negative.

Label the n orthonormal eigenvectors of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ as $\mathbf{v}_i$ with associated eigenvalues λ_i and assume that we have sorted them so that

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0. \tag{2.28}$$

Similarly, label the m orthonormal eigenvectors of $\mathbf{A}\mathbf{A}^{\mathrm{T}}$ as $\mathbf{u}_i$ with associated eigenvalues μ_i and sort them so that

$$\mu_1 \geq \mu_2 \geq \cdots \geq \mu_m \geq 0. \tag{2.29}$$

Consider the first eigenvector $\mathbf{v}_1$ of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ and suppose that λ_1 is not equal to zero. The vector $\mathbf{A}\mathbf{v}_1$ is then non-zero. We wish to show that $\mathbf{A}\mathbf{v}_1$ is in fact an eigenvector of $\mathbf{A}\mathbf{A}^{\mathrm{T}}$. To check this, we notice that

$$\left(\mathbf{A}\mathbf{A}^{\mathrm{T}}\right)(\mathbf{A}\mathbf{v}_1) = \mathbf{A}\left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)\mathbf{v}_1 = \lambda_1(\mathbf{A}\mathbf{v}_1). \tag{2.30}$$

This shows that $\mathbf{A}\mathbf{v}_1$ is indeed an eigenvector of $\mathbf{A}\mathbf{A}^{\mathrm{T}}$ which belongs to the eigenvalue λ_1. If we normalize $\mathbf{A}\mathbf{v}_1$ to have unit length by forming

$$\frac{\mathbf{A}\mathbf{v}_1}{\|\mathbf{A}\mathbf{v}_1\|}, \tag{2.31}$$

this is a normalized eigenvector of $\mathbf{A}\mathbf{A}^{\mathrm{T}}$ and so must be one of the $\mathbf{u}_i$ mentioned above provided that the eigenvalues of $\mathbf{A}\mathbf{A}^{\mathrm{T}}$ are not degenerate.

Continuing the above argument, we see that each non-zero eigenvalue λ_i of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ must also be an eigenvalue of $\mathbf{A}\mathbf{A}^{\mathrm{T}}$. A similar argument starting with an eigenvector $\mathbf{u}_i$ of $\mathbf{A}\mathbf{A}^{\mathrm{T}}$ belonging to a non-zero eigenvalue μ_i shows that the vector $\mathbf{A}^{\mathrm{T}}\mathbf{u}_1/\left\|\mathbf{A}^{\mathrm{T}}\mathbf{u}_1\right\|$ is a normalized eigenvector of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ with the same eigenvalue.

The conclusion to the above is that the non-zero eigenvalues of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ are the same as the non-zero eigenvalues of $\mathbf{A}\mathbf{A}^{\mathrm{T}}$ and vice versa. If there are r non-zero eigenvalues, this means that $\lambda_1 = \mu_1, ..., \lambda_r = \mu_r$ and that all subsequent eigenvalues must be zero, i.e., $\lambda_{r+1} = ... = \lambda_n = 0$ and that $\mu_{r+1} = ... = \mu_m = 0$.

It turns out to be possible to arrange that for all $k = 1, ..., r$,

$$\mathbf{u}_k = \frac{\mathbf{A}\mathbf{v}_k}{\|\mathbf{A}\mathbf{v}_k\|} \text{ and } \mathbf{v}_k = \frac{\mathbf{A}^{\mathrm{T}}\mathbf{u}_k}{\|\mathbf{A}^{\mathrm{T}}\mathbf{u}_k\|}. \tag{2.32}$$

This happens automatically if the non-zero eigenvalues of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ and $\mathbf{A}\mathbf{A}^{\mathrm{T}}$ are non-degenerate. Even if there are degeneracies, it is possible to choose the appropriate linear combinations in the degenerate eigenspaces so that these are true. The value of r is known as the *rank* of the matrix $\mathbf{A}$ (or of the matrix $\mathbf{A}^{\mathrm{T}}$). It should be clear that $r \leq m$ and $r \leq n$.

The norms in (2.32) may be evaluated,

$$\|\mathbf{A}\mathbf{v}_k\|^2 = (\mathbf{A}\mathbf{v}_k)^{\mathrm{T}}(\mathbf{A}\mathbf{v}_k) = \mathbf{v}_k^{\mathrm{T}}\left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)\mathbf{v}_k = \lambda_k, \tag{2.33}$$

where the last equality holds because $\mathbf{v}_k$ is an eigenvalue of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ belonging to λ_k, and because $\mathbf{v}_k$ is normalized so that $\mathbf{v}_k^{\mathrm{T}}\mathbf{v}_k = 1$. Similarly, $\left\|\mathbf{A}^{\mathrm{T}}\mathbf{u}_k\right\|^2 = \mu_k$. Since $\lambda_k = \mu_k > 0$, we may define σ_k to be the square root of the eigenvalue and write

$$\|\mathbf{A}\mathbf{v}_k\| = \left\|\mathbf{A}^{\mathrm{T}}\mathbf{u}_k\right\| = \sigma_k = \sqrt{\lambda_k} = \sqrt{\mu_k}, \tag{2.34}$$

for $k = 1, 2, ..., r$. Equation (2.32) then takes the simple form

$$\mathbf{A}\mathbf{v}_k = \sigma_k \mathbf{u}_k \tag{2.35}$$

$$\mathbf{A}^{\mathrm{T}}\mathbf{u}_k = \sigma_k \mathbf{v}_k. \tag{2.36}$$

The effect of the linear transformation $\mathbf{A}$ on the unit vector $\mathbf{v}_k \in \mathbb{R}^n$ is to take it to the vector $\sigma_k \mathbf{u}_k \in \mathbb{R}^m$ of length σ_k in the direction of the unit vector $\mathbf{u}_k \in \mathbb{R}^m$. The effect of the linear transformation $\mathbf{A}^{\mathrm{T}}$ on the unit vector $\mathbf{u}_k \in \mathbb{R}^m$ is to take it to the vector $\sigma_k \mathbf{v}_k \in \mathbb{R}^n$ of length σ_k in the direction of the unit vector $\mathbf{v}_k \in \mathbb{R}^n$.

On the other hand for $k > r$, the eigenvalue of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ associated with $\mathbf{v}_k$ is zero and so $\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{v}_k = 0$. Premultiplying this by $\mathbf{v}_k^{\mathrm{T}}$ shows that $\|\mathbf{A}\mathbf{v}_k\| = 0$ and hence that $\mathbf{A}\mathbf{v}_k = 0$. We thus have that

$$\mathbf{A}\mathbf{v}_k = 0 \text{ for } k = r+1, ..., n \tag{2.37}$$

and similarly,

$$\mathbf{A}^{\mathrm{T}}\mathbf{u}_k = 0 \text{ for } k = r+1, ..., m. \tag{2.38}$$

Equations (2.35) and (2.37) together describe how $\mathbf{A}$ acts on the vectors in the basis $\{\mathbf{v}_k\}$ for $k = 1, ..., n$. By linearity, any operator which has the same action as $\mathbf{A}$ on each of the vectors of the basis must be the same as $\mathbf{A}$. Thus we may write

$$\mathbf{A} = \sum_{k=1}^{r} \sigma_k \mathbf{u}_k \mathbf{v}_k^{\mathrm{T}}, \tag{2.39}$$

and it is easy to check (using the orthonormality of the basis $\{\mathbf{v}_k\}$) that the right hand side does have the same action as $\mathbf{A}$ on the basis.

Taking the transpose of (2.39) gives

$$\mathbf{A}^{\mathrm{T}} = \sum_{k=1}^{r} \sigma_k \mathbf{v}_k \mathbf{u}_k^{\mathrm{T}} \tag{2.40}$$

and it is again easy to check that this is consistent with (2.36) and (2.38).

The orthonormal vectors $\{\mathbf{v}_k\}$ are known as the *right singular vectors*, the vectors $\{\mathbf{u}_k\}$ are known as the *left singular vectors*, and the scalars $\{\sigma_k\}$ are called the *singular values* of the matrix $\mathbf{A}$.

We may write the column vectors $\mathbf{u}_k$ next to each other to form an orthogonal $m \times m$ matrix $\mathbf{U}$ and stack the row vectors $\mathbf{v}_k^{\mathrm{T}}$ on top of each other to form the orthogonal $n \times n$ matrix $\mathbf{V}^{\mathrm{T}}$. The equation (2.39) may then be written in matrix form as

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{T}}$$

where $\mathbf{S}$ is an $m \times n$ matrix whose only non-zero elements are the first r entries on the diagonal, i.e., $s_{kk} = \sigma_k$.

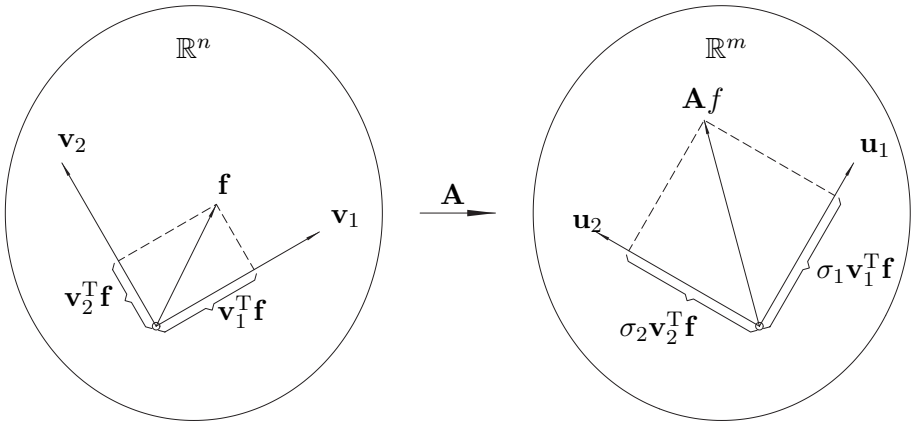


Figure 2.2: Effect of a rectangular matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ on a vector $\mathbf{f} \in \mathbb{R}^n$.

2.5 Interpretation of the singular value decomposition of a matrix

In this section, we discuss the interpretation of the singular value decomposition (2.39). When the matrix $\mathbf{A}$ acts on a vector $\mathbf{f}$, we may write the product as

$$\mathbf{A}\mathbf{f} = \sum_{k=1}^{r} \mathbf{u}_k \sigma_k \left(\mathbf{v}_k^{\mathrm{T}}\mathbf{f}\right), \tag{2.41}$$

this may again be understood as the sequence:

1. It resolves the input vector along each of the right singular vectors $\mathbf{v}_k$, the component of the input vector along the k^{th} singular vector being given by $\mathbf{v}_k^{\mathrm{T}}\mathbf{f}$,
2. The amount along the k^{th} direction is multiplied by the singular value σ_k,
3. The product tells us how much of the k^{th} left singular vector $\mathbf{u}_k$ is present in the product $\mathbf{A}\mathbf{f}$.

In effect the decomposition shows how a complicated operation such as matrix multiplication can be split into r *independent* multiplications, each of which takes a component along a vector in $\mathbb{R}^n$ and converts it into a component along a vector in $\mathbb{R}^m$. This result is all the more remarkable since $\{\mathbf{v}_k\}_{k=1}^r$ can be extended to an orthonormal basis $\{\mathbf{v}_k\}_{k=1}^n$ for $\mathbb{R}^n$ and $\{\mathbf{u}_k\}_{k=1}^r$ can be extended to an orthonormal basis $\{\mathbf{u}_k\}_{k=1}^m$ for $\mathbb{R}^m$. The action of $\mathbf{A}$ on a vector $\mathbf{f}$ is shown schematically in Figure 2.2. For convenience, we only show two of the n dimensions in the domain and two of the m dimensions in the range. This figure is rather similar to Figure 2.1, but notice that the bases in the two spaces are now *different*, even though they can both be chosen to be orthonormal.

The action of the transpose of $\mathbf{A}$ can also be worked out using the singular value decomposition. If $\mathbf{y} \in \mathbb{R}^m$, we see that,

$$\mathbf{A}^{\mathrm{T}}\mathbf{y} = \sum_{k=1}^{r} \mathbf{v}_k \sigma_k \left(\mathbf{u}_k^{\mathrm{T}}\mathbf{y}\right), \tag{2.42}$$

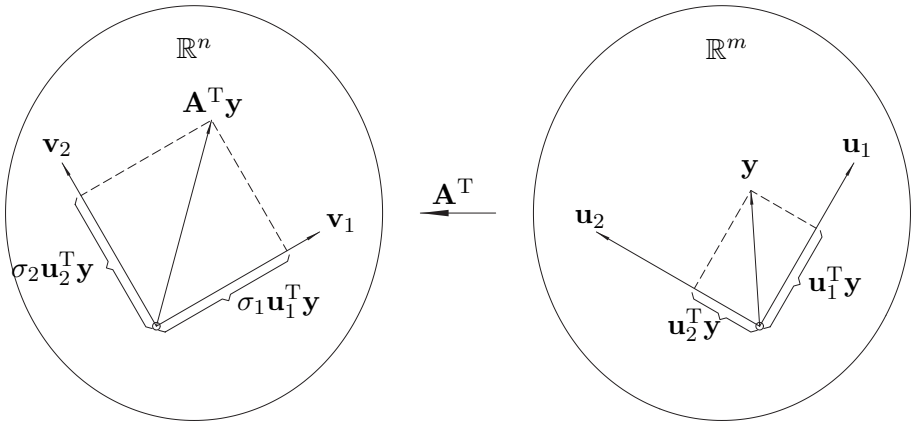


Figure 2.3: Effect of a rectangular matrix $\mathbf{A}^{\mathrm{T}} \in \mathbb{R}^{n\times m}$ on a vector $\mathbf{y} \in \mathbb{R}^m$.

which may be understood as the sequence:

1. Resolve the vector $\mathbf{y}$ along each of the left singular vectors $\mathbf{u}_k$, the component of the input vector along the k^{th} singular vector being given by $\mathbf{u}_k^{\mathrm{T}}\mathbf{y}$,
2. The amount along the k^{th} direction is multiplied by the singular value σ_k,
3. The product tells us how much of the k^{th} right singular vector $\mathbf{v}_k$ is present in the product $\mathbf{A}^{\mathrm{T}}\mathbf{y}$.

These steps are shown in Figure 2.3. Note that the singular values for $\mathbf{A}^{\mathrm{T}}$ are the same as those for $\mathbf{A}$.

2.6 Geometry of a linear transformation

Equations (2.35) and (2.38) tell us that the image of $\mathbf{A}$ is spanned by $\mathbf{u}_1, ..., \mathbf{u}_r$ and the null space of $\mathbf{A}^{\mathrm{T}}$ (i.e., the space which is mapped to zero under the action of $\mathbf{A}^{\mathrm{T}}$) is spanned by $\mathbf{u}_{r+1}, ..., \mathbf{u}_m$. Together they span all of $\mathbb{R}^m$, and we write $\mathbb{R}^m = \text{image}\,(\mathbf{A}) \oplus \text{null}\left(\mathbf{A}^{\mathrm{T}}\right)$. Similarly, from equations (2.37) and (2.36), the image of $\mathbf{A}^{\mathrm{T}}$ is spanned by $\mathbf{v}_1, ..., \mathbf{v}_r$ and the null space of $\mathbf{A}$ is spanned by $\mathbf{v}_{r+1}, ..., \mathbf{v}_n$. Together they span all of $\mathbb{R}^n$, and we write $\mathbb{R}^n = \text{image}\left(\mathbf{A}^{\mathrm{T}}\right) \oplus \text{null}\,(\mathbf{A})$. The symbol $\oplus$ denotes a *direct sum* of the spaces. If we write $C = A \oplus B$ where A, B and C are vector spaces, this means that any vector $\mathbf{c} \in C$ can be written in a unique way as the sum of a vector $\mathbf{a} \in A$ and a vector $\mathbf{b} \in B$. The above direct sums are also *orthogonal*, so that the vectors $\mathbf{a}$ and $\mathbf{b}$ are at right angles to each other. The dimensions satisfy $\dim C = \dim A + \dim B$.

It is convenient to visualize these relationships using the diagram of Figure 2.4. In the image space $\mathbb{R}^n$, there are r dimensions associated with the image of $\mathbf{A}^{\mathrm{T}}$ and $n - r$ dimensions associated with the null space of $\mathbf{A}$. Since we cannot draw more than two dimensions on a page, we represent each of these spaces by a single axis. Since the spaces are orthogonal, the axes are drawn at right angles to each other. Similarly, in data space $\mathbb{R}^m$, there are r dimensions associated with the image of $\mathbf{A}$ and $m - r$ dimensions associated with the null space of $\mathbf{A}^{\mathrm{T}}$. In the figure these are also represented schematically as two orthogonal axes.

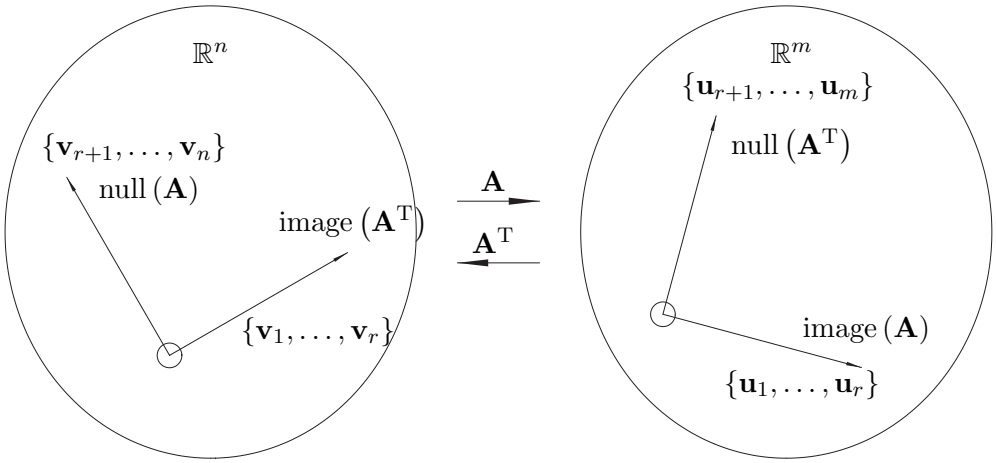


Figure 2.4: The image and null spaces of $\mathbf{A}$ and $\mathbf{A}^{\mathrm{T}}$ for a linear transformation of rank r.

The action of the linear transformation $\mathbf{A}$ is to map non-zero vectors in the space image $\left(\mathbf{A}^{\mathrm{T}}\right)$ into non-zero vectors in the space image $(\mathbf{A})$. All vectors which are orthogonal to image $\left(\mathbf{A}^{\mathrm{T}}\right)$ (i.e., those orthogonal to every *row* of $\mathbf{A}$) are necessarily in null $(\mathbf{A})$ and are mapped to zero under the action of $\mathbf{A}$.

Similarly, the action of the linear transformation $\mathbf{A}^{\mathrm{T}}$ is to map non-zero vectors in the space image $(\mathbf{A})$ into non-zero vectors in the space image $\left(\mathbf{A}^{\mathrm{T}}\right)$. All vectors which are orthogonal to image $(\mathbf{A})$ (i.e., those orthogonal to every *column* of $\mathbf{A}$) are necessarily in null $\left(\mathbf{A}^{\mathrm{T}}\right)$ and are mapped to zero under the action of $\mathbf{A}^{\mathrm{T}}$.

Let us now see how the singular value decomposition is useful in revealing how a linear transformation converts an "image" vector $\mathbf{f}$ into a "data" vector $\mathbf{Af}$. Any image vector $\mathbf{f}$ may be written as the sum of right singular vectors $\mathbf{v}_k$, the length of the component being given by the projection along the singular vector,

$$\mathbf{f} = \sum_{k=1}^{n} f_k \mathbf{v}_k \text{ where } f_k = \mathbf{v}_k^{\mathrm{T}} \mathbf{f}. \tag{2.43}$$

After this is converted into "data " by multiplication by $\mathbf{A}$, the result is

$$\mathbf{Af} = \sum_{k=1}^{r} (\sigma_k f_k)\, \mathbf{u}_k. \tag{2.44}$$

Information about the projection of $\mathbf{f}$ along $\mathbf{v}_k$ is "encoded" in the data as the component along the direction $\mathbf{u}_k$. The size of the projection is multiplied by σ_k to give the coefficient of $\mathbf{u}_k$, namely $\sigma_k f_k$. The singular value is the *factor* which tells us by how much each component defining the image is amplified or attenuated when it is converted into the data.

Notice that only the projections of the image $\mathbf{f}$ along the first r right singular vectors $\mathbf{v}_k$ play a role in determining the data $\mathbf{Af}$. If $r < n$, this means that the data are "blind" to certain aspects of the image: the data do not allow us to distinguish between images which have the same projections along the first r right singular vectors. Such images will look different, since they may differ in their projections along the remaining singular vectors. Equivalently, we may add to an image *any* element of the null space of $\mathbf{A}$ and the data will

not be changed in any way. If we now think of solving the inverse problem of reconstructing $\mathbf{f}$ from a measurement of $\mathbf{d}$, it is clear that at best, those components of $\mathbf{f}$ along the first r right singular vectors are determined by the data. However the data tell us nothing at all about the components of $\mathbf{f}$ along the remaining $n-r$ right singular vectors, and we have to use some other means for determining these components.

The above indicates that by calculating and plotting the right singular vectors, we can get an idea of what types of structure in the image will be visible in the data and also the types of structure which are invisible in the data.

2.7 The Singular Value Decomposition in Model Fitting Problems

In model-fitting problems, such as fitting of a straight line $y = f_0 + f_1 x$ to a collection of m data points $\{(x_k, y_k)\}_{k=1}^m$, the dimensionality of the image space is very low. The forward problem is

$$\underbrace{\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}}_{\mathbf{d}} = \underbrace{\begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_m \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} f_0 \\ f_1 \end{pmatrix}}_{\mathbf{f}}, \tag{2.45}$$

so image space is two-dimensional ($n = 2$), while data space is m dimensional. For model-fitting to give well-defined answers, the rank r of the matrix $\mathbf{A}$ is equal to n, so that the image of $\mathbf{A}$ is a two-dimensional subspace of $\mathbb{R}^m$. The data sets which lie in the image of $\mathbf{A}$ are those for which all the points lie exactly on some straight line. For most data sets, the points will not all lie on a straight line, and in these cases $\mathbf{d} \notin \text{image}(\mathbf{A})$. In the least-squares approach, the model parameters $\hat{\mathbf{f}}$ are chosen so that the $\mathbf{A}\hat{\mathbf{f}}$ is as close as possible to $\mathbf{d}$, i.e.,

$$\hat{\mathbf{f}} = \arg\min \|\mathbf{d} - \mathbf{A}\mathbf{f}\|^2 \tag{2.46}$$

Suppose we compute the singular value decomposition of $\mathbf{A}$, i.e., we find the left singular vectors $\{\mathbf{u}_k\}_{k=1}^m$, the right singular vectors $\{\mathbf{v}_k\}_{k=1}^n$ and the singular values σ_k such that:

$$\mathbf{A} = \sum_{k=1}^{r} \sigma_k \mathbf{u}_k \mathbf{v}_k^{\mathrm{T}} \tag{2.47}$$

Since $\{\mathbf{u}_k\}_{k=1}^m$ form a basis of data space, we may write the data $\mathbf{d}$ as a the linear combination:

$$\mathbf{d} = \sum_{k=1}^{m} \mathbf{u}_k \left(\mathbf{u}_k^{\mathrm{T}} \mathbf{d}\right). \tag{2.48}$$

Then, given any $\mathbf{f}$, we see that

$$\|\mathbf{d} - \mathbf{A}\mathbf{f}\|^2 = \left\| \sum_{k=1}^{m} \mathbf{u}_k \left(\mathbf{u}_k^{\mathrm{T}} \mathbf{d}\right) - \sum_{k=1}^{r} \sigma_k \mathbf{u}_k \left(\mathbf{v}_k^{\mathrm{T}} \mathbf{f}\right) \right\|^2 \tag{2.49}$$

$$= \left\| \sum_{k=1}^{r} \mathbf{u}_k \left\{\mathbf{u}_k^{\mathrm{T}} \mathbf{d} - \sigma_k \left(\mathbf{v}_k^{\mathrm{T}} \mathbf{f}\right)\right\} + \sum_{k=r+1}^{m} \mathbf{u}_k \left(\mathbf{u}_k^{\mathrm{T}} \mathbf{d}\right) \right\|^2. \tag{2.50}$$

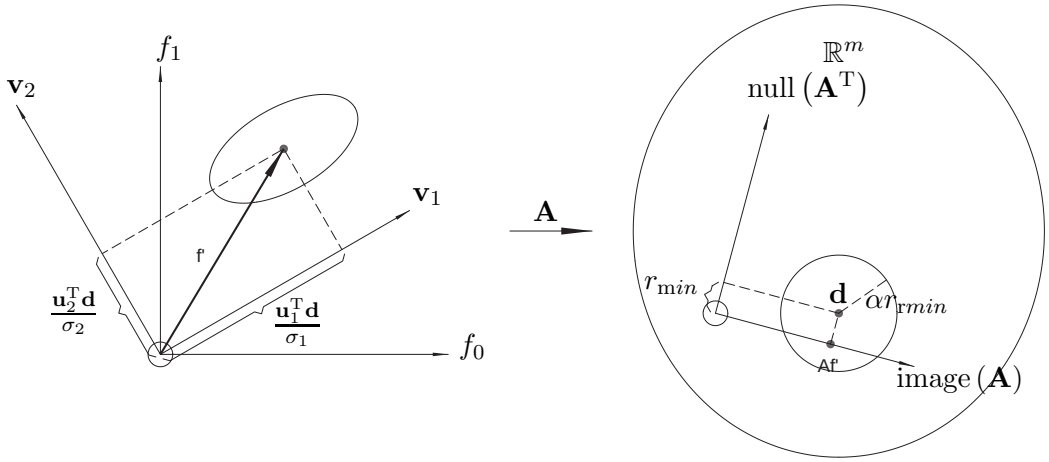


Figure 2.5: Geometry of a model-fitting problem

Using the theorem of Pythagorus (since the vectors $\{\mathbf{u}_k\}$ are orthogonal),

$$\|\mathbf{d}-\mathbf{A}\mathbf{f}\|^2 = \sum_{k=1}^{r} \left|\mathbf{u}_k^{\mathrm{T}}\mathbf{d} - \sigma_k\left(\mathbf{v}_k^{\mathrm{T}}\mathbf{f}\right)\right|^2 + \sum_{k=r+1}^{m} \left|\mathbf{u}_k^{\mathrm{T}}\mathbf{d}\right|^2. \tag{2.51}$$

Choosing $\hat{\mathbf{f}}$ so as to minimize $\|\mathbf{d}-\mathbf{A}\mathbf{f}\|^2$ is now straightforward. The second term on the right-hand side is the square of the perpendicular distance from $\mathbf{d}$ to the image of $\mathbf{A}$, and is completely unaffected by the choice of $\mathbf{f}$. The first term on the right hand side can be reduced to zero (its minimum possible value) by choosing $\hat{\mathbf{f}}$ such that

$$\mathbf{v}_k^{\mathrm{T}}\hat{\mathbf{f}} = \frac{\mathbf{u}_k^{\mathrm{T}}\mathbf{d}}{\sigma_k} \text{ for } k = 1, 2, \ldots, r. \tag{2.52}$$

Whether or not this completely determines $\hat{\mathbf{f}}$ depends on whether $r = n$ or $r < n$. For model fitting, $r = n$, and so the unique solution to the model fitting problem is:

$$\hat{\mathbf{f}} = \sum_{k=1}^{n} \mathbf{v}_k\left(\mathbf{v}_k^{\mathrm{T}}\hat{\mathbf{f}}\right) = \sum_{k=1}^{n} \mathbf{v}_k\left(\frac{\mathbf{u}_k^{\mathrm{T}}\mathbf{d}}{\sigma_k}\right) = \left(\sum_{k=1}^{n} \frac{1}{\sigma_k}\mathbf{v}_k\mathbf{u}_k^{\mathrm{T}}\right)\mathbf{d}. \tag{2.53}$$

This process is illustrated for the problem of fitting a straight line in Figure 2.5. Note that the two-dimensional image space is depicted in its entirety, but that image $(\mathbf{A})$, which is a two dimensional subspace in data space, is only depicted schematically by a line. The other $m-2$ dimensions in data space are also depicted as a line perpendicular to image $(\mathbf{A})$. The data $\mathbf{d}$ are shown as being slightly off image $(\mathbf{A})$ due to noise. The reconstucted model parameters $\hat{\mathbf{f}}$ are chosen so that $\mathbf{A}\hat{\mathbf{f}}$ is as close as possible to $\mathbf{d}$ in data space. The components of $\hat{\mathbf{f}}$ along the right singular vectors $\mathbf{v}_k$ are given by $\left(\mathbf{u}_k^{\mathrm{T}}\mathbf{d}\right)/\sigma_k$ as indicated by (2.53). The singular vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ are at right angles to each other, but may make an angle to the f_0 and f_1 axes which represent the intercept and gradient of the fitted straight line, respectively. The ellipse in image space depicts the uncertainty in the fitted parameter values, and will be discussed in a subsequent section.

2.7.1 Relationship to the Moore-Penrose inverse

We have already seen the Moore-Penrose inverse in the context of least-squares data fitting. In order to minimize the misfit $C = \|\mathbf{d} - \mathbf{A}\mathbf{f}\|^2$, we may write

$$C = \|\mathbf{d} - \mathbf{A}\mathbf{f}\|^2 = \sum_{k=1}^{m} \left(d_k - \sum_{l=1}^{n} a_{kl} f_l \right)^2 \tag{2.54}$$

Then

$$\frac{\partial C}{\partial f_i} = \sum_{k=1}^{m} 2 \left(d_k - \sum_{l=1}^{n} a_{kl} f_l \right) (-a_{ki}) = 0 \text{ for } i = 1, \ldots, n \tag{2.55}$$

for an extremum. This may be written as

$$\sum_{l=1}^{n} \left(\sum_{k=1}^{m} a_{ki} a_{kl} \right) f_l = \sum_{k=1}^{m} a_{ki} d_k, \tag{2.56}$$

which in matrix form is

$$\left(\mathbf{A}^{\mathrm{T}} \mathbf{A} \right) \mathbf{f} = \mathbf{A}^{\mathrm{T}} \mathbf{d}. \tag{2.57}$$

These are known as the *normal equations* of the least-squares problem. We obtain a unique solution provided that $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ is invertible, and find the best fit parameters $\hat{\mathbf{f}}$ using

$$\hat{\mathbf{f}} = \left(\mathbf{A}^{\mathrm{T}} \mathbf{A} \right)^{-1} \mathbf{A}^{\mathrm{T}} \mathbf{d} \tag{2.58}$$

The matrix $\left(\mathbf{A}^{\mathrm{T}} \mathbf{A} \right)^{-1} \mathbf{A}^{\mathrm{T}}$ is known as the Moore-Penrose inverse of $\mathbf{A}$.

It is easy to relate this to the singular-value decomposition. Using (2.39) and (2.40),

$$\mathbf{A}^{\mathrm{T}}\mathbf{A} = \left(\sum_{k=1}^{r} \sigma_k \mathbf{v}_k \mathbf{u}_k^{\mathrm{T}} \right) \left(\sum_{l=1}^{r} \sigma_l \mathbf{u}_l \mathbf{v}_l^{\mathrm{T}} \right) = \sum_{k=1}^{r} \sum_{l=1}^{r} \sigma_k \sigma_l \mathbf{v}_k \left(\mathbf{u}_k^{\mathrm{T}} \mathbf{u}_l \right) \mathbf{v}_l^{\mathrm{T}} = \sum_{k=1}^{r} \sigma_k^2 \mathbf{v}_k \mathbf{v}_k^{\mathrm{T}} \tag{2.59}$$

since $\mathbf{u}_k^{\mathrm{T}} \mathbf{u}_l = \delta_{kl}$. So $\left\{ \sigma_k^2 \right\}_{k=1}^{r}$ are the nonzero eigenvalues of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$. Since $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ is an $n \times n$ matrix, it is invertible iff $r = n$. If the matrix is invertible, then

$$\left(\mathbf{A}^{\mathrm{T}} \mathbf{A} \right)^{-1} = \sum_{k=1}^{r} \frac{1}{\sigma_k^2} \mathbf{v}_k \mathbf{v}_k^{\mathrm{T}}, \tag{2.60}$$

and

$$\left(\mathbf{A}^{\mathrm{T}} \mathbf{A} \right)^{-1} \mathbf{A}^{\mathrm{T}} = \left(\sum_{k=1}^{r} \frac{1}{\sigma_k^2} \mathbf{v}_k \mathbf{v}_k^{\mathrm{T}} \right) \left(\sum_{l=1}^{r} \sigma_l \mathbf{v}_l \mathbf{u}_l^{\mathrm{T}} \right) = \sum_{k=1}^{r} \frac{1}{\sigma_k} \mathbf{v}_k \mathbf{u}_k^{\mathrm{T}} \tag{2.61}$$

Comparing this with (2.53) shows that the least squares solution as calculated using the singular value decomposition is identical to that using the Moore Penrose inverse.

2.7.2 Effects of Noise on Model Parameter Estimates

It is instructive to consider how well the model parameters $\hat{\mathbf{f}}$ are determined in a least squares fitting procedure. The method is based on the idea that the value of $\|\mathbf{d} - \mathbf{A}\mathbf{f}\|^2$ is a measure

of how unlikely $\mathbf{f}$ is when the measured data are $\mathbf{d}$. The "best" parameter estimate $\hat{\mathbf{f}}$ is thus the one which miimizes $\|\mathbf{d}-\mathbf{A}\mathbf{f}\|^2$, i.e.,

$$\left\|\mathbf{d}-\mathbf{A}\hat{\mathbf{f}}\right\|^2 = \min_{\mathbf{f}\in\mathbb{R}^n}\left\|\mathbf{d}-\mathbf{A}\hat{\mathbf{f}}\right\|^2 = r_{\min}^2 \tag{2.62}$$

where $r_{\min}$ is the distance between the data $\mathbf{d}$ and the image of $\mathbf{A}$, (in which the data should lie, if noise were absent). The value of $r_{\min}$ allows us to estimate the amount of noise likely to be present. In order to get some idea of how confident we are about $\hat{\mathbf{f}}$, we consider the set of probable $\mathbf{f}$ values for the given data and noise level. This is known as the "feasible set",

$$F = \left\{\mathbf{f} : \|\mathbf{d}-\mathbf{A}\mathbf{f}\|^2 \le \alpha r_{\min}^2\right\}, \tag{2.63}$$

where α is chosen according to the confidence level required. We shall show that this set is an ellipse in image space centred about $\hat{\mathbf{f}}$ as shown in Figure.2.5.

Using the singular value decomposition of $\mathbf{A}$, we find from (2.51) that

$$\|\mathbf{d}-\mathbf{A}\mathbf{f}\|^2 = \sum_{k=1}^{r}\left|\mathbf{u}_k^{\mathrm{T}}\mathbf{d}-\sigma_k\left(\mathbf{v}_k^{\mathrm{T}}\mathbf{f}\right)\right|^2 + r_{\min}^2 \tag{2.64}$$

Substituting the values of $\mathbf{v}_k^{\mathrm{T}}\hat{\mathbf{f}}$ from (2.52),

$$\begin{aligned}\|\mathbf{d}-\mathbf{A}\mathbf{f}\|^2 &= r_{\min}^2 + \sum_{k=1}^{r}\left|\sigma_k\left(\mathbf{v}_k^{\mathrm{T}}\hat{\mathbf{f}}\right) - \sigma_k\left(\mathbf{v}_k^{\mathrm{T}}\mathbf{f}\right)\right|^2 \\ &= r_{\min}^2 + \sum_{k=1}^{r}\sigma_k^2\left|\mathbf{v}_k^{\mathrm{T}}\left(\mathbf{f}-\hat{\mathbf{f}}\right)\right|^2\end{aligned} \tag{2.65}$$

From the definition of the set F, we want those $\mathbf{f}$ which satisfy

$$\sum_{k=1}^{r}\sigma_k^2\left|\mathbf{v}_k^{\mathrm{T}}\left(\mathbf{f}-\hat{\mathbf{f}}\right)\right|^2 \le (\alpha-1)\, r_{\min}^2 \tag{2.66}$$

The boundary of this set is an ellipse centred at $\hat{\mathbf{f}}$ with principal axes aligned with the singular vectors $\mathbf{v}_k$. The length of the k'th principal semi-axis is

$$\frac{r_{\min}\sqrt{\alpha-1}}{\sigma_k}, \tag{2.67}$$

from which it is apparent that we are less confident about the parameters along the directions corresponding to small singular values.

As is apparent from Figure 2.5, the *independent* uncertainties along the singular vector directions $\mathbf{v}_k$ translate into *correlated* uncertaities along the axes f_0 and f_1 of the estimated parameters. We shall later consider in more detail how to quantify such correlated uncertainties which arise from parameter fitting.

2.8 The Singular Value Decomposition in General

As the number n of parameters in the model becomes large, it becomes difficult to ensure that they are all independent. Once there are dependent parameters, it becomes possible to achieve the same data vector from more than one set of parameters. Formally, the forward map is not one-one and $\exists \mathbf{f}_1, \mathbf{f}_2$ such that $\mathbf{A}\mathbf{f}_1 = \mathbf{A}\mathbf{f}_2$, but $\mathbf{f}_1 \neq \mathbf{f}_2$. As n becomes large, the problem of model fitting merges into that of indirect imaging. In the regime of indirect imaging, the number of points in image space is chosen to be so large that it can adequately represent the object of interest. Since there are now many images which map to the same data, it becomes necessary to choose from among them using an additional criterion of optimality.

In terms of the singular value decomposition, the rank r of the forward map $\mathbf{A}$ is less than n when the parameters of the model are not independent. This means that the null space of $\mathbf{A}$ contains some non-zero vectors. In fact, all vectors which are linear combinations of $\mathbf{v}_{r+1}, \ldots, \mathbf{v}_n$ are in the null space of $\mathbf{A}$, so that

$$\mathbf{A}\left(c_{r+1}\mathbf{v}_{r+1} + \cdots + c_n\mathbf{v}_n\right) = 0 \tag{2.68}$$

for every choice of coefficients $c_{r+1}, \ldots, c_n$.

Let us now consider what happens if we try to use the principle of least squares to reconstruct the image, when $r < n$. For measured data $\mathbf{d}$, and a trial image $\mathbf{f}$, (2.51) states that

$$\|\mathbf{d} - \mathbf{A}\mathbf{f}\|^2 = \sum_{k=1}^{r} \left|\mathbf{u}_k^{\mathrm{T}}\mathbf{d} - \sigma_k\left(\mathbf{v}_k^{\mathrm{T}}\mathbf{f}\right)\right|^2 + \sum_{k=r+1}^{m} \left|\mathbf{u}_k^{\mathrm{T}}\mathbf{d}\right|^2. \tag{2.69}$$

In order to minimize $\|\mathbf{d} - \mathbf{A}\mathbf{f}\|^2$ over all possible $\mathbf{f}$, the best that we can do is to ensure that the first term on the right hand side is equal to zero, i.e., that

$$\mathbf{v}_k^{\mathrm{T}}\mathbf{f} = \frac{\mathbf{u}_k^{\mathrm{T}}\mathbf{d}}{\sigma_k} \text{ for } k = 1, 2, \ldots, r. \tag{2.70}$$

If $r = n$, this completely defines $\mathbf{f}$, but if $r < n$, we see that only the projections of $\mathbf{f}$ along the first r right singular vectors $\mathbf{v}_1, \ldots, \mathbf{v}_r$ are determined. The projections on the remaining $n - r$ are completely arbitrary. Thus, instead of a single "best" solution $\hat{\mathbf{f}}$, all images of the form:

$$\sum_{k=1}^{r} \mathbf{v}_k \left(\frac{\mathbf{u}_k^{\mathrm{T}}\mathbf{d}}{\sigma_k}\right) + c_{r+1}\mathbf{v}_{r+1} + \cdots + c_n\mathbf{v}_n \tag{2.71}$$

for every choice of coefficients $c_{r+1}, \ldots, c_n$ will give the same minimum value for $\|\mathbf{d} - \mathbf{A}\mathbf{f}\|^2$. Needless to say, if some of the arbitrary coefficients are large, the reconstructed image will look terrible. As mentioned above, we need to select the arbitrary coefficients according to some other critereon of optimality, because the data do *not* determine them at all.

The situation is depicted schematically in Figure 2.6. If $r < m$, it is very likely that noise will cause the data $\mathbf{d}$ to lie outside the image of $\mathbf{A}$. If $\mathbf{y}$ is the point which is closest to $\mathbf{d}$ in image $(\mathbf{A})$, we find that an infinite number of images $\mathbf{f}$ map to the point $\mathbf{d}$. All such images differ by some vector in null $(\mathbf{A})$, and so the set S in image space which maps to $\mathbf{y}$ is represented schematically by a line parallel to the subspace null $(\mathbf{A})$. When we include the uncertainty in $\mathbf{d}$, as represented by the m-dimensional sphere about $\mathbf{d}$, we are led to consider the feasible set F (as defined by (2.63)) which maps to the intersection of the sphere and

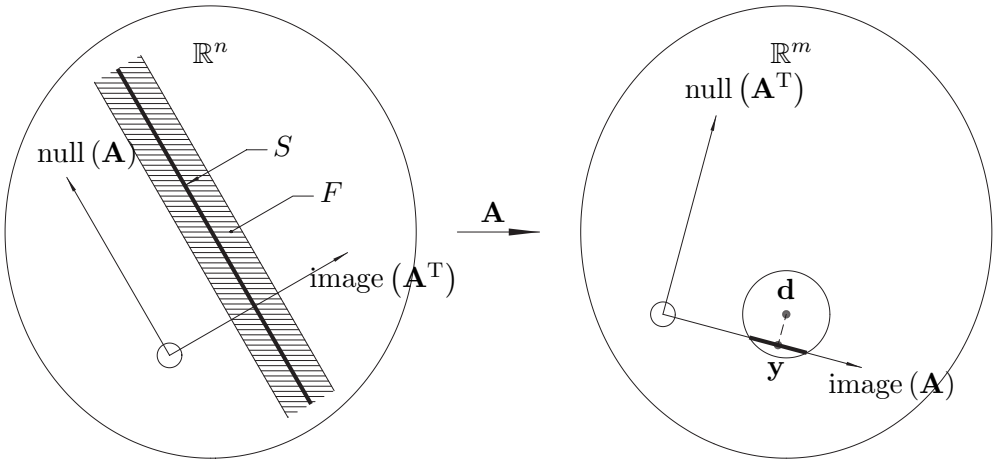


Figure 2.6: Geometry of a linear inverse problem.

image $(\mathbf{A})$. In the image space, the feasible set is represented by a strip as shown. Although feasible in the sense of fitting the data to some specified precision, most of the images in F have wildly oscillatory and non-physical properties (e.g., negative intensities, etc.) The essential goal of regularization methods to be discussed in the next chapter is to choose a "reasonable" solution within the feasible set.

2.9 Classifying linear operators

In the theory of inverse problems it is useful to classify operators $\mathbf{A}$ according to whether the system of equations $\mathbf{Af} = \mathbf{d}$ has solutions for various values of $\mathbf{d}$.

1. If the image of the operator $\mathbf{A}$ has smaller dimension than the data space, i.e., if $r < m$, there are vectors $\mathbf{d} \in \mathbb{R}^m$ which lie outside the range of $\mathbf{A}$. This means that for some possible $\mathbf{d}$, the equations $\mathbf{Af} = \mathbf{d}$ have no solution and it is necessary to use some condition such as least-squares to select a vector $\hat{\mathbf{f}}$ which minimizes the discrepancy between $\mathbf{Af}$ and $\mathbf{d}$. Whether or not this least-squares solution is unique or not depends on the next condition.

2. If the image of the operator $\mathbf{A}$ has smaller dimension than the image space, i.e., if $r < n$, then there are an infinite number of vectors $\mathbf{x}$ all of which map under $\mathbf{A}$ to the same vector $\mathbf{d}$. This means that for some choices of $\mathbf{d}$, the equations $\mathbf{Af} = \mathbf{d}$ have infinitely many solutions. Whether or not solutions exist for all values of $\mathbf{d}$ depends on the previous condition.

3. The only situation in which $\mathbf{Af} = \mathbf{d}$ has a unique solution for every $\mathbf{d}$ is if $r = m = n$ so that the matrix $\mathbf{A}$ is square and invertible. This situation almost *never* holds in practice and it is almost always a bad idea to try to force an inverse problem into a form with a square invertible matrix with a view of solving the problem by a solution of these equations. This reasons for this should become clearer in the following discussions.

Whenever the second condition holds, it is necessary to use additional information over and above the data collected in order to select a good reconstruction from among the possible

reconstructions. One way of doing this is by using the process of *regularization* which we shall examine in more detail later. In practice, it is often the case that both the first two conditions hold and r is strictly less than both m and n. When this is the case, there is no solution to $\mathbf{Af} = \mathbf{d}$ for some $\mathbf{d}$ while there are infinitely many solutions for other possible $\mathbf{d}$. If there is no solution for some $\mathbf{d}$, it makes sense to find $\mathbf{f}$ to minimize $\|\mathbf{d} - \mathbf{Af}\|^2$. It turns out that this minimization gives a unique $\mathbf{f}$ if $r = n$ but there are an infinity of vectors all of which give exactly the same minimum norm if $r < n$.

2.10 The effects of noise and small singular values

In the theory discussed so far, we have drawn a sharp distinction between the eigenvalues of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ which are non-zero and those which are zero. Indeed the rank r of $\mathbf{A}$ may be defined as the number of non-zero eigenvalues of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$. In practice of course, when the eigenvalues of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ are sorted in decreasing order, there is a smooth transition from the large eigenvalues through the small eigenvalues to the tiny eigenvalues and the actual rank is always equal to the smaller of m or n. A more useful concept is the effective rank, which depends on the threshold below which we consider the eigenvalue (or the corresponding singular value) to be negligible.

For typical measurement processes, large or slowly-varying portions in the image are well-represented in the data while structures on fine scales or with high frequency components tend to be poorly represented. This is because measurements can usually only be made with a certain spatial or temporal resolution. For example, if we are taking a photograph of an object with visible light, structures on the object with a scale smaller than the wavelength are invisible. The singular vectors in image space associated with the large singular values for such an operator will tend to be smooth, while those associated with the small singular values will tend to be highly irregular or oscillatory.

In order to understand how the small singular values affect the reconstruction process, we consider a simple model for the measurement uncertainty or noise that is always present in data. Let us suppose that the measured data $\mathbf{d}$ may be written as the sum of the transformed image $\mathbf{y} = \mathbf{Af}$ and a noise vector $\mathbf{n}$ so that

$$\mathbf{d} = \mathbf{Af} + \mathbf{n}. \tag{2.72}$$

The vector $\mathbf{f}$ represents the true underlying image and $\mathbf{y}$ is the data that would have been obtained in the absence of noise. Neither of these quantities is known in practice, but the aim of reconstruction is to find a vector approximating to $\mathbf{f}$. Substituting the singular-value decomposition of $\mathbf{A}$ into this yields

$$\mathbf{d} = \left(\sum_{k=1}^{n} \sigma_k \mathbf{u}_k \mathbf{v}_k^{\mathrm{T}} \right) \mathbf{f} + \mathbf{n} \tag{2.73}$$

where the rank has been taken to be n, the size of the image space (assumed to be smaller than m). The forward mapping is strictly 1-1, and so there is a unique least-squares solution which we have seen is given by:

$$\hat{\mathbf{f}} = \sum_{k=1}^{n} \left(\frac{\mathbf{u}_k^{\mathrm{T}} \mathbf{d}}{\sigma_k} \right) \mathbf{v}_k. \tag{2.74}$$

Substituting the expansion (2.73) gives

$$\hat{\mathbf{f}} = \sum_{k=1}^{n} \left(\mathbf{v}_k^{\mathrm{T}} \mathbf{f} + \frac{\mathbf{u}_k^{\mathrm{T}} \mathbf{n}}{\sigma_k} \right) \mathbf{v}_k \tag{2.75}$$

$$= \mathbf{f} + \sum_{k=1}^{n} \frac{\left(\mathbf{u}_k^{\mathrm{T}} \mathbf{n}\right)}{\sigma_k} \mathbf{v}_k \tag{2.76}$$

where we have made use of the fact that the $\{\mathbf{v}_k\}$ form a basis for $\mathbb{R}^n$. We see that the reconstruction is the sum of the true image and terms due to the noise. The error term along the direction of $\mathbf{v}_k$ in image space arises from the component of the noise in the direction of $\mathbf{u}_k$ in data space, *divided by* the singular value σ_k.

If we now suppose that some of the singular values σ_k are small, this division will give a very large random component, often completely swamping the component of $\mathbf{f}$ in that direction. Another way of thinking about this is to see that the small singular values correspond to directions in image space for which the data contain very little information about the image. In attempting to reconstruct those aspects of $\mathbf{f}$ which lie along these directions, we have to amplify the small signal buried in the data by dividing by the small singular value. Such a scheme is risky because the noise which is inevitably present in the data is also going to be amplified by a large factor, corrupting the reconstruction.

Thus when there are small singular values, the least squares method can give bad reconstructions. It is better to consider small singular values as being effectively zero, and to regard the components along such directions as being free parameters which are not determined by the data.

When the singular values of the measurement operator $\mathbf{A}$ are ranked in non-increasing order, the rate at which they decrease with index gives valuable information about how much we can hope to reconstruct from data taken using that measurement process for a given amount of noise in the data. The more rapid is the decrease, the less we can reconstruct reliably for a given noise level. Equivalently, in order to get good reconstructions when the singular values decrease rapidly, an extremely high signal-to-noise ratio in the data is required.

2.11 Continuous transformations

Our analysis of the forward map A via the SVD required that both image space and data space to be finite dimensional so the A is represented by a $m \times n$ matrix. When one or both spaces are continuous we can still perform a singular-value decomposition with the aid of a little more mathematical machinery, as presented in the next section. We will only deal fully with the case where just one of the spaces is continuous and investigate some of the issues for the continuous-continuous problem.in an assignment.

We deal with continuous spaces by first considering how the ideas of matrices may be extended to infinite dimensional spaces and then show how to construct the matrix (conjugate) transpose so that we are able to form the analogs of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ and $\mathbf{A}\mathbf{A}^{\mathrm{T}}$. Both these constructions are achieved by using the "star" operator defined in the next section.

2.11.1 Adjoint operators and the star notation

Adjoints of Operators

One of the most powerful and pervasive concepts in linear algebra is the concept of adjoints. Its generalization to function spaces will allow the use of matrix notation for continuous-discrete and continuous-continuous forward operators.

For fixed inner-products in $\mathbb{C}^N$ and $\mathbb{C}^M$, the adjoint of a linear operator $F : \mathbb{C}^N \to \mathbb{C}^M$ is a another linear operator

$$F^* : \mathbb{C}^M \to \mathbb{C}^N$$

satisfying the relation

$$\langle \mathbf{u}, F^*\mathbf{v} \rangle = \langle F\mathbf{u}, \mathbf{v} \rangle , \quad \text{for all } \mathbf{u} \in \mathbb{C}^N, \mathbf{v} \in \mathbb{C}^M. \tag{2.77}$$

Note that the first inner-product is in $\mathbb{C}^N$ while the second is in $\mathbb{C}^M$. Strictly, equation 2.77 is not a definition of F^* as it is actually a (non-trivial) consequence of the properties of inner-products that F^* must exist and be unique.

A concrete picture of adjoints can be gained by choosing bases $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_N\}$ of $\mathbb{C}^N$ and $\{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_M\}$ of $\mathbb{C}^M$ which are orthonormal with respect to the respective inner-products. Recall that the operator F is represented by the $M \times N$ matrix $[F_{mn}]$ and F^* is represented by the $N \times M$ matrix $[(F^*)_{nm}]$. The relationship between these two matrices can be found by considering the defining expressions

$$F\mathbf{b}_n = \sum_{m=0}^{M} \mathbf{d}_m F_{mn}, \quad F^*\mathbf{d}_m = \sum_{n=0}^{N} \mathbf{b}_n \left(F^*\right)_{nm}.$$

Since both bases are orthonormal, the inner-product of both sides of the first equation with $\mathbf{d}_k$, and the inner-product of both sides of the second equation with $\mathbf{b}_l$ gives

$$\langle \mathbf{d}_k, F\mathbf{b}_n \rangle = F_{kn}, \quad \langle \mathbf{b}_l, F^*\mathbf{d}_m \rangle = \left(F^*\right)_{lm}.$$

Setting $k = m$ and $l = n$ and using the Hermiticity property of the inner-product,

$$\left(F^*\right)_{nm} = \langle F\mathbf{b}_n, \mathbf{d}_m \rangle = \overline{\langle \mathbf{d}_m, F\mathbf{b}_n \rangle} = \overline{F_{mn}}.$$

Hence the matrix $[(F^*)_{nm}]$ representing F^* is the complex conjugate of the transpose[3] of the matrix $[F_{mn}]$ representing F. So a simple way of thinking of the adjoint operator is to think of the conjugate transpose of the representing matrix. In the general case of non-orthonormal bases, the relationship between matrices is more complicated. However we will typically only be considering orthonormal bases and so the simple picture is sufficient.

Adjoints of Vectors

The notion of the adjoint also applies to vectors when we associate the vector with its natural operation as a matrix. Think of the vector $\mathbf{u} \in \mathbb{C}^N$ as a simple operator that takes a complex number and produces a vector, i.e.,

$$\mathbf{u} : \mathbb{C} \to \mathbb{C}^N,$$

[3]The conjugate transpose is also called the Hermitian conjugate.

defined in the obvious way: for $c \in \mathbb{C}$,

$$\mathbf{u}c = c\mathbf{u}.$$

The left-hand side describes $\mathbf{u}$ acting on c while the right-hand side defines the result and is just the scalar multiplication of $\mathbf{u}$ by c.

Then the adjoint of $\mathbf{u}$ is

$$\mathbf{u}^*:\mathbb{C}^N \to \mathbb{C}$$

defined by equation 2.77, which implies[4] that for $\mathbf{v} \in \mathbb{C}^N$

$$\mathbf{u}^*\mathbf{v} = \langle \mathbf{u}, \mathbf{v} \rangle$$

So $\mathbf{u}^*$ is the operator that takes the inner product with $\mathbf{u}$, or equivalently, is the process of multiplying by the conjugate transpose of the representation of the vector. So, again, the adjoint may be thought of as the conjugate transpose. Since $\mathbf{u}$ is represented by a column vector, $\mathbf{u}^*$ is a row vector.

In the Physics literature another notation is often used in which one writes

$$\begin{array}{ccc} \mathbf{u}^* & \text{as} & \langle \mathbf{u}| \quad \text{(called a bra)} \\ \mathbf{v} & \text{as} & |\mathbf{v}\rangle \quad \text{(called a ket)} \end{array}$$

Finally, note that every linear operator that maps $\mathbb{C}^N \to \mathbb{C}$ can be written as $\mathbf{u}^*$ for some $\mathbf{u} \in \mathbb{C}^N$.

Adjoints of Functions

So far we appear to have done little more than observe that the inner product may be written as $\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u}^*\mathbf{v}$ where $\mathbf{u}^*$ is usually the conjugate transpose of $\mathbf{u}$ – though our notation does work for representations in non-orthonornmal bases as well. But the real power of the notation is that it works for infinite-dimensional vectors as well – such as those represented by functions of a continuous variable.

For example, a vector space of functions that we often use is the space of all square integrable functions defined on the real line. That is, the set of functions

$$L^2(\mathbb{R}) = \left\{ f : \mathbb{R} \to \mathbb{C} \text{ and } \|f\| \equiv \int_{-\infty}^{\infty} |f(x)|^2 \, \mathrm{d}x < \infty \right\}.$$

For $f, g \in L^2(\mathbb{R})$, the usual inner product is given by

$$\langle f, g \rangle = \int_{-\infty}^{\infty} \overline{f(x)} g(x) \, \mathrm{d}x.$$

In the same way as for finite dimensional vectors, we can think of $f(x)$ as a simple operator

$$f:\mathbb{C} \to L^2(\mathbb{R})$$

[4]Consider $\langle \mathbf{u}c, \mathbf{v} \rangle$ for any $c \in \mathbb{C}$ and $\mathbf{v} \in \mathbb{C}^N$. As the inner product is anti-linear in the first argument we have $\langle \mathbf{u}c, \mathbf{v} \rangle = \bar{c} \langle \mathbf{u}, \mathbf{v} \rangle$ while the definition of the $\mathbf{u}^*$ is $\langle \mathbf{u}c, \mathbf{v} \rangle = \langle c, \mathbf{u}^*\mathbf{v} \rangle = \bar{c}\mathbf{u}^*\mathbf{v}$. The implication follows.

defined by

$$fc = cf(x).$$

The adjoint of this operator is

$$f^*{:}L^2(\mathbb{R}) \to \mathbb{C}$$

which is given by, for any $g \in L^2(\mathbb{R})$,

$$f^* g = \langle f, g \rangle.$$

Again we note that any bounded linear functional[5] that maps $L^2(\mathbb{R})$ to $\mathbb{C}$ can be written as f^* for some function f.

Example – the Fourier Transform

The star notation provides a handy shorthand notation for the Fourier transform. For any frequency let

$$e_\nu(t) = \exp(2\pi \mathrm{i} \nu t)$$

then the Fourier transform of a function $f(t) \in L^2(\mathbb{R})$ is

$$F(\nu) = e_\nu^* f.$$

Example – first N moments

Consider reconstructing a function $f(x)$ defined on the interval $[0,1]$ from measurements of its first N moments

$$d_l = \int_0^1 x^{l-1} f(x)\, \mathrm{d}x \qquad l = 1, 2, \ldots, N.$$

By defining the functions

$$a_l(x) = x^{l-1} \quad \text{for } x \in [0,1],\ l = 1, 2, \ldots, N$$

we can write the forward map as

$$d_l = a_l^* f, \quad l = 1, 2, \ldots, N$$

or in the "matrix-vector" form –

$$\mathbf{d} = Af$$

where $\mathbf{d}$ is the N-dimensional vector of data. Here A is the discrete×continuous "matrix"

$$A = (a_l^*(x)) = \begin{pmatrix} \cdots & 1 & \cdots \\ \cdots & x & \cdots \\ \cdots & x^2 & \cdots \\ & \vdots & \\ \cdots & x^{N-1} & \cdots \end{pmatrix}$$

in which the rows are the indicated functions defined on $[0,1]$. The generalization of the matrices $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ and $\mathbf{A}\mathbf{A}^{\mathrm{T}}$are the representations of the operators $\mathbf{A}^*\mathbf{A}$ and $\mathbf{A}\mathbf{A}^*$ – the first of

[5] A functional is just a linear operator that takes a function as its argument.

which is continuous×continuous while the later is discrete×discrete. So naturally we choose the smaller matrix which is the $N \times N$ matrix which has kl component

$$(AA^*)_{kl} = a_k^* a_l = \int_0^1 x^{l-1} x^{k-1} \, \mathrm{d}x = \frac{1}{k+l-1},$$

where we have used the important relation that $(a_l^*)^* = a_l$.

First the simplest case, $N = 1$. Now $AA^* = (1)$ which has one eigenvalue of 1 with eigenvector (1). Consequently, A has one non-zero singular value, equalling 1, and the corresponding singular vector in image space is

$$A^* (1) = 1 (x),$$

i.e., the constant function with value 1. The left-hand side is the matrix A^* acting on the data vector (1) ,and since $A = a_1^* (x) = 1^*$, in this case, then $A^* = a_1 (x) = 1 (x)$ and the function $1 (x)$ acting on the number 1 simply results in the function $1 (x)$ which is the constant function with value 1.

A bit more substantial is the case $N = 3$: Now

$$AA^* = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{pmatrix}$$

$$\begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{pmatrix}$$

which has the normalized eigenvectors: $\mathbf{u}_1 = \begin{pmatrix} 0.82706 \\ 0.46039 \\ 0.32248 \end{pmatrix}$, $\mathbf{u}_2 = \begin{pmatrix} 0.54744 \\ -0.52822 \\ -0.64909 \end{pmatrix}$, and $\mathbf{u}_3 = \begin{pmatrix} 0.12766 \\ -0.71375 \\ 0.68868 \end{pmatrix}$, with eigenvalues 1.4083, 0.12233, and 2.6873×10^{-3}, respectively. The vectors are also the left singular vectors of A while the singular values are the square-roots of the eigenvalues and are: $\sigma_1 = 1.1867$, $\sigma_2 = 0.34976$, and $\sigma_3 = 5.1839 \times 10^{-2}$. The right singular vectors may be found, as in equation 2.32 or 2.36, by operating on each of the left singular vectors by A^* and dividing by the corresponding singular value. The resulting functions are

$$v_1 (x) = \frac{A^* \begin{pmatrix} 0.82706 \\ 0.46039 \\ 0.32248 \end{pmatrix}}{1.1867} = 0.69694 + 0.38796x + 0.27175x^2$$

$$v_2 (x) = \frac{A^* \begin{pmatrix} 0.54744 \\ -0.52822 \\ -0.64909 \end{pmatrix}}{0.34976} = 1.5652 - 1.5102x - 1.8558x^2$$

$$v_3 (x) = \frac{A^* \begin{pmatrix} 0.12766 \\ -0.71375 \\ 0.68868 \end{pmatrix}}{5.1839 \times 10^{-2}} = 2.4626 - 13.769x + 13.285x^2$$

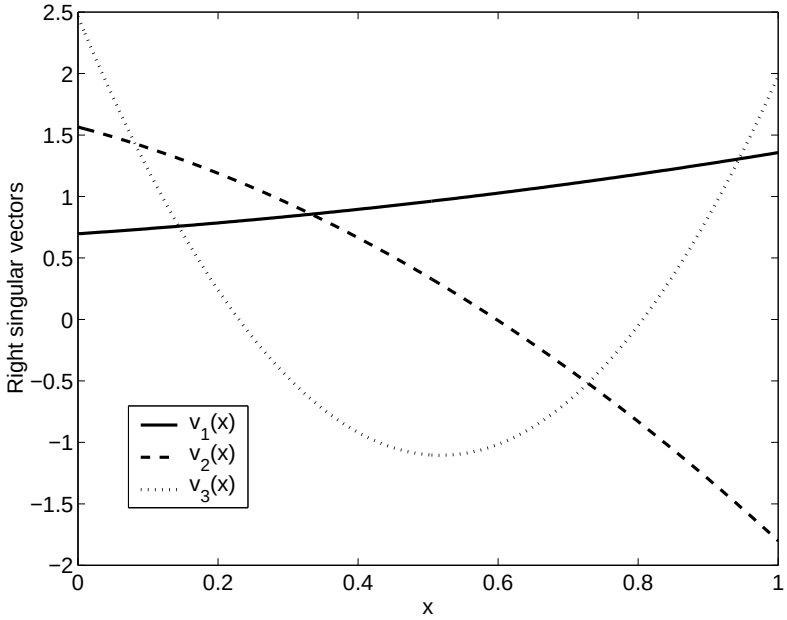


Figure 2.7: The three right singular vectors

With a little work you should be able to show that these functions are indeed orthonormal over the interval $[0,1]$. Just for the record, here is a graph of the three right singular vectors

These are the three structures in the image which contribute to the data. Note that the measurement process is about 20 times less sensitive to $v_3(x)$ than to $v_1(x)$.

In the case where 10 moments are measured we have $N = 10$ and

$$AA^* = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} & \frac{1}{14} \\ \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} & \frac{1}{14} & \frac{1}{15} \\ \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} & \frac{1}{14} & \frac{1}{15} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} & \frac{1}{14} & \frac{1}{15} & \frac{1}{16} & \frac{1}{17} \\ \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} & \frac{1}{14} & \frac{1}{15} & \frac{1}{16} & \frac{1}{17} & \frac{1}{18} \\ \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} & \frac{1}{14} & \frac{1}{15} & \frac{1}{16} & \frac{1}{17} & \frac{1}{18} & \frac{1}{19} \end{pmatrix}.$$

The square root of the eigenvalues give the singular values

index	**singular value**
1	1.3236
2	0.5856
3	0.18906
4	5.0308×10^{-2}
5	1.1347×10^{-2}
6	2.1748×10^{-3}
7	3.5057×10^{-4}
8	4.634×10^{-5}
9	4.761×10^{-6}
10	3.3064×10^{-7}

Notice how the singular values fall off rapidly. With a measurement accuracy of 1 part in 10^3, for example, we would expect to measure the first 6 components (to varying degrees) while the remaining components of the data will typically be smaller than the noise.

3

Regularization Methods for Linear Inverse Problems

The primary difficulty with linear ill-posed problems is that the inverse image is undetermined due to small (or zero) singular values of $\mathbf{A}$. Actually the situation is a little worse in practice because $\mathbf{A}$ depends on our model of the measurement process and that is typically not precisely known, leading to a slight imprecision in the singular values. Usually that is not significant for the large singular values, but may lead to ambiguity in the small singular values so that we do not know if they are small or zero.

As an introduction to regularization, which is one method for surmounting the problems associated with small singular vectors, we consider a framework for describing the quality of a reconstruction $\hat{\mathbf{f}}$ in an inverse problem.

3.1 The data misfit and the solution semi-norms

In the last chapter, we considered the linear problem

$$\mathbf{d} = \mathbf{A}\mathbf{f} + \mathbf{n}$$

and focused on the structure of the operator $\mathbf{A} \in \mathbb{R}^{m\times n}$. As far as the data are concerned, a reconstructed image $\hat{\mathbf{f}}$ is good provided that it gives rise to 'mock data' $\mathbf{A}\hat{\mathbf{f}}$ which are close to the observed data. Thus, one of the quantities for measuring the quality of $\hat{\mathbf{f}}$ is the *data misfit* function which is usually the square of the *residual norm*

$$C(\mathbf{f}) = \|\mathbf{d} - \mathbf{A}\mathbf{f}\|^2 . \tag{3.1}$$

However, from our previous considerations, we have seen that choosing $\hat{\mathbf{f}}$ so as to minimize $C(\mathbf{f})$ usually gives a poor reconstruction. If the rank of the operator $\mathbf{A}$ is less than n there are an infinite number of reconstructions, all of which minimize $C(\mathbf{f})$, since the data are not affected by adding to a reconstruction any vector which lies in the null space of $\mathbf{A}$. In the presence of noise, finding the (possibly non-unique) minimum of C is undesirable as it leads to amplification of the noise in the directions of the singular vectors with small singular values. Instead, we usually regard the data as defining a *feasible set* of reconstructions for which $C(\mathbf{f}) \leq C_0$ where C_0 depends on the 'level' of the noise. Any reconstruction within the feasible set is to be thought of as being consistent with the data.

Since the data do not give us any information about some aspects of $\mathbf{f}$, it is necessary to include additional information which allows us to select from among several feasible reconstructions. Analytical solutions are available if we choose sufficiently simple criteria. One way of doing this is to introduce a second function $\Omega(\mathbf{f})$ representing our aversion to a particular

reconstruction. For example, we may decide that the solution having minimum norm should be chosen from among the feasible set. This can be done by choosing

$$\Omega(\mathbf{f}) = \|\mathbf{f}\|^2 . \tag{3.2}$$

Sometimes, we have a preference for reconstructions which are close to some *default solution* $\mathbf{f}^\infty$. This may be appropriate if we have historical information about the quantity. This can be done by choosing

$$\Omega(\mathbf{f}) = \|\mathbf{f} - \mathbf{f}^\infty\|^2 . \tag{3.3}$$

More generally, it may not be the norm of $\mathbf{f} - \mathbf{f}^\infty$ which needs to be small, but some linear operator acting on this difference. Introducing the operator $\mathbf{L}$ for this purpose, we can set

$$\Omega(\mathbf{f}) = \|\mathbf{L}(\mathbf{f} - \mathbf{f}^\infty)\|^2 = (\mathbf{f} - \mathbf{f}^\infty)^{\mathrm{t}} \mathbf{L}^{\mathrm{t}} \mathbf{L}(\mathbf{f} - \mathbf{f}^\infty) . \tag{3.4}$$

If the image space is n dimensional and the data space is m dimensional, the matrix $\mathbf{A}$ is of size $m \times n$ and the matrix $\mathbf{L}$ is of size $p \times n$ where $p \leq n$. Typically, $\mathbf{L}$ is the identity matrix or a banded matrix approximation to the $(n-p)$'th derivative. For example, an approximation to the first derivative in 1-dimension is given by the matrix

$$\mathbf{L}_1 = \frac{1}{\Delta x} \begin{pmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{pmatrix},$$

while an approximation to the second derivative is

$$\mathbf{L}_2 = \frac{1}{(\Delta x)^2} \begin{pmatrix} 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \end{pmatrix}.$$

In other cases, it may be appropriate to minimize some combination of the derivatives such as

$$\Omega(\mathbf{f}) = \alpha_0 \|\mathbf{f} - \mathbf{f}^\infty\|^2 + \sum_{k=1}^{q} \alpha_k \|\mathbf{L}_k(\mathbf{f} - \mathbf{f}^\infty)\|^2 ,$$

where $\mathbf{L}_k$ is a matrix which approximates the k'th derivative, and α_k are non-negative constants. Such a quantity is also the square of a norm, called a *Sobolev norm.*

It is a simple result from linear algebra that any real symmetric positive semidefinite matrix may be factorized into a product of the form $\mathbf{L}^{\mathrm{t}}\mathbf{L}$ where $\mathbf{L}$ is a lower triangular matrix. A constructive proof of this result leads to the *Cholesky factorization* of the square matrix. The Sobolev norm above may thus also be written in the form of (3.4) for a suitable choice of $\mathbf{L}$.

There are many ways of balancing the often conflicting requirements of equations (3.4) and (3.1) which leads to a variety of regularization methods. We shall discuss two of these below.

3.2 Tikhonov regularization

This is perhaps the most common and well-known of regularization schemes. We form a weighted sum of $\Omega(\mathbf{f})$ and $C(\mathbf{f})$ using a weighting factor λ^2, and find the image $\hat{\mathbf{f}}_\lambda$ which minimizes this sum, i.e.,

$$\hat{\mathbf{f}}_\lambda = \arg\min_{\mathbf{f}} \left\{ \lambda^2 \left\| \mathbf{L}\left(\mathbf{f} - \mathbf{f}^\infty\right) \right\|^2 + \left\| \mathbf{d} - \mathbf{A}\mathbf{f} \right\|^2 \right\}. \tag{3.5}$$

This is a whole family of solutions parameterized by the weighting factor λ^2. We call λ the *regularization parameter.* If the regularization parameter is very large, the effect of the data misfit term $C(\mathbf{f})$ is negligible to that of $\Omega(\mathbf{f})$ and we find that $\lim_{\lambda\to\infty} \hat{\mathbf{f}}_\lambda = \mathbf{f}^\infty$. With a large amount of regularization, we effectively ignore the data (and any noise on the data) completely and try to minimize the solution semi-norm which is possible by choosing the default solution. On the other hand, if λ is small, the weighting placed on the solution semi-norm is small and the value of the misfit at the solution becomes more important. Of course, if λ is reduced to zero, the problem reduces to the least-squares case considered earlier with its extreme sensitivity to noise on the data.

A formal solution to the problem may readily be found. We set

$$\frac{\partial}{\partial f_k} \left\{ \lambda^2 \left(\mathbf{f} - \mathbf{f}^\infty\right)^{\mathrm{t}} \mathbf{L}^{\mathrm{t}}\mathbf{L}\left(\mathbf{f} - \mathbf{f}^\infty\right) + \left(\mathbf{d} - \mathbf{A}\mathbf{f}\right)^{\mathrm{t}} \left(\mathbf{d} - \mathbf{A}\mathbf{f}\right) \right\} = 0, \tag{3.6}$$

for $k = 1, 2, ..., n$. This leads to the simultaneous equations

$$2\lambda^2 \mathbf{L}^{\mathrm{t}}\mathbf{L}\left(\mathbf{f} - \mathbf{f}^\infty\right) - 2\mathbf{A}^{\mathrm{t}}\left(\mathbf{d} - \mathbf{A}\mathbf{f}\right) = 0, \tag{3.7}$$

or

$$\left(\lambda^2 \mathbf{L}^{\mathrm{t}}\mathbf{L} + \mathbf{A}^{\mathrm{t}}\mathbf{A}\right)\mathbf{f} = \lambda^2 \mathbf{L}^{\mathrm{t}}\mathbf{L}\mathbf{f}^\infty + \mathbf{A}^{\mathrm{t}}\mathbf{d}. \tag{3.8}$$

Setting $\lambda = 0$ reduces this system of equations to the normal equations associated with the usual least squares problem. For non-zero values of λ, the additional term $\lambda^2\mathbf{L}^{\mathrm{t}}\mathbf{L}$ in the matrix on the left-hand side alters the eigenvalues (and eigenvectors) from those of $\mathbf{A}^{\mathrm{t}}\mathbf{A}$ alone. So long that $\left(\lambda^2\mathbf{L}^{\mathrm{t}}\mathbf{L} + \mathbf{A}^{\mathrm{t}}\mathbf{A}\right)$ is non-singular, there is a unique solution. The problem of image reconstruction is thus reduced to solving a (large) system of simultaneous equations with a symmetric positive definite coefficient matrix, and we shall later discuss ways of solving such systems of equations.

3.3 Truncated singular value decomposition (TSVD)

Let us suppose that the operator $\mathbf{A}$ has the singular value decomposition

$$\mathbf{A} = \sum_{l=1}^{r} \sigma_l \mathbf{u}_l^{\mathrm{t}} \mathbf{v}_l. \tag{3.9}$$

The truncated singular value decomposition (TSVD) method is based on the observation that for the larger singular values of $\mathbf{A}$, the components of the reconstruction along the corresponding singular vector is well-determined by the data, but the other components are not well-determined. An integer $k \leq n$ is chosen for which the singular values are deemed to be significant and the solution vector $\hat{\mathbf{f}}$ is chosen so that

$$\mathbf{v}_l^{\mathrm{T}}\hat{\mathbf{f}} = \frac{\mathbf{u}_l^{\mathrm{T}}\mathbf{d}}{\sigma_l} \text{ for } l = 1, ..., k. \tag{3.10}$$

The components along the remaining singular-vector directions $\{\mathbf{v}_l\}$ for $l = k+1, ..., n$ are then chosen so that the *total* solution vector $\hat{\mathbf{f}}$ satisfies some criterion of optimality, such as the minimization of a solution semi-norm of the form $\Omega(\mathbf{f}) = \|\mathbf{L}(\mathbf{f} - \mathbf{f}^\infty)\|^2$ as above. Let us denote by $\mathbf{V}_k$ the $n \times (n-k)$ matrix whose columns are $\{\mathbf{v}_l\}$ for $l = k+1, ..., n$ so that $\mathbf{V}_k$ is the matrix whose columns span the effective null space of $\mathbf{A}$. The reconstruction which has zero projection in this effective null space is

$$\mathbf{f}' = \sum_{l=1}^{k} \left(\frac{\mathbf{u}_l^{\mathrm{T}} \mathbf{d}}{\sigma_l} \right) \mathbf{v}_l \ . \tag{3.11}$$

The desired reconstruction $\hat{\mathbf{f}}$ must be equal to the sum of $\mathbf{f}'$ and a vector which is the superposition of the columns of $\mathbf{V}_k$. This may be written as

$$\hat{\mathbf{f}} = \mathbf{f}' + \sum_{l=k+1}^{n} c_l \mathbf{v}_l = \mathbf{f}' + \mathbf{V}_k \mathbf{c} \tag{3.12}$$

for a $n-k$ element column vector $\mathbf{c}$. The solution semi-norm of this reconstruction is

$$\Omega\left(\hat{\mathbf{f}}\right) = \left\|\mathbf{L}\left(\hat{\mathbf{f}} - \mathbf{f}^\infty\right)\right\|^2 = \left\|\mathbf{L}\left(\mathbf{f}' + \mathbf{V}_k\mathbf{c} - \mathbf{f}^\infty\right)\right\|^2 = \left\|\mathbf{L}\left(\mathbf{f}' - \mathbf{f}^\infty\right) + \left(\mathbf{L}\mathbf{V}_k\right)\mathbf{c}\right\|^2$$

The vector $\mathbf{c}$ which minimizes this semi-norm is

$$\mathbf{c} = -\left(\mathbf{L}\mathbf{V}_k\right)^\dagger \mathbf{L}\left(\mathbf{f}' - \mathbf{f}^\infty\right)$$

where the dagger represents the Moore-Penrose inverse. i.e., for any matrix $\mathbf{A}$, we define $\mathbf{A}^\dagger = \left(\mathbf{A}^{\mathrm{t}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathrm{t}}$. This gives an explicit expression for the truncated singular value decomposition solution, namely

$$\hat{\mathbf{f}} = \mathbf{f}' - \mathbf{V}_k\left(\mathbf{L}\mathbf{V}_k\right)^\dagger \mathbf{L}\left(\mathbf{f}' - \mathbf{f}^\infty\right) \tag{3.13a}$$

where $\mathbf{f}'$ is given by (3.11) above.

Note that some authors use the terminology *truncated singular value decomposition* to refer to the special case where $\mathbf{L}$ is chosen to be the identity matrix, and call the general case derived above the *modified truncated singular value decomposition.*

3.4 Filter factors

In any regularization scheme, there is a *regularization parameter* which is a quantity that can be adjusted in order to change the degree of regularization of the solution. For values of this parameter at one end of its range, the solution is usually smoother, more similar to the default solution and less affected by noise on the data whereas for values of this parameter at the other end, the solution can be very sensitive to noise as it is primarily determined by the requirement of minimizing the data residual. In the case of Tikhonov regularization, the parameter is the quantity λ while in the case of the TSVD method, it is the choice of k at which the singular values are deemed to be negligible.

It is useful to be able to look at the range of solutions which result as the regularization parameter is varied. This can always be done by recomputing the solution from scratch for each value of the parameter, but this is computationally very intensive as we often need to invert a large matrix for each choice of the regularization parameter. An advantage of studying the

singular value decomposition is that it provides a convenient way of investigating the family of regularized solutions without having to reinvert large matrices. We shall thus re-examine the regularization methods described above in terms of the singular value decomposition.

Tikhonov regularization can be analyzed in this way when the matrix $\mathbf{L}$ happens to be the identity. For more general $\mathbf{L}$, an extension of the singular value decomposition called the generalized singular value decomposition (GSVD) may be defined, but we shall not be considering it in this course. The solution to the problem is given by

$$\left(\lambda^2\mathbf{I} + \mathbf{A}^{\mathrm{T}}\mathbf{A}\right)\hat{\mathbf{f}} = \lambda^2\mathbf{f}^{\infty} + \mathbf{A}^{\mathrm{t}}\mathbf{d} \tag{3.14}$$

Let us suppose that we have computed the singular value decomposition of $\mathbf{A}$ in the usual form

$$\mathbf{A} = \sum_{l=1}^{r} \sigma_l \mathbf{u}_l \mathbf{v}_l^{\mathrm{t}} \tag{3.15}$$

then

$$\left(\lambda^2\mathbf{I} + \mathbf{A}^{\mathrm{t}}\mathbf{A}\right)\hat{\mathbf{f}} = \lambda^2 \sum_{l=1}^{n} \hat{f}_l \mathbf{v}_l + \sum_{l=1}^{r} \sigma_l^2 \hat{f}_l \mathbf{v}_l \tag{3.16}$$

$$= \sum_{l=1}^{r} \left(\lambda^2 + \sigma_l^2\right) \hat{f}_l \mathbf{v}_l + \lambda^2 \sum_{l=r+1}^{n} \hat{f}_l \mathbf{v}_l \tag{3.17}$$

where $\hat{f}_l = \mathbf{v}_l^{\mathrm{t}}\hat{\mathbf{f}}$ and

$$\lambda^2\hat{\mathbf{f}}^{\infty} + \mathbf{A}^{\mathrm{t}}\mathbf{d} = \lambda^2 \sum_{l=1}^{n} f_l^{\infty} \mathbf{v}_l + \sum_{l=1}^{r} \sigma_l d_l \mathbf{v}_l \tag{3.18}$$

$$= \sum_{l=1}^{r} \left[\lambda^2 f_l^{\infty} + \sigma_l^2 \left(\frac{d_l}{\sigma_l}\right)\right] \mathbf{v}_l + \lambda^2 \sum_{l=r+1}^{n} f_l^{\infty} \mathbf{v}_l \tag{3.19}$$

where $f_l^{\infty} = \mathbf{v}_l^{\mathrm{t}}\mathbf{f}^{\infty}$, $d_l = \mathbf{u}_l^{\mathrm{t}}\mathbf{d}$ and we have made use of the fact that

$$\mathbf{I} = \sum_{l=1}^{n} \mathbf{v}_l \mathbf{v}_l^{\mathrm{t}}$$

since the $\{\mathbf{v}_l\}_{l=1}^{n}$ form an orthonormal basis of $\mathbb{R}^n$. Equating (3.17) and (3.19) and using the linear independence of the vectors $\mathbf{v}_l$ we see that

$$\hat{f}_l = \begin{cases} \frac{\lambda^2}{\lambda^2+\sigma_l^2} f_l^{\infty} + \frac{\sigma_l^2}{\lambda^2+\sigma_l^2}\left(\frac{d_l}{\sigma_l}\right) & \text{for } l = 1, 2, ..., r, \\ f_l^{\infty} & \text{for } l = r+1, ..., n. \end{cases} \tag{3.20}$$

This displays the solutions to the regularization problem for all values of λ in a convenient form. In the directions of the singular vectors $\mathbf{v}_{r+1}, ..., \mathbf{v}_n$ which span the null space of $\mathbf{A}$, the projections $\hat{f}_l$ of the regularized solution are equal to the projections of the default solution f_l^{∞}. This is not unreasonable as the data do not give us any information about those aspects of the image. On the other hand, along each of the directions $\mathbf{v}_1, ..., \mathbf{v}_r$ for which the data do give us some information, the regularized solution is a weighted linear combination of f_l^{∞}, which is what the default solution would have us take, and of d_l/σ_l which is what the data

alone would have suggested. Notice that since the weights add up to one, the regularized solution lies along the line in n space joining these points. The value of the weights is also of interest. As λ becomes large, the solution is pulled towards the default solution, as expected, but it should be noticed that where along the line the solution ends up at depends on the relative values of λ and of σ_l. The larger is the singular value σ_l, the smaller is the relative effect of the regularization parameter λ. In fact we need to have $\lambda = \sigma_l$ in order to pull the component of the solution to the midpoint of the line joining $f_l^{\infty}\mathbf{v}_l$ and $(d_l/\sigma_l)\,\mathbf{v}_l$. This is desirable since it is precisely in the directions corresponding to the large singular values that the data give us the greatest information.

The quantities $\sigma_l^2/\left(\lambda^2+\sigma_l^2\right)$ which multiply the data coefficient (d_l/σ_l) are called *filter factors.* They show how the algorithm reduces the weighting for the data which are associated with the small singular values. Depending on the level of the noise on the data, we need different amounts of protection against the noise-amplifying effects of reconstruction using the small singular values.

By contrast to the Tikhonov regularization algorithm in which the filter factors smoothly decrease to zero as the singular values gets smaller, the truncated singular value algorithms have filter factors which are equal to unity for those singular values which are deemed to be non-negligible ($l \leq k$) and to zero for those singular values which are negligible ($l > k$). The value of the components of the regularized solution in the directions of the significant singular values are completely determined by the data, as

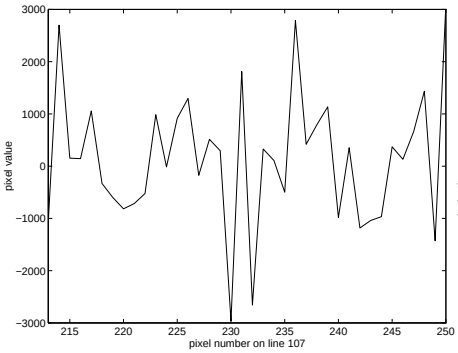
$$\hat{f}_l = \frac{d_l}{\sigma_l} \text{ for } l = 1, 2, ..., k$$

The other components $\hat{f}_{k+1}, ..., \hat{f}_n$ are adjusted so as to minimize the solution semi-norm.

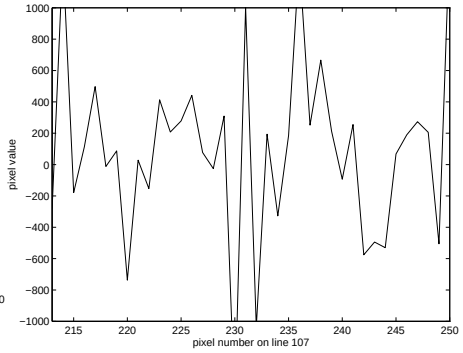
3.5 Smoothness versus data-fitting in the deblurring example

The regularizing parameter, λ, can be thought of as controlling the balance between minimizing the regularizing term – which is often a criterion of smoothness of the reconstructed image – and minimizing the term which corresponds to fitting the data. When λ is small, there is little weight put on the regularizing term, the data is fitted well and the reconstructed image is not smooth. Conversely when λ is large, the regularizer dominates the minimization and the reconstructed image is smooth – at the expense of not fitting the data so well.

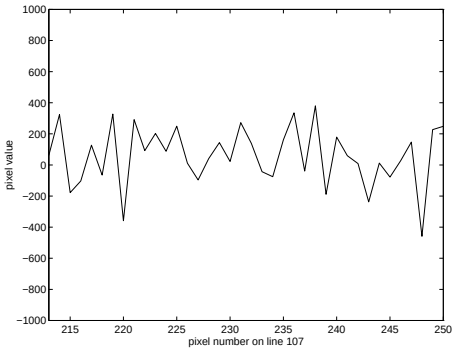
As an example, I have plotted some pixel values from the deblurring example in Chapter 1. The pixels shown are those with indexes `(107,210:250)` – to use the Matlab notation – and are the pixels through the middle of the word 'way'.



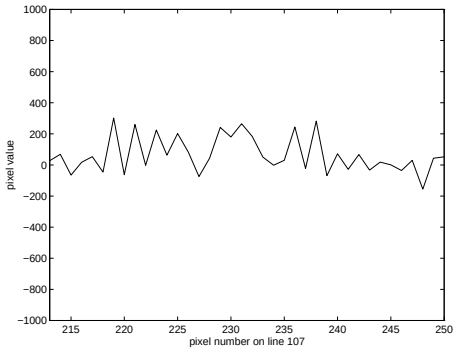
Pixel values in unregularized inverse.



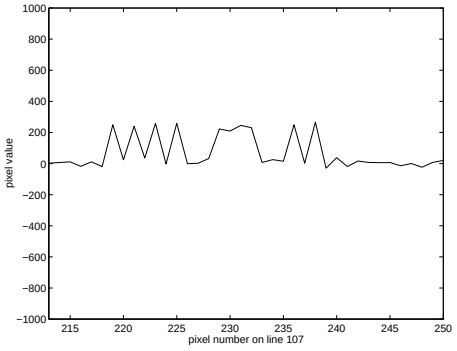
Regularized inverse: $\lambda = 0.1$



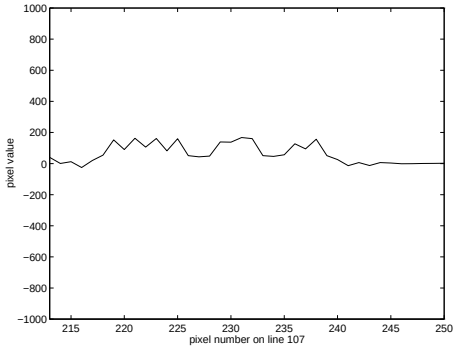
Regularized inverse: $\lambda = 1$



Regularized inverse: $\lambda = 10$



Regularized inverse: $\lambda = 100$



Regularized inverse: $\lambda = 1000$

Note how the graphs of pixel value become more smooth as λ is increased.

3.6 Choosing the regularization parameter

We have seen that λ sets the balance between minimizing the residual norm $\|\mathbf{d} - \mathbf{A}\mathbf{f}_\lambda\|_2$and minimizing the roughness $\|\mathbf{f}_\lambda - \mathbf{f}^\infty\|_2$ .The big question now is "how to choose λ"?

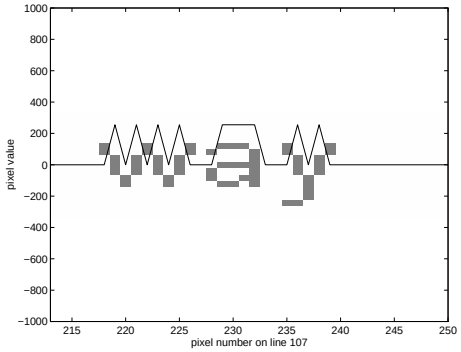


Figure 3.1: Pixel values in original (unblurred and un-noisy) image with the text, as reference.

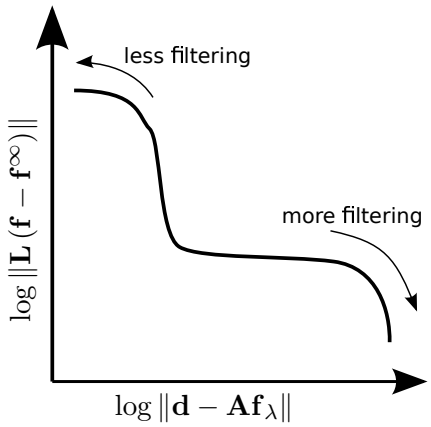


Figure 3.2: The generic form of the L-curve.

3.6.1 The L-Curve

Perhaps the most convenient graphical tool for setting λ is the "L-curve". When we plot $\log \|\mathbf{d} - \mathbf{A}\mathbf{f}_\lambda\|$ versus $\log \|\mathbf{f}_\lambda - \mathbf{f}^\infty\|$ (for a discrete problem) we get the characteristic L-shaped curve with a (often) distinct corner separating vertical and horizontal parts of the curve.

The rationale for using the L curve is that regularization is a trade-off between the data misfit and the solution seminorm. In the vertical part of the curve the solution seminorm $\|\mathbf{L}(\mathbf{f} - \mathbf{f}^\infty)\|$ is a very sensitive function of the regularization parameter because the solution is undergoing large changes with λ in an attempt to fit the data better. At these low levels of filtering, there are still components in the solution which come from dividing by a small singular value, which corresponds to having inadequate regularization. On the horizontal part, the solution is not changing by very much (as measured by the solution seminorm) as λ is changed. However, the data misfit is increasing sharply with more filtering. Along this portion, our preference for the more highly filtered solutions increases only slightly at the cost of a rapidly rising data misfit, and so it is desirable to choose a solution which lies not too far to the right of the corner.

The following figure shows the L-curve that is associated with the deblurring example in chapter 1.

The curve is labeled parametrically with the values of the regularization parameter. In this

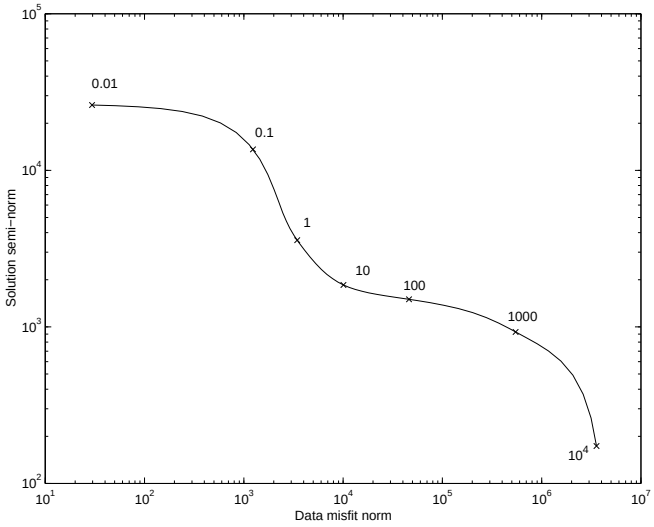


Figure 3.3: L-curve for the deblurring example

example, the bend in the curve is not very sharp, and the solution which appears optimal visually lies slightly to the right of the position of largest upwards-pointing curvature.

3.6.2 The Discrepancy Principle

This is an alternative criterion advocated by those who believe that "the data come first". The value of λ is chosen so as to place the solution on the edge of the feasible set as defined by $C(\mathbf{f}) \leq C_0$. Since $C(\mathbf{f})$ is the sum of squares of noise terms, its expectation value can be calculated if one knows the amount of noise that is present. The value of C_0 is selected such that the probability of exceeding this value due to chance alone is smaller than some threshold, say one percent. A problem with the practical use of this method is that there are often systematic errors as well (such as an inaccurate characterization of the forward problem) in the observation process, so that attempting to fit the data to within the precision allowed by the random errors alone can sometimes lead to under-regularization. This corresponds to finding a solution on the vertical part of the L curve.

3.7 Three Pictorial Examples

3.7.1 Deblurring with missing data

The figures below show a reconstruction based on the blurred image discussed in the first chapter, with the added complication that half the points of the blurred image are now assumed to be unmeasured or corrupted. In Figure 3.4, all unmeasured points are shown as black. The forward problem is as before, except that after the convolution is complete, the unmeasured points are discarded before comparison with the data. In this case, the Fourier division method is inapplicable because the forward map is not even square; Atempting Fourier division by setting missing values to zero fails completely. However, a regularized reconstruction still allows the message to be read as shown in Figure 3.5. In this case the minimization required in Eqn. 3.5 is implemented directly, using the algorithm in Section 3.A.

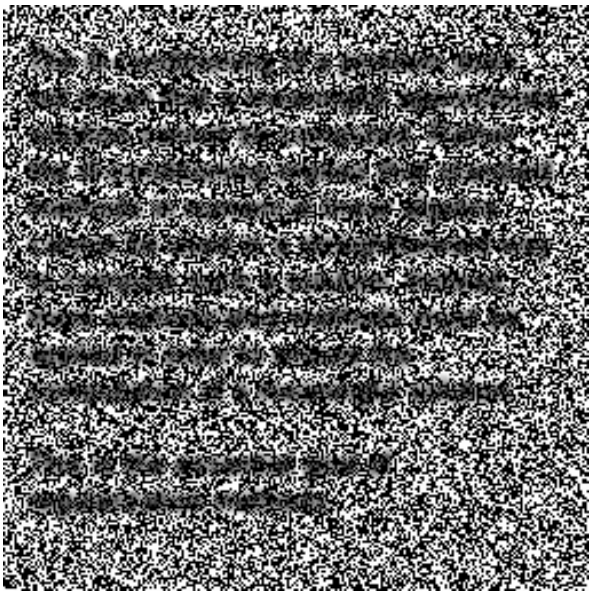


Figure 3.4: Blurred image with missing data. Every black pixel indicates a data point that was unmeasured. In this image, approximately half the pixels are unmeasured.

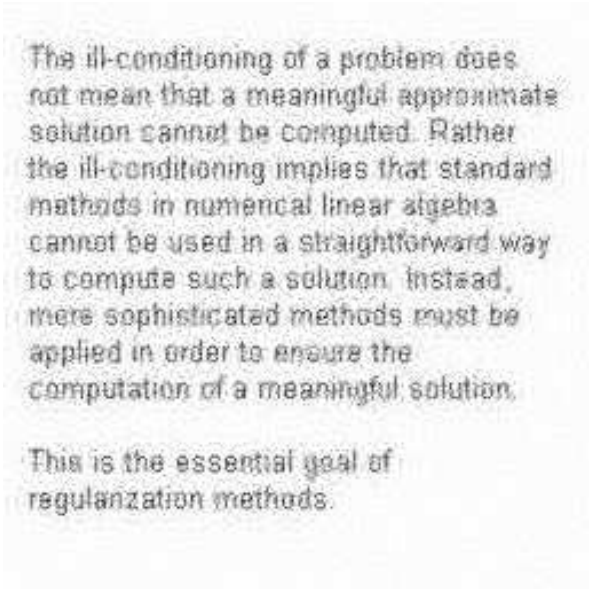
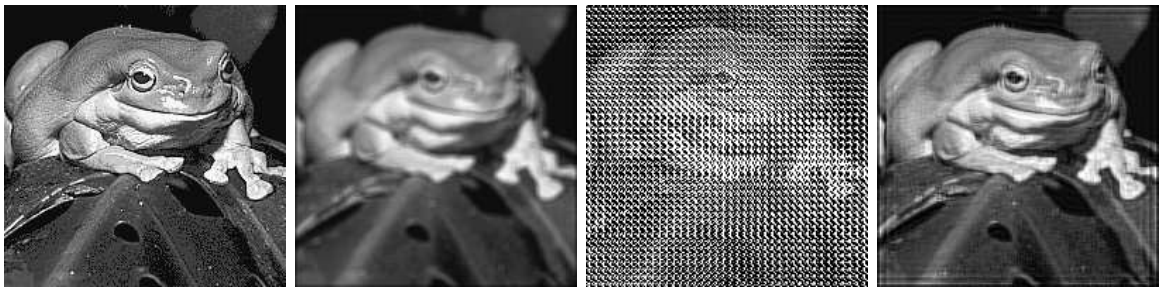


Figure 3.5: Reconstruction from the above data set

3.8 Deblurring with model error and measurement error

Here are two further examples of regularized inversion, showing the effect of model error and measurement error.

Figure 3.6 shows a tableau of images, starting with the original (true) image on the left. A blurred version was created using the MatLab command



original blurred direct inverse regularized inverse

Figure 3.6: An image, a blurred version, inverse by Fourier deconvolution, and the Tikhonov regularized reconstruction.

```
blurred = conv2(frog,ones(5,5)/25,'same');.
```

(`conv2` implements 2-dimensional convolution with the point-spread function `ones(5,5)/25` that is normalized so constant images are unchanged. The argument `'same'` returns the portion of the convolution that is the same size as the original image. No noise was added.

Fourier deconvolution was implemented as for the example in section 1.4, with the result shown as 'direct inverse'. As you can see, it looks awful. Can you work out what went wrong?

The problem is that the forward map is not *exactly* convolution, because of the truncation of the convolution result using the `'same'` argument. That truncation means that in the border of width 4 pixels[1] around the image, the forward map is not simply a shifted version of the forward map in the middle of the image. Hence the Fourier transform does not *exactly* diagonalize the forward map, and the inversion is not exact. That error in the model for the forward map – that is only inaccurate in the 4 pixel wide border – is enough to produce artifacts throughout the image, that has a hash pattern related to the width of the border.

We have to conclude that model error can make the inverse no use!

You might wonder why the same problem did not occur in the deblurring example in section 1.4. We avoided this problem by cooking the example, by putting a border of zeros in the original text image, with width greater than the width of the point spread function. Then when the blurred image is zero padded, the padded values actually fill in the missing values very accurately. For the frog image, zero padding does not fill in the truncated values accurately.

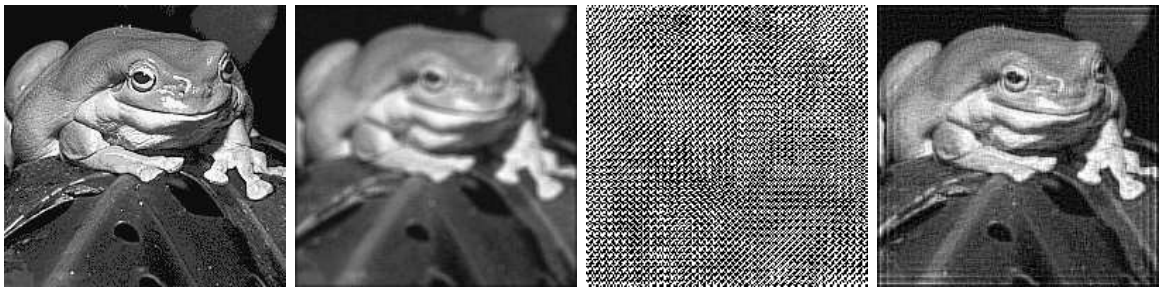
Figure 3.7 shows a similar tableau of images,with the original (true) image on the left. This time the blurred version was created using the MatLab command

```
blurred = conv2(frog,ones(5,5)/25,'same') + 10*rand(size(frog));
```

so noise has also been added to the blurred image.

As you should expect, the direct inversion using Fourier deconvolution is still awful, and is somewhat worse compared to the case where no noise was added. Again, Tikhonov regularized inversion is able to produce a visually acceptable reconstruction, thought the result is not quite as good as the case where no noise is added.

[1] the point-spread function is 5 pixels wide



original blurred + noise direct inverse regularized inverse

Figure 3.7: An image, a noisy blurred version, inverse by Fourier deconvolution, and the Tikhonov regularized reconstruction.

3.9 Why look beyond least-squares and regularization?

The deblurring examples, in sections 1.4 and 3.7.1, show that regularized inversion can do a good job of damping the spurious components in a reconstruction that arise because of small singular values. At least the reconstructed image is useful in the sense that it can be identified, and the text read.

However, inversion by regularization suffers from several major drawbacks that make it of limited value in the wider setting of inverse problems. Probably the most severe limitation is that the regularized inverse is not *quantitatively* accurate in the sense that values, or properties, of the true image are not accurately recovered. We will see this property when we consider the distribution over measurement errors more closely. In particular, uncertainties in the regularized reconstruction do not have a sensible probabilistic interpretation. We will see in the following 'contrived' example, how the apparent accuracy of the reconstruction can be quite misleading, and far from the best we can do.

3.9.1 An example showing stochastic bias

Lemma 3.9.1 If z is a random variable from *any* distribution with mean μ and variance σ^2, then $\mathrm{E}\left[z^2\right] = \mu^2 + \sigma^2$.

Proof. $\mathrm{Var}\left[z\right] = \mathrm{E}\left[z^2\right] - \left(\mathrm{E}\left[z\right]\right)^2$. ∎

This is an example of how expectation does not commute with general non-linear functions. (In this case, $\left(\mathrm{E}\left[z\right]\right)^2 \neq \mathrm{E}\left[z^2\right]$.) We all know this, but somehow manage to assume it is true when applying least-squares estimation to general problems. Here is an example that explicitly demonstrates the resulting error made by least-squares estimation.

In this example, we want to estimate a scalar x_* from N noisy measurements of $\sqrt{x_*}/i$ for some value of x_*. That is, we measure

$$d_i = \frac{\sqrt{x_*}}{i} + n_i, \quad i = 1, 2, \ldots, N.$$

The least squares estimate of x_* is

$$\hat{x} = \arg\min_x \sum_{i=1}^{N} \left(d_i - \frac{\sqrt{x}}{i} \right)^2$$

given by normal equations

$$\sum_{i=1}^{N} f_i(\hat{x}) f_i'(\hat{x}) = \sum_{i=1}^{N} d_i f_i'(\hat{x})$$

where, for convenience, we defined the function $f_i(x) = \frac{\sqrt{x}}{i}$. For the particular function we are fitting, the normal equations can be solved to give the estimate

$$\hat{x} = \left[\frac{\sum_{i=1}^{N} d_i/i}{\sum_{i=1}^{N} 1/i^2} \right]^2.$$

We will denote the term in the square brackets by w. Note that w is the sum of normal random variables, hence $w \sim N\left(\sqrt{x_*}, \sigma^2/\sum_{i=1}^{N} 1/i^2\right) \to N\left(\sqrt{x_*}, 6\sigma^2/\pi^2\right)$ as $N \to \infty$. Thus, in the limit of an infinite number of measurements

$$\mathrm{E}[\hat{x}] = x_* + 6\sigma^2/\pi^2,$$

i.e. the least-squares estimate is biased. Actually, the estimate is biased for all N, and the bias actually *increases* as we make more measurements. That's not good!

Alternatively, if we want to estimate $w = \sqrt{x}$, so $f_i(w) = w/i$, then the normal equations for w give

$$\hat{w} = \frac{\sum_{i=1}^{N} d_i/i}{\sum_{i=1}^{N} 1/i^2} \qquad \mathrm{E}[\hat{w}] = \sqrt{x_*}$$

which is not biased ($\forall N$).

To summarize, in the limit $N \to \infty$,

$$\hat{w} = \tfrac{6}{\pi^2} \textstyle\sum_{i=1}^{N} d_i/i \qquad \text{is an unbiased estimate of } \sqrt{x_*}$$

$$\hat{x} = \hat{w}^2 \qquad \text{is a biased estimate of } x_*$$

Note the following:

- It is easy to remove bias in $\hat{x}$ (just subtract $6\sigma^2/\pi^2$). The resulting estimate is better in all ways (same variance but reduced bias) and is explicitly not the least-squares fit to data! In this case *the best fit to data is not the best fit to parameters.*

- The unbiased estimate of x_* is not the square of the unbiased estimate of $\sqrt{x_*}$. In undergraduate statistics courses you would learn the more general notion that: "Conditioning on estimates gives poor predictive densities".

We can also calculate the Tikhonov regularized estimates[2] in this case.

$$\hat{x}_\lambda = \arg\min_x \left(\sum_{i=1}^{N} \left(d_i - f_i\left(x\right)\right)^2 + \lambda^2 \sum_{i=1}^{N} \left(f_i\left(x\right)\right)^2 \right)$$

for $N \to \infty$ ($\hat{x}_0 = \hat{x}_{\text{ls}}$, $\hat{w}_0 = \hat{w}_{\text{ls}}$)

$$\hat{x}_\lambda = \frac{\hat{x}_\lambda}{\left(1+\lambda^2\right)^2}, \qquad \hat{w}_\lambda = \frac{\hat{w}_\lambda}{1+\lambda^2}$$

$$\mathrm{E}\left[\hat{x}_\lambda\right] = \frac{x_* + 6\sigma^2/\pi^2}{\left(1+\lambda^2\right)^2}, \qquad \mathrm{E}\left[\hat{w}_\lambda\right] = \frac{\sqrt{x_*}}{1+\lambda^2}$$

Note that now both estimates are biased. Further, the bias depends on the unknown x_*, which means it is harder to fix the bias. So, in this case, regularization has actually made the estimates worse for the purposes of quantifying errors on estimates.

3.9.2 Least-squares with uniform noise

Our second simple example that shows up problems with least-squares estimation considers estimation of a scalar quantity that is measured subject to uniform noise. Uniform noise with known range is often used as a simple model for digitization error, though we are not suggesting that for the current example.

Assume that K direct measurements of the scalar μ are made, subject to independent uniform noise with range $[-1, 1]$; that is, the i^{th} measurement is

$$d_i = \mu + n_i$$

each $n_i \sim U\left(-1, 1\right)$.

It is possible to bound the value of μ by simple considerations; Since $\mu - 1 \leq d_i \leq \mu + 1$ for all i, it follows that

$$\max\left\{d_i\right\} - 1 \leq \mu \leq \min\left\{d_i\right\} + 1.$$

This turns out to be precisely the Bayesian likelihood function for μ based on measurements $\{d_i : i = 1, 2, \ldots, K\}$.

The least-squares estimate of μ from K measurements is

$$\hat{\mu}_{\text{ls}} = \frac{1}{K} \sum_{i=1}^{K} d_i$$

and is has the mean-square-error $1/\sqrt{3K}$, that is usually quoted as the error in this estimate. How does this estimate and error perform over repeated experiments consisting of K measurements each?

The following figure shows the result of 10 experiments (numbered 1 to 10 horizontally) with the value of the $K = 10$ measured values shown vertically, for the case $\mu = 0$.

[2] Tikhonov regularized estimates are sometimes referred to as *damped least-squares* estimates.

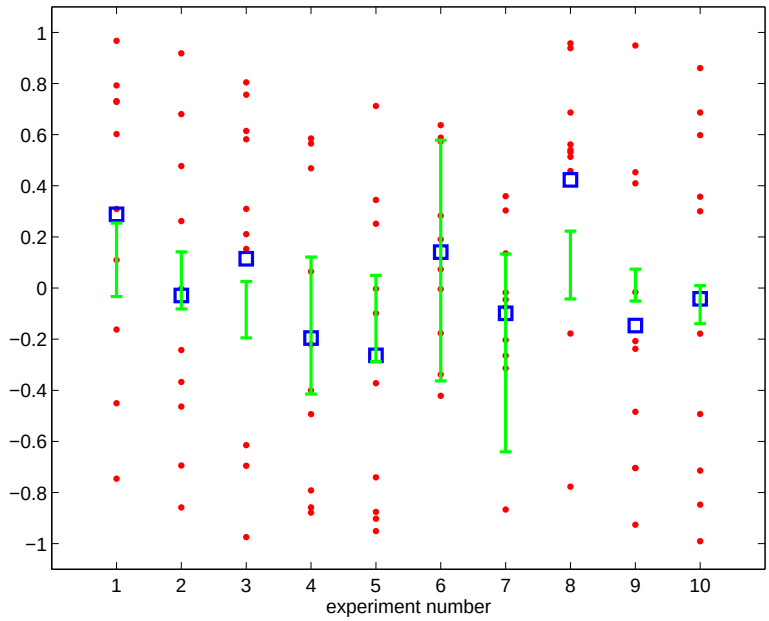


Figure 3.8: Least-squares estimates for $K = 10$ measurements subject to uniform noise of known range, for the case $\mu = 0$. Red dots: data; Squares: least-squares estimate; Green error bars: feasible interval.

In 30% of experiments (this is the theoretically derived percentage), the least-squares estimate is not even feasible. Perhaps a more serious problem, for applications with multiple noise sources, is that the least-squares 'error' of ± 0.4082 has nothing to do with actual width of the feasible interval; sometimes it is smaller, sometime larger. Note that the least-squares error *does not depend on the data set* (this is often stated as an advantage of the least-squares method) and so cannot reflect the actual length of the feasible interval, that does depend on the particular data set.

3.A Solving large systems of equations for regularization problems

This is a very large subject in its own right which we cannot treat in detail. Although the singular value decomposition (and the generalized SVD) is useful for the theoretical study of inverse problems, it is rather numerically intensive to calculate in practice. As a rough guide, finding the SVD is feasible when there are up to a few hundred variables, and so is of limited use for problems involving several thousand to several million components in the image and data vectors.

The mapping from the image to the data is specified by multiplication by the matrix $\mathbf{A}$. For large problems, it is often not feasible to store the matrix $\mathbf{A}$ in the computer, or to compute its action by direct multiplication. If for example, we take the problem of blurring an image consisting of a 256×256 pixel scene to produce a data photograph of the same size, the sizes of image and data space are each 65536 dimensional, and the matrix $\mathbf{A}$ has 256^4 elements. In the case of blurring, we know that the two-dimensional convolution corresponding to the action of $\mathbf{A}$ can be computed efficiently via the trick of multiplying together Fourier transforms, and so

the matrix $\mathbf{A}$ is never formed explicitly. When writing algorithms for solving such problems, we cannot use methods which involve manipulating the full matrices, and we should regard even the storage of vectors in image and data space being rather costly. For generality, we assume that the user will provide us with a function that will calculate $\mathbf{Av}$ and $\mathbf{Lv}$ when provided with a vector $\mathbf{v} \in \mathbb{R}^n$. As we shall see, we also need to be provided with routines to calculate $\mathbf{A}^{\mathrm{t}}\mathbf{u}$ for $\mathbf{u} \in \mathbf{R}^m$ and $\mathbf{L}^{\mathrm{t}}\mathbf{w}$ for $\mathbf{w} \in \mathbb{R}^p$. These functions will usually employ some trick such as using Fourier transforms or exploiting the sparsity of the matrices in order to do the calculation in a reasonable time.

In the above, we derived the simultaneous equations for Tikhonov regularization from the minimization of the function

$$\mathcal{L}(\mathbf{f}) = \lambda^2 \|\mathbf{L}(\mathbf{f} - \mathbf{f}^{\infty})\|^2 + \|\mathbf{d} - \mathbf{Af}\|^2$$

For the linear inverse problem, this is a *quadratic* in the image $\mathbf{f}$. Before describing the algorithm, we need to consider some properties of such quadratic expressions.

3.A.1 Minimizing a quadratic in many dimensions

A quadratic expression in a vector $\mathbf{x} \in \mathbb{R}^n$ has the general form

$$Q(\mathbf{x}) = \mathbf{x}^{\mathrm{t}}\mathbf{Hx} - \mathbf{x}^{\mathrm{t}}\mathbf{G} - \mathbf{G}^{\mathrm{t}}\mathbf{x} + Q_0 \tag{3.21}$$

where $\mathbf{H} \in \mathbb{R}^{n \times n}$ is a real symmetric matrix, $\mathbf{G} \in \mathbb{R}^n$ is a vector and $Q_0 \in \mathbb{R}$. This quadratic has a stationary point whenever

$$\frac{\partial Q}{\partial x_i} = 2\sum_j h_{ij}x_j - 2g_i = 0 \tag{3.22}$$

for all $i \in \{1, ..., n\}$. i.e., stationary points $\mathbf{x}_s$ of Q satisfy the set of linear equations

$$\mathbf{Hx}_s = \mathbf{G} \tag{3.23}$$

Depending on the nature of the matrix $\mathbf{H}$, this can have none, one or infinitely many solutions. A special case of interest is when $\mathbf{H}$ is *positive definite,* which means that $\mathbf{x}^{\mathrm{t}}\mathbf{Hx} \geq 0$ for all $\mathbf{x}$, and that $\mathbf{x}^{\mathrm{t}}\mathbf{Hx} = 0$ only if $\mathbf{x} = \mathbf{0}$. If $\mathbf{H}$ is positive definite, it is invertible and there is a unique stationary point

$$\mathbf{x}_s = \mathbf{H}^{-1}\mathbf{G} \tag{3.24}$$

The original quadratic may be written as

$$Q(\mathbf{x}) = (\mathbf{x} - \mathbf{x}_s)^{\mathrm{t}}\mathbf{H}(\mathbf{x} - \mathbf{x}_s) + \left(Q_0 - \mathbf{x}_s^{\mathrm{t}}\mathbf{Hx}_s\right) \tag{3.25}$$

from which it is clear that $\mathbf{x}_s$ is the global minimum of Q. When $\mathbf{H}$ is positive definite, we may regard solving the system of equations (3.23) and minimizing the quadratic (3.21) as being equivalent problems.

Let us now consider the problem of minimizing $Q(\mathbf{x})$ when n is so large that computing $\mathbf{H}^{-1}$ explicitly is not feasible. We can consider an iterative algorithm which proceeds from an initial guess $\mathbf{x}_0$ of the minimum to a point $\mathbf{x}_1$ which is hopefully a better estimate of $\mathbf{x}_s$. One way of doing this is to consider the direction of *steepest descent* from $\mathbf{x}_0$. This is along the direction opposite to the gradient of Q at $\mathbf{x}_0$. We see that

$$\nabla Q = 2(\mathbf{Hx} - \mathbf{G})$$

and so the direction of steepest descent is along $-\nabla Q$ which is parallel to $\mathbf{s}_1 = \mathbf{G} - \mathbf{H}\mathbf{x}_0$. We now proceed along the line $\mathbf{x}_0 + c_1\mathbf{s}_1$ trying to find a better approximation to $\mathbf{x}_s$. It is easy to find out how Q behaves on this line, since

$$\begin{aligned} Q\left(\mathbf{x}_0 + c_1\mathbf{s}_1\right) &= \left(\mathbf{x}_0 + c_1\mathbf{s}_1\right)^{\mathrm{t}} \mathbf{H}\left(\mathbf{x}_0 + c_1\mathbf{s}_1\right) - \left(\mathbf{x}_0 + c_1\mathbf{s}_1\right)^{\mathrm{t}} \mathbf{G} - \mathbf{G}^{\mathrm{t}}\left(\mathbf{x}_0 + c_1\mathbf{s}_1\right) + Q_0 \\ &= \left(\mathbf{s}_1^{\mathrm{t}}\mathbf{H}\mathbf{s}_1\right) c_1^2 - \left\{\mathbf{s}_1^{\mathrm{t}}\left(\mathbf{G} - \mathbf{H}\mathbf{x}_0\right) + \left(\mathbf{G} - \mathbf{H}\mathbf{x}_0\right)^{\mathrm{t}} \mathbf{s}_1\right\} c_1 + Q\left(\mathbf{x}_0\right) \end{aligned} \tag{3.26}$$

This is a quadratic in c_1 whose minimum is readily found to be at

$$c_1 = \frac{\mathbf{s}_1^{\mathrm{t}}\left(\mathbf{G} - \mathbf{H}\mathbf{x}_0\right) + \left(\mathbf{G} - \mathbf{H}\mathbf{x}_0\right)^{\mathrm{t}} \mathbf{s}_1}{2\left(\mathbf{s}_1^{\mathrm{t}}\mathbf{H}\mathbf{s}_1\right)} = \frac{\mathbf{s}_1^{\mathrm{t}}\mathbf{s}_1}{\mathbf{s}_1^{\mathrm{t}}\mathbf{H}\mathbf{s}_1} \tag{3.27}$$

We now set $\mathbf{x}_1 = \mathbf{x}_0 + c_1\mathbf{s}_1$ as our next estimate of the position of $\mathbf{x}_s$. Notice that in order to compute c_1, all that we need to be able to do is to calculate $\mathbf{H}\mathbf{s}_1$ and to carry out arithmetic with *vector* quantities. So long that $\mathbf{H}\mathbf{s}_1$ can be computed efficiently, no large matrices need to be stored. We can proceed iteratively, finding $-\nabla Q\left(\mathbf{x}_1\right)$ and searching along this direction from $\mathbf{x}_1$ to find the point $\mathbf{x}_2$ which minimizes Q along this line. This is known as the *steepest descent algorithm* for minimizing a function. At each iteration, a one dimensional search is carried out, and the hope is that a succession of these will ultimately lead to the minimization of the n dimensional quadratic. Despite its intuitive appeal, it is a very inefficient way of minimizing a function of many variables. Unless the contours of Q are spherical, it requires many more than n iterations to find the minimum due to a phenomenon called "hem-stitching" in which the succession of iterates slowly crawls down the walls of a long elliptical valley. In effect, each successive search largely undoes the work of the previous step, and the result is only a very gradual reduction in Q. For large n, this algorithm is essentially useless.

Instead of starting at an initial guess and searching along a single line for the minimum of Q, we can consider searching in a larger *subspace* starting from an initial guess. For the moment, suppose that someone gives us a list of linearly independent *search directions,* $\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_k$ and asks us to minimize Q within the subspace

$$\mathbf{x}_0 + c_1\mathbf{s}_1 + ... + c_n\mathbf{s}_n = \mathbf{x}_0 + \mathbf{S}\mathbf{c} \tag{3.28}$$

where $\mathbf{S}$ is the matrix whose columns are the search directions and $\mathbf{c}$ is the column vector of the coefficients. Within this k dimensional affine space, we see that

$$Q\left(\mathbf{x}_0 + \mathbf{S}\mathbf{c}\right) = \left(\mathbf{x}_0 + \mathbf{S}\mathbf{c}\right)^{\mathrm{t}} \mathbf{H}\left(\mathbf{x}_0 + \mathbf{S}\mathbf{c}\right) - \left(\mathbf{x}_0 + \mathbf{S}\mathbf{c}\right)^{\mathrm{t}} \mathbf{G} - \mathbf{G}^{\mathrm{t}}\left(\mathbf{x}_0 + \mathbf{S}\mathbf{c}\right) + Q_0 \tag{3.29}$$

$$= \mathbf{c}^{\mathrm{t}}\left(\mathbf{S}^{\mathrm{t}}\mathbf{H}\mathbf{S}\right)\mathbf{c} - \mathbf{c}^{\mathrm{t}}\mathbf{S}^{\mathrm{t}}\left(\mathbf{G} - \mathbf{H}\mathbf{x}_0\right) - \left(\mathbf{G} - \mathbf{H}\mathbf{x}_0\right)^{\mathrm{t}}\mathbf{S}\mathbf{c} + \left(Q_0 + \mathbf{x}_0^{\mathrm{t}}\mathbf{H}\mathbf{x}_0 - \mathbf{x}_0^{\mathrm{t}}\mathbf{G} - \mathbf{G}^{\mathrm{t}}\mathbf{x}_0\right) \tag{3.30}$$

$$= \mathbf{c}^{\mathrm{t}}\tilde{\mathbf{H}}\mathbf{c} - \mathbf{c}^{\mathrm{t}}\tilde{\mathbf{G}} - \tilde{\mathbf{G}}^{\mathrm{t}}\mathbf{c} + \tilde{Q}_0 \tag{3.31}$$

which is also a quadratic in $\mathbf{c}$, with the matrices

$$\tilde{\mathbf{H}} = \mathbf{S}^{\mathrm{t}}\mathbf{H}\mathbf{S} \tag{3.32}$$

$$\tilde{\mathbf{G}} = \mathbf{S}^{\mathrm{t}}\left(\mathbf{G} - \mathbf{H}\mathbf{x}_0\right) \tag{3.33}$$

$$\tilde{Q}_0 = Q\left(\mathbf{x}_0\right) = Q_0 + \mathbf{x}_0^{\mathrm{t}}\mathbf{H}\mathbf{x}_0 - \mathbf{x}_0^{\mathrm{t}}\mathbf{G} - \mathbf{G}^{\mathrm{t}}\mathbf{x}_0 \tag{3.34}$$

Since the columns of $\mathbf{S}$ are linearly independent and the matrix $\mathbf{H}$ is positive definite, so is the matrix $\tilde{\mathbf{H}}$. By the above discussion, the minimum of $Q\left(\mathbf{x}_0 + \mathbf{S}\mathbf{c}\right)$ in this affine subspace is located at

$$\hat{\mathbf{c}} = \tilde{\mathbf{H}}^{-1}\tilde{\mathbf{G}} \tag{3.35}$$

This can be readily computed since the matrix $\tilde{\mathbf{H}}$ is only of size $k \times k$ where k is the number of search directions in the subspace. When using a subspace search, the next guess at the minimizing point $\mathbf{x}_s$ is

$$\mathbf{x}_1 = \mathbf{x}_0 + \mathbf{S}\hat{\mathbf{c}} = \mathbf{x}_0 + \mathbf{S}\tilde{\mathbf{H}}^{-1}\tilde{\mathbf{G}} \tag{3.36}$$

We can then proceed iteratively to get a succession of estimates to $\mathbf{x}_s$.

The efficiency of the resulting algorithm depends critically on making an appropriate choice for the search directions. One might imagine starting at $\mathbf{x}_0$ and searching in the direction of $\mathbf{s}_1 = -\nabla Q(\mathbf{x}_0)$, thus reaching $\mathbf{x}_1$. From there, we find $\mathbf{s}_2 = -\nabla Q(\mathbf{x}_1)$, but instead of simply searching along this line, we can search in the affine subspace spanned by $\mathbf{s}_1$ and $\mathbf{s}_2$ starting at $\mathbf{x}_1$. Having found the minimum of Q in this subspace at $\mathbf{x}_2$, we then set $\mathbf{s}_3 = -\nabla Q(\mathbf{x}_2)$ and search the subspace spanned by $\mathbf{s}_1$, $\mathbf{s}_2$ and $\mathbf{s}_3$. In this way, we search for the minimum over larger and larger spaces, which ensures that each search does not undo the work of the previous searches. The spaces generated in this manner are known as the *Krylov subspaces.* The algorithm, as described would rapidly become unworkable since the search space increases in size on each iteration. However, due to a truly amazing result which we do not have time to prove, it turns out that for quadratics, it is possible to achieve the same result as searching over all the space spanned by all the previous search directions by searching firstly along $-\nabla Q(\mathbf{x}_0)$ and subsequently over only a two-dimensional space on each iteration.

Suppose we have reached the point $\mathbf{x}_l$ after the l'th iteration. We calculate $-\nabla Q(\mathbf{x}_l)$ as before and search within the affine space spanned by this vector and the vector $\mathbf{x}_l - \mathbf{x}_{l-1}$. It can be shown that in the absence of round-off errors, the effect is the same as if we had searched in the space spanned by $-\nabla Q(\mathbf{x}_0), -\nabla Q(\mathbf{x}_1), ..., -\nabla Q(\mathbf{x}_l)$. With this algorithm and perfect arithmetic, one can reach the minimum of an n dimensional quadratic in at most n steps.

The following Matlab code illustrates how a search can be carried out over an arbitrary affine subspace around the current point. The user specifies the search directions as the columns of the matrix $\mathbf{S}$.

```
function [xnew,Qpred] = search1(x0,res,Hfunc,Qnow,S)
% Searches in an affine subspace for the minimum of the
%  quadratic
%   Q(x) = x'*H*x-x'*G-G'*x+Q0
% x0    = Initial guess of location of the minimum
% res   = G-H*x0
% Hfunc = Name of function which calculates H*x for given x
% Qnow  = Value of Q(x0)
% S     = matrix with search directions as its columns

% Sze Tan, University of Auckland, July 1998

nsrch = size(S,2);
HS = zeros(length(x0),nsrch);
fprintf('Value of quadratic (current)   = %f\n',Qnow);
for k = 1 : nsrch
   HS(:,k) = feval(Hfunc,S(:,k));
end
```

```
Hq = S'*HS;
Gq = S'*res;
c  = Hq\Gq;
Qpred = Qnow + c'*Hq*c - c'*Gq - Gq'*c;
fprintf('Value of quadratic (predicted) = %f\n',Qpred);
xnew  = x0 + S * c;
```

In this code, the function whose name is in `Hfunc` is used to apply the matrix $\mathbf{H}$ to an arbitrary vector. In general, this will compute the product in some indirect way for efficiency. However, as an illustrative example, the following shows how we can solve the problem $\mathbf{Hx} = \mathbf{G}$ for a small matrix $\mathbf{H}$ by making the function `Hfunc` do an explicit matrix multiplication. Note that $\mathbf{H}$ has to be positive definite for the algorithm to work. In this algorithm, we predict the value that $Q(\mathbf{x})$ is going to have on the next iteration by examining its behaviour in the subspace. On the next iteration the value of Q is recalculated at this point and it is a good check to see that the actual value agrees with the predicted value.

```
global H
% Set up a random positive definite matrix H and a right-hand side
%  of the equation
neq = 20;
H = rand(neq,neq);
H = H'*H + 1e-6*eye(neq,neq);
G = randn(neq,1);
% Hmult is the name of a simple function that multiplies by H
Hfunc = 'Hmult';
S = [];
% Random starting guess
x0 = randn(neq,1);
while 1,
   Hx = feval(Hfunc,x0);
   res = G - Hx;
   fprintf('Norm of residual = %f\n',norm(res));
   Qnow = x0'*Hx - x0'*G - G'*x0;
   S(:,1) = 2*res; % Search direction along negative gradient
   [xnew,Qpred] = search1(x0,res,Hfunc,Qnow,S);
   S(:,2) = xnew - x0; % Second search direction
   x0 = xnew;
   keyboard % Type "return" to continue to next iteration
end
```

Notice how the matrix of search directions $\mathbf{S}$ is set up. On the first iteration, it consists of a single column containing $2(\mathbf{G} - \mathbf{Hx}_0)$. This is the negative gradient at the starting guess. After the first iteration, the second column of $\mathbf{S}$ is set to $\mathbf{x}_1 - \mathbf{x}_0$. On the second iteration, the first column is set to $2(\mathbf{G} - \mathbf{Hx}_1) \equiv -\nabla Q(\mathbf{x}_1)$ so that the function `search1` finds the minimum in the two dimensional space spanned by $-\nabla Q(\mathbf{x}_1)$ and $\mathbf{x}_1 - \mathbf{x}_0$, as required. This process continues on subsequent iterations.

The function `Hmult` is simply

```
function y = Hmult(x)
global H
y = H*x;
```

3.A.2 Application to Tikhonov Regularization

For Tikhonov regularization, we need to find the minimum of the quadratic expression

$$\mathcal{L}(\mathbf{f}) = \lambda^2 \|\mathbf{L}(\mathbf{f}-\mathbf{f}^\infty)\|^2 + \|\mathbf{d}-\mathbf{A}\mathbf{f}\|^2 \tag{3.37}$$

$$= \lambda^2 \Omega(\mathbf{f}) + C(\mathbf{f}) \tag{3.38}$$

where we shall assume that λ has been given. One can apply the algorithm discussed above directly to this problem by multiplying out the norms in order to find the matrices $\mathbf{H}$ and $\mathbf{G}$, but it is convenient to use the special form of the expression and to assume that the user can provide functions which will apply the forward operator $\mathbf{A}$ and the operator $\mathbf{L}$ to any given vector.

Starting from $\mathbf{f}_0$ and searching within a subspace spanned by the columns of $\mathbf{S}$ as before, we wish to consider

$$\mathcal{L}(\mathbf{f}_0+\mathbf{S}\mathbf{c}) = \lambda^2 \Omega(\mathbf{f}_0+\mathbf{S}\mathbf{c}) + C(\mathbf{f}_0+\mathbf{S}\mathbf{c}) \tag{3.39}$$

Substituting into the expression for Ω, we find

$$\Omega(\mathbf{f}_0+\mathbf{S}\mathbf{c}) = \|\mathbf{L}(\mathbf{f}_0+\mathbf{S}\mathbf{c}-\mathbf{f}^\infty)\|^2 \tag{3.40}$$

$$= (\mathbf{f}_0+\mathbf{S}\mathbf{c}-\mathbf{f}^\infty)^{\mathrm{t}}\mathbf{L}^{\mathrm{t}}\mathbf{L}(\mathbf{f}_0+\mathbf{S}\mathbf{c}-\mathbf{f}^\infty) \tag{3.41}$$

$$= \mathbf{c}^{\mathrm{t}}\mathbf{S}^{\mathrm{t}}\mathbf{L}^{\mathrm{t}}\mathbf{L}\mathbf{S}\mathbf{c} + \mathbf{c}^{\mathrm{t}}\mathbf{S}^{\mathrm{t}}\mathbf{L}^{\mathrm{t}}\mathbf{L}(\mathbf{f}_0-\mathbf{f}^\infty) + (\mathbf{f}_0-\mathbf{f}^\infty)^{\mathrm{t}}\mathbf{L}^{\mathrm{t}}\mathbf{L}\mathbf{S}\mathbf{c} + \Omega(\mathbf{f}_0) \tag{3.42}$$

$$= \mathbf{c}^{\mathrm{t}}\tilde{\mathbf{H}}_\Omega\mathbf{c} - \mathbf{c}^{\mathrm{t}}\tilde{\mathbf{G}}_\Omega - \tilde{\mathbf{G}}_\Omega^{\mathrm{t}}\mathbf{c} + \Omega(\mathbf{f}_0) \tag{3.43}$$

where

$$\tilde{\mathbf{H}}_\Omega = \mathbf{S}^{\mathrm{t}}\mathbf{L}^{\mathrm{t}}\mathbf{L}\mathbf{S} = (\mathbf{L}\mathbf{S})^{\mathrm{t}}(\mathbf{L}\mathbf{S}) \tag{3.44}$$

$$\tilde{\mathbf{G}}_\Omega = -\mathbf{S}^{\mathrm{t}}\mathbf{L}^{\mathrm{t}}\mathbf{L}(\mathbf{f}_0-\mathbf{f}^\infty) = -(\mathbf{L}\mathbf{S})^{\mathrm{t}}\mathbf{L}(\mathbf{f}_0-\mathbf{f}^\infty) \tag{3.45}$$

Similarly using the expression for C, we find

$$C(\mathbf{f}_0+\mathbf{S}\mathbf{c}) = \|\mathbf{d}-\mathbf{A}(\mathbf{f}_0+\mathbf{S}\mathbf{c})\|^2 \tag{3.46}$$

$$= \{\mathbf{d}-\mathbf{A}(\mathbf{f}_0+\mathbf{S}\mathbf{c})\}^{\mathrm{t}}\{\mathbf{d}-\mathbf{A}(\mathbf{f}_0+\mathbf{S}\mathbf{c})\} \tag{3.47}$$

$$= \mathbf{c}^{\mathrm{t}}\mathbf{S}^{\mathrm{t}}\mathbf{A}^{\mathrm{t}}\mathbf{A}\mathbf{S}\mathbf{c} - \mathbf{c}^{\mathrm{t}}\mathbf{S}^{\mathrm{t}}\mathbf{A}^{\mathrm{t}}(\mathbf{d}-\mathbf{A}\mathbf{f}_0) - (\mathbf{d}-\mathbf{A}\mathbf{f}_0)^{\mathrm{t}}\mathbf{A}\mathbf{S}\mathbf{c} + C(\mathbf{f}_0) \tag{3.48}$$

$$= \mathbf{c}^{\mathrm{t}}\tilde{\mathbf{H}}_C\mathbf{c} - \mathbf{c}^{\mathrm{t}}\tilde{\mathbf{G}}_C - \tilde{\mathbf{G}}_C^{\mathrm{t}}\mathbf{c} + C(\mathbf{f}_0) \tag{3.49}$$

where

$$\tilde{\mathbf{H}}_C = \mathbf{S}^{\mathrm{t}}\mathbf{A}^{\mathrm{t}}\mathbf{A}\mathbf{S} = (\mathbf{A}\mathbf{S})^{\mathrm{t}}(\mathbf{A}\mathbf{S}) \tag{3.50}$$

$$\tilde{\mathbf{G}}_C = \mathbf{S}^{\mathrm{t}}\mathbf{A}^{\mathrm{t}}(\mathbf{d}-\mathbf{A}\mathbf{f}_0) = (\mathbf{A}\mathbf{S})^{\mathrm{t}}\mathbf{r}_0 \text{ where } \mathbf{r}_0 = \mathbf{d}-\mathbf{A}\mathbf{f}_0 \tag{3.51}$$

Thus

$$\mathcal{L}(\mathbf{f}_0+\mathbf{S}\mathbf{c}) = \mathbf{c}^{\mathrm{t}}\left(\lambda^2\tilde{\mathbf{H}}_\Omega + \tilde{\mathbf{H}}_C\right)\mathbf{c} - \mathbf{c}^{\mathrm{t}}\left(\lambda^2\tilde{\mathbf{G}}_\Omega + \tilde{\mathbf{G}}_C\right) - \left(\lambda^2\tilde{\mathbf{G}}_\Omega + \tilde{\mathbf{G}}_C\right)^{\mathrm{t}}\mathbf{c} + \mathcal{L}(\mathbf{f}_0) \tag{3.52}$$

The minimum in the subspace occurs where

$$\hat{\mathbf{c}} = \left(\lambda^2 \tilde{\mathbf{H}}_\Omega + \tilde{\mathbf{H}}_C\right)^{-1} \left(\lambda^2 \tilde{\mathbf{G}}_\Omega + \tilde{\mathbf{G}}_C\right) \tag{3.53}$$

and so we set

$$\mathbf{f}_1 = \mathbf{f}_0 + \mathbf{S}\left(\lambda^2 \tilde{\mathbf{H}}_\Omega + \tilde{\mathbf{H}}_C\right)^{-1} \left(\lambda^2 \tilde{\mathbf{G}}_\Omega + \tilde{\mathbf{G}}_C\right) \tag{3.54}$$

These considerations are illustrated in the Matlab program included below

```
function [fnew,cpred,wpred] = subsearch(f0,res,fdef,Afunc,Lfunc,lambda,S)
% Searches in subspace spanned by columns of S for the optimal solution
%  of the regularization problem
%   (lambda)^2*||L*(f-fdef)||^2 + ||d-A*f||^2
% f0     = Initial guess at location of the minimum
% res    = d - A*f0, the current residual
% fdef   = The default image
% Afunc  = Name of function which applies A to a vector
% Lfunc  = Name of function which applies L to a vector
% lambda = Weighting between solution seminorm and data misfit
% S      = matrix with search directions as its columns
%
% fnew   = Position of minimum within subspace
% cpred  = Predicted value of ||d-A*fnew||^2
% wpred  = Predicted value of ||L*(fnew-fdef)||^2

% Sze Tan, University of Auckland, July 1998

nsrch = size(S,2);
pref  = feval(Lfunc,f0-fdef);
w0 = pref' * pref;
c0 = res' * res;
fprintf('Square of regularization semi-norm (current)   = %f\n',w0);
fprintf('Square of data misfit norm (current)           = %f\n',c0);
AS = zeros(length(res),nsrch);
LS = zeros(length(pref),nsrch);
for k = 1 : nsrch
   AS(:,k) = feval(Afunc,S(:,k));
   LS(:,k) = feval(Lfunc,S(:,k));
end
Hc = AS' * AS;  Hw = LS' * LS;
Gc = AS' * res; Gw = -LS'*pref;
c = (Hc + lambda^2 * Hw) \ (Gc + lambda^2 *Gw);
cpred = c0 + c'*Hc*c - c'*Gc - Gc'*c;
wpred = w0 + c'*Hw*c - c'*Gw - Gw'*c;
fprintf('Square of regularization semi-norm (predicted) = %f\n',wpred);
fprintf('Square of data misfit norm (predicted)         = %f\n\n',cpred);
fnew = f0 + S * c;
```

Notice that for the minimization in the subspace, it is only necessary to be able to apply $\mathbf{A}$ and $\mathbf{L}$ to vectors. It only remains for us to calculate the search directions for the minimization. The negative gradient of $\mathcal{L}$ is

$$-\nabla\mathcal{L}(\mathbf{f}) = -\lambda^2\mathbf{L}^{\mathrm{t}}\mathbf{L}(\mathbf{f}-\mathbf{f}^{\infty}) + \mathbf{A}^{\mathrm{t}}(\mathbf{d}-\mathbf{A}\mathbf{f})$$

In order to calculate this, we also need functions which will apply $\mathbf{L}^{\mathrm{t}}$ and $\mathbf{A}^{\mathrm{t}}$ to a vector. A method which works reasonably well in practice is to calculate $\mathbf{L}^{\mathrm{t}}\mathbf{L}(\mathbf{f}_l-\mathbf{f}^{\infty})$, $\mathbf{A}^{\mathrm{t}}(\mathbf{d}-\mathbf{A}\mathbf{f}_l)$ as search directions on the first iteration ($l=0$), and to append the direction $\mathbf{f}_l-\mathbf{f}_{l-1}$ on subsequent iterations. This is illustrated in the code fragment below:

```
S = [];
while 1,
   pref = feval(Lfunc,f - fdef);
   res  = data - feval(Afunc,f);
   S(:,1) = feval(Ahfunc,res); S(:,2) = -feval(Lhfunc,pref);
   test = 1 - abs(S(:,1)'*S(:,2)./(norm(S(:,1))*norm(S(:,2))));
   fprintf('Test statistic = %f\n',test);
   [fnew,cpred,spred] = subsearch(f,res,fdef,Afunc,Lfunc,lambda,S);
   S(:,3) = fnew - f;
   f = fnew;
   keyboard   % Pause to allow user to view result
end
```

In this example, the functions whose names are in `Afunc` and `Lfunc` apply the matrices $\mathbf{A}$ and $\mathbf{L}$ to a vector, while the functions whose names are in `Ahfunc` and `Lhfunc` apply the (conjugate) transposes ($\mathbf{A}^{\mathrm{H}}$ and $\mathbf{L}^{\mathrm{H}}$) to a vector. The search directions are placed in the columns of $\mathbf{S}$. The quantity `test` indicates whether the vectors ∇C and $\nabla\Omega$ are parallel to each other, as they should be at the optimal point. When the value of `test` is sufficiently small, the algorithm may be deemed to have converged.

4

Elements of Probability and Statistics

"the true logic for this world is the calculus of Probabilities, which takes account of the magnitude of the probability which is, or ought to be, in a reasonable man's mind." *James Clerk Maxwell, 1850*

4.1 The role of probability in inverse problems

So far we have looked at the deterministic part of inverse problem theory, which involves examining how an 'image' is transformed into the 'data' via the physics of the measurement process. That defines the *forward map*, with the naïve reconstruction method being to invert that map.

When the forward map is ill-posed, or ill-conditioned, direct inversion leads to a poor solution since errors on the measured data, and other uncertainties, are amplified by the inverse map to the point where the recovered image is dominated by those errors. The methods of regularization and truncation in chapter 3 provide algorithms to tame ill-posed computational problems, and can produce very useful solutions, as we have seen. But a crucial component is still missing: how to produce a proper analysis of the certainty, or rather uncertainty, of the estimates? How much, indeed, can we trust the predictions given by our models, often simulating complex physical systems? Here is the main contribution that probilistic methods can provide.

The examples in section 3.9 showed how regularization can perform very poorly in terms of quantifying uncertainties in estimates, even in very simple examples. We discovered that, in the presence of uncertainties, there is no single notion of 'best'. Thus, the paradigm that "inverse problems can be solved by selecting a criterion for 'best' and then optimizing with respect to it" is fundamantally flawed (though may be useful in some cases).

All available data contains measurement errors, so the estimated image is uncertain to some degree. A natural question then arises: if measurement noise corrupting the data follows some statistics, what is the *distribution* over the possible solutions after the estimation procedure? The same question can be extended to model error: given that our model of the physical measurement process is always just that, a model, what is the uncertainty in a reconstructed image or in subsequently calculated properties of the image?

We take a 'Bayesian' approach, in which all uncertainties are modelled via probability. The main sources of uncertainty that we will consider are: measurement errors, model misspecification, and *a priori* uncertainty in the solution[1]. We do not distinguish between the

[1] If we were certain about the solution in advance, there would be no difficulty in finding the solution.

possible origins of uncertainty, i.e., whether the true value is unknowable, or we are simply unwilling or unable to determine its value. This means that *all* uncertain quantities are modelled as a *random variable* with a distribution of its own. The (*a priori*) stochastic modelling of the unknown image provides a flexible way of stating constraints or knowledge about the image, and generalizes the role of the regularizing functional by supplying information about the unknown image in directions that are not determined by the measurements.

Bayesian analysis adopts the *subjective* notion of probability, emphasizing, as we have above, the *state of knowledge* we can have about uncertain quantities or predictions. This fits the physics and engineering notion of learning from measurement that corresponds to refining knowledge about an underlying physical reality. Many Bayesian physicists adopt Bayesian methods because of the work of the physicist R. T. Cox who proved that any consistent manipulation of degrees of plausibility represented by real numbers is equivalent to the Bayesian calculus.

An alternative is the *frequentist* notion that probability refers to *frequency in a random experiment*. The regularization methods we have seen are exactly the 'ridge regression' estimators in frequentist statistics. A lively debate has taken place within the discipline of statistics for the past hunderd years, or so, over which of the Bayesian and frequentist notions is 'correct'. A practically oriented researcher might find the dispute somewhat academic, since in a real modeling project we are seldom concerned about the 'true' interpretation of parameters, notwithstanding that estimates for unknowns should be physically plausible. More often we are interested in the reliability of model predictions. The debate in statistics seems to be reaching a maturity, with some hints that the outcome may be that Bayesian and frequentist notions represent contrasting, in some sense 'dual', paradigms for inference.

The goal in any probabilistic analysis is to characterise *all* solutions that are consistent with data and models. This is quite different to the deterministic methods that seek a single 'best' solution. The distribution over possible solutions is usually too unweildy to be considered a useful solution. Instead, useful solutions are provided by summary statistics that give estimates and uncertainties of the unknown true image, and other quantities of interest. The task of calculating summary statistics is always well defined, even if the original inverse problem is ill-posed. In particular, there being no need to formulate an inverse with a unique solution, as in regularization methods.

One substantial *practical* advantage of the Bayesian framework is that it allows our inference to be informed by physical modelling, in a way that permits not only useful engineering solutions but also valid scientific answers as well. It is strightforward to consider non-linear forward problems and non-additive noise processes, as well as the mid-level and high-level representations.Those stochastic models for the unknown image have a role that is analagous to the regularizing functional, of providing information about the unknown image for aspects that are not determined by the measurements. These advantages come at the cost of increased modelling and computation. Those features of Bayesian modeling, inference, and computation will be discussed in later chapters. In the remainder of this chapter we develop some of the elementary tools from probability and statistics that we will need.

4.2 Random variables and their properties

A (scalar, vector-valued) quantity with uncertain value is acalled a (scalar, multivariate) *random variable*.

For example, the outcome of a coin toss is uncertain before the coin is tossed, and could take either of the values 'heads' or 'tails'. If we denote the outcome by X, then X is a random variable taking values in the set $\{\text{heads}, \text{tails}\}$. After the coin toss, X will take one of these values, that we call a *realization* of the random variable. When we are being careful, we denote random variables by an *upper case* letter, such as X above, and realizations by the lower case letter, such as $x =$ 'heads'.

The probability of the outcome that the random variable X takes the value x is written $\mathbb{P}(X = x)$ which is read as 'the probability that X is equal to x'. For a fair coin, this probability is $1/2$ in both of the cases $x =$ 'heads' and $x =$ 'tails'. More generally, we write the probability that X lies in some set of outcomes A, called an *event*, as

$$\mathbb{P}(X \in A).$$

The *distribution* over the random variable X is the rule $\mathbb{P}$ that assigns a (real, positive) probability to each event.

4.2.1 Probability density functions

Continuous random variables

The *probability density function* $\pi_X(x)$ of a real random variable X allows the probability that X lies in the range $a \leq X < b$ to be evaluated as the integral of the density function between a and b. i.e.,

$$\mathbb{P}(a \leq X < b) = \int_a^b \pi_X(x)\,\mathrm{d}x \tag{4.1}$$

Probability density functions are real, non-negative and normalized so that

$$\int_{-\infty}^{\infty} \pi_X(x)\,\mathrm{d}x = 1 \tag{4.2}$$

Discrete random variables

If we allow the probability density to be a generalized function, it is possible to use the same formalism for discrete random variables. If π_k is the probability that $X = k$ where k comes from a discrete set K, the probability density function for X is

$$\pi_X(x) = \sum_{k \in K} \pi_k \delta(x - k) \tag{4.3}$$

Multivariate random variables

The generalization to several random variables is immediate. For example if X_1, X_2, ..., X_n are random variables, the probability density $\pi_{X_1 X_2 \ldots X_n}$ is defined so that an integral over a n-dimensional region gives the joint probability that the point $(X_1, X_2, ..., X_n)$ lies in the specified region. i.e.,

$$\begin{aligned}\Pr(a_1 \leq X_1 < b_1 \text{ and } a_2 \leq X_2 < b_2 \text{ and } \ldots \text{ and } a_n \leq X_n < b_n) = \\ \int_{a_1}^{b_1} \mathrm{d}x_1 \int_{a_2}^{b_2} \mathrm{d}x_2 \ldots \int_{a_n}^{b_n} \mathrm{d}x_n\, \pi_{X_1 X_2 \ldots X_n}(x_1, x_2, ..., x_n)\end{aligned} \tag{4.4}$$

We often use a vector notation, writing $\mathbf{X}$ for the random variable and $\pi_{\mathbf{X}}(\mathbf{x})$ for the probability density.

Starting from a joint probability density, we can find the probability density of a subset of the variables by integrating over all possible values of the variable(s) we do not want, e.g.,

$$\pi_X(x) = \int_{-\infty}^{\infty} \mathrm{d}y \int_{-\infty}^{\infty} \mathrm{d}z \; \pi_{XYZ}(x, y, z) \tag{4.5}$$

This process is called *marginalization* and $\pi_X(x)$ is called a *marginal probability density*.

Given random variables X and Y, the *conditional probability* of X given Y is defined by

$$\pi_{X|Y}(x|y) = \frac{\pi_{XY}(x, y)}{\pi_Y(y)} \tag{4.6}$$

In the joint space of possible values of X and Y, we are effectively restricting our attention to cases in which $Y = y$. Out of these, we are interested in the probability that X is equal to x.

From the definition, it is easy to see that

$$\pi_{XY}(x, y) = \pi_{X|Y}(x|y)\pi_Y(y) = \pi_{Y|X}(y|x)\pi_X(x) \tag{4.7}$$

This relationship between the two conditional probabilities $\pi_{Y|X}$ and $\pi_{X|Y}$ is called *Bayes' theorem*. As we shall see later, this is a result that we will use redeatedly.

4.2.2 Cumulative distribution functions

The cumulative distribution function $\Pi_X(x)$ of a single real-valued random variable X gives the probability that X is less than some specified value, i.e.,

$$\Pr(X < x_0) = \Pi_X(x_0) \tag{4.8}$$

This is related to the probability density function $\pi_X(x)$ by

$$\Pi_X(x) = \int_{-\infty}^{x} \pi_X(\xi) \, \mathrm{d}\xi \tag{4.9}$$

It is then easy to see that

1. $\Pi_X(-\infty) = 0$
2. $\Pi_X(\infty) = 1$
3. Π_X is a monotonically non-decreasing function
4. $\pi_X(x) = \mathrm{d}\Pi_X(x)/\mathrm{d}x$

For discrete valued random variables, the cumulative distribution function has step discontinuities. This is consistent with the delta functions in the probability density function.

In several dimensions, the cumulative distribution function $\Pi_{X_1 X_2 \ldots X_n}(x_1, x_2, ..., x_n)$ is simply the probability that $X_1 < x_1$ and $X_2 < x_2$ and ... and $X_n < x_n$. Thus

$$\pi_{X_1 X_2 \ldots X_n}(x_1, x_2, ..., x_n) = \frac{\partial^n \Pi_{X_1 X_2 \ldots X_n}}{\partial x_1 \partial x_2 \ldots \partial x_n} \tag{4.10}$$

4.2.3 Transformation of a random variable

Suppose that X is a random variable and that $Y = \pi(X)$ is the random variable which is obtained by applying the function f to X. Given the probability density $\pi_X(x)$, we wish to determine the probability density $\pi_Y(y)$ of Y. It is easy to find the cumulative distribution function of Y since

$$\Pr(Y < y) = \Pr(f(X) < y) \tag{4.11}$$

$$= \int_{-\infty}^{\infty} u(y - f(x)) \pi_X(x) \, \mathrm{d}x, \tag{4.12}$$

where $u(x)$ is the unit step. The probability density of Y is found by differentiation

$$\pi_Y(y) = \frac{\partial}{\partial y} \left(\int_{-\infty}^{\infty} u(y - f(x)) \pi_X(x) \, \mathrm{d}x \right) \tag{4.13}$$

$$= \int_{-\infty}^{\infty} \delta(y - f(x)) \pi_X(x) \, \mathrm{d}x. \tag{4.14}$$

In order to be able to apply this result, we need to be able to handle δ functions with non-trivial arguments. Recall that in distribution theory, the idea is to define the action of a distribution on a test function in such a way that the usual formal algebraic manipulations can still be carried out. Let us consider the meaning of $\delta(g(x))$ where $g(x)$ is differentiable and has a single zero at x_0 at which $g'(x_0)$ is non-zero. Given a test function $\phi(x)$, we require that

$$\langle \delta(g(x)), \phi(x) \rangle = \lim_{h \to 0} \langle \delta_h(g(x)), \phi(x) \rangle = \lim_{h \to 0} \frac{1}{h} \int_{\{x : |g(x)| < h/2\}} \phi(x) \, \mathrm{d}x \tag{4.15}$$

where $\delta_h(x)$ is equal to $1/h$ in the interval $|x| < h/2$ and is zero elsewhere. Since g has an isolated zero at x_0, for sufficiently small h, the only values of x of interest are those in a small interval around x_0. Within this interval we may approximate $g(x)$ by its Taylor series about x_0, namely

$$g(x) \approx g(x_0) + (x - x_0) g'(x_0) = (x - x_0) g'(x_0) \tag{4.16}$$

and so to this order of approximation

$$|g(x)| < \frac{h}{2} \text{ iff } |x - x_0| < \frac{h}{2|g'(x_0)|} \tag{4.17}$$

Thus

$$\langle \delta(g(x)), \phi(x) \rangle = \lim_{h \to 0} \frac{1}{h} \int_{|x - x_0| < \frac{h}{2|g'(x_0)|}} \phi(x) \, \mathrm{d}x = \frac{\phi(x_0)}{|g'(x_0)|}, \tag{4.18}$$

and so under these conditions,

$$\delta(g(x)) = \frac{\delta(x - x_0)}{|g'(x_0)|}. \tag{4.19}$$

If we find that $g(x)$ has several zeros $\{x_i\}$ within the interval of integration, this readily generalizes to

$$\delta(g(x)) = \sum_i \frac{\delta(x - x_i)}{|g'(x_i)|} \tag{4.20}$$

Examples

1. Suppose that the random variable Θ is uniformly distributed in the range $[0, 2\pi)$ and that $X = \tan\Theta$. Find the probability density for X and check that it is properly normalized.

   Since Θ is uniformly distributed, we see that $\pi_\Theta(\theta) = (2\pi)^{-1}$. By the above result,

$$\pi_X(x) = \int_0^{2\pi} \delta(x - \tan\theta)\, \pi_\Theta(\theta)\, \mathrm{d}\theta \tag{4.21}$$

$$= \frac{1}{2\pi} \int_0^{2\pi} \delta(x - \tan\theta)\, \mathrm{d}\theta \tag{4.22}$$

   For a given value of $x > 0$, there are two values of θ within the range $[0, 2\pi)$ which satisfy $x - \tan\theta = 0$, namely $\theta_1 = \tan^{-1} x$ and $\theta_2 = \pi + \tan^{-1} x$. If we set $g(\theta) = x - \tan\theta$, we find that

$$g'(\theta) = -\sec^2\theta \tag{4.23}$$

   and so

$$\left|g'(\theta_1)\right| = \left|g'(\theta_2)\right| = \sec^2\left(\tan^{-1} x\right) = 1 + x^2 \tag{4.24}$$

   Thus

$$\delta(x - \tan\theta) = \frac{\delta\left(\theta - \tan^{-1} x\right)}{1 + x^2} + \frac{\delta\left(\theta - \pi - \tan^{-1} x\right)}{1 + x^2}. \tag{4.25}$$

   Substituting into the integral (4.22) yields

$$\pi_X(x) = \frac{1}{2\pi} \int_0^{2\pi} \left[\frac{\delta\left(\theta - \tan^{-1} x\right)}{1 + x^2} + \frac{\delta\left(\theta - \pi - \tan^{-1} x\right)}{1 + x^2}\right] \mathrm{d}\theta \tag{4.26}$$

$$= \frac{1}{\pi(1 + x^2)} \tag{4.27}$$

   Similarly, it is easy to check that this expression also gives the probability density for $x < 0$. The integral of $\pi_X(x)$ over all x yields unity, indicating that it is properly normalized.

2. Suppose that X is distributed with probability density

$$\pi_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \tag{4.28}$$

   and $Y = X^2$, determine the probability density of Y.

   By the theorem,

$$\pi_Y(y) = \int \delta\left(y - x^2\right) \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \mathrm{d}x \tag{4.29}$$

   For $y > 0$, we see that there are two values of x, namely $\pm\sqrt{y}$ which satisfy $y - x^2 = 0$. Furthermore we find that

$$\left[\frac{\partial}{\partial x}\left(y - x^2\right)\right]_{x=\pm\sqrt{y}} = \left[-2x\right]_{x=\pm\sqrt{y}} = \mp 2\sqrt{y} \tag{4.30}$$

   Hence

$$\delta\left(y - x^2\right) = \frac{\delta\left(x - \sqrt{y}\right)}{\left|-2\sqrt{y}\right|} + \frac{\delta\left(x + \sqrt{y}\right)}{\left|2\sqrt{y}\right|}. \tag{4.31}$$

Substituting into (4.29) yields

$$\begin{aligned}\pi_Y(y) &= \int \left(\frac{\delta\left(x-\sqrt{y}\right)+\delta\left(x+\sqrt{y}\right)}{2\sqrt{y}} \right) \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \mathrm{d}x \\ &= \frac{1}{\sigma\sqrt{2\pi y}} \exp\left(-\frac{y}{2\sigma^2}\right). \end{aligned} \tag{4.32}$$

Exercise

Show that if $\Pi_X(x)$ is the cumulative distribution function of a random variable X, it is possible to generate samples of X by starting with a random variable Y which is uniformly distributed within the range $[0,1]$ and setting $X = \Pi_X^{-1}(Y)$.

4.2.4 Multivariate Transformations

If we have a set of random variables $X_1, X_2, ..., X_n$ and a tranformation $f : \mathbb{R}^n \to \mathbb{R}^m$, we can find the joint probability density of $Y_1, Y_2, ..., Y_m$ where $Y_i = \pi_i(X_1, X_2, ..., X_n)$ by computing

$$\begin{aligned}\pi_{Y_1...Y_m}(y_1, ..., y_m) = \int ... \int \delta(y_1 - \pi_1(x_1, ..., x_n)) ... \delta(y_m - \pi_m(x_1, ..., x_n)) \\ \times \pi_{X_1...X_n}(x_1, ..., x_n)\, \mathrm{d}x_1 ... \mathrm{d}x_n \end{aligned} \tag{4.33}$$

An important example of the use of this theorem is to find the probability density of the sum of two random variables, i.e., when $Y = X_1 + X_2$. In this case

$$\begin{aligned}\pi_Y(y) &= \int\int \delta(y - (x_1 + x_2))\, \pi_{X_1X_2}(x_1, x_2)\, \mathrm{d}x_1\, \mathrm{d}x_2 \\ &= \int \pi_{X_1X_2}(x_1, y - x_1)\, \mathrm{d}x_1 \end{aligned} \tag{4.34}$$

where we have carried out the integration over x_2 which collapses due to the presence of the δ function. If further we assume that the random variables are independent, so that $\pi_{X_1X_2}(x_1, x_2) = \pi_{X_1}(x_1)\, \pi_{X_2}(x_2)$, this reduces to

$$\pi_Y(y) = \int \pi_{X_1}(x_1)\, \pi_{X_2}(y - x_1)\, \mathrm{d}x_1 \tag{4.35}$$

which is seen to be the *convolution* of the two probability densities.

4.2.5 Example: The χ^2 probability density

Consider the probability density of $Y = X_1^2 + X_2^2$ where

$$\pi_{X_1X_2}(x_1, x_2) = \frac{1}{2\pi} \exp\left(-\frac{1}{2}\left(x_1^2 + x_2^2\right)\right) \tag{4.36}$$

By the transformation rule,

$$\pi_Y(y) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \delta\left(y - x_1^2 - x_2^2\right) \exp\left(-\frac{1}{2}\left(x_1^2 + x_2^2\right)\right) \mathrm{d}x_1 \mathrm{d}x_2 \tag{4.37}$$

It is convenient to change to polar coordinates. This yields

$$\pi_Y(y) = \frac{1}{2\pi}\int_0^{2\pi}\int_0^{\infty}\delta\left(y-r^2\right)\exp\left(-\frac{r^2}{2}\right)\,r\,\mathrm{d}r\,\mathrm{d}\theta \tag{4.38}$$

Converting the delta function, we see that there the argument is zero if $r=\sqrt{y}$. At this point

$$\frac{\partial}{\partial r}\left(y-r^2\right) = -2r \tag{4.39}$$

Hence

$$\delta\left(y-r^2\right) = \frac{\delta\left(r-\sqrt{y}\right)}{2\sqrt{y}} \tag{4.40}$$

and so

$$\begin{aligned}\pi_Y(y) &= \frac{1}{2\pi}\int_0^{2\pi}\int_0^{\infty}\frac{\delta\left(r-\sqrt{y}\right)}{2\sqrt{y}}\exp\left(-\frac{r^2}{2}\right)\,r\,\mathrm{d}r\,\mathrm{d}\theta \\ &= \frac{1}{2}\exp\left(-\frac{y}{2}\right)\ \text{for } y>0\end{aligned} \tag{4.41}$$

This is called the χ^2 probability density with two degrees of freedom, being the sum of squares of two independent zero-mean unit-variance Gaussian distributions. In more dimensions, the sum of squares of N independent zero-mean unit-variance Gaussian distributions has probability density

$$\pi_Y(y) = \frac{1}{2^{N/2}\Gamma\left(N/2\right)}y^{\frac{N}{2}-1}\exp\left(-\frac{y}{2}\right) \tag{4.42}$$

which is the χ^2 probability density with N degrees of freedom. This may readily be derived from the fact that the volume element in N dimensions for integrands with spherical symmetry may be written as

$$\mathrm{d}x_1\mathrm{d}x_2....\mathrm{d}x_N = \frac{2\pi^{N/2}}{\Gamma\left(N/2\right)}r^{N-1}\,\mathrm{d}r. \tag{4.43}$$

4.2.6 Expected values

The *expected value* of a function h of the random variable X is an average of the function values $h(x)$ weighted by the probability that X takes on the value x, i.e.,

$$\mathrm{E}[h(X)] = \int_{-\infty}^{\infty}\pi_X(x)h(x)\,\mathrm{d}x \tag{4.44}$$

Similarly, if we have a function of more than one random variable, the weighted average is taken over the joint probability density of the variables, i.e.,

$$\mathrm{E}[\pi(X_1,X_2,...,X_n)] = \int_{-\infty}^{\infty}\mathrm{d}x_1\int_{-\infty}^{\infty}\mathrm{d}x_2...\int_{-\infty}^{\infty}\mathrm{d}x_n\;\pi_{X_1X_2...X_n}(x_1,x_2,...,x_n)h(x_1,x_2,...,x_n) \tag{4.45}$$

For example, the expectation value of the product of two random variables X and Y is

$$\mathrm{E}[XY] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}xy\pi_{XY}(x,y)\,\mathrm{d}x\mathrm{d}y \tag{4.46}$$

The expected value of a random variable X is called the *mean* of X, and is denoted μ_X. The expected value of $(X-\mu)^2$ is called the *variance* of X and is denoted σ_X^2. The *n'th moment* m_n of X is the expected value of X^n, i.e.,

$$m_n = \mathrm{E}[X^n] = \int_{-\infty}^{\infty} x^n \pi_X(x)\,\mathrm{d}x \tag{4.47}$$

We see that $m_0 = 1$, $m_1 = \mu$ and $m_2 = \sigma^2 + \mu^2$.

The operation of taking the expected value is linear, hence

$$\mathrm{E}[ah(X) + bg(Y)] = a\mathrm{E}[h(X)] + b\mathrm{E}[g(Y)] \tag{4.48}$$

This follows directly from the linearity of the integral.

An important pathological case

It is *not* always the case that the moments of a probability density exist and are finite. A simple example is the Cauchy probability density

$$\pi_i(x) = \frac{a}{\pi\,(a^2 + x^2)} \tag{4.49}$$

The second moment of this probability density is infinite.

4.2.7 Independent and uncorrelated random variables

Two random variables are said to be *independent* if their joint probability density is equal to the product of the individual probability densities. Thus random variables X and Y are independent if and only if

$$\pi_{XY}(x,y) = \pi_X(x)\pi_Y(y) \tag{4.50}$$

From the definition of conditional probability, X and Y are independent if and only if

$$\pi_{Y|X}(y|x) = \pi_Y(y) \tag{4.51}$$

Physically, this means that knowledge of the value of one of the random variables X gives no information about the value of the other random variable Y since our state of knowledge of Y conditional on knowing that $X = x$ is the same as if we had no information about X.

Similarly, a collection of random variables $X_1, X_2, ..., X_n$ is said to be independent iff their joint probability density function factorizes

$$\pi_{X_1 X_2 ... X_n}(x_1, x_2, ..., x_n) = \pi_{X_1}(x_1)\pi_{X_2}(x_2)...\pi_{X_n}(x_n) \tag{4.52}$$

Theorem 4.1 If X and Y are independent random variables, $\mathrm{E}[XY] = \mathrm{E}[X]\mathrm{E}[Y]$

Proof Do as an exercise.

Two random variables X and Y are said to be *uncorrelated* if $\mathrm{E}[XY] = \mathrm{E}[X]\mathrm{E}[Y]$. Thus independent random variables are uncorrelated, but the converse is **not** true.

Exercise 4.1 Construct two uncorrelated random variables which are not independent.

Example 4.1 The maximum of N independent random variables

Suppose that $X_1, ..., X_N$ are a set of N independent identically-distributed random variables each with probability density $\pi_X(x)$ and suppose that $Y = \max(X_1, ..., X_n)$. Find the probability density of Y.

It is easiest to consider the cumulative distribution function. The probability that $Y < y$ is the probability that all of X_1, $X_2, ..., X_N$ are less than y. Thus

$$\begin{aligned}\Pi_Y(y) &= \Pi_{X_1}(y)\,\Pi_{X_2}(y)\,...\Pi_{X_N}(y) \\ &= \left(\int_{-\infty}^{y} \pi_X(x)\,\mathrm{d}x\right)^N \end{aligned} \tag{4.53}$$

Differentiating to get the probability density,

$$\pi_Y(y) = \Pi_Y'(y) = N\pi_X(y)\left(\int_{-\infty}^{y} \pi_X(x)\,\mathrm{d}x\right)^{N-1} \tag{4.54}$$

4.2.8 Probability density of the sum of independent random variables

Let X and Y be random variables with joint probability function $\pi_{XY}(x, y)$. We wish to find the probability density $\pi_Z(z)$ of the random variable Z which is the sum of X and Y.

Consider first the cumulative distribution function $F_Z(z)$. By definition,

$$F_Z(z) = \Pr(Z < z) = \Pr(X + Y < z) = \int_{-\infty}^{\infty} \mathrm{d}x \int_{-\infty}^{z-x} \mathrm{d}y\, \pi_{XY}(x, y) \tag{4.55}$$

where the double integral is taken over the portion of the (x, y) plane for which $x + y < z$. Substituting $y' = x + y$ in the second integral yields

$$F_Z(z) = \int_{-\infty}^{\infty} \mathrm{d}x \int_{-\infty}^{z} \mathrm{d}y'\, \pi_{XY}\left(x, y' - x\right) \tag{4.56}$$

Differentiating with respect to z yields the desired probability density function

$$\pi_Z(z) = \int_{-\infty}^{\infty} \mathrm{d}x\, \pi_{XY}(x, z - x)\,. \tag{4.57}$$

If X and Y are *independent*, the joint density function factorizes and so

$$\pi_Z(z) = \int_{-\infty}^{\infty} \mathrm{d}x\, \pi_X(x)\, \pi_Y(z - x) = (\pi_X * \pi_Y)(z) \tag{4.58}$$

which we recognize as the convolution of the probability density functions. The characteristic function of Z is thus the product of the characteristic functions of X and Y

$$\chi_Z(s) = \chi_X(s)\chi_Y(s). \tag{4.59}$$

This result generalizes to the situation of more than two independent variables.

Another way of seeing that this result holds is by considering the algebra of expectation values

$$\begin{aligned}\chi_Z(s) &= \mathrm{E}\left[\exp(isZ)\right] = \mathrm{E}\left[\exp\left(is\left(X + Y\right)\right)\right] \\ &= \mathrm{E}\left[\exp(isX)\exp(isY)\right] = \mathrm{E}\left[\exp(isX)\right]\mathrm{E}\left[\exp(isY)\right] \\ &= \chi_X(s)\chi_Y(s). \end{aligned} \tag{4.60}$$

where the factorization of the expectation value is possible because of the independence of the random variables.

Similarly if we consider $Z = aX + bY$, you should check that $\chi_Z(s) = \chi_X(as)\chi_Y(bs)$ and that the inverse transform of this characteristic function yields the probability density

$$\pi_Z(z) = \frac{1}{|ab|} \int \pi_X\left(\frac{u}{a}\right) \pi_Y\left(\frac{z-u}{b}\right) \mathrm{d}u \tag{4.61}$$

4.3 Some special probability distributions

4.3.1 The Gaussian probability density

The random variable X is said to be *Gaussian* or *normally* distributed if its probability density is of the form

$$\pi_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{4.62}$$

where μ and σ^2 are the mean and variance of X respectively. The corresponding characteristic function is

$$\chi_X(s) = \exp(\mathrm{j}s\mu) \exp\left(-\frac{1}{2}\sigma^2 s^2\right) \tag{4.63}$$

For a Gaussian random variable, the probability density is completely specified once we know the mean and the variance.

Now consider the sum of two Gaussian distributed random variables. If μ_X, μ_Y, σ_X^2 and σ_Y^2 are the means and variances of X and Y respectively, the characteristic function of the sum $Z = X + Y$ is the product of the individual characteristic functions. That is,

$$\chi_Z(s) = \exp(\mathrm{j}s\mu_X) \exp\left(-\frac{1}{2}\sigma_X^2 s^2\right) \exp(\mathrm{j}s\mu_Y) \exp\left(-\frac{1}{2}\sigma_Y^2 s^2\right) \tag{4.64}$$

$$= \exp[\mathrm{j}s(\mu_X + \mu_Y)] \exp\left[-\frac{1}{2}(\sigma_X^2 + \sigma_Y^2)s^2\right] \tag{4.65}$$

It is easy to see that Z is also Gaussian distributed. The mean of Z is $\mu_X + \mu_Y$ and the variance of Z is $\sigma_X^2 + \sigma_Y^2$. Similarly, the sum of more than two independent Gaussian distributed random variables is also Gaussian distributed. The mean of the sum is the sum of the means and the variance of the sum is the sum of the variances.

Exercise: Note that this last result is more generally true. By linearity of the expectation value, it is easy to see that the mean of the sum of two random variables is always the sum of the individual means, whether or not the random variables are independent. Show that the variance of the sum of two random variables is equal to the sum of the variances of the individual variables, provided that the random variables are *uncorrelated*.

4.4 The central limit theorem

Let us now consider what happens when we add together N zero-mean independent identically distributed random variables. We shall assume that each of the random variables possesses n'th cumulants κ_n for every n.

Let $Z = X_1 + X_2 + ...{}_+ X_N$. It is clear that this has zero mean and that the n'th cumulant of Z is $N\kappa_n$. In particular, the variance of Z is $\sigma_Z^2 = N\kappa_2$. Consider the normalized random variable Z/σ_Z. This has unit variance and its n'th cumulant is

$$\frac{N\kappa_n}{\sigma_Z^n} = N^{1-\frac{n}{2}} \frac{\kappa_n}{\kappa_2^{n/2}} \tag{4.66}$$

As N becomes large, we see that the n'th cumulant of the normalized random variable tends to zero for all $n > 2$. We thus conclude that for large N, Z/σ_Z tends to a Gaussian random variable with zero mean and unit variance. This is a special case of the *central limit theorem*. In its more general form which applies to non-identically distributed random variables, it essentially states that

> The probability density of the sum of N well-behaved *independent* random variables tends to a Gaussian distribution whose mean is the sum of the individual means and whose variance is the sum of the individual variances

More precisely if $Z = X_1 + X_2 + ... + X_N$, $\mu = \mu_1 + \mu_2 + ... + \mu_N$ is the sum of the means and $\sigma^2 = \sigma_1^2 + \sigma_2^2 + ... + \sigma_N^2$ is the sum of the variances of $X_1, X_2, ..., X_N$, then the probability density of $(Z - \mu)/\sigma$ tends to a zero-mean Gaussian distributed variable of unit variance as N becomes large.

Note that the individual probability density functions need not be Gaussian nor identically distributed.

To make this more general statement true, it is necessary to restrict the individual random variables so that each has a finite variance and that the probability for $|X_i|$ to be large is very small. These are contained in the *Lindeberg condition* which requires that for all $t > 0$,

$$\lim_{N\to\infty} \frac{1}{\sigma^2} \sum_{i=1}^{N} \int_{|x-\mu_i|>t\sigma} \mathrm{d}x\, (x-\mu_i)^2 \pi_i(x-\mu_i) = 0 \tag{4.67}$$

where π_i is the probability density of the i'th random variable and σ^2 is the sum of the N variances.

A rigorous proof of the central limit theorem under these general conditions is quite difficult since we do not even assume the existence of the cumulants.

Notes:

1. It is interesting to repeatedly convolve a uniform probability density with itself repeatedly to see how the probability density of the sum approaches a Gaussian. If we add together 12 independent random numbers each generated from a uniform distribution in the range $\left[-\frac{1}{2}, \frac{1}{2}\right]$, the sum closely approximates a zero-mean unit variance Gaussian distributed variable. (This is sometimes used for computer generation of normal random variables, but the method is quite slow).

2. As an example showing how the central limit theorem can fail (through violation of the Lindeberg condition), consider the sum of N identical independent Cauchy distributed random variables with
$$\pi_i(x) = \frac{a}{\pi\,(a^2 + x^2)} \tag{4.68}$$

Show (as an exercise) that the variance of the Cauchy distribution is infinite and that the sum of any number of such distributions is also a Cauchy distribution and does not tend to the normal distribution.

4.5 Vector-valued random variables

A vector-valued random variable $\mathbf{X}$ with n components is simply a convenient notation for a collection of n random variables. The probability density $\pi_{\mathbf{X}}(\mathbf{x})$ is a joint probability density as defined above.

The *mean vector* (denoted by $\mu_{\mathbf{X}}$) is simply the vector of the mean values of the components of $\mathbf{X}$. This is the first moment of $\mathbf{X}$

$$\mu_{\mathbf{X}} = \mathrm{E}[\mathbf{X}] = \int \mathbf{x}\, \pi_{\mathbf{X}}(\mathbf{x})\ \mathrm{d}^n\mathbf{x} \tag{4.69}$$

The k'th component of $\mathrm{E}[\mathbf{X}]$ is $\mathrm{E}[X_k]$.

The second moment of $\mathbf{X}$ is the expectation value of products of pairs of components of $\mathbf{X}$. For an n component random vector, there are n^2 pairs which can be conveniently arranged in an $n \times n$ matrix called the correlation matrix $\mathbf{\Phi_{XX}}$

$$\mathbf{\Phi_{XX}} = \mathrm{E}[\mathbf{X}\,\mathbf{X}^t] = \int \mathbf{x}\,\mathbf{x}^t\, \pi_{\mathbf{X}}(\mathbf{x})\ \mathrm{d}^n\mathbf{x} \tag{4.70}$$

The kl'th component of $\mathbf{\Phi_{XX}}$ is $\mathrm{E}[X_k X_l]$. It is clear that the correlation matrix is symmetric.

Just as we defined the variance in the case of a scalar-valued random variable, we define the *covariance matrix* of a vector-valued random variable. This is also an $n \times n$ matrix $\mathbf{\Gamma_{XX}}$

$$\mathbf{\Gamma_{XX}} = E[(\mathbf{X} - \mu_{\mathbf{X}})(\mathbf{X} - \mu_{\mathbf{X}})^t] = \mathbf{\Phi_{XX}} - \mu_{\mathbf{X}}\mu_{\mathbf{X}}^t \tag{4.71}$$

The kl'th component of $\mathbf{\Gamma_{XX}}$ is $\mathrm{E}[X_k X_l] - \mathrm{E}[X_k]\mathrm{E}[X_l]$. Like the correlation matrix, the covariance matrix is also symmetric. The diagonal elements of the covariance matrix are the variances of the random variables.

Higher order moments are more complicated to write down as the m'th moment is a rank m tensor with n^m components. The $k_1 k_2 ... k_m$'th component of the m'th moment tensor is $\mathrm{E}[X_{k_1} X_{k_2} ... X_{k_m}]$.

The multivariate form of the characteristic function is a scalar-valued function of the n dimensional vector variable $\mathbf{s}$ defined by

$$\chi_{\mathbf{X}}(\mathbf{s}) = \mathrm{E}[\exp(\mathrm{j}\mathbf{s}^t\mathbf{X})] \tag{4.72}$$

If we expand the exponential as a power series as in the scalar case, we see the successive moments appearing in the expansion. The first three terms are

$$\chi_{\mathbf{X}}(\mathbf{s}) = 1 + \mathrm{j}\mathbf{s}^t\mu_{\mathbf{X}} - \frac{1}{2!}\mathbf{s}^t\mathbf{\Phi_{XX}}\mathbf{s} - ... \tag{4.73}$$

The inverse relationship which expresses the probability density of $\mathbf{X}$ in terms of $\chi_{\mathbf{X}}(\mathbf{s})$ is

$$\begin{aligned}\pi_{\mathbf{X}}(\mathbf{x}) &= \frac{1}{(2\pi)^n}\int \chi_{\mathbf{X}}(\mathbf{s})\exp(-\mathrm{j}\mathbf{s}^t\mathbf{x})\ \mathrm{d}^n\mathbf{s} \\ &= \frac{1}{(2\pi)^n}\int_{-\infty}^{\infty}\mathrm{d}s_1 ... \int_{-\infty}^{\infty}\mathrm{d}s_n\ \chi_{\mathbf{X}}(s_1,...,s_n)\exp[-\mathrm{j}(s_1x_1 + s_2x_2 + ... + s_nx_n)]\end{aligned} \tag{4.74}$$

This is essentially an n dimensional inverse Fourier transform (except for the sign of the exponent).

If the components of the vector-valued random variable $\mathbf{X}$ are *independent*, the joint probability factorizes

$$\pi_{\mathbf{X}}(\mathbf{x}) = \pi_{X_1}(x_1)\pi_{X_2}(x_2)...\pi_{X_n}(x_n) \tag{4.75}$$

As a consequence $\mathrm{E}[X_k X_l] = \mathrm{E}[X_k]\mathrm{E}[X_l]$ if $k \neq l$. The covariance matrix $\mathbf{\Gamma_{XX}}$ is then diagonal with the variances on the diagonal. The characteristic function also factorizes as

$$\chi_{\mathbf{X}}(\mathbf{s}) = \chi_{X_1}(s_1)\chi_{X_2}(s_2)...\chi_{X_n}(s_n) \tag{4.76}$$

4.6 Linear transformations and correlations

In this section we consider the generalization of the result that the characteristic function of the sum of two independent random variables is the product of the two characteristic functions. We shall see that the effect of a linear transformation is to change the *correlations* between the various components of a vector-valued random variable.

Theorem: If $\chi_{\mathbf{X}}(\mathbf{s})$ is the characteristic function of the n dimensional vector-valued random variable $\mathbf{X}$ and the m dimensional vector-valued random variable $\mathbf{Y}$ is related to $\mathbf{X}$ by the linear transformation

$$\mathbf{Y} = \mathbf{A}\mathbf{X} \tag{4.77}$$

where $\mathbf{A}$ is an m by n matrix, the characteristic function of $\mathbf{Y}$ is given by

$$\chi_{\mathbf{Y}}(\mathbf{s}) = \chi_{\mathbf{X}}(\mathbf{A}^t\mathbf{s}) \tag{4.78}$$

Proof:

$$\begin{aligned}\chi_{\mathbf{Y}}(\mathbf{s}) &= \mathrm{E}[\exp(\mathrm{j}\mathbf{s}^t\mathbf{Y})] = \mathrm{E}[\exp(\mathrm{j}\mathbf{s}^t\mathbf{A}\mathbf{X})] = \mathrm{E}[\exp(\mathrm{j}\{\mathbf{A}^t\mathbf{s}\}^t\mathbf{X})] \\ &= \chi_{\mathbf{X}}(\mathbf{A}^t\mathbf{s})\end{aligned} \tag{4.79}$$

It is worthwhile to consider in more detail the consequences of this deceptively simple result. We first note that it is a generalization of the result for the sum of two independent random variables in two ways. Firstly, there can be an arbitrary number of random variables contained in $\mathbf{X}$ and these need not be independent. Secondly, instead of a simple summation, we can now handle an arbitrary linear combination which can result in several random variables contained in $\mathbf{Y}$.

To see how this reduces to the previous result, suppose that $n = 2$, $m = 1$ and that $\mathbf{A} = (1\ 1)$. Then $\mathbf{Y} = \mathbf{A}\mathbf{X}$ becomes $Y = X_1 + X_2$. By the theorem,

$$\chi_Y(s) = \chi_{\mathbf{X}}(\mathbf{A}^t s) = \chi_{\mathbf{X}}\left(\begin{pmatrix} s \\ s \end{pmatrix}\right) \tag{4.80}$$

Since the components of $\mathbf{X}$ are independent, the characteristic function factorizes, i.e.,

$$\chi_{\mathbf{X}}\left(\begin{pmatrix} s_1 \\ s_2 \end{pmatrix}\right) = \chi_{X_1}(s_1)\chi_{X_2}(s_2) \tag{4.81}$$

Hence

$$\chi_Y(s) = \chi_{X_1}(s)\chi_{X_2}(s) \tag{4.82}$$

which is just the product of the two characteristic functions. The probability densities are thus related by a convolutional relationship.

Another important consequence of this theorem may be seen by expanding the characteristic functions as power series in $\mathbf{s}$ just as in (4.73). On the left-hand side we have

$$\chi_{\mathbf{Y}}(\mathbf{s}) = 1 + \mathrm{j}\mathbf{s}^t\mu_{\mathbf{Y}} - \frac{1}{2!}\mathbf{s}^t\mathbf{\Phi_{YY}}\mathbf{s} - ... \tag{4.83}$$

and on the right-hand side,

$$\chi_{\mathbf{X}}(\mathbf{A}^t\mathbf{s}) = 1 + \mathrm{j}(\mathbf{A}^t\mathbf{s})^t\mu_{\mathbf{X}} - \frac{1}{2!}(\mathbf{A}^t\mathbf{s})^t\mathbf{\Phi_{XX}}(\mathbf{A}^t\mathbf{s}) - ... \tag{4.84}$$

$$= 1 + \mathrm{j}\mathbf{s}^t(\mathbf{A}\mu_{\mathbf{X}}) - \frac{1}{2!}\mathbf{s}^t(\mathbf{A}\mathbf{\Phi_{XX}}\mathbf{A}^t)\mathbf{s} - ... \tag{4.85}$$

Comparing these two expansions we see that

$$\mu_{\mathbf{Y}} = \mathbf{A}\mu_{\mathbf{X}} \tag{4.86}$$

$$\mathbf{\Phi_{YY}} = \mathbf{A}\mathbf{\Phi_{XX}}\mathbf{A}^t \tag{4.87}$$

These results can also be seen more directly from the definitions. For example, if $\mathbf{Y} = \mathbf{AX}$,

$$\begin{aligned}\mathbf{\Phi_{YY}} &= \mathrm{E}\left[\mathbf{Y}\mathbf{Y}^t\right] = \mathrm{E}\left[\mathbf{AX}\left(\mathbf{AX}\right)^t\right] = \mathrm{E}\left[\mathbf{A}\mathbf{X}\mathbf{X}^t\mathbf{A}^t\right] \\ &= \mathbf{A}\mathrm{E}\left[\mathbf{X}\mathbf{X}^t\right]\mathbf{A}^t = \mathbf{A}\mathbf{\Phi_{XX}}\mathbf{A}^t.\end{aligned}$$

Exercise: Show that the covariances are also related by

$$\mathbf{\Gamma_{YY}} = \mathbf{A}\mathbf{\Gamma_{XX}}\mathbf{A}^t \tag{4.88}$$

Thus we see precisely how a linear transformation affects the moments of the random variables. The relationship for the mean is exactly as we would expect since the process of taking an expected value is linear. Higher-order moments are similarly related via further terms in the expansions.

Exercise: Show that if $X_1, X_2, ..., X_n$ are independent random variables with means $\mu_1, \mu_2, ..., \mu_n$ and variances $\sigma_1^2, \sigma_2^2, ..., \sigma_n^2$, then if $Z = c_1X_1 + c_2X_2 + ... + c_nX_n$, the mean of Z is $c_1\mu_1 + c_2\mu_2 + ... + c_n\mu_n$ and the variance of Z is $c_1^2\sigma_1^2 + c_2^2\sigma_2^2 + ... + c_n^2\sigma_n^2$.

4.6.1 Physical meaning of the covariance

In order to more fully appreciate the above result, we pause to consider what the covariance matrix is telling us about the random variables that make up the vector $\mathbf{X}$. For simplicity suppose that $n = 2$ so that the pair of random variables (X_1, X_2) may be represented by a point on a plane. On successive trials, we obtain a scatter of points whose density on the plane is given by the probability density. The mean (μ_1, μ_2) is the centroid of these points. The variance of X_i is $\Gamma_{ii} = \mathrm{E}[(X_i - \mu_i)^2]$ which is (proportional to) the moment of inertia of the points when the plane is rotated about the axis $X_i = \mu_i$. These give the diagonal terms of the covariance matrix.

The off-diagonal covariance term Γ_{12} is $\mathrm{E}[(X_1 - \mu_1)(X_2 - \mu_2)]$. For a given point, the product $(X_1 - \mu_1)(X_2 - \mu_2)$ is positive in the first and third quadrants (respectively negative in the second and fourth quadrants) where the two deviations $(X_1 - \mu_1)$ and $(X_2 - \mu_2)$ have the

same (respectively opposite) signs. The variables are *uncorrelated* and $\Gamma_{12} = 0$ if on average the deviations are as likely to have the same signs as opposite signs. $\Gamma_{12} > 0$ and we say the variables are *positively correlated* if on average the points lie in the first and third quadrants rather than in the second and fourth quadrants. This means that if one of the variables is on one side of its mean (say $X_1 > \mu_1$), on average we expect the other variable to be on the same side of its mean (i.e., $X_2 > \mu_2$). We are not certain that this will be the case, only that as the variables become more highly positively correlated, the sign of the deviation of one of the variables becomes a more reliable indication that the sign of the other deviation is the same. The opposite holds true if $\Gamma_{12} < 0$ and the variables are *negatively correlated*. In this case, the sign of one deviation makes it likely that the other deviation is of the opposite sign.

Exercise 4.2 By expanding $\mathrm{E}\big[((X_1 - \mu_1) + \alpha(X_2 - \mu_2))^2\big]$ as a quadratic in α show that $\Gamma_{12}^2 \leq \Gamma_{11}\Gamma_{22}$ where $\Gamma_{ij} = \mathrm{E}[(X_i - \mu_i)(X_j - \mu_j)]$ are the components of the covariance matrix.

(*Hint:* For all α the expectation value must be non-negative. This leads to a condition on the discriminant of the quadratic in α.)

Exercise 4.3 Show that the correlation and covariance matrices of a vector-valued random variable are positive definite matrices. (**Note:** A real-valued n by n matrix $\mathbf{A}$ is positive definite if it is symmetric and for all non-zero n dimensional column vectors $\mathbf{x}$, $\mathbf{x}^t\mathbf{A}\mathbf{x}$ is positive.)

4.7 The multivariate Gaussian and its characteristic function

First let us consider the probability density and characteristic function of n independent identically distributed Gaussian random variables with zero mean and unit variance. The probability density and characteristic function of the k'th random variable X_k are

$$\pi_k(x_k) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}x_k^2\right) \tag{4.89}$$

$$\chi_k(s_k) = \exp\left(-\frac{1}{2}s_k^2\right) \tag{4.90}$$

Since the random variables are independent, the joint probability density and characteristic function are the product of those for the individual variables

$$\pi_{\mathbf{X}}(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}} \exp\left(-\frac{1}{2}(x_1^2 + x_2^2 + ...x_n^2)\right) \tag{4.91}$$

$$\chi_{\mathbf{X}}(\mathbf{s}) = \exp\left(-\frac{1}{2}(s_1^2 + s_2^2 + ... + s_n^2)\right) = \exp\left(-\frac{1}{2}\mathbf{s}^t\mathbf{s}\right) \tag{4.92}$$

The mean vector is $\mu_{\mathbf{X}} = \mathbf{0}$ and the correlation and covariance matrix are $\mathbf{\Gamma_{XX}} = \mathbf{\Phi_{XX}} = \mathbf{I}$ the n by n identity matrix. Now consider applying the linear transformation defined by the non-singular n by n matrix $\mathbf{A}$. i.e., we consider $\mathbf{Y} = \mathbf{A}\mathbf{X}$. The mean and correlation matrix of $\mathbf{Y}$ are given as above by

$$\mu_{\mathbf{Y}} = \mathbf{0} \qquad \text{and} \qquad \mathbf{\Gamma_{YY}} = \mathbf{\Phi_{YY}} = \mathbf{A}\mathbf{A}^t \tag{4.93}$$

By the theorem the characteristic function of $\mathbf{Y}$ is

$$\chi_{\mathbf{Y}}(\mathbf{s}) = \exp\left(-\frac{1}{2}(\mathbf{A}^t\mathbf{s})^t(\mathbf{A}^t\mathbf{s})\right) = \exp\left(-\frac{1}{2}\mathbf{s}^t\mathbf{\Gamma_{YY}}\mathbf{s}\right) \tag{4.94}$$

The probability density is found from the characteristic function by calculating (4.74). As shown in the appendix, the result is

$$\pi_{\mathbf{Y}}(\mathbf{y}) = \frac{1}{\sqrt{(2\pi)^n \det(\mathbf{\Gamma_{YY}})}} \exp\left(-\frac{1}{2}\mathbf{y}^t \mathbf{\Gamma}_{\mathbf{YY}}^{-1} \mathbf{y}\right) \tag{4.95}$$

Notice how the covariance matrix appears in the expression for the characteristic function while the inverse of the covariance matrix appears in the probability density.

The exponent of a multivariate Gaussian is a *quadratic form* in the variable $\mathbf{y}$. We may write

$$\pi_{\mathbf{Y}}(\mathbf{y}) = \frac{1}{\sqrt{(2\pi)^n \det(\mathbf{\Gamma_{YY}})}} \exp\left(-\frac{1}{2}Q(\mathbf{y})\right) \tag{4.96}$$

where $Q(\mathbf{y}) = \mathbf{y}^t \mathbf{\Gamma}_{\mathbf{YY}}^{-1} \mathbf{y}$. Since the matrix $\mathbf{\Gamma}_{\mathbf{YY}}^{-1}$ is positive definite (being the inverse of a positive definite matrix), the contours $Q(\mathbf{y})$ =const form ellipsoids whose principal axes are along the eigenvectors of $\mathbf{\Gamma_{YY}}$ and whose principal axis lengths are proportional to the square roots of the eigenvalues of $\mathbf{\Gamma_{YY}}$. These contours join points of equal probability density.

If the mean of Y_k is μ_k rather than zero, the probability density and characteristic function become

$$\pi_{\mathbf{Y}}(\mathbf{y}) = \frac{1}{\sqrt{(2\pi)^n \det(\mathbf{\Gamma_{YY}})}} \exp\left(-\frac{1}{2}(\mathbf{y}-\mu_{\mathbf{Y}})^t \mathbf{\Gamma}_{\mathbf{YY}}^{-1} (\mathbf{y}-\mu_{\mathbf{Y}})\right) \tag{4.97}$$

$$\chi_{\mathbf{Y}}(\mathbf{s}) = \exp\left(\mathrm{j}\mathbf{s}^t \mu_{\mathbf{Y}} - \frac{1}{2}\mathbf{s}^t \mathbf{\Gamma_{YY}} \mathbf{s}\right) \tag{4.98}$$

These describe a general multivariate Gaussian random variable.

Exercise: If we start from an n dimensional multivariate Gaussian random variable $\mathbf{Y}$ and take a linear combination $\mathbf{Z} = \mathbf{AY}$, show that $\mathbf{Z}$ also has the form of a multivariate Gaussian. Thus an arbitrary linear combination of Gaussian variables is Gaussian.

4.A Characteristic functions

The *characteristic function* of a real continuous random variable X is defined by

$$\chi_X(s) = \mathrm{E}[\exp(\mathrm{j}sX)] = \int_{-\infty}^{\infty} \exp(\mathrm{j}sx)\, \pi_X(x)\, \mathrm{d}x \tag{4.99}$$

This is almost the same as the Fourier transform of $\pi_X(x)$ except for the sign of the exponent. The inverse transform relationship is

$$\pi_X(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \exp(-\mathrm{j}sx)\, \chi_X(s)\, \mathrm{d}s. \tag{4.100}$$

If we differentiate $\chi_X(s)$ with respect to s, the effect in the integral is to multiply the integrand by $\mathrm{j}x$. Thus,

$$\chi'_X(s) = \int_{-\infty}^{\infty} \mathrm{j}x\, \pi_X(x)\, \exp(\mathrm{j}sx)\, \mathrm{d}x \tag{4.101}$$

$$\chi'_X(0) = \int_{-\infty}^{\infty} \mathrm{j}x\, \pi_X(x)\, \mathrm{d}x = \mathrm{j}m_1 \tag{4.102}$$

Evaluating the derivative at $s = 0$ gives j times the mean (the first moment) of X. Successive differentiation leads to the rule

$$\chi_X^{(k)}(0) = \mathrm{j}^k m_k \tag{4.103}$$

Thus all the moments of the random variable can be derived from the characteristic function. We can also see this by expanding the exponential in the definition of the characteristic function as a power series.

$$\chi_X(s) = \mathrm{E}[\exp(\mathrm{j}sX)] = \mathrm{E}\left[\sum_{k=0}^{\infty} \frac{(\mathrm{j}sX)^k}{k!}\right] \tag{4.104}$$

$$= \sum_{k=0}^{\infty} \frac{\mathrm{j}^k \mathrm{E}\left[X^k\right]}{k!} s^k \tag{4.105}$$

$$= \sum_{k=0}^{\infty} \frac{\mathrm{j}^k m_k}{k!} s^k \tag{4.106}$$

This is the Taylor series expansion of $\chi_X(s)$ about $s = 0$. The coefficient of s^k is $\chi_X^{(k)}(0)/k!$ which again leads to the relationship (4.103).

An important pathological case

It is *not* the case that a complete set of moments defines the characteristic function uniquely. This is because two characteristic functions can differ by a function whose derivatives of all orders vanish at zero. Indeed it is possible to find two different probability densities which have exactly the same (finite) moments of all orders.

4.A.1 Inversion of a Gaussian characteristic function

We need to calculate the integral

$$\pi_{\mathbf{Y}}(\mathbf{y}) = \frac{1}{(2\pi)^n} \int \exp\left(-\frac{1}{2}\mathbf{s}^t\mathbf{\Gamma}\mathbf{s} - \mathrm{j}\mathbf{s}^t\mathbf{y}\right)\, \mathrm{d}^n\mathbf{s} \tag{4.107}$$

The first step is to complete the square in the exponent. Consider the matrix analogue of a perfect square

$$\frac{1}{2}(\mathbf{s}-\mathbf{s}_0)^t\mathbf{\Gamma}(\mathbf{s}-\mathbf{s}_0) = \frac{1}{2}\mathbf{s}^t\mathbf{\Gamma}\mathbf{s} - \mathbf{s}^t\mathbf{\Gamma}\mathbf{s}_0 + \frac{1}{2}\mathbf{s}_0^t\mathbf{\Gamma}\mathbf{s}_0 \tag{4.108}$$

where we have used the fact that $\mathbf{s}^t\mathbf{\Gamma}\mathbf{s}_0 = \mathbf{s_0}^t\mathbf{\Gamma}\mathbf{s}$ since they are both scalars. Rearranging this gives

$$-\frac{1}{2}\mathbf{s}^t\mathbf{\Gamma}\mathbf{s} + \mathbf{s}^t\mathbf{\Gamma}\mathbf{s}_0 = -\frac{1}{2}(\mathbf{s}-\mathbf{s}_0)^t\mathbf{\Gamma}(\mathbf{s}-\mathbf{s}_0) + \frac{1}{2}\mathbf{s}_0^t\mathbf{\Gamma}\mathbf{s}_0 \tag{4.109}$$

This can be made equal to the exponent in the integrand if we set $-\mathrm{j}\mathbf{y} = \mathbf{\Gamma}\mathbf{s}_0$ or $\mathbf{s}_0 = -\mathrm{j}\mathbf{\Gamma}^{-1}\mathbf{y}$. The integral thus becomes

$$\pi_{\mathbf{Y}}(\mathbf{y}) = \frac{1}{(2\pi)^n}\left\{\int \exp\left(-\frac{1}{2}(\mathbf{s}-\mathbf{s}_0)^t\mathbf{\Gamma}(\mathbf{s}-\mathbf{s}_0)\right)\, \mathrm{d}^n\mathbf{s}\right\} \exp\left(-\frac{1}{2}\mathbf{y}^t\mathbf{\Gamma}^{-1}\mathbf{y}\right) \tag{4.110}$$

We finally consider the integral in the braces. Remembering that $\mathbf{\Gamma} = \mathbf{A}\mathbf{A}^t$ where $\mathbf{A}$ is non-singular, we introduce the new variables

$$\mathbf{u} = \mathbf{A}^t(\mathbf{s}-\mathbf{s}_0) \tag{4.111}$$

The integral is over all of $\mathbf{s}$ space which maps to all of $\mathbf{u}$ space. The Jacobian determinant for the transformation relating the volume elements in the two spaces is

$$\mathrm{d}^n\mathbf{u} = \det(\mathbf{A}^t)\mathrm{d}^n\mathbf{s} \tag{4.112}$$

Hence

$$\begin{aligned}\int \exp\left(-\frac{1}{2}(\mathbf{s}-\mathbf{s}_0)^t\mathbf{\Gamma}(\mathbf{s}-\mathbf{s}_0)\right)\,\mathrm{d}^n\mathbf{s} &= \int \frac{\exp\left(-\frac{1}{2}(\mathbf{u}^t\mathbf{u})\right)}{\det(\mathbf{A}^t)}\,\mathrm{d}^n\mathbf{u} \\ &= \frac{(2\pi)^{n/2}}{\det(\mathbf{A}^t)}\end{aligned} \tag{4.113}$$

Since $\det(\mathbf{A}) = \det(\mathbf{A}^t)$ and $\det(\mathbf{A})\det(\mathbf{A}^t) = \det(\mathbf{\Gamma})$, we see that $\det(\mathbf{A}^t) = \sqrt{\det(\mathbf{\Gamma})}$. Hence

$$\pi_{\mathbf{Y}}(\mathbf{y}) = \frac{1}{\sqrt{(2\pi)^n\det(\mathbf{\Gamma})}}\exp\left(-\frac{1}{2}\mathbf{y}^t\mathbf{\Gamma}^{-1}\mathbf{y}\right) \tag{4.114}$$

as claimed.

4.A.2 Characteristic function of the χ^2 probability density

Let us consider first the χ^2 density with one degree of freedom. This is the probability density of the square of a zero-mean unit-variance Gaussian distribution which is

$$\pi_Y(y) = \frac{1}{\sqrt{2\pi y}}\exp\left(-\frac{y}{2}\right)\ \text{for } y > 0 \tag{4.115}$$

We wish to calculate the characteristic function which is

$$\begin{aligned}\mathrm{E}\left[\exp(\mathrm{j}sY)\right] &= \int_0^\infty \frac{1}{\sqrt{2\pi y}}\exp\left(-\frac{y}{2}\right)\exp(\mathrm{j}sy)\,\mathrm{d}y \\ &= \sqrt{\frac{1}{2\pi}}\int_{-\infty}^{\infty}\exp\left(\frac{1}{2}(2\mathrm{j}s-1)\,u^2\right)\mathrm{d}u \\ &= \sqrt{\frac{1}{2\pi}}\sqrt{\frac{2\pi}{1-2\mathrm{j}s}} = \frac{1}{\sqrt{1-2\mathrm{j}s}}.\end{aligned} \tag{4.116}$$

where we have used the change of variable $y = u^2$ and the fact that the integrand is even in the second line. For the χ^2 distribution with N degrees of freedom, we simply take the sum of N independent variables, each distributed as χ^2 with one degree of freedom. By the rule for the sum of random variables, the characteristic function is

$$\chi_Y(s) = \frac{1}{(1-2\mathrm{j}s)^{N/2}} \tag{4.117}$$

4.B Cumulants of a random variable

When independent random variables are added together, the mean of the sum is the sum of the means and the variance of the sum is the sum of the variances. The mean and variance are the first two of a set of quantities called *cumulants* which add together when independent random variables are added together.

The cumulants κ_n of a random variable X with probability density $F_X(x)$ are defined by

$$\log \chi_X(s) = \sum_{n=1}^{\infty} \kappa_n \frac{(\mathrm{j}s)^n}{n!} \tag{4.118}$$

They are just the coefficients of the power series expansion of the natural logarithm of the characteristic function. When two independent random variables are added together, we *multiply* together their characteristic functions. This corresponds to the *addition* of the logarithms of the characteristic functions. Thus the n'th cumulant of the sum is simply the sum of the n'th cumulants of the individual probability densities.

The first few cumulants are related to the moments as follows

$$\kappa_1 = m_1 \tag{4.119}$$

$$\kappa_2 = m_2 - m_1^2 \tag{4.120}$$

$$\kappa_3 = m_3 - 3m_2m_1 + 2m_1^3 \tag{4.121}$$

$$\kappa_4 = m_4 - 3m_2^2 - 4m_3m_1 + 12m_2m_1^2 - 6m_1^4 \tag{4.122}$$

These expressions are considerably simpler for random variables with zero means ($m_1 = 0$). To gain a physical picture of the significance of the first four moments, κ_1 is the mean, κ_2 is the variance, $\kappa_3/\kappa_2^{3/2}$ is the skewness and κ_4/κ_2^2 is the excess or kurtosis which measures whether the "skirts" of the probability density are broader ($\kappa_4 > 0$) or narrower ($\kappa_4 < 0$) than for a Gaussian of the same mean and variance.

Important note: For a Gaussian probability density, only the first two cumulants are non-zero since the logarithm of the characteristic function is a quadratic in s.

Exercise: Show that if X is a random variable and $Y = aX$ for some $a > 0$, the n'th cumulant of Y is a^n times the corresponding cumulant of X.

(*Hint:* First show that the probability density of Y is $\pi_Y(y) = (1/a)\pi_X(y/a)$.)

4.C Exercises

1. Show that if $Y = aX_1 + bX_2$, then

$$\pi_Y(y) = \frac{1}{|ab|} \int \pi_{X_1X_2}\left(\frac{u}{a}, \frac{y-u}{b}\right) \mathrm{d}u. \tag{4.123}$$

   Hence find the probability density function of $3X_1 + 4X_2$ when each of X_1 and X_2 is uniformly distributed in the range $[0, 1]$.

2. Find the probability density of $Z = X/Y$ if X and Y have joint probability density

$$\pi_{XY}(x, y) = \frac{1}{2\pi} \exp\left(-\frac{x^2+y^2}{2}\right). \tag{4.124}$$

   Answer: $\pi_Y(y) = \frac{1}{\pi(1+y^2)}$.

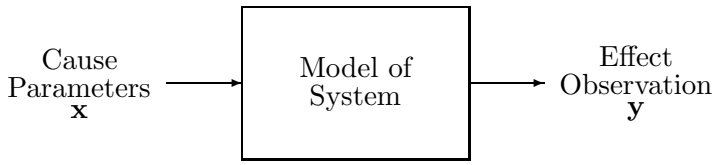
3. Find the probability density of $R = \sqrt{X^2+Y^2}$ if X and Y are distributed according to the density (4.124). Answer: $\pi_R(r) = r\exp\left(-r^2/2\right)$, or $r > 0$.

5

Bayesian statistical inference and parameter estimation

5.1 Forward and inverse probability

Bayesian statistics provides a theory of inference which enables us to relate the results of observation with theoretical predictions. Consider the process of trying to understand some physical system. Theoretical physics constructs a model which tells us what observations to expect if certain causes are present. Abstractly, a set of causes can be represented by a parameter vector $\mathbf{x}$, and the result of the observations by an observation vector $\mathbf{y}$.



Theory tells us $f(\mathbf{y}|\mathbf{x})$, the conditional probability of the observation given the cause. This is usually called the **forward probability** density. Based on observations (and possibly control of some of the parameters), experimental physics involves trying to deduce the values of the parameters, under the assumption that the model is valid. Experimentalists want $f(\mathbf{x}|\mathbf{y})$, which is the conditional probability of the possible causes, given that some effect has been observed. This **inverse probability** represents our state of knowledge of $\mathbf{x}$ after measuring $\mathbf{y}$. In the context of inverse problem theory, $\mathbf{x}$ is the image and $\mathbf{y}$ is the data.

Examples:

1. A rod of length x is measured with precision σ. If we assume Gaussian errors in the measurement model, theory predicts that an observation y has the probability density

$$f(y|x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-x}{\sigma}\right)^2\right] \tag{5.1}$$

   Given one or more measurements y_i, what is our state of knowledge of x?

2. A radioactive solid with a long half-life decays at rate x disintegrations per second so that in time T, the probability of obtaining y counts is Poisson distributed, namely

$$f(y|x) = \frac{\exp(-xT)(xT)^y}{y!} \tag{5.2}$$

In various intervals each of duration T, y_i counts were observed. What can we say about x?

3. A photograph $\mathbf{y}$ is taken of a scene $\mathbf{x}$ with an out-of-focused camera so that

$$\mathbf{y} = F(\mathbf{x}) + \mathbf{n} \tag{5.3}$$

where F denotes a "blurring operator" and $\mathbf{n}$ denotes a noise vector. The forward probability density is given by

$$f(\mathbf{y}|\mathbf{x}) = f_N(\mathbf{n} = \mathbf{y} - F(\mathbf{x})) \tag{5.4}$$

where the noise statistics and hence pdf f_N are assumed to be known. Given $\mathbf{y}$, how can we process it to recover $\mathbf{x}$, and how confident can we be of the result?

5.2 Bayes' theorem

The central result that helps us solve all of these problems is Bayes' theorem, which is based on the relationship between joint and conditional probabilities.

Given events A and B,

$$\Pr(A, B) = \Pr(A|B)\Pr(B) = \Pr(B|A)\Pr(A) \tag{5.5}$$

Hence,

$$\Pr(A|B) = \frac{1}{\Pr(B)}\Pr(B|A)\Pr(A) \tag{5.6}$$

This (and extensions to include more than 2 events) is called Bayes' theorem. It relates forward probabilities $\Pr(B|A)$ to inverse probabilities $\Pr(A|B)$.

In terms of continuous parameters and observations, the density functions satisfy

$$f(\mathbf{x}|\mathbf{y}) = \frac{1}{f(\mathbf{y})} f(\mathbf{y}|\mathbf{x}) f(\mathbf{x}) \tag{5.7}$$

We shall interpret this equation as telling us how we should change our state of knowledge of $\mathbf{x}$ as a result of making an observation which yields the result $\mathbf{y}$.

- On the right-hand side, $f(\mathbf{x})$ is the probability function which represents what we know (or believe) about the parameter $\mathbf{x}$ **before** making the observation. It is known as the **prior probability** and summarizes our initial state of knowledge about the parameter.
- On the left-hand side, $f(\mathbf{x}|\mathbf{y})$ is the probability function which tells us what we know about the parameter $\mathbf{x}$ **after** making the observation. It is called the **posterior probability**.
- The way we change the prior probability into the posterior probability is to multiply by two factors. One of these is $f(\mathbf{y}|\mathbf{x})$ which is just the forward probability function which is determined by theoretical physics. Notice that in this application we think about this as a function of $\mathbf{x}$ for a fixed observation $\mathbf{y}$. When viewed in this way, the forward probability is called the **likelihood function**.

- The remaining factor $f(\mathbf{y})^{-1}$ can be determined by normalization since we know that the sum of the posterior probability distribution function over all possible causes must be equal to one.

We now consider a specific example to illustrate the operation of this single most important result in statistical inference and data analysis.

5.2.1 The tramcar problem

A town has n tramcars labelled from 1 to n. On k separate occasions, I see trams numbered m_1, m_2,..., m_k. Based on this information, what can I say about the number of tramcars in the town?

At first sight, all that one can say as a result of seeing tramcar m is that there are at least m tramcars in the town. However one intuitively feels that if one has lived in the town for some time the highest numbered tramcar that one has ever seen will be close to the actual number of tramcars. We shall now see how Bayes' theorem can make these considerations precise.

Let us start off by analyzing the effect of the first observation, which is of tramcar m_1. The parameter we wish to estimate is n so writing down Bayes' theorem we obtain

$$\Pr(n|m_1) = \frac{1}{\Pr(m_1)} \times \Pr(m_1|n) \times \Pr(n) \tag{5.8}$$

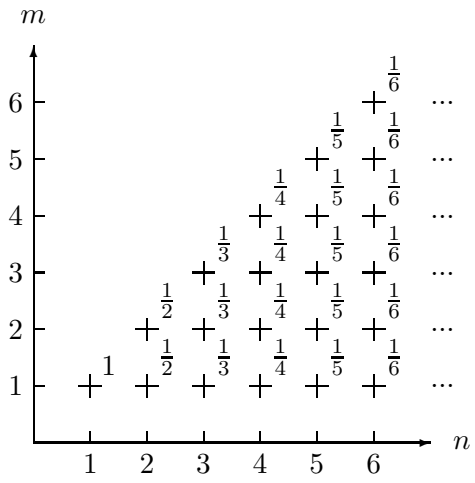
Prior modelling: Before making any observations, let us suppose that I believe that it is equally probable that there are any number between 1 and N tramcars. Later we shall consider the effect of changing N. This corresponds to the choice

$$\Pr(n) = \begin{cases} 1/N & \text{if } n \leq N \\ 0 & \text{otherwise} \end{cases} \tag{5.9}$$

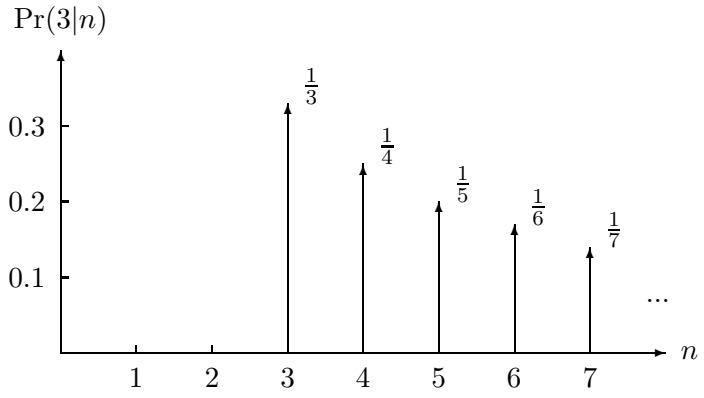
Likelihood: This is just the forward probability function. If there happen to be n tramcars in town, the probability that I see tramcar m is $1/n$ if $m \leq n$ and it is impossible to see a tramcar numbered $> n$. Thus

$$\Pr(m|n) = \begin{cases} 1/n & \text{if } m \leq n \\ 0 & \text{otherwise} \end{cases} \tag{5.10}$$

This likelihood function can be represented by the following diagram. Each cross denotes particular values of n and m and the function value associated with each cross is written next to it.



We consider the posterior probability as a function of n for a fixed observation m_1. From the expression (5.8) for the posterior probability, this involves looking across a row of the likelihood function. For example, if $m_1 = 3$, the section through the likelihood function is $\Pr(3|n)$ which appears as shown.



The posterior probability function is found by multiplying together the prior and the likelihood. The normalization term $1/\Pr(m_1)$ does not depend on n and so we can say that $\Pr(n|m_1)$ is proportional to

$$\Pr(n|m_1) \propto \begin{cases} 1/(nN) & \text{if } m_1 \leq n \leq N \\ 0 & \text{otherwise} \end{cases} \tag{5.11}$$

Since the sum over all n must be unity, we easily determine $\Pr(m_1) = \sum_{n=m_1}^{N} (nN)^{-1}$. From the posterior probability, we notice that

- It is not possible for there to be fewer than m_1 tramcars,
- The posterior probability is a sampled section of a hyperbola from m_1 to N,

- If we approximate the discrete posterior probability function by a continuous one, the mean of the posterior probability function is

$$\mu \approx \int_{m_1}^{N} n \Pr(n|m_1)\,\mathrm{d}n = \frac{N - m_1}{\log N - \log m_1} \tag{5.12}$$

  At this stage the mean tends to infinity if we let N tend to infinity. This indicates that the posterior probability function is still strongly dependent on the prior probability that we chose. With only one observation, we have not yet learnt very much and are still highly influenced by our prior suppositions.

5.3 Multiple Observations

Now let us suppose that we see tramcar number m_2. We want to compute $\Pr(n|m_1, m_2)$. Again using Bayes' theorem, assuming that the observations are independent,

$$\begin{aligned}\Pr(n|m_1, m_2) &= \frac{\Pr(n, m_1, m_2)}{\Pr(m_1, m_2)} = \frac{\Pr(m_2|n, m_1)\ \Pr(n, m_1)}{\Pr(m_2)\ \Pr(m_1)} \\ &= \frac{1}{\Pr(m_2)}\ \Pr(m_2|n)\ \Pr(n|m_1)\end{aligned} \tag{5.13}$$

So for independent observations, we can use the posterior probability for the first observation $\Pr(n|m_1)$ as the prior probability for the second observation. In effect, the likelihood functions are multiplied together. For k independent observations,

$$\Pr(n|m_1, m_2, ..., m_k) = \frac{1}{\Pr(m_1)\Pr(m_2)...\Pr(m_k)} \times \Pr(m_k|n)\ \Pr(m_{k-1}|n)...\Pr(m_1|n) \times \Pr(n) \tag{5.14}$$

This formalizes the process of how we learn from a succession of independent observations.

For the tramcar problem, after k observations we find that the posterior probability function is

$$\Pr(n|m_1, m_2, ..., m_k) = \begin{cases} N/n^k & \text{if } \max(m_1, m_2, ..., m_k) \le n \le N \\ 0 & \text{otherwise} \end{cases} \tag{5.15}$$

The normalization constant $\mathcal{N}$ is chosen so these probabilities sum to one. We see that

- $M = \max(m_1, m_2, ..., m_k)$ is a lower bound for the number of tramcars.
- With more observations (k large), the posterior probability falls off more sharply with n
- In the approximation in which the discrete probability function is approximated by a continuous probability density, the mean of the posterior probability is

$$\mu = \left(\frac{k-1}{k-2}\right) M \left[\frac{1 - (M/N)^{k-2}}{1 - (M/N)^{k-1}}\right] \qquad \text{for } k > 2 \tag{5.16}$$

  As $N \to \infty$, $\mu \to (k-1)M/(k-2)$ which is just slightly larger than M. We see that as k is increased, the cutoff N in the prior makes little difference. This means that the observations are becoming more important to the inference than our initial presuppositions.

- For large N, the variance of the posterior probability is approximately

$$\sigma^2 = \left(\frac{k-1}{k-3}\right)\left[\frac{M}{k-2}\right]^2 \tag{5.17}$$

  This becomes small as k increases. We can thus be more confident of the total number of tramcars as we see more of them.

- The likelihood function for k observations $\Pr(m_1, m_2, ..., m_k|n)$ depends on the observations $m_1, ..., m_k$ only through the two quantities k and $M = \max(m_1, m_2, ..., m_k)$. When a likelihood function is completely determined by a set of quantities, these quantities are said to form a set of **sufficient statistics** for the estimation process. In other words, a set of sufficient statistics summarizes all the information present in the set of data which is relevant for the estimation of the desired quantities.

- In the Bayesian viewpoint, our **complete state of knowledge** of the parameter(s) after the observations is given by the posterior probability density function. Often, we are asked for a "best estimate" of the parameters rather than the entire representation of our state of knowledge. The procedure for selecting such an estimate is **not** part of the framework and can sometimes be problematical. Various choices are to use the MAP (maximum *à posteriori*) estimate, the mean of the posterior probability, the maximum likelihood estimate or some other *ad hoc* estimator.

Exercises

1. In one of three boxes there are two gold coins, in another there are two silver coins and in the third there are one gold coin and one silver coin. I go to one of the boxes at random and extract a coin. Given that this coin is gold, what is the probability that the other coin in that box is also gold? Repeat the problem if there are a hundred coins in each box, all gold in the first, all silver in the second and one gold and ninety-nine silver in the third.

2. In a town, 80 percent of all taxis are blue and the other 20 percent are green. One night, an accident involving a taxi occurred, and a witness who is able to identify the colour of a taxi under the lighting conditions with 75 percent accuracy testifies that the colour of the taxi involved was green. Compute the posterior probability of the colour of the taxi after receiving the testimony.

3. A bag contains 80 fair coins and 20 double-headed coins. A coin is withdrawn at random and tossed, yielding a head. What is the probability that the coin is fair? How many times must the coin be tossed before we can be 99 percent sure that it is double-headed?

We shall now embark on a series of examples showing how these principles may be applied to a variety of problems.

5.4 Estimating a quantity with Gaussian measurement errors

Consider the problem of measuring the length x of a rod using a measuring instrument for which

$$f(y|x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y-x)^2}{2\sigma^2}\right) \tag{5.18}$$

This means that each observation comes from a Gaussian of standard deviation σ centred about the true value of x.

Experimentally, we measure the length several times and obtain an independent sequence of measurements $\{y_1, y_2, ..., y_N\}$. The likelihood function for these is

$$f(y_1, y_2, ..., y_N|x) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left(-\sum_{i=1}^{N} \frac{(y_i-x)^2}{2\sigma^2}\right) \tag{5.19}$$

The posterior probability for x starting with a prior probability of the form $f(x)$ is

$$f(x|y_1, ..., y_N) = \frac{\mathcal{N}}{(2\pi\sigma^2)^{N/2}} \exp\left(-\sum_{i=1}^{N} \frac{(y_i-x)^2}{2\sigma^2}\right) f(x) \tag{5.20}$$

where $\mathcal{N} = 1/f(y_1, ..., y_N)$ is a normalization constant. We are interested in seeing the right-hand side as a function of x. This is facilitated by expanding the squares and collecting terms in various powers of x yielding

$$f(x|y_1, ..., y_N) = \frac{\mathcal{N}}{(2\pi\sigma^2)^{N/2}} \exp\left[-\left(\frac{N}{2\sigma^2}\right)x^2 + \left(\frac{1}{\sigma^2}\right)\sum_{i=1}^{N} y_i x - \left(\frac{1}{2\sigma^2}\right)\sum_{i=1}^{N} y_i^2\right] f(x) \tag{5.21}$$

$$\propto \exp\left[-\frac{N}{2\sigma^2}\left(x - \frac{1}{N}\sum_{i=1}^{N} y_i\right)^2\right] f(x) \tag{5.22}$$

where all terms not explicitly dependent on x have been included in the proportionality. We see that the effect of collecting the data is to multiply the prior $f(x)$ by a Gaussian of mean $\sum y_i/N$ and standard deviation $\sigma/\sqrt{N}$. If the prior probability $f(x)$ before making the measurement is approximately uniform in a sufficiently large interval around $\sum y_i/N$, the posterior probability function will be almost completely determined by the data. For a Gaussian posterior probability density, the mean, median and mode all coincide, so there is little doubt as to what should be quoted as the estimate. The variance of the posterior probability represents our confidence in the result. We thus quote the mean of the data points $m = \sum y_i/N$ as the best estimate for x and give its uncertainty as $\sigma/\sqrt{N}$.

5.4.1 Estimating the measurement error as well as the quantity

In the above we assumed that the error in each datum σ is known. Often it is unknown and we seek to estimate σ as well as x from the data. This can be readily done within the Bayesian formulation by considering σ as an additional parameter. We can write

$$f(x, \sigma|y_1, y_2, ..., y_N) = \mathcal{N}\, f(y_1, y_2, ..., y_N|x, \sigma)\, f(x, \sigma) \tag{5.23}$$

where the likelihood $f(y_1, y_2, ..., y_N|x, \sigma)$ has the same form as (5.19) above. Evaluating this yields

$$f(x, \sigma|y_1, y_2, ..., y_N) = \frac{\mathcal{N}}{(2\pi\sigma^2)^{N/2}} \exp\left(-\frac{N}{2\sigma^2}\left[(x-m)^2 + s^2\right]\right) f(x, \sigma) \tag{5.24}$$

where $m = (1/N)\sum y_i$ and $s^2 = (1/N)\sum y_i^2 - m^2$. This is a joint distribution for x and σ. Again if the prior probability is approximately constant, the first factor (essentially the likelihood function) determines the posterior probability. In the Bayesian framework, this posterior probability density summarizes all that we know about the parameters after making the measurement. In the following, we shall find the following integral useful

$$\int_0^\infty \frac{1}{\sigma^k} \exp\left(-\frac{A}{\sigma^2}\right) \mathrm{d}\sigma = \frac{\Gamma\left(\frac{k-1}{2}\right)}{2\sqrt{A^{k-1}}}. \tag{5.25}$$

For a flat prior, the normalized form of the posterior probability is given by

$$f(x, \sigma|y_1, y_2, ..., y_N) = \sqrt{\frac{8}{N\pi}} \left(\frac{Ns^2}{2}\right)^{N/2} \frac{1}{s^2\Gamma\left(\frac{1}{2}N - 1\right)\sigma^N} \exp\left(-\frac{1}{2}\frac{N}{\sigma^2}\left[(x-m)^2 + s^2\right]\right). \tag{5.26}$$

The peak of this function is given by

$$x_{\mathrm{MAP}} = m \tag{5.27}$$

$$\sigma_{\mathrm{MAP}} = s \tag{5.28}$$

where "MAP" stands for **maximum à posteriori estimate**, i.e., the mode of the posterior probability density. This is also the **maximum likelihood estimate,** since the prior is assumed to be flat.

From the joint posterior probability density, we can find the marginal probability densities by integrating over the variable(s) which we do not wish to consider. The results are

$$f\left(x|y_1, y_2, ..., y_N\right) = \frac{\Gamma\left(\frac{1}{2}N - \frac{1}{2}\right)}{\Gamma\left(\frac{1}{2}N - 1\right)} \frac{s^{N-2}}{\pi^{\frac{1}{2}}\left[(x-m)^2 + s^2\right]^{\frac{N-1}{2}}}, \tag{5.29}$$

$$f\left(\sigma|y_1, y_2, ..., y_N\right) = \frac{2}{\Gamma\left(\frac{1}{2}N - 1\right)} \left(\frac{Ns^2}{2}\right)^{N/2-1} \frac{1}{\sigma^{N-1}} \exp\left(-\frac{1}{2}\frac{N}{\sigma^2}s^2\right). \tag{5.30}$$

In Figures 5.1 and 5.2 we show the joint and marginal posterior probability densities for the cases of $N = 3$ and $N = 50$ measured data points. These graphs are plotted in terms of the variables $(x - m)/s$ and σ/s in order to make them independent of m and s. The joint posterior probability density is shown as a contour diagram. The contour label λ indicates the contour at which the joint posterior probability density has fallen to a value of $\exp\left(-\lambda^2/2\right)$ of the peak value.

When giving estimates of x and σ, the peak of the posterior probability may not be a good representative of the probability distribution. This is especially the case for σ when N is small, since the posterior probability of σ is quite asymmetrical. It is possible (with some effort) to find analytic expressions for the mean of these probability densities,

$$\mathrm{E}\left[x|\mathbf{y}\right] = m, \tag{5.31}$$

$$\mathrm{E}\left[\sigma|\mathbf{y}\right] = \sqrt{\frac{N}{2}}\frac{\Gamma\left(\frac{1}{2}N - \frac{3}{2}\right)}{\Gamma\left(\frac{1}{2}N - 1\right)}s \sim \left(1 + \frac{7}{4N} + \frac{145}{32N^2} + ...\right)s. \tag{5.32}$$

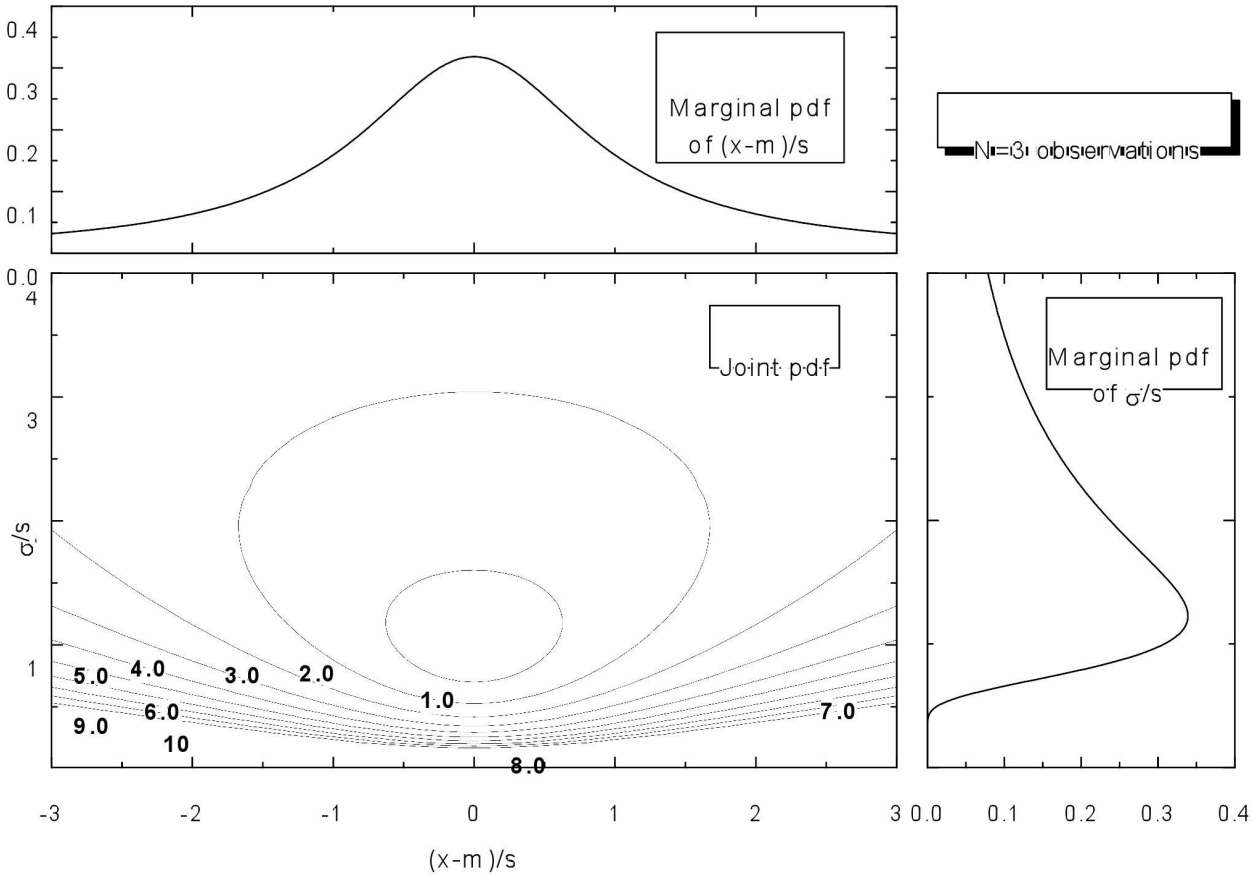


Figure 5.1: Posterior probability densities after $N = 3$ observations.

For finite N, the asymmetry in the posterior probability for σ pushes the mean higher than the mode which is at s. The covariance of the posterior probability distribution is

$$\mathrm{E}\left[(\Delta x)^2 |\mathbf{y}\right] = \mathrm{E}\left[x^2|\mathbf{y}\right] - (\mathrm{E}\left[x|\mathbf{y}\right])^2 = \frac{s^2}{N-4} \sim \left(\frac{1}{N} + \frac{4}{N^2} + ...\right) s^2, \tag{5.33}$$

$$\mathrm{E}\left[(\Delta x)(\Delta\sigma)|\mathbf{y}\right] = \mathrm{E}\left[x\sigma|\mathbf{y}\right] - \mathrm{E}\left[x|\mathbf{y}\right]\mathrm{E}\left[\sigma|\mathbf{y}\right] = 0, \tag{5.34}$$

$$\begin{aligned}\mathrm{E}\left[(\Delta\sigma)^2 |\mathbf{y}\right] &= \mathrm{E}\left[\sigma^2|\mathbf{y}\right] - (\mathrm{E}\left[\sigma|\mathbf{y}\right])^2 \\ &= \left[\frac{1}{N-4} - \frac{1}{2}\left(\frac{\Gamma\left(\frac{1}{2}N - \frac{3}{2}\right)}{\Gamma\left(\frac{1}{2}N - 1\right)}\right)^2\right] Ns^2 \qquad (5.35) \\ &\sim \left(\frac{1}{2N} + \frac{31}{8N^2} + ...\right) s^2. \qquad (5.36)\end{aligned}$$

The asymptotic approximations hold for large values of N and are based on Stirling's approximation for the Γ function, namely

$$\log\Gamma(z) \sim \frac{1}{2}\log(2\pi) - z + \left(z - \frac{1}{2}\right)\log z + \frac{1}{12z} + \mathrm{O}\left(\frac{1}{z^3}\right)$$

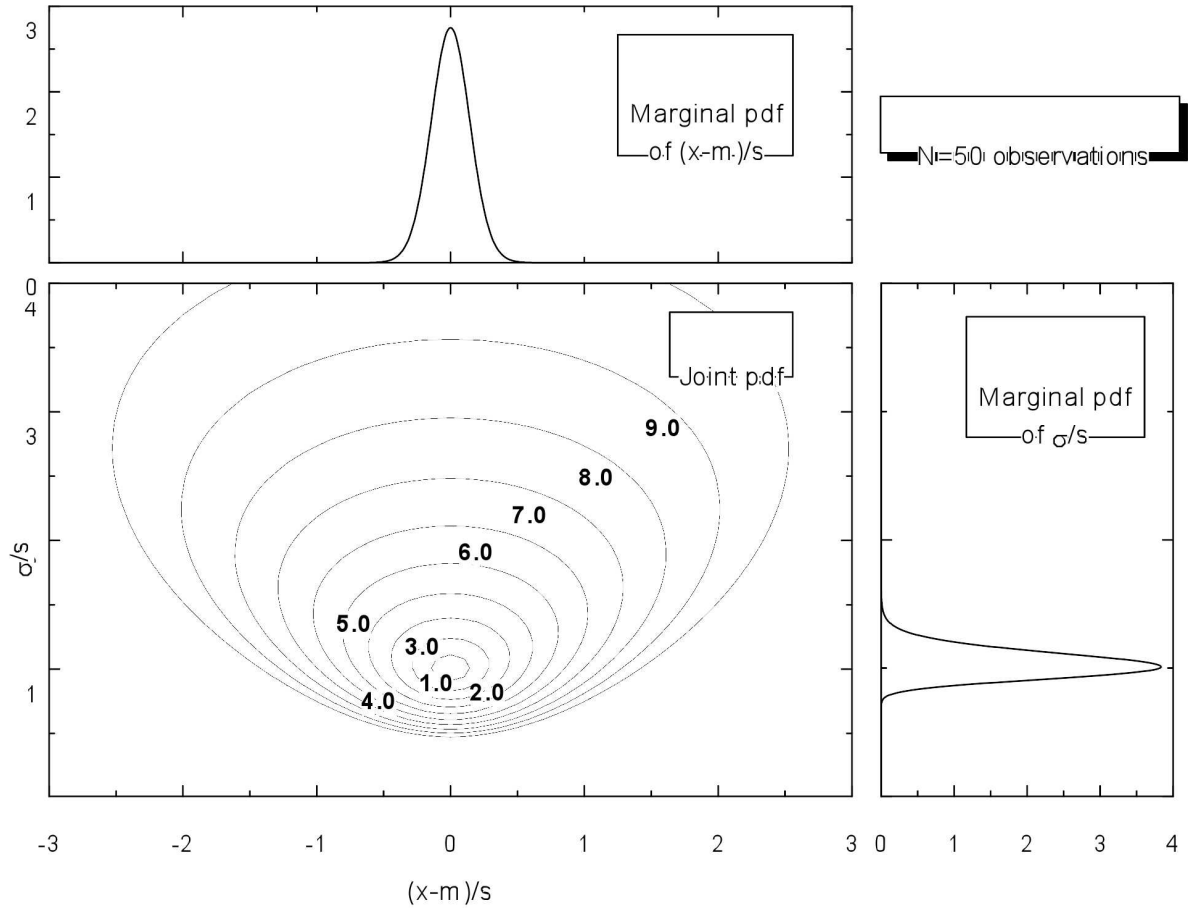


Figure 5.2: Posterior probability densities after $N = 50$ observations

from which we deduce that for large z,

$$z^{b-a}\frac{\Gamma(z+a)}{\Gamma(z+b)} \sim 1 + \frac{(a-b)(a+b-1)}{2z} \tag{5.37}$$

$$+ \frac{1}{12}\frac{(a-b)(a-b-1)\left(3(a+b-1)^2 - a + b - 1\right)}{2z^2} + \mathrm{O}\left(\frac{1}{z^3}\right). \tag{5.38}$$

We see that it is not until $N > 4$ that these probability densities have finite variances. Using the above expressions, we can give estimates of x and σ as well as the uncertainties in these quantities. Notice that the uncertainty of our estimate of x still falls as $N^{-1/2}$ for large N just as when the value of σ is known. However, for smaller values of N, this uncertainty is larger than when σ is known.

5.5 Estimating radioactive source strength and half-life

Suppose that a radioactive source has strength which decays with time according to the law

$$S(t) = S_0 \exp(-\alpha t) \tag{5.39}$$

where the half-life is $(\ln 2)/\alpha$. At time $t = 0$, we turn on an ideal Geiger counter and record the counts which occur until time T. From the record of counts, how should we estimate the values of S_0 and of α?

In order to carry out a Bayesian analysis, we want the posterior probability of the parameters S_0 and α given a particular record of counts. The forward problem requires us to find the probability of getting the particular sequence of counts given values of S_0 and α. We note that we can record the times at which the counts occur. Let us divide the interval $[0, T]$ into short subintervals of duration Δt starting at t_0, $t_1, ..., t_k, ..., t_{T/\Delta t-1}$. Each of these intervals is assumed to be so short that there is at most one count in an interval. The probability that a count occurs in the subinterval starting at t_k is $S(t_k)\,\Delta t$.

Let us suppose that there were a total of N counts in the interval $[0, T]$ and that they occured in the subintervals starting at $t_{k_1}, t_{k_2}, ..., t_{k_N}$. The probability of this particular record is the product of the probabilities that counts *did* occur in the specified subintervals and that they *did not* occur in the others. This is

$$\Pr\left(t_{k_1}, t_{k_2}, ..., t_{k_N} | S_0, \alpha\right) = (\Delta t)^N \left[\prod_{i=1}^{N} S\left(t_{k_i}\right)\right] \prod_{k \neq k_i} \left[1 - S\left(t_k\right)\Delta t\right] \tag{5.40}$$

By Bayes' theorem,

$$f\left(S_0, \alpha | t_{k_1}, ..., t_{k_N}\right) \propto f\left(S_0, \alpha\right) \left[\prod_{i=1}^{N} S_0 \mathrm{e}^{-\alpha t_{k_i}}\right] \prod_{k \neq k_i} \left[1 - S_0 \mathrm{e}^{-\alpha t_k}\Delta t\right] \tag{5.41}$$

$$\begin{aligned} \log\, f\left(S_0, \alpha | t_{k_1}, ..., t_{k_N}\right) &= \text{const} + \log\, f\left(S_0, \alpha\right) + N \log S_0 \\ &\quad - \alpha \left(\sum_{i=1}^{N} t_{k_i}\right) + \sum_{k \neq k_i} \log\left[1 - S_0 \mathrm{e}^{-\alpha t_k}\Delta t\right] \end{aligned} \tag{5.42}$$

As Δt becomes small, we can expand the last logarithm and retain only the linear term

$$\sum_{k \neq k_i} \log\left[1 - S_0 \mathrm{e}^{-\alpha t_k}\Delta t\right] \approx -S_0 \sum_{k \neq k_i} \mathrm{e}^{-\alpha t_k}\Delta t \to -S_0 \int_0^T \mathrm{e}^{-\alpha t}\,\mathrm{d}t = -\frac{S_0}{\alpha}\left(1 - \mathrm{e}^{-\alpha T}\right) \tag{5.43}$$

and so

$$\log\, f\left(S_0, \alpha | t_{k_1}, ..., t_{k_N}\right) = \text{const} + \log\, f\left(S_0, \alpha\right) + N \log S_0 - \alpha \left(\sum_{i=1}^{N} t_{k_i}\right) - \frac{S_0}{\alpha}\left(1 - \mathrm{e}^{-\alpha T}\right). \tag{5.44}$$

The *log likelihood function* consists of all terms on the right-hand side excluding that containing the prior probability density. From the form of this function, it is clear that the sufficient statistics for this problem are N, the number of counts in the interval and $\sum t_{k_i}$ which is the *sum* of the decay times. For any data set, we only need to calculate these sufficient statistics in order to completely determine the likelihood function.

If we assume that the prior is flat, we may examine how the log likelihood varies with the parameters S_0 and α. One possible strategy for estimating S_0 and α is to maximize the likelihood function for the given data, this gives the so-called *maximum likelihood* estimator.

In this example the maximum likelihood estimator of α is the solution of

$$\frac{1 - \mathrm{e}^{-\alpha T}\,(1 + \alpha T)}{\alpha\,(1 - \mathrm{e}^{-\alpha T})} = \frac{1}{N}\sum_{i=1}^{N} t_{k_i} \tag{5.45}$$

and having found α, the estimate for S_0 is

$$S_0 = \frac{N\alpha}{1 - \mathrm{e}^{-\alpha T}}. \tag{5.46}$$

We can check that this result is reasonable by considering the limit in which the source half life is very long compared to the measurement time T. In this case, the rate of counts is constant over the interval and we expect the mean of the count times on the right hand side of (5.45) to be equal to $T/2$. It is easy to check that the solution for α in this situation is $\alpha = 0$. Substituting into (5.46) gives

$$S_0 = \frac{N}{T}. \tag{5.47}$$

So the maximum likelihood estimate for the source strength when the decay is negligible is just the total number of counts divided by the counting time, as we might have expected.

In Figure (5.3), we show the contours of the log likelihood function for a source with $S_0 = 100$ and $\alpha = 0.5$. The counting time was 10 sec, during which time $N = 215$ counts were detected and the sum of the decay times was 435.2. If the likelihood is approximated by a Gaussian, the $n\sigma$ level corresponds to the probability density falling to $\exp\left(-n^2/2\right)$ of the peak. In the figure, contour lines are drawn at the maximum value of the log likelihood minus $n^2/2$. Recall that the quoted standard error is the *projection* of the 1σ uncertainty ellipse onto the coordinate axes. Notice that in this example there is a positive correlation between the values of S_0 and of α. This means that if we increase our estimate of S_0, it is also necessary to increase our estimate of the decay rate in order to maintain the same data misfit.

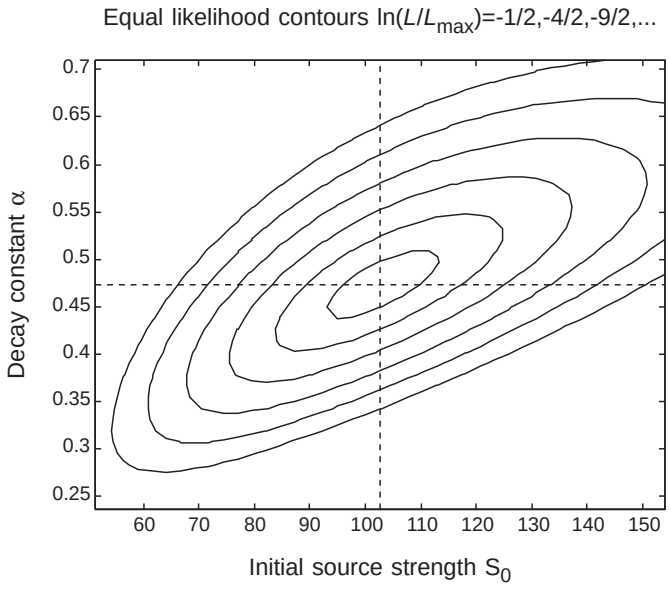


Figure 5.3: Contour plot of log likelihood for radioactive decay problem

5.6 Approximation of unimodal probability densities by Gaussians

It is often inconvenient to have to display an entire posterior probability density. If the posterior probability has a single well-defined peak and falls away from the maximum in an approximately Gaussian fashion, it is usual to approximate the probability density by a Gaussian. The key advantage of this is that a Gaussian is *completely determined* when one specifies the mean vector and the covariance matrix. Often, the user is not interested in the off-diagonal elements of the covariance matrix and only the diagonal elements (namely the variances) are required.

A Gaussian probability density is unimodal and has the property that its logarithm is a quadratic function of the variables. The maximum of this quadratic form gives the position of the mean and the *curvature* (second derivative) at the maximum gives information about the variance. In order to approximate a unimodal probability density $f(\mathbf{x})$ by a Gaussian $g(\mathbf{x})$, we adopt the following procedure:

1. Find the logarithm of the probability density $\log f(\mathbf{x})$ and find the position of its maximum $\hat{\mathbf{x}}$. This gives the mean of the approximate Gaussian.

2. Expand $\log f(\mathbf{x})$ using a Taylor series to second order about the point $\hat{\mathbf{x}}$. The linear terms vanish since $\hat{\mathbf{x}}$ is an extremum. Thus we find

$$\log f(\mathbf{x}) = \log f(\hat{\mathbf{x}}) - \frac{1}{2}(\mathbf{x}-\hat{\mathbf{x}})^t \mathbf{Q} (\mathbf{x}-\hat{\mathbf{x}}) \tag{5.48}$$

   where $\mathbf{Q}$ is the negative of the matrix of second derivatives with components of $\log f(\mathbf{x})$, i.e.,

$$Q_{ij} = -\frac{\partial^2 \log p}{\partial x_i \partial x_j} \tag{5.49}$$

3. The approximating Gaussian is

$$g(\mathbf{x}) = \mathcal{N} \exp\left[-\frac{1}{2}(\mathbf{x}-\hat{\mathbf{x}})^t \mathbf{Q} (\mathbf{x}-\hat{\mathbf{x}})\right] \tag{5.50}$$

   where $\mathcal{N}$ is a normalization factor. The covariance matrix for the approximating Gaussian is $\mathbf{Q}^{-1}$.

5.6.1 Joint estimation of quantity and measurement error problem

We now return to the posterior probability function for the problem of measuring a constant with Gaussian distributed errors

$$f(x,\sigma|y_1,y_2,...,y_N) = \frac{\mathcal{N}}{(2\pi\sigma^2)^{N/2}} \exp\left(-\frac{N}{2\sigma^2}\left[(x-m)^2+s^2\right]\right) f(x,\sigma)$$

If we suppose that the prior $f(x,\sigma)$ is flat,

$$\log f(x,\sigma|y_1,y_2,...,y_N) = \text{const} \; - N\log\sigma - \frac{N}{2\sigma^2}\left[(x-m)^2+s^2\right] \tag{5.51}$$

We find that

$$\frac{\partial \log L}{\partial x} = -\frac{N}{\sigma^2}(x-m) \tag{5.52}$$

$$\frac{\partial \log L}{\partial \sigma} = -\frac{N}{\sigma} + \frac{N}{\sigma^3}\left[(x-m)^2 + s^2\right] \tag{5.53}$$

and so the maximum occurs at $\hat{x} = m$ and $\hat{\sigma} = s$. Calculating the negative of the matrix of second derivatives and xevaluating this at $(\hat{x}, \hat{\sigma})$ yields

$$\mathbf{Q} = \begin{pmatrix} N/s^2 & 0 \\ 0 & 2N/s^2 \end{pmatrix} \tag{5.54}$$

The approximating Gaussian thus has mean (m, s) and a diagonal covariance matrix $\mathbf{Q}^{-1}$ with variance s^2/N in x and variance $s^2/(2N)$ in σ. Comparing these with the mean and covariance of the actual joint posterior probability density given above in equations (5.31) through (5.36), we see that the answers approach each other when N is large, but there are significant differences for small N.

5.6.2 Radioactive decay problem

The logarithm of the posterior probability in this case for a flat prior was

$$\log\, f\left(S_0, \alpha | t_{k_1}, ..., t_{k_N}\right) = \text{const} + N \log S_0 - \alpha \left(\sum_{i=1}^{N} t_{k_i}\right) - \frac{S_0}{\alpha}\left(1 - \mathrm{e}^{-\alpha T}\right). \tag{5.55}$$

The second derivatives are

$$\frac{\partial^2}{\partial S_0^2}\left(N \log S_0 - \alpha \left(\sum_{i=1}^{N} t_{k_i}\right) - \frac{S_0}{\alpha}\left(1 - \mathrm{e}^{-\alpha T}\right)\right) = -\frac{N}{S_0^2}$$

$$\frac{\partial^2}{\partial S_0 \partial \alpha}\left(N \log S_0 - \alpha \left(\sum_{i=1}^{N} t_{k_i}\right) - \frac{S_0}{\alpha}\left(1 - \mathrm{e}^{-\alpha T}\right)\right) = -\frac{1}{\alpha^2}\left[(1+\alpha T)\,\mathrm{e}^{-\alpha T} - 1\right]$$

$$\frac{\partial^2}{\partial \alpha^2}\left(N \log S_0 - \alpha \left(\sum_{i=1}^{N} t_{k_i}\right) - \frac{S_0}{\alpha}\left(1 - \mathrm{e}^{-\alpha T}\right)\right) = -\frac{S_0}{\alpha^3}\left[2 - \mathrm{e}^{-\alpha T}\left(T^2\alpha^2 + 2T\alpha + 2\right)\right]$$

For the data given above, the inverse covariance matrix is

$$\mathbf{Q} = \begin{pmatrix} 0.0204 & -4.24 \\ -4.24 & 1650 \end{pmatrix}, \tag{5.56}$$

and the covariance matrix is

$$\mathbf{Q}^{-1} = \begin{pmatrix} 105 & 0.271 \\ 0.271 & 0.00130 \end{pmatrix}. \tag{5.57}$$

The square roots of the diagonal elements give the standard errors in the estimates. Thus for the data set in the example

$$S_0 = 103 \pm 10 \tag{5.58}$$

$$\alpha = 0.47 \pm 0.04. \tag{5.59}$$

The graph of the Figure 5.4 shows the contours of the approximate Gaussian posterior probability density (thin lines) superimposed upon the actual posterior probability (thick lines).

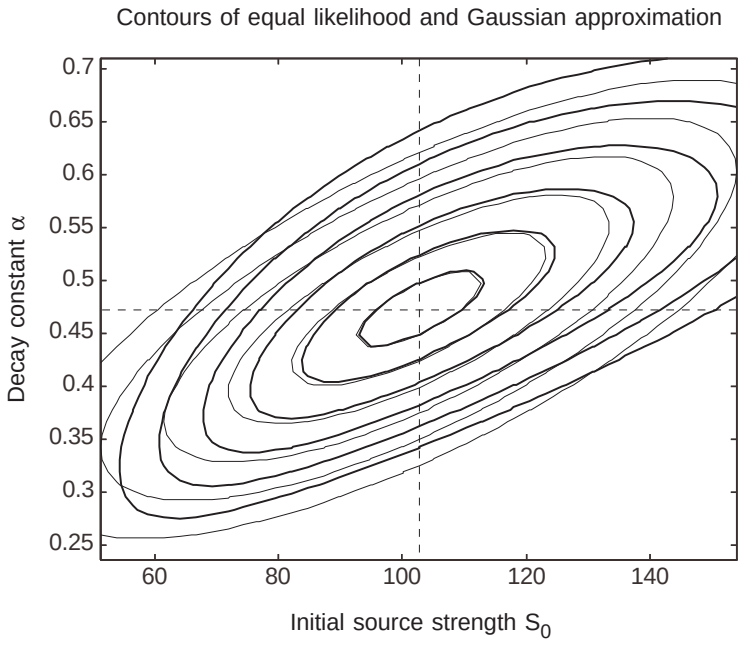


Figure 5.4: Gaussian approximation to likelihood function for radioactive decay problem

5.6.3 Interpretation of the covariance matrix

Figure 5.5 shows contours of equal probability of the bivariate Gaussian

$$f(x_1, x_2) = \frac{1}{2\pi\sqrt{\det(\mathbf{R})}} \exp\left[-\frac{1}{2}\begin{pmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{pmatrix}^t \begin{pmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{pmatrix} \begin{pmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{pmatrix}\right] \tag{5.60}$$

where

$$\mathbf{Q} = \begin{pmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{pmatrix} \tag{5.61}$$

is the inverse covariance matrix and

$$\mathbf{R} = \mathbf{Q}^{-1} = \begin{pmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{pmatrix} \tag{5.62}$$

is the covariance matrix.

Remember that this is a representation of our **state of knowlege** of the parameters x_1 and x_2. We note the following

- Corresponding to a range of parameter values such as $\mu - k\sigma$ to $\mu + k\sigma$ for a one-dimensional Gaussian, we have an **error ellipsoid** in the parameter space. These are bounded by contours of $E = (\mathbf{x}-\mu)^t\mathbf{Q}(\mathbf{x}-\mu)$ and the probability that $E \leq \chi^2$ is given by the χ^2 distribution with ν degrees of freedom where ν is the number of components in $\mathbf{x}$.
- The diagonal components of the inverse covariance matrix gives information about the intersections of the error ellipsoid with the axes. In general, these are **not** useful as estimates of the error in the parameter values.

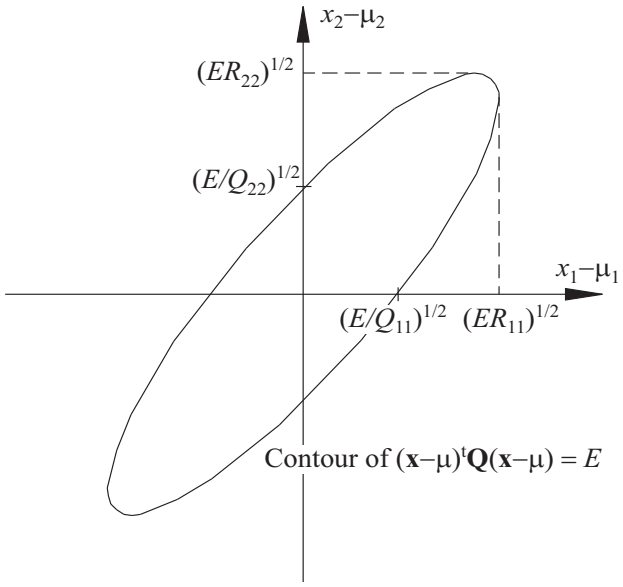


Figure 5.5: The error ellipse

- If we calculate the **marginal** probability density of one of the parameters, say x_k, the **variance** of x_k is given by the diagonal element R_{kk} of the covariance matrix. The standard error of x_k is $\sqrt{R_{kk}}$ and this is given by the projection of the error ellipsoid $E = 1$ on the k axis.

- The directions of the principal axes of the error ellipsoids are given by the eigenvectors of $\mathbf{R}$ (or of $\mathbf{Q}$). The lengths of the principal axes are related to the eigenvalues of $\mathbf{R}$.

You should be able to prove the above results.

5.7 Estimators and parameter estimation

As mentioned previously, in the Bayesian framework, our state of knowledge of a parameter or set of parameters $\mathbf{x}$ is given by the posterior probability density $f(\mathbf{x}|\mathbf{y})$. However, we are sometimes asked to give a single **estimate** of the quantity of interest. We have seen various ways of generating such estimates such as the **maximum à posteriori** estimate, the mean of the posterior probability, the maximum likelihood estimate and so on. Besides these estimators which are based on Bayesian ideas, we may consider any other method of generating an estimate. Abstractly, an **estimator** is a function $\hat{\mathbf{X}} : \mathbf{y} \to \hat{\mathbf{X}}(\mathbf{y})$ that converts the data vector into a number (or vector) which is our estimate of the quantity of interest.

In the Bayesian approach, we focus on the data that have been collected and try to discover the parameter values which best account for these data. In more conventional approaches to statistics, we decide on the **estimator** and then work out how well this estimator performs well in the long run. More specifically, we first suppose that the true value of the parameters $\mathbf{x}$ are given and calculate the **sampling distribution** $f(\mathbf{y}|\mathbf{x})$ of the possible data sets. Using our estimator, we calculate for each $\mathbf{y}$ the estimate $\hat{\mathbf{X}}(\mathbf{y})$. Then by the rule for transformation

of variables, we can find the probability density of the estimator

$$f\left(\hat{\mathbf{x}}|\mathbf{x}\right) = \int \delta\left(\hat{\mathbf{x}} - \hat{\mathbf{X}}\left(\mathbf{y}\right)\right) f\left(\mathbf{y}|\mathbf{x}\right)\, \mathrm{d}\mathbf{y}. \tag{5.63}$$

If the estimator $\hat{\mathbf{X}}$ is good, we would expect that $f\left(\hat{\mathbf{x}}|\mathbf{x}\right)$ is strongly peaked around $\hat{\mathbf{x}} = \mathbf{x}$. We would like the estimator to be good for all the parameters $\mathbf{x}$ that we are likely to encounter. It is usual to quantify the distribution of $\hat{\mathbf{x}}$ about $\mathbf{x}$ in terms of the first few moments. We define

- The **bias** of the estimator for a given true parameter vector $\mathbf{x}$ by

$$\mathcal{B}_{\hat{\mathbf{X}}}\left(\mathbf{x}\right) = \mathrm{E}\left[\hat{\mathbf{x}}|\mathbf{x}\right] - \mathbf{x} \tag{5.64}$$

$$= \int \left(\hat{\mathbf{X}}\left(\mathbf{y}\right) - \mathbf{x}\right) f\left(\mathbf{y}|\mathbf{x}\right)\, \mathrm{d}\mathbf{y}. \tag{5.65}$$

  This gives the difference between the mean of the estimator over $f\left(\mathbf{y}|\mathbf{x}\right)$and the true value of the parameters $\mathbf{x}$. An **unbiassed estimator** is one for which the bias is zero.

- The **mean-square error** of an estimator for a given true parameter vector $\mathbf{x}$ by

$$\mathrm{m.s.e.}_{\hat{\mathbf{X}}}\left(\mathbf{x}\right) = \mathrm{E}\left[\left(\hat{\mathbf{x}} - \mathbf{x}\right)\left(\hat{\mathbf{x}} - \mathbf{x}\right)^{t}\middle|\, \mathbf{x}\right] \tag{5.66}$$

$$= \int \left(\hat{\mathbf{X}}\left(\mathbf{y}\right) - \mathbf{x}\right)\left(\hat{\mathbf{X}}\left(\mathbf{y}\right) - \mathbf{x}\right)^{t} f\left(\mathbf{y}|\mathbf{x}\right)\, \mathrm{d}\mathbf{y}. \tag{5.67}$$

  For a scalar parameter x we have

$$\mathrm{m.s.e.}_{\hat{X}}\left(x\right) = \mathrm{E}\left[\left(\hat{x} - x\right)^{2}\middle|\, x\right] \tag{5.68}$$

$$= \int \left(\hat{X}\left(\mathbf{y}\right) - x\right)^{2} f\left(\mathbf{y}|x\right)\, \mathrm{d}\mathbf{y}. \tag{5.69}$$

- The **variance** of an estimator for a given true parameter vector $\mathbf{x}$ by

$$\mathrm{var}_{\hat{\mathbf{X}}}\left(\mathbf{x}\right) = \mathrm{E}\left[\left(\hat{\mathbf{x}} - \mathrm{E}\left[\hat{\mathbf{x}}|\mathbf{x}\right]\right)\left(\hat{\mathbf{x}} - \mathrm{E}\left[\hat{\mathbf{x}}|\mathbf{x}\right]\right)^{t}\middle|\, \mathbf{x}\right] \tag{5.70}$$

$$= \int \left(\hat{\mathbf{X}}\left(\mathbf{y}\right) - \mathrm{E}\left[\hat{\mathbf{x}}|\mathbf{x}\right]\right)\left(\hat{\mathbf{X}}\left(\mathbf{y}\right) - \mathrm{E}\left[\hat{\mathbf{x}}|\mathbf{x}\right]\right)^{t} f\left(\mathbf{y}|\mathbf{x}\right)\, \mathrm{d}\mathbf{y}. \tag{5.71}$$

  For a scalar parameter x we have

$$\mathrm{var}_{\hat{X}}\left(x\right) = \mathrm{E}\left[\left(\hat{x} - \mathrm{E}\left[\hat{x}|x\right]\right)^{2}\middle|\, \mathbf{x}\right] \tag{5.72}$$

$$= \int \left(\hat{X}\left(\mathbf{y}\right) - \mathrm{E}\left[\hat{x}|x\right]\right)^{2} f\left(\mathbf{y}|\mathbf{x}\right)\, \mathrm{d}\mathbf{y}. \tag{5.73}$$

It is easy to show that (check this as an exercise) for any estimator $\hat{\mathbf{X}}$ and for any $\mathbf{x}$,

$$\mathrm{m.s.e.}_{\hat{\mathbf{X}}}\left(\mathbf{x}\right) = \mathrm{var}_{\hat{\mathbf{X}}}\left(\mathbf{x}\right) + \mathcal{B}_{\hat{\mathbf{X}}}\left(\mathbf{x}\right)\mathcal{B}_{\hat{\mathbf{X}}}\left(\mathbf{x}\right)^{t}. \tag{5.74}$$

and for a scalar parameter x, this reduces to

$$\mathrm{m.s.e.}_{\hat{X}}\left(x\right) = \mathrm{var}_{\hat{X}}\left(x\right) + \mathcal{B}_{\hat{X}}\left(x\right)^{2} \tag{5.75}$$

Of course, the best estimators are those which have small bias, variance and mean-square errors.

5.7.1 Examples

1. Suppose that we have samples $\mathbf{y}$ drawn from a uniformly distributed random variable which extends from zero to x. We wish to estimate x from the data $y_1, ..., y_N$. Let us first discuss the properties of the estimator

$$\hat{X}(\mathbf{y}) = \frac{2}{N}(y_1 + y_2 + ... + y_N). \tag{5.76}$$

Since this is just a linear combination of the data, it is easy to calculate the moments of $\hat{X}$ in terms of the moments of the data. For a uniform random variable Y extending from zero to x, we know that

$$\mathrm{E}[y|x] = \frac{1}{2}x, \tag{5.77}$$

$$\mathrm{E}\left[y^2|x\right] = \frac{1}{3}x^2 \Rightarrow \mathrm{var}[y] = \frac{1}{12}x^2 \tag{5.78}$$

Hence when N of these independent random variables are added together, the mean and variances for each variable are just added together. Thus

$$\mathrm{E}\left[\hat{X}|x\right] = \frac{2}{N}\left(\frac{N}{2}x\right) = x \tag{5.79}$$

$$\mathrm{var}\left[\hat{X}|x\right] = \frac{4}{N^2}\left(\frac{N}{12}x^2\right) = \frac{x^2}{3N} \tag{5.80}$$

The estimate is **unbiassed**, and the variance and mean square error are both equal to $x^2/(3N)$.

2. Using the same data as above, let us consider instead the estimator

$$\hat{X}(\mathbf{y}) = \max(y_1, y_2, ..., y_N) \tag{5.81}$$

which happens to be the maximum likelihood estimator of x. In order to find the probability density of $\hat{X}$, we make use of the result in the previous chapter for the maximum of a set of independent identically distributed random variables. This is

$$f(\hat{x}|x) = Np_Y(\hat{x}|x)\left(\int_{-\infty}^{\hat{x}} p_Y(y|x)\,\mathrm{d}y\right)^{N-1} \tag{5.82}$$

$$= \frac{N}{x}\left(\frac{\hat{x}}{x}\right)^{N-1} \text{ for } 0 \le \hat{x} \le x \tag{5.83}$$

The mean of this distribution is

$$\mathrm{E}\left[\hat{X}\middle|\, x\right] = \int_0^x \hat{x}\frac{N}{x}\left(\frac{\hat{x}}{x}\right)^{N-1}\mathrm{d}\hat{x} = \frac{Nx}{N+1} \tag{5.84}$$

The variance of the distribution is

$$\mathrm{E}\left[\left(\hat{X} - \frac{Nx}{N+1}\right)^2\middle| x\right] = \frac{Nx^2}{(N+2)(N+1)^2} \tag{5.85}$$

We see that the estimator is biassed and that the bias is

$$\mathcal{B}_{\hat{X}}(x) = \left(\frac{N}{N+1}\right) x - x = -\frac{x}{N+1} \tag{5.86}$$

The mean square error is

$$\begin{aligned} \text{m.s.e.}_{\hat{X}}(x) &= \frac{Nx^2}{(N+2)(N+1)^2} + \left(-\frac{x}{N+1}\right)^2 \\ &= \frac{2x^2}{(N+1)(N+2)} \end{aligned} \tag{5.87}$$

Note that as N becomes large, the mean-square error of this estimator is much smaller than that for the estimator which is twice the mean of the y_k.

Exercise: Consider the estimator $\hat{X}(\mathbf{y}) = \left(\frac{N+1}{N}\right) \max(y_1, y_2, ..., y_N)$. Show that this is unbiassed and that its variance and mean-square error are $x^2/[N(N+2)]$. The variance of this estimator is larger than the maximum likelihood estimator but its mean-square error is smaller.

5.8 Optimal Estimators

5.8.1 The minimum mean-square error estimator

As defined above, the value of the mean-square error is a function of the true value of the parameters $\mathbf{x}$. If we have a prior probability density $f(\mathbf{x})$ which describes how we believe the parameters are distributed, we can consider the problem of finding the estimator which minimizes the prior probability weighted average of the mean-square errors, i.e., we choose the estimator so as to minimize

$$E = \int f(\mathbf{x}) \, \text{m.s.e.}_{\hat{\mathbf{X}}}(\mathbf{x}) \, \mathrm{d}\mathbf{x}. \tag{5.88}$$

Substituting the definition of the mean-square error, we see that

$$E = \int\int \left\| \hat{\mathbf{X}}(\mathbf{y}) - \mathbf{x} \right\|^2 f(\mathbf{y}|\mathbf{x}) \, f(\mathbf{x}) \, \mathrm{d}\mathbf{y} \, \mathrm{d}\mathbf{x} \tag{5.89}$$

$$= \int\int \left\| \hat{\mathbf{X}}(\mathbf{y}) - \mathbf{x} \right\|^2 f(\mathbf{x}, \mathbf{y}) \, \mathrm{d}\mathbf{y} \, \mathrm{d}\mathbf{x}. \tag{5.90}$$

To minimize this, we adopt a variational approach. We consider perturbing the estimator function

$$\hat{\mathbf{X}}(\mathbf{y}) \to \hat{\mathbf{X}}(\mathbf{y}) + \varepsilon \hat{\mathbf{F}}(\mathbf{y}), \tag{5.91}$$

so that

$$E(\varepsilon) = \int\int \left\| \hat{\mathbf{X}}(\mathbf{y}) + \varepsilon \hat{\mathbf{F}}(\mathbf{y}) - \mathbf{x} \right\|^2 f(\mathbf{x}, \mathbf{y}) \, \mathrm{d}\mathbf{y} \, \mathrm{d}\mathbf{x}. \tag{5.92}$$

For the optimal estimator

$$0 = \left[\frac{\partial E}{\partial \varepsilon}\right]_{\varepsilon=0} = 2 \int\int \hat{\mathbf{F}}(\mathbf{y})^t \left[\hat{\mathbf{X}}(\mathbf{y}) - \mathbf{x}\right] f(\mathbf{x}, \mathbf{y}) \, \mathrm{d}\mathbf{y} \, \mathrm{d}\mathbf{x}. \tag{5.93}$$

Since this has to hold for every choice of perturbing function $\hat{\mathbf{F}}(\mathbf{y})$, we see that

$$2\int \left[\hat{\mathbf{X}}(\mathbf{y}) - \mathbf{x}\right] f(\mathbf{x},\mathbf{y})\ \mathrm{d}\mathbf{x} = \mathbf{0}. \tag{5.94}$$

Thus

$$\hat{\mathbf{X}}(\mathbf{y})\, f(\mathbf{y}) = \int \mathbf{x} f(\mathbf{x},\mathbf{y})\ \mathrm{d}\mathbf{x}, \tag{5.95}$$

or

$$\hat{\mathbf{X}}(\mathbf{y}) = \int \mathbf{x}\frac{f(\mathbf{x},\mathbf{y})}{f(\mathbf{y})}\,\mathrm{d}\mathbf{x} = \int \mathbf{x} f(\mathbf{x}|\mathbf{y})\,\mathrm{d}\mathbf{x}. \tag{5.96}$$

The optimal estimator in this sense is just the mean over the posterior probability density.

5.8.2 The Cramér-Rao lower bound

The Cramér-Rao lower bound is a relation which gives the minimum variance that an estimator can have for a given bias. It does not give a construction for such minimum-variance estimators, but is useful for evaluating how near an estimator is to the ideal. The bound is expressed completely in terms of the forward probability for the data given the parameter $f(\mathbf{y}|x)$ and makes no reference to the ideas of prior probability. For simplicity, let us consider the case of a single **scalar** parameter x.

The result is based on the Cauchy-Schwarz inequality which may be stated in the form that if $\mathbf{y}$ is a vector-valued random variable and if $F(\mathbf{y})$ and $G(\mathbf{y})$ are scalar-valued functions of $\mathbf{y}$, then

$$\left|\mathrm{E}\left[F(\mathbf{y})\,G(\mathbf{y})\right]\right|^2 \le \mathrm{E}\left[F(\mathbf{y})^2\right]\mathrm{E}\left[G(\mathbf{y})^2\right] \tag{5.97}$$

We suppose that we have some estimator $\hat{X}(\mathbf{y})$whose variance we wish to bound. Let us consider the following choice of F and G and suppose that the expectation values are being taken over the probability density $f(\mathbf{y}|x)$ for some fixed x.

$$F(\mathbf{y}) = \hat{X}(\mathbf{y}) - \mathrm{E}\left[\hat{X}|x\right] \tag{5.98}$$

$$G(\mathbf{y}) = \frac{\partial}{\partial x}\log f(\mathbf{y}|x) \tag{5.99}$$

Then

$$F(\mathbf{y})\,G(\mathbf{y}) = \left(\hat{X}(\mathbf{y}) - \mathrm{E}\left[\hat{X}|x\right]\right)\frac{\partial}{\partial x}\log f(\mathbf{y}|x) \tag{5.100}$$

and

$$\begin{aligned}
\mathrm{E}\left[F(\mathbf{y})\,G(\mathbf{y})\right] &= \int \left\{\left(\hat{X}(\mathbf{y}) - \mathrm{E}\left[\hat{X}|x\right]\right)\frac{\partial}{\partial x}\log f(\mathbf{y}|\mathbf{x})\right\} f(\mathbf{y}|\mathbf{x})\ \mathrm{d}\mathbf{y} \\
&= \int \left(\hat{X}(\mathbf{y}) - \mathrm{E}\left[\hat{X}|x\right]\right)\frac{\partial}{\partial x} f(\mathbf{y}|\mathbf{x})\ \mathrm{d}\mathbf{y} \text{ since } \frac{\partial}{\partial x}\log f(\mathbf{y}|x) = \frac{\frac{\partial}{\partial x} f(\mathbf{y}|x)}{f(\mathbf{y}|x)} \\
&= \int \hat{X}(\mathbf{y})\frac{\partial}{\partial x} f(\mathbf{y}|x)\ \mathrm{d}\mathbf{y} \text{ since } \frac{\partial}{\partial x}\int f(\mathbf{y}|x)\ \mathrm{d}\mathbf{y} = \mathbf{0} \\
&= \frac{\partial}{\partial x}\left(\int \hat{X}(\mathbf{y})\, f(\mathbf{y}|x)\ \mathrm{d}\mathbf{y}\right) = \frac{\partial \mathrm{E}\left[\hat{X}|x\right]}{\partial x}
\end{aligned} \tag{5.101}$$

Further, we see that

$$\mathrm{E}\left[F(\mathbf{y})^2\right] = \mathrm{E}\left[\left(\hat{X}(\mathbf{y}) - \mathrm{E}\left[\hat{X}|x\right]\right)^2 \middle| \mathbf{x}\right] = \mathrm{var}_{\hat{X}}(x) \tag{5.102}$$

$$\mathrm{E}\left[G(\mathbf{y})^2\right] = \mathrm{E}\left[\left(\frac{\partial}{\partial x}\log f(\mathbf{y}|x)\right)^2 \middle| x\right] \tag{5.103}$$

and so substituting into the Cauchy-Schwarz inequality we have

$$\left(\frac{\partial \mathrm{E}\left[\hat{X}|x\right]}{\partial x}\right)^2 \leq \mathrm{var}_{\hat{X}}(x)\,\mathrm{E}\left[\left(\frac{\partial}{\partial x}\log f(\mathbf{y}|x)\right)^2 \middle| x\right] \tag{5.104}$$

By the definition of the bias,

$$\mathcal{B}_{\hat{X}}(x) = \mathrm{E}\left[\hat{X}|x\right] - x \tag{5.105}$$

$$\mathrm{var}_{\hat{X}}(x) \geq \frac{\left(1 + \mathcal{B}'_{\hat{X}}(x)\right)^2}{\mathrm{E}\left[\left(\frac{\partial}{\partial x}\log f(\mathbf{y}|x)\right)^2 |x\right]} \tag{5.106}$$

This is the statement of the **Cramér-Rao** lower bound (CRLB). Notice that it gives a minimum possible value for the variance of any estimator for a given value of x. For the special case of an **unbiassed** estimator, we have that

$$\mathrm{var}_{\hat{X}}(x) \geq \frac{1}{\mathrm{E}\left[\left(\frac{\partial}{\partial x}\log f(\mathbf{y}|x)\right)^2 |x\right]} \tag{5.107}$$

An alternative form of the denominator is often convenient. Consider

$$\frac{\partial^2}{\partial x^2}\log f(\mathbf{y}|x) = \frac{\partial}{\partial x}\left(\frac{1}{p}\frac{\partial p}{\partial x}\right) = \frac{1}{p}\frac{\partial^2 p}{\partial x^2} - \frac{1}{p^2}\left(\frac{\partial p}{\partial x}\right)^2 = \frac{1}{p}\frac{\partial^2 p}{\partial x^2} - \left(\frac{\partial}{\partial x}\log p\right)^2 \tag{5.108}$$

So taking the expectation over $f(\mathbf{y}|x)$ we get

$$\mathrm{E}\left[\frac{\partial^2}{\partial x^2}\log f(\mathbf{y}|x)\middle| x\right] = -\mathrm{E}\left[\left(\frac{\partial}{\partial x}\log f(\mathbf{y}|x)\right)^2 \middle| x\right] \tag{5.109}$$

since

$$\mathrm{E}\left[\frac{1}{p}\frac{\partial^2 f(\mathbf{y}|x)}{\partial x^2}\middle| x\right] = \int \frac{\partial^2 f(\mathbf{y}|x)}{\partial x^2}\mathrm{d}\mathbf{y} = \frac{\partial^2}{\partial x^2}\int f(\mathbf{y}|x)\,\mathrm{d}\mathbf{y} = 0. \tag{5.110}$$

Thus, the CRLB may also be written as

$$\mathrm{var}_{\hat{X}}(x) \geq \frac{\left(1 + \mathcal{B}'_{\hat{X}}(x)\right)^2}{\mathrm{E}\left[-\frac{\partial^2}{\partial x^2}\log f(\mathbf{y}|x)\middle| x\right]} \tag{5.111}$$

This has an appealing interpretation since it states that the bound is related to the expectation value of the **curvature** of the **log likelihood** function. The term in the denominator is large when the likelihood function is sharply peaked. For such likelihood functions, it is possible for estimators to achieve a lower variance than for situations in which the likelihood function is broad.

Note that in order for us to be able to use the Cramér-Rao lower bound, the function $\log f(\mathbf{y}|x)$ must be differentiable with respect to x and not have any singularities in the derivative.

5.8.3 Examples

1. Let us consider estimating the variance v from a set of N Gaussian random variables with

$$f\left(y_1,...,y_N|v\right)=\frac{1}{\left(2\pi v\right)^{N/2}}\exp\left(-\frac{1}{2v}\sum_{k=1}^{N}\left(y_k-\mu\right)^2\right). \tag{5.112}$$

From this we see that

$$\begin{aligned}\frac{\partial^2}{\partial v^2}\log f\left(\mathbf{y}|v\right)&=\frac{\partial^2}{\partial v^2}\log\left(\frac{1}{\left(2\pi v\right)^{N/2}}\exp\left(-\frac{1}{2v}\sum_{k=1}^{N}\left(y_k-\mu\right)^2\right)\right)\\&=\frac{1}{2v^3}\left(Nv-2\sum_{k=1}^{N}\left(y_k-\mu\right)^2\right)\end{aligned} \tag{5.113}$$

Taking the expectation value over $f\left(\mathbf{y}|v\right)$ yields

$$\begin{aligned}\mathrm{E}\left[\frac{\partial^2}{\partial v^2}\log f\left(\mathbf{y}|v\right)\middle|v\right]&=\frac{N}{2v^2}-\frac{1}{v^3}\sum_{k=1}^{N}\mathrm{E}\left[\left(y_k-\mu\right)^2\right]\\&=\frac{N}{2v^2}-\frac{N}{v^2}=-\frac{N}{2v^2}\end{aligned} \tag{5.114}$$

So the CRLB for the estimator is

$$\mathrm{var}_{\hat{X}}\left(x\right)\geq\frac{2v^2}{N}\left|1+\mathcal{B}'_{\hat{X}}\left(x\right)\right|^2$$

If we use an estimator for the variance given by

$$\hat{V}=\frac{1}{N}\sum_{k=1}^{N}\left[y_k-\left(\frac{1}{N}\sum_{k=1}^{N}y_k\right)\right]^2, \tag{5.115}$$

it can be shown that (check!) the estimator has bias

$$\mathcal{B}_{\hat{V}}\left(v\right)=-\frac{v}{N}, \tag{5.116}$$

and so by the CRLB

$$\mathrm{var}_{\mathrm{CRLB}}\geq\frac{2v^2}{N}\left(1-\frac{1}{N}\right)^2=\frac{2\left(N-1\right)^2v^2}{N^3}$$

The actual variance of the estimator can be shown to be (check again!)

$$\mathrm{var}_{\hat{V}}\left(v\right)=\frac{2\left(N-1\right)v^2}{N^2} \tag{5.117}$$

The ratio of the CRLB to the actual variance is called the **efficiency** of the estimate. In this case

$$\frac{\mathrm{var}_{\mathrm{CRLB}}}{\mathrm{var}_{\hat{V}}\left(v\right)}=\frac{N-1}{N} \tag{5.118}$$

As N becomes large, this efficiency approaches unity, and the estimator is said to be **asymptotically efficient.**

2. One **cannot** apply the CRLB to the estimators associated with finding the width of a uniform distribution since the log likelihood function is $-\infty$ in certain regions, and there are discontinuities at which it fails to be differentiable.

Note that it is possible to evaluate the variance of an estimator numerically by simulation and to compare the result with that given by the Cramér-Rao lower bound.

5.9 Data Modelling

We now wish to address some of the practical issues involved in *data modelling*, which may be regarded as a way of summarizing data $\mathbf{y}$ by fitting it to a "model" which depends on a set of adjustable parameters $\mathbf{x}$. This model may result from some underlying theory that the data are supposed to arise from, or it may simply be a member of a convenient class of functions (such as a polynomial, or sum of sinusoids of unknown amplitude, frequency and phase). We have seen that the Bayesian approach is to calculate the posterior probability density for $\mathbf{x}$ by using the familiar rule

$$f(\mathbf{x}|\mathbf{y}) \propto f(\mathbf{y}|\mathbf{x})\, f(\mathbf{x}) \tag{5.119}$$

we then estimate $\mathbf{x}$ by choosing some measure of the "centre" of the posterior probability function. Although this is straightforward in principle, it is often difficult to display the posterior probability density function or to calculate its statistics, because the number of parameters $\mathbf{x}$ is often rather large, and the topology of the posterior probability function may be complicated and have many local maxima. A variety of approximate methods are often employed, some of which we shall consider.

By taking the logarithm of the above equation, we may write

$$\log f(\mathbf{x}|\mathbf{y}) = \text{const} \; - \frac{1}{2}\mathcal{E}(\mathbf{x};\mathbf{y}) + \frac{1}{2}\mathcal{S}(\mathbf{x}) \tag{5.120}$$

where $\mathcal{S}(\mathbf{x}) = 2\log f(\mathbf{x})$ and $\mathcal{E}(\mathbf{x};\mathbf{y}) = -2\log f(\mathbf{y}|\mathbf{x})$. The quantity

$$\mathcal{E}(\mathbf{x};\mathbf{y}) - \mathcal{S}(\mathbf{x}) \tag{5.121}$$

may be regarded as a **figure-of-merit function** (or merit function, for short) which is small when the posterior probability is large. This merit function has two terms, the first $\mathcal{E}(\mathbf{x};\mathbf{y})$ depends both on the data and the parameters, and may be interpreted naturally as a measure of the **misfit** between the actual data and the predictions of the model. The second term $\mathcal{S}(\mathbf{x})$ which depends on the prior may be interpreted as a **preference** function, which is large when the parameters conform to our preconceptions. It should be clear that finding $\mathbf{x}$ which minimizes $\mathcal{E}(\mathbf{x};\mathbf{y})$ alone corresponds to the maximum-likelihood estimate while minimizing $\mathcal{E}(\mathbf{x};\mathbf{y}) - \mathcal{S}(\mathbf{x})$ corresponds to the maximum *à posteriori* estimate.

Besides estimating the **values** of the parameters, there are two additional important issues. One is to assess whether or not the model is appropriate for explaining the data — this involves testing the **goodness of fit** against some statistical standard, and the other is to obtain an indication of the **uncertainties** in the estimated parameter values.

5.10 Least-Squares for Parameter Estimation

Let us suppose that the noise process is **additive** and **Gaussian** distributed so that the actual data may be written as

$$\mathbf{y} = \hat{\mathbf{y}}(\mathbf{x}) + \mathbf{n}$$

where $\hat{\mathbf{y}}(\mathbf{x})$ is the **mock data** which would have been generated in the absence of noise if the true parameter vector was $\mathbf{x}$ and $\mathbf{n}$ represents the noise. The likelihood function is

$$f(\mathbf{y}|\mathbf{x}) = f(\mathbf{n} = \mathbf{y} - \hat{\mathbf{y}}(\mathbf{x})) = \frac{1}{(2\pi)^{N/2}\sqrt{\det \mathbf{\Gamma}}} \exp\left[-\frac{1}{2}(\mathbf{y} - \hat{\mathbf{y}}(\mathbf{x}))^t \mathbf{\Gamma}^{-1}(\mathbf{y} - \hat{\mathbf{y}}(\mathbf{x}))\right], \tag{5.122}$$

where the noise is assumed to have zero mean and covariance matrix $\mathbf{\Gamma}$. Ignoring a constant, the misfit function is given by

$$\mathcal{E}(\mathbf{x};\mathbf{y}) = (\mathbf{y} - \hat{\mathbf{y}}(\mathbf{x}))^t \mathbf{\Gamma}^{-1}(\mathbf{y} - \hat{\mathbf{y}}(\mathbf{x})). \tag{5.123}$$

If the noise samples are independent, the matrix $\mathbf{\Gamma}$ is diagonal with the diagonal elements being given by the variances. If further all the variances are equal to σ^2, then the likelihood function has the particularly simple form

$$f(\mathbf{y}|\mathbf{x}) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left[-\frac{1}{2\sigma^2}(\mathbf{y} - \hat{\mathbf{y}}(\mathbf{x}))^t (\mathbf{y} - \hat{\mathbf{y}}(\mathbf{x}))\right] \tag{5.124}$$

and the misfit is

$$\mathcal{E}(\mathbf{x};\mathbf{y}) = \frac{(\mathbf{y} - \hat{\mathbf{y}}(\mathbf{x}))^t (\mathbf{y} - \hat{\mathbf{y}}(\mathbf{x}))}{\sigma^2} = \sum_{k=1}^{N} \frac{1}{\sigma^2}(y_k - \hat{y}_k(\mathbf{x}))^2 \tag{5.125}$$

which is simply a sum of squares. We shall investigate the process of minimizing this misfit, or equivalently, maximizing the likelihood function. Thus we see that

Least squares ≡ maximum likelihood with independent Gaussian noise

In order to illustrate this process, we consider some specific examples.

5.10.1 Estimating amplitude of a known signal in additive Gaussian noise

Let us suppose that the data consist of N samples from the signal

$$y(t) = As(t) \tag{5.126}$$

taken at times $t_1, t_2, ..., t_N$ which need not necessarily be evenly spaced. We shall assume that $s(t)$ is known but that the amplitude A is to be determined. The components of the data vector $\mathbf{y} \in \mathbb{R}^N$ are given by

$$y_k = As(t_k) + n_k, \tag{5.127}$$

where n_k are samples of the noise. The mock data is

$$\hat{y}_k(A) = As(t_k) \equiv As_k \tag{5.128}$$

If we suppose that the noise samples are independent and each have variance σ^2, the misfit function for a given data vector $\mathbf{y}$ is

$$\mathcal{E}(\mathbf{y}|A) = \frac{(\mathbf{y} - \hat{\mathbf{y}}(A))^t (\mathbf{y} - \hat{\mathbf{y}}(A))}{\sigma^2} = \sum_{k=1}^{N} \frac{1}{\sigma^2} (y_k - As_k)^2 \tag{5.129}$$

$$= \frac{1}{\sigma^2} \left[\left(\sum_{k=1}^{N} s_k^2 \right) \left(A - \frac{\sum_{k=1}^{N} y_k s_k}{\sum_{k=1}^{N} s_k^2} \right)^2 + \left(\sum_{k=1}^{N} y_k^2 - \frac{\left(\sum_{k=1}^{N} y_k s_k \right)^2}{\sum_{k=1}^{N} s_k^2} \right) \right] \tag{5.130}$$

where we have completed the square in order to show more clearly the dependence on A. The maximum-likelihood estimate is given by maximizing the exponent. This leads to the estimate

$$\hat{A} = \frac{\sum_{k=1}^{N} y_k s_k}{\sum_{k=1}^{N} s_k^2} \tag{5.131}$$

We see that in order to obtain the estimate of A, the only function of the data that needs to be calculated is

$$\sum_{k=1}^{N} y_k s_k \tag{5.132}$$

This may be interpreted as multiplying the measured data by a copy of the known signal and summing the result (or integrating, in the continuous case). This is the basis of correlation detectors or lock-in amplifiers.

5.10.2 Estimating parameters of two sinusoids in noise

Let us suppose that the data consist of N samples from the signal

$$y(t) = A_1 \cos \omega_1 t + B_1 \sin \omega_1 t + A_2 \cos \omega_2 t + B_2 \sin \omega_2 t, \tag{5.133}$$

taken at times $t_1, t_2, ..., t_N$ which need not be evenly spaced. The quantities A_1, A_2, B_1, B_2, ω_1 and ω_2 are regarded as the unknown parameters which we wish to estimate. We shall refer to these parameters collectively by the vector $\mathbf{x} \in \mathbb{R}^M$ and the components of the data vector $\mathbf{y} \in \mathbb{R}^N$ are given by

$$y_k = A_1 \cos \omega_1 t_k + B_1 \sin \omega_1 t_k + A_2 \cos \omega_2 t_k + B_2 \sin \omega_2 t_k + n_k, \tag{5.134}$$

where n_k are samples of the noise. The mock data is

$$\hat{y}_k(A_1, A_2, B_1, B_2, \omega_1, \omega_2) = A_1 \cos \omega_1 t_k + B_1 \sin \omega_1 t_k + A_2 \cos \omega_2 t_k + B_2 \sin \omega_2 t_k. \tag{5.135}$$

If we suppose that the noise samples are independent and each have variance σ^2, the misfit function for a given data vector $\mathbf{y}$ is

$$\mathcal{E}(\mathbf{y}|\mathbf{x}) = \frac{(\mathbf{y} - \hat{\mathbf{y}}(\mathbf{x}))^t (\mathbf{y} - \hat{\mathbf{y}}(\mathbf{x}))}{\sigma^2} = \sum_{k=1}^{N} \frac{1}{\sigma^2} (y_k - \hat{y}_k(A_1, A_2, B_1, B_2, \omega_1, \omega_2))^2. \tag{5.136}$$

Minimizing this as a function of the $M = 6$ parameters may be regarded as an exercise in multidimensional non-linear optimization. A variety of iterative methods are available, which generate a sequence of iterates $\mathbf{x}_1, \mathbf{x}_2, ...$ which converge to $\arg\min_{\mathbf{x}} \mathcal{E}(\mathbf{x}; \mathbf{y})$. Some of these are

1. Non-linear simplex methods: These work on the principle of evaluating the merit function on a set of $M+1$ points called an M simplex. An M simplex is the simplest entity which encloses a "volume" in M dimensions (e.g., a 2–simplex is a triangle and a 3–simplex is a tetrahedron), and the idea is to try to enclose the position of the minimum of the merit function within the volume of the simplex. By applying a series of transformations based on the function values at the vertices, the simplex moves downhill and (hopefully) shrinks until it is small enough to specify the minimum position to the desired accuracy. The main advantages of the simplex method are its simplicity, in that it only requires function evaluations, and its robustness to non-smooth merit functions. The main disadvantage is its often prohibitively slow speed when M is moderate or large. The Matlab routine `fmins` uses the simplex algorithm.

2. Gradient-based methods: Besides using function evaluations, these methods also require the user to supply the **gradient** or first derivative of the merit function so that the algorithm knows how to proceed downhill. The naïve **steepest descents** algorithm simply computes the direction of steepest descents at the current $\mathbf{x}_n$ and proceeds as far along this direction, i.e., along the line

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \beta_n \nabla \mathcal{E}\left(\mathbf{x}_n; \mathbf{y}\right) \tag{5.137}$$

until β_n is such that a minimum is achieved, i.e.,

$$\beta_n = \arg\min_{\beta>0} \mathcal{E}\left(\mathbf{x}_n - \beta \nabla \mathcal{E}\left(\mathbf{x}_n; \mathbf{y}\right)\right). \tag{5.138}$$

It then repeats this procedure starting at $\mathbf{x}_{n+1}$ to find $\mathbf{x}_{n+2}$ and so on. Although the merit function keeps decreasing on each iterate, this algorithm is **extremely inefficient** when the contours of the merit function are long ellipses.

3. Conjugate gradient algorithm: The steepest descents method is an example of an iterative method which is based on a sequence of line searches along vectors $\mathbf{p}_n$ called "search directions", i.e.,

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \beta_n \mathbf{p}_n \tag{5.139}$$

Ideally, we would like the search directions to be such that the sequence of minimizations starting from $\mathbf{x}_0$ along the directions $\mathbf{p}_0, ..., \mathbf{p}_{K-1}$ should give the **same result** as a multidimensional minimization over the space

$$S_K = \{\mathbf{x}_0 + b_0\mathbf{p}_0 + ... + b_{K-1}\mathbf{p}_{K-1} : b_0, ..., b_{K-1} \in \mathbb{R}\} \tag{5.140}$$

Unfortunately this is not the case unless the search directions $\mathbf{p}_n$ form a **mutually conjugate set.** In the conjugate gradient algorithm, the new search direction on the iteration $n+1$ is not simply $-\nabla \mathcal{E}\left(\mathbf{x}_{n+1}\right)$. Instead, this negative gradient is combined with a multiple of the **previous** search direction so that the new search direction is conjugate to the last

$$\mathbf{p}_{n+1} = -\nabla \mathcal{E}\left(\mathbf{x}_{n+1}; \mathbf{y}\right) + \gamma_n \mathbf{p}_n \tag{5.141}$$

where γ_n is chosen to give conjugacy. It turns out that if the merit function happens to be exactly quadratic, this procedure ensures that **all** the $\mathbf{p}_n$ are mutually conjugate and so the algorithm reaches the minimum in no more than M iterations. In practise, inexact arithmetic and the non-quadratic nature of the merit function mean that this is not always achieved. The advantage of the conjugate gradient algorithm is that it can be implemented without using very much memory even for large problems. Some minor disadvantages are the need to carry out line searches and to calculate the derivatives.

4. Newton based methods: If we expand the function we wish to minimize in a Taylor series about the current iterate $\mathbf{x}_n$, we find

$$\mathcal{E}(\mathbf{x};\mathbf{y}) \approx \mathcal{E}(\mathbf{x}_n;\mathbf{y}) + \nabla\mathcal{E}(\mathbf{x}_n;\mathbf{y})^t(\mathbf{x}-\mathbf{x}_n) + \frac{1}{2}(\mathbf{x}-\mathbf{x}_n)^t\nabla\nabla\mathcal{E}(\mathbf{x}_n;\mathbf{y})(\mathbf{x}-\mathbf{x}_n) + ... \tag{5.142}$$

where $\nabla\nabla\mathcal{E}(\mathbf{x}_n;\mathbf{y})$ denotes the Hessian matrix of second derivatives taken with respect to $\mathbf{x}$. The minimum of the quadratic form found by truncating the Taylor series at the quadratic term is where

$$\mathbf{0} = \nabla\mathcal{E}(\mathbf{x};\mathbf{y}) \approx \nabla\mathcal{E}(\mathbf{x}_n;\mathbf{y}) + \nabla\nabla\mathcal{E}(\mathbf{x}_n;\mathbf{y})(\mathbf{x}-\mathbf{x}_n), \tag{5.143}$$

This has the solution

$$\mathbf{x} = \mathbf{x}_n - [\nabla\nabla\mathcal{E}(\mathbf{x}_n;\mathbf{y})]^{-1}\nabla\mathcal{E}(\mathbf{x}_n;\mathbf{y}) \tag{5.144}$$

Since this is only an approximation to the minimum, given that merit functions are in general non-quadratic, this is used as the next iterate $\mathbf{x}_{n+1}$. Newton methods converge quadratically in a neighbourhood of a minimum, but can give bad results far from a minimum where $\nabla\nabla\mathcal{E}(\mathbf{x}_n;\mathbf{y})$ may be nearly singular. It is preferable to use a method which has the features of the steepest descents algorithm (which is guaranteed to reduce the merit function on each iterate) while still far from the minimum, but which switches to a Newton based method near the minimum. One such algorithm is called the **Levenberg-Marquardt** method which sets

$$\mathbf{x}_{n+1} = \mathbf{x}_n - [\lambda\mathbf{D}(\mathbf{x}_n;\mathbf{y}) + \nabla\nabla\mathcal{E}(\mathbf{x}_n;\mathbf{y})]^{-1}\nabla\mathcal{E}(\mathbf{x}_n;\mathbf{y})$$

where the quantity λ is chosen to be small once we are near the minimum. $\mathbf{D}$ is a diagonal matrix, usually chosen to be the diagonal part of $\nabla\nabla\mathcal{E}(\mathbf{x}_n;\mathbf{y})$. This ensures that for large λ, the algorithm takes a small step in a direction which decreases $\mathcal{E}$. Note that Newton based methods require the storage and inversion of an $M \times M$ matrix on each iteration, which may be prohibitive if M is large.

Computing the gradient and Hessian matrix of $\mathcal{E}(\mathbf{x};\mathbf{y})$ is straightforward for the least-squares problem. We see that if

$$\mathcal{E}(\mathbf{x};\mathbf{y}) = \sum_{k=1}^{N}\frac{1}{\sigma^2}(y_k - \hat{y}_k(\mathbf{x}))^2, \tag{5.145}$$

then

$$\frac{\partial\mathcal{E}}{\partial x_r} = -2\sum_{k=1}^{N}\frac{[y_k - \hat{y}_k(\mathbf{x})]}{\sigma^2}\frac{\partial\hat{y}_k(\mathbf{x})}{\partial x_r} \tag{5.146}$$

and

$$\frac{\partial^2\mathcal{E}}{\partial x_r\partial x_s} = 2\sum_{k=1}^{N}\frac{1}{\sigma^2}\left[\frac{\partial\hat{y}_k(\mathbf{x})}{\partial x_r}\frac{\partial\hat{y}_k(\mathbf{x})}{\partial x_s} - [y_k - \hat{y}_k(\mathbf{x})]\frac{\partial^2\hat{y}_k(\mathbf{x})}{\partial x_r\partial x_s}\right] \tag{5.147}$$

In practise, the second term in the sum is small compared to the first, either because the model is weakly non-linear or because the $y_k - \hat{y}_k(\mathbf{x})$ is essentially noise and so they tend to be as often positive as negative, leading to cancellation when the sum is taken over k. Thus it is usual to use

$$\frac{\partial^2\mathcal{E}}{\partial x_r\partial x_s} \approx 2\sum_{k=1}^{N}\frac{1}{\sigma^2}\left[\frac{\partial\hat{y}_k(\mathbf{x})}{\partial x_r}\frac{\partial\hat{y}_k(\mathbf{x})}{\partial x_s}\right] \tag{5.148}$$

for Newton based algorithms such as the Levenberg-Marquardt method. We see that both first and second derivatives of $\mathcal{E}$ may be formed from the Jacobian matrix

$$\left(\mathbf{J}_{\mathbf{x}}\hat{\mathbf{y}}\right)_{ij} = \frac{\partial \hat{y}_i\left(\mathbf{x}\right)}{\partial x_j}$$

Linear and Non-linear parameters

In all of the methods for minimizing the merit function, the time needed for finding the minimum increases as the number of parameters is increased. It is therefore sometimes advantageous to try to reduce the size of the numerical minimization problem by doing a part of the minimization analytically. If we return to the problem of the two sinusoids in noise, we see that the parameters may be divided into two classes. In the first class we have A_1, A_2, B_1 and B_2 on which $\hat{\mathbf{y}}$ depends linearly and in the second class we have ω_1 and ω_2 on which $\hat{\mathbf{y}}$ depends non-linearly. It turns out to be possible to carry out the minimization over the linear parameters **analytically,** thus reducing the size of the search space for our optimization routines to the number of non-linear parameters alone.

We may write the dependence of $\hat{\mathbf{y}}$ on the parameters as in Eq (5.135) in the form

$$\hat{y}_k = \begin{pmatrix} \cos\omega_1 t_k & \sin\omega_1 t_k & \cos\omega_2 t_k & \sin\omega_2 t_k \end{pmatrix} \begin{pmatrix} A_1 \\ B_1 \\ A_2 \\ B_2 \end{pmatrix} \tag{5.149}$$

or

$$\begin{pmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_N \end{pmatrix} = \begin{pmatrix} \cos\omega_1 t_1 & \sin\omega_1 t_1 & \cos\omega_2 t_1 & \sin\omega_2 t_1 \\ \cos\omega_1 t_2 & \sin\omega_1 t_2 & \cos\omega_2 t_2 & \sin\omega_2 t_2 \\ \vdots & \vdots & \vdots & \vdots \\ \cos\omega_1 t_N & \sin\omega_1 t_N & \cos\omega_2 t_N & \sin\omega_2 t_N \end{pmatrix} \begin{pmatrix} A_1 \\ B_1 \\ A_2 \\ B_2 \end{pmatrix} \tag{5.150}$$

$$\hat{\mathbf{y}} = \mathbf{C}\left(\mathbf{x}_{\text{nonlin}}\right)\mathbf{x}_{\text{lin}} \tag{5.151}$$

where $\mathbf{x}_{\text{lin}}$ represents the linear parameters $\{A_1, A_2, B_1, B_2\}$ and $\mathbf{x}_{\text{nonlin}}$ represents the non-linear parameters $\{\omega_1, \omega_2\}$. The misfit function is

$$\begin{aligned} \mathcal{E}\left(\mathbf{x};\mathbf{y}\right) = \mathcal{E}\left(\mathbf{x}_{\text{lin}}, \mathbf{x}_{\text{nonlin}};\mathbf{y}\right) &= \frac{1}{\sigma^2}\left(\mathbf{y} - \hat{\mathbf{y}}\left(\mathbf{x}\right)\right)^t\left(\mathbf{y} - \hat{\mathbf{y}}\left(\mathbf{x}\right)\right) \\ &= \frac{1}{\sigma^2}\left(\mathbf{y} - \mathbf{C}\left(\mathbf{x}_{\text{nonlin}}\right)\mathbf{x}_{\text{lin}}\right)^t\left(\mathbf{y} - \mathbf{C}\left(\mathbf{x}_{\text{nonlin}}\right)\mathbf{x}_{\text{lin}}\right) \end{aligned} \tag{5.152}$$

The derivatives with respect to the linear parameters may be computed and set equal to zero. This leads to the following simultaneous equations for $\mathbf{x}_{\text{lin}}$

$$\mathbf{C}^t\mathbf{C}\mathbf{x}_{\text{lin}} = \mathbf{C}^t\mathbf{y} \tag{5.153}$$

or

$$\mathbf{x}_{\text{lin}} = \left(\mathbf{C}^t\mathbf{C}\right)^{-1}\mathbf{C}^t\mathbf{y} \tag{5.154}$$

where $\mathbf{C}$ is evaluated at the value of the non-linear parameters. Having found the linear parameters for a particular choice of the non-linear parameters, we can write the misfit function as

$$\mathcal{E}\left(\mathbf{x}_{\text{lin}}\left(\mathbf{x}_{\text{nonlin}}\right), \mathbf{x}_{\text{nonlin}};\mathbf{y}\right) = \frac{\mathbf{y}^t\mathbf{y} - \mathbf{y}^t\mathbf{C}\left(\mathbf{x}_{\text{nonlin}}\right)\left[\mathbf{C}\left(\mathbf{x}_{\text{nonlin}}\right)^t\mathbf{C}\left(\mathbf{x}_{\text{nonlin}}\right)\right]^{-1}\mathbf{C}\left(\mathbf{x}_{\text{nonlin}}\right)^t\mathbf{y}}{\sigma^2} \tag{5.155}$$

this is only a function of the non-linear parameters which often makes it more convenient for optimization. We use `fmins` or some other convenient method of finding $\mathbf{x}_{\text{nonlin}}$ and then calculate the linear parameters from this.

The Matlab code below shows how this process is applied to the problem of estimating the angular frequencies of the two sinusoids and then finding the amplitudes.

```
% List of times at which data are measured
tlist = linspace(0,6,41)';
% Synthesize some data with linear parameters xlin = [A_1;B_1;A_2;B_2]
%  and non-linear parameters w1 and w2
xlin = [1;1;2;0]; w1 = 1.5; w2 = 2.5;
C = makeC([w1;w2],tlist); yhat = C * xlin;
%  and add noise
sigma = 0.5; y = yhat + sigma*randn(size(yhat));
% Use fmins to find the optimal parameters
xnlbest = fmins('misfit',[1;2],1,[],tlist,y);
C = makeC(xnlbest,tlist); xlinbest = (C'*C)\(C'*y);
Ebest = misfit(xnlbest,tlist,y)
figure(1); plot(tlist,y,'x',tlist,C*xlinbest);
xlabel('Time'); ylabel('Samples and best fit');
% Grid of points for calculating misfit
nl = 30;
wlist = linspace(0.1,3.0,nl);
[w1,w2] = meshgrid(wlist,wlist+1e-3);
E = zeros(nl,nl);
for k = 1:nl
   w1t = w1(1,k);
   for l = 1:nl
      w2t = w2(l,1);
      E(l,k) = misfit([w1t;w2t],tlist,y);
   end
end
figure(2); contour(w1,w2,E,min(min(E))+[1:2:40]);
hold on
plot(xnlbest(1),xnlbest(2),'+',xnlbest(2),xnlbest(1),'+');
hold off
xlabel('w1'); ylabel('w2');
save fitdata tlist y xnlbest xlinbest Ebest sigma
```

This code requires the two supporting functions listed below

```
function E = misfit(xnl,tlist,y)
%
C = makec(xnl,tlist);
xlin = (C'*C)\(C'*y);
E = sum(abs(y-C*xlin).^2);
```

```
function C = makec(xnl,tlist)
%
w1 = xnl(1); w2 = xnl(2);
C = [cos(w1*tlist) sin(w1*tlist) cos(w2*tlist) sin(w2*tlist)];
```

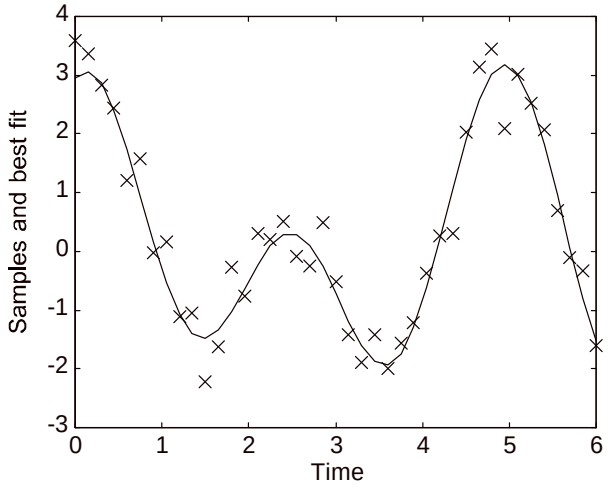


Figure 5.6: Samples of the sum of two sinusoids and best fitting model

In Figure 5.6, the noisy samples are drawn together with the best fitting model. Note that the standard deviation of the noise at each point is taken to be $\sigma = 0.5$. The true values of the parameters and those of the best fitting model are

	ω_1	ω_2	A_1	B_1	A_2	B_2
True value	1.5	2.5	1	1	2	0
Estimate	1.40	2.57	1.25	0.73	1.70	0.30

In Figure 5.7, the merit surface as a function of the non-linear parameters ω_1 and ω_2 is shown to give an idea of the topology of the function for which the minimization is carried out. We see that even in this simple example, the surface is quite complicated and there is the danger of missing the minimum if we start off with an inaccurate initial estimate. At the two (equal) minima, we find that $\mathcal{E}_{\min} = 7.4$

5.10.3 Determining the adequacy of the model and error estimates for the parameters

In the Bayesian formalism, we should investigate the posterior probability function (which is related to the merit function) to see how the probability for the parameter estimates change away from the point at which the merit function is minimized. In particular, we are concerned that the posterior probability density may be such that its maximum is not a good representative of the function, for example because there may be multiple maxima, or because most of the probability may in fact be located in a low, broad peak away from the

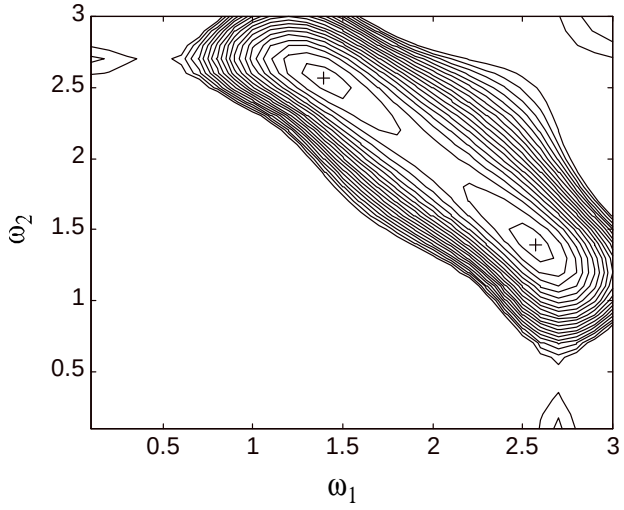
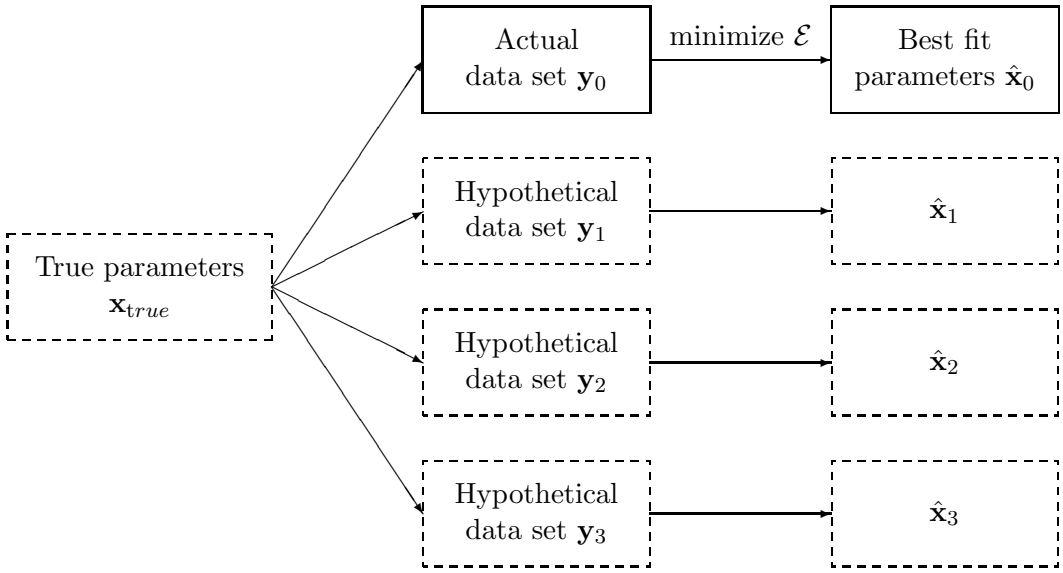


Figure 5.7: Merit function surface for problem of fitting two sinusoids.

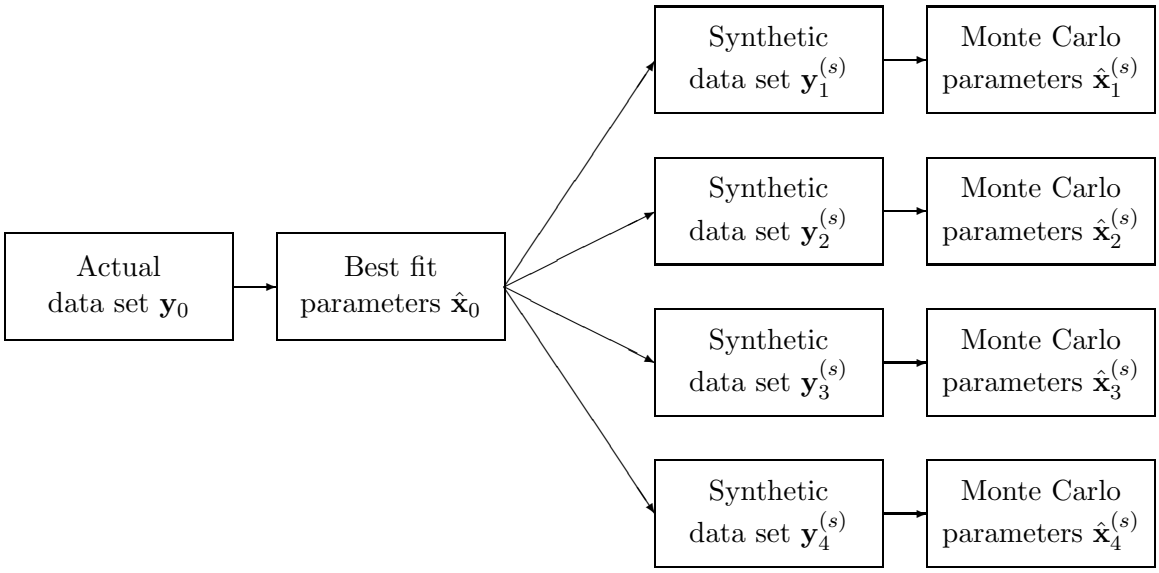
maximum. Since the posterior probability (or the likelihood) is a function of many variables ($M = 6$ in our example), it is often very difficult to visualize, unless there is a simple analytic form for the result. One solution is to try to find an algorithm which generates samples of M variables which are drawn from the posterior probability function. This is a technique which is possible for some classes of posterior probability functions, and will be discussed in more detail in the third part of this course.

Often, however, we are unable to handle the posterior probability function because of its high dimensionality but still wish to make a statement about the quality of our parameter estimates. To do this, we adopt a different point of view introduced at the end of the last chapter. Instead on focussing on the **particular data set** that we collected, we ask what is the **likely range** of possible data sets given that the parameters have **specified** values.



In the above diagram, we show the conceptual framework. There are some "true" parameter values $\mathbf{x}_{\text{true}}$ which generate the data using the model defined by the forward probability $f(\mathbf{y}|\mathbf{x}_{\text{true}})$. The actual data set $\mathbf{y}_0$ we collected is a **particular sample** from this forward probability. However, since the noise could have been different, other possible (hypothetical) data sets are $\mathbf{y}_1$, $\mathbf{y}_2, ...$, as shown. From a data set, we have an algorithm for estimating the parameters $\hat{\mathbf{x}}$ which may involve defining a merit function and minimizing this, just as described above. Using this algorithm, we compute the estimate $\hat{\mathbf{x}}_0$ from the actual data set $\mathbf{y}_0$. Conceptually, however, we can also compute estimates $\hat{\mathbf{x}}_1$, $\hat{\mathbf{x}}_2, ...$ from the hypothetical data sets $\mathbf{y}_1$, $\mathbf{y}_2, ...$ We now look at the **width** of the **distributions** of the estimates $\hat{\mathbf{x}}_0$, $\hat{\mathbf{x}}_1$, $\hat{\mathbf{x}}_2, ...$ about $\mathbf{x}_{\text{true}}$ in order to quantify the accuracy of the estimates.

Unfortunately, this strategy requires us to know the value of $\mathbf{x}_{\text{true}}$, which is of course unavailable. What we do instead is to first calculate $\hat{\mathbf{x}}_0$ from the actual data set $\mathbf{y}_0$ and pretend that this is the true value of $\mathbf{x}$. We then construct "synthetic data sets" $\mathbf{y}_1^{(s)}$, $\mathbf{y}_2^{(s)}, ...$ by drawing from the forward probability function $f(\mathbf{y}|\hat{\mathbf{x}}_0)$. This requires us to have a reasonable idea of the noise process. From each of the synthetic data sets, we carry out the estimation process to find $\hat{\mathbf{x}}_1^{(s)}$, $\hat{\mathbf{x}}_2^{(s)}, ...$ which we call **Monte Carlo parameters.** This process is shown in the diagram.



We then study the distribution of $\hat{\mathbf{x}}_1^{(s)}$, $\hat{\mathbf{x}}_2^{(s)}$, ... about $\hat{\mathbf{x}}_0$ to tell us about the accuracy of estimation process. When forming each Monte Carlo parameter estimate, we obtain a value for the merit function at the minimum. By plotting a histogram of these minima, we can see whether $\mathcal{E}(\hat{\mathbf{x}}_0; \mathbf{y}_0) = \min \mathcal{E}(\mathbf{x}; \mathbf{y}_0)$ is reasonable. This gives an indication of the **model adequacy.** Note that we are making the (possibly big) **assumption** that the distribution of $\hat{\mathbf{x}}_k^{(s)}$ about $\hat{\mathbf{x}}_0$ is not too different from the (inaccessible) distribution of $\hat{\mathbf{x}}_k$ about $\mathbf{x}_{\text{true}}$.

The Matlab program below illustrates how Monte Carlo simulation may be carried out for the fitting problem

```
clear
load fitdata
Nmc = 100;
ymock = makeC(xnlbest,tlist) * xlinbest;
fid = fopen('fitmc.dat','w');

for k = 1:Nmc
% Make synthetic data
   ysyn = ymock + sigma*randn(size(y));
   xnlMC = fmins('misfit',xnlbest,0,[],tlist,ysyn);
   xnlMC = sort(xnlMC);
   C = makeC(xnlMC,tlist);
   xlinMC = (C'*C)\(C'*ysyn);
   EMC = misfit(xnlMC,tlist,ysyn);
   fprintf(fid,'%f  ',xnlMC,xlinMC,EMC);
   fprintf(fid,'\n');
end
fclose(fid);
```

The results of the simulation are stored in the file `fitmc.dat` and are summarized in the following table. Each row corresponds to a new synthetic data set, and the estimates of the parameters and the minimum value of the merit function are tabulated for that data.

	$\omega_1^{(s)}$	$\omega_2^{(s)}$	$A_1^{(s)}$	$B_1^{(s)}$	$A_2^{(s)}$	$B_2^{(s)}$	$\mathcal{E}_{\min}$
Data realization $\mathbf{y}_1^{(s)}$	1.33	2.63	1.32	0.50	1.71	0.49	8.61
Data realization $\mathbf{y}_2^{(s)}$	1.36	2.58	1.38	0.61	1.91	0.31	7.50
Data realization $\mathbf{y}_3^{(s)}$	1.45	2.48	1.40	0.81	1.86	−0.22	14.66
	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Data realization $\mathbf{y}_{100}^{(s)}$	1.40	2.60	1.25	0.93	1.70	0.61	12.14
Mean	1.38	2.57	1.27	0.71	1.71	0.29	8.73
Standard deviation	0.08	0.07	0.16	0.34	0.16	0.35	2.19

From this table, we can place error bars on the estimates found previously. We write

$$\begin{aligned}\omega_1 &= 1.40 \pm 0.08\\ \omega_2 &= 2.57 \pm 0.07\\ A_1 &= 1.2 \pm 0.2\\ B_1 &= 0.7 \pm 0.3\\ A_2 &= 1.7 \pm 0.2\\ B_2 &= 0.3 \pm 0.4\end{aligned}$$

Since the means over the Monte Carlo runs lie within these error bars, we have no evidence that the estimates are biassed. Notice that the minimum value of $\mathcal{E}$ obtained in the original data fit is 7.4, which is well within the range of $\mathcal{E}_{\min}$ obtained during the Monte Carlo simulations. Note that the distribution of $\mathcal{E}_{\min}$ is **not** Gaussian, in general, and so we would usually not be too alarmed even if the value of $\mathcal{E}_{\min}$ that we obtained in the original fit is several standard deviations larger than the average in the Monte Carlo simulations. We would become concerned about using an inadequate model only if the probability of getting the value of $\mathcal{E}_{\min}$ found in the original fit is less than about 10^{-3}.

5.10.4 The Joint Gaussian Approximation

The above method of using Monte Carlo simulation of many data sets is very general and allows us to consider non-linear models and non-Gaussian noise processes. However, the need to produce many synthetic data sets can be quite time consuming. If the posterior probability function can be approximated by a multivariate Gaussian, it is possible to obtain error estimates on the parameters by using the technique described previously. Since we are approximating the posterior probability by

$$\mathcal{N} \exp\left(-\frac{1}{2}\left[\mathcal{E}\left(\mathbf{x};\mathbf{y}\right) - \mathcal{S}\left(\mathbf{x}\right)\right]\right) \tag{5.156}$$

or the likelihood function by

$$\mathcal{N} \exp\left(-\frac{1}{2}\mathcal{E}\left(\mathbf{x};\mathbf{y}\right)\right) \tag{5.157}$$

the **formal covariance matrix** is $\mathbf{Q}^{-1}$ where

$$Q_{ij}=\frac{\partial^2 L}{\partial x_i \partial x_j}, \tag{5.158}$$

and $L(\mathbf{x})$ is either $\mathcal{E}(\mathbf{x};\mathbf{y}) - \mathcal{S}(\mathbf{x})$ for the MAP estimator or is $\mathcal{E}(\mathbf{x};\mathbf{y})$ for the maximum likelihood estimator. The formal variances of the individual parameters are found by taking the diagonal elements of $\mathbf{Q}^{-1}$.

In order to assess the model adequacy, it can be shown that in the special case of **additive Gaussian noise** of covariance $\mathbf{\Gamma}$ and a purely linear model $\hat{\mathbf{y}}(\mathbf{x}) = \mathbf{C}\mathbf{x}$, for some $N \times M$ matrix $\mathbf{C}$, it can be shown if we define

$$\mathcal{E}(\mathbf{x};\mathbf{y}) = (\mathbf{y} - \hat{\mathbf{y}}(\mathbf{x}))^t \mathbf{\Gamma}^{-1} (\mathbf{y} - \hat{\mathbf{y}}(\mathbf{x}))$$

then for a given data set $\mathbf{y}$, the distribution of

$$\mathcal{E}_{\min} \equiv \min_{\mathbf{x}} \mathcal{E}(\mathbf{x};\mathbf{y})$$

is χ^2 with $N - M$ degrees of freedom. By using tables of the χ^2 distribution, one can judge if the value of $\mathcal{E}_{\min}$ obtained in the fit is reasonable.

These results are often used in practise even for non-linear fitting problems, although this can be dangerous in pathological cases. In particular, the quantitative confidence levels for Gaussian distributions will generally differ from those of the true posterior probability, even though the shapes of the contours of equal posterior probability (which are assumed to be ellipsoids) may be reasonable near the optimum parameter values.

6

Stochastic Simulation

6.1 Markov Chains

6.1.1 Introduction

Suppose that $\mathcal{M} = \{X_n\}_{n=0}^{\infty}$, (or perhaps $\mathcal{M} = \{X_n\}_{n=-\infty}^{\infty}$) is a sequence of correlated random variables, where each X_n comes from some set Ω, *.called* the **state space.** We assume that states in Ω can be labeled by the integers, i.e., Ω is discrete. The process is a **Markov chain** if it satisfies the **Markov condition**

$$\Pr\left(X_{n+1} = j | X_n = i, X_{n-1} = x_{n-1}, ..., X_0 = x_0\right) = \Pr\left(X_{n+1} = j | X_n = i\right). \tag{6.1}$$

Fixing an **initial distribution** $\Pr\left(X_0 = i\right)$ for X_0 and the **transition probability** for X_{n+1} given X_n, $\Pr\left(X_{n+1} = j | X_n = i\right)$ determines a Markov chain.

If the transition probability does not depend on n, i.e., if

$$\Pr\left(X_{n+m+1} = j | X_{n+m} = i\right) = \Pr\left(X_{n+1} = j | X_n = i\right) \text{ for all } m \in \mathbb{Z}, \tag{6.2}$$

we say that $\mathcal{M}$ is **homogeneous** and we write the transition probability as a matrix $\mathbf{P}$ where

$$P_{ij} = \Pr\left(X_{n+1} = j | X_n = i\right). \tag{6.3}$$

Note that P_{ij} denotes the conditional probability to enter state j on the next step, given that the current state is i. The transition probabilities satisfy the normalization condition

$$\sum_{j \in \Omega} P_{ij} = 1 \tag{6.4}$$

since the chain must be in **some** state on the next step. A matrix with rows which sum to one is called **stochastic**.

Example 6.1 Suppose that $\Omega = \{1, 2, 3\}$, the transition matrix is

$$\mathbf{P} = \begin{pmatrix} \frac{2}{5} & \frac{1}{2} & \frac{1}{10} \\ \frac{1}{5} & \frac{7}{10} & \frac{1}{10} \\ \frac{2}{5} & \frac{2}{5} & \frac{1}{5} \end{pmatrix},$$

and that the initial distribution is $\Pr\left(X_0 = i\right) = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$. We may represent the transition matrix of the Markov chain as a graph with a vertex for each state and a directed edge from vertex i to vertex j when there is a nonzero transition probability P_{ij} from i to j. Thus for the Markov chain above, we would have the digraph (directed graph) shown in Figure 6.1.

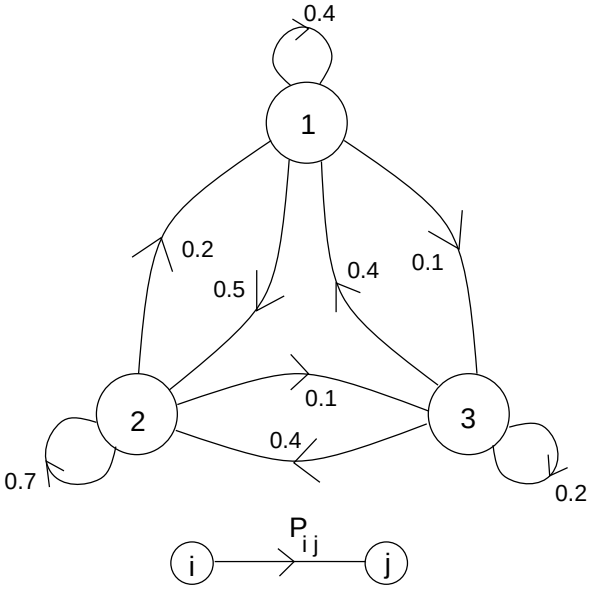


Figure 6.1: Digraph corresponding to the transition matrix of Example 6.1.

Simulation Note that if $a + b + c = 1$ and we wish to pick a state in $\{1, 2, 3\}$ with probability a for 1, b for 2 and c for 3, we need only generate a random number p distributed uniformly on $[0, 1]$ and if $p < a$, pick 1, if $a < p < a+b$, pick 2 and if $a+b < p < a+b+c = 1$ pick c.

We will use this technique to simulate the chain in Example 6.1. The random numbers

u_1	u_2	u_3	u_4	u_5	u_6
0.429	0.156	0.146	0.951	0.921	0.644

have been sampled from a uniform distribution on $[0, 1]$. Our simulation proceeds as follows:

1. Pick X_0 using the initializing distribution $\Pr(X_0 = i) = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$. Since we have $a = b = c = \frac{1}{3}$ and $u_1 = 0.429$, we select $X_0 = 2$.

2. We are now in state 2. We must choose a new state by jumping from state 2. Since $P_{2j} = \left(\frac{1}{5}, \frac{7}{10}, \frac{1}{10}\right)$, we have $a = \frac{1}{5}$, $b = \frac{7}{10}$ and $c = \frac{1}{10}$. Since $u_2 = 0.156 < a$, we select $X_1 = 1$.

3. We are now in state 1. Iterating, since $P_{1j} = \left(\frac{2}{5}, \frac{1}{2}, \frac{1}{10}\right)$, we have $a = \frac{2}{5}$, $b = \frac{1}{2}$ and $c = \frac{1}{10}$. Since $u_3 = 0.146 < a$, we select $X_2 = 1$.

We thus obtain

X_0	X_1	X_2	X_3	X_4	X_5	...
2	1	1	3	3	2	...

which we call a a **realization** of the chain. By simulating the process we obtain a realization of the process. We see from the simulation procedure that the stochastic process satisfies the Markov condition: we only used the value of the last state X_n to calculate the distribution forX_{n+1}.

6.1.2 The Distribution of X_n

Consider $\Pr(X_n = j)$, which is the probability that after a simulation of a Markov chain for n steps, the state reached is $X_n = j$. These probabilities may be arranged in a **row vector** $\pi^{(n)}$ where, by definition $\pi_j^{(n)} = \Pr(X_n = j)$. When $n = 1$, we see that

$$\begin{aligned}\Pr(X_1 = j) &= \sum_{i\in\Omega} \Pr(X_1 = j,\ X_0 = i) \\ &= \sum_{i\in\Omega} \Pr(X_1 = j | X_0 = i) \Pr(X_0 = i). \end{aligned} \tag{6.5}$$

This may be written in matrix form as

$$\pi_j^{(1)} = \sum_{i\in\Omega} \pi_i^{(0)} P_{ij} \text{ or } \pi^{(1)} = \pi^{(0)}\mathbf{P}. \tag{6.6}$$

Similarly,

$$\pi^{(n)} = \pi^{(n-1)}\mathbf{P}. \tag{6.7}$$

Suppose that for some π, we have that

$$\pi = \pi\mathbf{P} \tag{6.8}$$

i.e., π is a left eigenvector of $\mathbf{P}$ with eigenvalue 1, normalized such that $\sum \pi_i = 1$. Then π is called a **stationary distribution** for $\mathbf{P}$, since if $\pi^{(n)} = \pi$, then $\pi^{(n+1)} = \pi^{(n)}\mathbf{P} = \pi$ also, i.e., once the chain is in distribution π, it stays in that distribution.

Example 6.2 $\pi = \left(\frac{5}{18}, \frac{11}{18}, \frac{1}{9}\right)$ is the stationary distribution for the matrix in the first example, i.e.,

$$\begin{pmatrix} \frac{5}{18} & \frac{11}{18} & \frac{1}{9} \end{pmatrix} = \begin{pmatrix} \frac{5}{18} & \frac{11}{18} & \frac{1}{9} \end{pmatrix} \begin{pmatrix} \frac{2}{5} & \frac{1}{2} & \frac{1}{10} \\ \frac{1}{5} & \frac{7}{10} & \frac{1}{10} \\ \frac{2}{5} & \frac{2}{5} & \frac{1}{5} \end{pmatrix} \tag{6.9}$$

Definition 6.1 Suppose that $\pi^{(n)} \to \pi$ as $n \to \infty$ for any $\pi^{(0)}$. Then π is the **equilibrium distribution** of the chain $\mathcal{M}$ and the chain is said to be **ergodic.**

For an ergodic chain, and for sufficiently large n, the states of $\mathcal{M}$ are distributed like π and the system is "in equilibrium".

Exercise 6.1 Suppose that $\mathcal{M}$ is ergodic with equilibrium distribution π. Show that as $n \to \infty$,

$$\mathbf{P}^n \to \begin{pmatrix} \cdots & \pi & \cdots \\ \cdots & \pi & \cdots \\ & \vdots & \\ \cdots & \pi & \cdots \end{pmatrix}. \tag{6.10}$$

where $\mathbf{P}^n = \mathbf{P}\mathbf{P}\ldots\mathbf{P}$ matrix-multiplied n times.

Solution: By iterating $\pi^{(n)} = \pi^{(n-1)}\mathbf{P}$ we see that $\pi^{(n)} = \pi^{(0)}\mathbf{P}^n$. If $\mathcal{M}$ is ergodic, then $\pi^{(n)} \to \pi$ for any $\pi^{(0)}$. In particular if we choose $\pi^{(0)} = (0, 0, ..., 1, ..., 0)$ where the 1 is in the k'th location, $\pi^{(0)}\mathbf{P}^n$ is equal to the k'th row of $\mathbf{P}^n$ and, by assumption, this tends to π as $n \to \infty$. Since this holds for every k, the matrix $\mathbf{P}^n$ has the form shown.

Exercise 6.2 Verify that

$$\begin{pmatrix} \frac{2}{5} & \frac{1}{2} & \frac{1}{10} \\ \frac{1}{5} & \frac{7}{10} & \frac{1}{10} \\ \frac{2}{5} & \frac{2}{5} & \frac{1}{5} \end{pmatrix}^n \to \frac{1}{18} \begin{pmatrix} 5 & 11 & 2 \\ 5 & 11 & 2 \\ 5 & 11 & 2 \end{pmatrix}$$

as $n \to \infty$ as follows from the examples above.

Exercise 6.3 Determine the stationary distribution if $\mathbf{P} = \begin{pmatrix} 1-a & a \\ b & 1-b \end{pmatrix}$ where $0 \leq a, b \leq 1$.

Solution: If $0 < a+b < 2$, then $\pi = \begin{pmatrix} b/(a+b) & a/(a+b) \end{pmatrix}$ is the stationary distribution. If $a = b = 0$, then $\Omega = \{0, 1\}$ is **reducible** under $\mathbf{P}$, i.e., depending upon the initial state, $\pi^{(n)} \to \begin{pmatrix} 1 & 0 \end{pmatrix}$ or $\pi^{(n)} \to \begin{pmatrix} 0 & 1 \end{pmatrix}$ and $\pi = c \begin{pmatrix} 0 & 1 \end{pmatrix} + d \begin{pmatrix} 0 & 1 \end{pmatrix}$ is a stationary distribution for any c and d. On the other hand, if $a = b = 1$, the Markov chain is periodic.

6.1.3 Equilibrium Distributions

Under what circumstances does a stationary distribution exist ? If one exists, is it unique ? Does $\pi^{(n)} \to \pi$ for any initial $\pi^{(0)}$? We seek sufficient conditions for ergodicity.

Irreducibility

If we can find a sequence of states

$$i \to k_1 \to k_2 \to \cdots \to k_n \to j \tag{6.11}$$

such that the transition probabilities $P_{i,k_1} \neq 0$, $P_{k_m,k_{m+1}} \neq 0$, $P_{k_n,j} \neq 0$, then there is a sequence of states from i to j with a non-zero probability of occurring in $\mathcal{M}$. We say that "i and j communicate" and write $i \to j$. If j and i also communicate, i.e., if $j \to i$, we say that i and j "intercommunicate" and write $i \leftrightarrow j$. Sets of intercommunicating states form equivalence classes, since $i \leftrightarrow m$ and $m \leftrightarrow j \Rightarrow i \leftrightarrow j$, and likewise $i \leftrightarrow m$ but $m \not\leftrightarrow j \Rightarrow i \not\leftrightarrow j$.

If all states in Ω intercommunicate, then Ω is said to be **irreducible under P**, i.e., for any two states i and j in Ω, there is a path with non-zero probability which links i to j and a path with non-zero probability which links j to i. Otherwise, Ω is **reducible under P**.

If there is more than one distinct equivalence class of intercommunicating states in $\boldsymbol{\Omega}$, the Markov chain is reducible under $\mathbf{P}$ and a stationary distribution need not be unique.

Example 6.3 Suppose that $\Omega = \{1, 2, 3, 4\}$ and that

$$\mathbf{P} = \begin{pmatrix} 0.4 & 0.6 & 0 & 0 \\ 0.2 & 0.8 & 0 & 0 \\ 0 & 0 & 0.4 & 0.6 \\ 0 & 0 & 0.2 & 0.8 \end{pmatrix} \tag{6.12}$$

Then it is clear (from the associated digraph) that $\{1, 2\}$ and $\{3, 4\}$ are the equivalence classes of intercommunicating states. There are two left eigenvectors of $\mathbf{P}$ with eigenvalue 1, namely $\sigma = \begin{pmatrix} \frac{1}{4} & \frac{3}{4} & 0 & 0 \end{pmatrix}$ and $\rho = \begin{pmatrix} 0 & 0 & \frac{1}{4} & \frac{3}{4} \end{pmatrix}$. If the initial state $X_0 \in \{1, 2\}$, the stationary distribution is σ and if $X_0 \in \{3, 4\}$ the stationary distribution is ρ.

Reversibility

If I give you a realization of a reversible Markov process, but don't tell you in which direction the chain was simulated, you will not be able to figure out the simulation direction by looking at the sequence of states in the realization. This is a rare and special property.

Consider, for example, a more hum-drum chain on $\Omega = \{1, 2, 3\}$ with the transition matrix

$$\mathbf{P} = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}. \tag{6.13}$$

This defines an irreducible chain. We notice that the sequence $1 \to 3 \to 2 \to 1$ is possible in this chain but the reverse sequence $1 \to 2 \to 3 \to 1$ is not possible since $P_{23} = 0$. Thus there is a sequence of states for which it is possible to tell in which direction the simulation occurred and the chain is not reversible.

Lemma 6.1.1 Let $\mathcal{M} = \{X_n\}_{n=-\infty}^{\infty}$ be a homogeneous Markov chain with transition matrix $\mathbf{P}$. Then if we define the random variables $Y_n = X_{-n}$, the sequence $\mathcal{M}' = \{Y_n\}_{n=-\infty}^{\infty}$ is also a Markov chain, called the **reversed chain** of $\mathcal{M}$.

Proof: Consider the conditional probability $\Pr(Y_{n+1}|Y_n, Y_{n-1}, ...)$. We wish to show that this is equal to $\Pr(Y_{n+1}|Y_n)$. By definition of conditional probabilities, we see that

$$\begin{aligned} \Pr(Y_{n+1}|Y_n, Y_{n-1}, ...) &= \frac{\Pr(Y_{n+1}, Y_n, Y_{n-1}, ...)}{\Pr(Y_n, Y_{n-1}, ...)} \\ &= \frac{\Pr(Y_{n-1}, Y_{n-2}, ...|Y_{n+1}, Y_n)\Pr(Y_{n+1}, Y_n)}{\Pr(Y_n, Y_{n-1}, ...)} \\ &= \frac{\Pr(X_{-n+1}, X_{-n+2}, ...|X_{-n-1}, X_{-n})\Pr(Y_{n+1}, Y_n)}{\Pr(Y_n, Y_{n-1}, ...)} \end{aligned} \tag{6.14}$$

Using the Markov property of $\mathcal{M}$ we have that

$$\Pr(X_{-n+1}, X_{-n+2}, ...|X_{-n-1}, X_{-n}) = \Pr(X_{-n+1}, X_{-n+2}, ...|X_{-n}), \tag{6.15}$$

hence

$$\begin{aligned} \Pr(Y_{n+1}|Y_n, Y_{n-1}, ...) &= \frac{\Pr(X_{-n+1}, X_{-n+2}, ...|X_{-n})\Pr(Y_{n+1}, Y_n)}{\Pr(Y_n, Y_{n-1}, ...)} \\ &= \frac{\Pr(Y_{n-1}, Y_{n-2}, ...|Y_n)\Pr(Y_{n+1}, Y_n)}{\Pr(Y_n, Y_{n-1}, ...)} \\ &= \frac{\Pr(Y_{n+1}, Y_n)}{\Pr(Y_n)} = \Pr(Y_{n+1}|Y_n). \end{aligned} \tag{6.16}$$

Thus $\mathcal{M}'$ is also a Markov chain.

Definition 6.2 A homogeneous Markov chain $\mathcal{M}$ is **reversible** if the transition matrix for the reversed chain $\mathcal{M}'$ coincides with that for $\mathcal{M}$, so that

$$\Pr(X_{n+1} = j|X_n = i) = \Pr(X_n = j|X_{n+1} = i)$$

If the transition matrices are the same, the value of any statistic we care to measure on a realization will have the same distribution, whether the realization came from the forward or reversed chain. That's why we cant determine what the simulation direction was by looking at the output.

Necessary and sufficient conditions for reversibility

Theorem 6.1 Suppose that $\mathcal{M} = \{X_n\}_{n=-\infty}^{\infty}$ is a Markov chain with transition matrix $\mathbf{P}$, unique stationary distribution π, and that for all n, X_n is distributed as π. $\mathcal{M}$ is reversible iff

$$\pi_i P_{ij} = \pi_j P_{ji} \text{ for all } i, j \in \Omega. \tag{6.17}$$

This is often called the **detailed balance condition.**

Proof: Let $\mathbf{Q} = (Q_{ij})$ be the transition matrix of the reversed chain $\mathcal{M}' = \{Y_n\}_{n=-\infty}^{\infty}$ where $Y_n = X_{-n}$, i.e.,

$$Q_{ij} = \Pr\left(Y_{n+1} = j | Y_n = i\right). \tag{6.18}$$

We must prove that $Q_{ij} = P_{ij}$ iff detailed balance holds. By definition of the process Y,

$$\begin{aligned} Q_{ij} &= \Pr\left(X_{-n-1} = j | X_{-n} = i\right) \\ &= \frac{\Pr\left(X_{-n-1} = j, X_{-n} = i\right)}{\Pr\left(X_{-n} = i\right)} \\ &= \frac{\Pr\left(X_{-n} = i | X_{-n-1} = j\right) \Pr\left(X_{-n-1} = j\right)}{\Pr\left(X_{-n} = i\right)} \end{aligned} \tag{6.19}$$

which is essentially a statement of Bayes' theorem. Since each X_n is distributed as π for all n,

$$\begin{aligned} Q_{ij} &= \frac{\Pr\left(X_{-n} = i | X_{-n-1} = j\right) \pi_j}{\pi_i} \\ &= \frac{P_{ji}\pi_j}{\pi_i}, \end{aligned} \tag{6.20}$$

by definition of the transition matrix $\mathbf{P}$ of $\mathcal{M}$. Then $Q_{ij} = P_{ij}$ and the chain is reversible if and only if detailed balance holds.

Example 6.4 Ehrenfest model of Diffusion

Given the microscopic dynamics of a reversible physical process, this example shows how the stationary distribution may be found by using the detailed balance condition.

Consider two boxes A and B connected to each other. Within the two boxes are m particles. At time $t \in \mathbb{Z}$, let X_t represent the number of particles in box A, so that there are $m - X_t$ particles in box B. Select one of the m particles at random and transfer it to the other box (whichever that is). This leads to the situation at time $t + 1$.

We shall verify that $\mathcal{M} = \{X_t\}_{t=0}^{\infty}$ is a homogeneous Markov chain. Its transition matrix $\mathbf{P}$ satisfies $P_{ij} = 0$ unless $j = i - 1$ or $j = i + 1$. Now, $P_{i,i-1}$ is the probability that $X_{t+1} = X_t - 1 = i - 1$ or the probability that the randomly selected particle is in box A which is i/m. Similarly, $P_{i,i+1} = (m - i)/m$.

Any given sequence of moves is equally likely to occur forward or in reverse (imagine putting a label on each ball), so the process is reversible. Thus the stationary distribution π must satisfy the detailed balance condition. Setting $j = i + 1$ in the condition,

$$\pi_i P_{i,i+1} = \pi_{i+1} P_{i+1,i} \tag{6.21}$$

or

$$\pi_i \left(\frac{m-i}{m} \right) = \pi_{i+1} \left(\frac{i+1}{m} \right)$$

$$\pi_{i+1} = \left(\frac{m-i}{i+1} \right) \pi_i \tag{6.22}$$

Using this relationship, we see that

$$\pi_1 = m\pi_0,\ \pi_2 = \frac{m-1}{2}\pi_1,\ \pi_3 = \frac{m-2}{3}\pi_2, ... \tag{6.23}$$

so that

$$\pi_i = \frac{m\,(m-1)\cdots(m-i+1)}{i!}\pi_0 = \binom{m}{i}\pi_0. \tag{6.24}$$

We can find all π_i by the normalization condition since

$$1 = \sum_{i=0}^{m} \pi_i = \sum_{i=0}^{m} \binom{m}{i}\pi_0 = 2^m \pi_{0.} \tag{6.25}$$

Hence the desired stationary distribution is

$$\pi_i = 2^{-m}\binom{m}{i}, \tag{6.26}$$

which is a binomial distribution $B\left(m, \frac{1}{2}\right)$.

Periodic chains (a technical detail)

If the transition matrix $\mathbf{P}$ of a Markov chain has a zero diagonal, i.e., if $P_{ii} = 0$ for all i, the chain may be **periodic**. Let $\mathcal{M}$ be an irreducible Markov chain with transition matrix $\mathbf{P}$ and let i be a fixed state. Define the set

$$T = \left\{ k : \left(\mathbf{P}^k \right)_{ii} > 0,\ k > 0 \right\}. \tag{6.27}$$

These are the steps on which it is possible for a chain which starts in state i to revisit i. The greatest common divisor of the integers in T is called the **period** of the state i.

It is possible to show that the period of an irreducible chain for the state i is a constant independent of i. The chain is said to be periodic if the period of any of its states is greater than one.

Example 6.5 Consider the chain defined by the state space $\Omega = \{1, 2, 3, 4\}$, the transition matrix

$$\mathbf{P} = \begin{pmatrix} 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{pmatrix}, \tag{6.28}$$

and the initial state $X_0 = 1$.

This is periodic with $d = 2$ since

$$\Pr\left(X_n = 2 | X_0 = 1\right) = \begin{cases} 0 & \text{if } n \text{ is even} \\ \frac{1}{2} & \text{if } n \text{ is odd} \end{cases} \tag{6.29}$$

Periodic chains are not ergodic, though the difficulty can be eliminated by sub-sampling.

Ergodicity theorem for reversible chains

Theorem 6.2 For an irreducible, aperiodic Markov chain $\mathcal{M}$ on a countable state space with transition matrix $\mathbf{P}$, if there exists $\pi = (\pi_i)$ such that $0 \leq \pi_i \leq 1$, $\sum_i \pi_i = 1$ and

$$\pi_i P_{ij} = \pi_j P_{ji}, \tag{6.30}$$

then $\mathcal{M}$ is reversible and ergodic with unique equilibrium distribution π.

Notice that if we sum the detailed balance condition with respect to j, we obtain

$$\pi_i \left(\sum_j P_{ij} \right) = \sum_j \pi_j P_{ji} \tag{6.31}$$

or

$$\pi_i = \sum_j \pi_j P_{ji} \tag{6.32}$$

which shows that π is a stationary distribution. In general, given a stochastic process with transition matrix $\mathbf{P}$, it is very difficult to find the equilibrium distribution π unless the chain is reversible. We have also assumed that Ω is countable and that $\mathcal{M}$ is homogeneous.

Exercise 6.4 Suppose $\mathcal{M}$ is an irreducible Markov chain on a finite space of N states, with $P_{ji} = P_{ij}$, *ie* the transition matrix is symmetric. Prove $\mathcal{M}$ is reversible, and that the uniform distribution $\pi_i = 1/N$ is its unique equilibrium distribution.

Solution: See Example 6.6 in the next section.

Reading

G.R. Grimmet and D.R. Stirzaker, *Probability and Random Processes.*

6.2 Markov Chain Monte Carlo

6.2.1 Some special cases

In the previous section, we considered the problem of finding the equilibrium distribution π of a chain with a given transition matrix $\mathbf{P}$. In this section, we consider the reverse problem. Given a distribution π, how do we construct the transition matrix $\mathbf{P}$ of a Markov chain so that the equilibrium distribution of this Markov chain is π?

A Markov chain can be specified by a transition matrix, or by giving the microscopic dynamics (*ie*, an algorithm which determines X_{n+1} given X_n). The algorithm implicitly fixes the transition matrix. Real problems are usually too complex for a transition matrix to be given explicitly. So the problem of specifying a MC with a given desired equilibrium distribution will boil down to the problem of providing an algorithm which we can prove generates a MC with the right equilibrium distribution. We use the idea of **reversibility.**

Example 6.6 Construct an ergodic, reversible Markov chain $\mathcal{M}$ on the state space $\Omega = \{1, 2, 3, 4\}$ with equilibrium distribution $\pi_i = \frac{1}{4}$, (i.e., uniform on Ω.)

Since $\mathcal{M}$ is to be reversible, the transition matrix must satisfy

$$\pi_i P_{ij} = \pi_j P_{ji}. \tag{6.33}$$

In order for $\pi_i = \pi_j$, we have $P_{ij} = P_{ji}$. If Ω is irreducible under $\mathbf{P}$ (i.e., the chain can get to any state in Ω) then we have satisfied the conditions of the ergodicity theorem for reversible chains. So any **symmetric, irreducible** transition matrix $\mathbf{P}$ will do the job.

If for example,

$$\mathbf{P} = \begin{pmatrix} 3/4 & 1/4 & 0 & 0 \\ 1/4 & 1/2 & 1/4 & 0 \\ 0 & 1/4 & 1/2 & 1/4 \\ 0 & 0 & 1/4 & 3/4 \end{pmatrix} \tag{6.34}$$

we satisfy $\sum_j P_{ij} = 1$ and $P_{ij} = P_{ji}$. Now $\pi = \begin{pmatrix} 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix}$ is a left eigenvector of $\mathbf{P}$. All the conditions for ergodicity are satisfied so we expect that $\pi^{(n)} = \pi^{(0)}\mathbf{P}^n$ tends to $\begin{pmatrix} 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix}$ as $n \to \infty$ from any start. Explicitly, we find that

$$\mathbf{P}^2 = \begin{pmatrix} .625 & .3125 & .0625 & 0 \\ .3125 & .375 & .25 & .0625 \\ .0625 & .25 & .375 & .3125 \\ 0 & .0625 & .3125 & .625 \end{pmatrix}, \tag{6.35}$$

$$\mathbf{P}^4 = \begin{pmatrix} .49219 & .32813 & .14063 & .03906 \\ .32813 & .30469 & .22656 & .14063 \\ .14063 & .22656 & .30469 & .32813 \\ .03906 & .14063 & .32813 & .49219 \end{pmatrix}, \tag{6.36}$$

$$\mathbf{P}^{100} = \begin{pmatrix} .25000\,00567 & .25000\,00235 & .24999\,99765 & .24999\,99433 \\ .25000\,00235 & .25000\,00097 & .24999\,99903 & .24999\,99765 \\ .24999\,99765 & .24999\,99903 & .25000\,00097 & .25000\,00235 \\ .24999\,99433 & .24999\,99765 & .25000\,00235 & .25000\,00567 \end{pmatrix}. \tag{6.37}$$

Note that this problem is trivial as we could have simply taken $P_{ij} = 1/4$ for all i, j in Ω. At each update, we sample uniformly on Ω. Then $\pi^{(n)} = \pi$ for all n.

Exercise 6.5 Consider the state space $\Omega = \{(a, b, c) : a + b + c = 0 \text{ and } a, b, c \in \{-9, -8, ..., 8, 9\}\}$. Construct a reversible Markov chain with an equilibrium distribution which is uniform on Ω.

Solution: In order to move from state $X_n = i$ to state $X_{n+1} = j$ where i and j are in Ω, pick a vector $\mathbf{u}$ with elements $(0, 1, -1)$ in random order, and set $j = i + \mathbf{u}$. This move respects the constraint that $a + b + c = 0$. If $i + \mathbf{u}$ is not in Ω, the move is not made and we set $X_{n+1} = X_n = i$. There are six choices for $\mathbf{u}$. We notice that

1. For each pair of states i and j, there is either one or no vector $\mathbf{u}$ of the above form which relates them. If $j = i + \mathbf{u}$, then $i = j + (-\mathbf{u})$. Since the probability to pick $\mathbf{u}$ is the same as the probability to pick $-\mathbf{u}$, we see that $P_{ij} = P_{ji} = \frac{1}{6}$ if there is a $\mathbf{u}$ such that $i = j + \mathbf{u}$. Also, $P_{ij} = P_{ji} = 0$ if no such $\mathbf{u}$ exists.

2. The rule that we do not make moves taking us outside Ω only affects the probability P_{ii} for some i and does not spoil the symmetry of the transition matrix $\mathbf{P}$.

Since $\mathbf{P}$ is symmetric, it is clear that a uniform distribution π on Ω will satisfy $\pi_i P_{ij} = \pi_j P_{ji}$. The chain is clearly irreducible as for any pair of states i and j in Ω, there is a sequence of valid vectors $\mathbf{u}$ which can take us from i to j. By the ergodicity theorem for reversible chains,

$$\Pr(X_n = i) \to \frac{1}{|\Omega|}, \text{ i.e., uniform on } \Omega \tag{6.38}$$

for any initial distribution. Note that $|\Omega|$ denotes the number of states in Ω.

6.2.2 Metropolis-Hastings Markov Chain Monte Carlo

We seem to be able to handle requests for samples from a uniform distribution. How about generating non-uniform distributions ? Metropolis-Hastings Markov Chain Monte Carlo is a certain type of algorithm which generates a MC with equilibrium distribution π. The user gets to say what kind of pie they want. The algorithm is as follows:

Let $X_n = i$. X_{n+1} is determined in the following way.

1. **Generation step:** Generate a candidate state j from i with some distribution $g(j|i)$. $g(j|i)$ is a fixed distribution that we are free to choose, so long that it satisfies the conditions

   a) $g(j|i) = 0 \Rightarrow g(i|j) = 0$, (cant go forward implies cant go back)

   b) $g(j|i)$ is the transition matrix of an irreducible Markov chain on Ω.

2. **Acceptance step:** With probability

$$\alpha(j|i) \equiv \min\left\{1, \frac{\pi_j}{\pi_i}\frac{g(i|j)}{g(j|i)}\right\}, \tag{6.39}$$

   set $X_{n+1} = j$ (i.e., "accept" j), otherwise set $X_n = i$ (i.e., "reject" j).

Note that since the generation and acceptance steps are independent, the transition probability to go from i to j is

$$\Pr(X_{n+1} = j | X_n = i) = g(j|i)\,\alpha(j|i) \text{ provided that } i \neq j. \tag{6.40}$$

This says that the probability to land in state j from the current state i is equal to the probability to generate j from the current state i times the probability that the new state j is accepted. The probability P_{ii} can be found from the requirement that $\sum_j P_{ij} = 1$.

Assertion: (Metropolis *et al.* 1953, Hastings 1970)

Let π be a given probability distribution. The Markov chain simulated by the Metropolis-Hastings algorithm is reversible with respect to π. If it is also irreducible and aperiodic, then it defines an ergodic Markov chain with unique equilibrium distribution π.

Proof: We have to show that the transition matrix $\mathbf{P}$ determined by the MH algorithm satisfies

$$\pi_i P_{ij} = \pi_j P_{ji} \tag{6.41}$$

for all $i \neq j$ (since the case for $i = j$ is trivial). If this is the case, then the chain is reversible and the rest of the assertion follows from the ergodicity theorem of the first section.

Assume without loss of generality that

$$\pi_j g\left(i|j\right) > \pi_i g\left(j|i\right). \tag{6.42}$$

Since

$$\begin{aligned} P_{ij} &= g\left(j|i\right)\alpha\left(j|i\right) \\ &= g\left(j|i\right)\min\left\{1, \frac{\pi_j}{\pi_i}\frac{g\left(i|j\right)}{g\left(j|i\right)}\right\} = g\left(j|i\right), \end{aligned} \tag{6.43}$$

by assumption. On the other hand

$$\begin{aligned} P_{ji} &= g\left(i|j\right)\alpha\left(i|j\right) \\ &= g\left(i|j\right)\min\left\{1, \frac{\pi_i}{\pi_j}\frac{g\left(j|i\right)}{g\left(i|j\right)}\right\} = g\left(i|j\right)\frac{\pi_i}{\pi_j}\frac{g\left(j|i\right)}{g\left(i|j\right)} \\ &= \frac{\pi_i}{\pi_j}g\left(j|i\right) \end{aligned} \tag{6.44}$$

again by assumption. From these expressions, it is clear that $\pi_i P_{ij} = \pi_j P_{ji}$ as required.

Note that if $g\left(j|i\right)$ is not irreducible, then P_{ij} is not irreducible. However, irreducibility of $g\left(j|i\right)$ is not sufficient to guarantee irreducibility of P_{ij} since it may be that $\alpha\left(j|i\right) = 0$ (if $\pi_j = 0$), and this may prevent the chain from reaching all states in Ω.

Example 6.7 Use the Metropolis-Hastings method to construct a reversible, ergodic Markov chain on $\Omega = \{1, 2, 3\}$ with equilibrium distribution $\pi = \left(\frac{5}{18}, \frac{11}{18}, \frac{2}{18}\right)$.

Solution: Let us suppose that $g_{ij} \equiv g\left(j|i\right) = \frac{1}{3}$ for all i and j, i.e., the generation step picks a candidate state uniformly from Ω. By the definition of $\alpha_{ij} \equiv \alpha\left(j|i\right)$, we see that

$$\alpha_{ij} = \min\left\{1, \frac{\pi_j}{\pi_i}\right\} = \begin{pmatrix} 1 & 1 & 2/5 \\ 5/11 & 1 & 2/11 \\ 1 & 1 & 1 \end{pmatrix}. \tag{6.45}$$

Since $P_{ij} = g_{ij}\alpha_{ij}$ for $i \neq j$ and $P_{ii} = 1 - \sum_{j \neq i} P_{ij}$, gives the diagonal entries, we calculate

$$\mathbf{P} = \begin{pmatrix} 8/15 & 1/3 & 2/15 \\ 5/33 & 26/33 & 2/33 \\ 1/3 & 1/3 & 1/3 \end{pmatrix}.$$

It is easy to check that $\pi\mathbf{P} = \pi$ and that the rows of $\mathbf{P}^n$ tend to π as n becomes large.

Example 6.8 Construct the transition matrix $\mathbf{P}$ of a Markov chain with state space $\{0, 1, 2, ...\}$ and equilibrium distribution

$$\pi_i = \frac{\mu^i \exp\left(-\mu\right)}{i!}, \tag{6.46}$$

i.e., a Poisson distribution with mean $\mu \in \mathbb{R}^+$.

As usual, there are two parts, an update scheme to generate candidate states and a scheme to accept or reject these states.

Let $X_n = i$. X_{n+1} is determined in the following way.

1. **Generation:** Any simple scheme will do, e.g.,

$$g(j|i) = \begin{cases} 1/2 & \text{if } j = i+1 \\ 1/2 & \text{if } j = i-1 \\ 0 & \text{otherwise} \end{cases} \tag{6.47}$$

i.e., given $X_n = i$, the candidate state for X_{n+1} is chosen uniformly from $\{i+1, i-1\}$.

2. **Accept / reject:** The acceptance probability is determined by the Metropolis-Hastings formula

$$\alpha(j|i) = \min\left\{1, \frac{\pi_j}{\pi_i}\frac{g(i|j)}{g(j|i)}\right\}. \tag{6.48}$$

Now $g(i|j) = g(j|i) = 1/2$ so

$$\alpha(i+1|i) = \min\left\{1, \frac{\pi_{i+1}}{\pi_i}\right\} = \min\left\{1, \frac{\mu}{i+1}\right\}, \tag{6.49}$$

and

$$\alpha(i-1|i) = \min\left\{1, \frac{\pi_{i-1}}{\pi_i}\right\} = \min\left\{1, \frac{i}{\mu}\right\}. \tag{6.50}$$

Note that if $X_n = 0$ and the candidate state $j = -1$ is chosen, we reject it and set $X_{n+1} = 0$. This affects P_{ii} when $i = 0$ but leaves P_{ij} correct for detailed balance to hold.

Exercise 6.6 Implement the algorithm for the above example in you favorite computer language. Using $\mu = 1$, check the mean and variance of the samples $X_0 \ldots X_n$ are close to μ when n is large.

Example 6.9 The Binary Markov Random Field

Consider the space $\Omega = \{(x_1, x_2, ..., x_{N^2}) : x_m \in \{-1, 1\}\}$. We may regard x_m as the colour of the m'th pixel where $x_m = -1$ denotes a black pixel and $x_m = 1$ denotes a white pixel. Then Ω is the space of all black and white images (Figure 6.2) with N^2 pixels. Suppose that

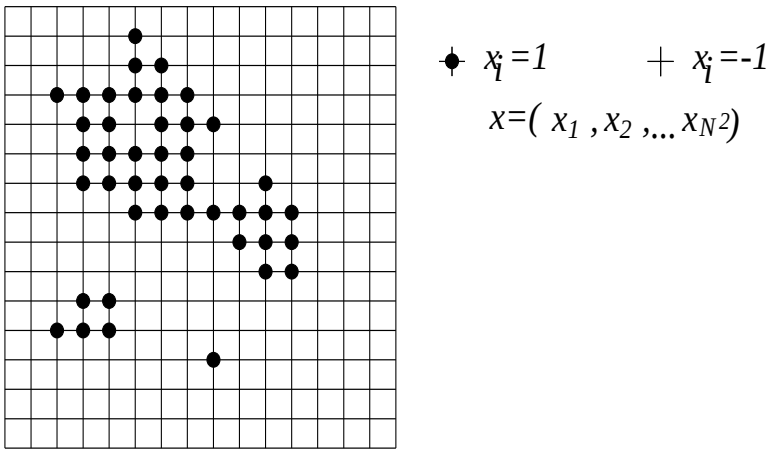


Figure 6.2: The Ising model on a square lattice with free boundaries.

the probability of $x \in \Omega$ is given by

$$\Pr(x) = \frac{1}{\mathcal{Z}'} \exp\left(J \sum_{\langle m,n \rangle} x_m x_n \right) \tag{6.51}$$

$$= \frac{1}{\mathcal{Z}} \exp(-2J\#x) \tag{6.52}$$

for some $J > 0$ and where $\langle m, n \rangle$ indicates a sum over pixels m and n which are adjacent on the image lattice, and $\#x$ is the number of edges connecting pixels of unequal value. The constant $\mathcal{Z}$ normalizes the probability distribution.

Exercise 6.7 Show that $\sum_{\langle m,n \rangle} x_m x_n = -2\#x + 2N^2 - 2N$. What happened to $\exp(2N^2 - 2N)$?

The probability distribution above favours smooth images. We wish to generate samples $X \sim \Pr(X = x)$ from this distribution. Each sample X is an array of N^2 binary values $x = (x_1, x_2, ..., x_{N^2})$. The normalising constant $\mathcal{Z}$ is not in general available (though see L. Onsager, Phys Rev, vol65, pp117, 1944) and classical sampling methods will not work. The distribution (6.51) is called a binary Markov random field, or "Ising" model.

Exercise 6.8 Construct a reversible Markov chain on Ω with equilibrium distribution $\pi_x = \Pr(x)$.

Solution: We use the Metropolis Hastings prescription:

Let $X_n = x$. X_{n+1} is determined in the following way.

1. Generate a candidate state x' from x using some process we can choose. Here is a simple method: given $x = (x_1, x_2, ..., x_{N^2})$, pick a pixel n at random from $1, 2, ..., N^2$. Set $x' = (x_1, x_2, ..., -x_n, ..., x_{N^2})$. Notice that

   a) we can get to any state in Ω by a sequence of such updates,

   b) if x' and x differ by more than one pixel, $g(x'|x) = g(x|x') = 0$,

   c) if x' and x differ by exactly one pixel, $g(x'|x) = g(x|x') = 1/N^2$,

   so our generation scheme is suitable for MH MCMC.

2. Work out the Metropolis-Hastings acceptance probability with $\pi_x = \Pr(x)$.

$$\alpha(x'|x) = \min\left\{1, \frac{\pi_{x'}}{\pi_x} \frac{g(x|x')}{g(x'|x)}\right\}$$

$$= \min\left\{1, \exp\left(-2J(\#x' - \#x)\right)\right\} \tag{6.53}$$

   Notice that both the g's and the normalization $\mathcal{Z}$ cancel. Since x and x' are the same except at x_n where $x'_n = -x_n$,, we write $\#x' - \#x = \#x'_n - \#x_n$ where $\#x_n$ is the number of disagreeing neighbours around x_n. So we set $X_{n+1} = x'$ with probability

$$\alpha(x'|x) = \min\{1, \exp(-2J\#\Delta_n)\} \tag{6.54}$$

   (where $\#\Delta_n = \#x'_n - \#x_n$ is the change in the number of disagreeing neighbours at pixel n). Otherwise we set $X_{n+1} = x$, *ie* no change.

Notice that

1. If $\#\Delta_n < 0$, i.e., the change from x to x' gives a smoother image with fewer disagreeing neighbours, then $\alpha = 1$ and the proposed change is accepted with probability 1.
2. If $\#\Delta_n > 0$, the change leads to a more irregular image with fewer agreeing neighbours. Then $\alpha = \exp(-2J\#\Delta_n)$ and the proposed change is accepted with probability < 1.

Algorithm for generating a realization of the Markov chain

A Matlab script file for generating realizations from a random binary Markov field is

```
Xpix = 64;
Ypix = 64;
J = 1;
l = 0;
F = -ones(Ypix,Xpix);
while 1,
   for k = 1:4096
      % Select a pixel at random
      ix = ceil(Xpix*rand(1)); iy = ceil(Ypix*rand(1));
      Fc = F(iy,ix); pos = (ix-1)*Ypix + iy; % Index of pixel
      nbhrs = pos + [-1 1 -Ypix Ypix]; % Find indicies of neighbours
      nbhrs(find([iy==1 iy==Ypix ix==1 ix==Xpix])) = []; % Remove those outside picture
      nagree = sum(Fc==F(nbhrs)); ndisagree = sum(Fc~=F(nbhrs));
      change = nagree - ndisagree;
      if rand(1)<exp(-2*J*change) % if change<0, this happens with certainty
         F(iy,ix) = -Fc;
      end
      l = l + 1;
   end
   figure(1); image(63*F);colormap(gray); title(['Iteration ' num2str(l)]);
   drawnow
end
```

The only non-straightforward part is handling pixels on the edges which have fewer than four neighbours. If you run it you will see images sampled from $\Pr(x)$. Increasing the value J leads to images dominated by one color, as the average number of disagreeing edges decreases with increasing J. Some examples are given in Figure 6.3.

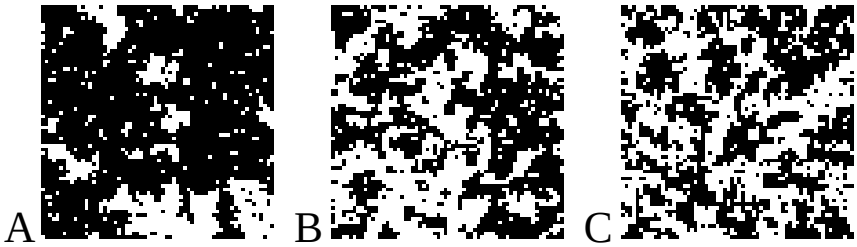


Figure 6.3: Samples $x \sim \exp(-2J\#x)/\mathcal{Z}$ with $N = 64$ and (A) $J = 0.45$ (B) $J = 0.4$ (C) $J = 0.35$.

7

Sampled solutions to Inverse Problems

7.1 Introduction

The end result of modeling in the Bayesian formalism is a posterior distribution $\Pr(\mathbf{f}|\mathbf{d})$ where $\mathbf{d}$ is a vector of data and $\mathbf{f}$ is a vector of parameter values of interest which is transformed by the physical process into the data. Observation of the data $\mathbf{d}$ involve a loss of information, as the transformation of $\mathbf{f}$ into $\mathbf{d}$ (the forward problem) may have random components ("noise") as well as a possible reduction of dimension. We assume that the physical process leading from $\mathbf{f}$ to $\mathbf{d}$ is understood, at least in a probabilistic sense, so that the forward probability function $\Pr(\mathbf{d}|\mathbf{f})$ may be calculated. In the Bayesian formalism, the forward probability function is regarded as a function of $\mathbf{f}$, and enters Bayes' theorem as the likelihood function. We then have

$$\Pr(\mathbf{f}|\mathbf{d}) \propto \Pr(\mathbf{d}|\mathbf{f}) \Pr(\mathbf{f}) \tag{7.1}$$

up to a normalizing constant independent of $\mathbf{f}$.

In many cases of interest, the full posterior probability distribution is hopelessly analytically intractable, since the number of components in $\mathbf{f}$ may be very large and the prior probability function $\Pr(\mathbf{f})$ may involve information which is difficult to express in analytic terms (e.g. in an image reconstruction problem, the true object $\mathbf{f}$ may be known to consist of a single "blob" of material so that its boundary is simply connected). How then do we get at $\mathbf{f}$**?** What are the most likely values for $\mathbf{f}$ given the data, and what are typical or "average" values? Our problem may be treated by simulation: it is enough that we can test the plausibility of any particular guess at $\mathbf{f}$ by simulating the physical process leading from $\mathbf{f}$ to $\mathbf{d}$. We draw samples from the set Ω of all possible $\mathbf{f}$'s, each sample drawn with probability $\Pr(\mathbf{f}|\mathbf{d})$. In this way we get a set $\mathbf{\Theta} = \{\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_N\}$ of samples distributed like the posterior distribution. Inference on $\Pr(\mathbf{f}|\mathbf{d})$ becomes inference on $\{\mathbf{f}_i\}_{i=1}^N$. For example, the mode and mean of the samples in $\mathbf{\Theta}$ give us an estimate of the most likely and average values of $\mathbf{f}$. Given some set of parameters A, we can estimate $\Pr(\mathbf{f} \in A|\mathbf{d})$ by calculating the fraction of samples which lie in the set A. Sampling-based methods are called **Monte-Carlo** methods.

Again, however, standard sampling techniques are useless when the posterior involves many variables and is otherwise intractable. In particular, one generally needs to have a closed form for the constant which normalizes the probability distribution we want to sample. (The exception to this, rejection sam1pling, is restricted in other respects). Markov chain sampling methods enable us to sample the kinds of complex distributions that "natural" models of physical processes tend to generate. In the previous chapter, we have considered Markov chains on discrete state spaces. In this chapter we shall also extend the MCMC method to continuous and mixed state spaces, allowing the solution of parameter estimation problems.

7.2 Recovering a binary matrix from noisy data

We first consider an example involving a finite (but large) state space based on the binary Markov random field described in the previous chapter. Consider the problem of reconstructing a binary image (i.e., one in which each pixel is either black or white) from observations of the image which are corrupted by independent zero-mean Gaussian noise. Let the image $\mathbf{f}$ be of size $M \times N$ pixels, and denote the value of the (m,n)'th pixel by $f_{mn} \in \{-1,1\}$. The state space of all images Ω thus has 2^{MN} elements. The observed data $\mathbf{d} = \{d_{mn}\}$ are related to $\mathbf{f}$ via

$$d_{mn} = f_{mn} + \varepsilon_{mn}$$

where each ε_{mn} is an independent sample of zero-mean Gaussian noise of variance σ^2. Note that although each f_{mn} is black (-1) or white $(+1)$, the values of d_{mn} can take on any real value. Figure 7.1 shows an example of the true image $\mathbf{f}$ and a data set $\mathbf{d}$ obtained for $\sigma = 2$

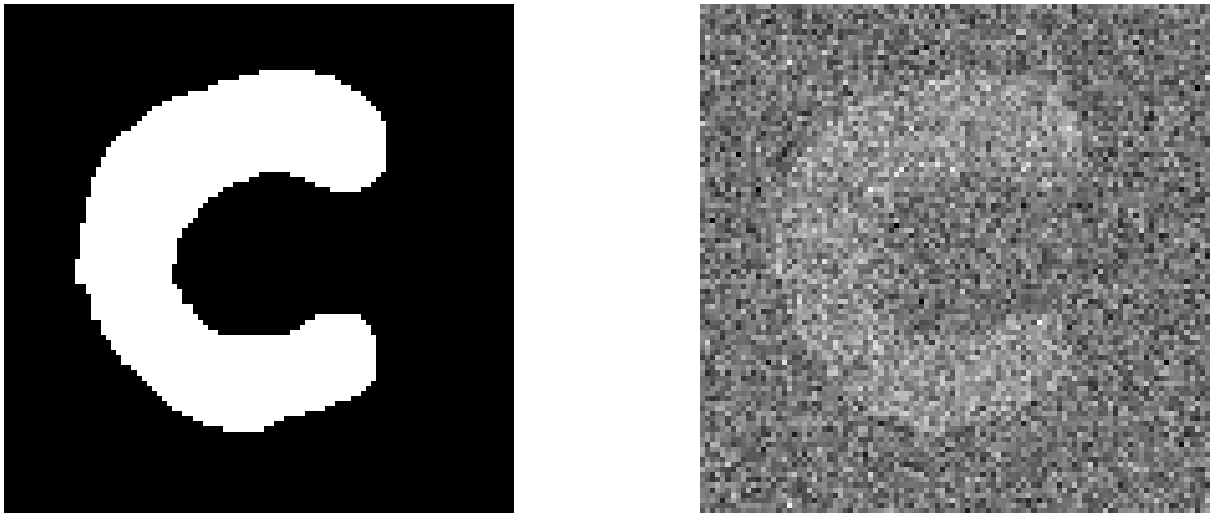


Figure 7.1: A binary image $\mathbf{f}$ and a noise-corrupted data set $\mathbf{d}$ obtained by adding samples of independent zero mean Gaussian noise of standard deviation $\sigma = 2$ to the image.

The likelihood function for this problem is given by

$$\Pr(\mathbf{d}|\mathbf{f}) \propto \exp\left[-\frac{1}{2\sigma^2}\sum_{m,n}(d_{mn} - f_{mn})^2\right].$$

In order to find the posterior probability function, a prior probability distribution $\Pr(\mathbf{f})$ is required. This encodes our state of knowledge about which images are (à priori, without any data) more likely to occur. For example, we might

1. have no prior prejudice whatsoever, i.e., $\Pr(\mathbf{f}) = 2^{-MN}$, which is uniform on Ω.
2. favour smooth images: since the material is likely to be in lumps, we regard reconstructions in which the 1's and -1's separate out in blobs as *a priori* more probable. In this case, the binary Markov random field of the last section might make a reasonable choice:

$$\Pr(\mathbf{f}) = \frac{1}{\mathcal{Z}}\exp(-2J\#\mathbf{f}), \tag{7.2}$$

where J is our lumping parameter. When $J = 0$, there is no smoothing, whereas if J is large, we favour a uniform image of a single colour. A model which favours "simple" reconstructions is called parsimonious.

7.2.1 Uniform Prior

If we use the uniform prior, the posterior probability is equal to the likelihood. An implementation of the Metropolis-Hastings Markov Chain Monte Carlo (MH MCMC) algorithm which draws samples from the posterior probability involves the following steps:

1. Let $X_n = \mathbf{f}$ denote the current state of the Markov chain. A pixel with coordinates kl is selected at random and the colour of the pixel is flipped, producing a candidate state $\mathbf{f}'$ where
$$f'_{ij} = \begin{cases} -f_{kl} & \text{if } i = k \text{ and } j = l \\ f_{ij} & \text{otherwise} \end{cases} \tag{7.3}$$
The generation probability $g(\mathbf{f}'|\mathbf{f})$ is zero if $\mathbf{f}'$ and $\mathbf{f}$ differ by more than one pixel, and is equal to $1/(MN)$ if they differ by exactly one pixel.

2. Calculate the acceptance probability $\alpha(\mathbf{f}'|\mathbf{f})$ using
$$\alpha\left(\mathbf{f}'|\mathbf{f}\right) = \min\left\{1, \frac{\Pr(\mathbf{f}'|\mathbf{d})}{\Pr(\mathbf{f}|\mathbf{d})} \frac{g\left(\mathbf{f}|\mathbf{f}'\right)}{g(\mathbf{f}'|\mathbf{f})}\right\} \tag{7.4}$$
If $\mathbf{f}'$ is generated as described above, the ratio $g(\mathbf{f}'|\mathbf{f})/g\left(\mathbf{f}|\mathbf{f}'\right) = 1$. The ratio of posterior probabilities is
$$\frac{\Pr(\mathbf{f}'|\mathbf{d})}{\Pr(\mathbf{f}|\mathbf{d})} = \frac{\Pr\left(\mathbf{d}|\mathbf{f}'\right)}{\Pr(\mathbf{d}|\mathbf{f})} \frac{\Pr(\mathbf{f}')}{\Pr(\mathbf{f})} \tag{7.5}$$
which for a uniform prior reduces to the likelihood ratio $\Pr\left(\mathbf{d}|\mathbf{f}'\right)/\Pr(\mathbf{d}|\mathbf{f})$.

When using the MH MCMC algorithm, calculating the ratio of posterior probabilities occurs on every step, and so should be done as efficiently as possible. For additive Gaussian noise,
$$\frac{\Pr\left(\mathbf{d}|\mathbf{f}'\right)}{\Pr(\mathbf{d}|\mathbf{f})} = \exp\left[-\frac{1}{2\sigma^2}\sum_{i,j}\left\{\left(d_{ij} - f'_{ij}\right)^2 - \left(d_{ij} - f_{ij}\right)^2\right\}\right]. \tag{7.6}$$
We can compute this much more efficiently by using the fact that the only term in the sum which changes when $\mathbf{f}$ is replaced by $\mathbf{f}'$ is that which involves the single pixel with indices k, l. Hence
$$\frac{\Pr\left(\mathbf{d}|\mathbf{f}'\right)}{\Pr(\mathbf{d}|\mathbf{f})} = \exp\left[-\frac{1}{2\sigma^2}\left\{\left(d_{kl} - f'_{kl}\right)^2 - \left(d_{ij} - f_{kl}\right)^2\right\}\right] = \exp\left[-\frac{1}{2\sigma^2}\left\{-2d_{kl}\left(f'_{kl} - f_{kl}\right) + \left(f'^2_{kl} - f^2_{kl}\right)\right\}\right] \tag{7.7}$$
Finally, using the fact that $f'_{kl} = -f_{kl}$, we find
$$\frac{\Pr\left(\mathbf{d}|\mathbf{f}'\right)}{\Pr(\mathbf{d}|\mathbf{f})} = \exp\left[\frac{d_{kl}\left(f'_{kl} - f_{kl}\right)}{\sigma^2}\right], \tag{7.8}$$
which can be computed very quickly.

3. Accept the candidate state $\mathbf{f}'$ with probability $\alpha(\mathbf{f}'|\mathbf{f})$. If the state is accepted, set $X_{n+1} = \mathbf{f}'$, otherwise set $X_{n+1} = \mathbf{f}$.

A Matlab implementation of the algorithm is given below, and the result of the reconstruction is shown in Figure 7.2.

```
F = imread('blob1.bmp');
[M,N] = size(F);
sigma = 2;
d = double(F); d(find(d==0))=-1;
d = d + sigma*randn(size(d));
figure(1);
subplot(1,2,1);imagesc(F);set(gca,'Visible','off');colormap gray; axis square;
subplot(1,2,2);imagesc(d);set(gca,'Visible','off');colormap gray; axis square;
drawnow
J = 0.5;
f = ones(M,N);
figure(3);
subplot(1,2,1); hf=imagesc(f);set(gca,'Visible','off');
colormap gray; axis square; drawnow;
mf = zeros(M,N);
subplot(1,2,2); hm=imagesc(mf);set(gca,'Visible','off');
colormap gray; axis square; drawnow;
SS = 10000;
misfit = [];
adj = [-1 1 0 0; 0 0 -1 1];
iter = 0;
while 1
  ix = ceil(N*rand(1)); iy = ceil(M*rand(1));
  pos = iy + M*(ix-1); fp = -f(pos);
  LkdRat = exp(d(pos)*(fp - f(pos))/sigma.^2);
  alpha = LkdRat;
  if rand<alpha % Prob of acceptance = min(1,alpha)
    f(pos) = fp;
  end
  iter = iter + 1;
  if rem(iter,SS) == 0,
     mf = mf+f; NS = iter/SS; iter
     set(hf,'CData',f);
     set(hm,'CData',mf); drawnow
     if iter/SS > length(misfit)
        misfit = [misfit,zeros(100,1)];
        misfit(iter/SS) = sum(sum((d-f).^2))/sigma;
     end
  end
end
```

Figure 7.2: Left hand figure shows a single sample from the MH MCMC algorithm, using a uniform prior and right hand figure shows the mean of 600 sweeps, each consisting of 10000 iterations of the algorithm.

From the reconstruction, it is clear that the uniform prior does not significantly reduce the noise, since the noise amplitude is so large that there is a significant probability that a given measured value arose from either a $+1$ or -1. For a uniform prior, the state of each pixel is independent of every other, so no inter-pixel correlations are used to improve the reconstruction.

7.2.2 Markov Radom Field with Ising Prior

As discussed in the previous chapter, an Ising prior on the space of binary images is one of the form

$$\Pr(\mathbf{f}) = \frac{1}{\mathcal{Z}_J} \exp(-2J\#\mathbf{f}) \tag{7.9}$$

where $\#\mathbf{f}$ is the number of edges linking disagreeing pixels in $\mathbf{f}$, $J > 0$ is a constant and $\mathcal{Z}_J$ is the normalization constant. In the MH MCMC algorithm, when comparing the current state $\mathbf{f}$ with a candidate state $\mathbf{f}'$ which differs from $\mathbf{f}$ by a single pixel, the ratio of posterior probabilities is given by

$$\frac{\Pr(\mathbf{f}'|\mathbf{d})}{\Pr(\mathbf{f}|\mathbf{d})} = \frac{\Pr(\mathbf{d}|\mathbf{f}')}{\Pr(\mathbf{d}|\mathbf{f})} \frac{\Pr(\mathbf{f}')}{\Pr(\mathbf{f})} = \exp\left[\frac{d_{kl}(f'_{kl} - f_{kl})}{\sigma^2}\right] \exp\left[2J\left(\#\mathbf{f} - \#\mathbf{f}'\right)\right]. \tag{7.10}$$

Notice how the normalization constant $\mathcal{Z}_J$ (which is difficult to evaluate) has cancelled in this ratio. The quantity $\#\mathbf{f} - \#\mathbf{f}'$ is easy to evaluate for a single-pixel change as it only involves examining at most four edges which link the pixel to its neighbours.

The Matlab code for this problem is essentially the same as for the uniform prior, except that the calculation of the acceptance probability is modified to the following:

```
LkdRat = exp(d(pos)*(fp - f(pos))/sigma.^2);
```

```
nbrs = pos + [-1,1,-M,M];  nbrs(find([iy==1,iy==M,ix==1,ix==N])) = [];
disagreef = sum(f(nbrs)~=f(pos));  disagreefp = sum(f(nbrs)~=fp);
DelLogPr = 2*J*(disagreef - disagreefp);
alpha = exp(DelLogPr)*LkdRat;
if rand<alpha
   f(pos) = fp;
end
```

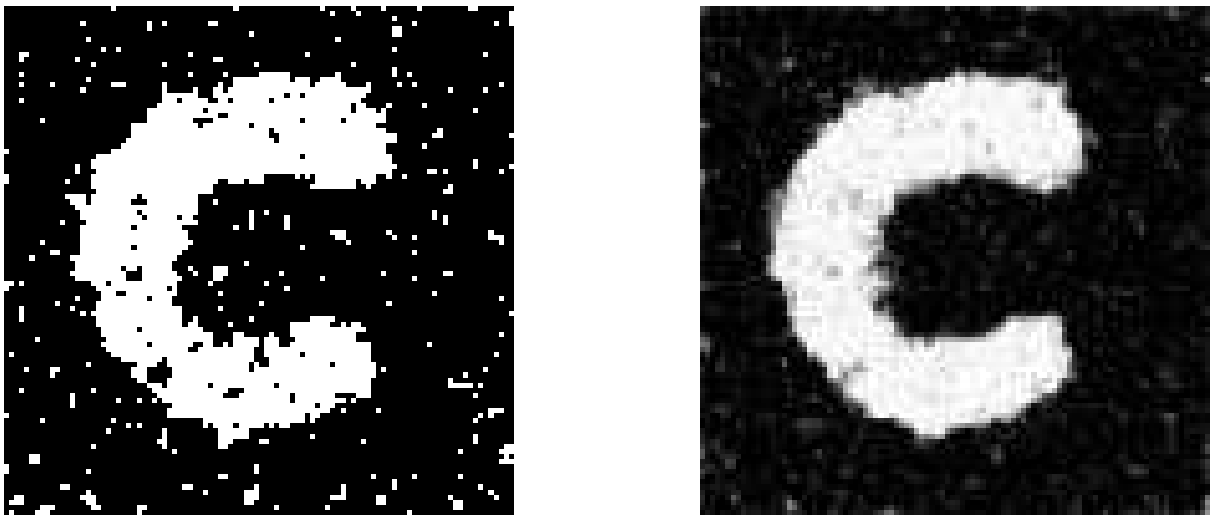


Figure 7.3: Left hand figure shows a single sample from the MH MCMC algorithm, using a Ising prior with $J = 0.5$ and right hand figure shows the mean of 600 sweeps, each consisting of 10000 iterations of the algorithm.

From the reconstruction shown in Figure 7.3, the clumping encouraged by the Ising prior is apparent. This example illustrates the power of using prior information to help constrain the reconstruction in image processing.

7.3 Recovering a binary matrix from its row and column sums

In this section we consider an artificial problem: the estimation of a matrix (or "image") of zeros and ones from its row and column sums, as observed in the presence of noise.

> An archaeologist is able to X-ray a square pillar from two sides only. We are given a single slice of this data - two 64-pixel long sequences of noisy intensity readings representing projections onto two orthogonal axes parallel to the sides of the pillar, at a fixed height up the pillar. Suppose that the intensity of the rays falls off from its initial value in proportion to the total amount of mass in the path of the x-rays incident a given pixel. The pillar is expected to be composed of two types of material of known density. The materials are expected to be separated out in lumps of unknown position.

The problem might be recast: give a Bayesian formulation for the problem of reconstructing an $N \times N$ matrix $\mathbf{f} = [f_{mn}]$ given that $f_{mn} \in \{-1, 1\}$ and given the row and column sums of $\mathbf{f}$. Each row and each column sum is an independent measurement, with an uncertainty which is assumed to be Gaussian, mean zero, and standard deviation σ.

Let the data be $\mathbf{d} = (\mathbf{r}, \mathbf{c})$ where $\mathbf{r}$ and $\mathbf{c}$ are N-component vectors. In terms of $\mathbf{f}$,

$$r_m = \sum_{n=1}^{N} f_{mn} + \epsilon_m^r, \tag{7.11}$$

$$c_n = \sum_{m=1}^{N} f_{mn} + \epsilon_n^c. \tag{7.12}$$

where ϵ_m^r and ϵ_n^c are normally distributed r.v. mean zero and standard deviation σ, i.e.,

$$\epsilon_m^r \sim \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\epsilon_m^r)^2/2\sigma^2} \qquad \epsilon_n^c \sim \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\epsilon_n^c)^2/2\sigma^2}, \text{ for } m, n = 1 \ldots N$$

Given an image $\mathbf{f}$, let $[\mathbf{f}]_m$ denote the sum over the m'th row of $\mathbf{f}$ and let $[\mathbf{f}]^n$ denote the sum over the n'th column of $\mathbf{f}$. The posterior probability is given by

$$\Pr(\mathbf{f}|\mathbf{d}) \propto \Pr(\mathbf{d}|\mathbf{f}) \Pr(\mathbf{f}) \tag{7.13}$$

The likelihood function is given by

$$\Pr(\mathbf{d}|\mathbf{f}) = \prod_{m=1}^{N} \Pr(\text{observe } r_m|[\mathbf{f}]_m) \prod_{n=1}^{N} \Pr(\text{observe } c_n|[\mathbf{f}]^n) \tag{7.14}$$

$$\propto \exp\left(-\frac{1}{2\sigma^2} \sum_{m=1}^{N} (r_m - [\mathbf{f}]_m)^2 - \frac{1}{2\sigma^2} \sum_{n=1}^{N} (c_n - [\mathbf{f}]^n)^2 \right). \tag{7.15}$$

If we use an Ising prior for $\mathbf{f}$, the posterior probability is

$$\Pr(\mathbf{f}|\mathbf{d}) = \frac{1}{\mathcal{Z}_{J,\sigma}} \exp\left(-2J\#\mathbf{f} - \frac{1}{2\sigma^2} \sum_{m=1}^{N} (r_m - [\mathbf{f}]_m)^2 - \frac{1}{2\sigma^2} \sum_{n=1}^{N} (c_n - [\mathbf{f}]^n)^2 \right) \tag{7.16}$$

where $\mathcal{Z}_{J,\sigma}^{-1}$ is again an unknown, intractable normalizing constant. We can sample from this posterior probability function using Markov chain Monte Carlo and the Metropolis-Hastings update rule as before. We aim to produce a Markov chain with equilibrium distribution $\pi_{\mathbf{f}} = \Pr(\mathbf{f}|\mathbf{d})$, $\mathbf{f} \in \Omega$. We can use much the same algorithm as above:

Let $X_n = \mathbf{f}$. X_{n+1} is determined in the following way:

1. Given $\mathbf{f} = (f_{1,1}, f_{1,2}, \ldots, f_{N,N})$, pick one of the N^2 pixels (m, n) at random.
   Set $\mathbf{f}' = (f_{1,1}, f_{1,2}, \ldots, -f_{m,n}, \ldots, f_{N,N})$. Our generation scheme is suitable for MH MCMC.

2. With probability

$$\begin{aligned}
\alpha(\mathbf{f}'|\mathbf{f}) &= \min\left\{1, \frac{\Pr(\mathbf{f}'|\mathbf{d}) g(\mathbf{f}|\mathbf{f}')}{\Pr(\mathbf{f}|\mathbf{d}) g(\mathbf{f}'|\mathbf{f})}\right\} \\
&= \min\left\{1, \exp\left(\begin{array}{c} -2J\left(\#\mathbf{f}' - \#\mathbf{f}\right) - \frac{1}{2\sigma^2}\sum_{m=1}^{N}\left[(r_m - [\mathbf{f}']_m)^2 - (r_m - [\mathbf{f}]_m)^2\right] \\ -\frac{1}{2\sigma^2}\sum_{n=1}^{N}\left[(c_n - [\mathbf{f}']^n)^2 - (c_n - [\mathbf{f}]^n)^2\right] \end{array} \right)\right\} \\
&= \min\left\{1, \exp\left(-2J\left(\#\mathbf{f}' - \#\mathbf{f}\right) - \frac{2f_{m,n}}{\sigma^2}\left(r_m + c_n - [\mathbf{f}]_m - [\mathbf{f}]^n + 2f_{m,n}\right)\right)\right\}
\end{aligned}$$

set $X_{n+1} = \mathbf{f}'$. Otherwise we set $X_{n+1} = \mathbf{f}$.

The above algorithm was implemented in *C*. The row and column sums of a matrix of side 64 were observed under i.i.d. Gaussian noise, mean zero, standard deviation 2.5. A sequence of states from MCMC sampling of the posterior for prior parameter $J = 0.5$ are shown in Figure 7.4.

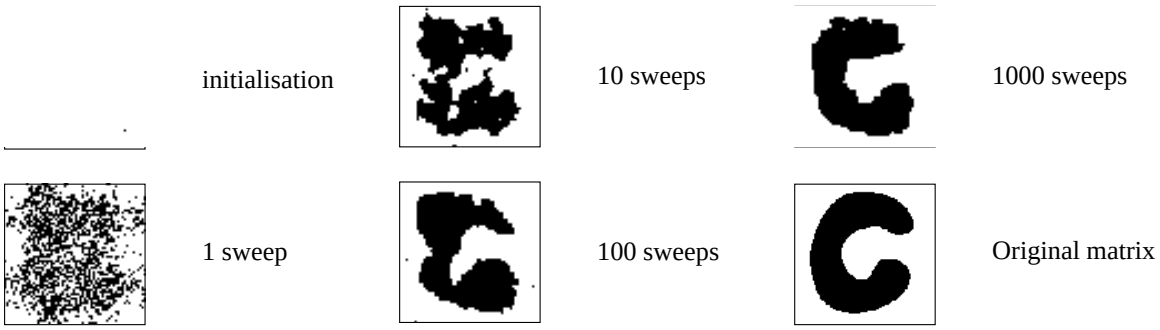


Figure 7.4: Sample from posterior for matrix reconstruction problem. 1 sweep = 4096 MC steps.

7.3.1 References

1. D. Geman and S. Geman (1984) "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images." IEEE Transactions on Pattern Analysis and Machine Intelligence, **6,** 721–741.

2. D.M. Grieg, B.T. Porteous and A.H. Seheult (1989) "Exact maximum à posteriori estimation for binary images", J. Royal Stat Soc B, **51,** 271–279.

7.4 Markov Chain Monte Carlo on Continuous State Spaces

Up to here, Ω has been discrete. Irreducibility is hard to define for continuous state spaces. We shall regard the continuous case as approximated by a distribution on a discrete space.

Problem: How to construct a Markov chain $\{X_n\}_{n=0}^{\infty}$ given an equilibrium probability density $q(x)$?

Definition: The kernel of a Markov chain on a continuous space Ω, corresponding to the transition matrix P_{ij} of the discrete chain is the density $p(x'|x)$ where

$$p\left(x'|x\right)\,\mathrm{d}x' = \Pr\left(x' \leq X_{n+1} < x' + dx'|X_n = x\right), \tag{7.17}$$

i.e., given that we are in state $X_n = x$ at update n, $p(x'|x)\,\mathrm{d}x'$ gives the probability to land in the interval $[x', x' + \mathrm{d}x')$ at the next update.

For correct normalization, we have

$$\int_{\Omega} p\left(x'|x\right)\,\mathrm{d}x' = 1 \tag{7.18}$$

since X_{n+1} must take **some** value. Also if $\pi^{(n)}(x)$ is the probability density, so that $\pi^{(n)}(x)\,\mathrm{d}x$ is the probability to be in the set $[x, x+\mathrm{d}x)$ after n steps, then

$$\pi^{(n+1)}\left(x'\right) = \int_{\Omega} \pi^{(n)}(x)\, p\left(x'|x\right)\,\mathrm{d}x \tag{7.19}$$

is the probability density at the next step. This should be compared with

$$\pi_j^{(n+1)} = \sum_i \pi_i^{(n)} P_{ij} \tag{7.20}$$

for discrete Ω.

Definition: A probability density $\pi(x)$ is stationary for $p(x'|x)$ if

$$\pi\left(x'\right) = \int_{\Omega} p\left(x'|x\right)\pi(x)\,\mathrm{d}x, \tag{7.21}$$

i.e., if the update preserves the distribution.

Assertion: The kernel $p(x \to dx')$ is reversible with respect to the distribution $\pi(x)$ iff

$$p\left(x'|x\right)\pi(x) = p\left(x|x'\right)\pi\left(x'\right) \tag{7.22}$$

Notice that if p is reversible with respect to π then π is stationary for p (integrate both sides $\mathrm{d}x$).

Irreducibility is harder to state.

Definition: $p(x'|x)$ is π-irreducible if for any set $A \subset \Omega$ with $\int_A \pi(x)\,\mathrm{d}x > 0$, $\Pr\left(X_n \in A \text{ for some finite } n|X\right.$ 0, so that the chain can hit any set that has finite probability in π.

Note that the last definition is not quite precise—see e.g., Tierney in "Markov Chain Monte Carlo in practice" eds. Gilks, Richardson and Spiegelhalter.

Theorem (ergodicity from reversibility)
Let $q(x)$ be a given probability density on Ω. If $p(x'|x)$ is q irreducible and if p is reversible and aperiodic with respect to q, then $\int_A \pi^{(n)}(x)\,dx \to \int_A \pi(x)\,dx$ as $n \to \infty$ for any set $A \subset \Omega$ and starting distribution $\pi^{(0)}$.

We can get reversibility using the Metropolis-Hastings prescription as before. Suppose we wish to generate a MC with unique probability density $q(x)$. Let $X_n = x$. X_{n+1} is determined in the following way:

1. Select x' with probability density $g(x'|x)$.

2. Accept x' (i.e., set $X_{n+1} = x'$) with probability

$$\alpha\left(x'|x\right) = \min\left\{1, \frac{q\left(x'\right)}{q(x)}\frac{g\left(x|x'\right)}{g\left(x'|x\right)}\right\}$$

   If x' is not accepted, then set $X^{n+1} = x$.

7.5 Estimating the parameters of a Gaussian Distribution

Suppose that we collect K samples $\mathbf{y} = \{y_1, \ldots, y_K\}$ from a normal distribution with mean μ and standard deviation σ. We wish to estimate μ and σ from the sample. This a somewhat contrived example in that there are only two parameters involved, and it is easy to visualize the posterior probability $p(\mu, \sigma|\mathbf{y})$ without using Monte-Carlo methods. Nevertheless, it is instructive to see how the problem may be solved via sampling from the posterior probability.

By Bayes' theorem,

$$p(\mu, \sigma|y_1, \ldots, y_K) \propto p(y_1, \ldots, y_K|\mu, \sigma)\, p(\mu, \sigma) \tag{7.23}$$

The likelihood function is determined by the forward problem. For a single sample from a normal distribution,

$$p(y_i|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y_i - \mu}{\sigma}\right)^2\right] \tag{7.24}$$

and so for K independent samples,

$$p(y_1, \ldots, y_K|\mu, \sigma) = \frac{1}{\sigma^K (2\pi)^K} \exp\left[-\frac{1}{2}\sum_{i=1}^{K}\left(\frac{y_i - \mu}{\sigma}\right)^2\right]. \tag{7.25}$$

Let us consider a prior probability which is the product of a non-informative prior for μ and a non-informative prior for σ. If we initially consider any value of μ as being equally likely as any other, we need to set $p(\mu)$ to be a constant. On the other hand, since σ measures the width of the distribution, it must be non-negative. A possible choice is to make the prior for σ "scale-invariant". This means that we initially think that $\Pr(a \leq \sigma < ka)$ to be independent of a. This is equivalent to making the probability density uniform as a function of $\log\sigma$, or $p(\sigma) \propto 1/\sigma$. Note that both of these priors are non-normalizable. This need not be a problem if the likelihood function is sharp enough, as the posterior probability density will be essentially independent of any limits we apply to the ranges of μ or σ in order to make the priors normalizable. We thus choose $p(\mu, \sigma) \propto 1/\sigma$.

Substituting into Bayes' theorem yields

$$p(\mu, \sigma|y_1, \ldots, y_K) \propto \frac{1}{\sigma^{K+1}} \exp\left[-\frac{1}{2}\sum_{i=1}^{K}\left(\frac{y_i - \mu}{\sigma}\right)^2\right] \tag{7.26}$$

Although this is not normalized, we already have enough information to use the MCMC method to sample from the posterior density.

Let $X_n = (\mu, \sigma)$. X_{n+1} is found as follows

1. Let r_1 and r_2 be drawn from a uniform distribution on $[0, 1]$. Let w_1 and w_2 be positive constants. Set

$$\mu' = \mu + w_1(2r_1 - 1) \tag{7.27}$$
$$\sigma' = \sigma + w_2(2r_2 - 1) \tag{7.28}$$

   This means that the proposal density function is

$$g(\mu', \sigma'|\mu, \sigma) = \begin{cases} \frac{1}{4w_1w_2} & \text{if } |\mu' - \mu| < w_1 \text{ and } |\sigma' - \sigma| < w_2 \\ 0 & \text{otherwise} \end{cases} \tag{7.29}$$

   Clearly $g(\mu', \sigma'|\mu, \sigma) = g(\mu, \sigma|\mu', \sigma')$.

2. With probability

$$\begin{aligned}\alpha\left(\mu',\sigma'|\mu,\sigma\right) &= \min\left\{1,\frac{p\left(\mu',\sigma'|y_1,\ldots,y_K\right)}{p\left(\mu,\sigma|y_1,\ldots,y_K\right)}\frac{g\left(\mu',\sigma'|\mu,\sigma\right)}{g\left(\mu,\sigma|\mu',\sigma'\right)}\right\} \\ &= \min\left\{1,\left(\frac{\sigma}{\sigma'}\right)^{K+1}\exp\left[-\frac{1}{2}\sum_{i=1}^{K}\left\{\left(\frac{y_i-\mu'}{\sigma'}\right)^2-\left(\frac{y_i-\mu}{\sigma}\right)^2\right\}\right]\right\} \quad (7.30)\end{aligned}$$

set $X_{n+1}=(\mu',\sigma')$, otherwise set $X_{n+1}=(\mu,\sigma)$. Note that if $\sigma'<0$ the posterior probability $p\left(\mu',\sigma'|y_1,\ldots,y_K\right)$ is zero (since $p\left(\mu',\sigma'\right)=0$) and we simply reject the move.

From the samples $\{X_n\}$ drawn from the posterior probability, we can plot histograms or calculate statistics of interest.

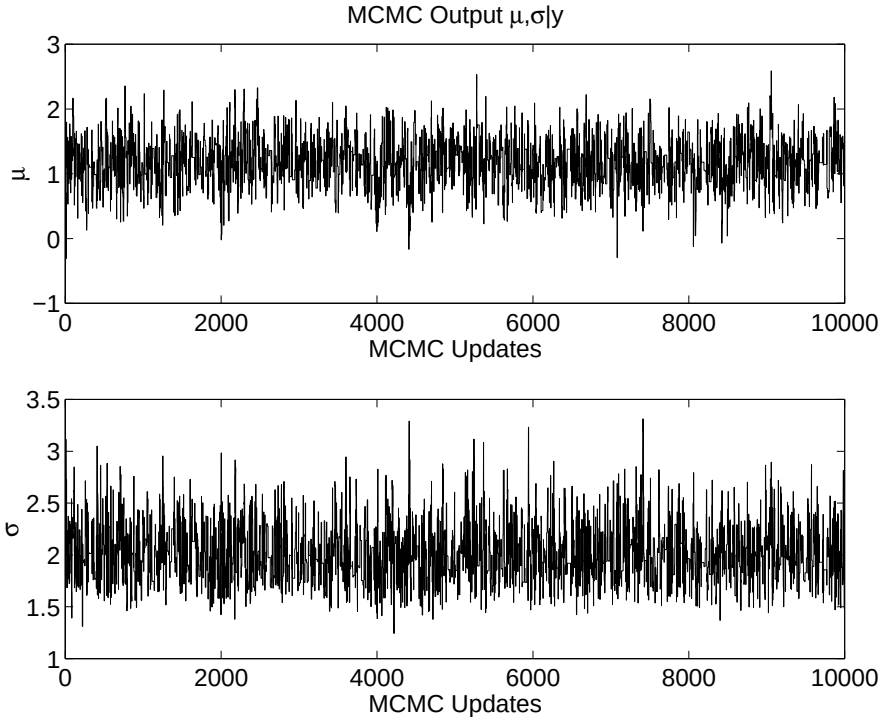


Figure 7.5: Samples from posterior probability distribution for the parameters of a Gaussian obtained from $K=30$ data points.

The Matlab code to carry out this simulation is:

```
K = 30;
mu = 1; sigma = 2;
y = mu + sigma*randn(K,1);

N = 10000;

m = 0; s = 1;
```

```
X = zeros(2,N);
w = 1;
for n = 1:N
mp = m + w*(2*rand-1);
sp = s + w*(2*rand-1);
 ratio = (s/sp)^(K+1)*exp(-0.5*sum(((y-mp)/sp).^2-((y-m)/s).^2));
if sp>0 & rand<ratio
 m = mp; s = sp;
end
X(:,n) = [m;s];
end
figure(1);
subplot(2,1,1); plot(X(1,:))
set(gca,'Fontsize',14); xlabel('MCMC Updates'); ylabel('\mu');
title('MCMC Output \mu,\sigma|y');
subplot(2,1,2); plot(X(2,:))
set(gca,'Fontsize',14); xlabel('MCMC Updates'); ylabel('\sigma');
figure(2);
subplot(1,2,1); hist(X(1,:),30)
set(gca,'Fontsize',14); xlabel('\mu'); ylabel('Frequency');
subplot(1,2,2); hist(X(2,:),30)
set(gca,'Fontsize',14); xlabel('\sigma'); ylabel('Frequency');
```

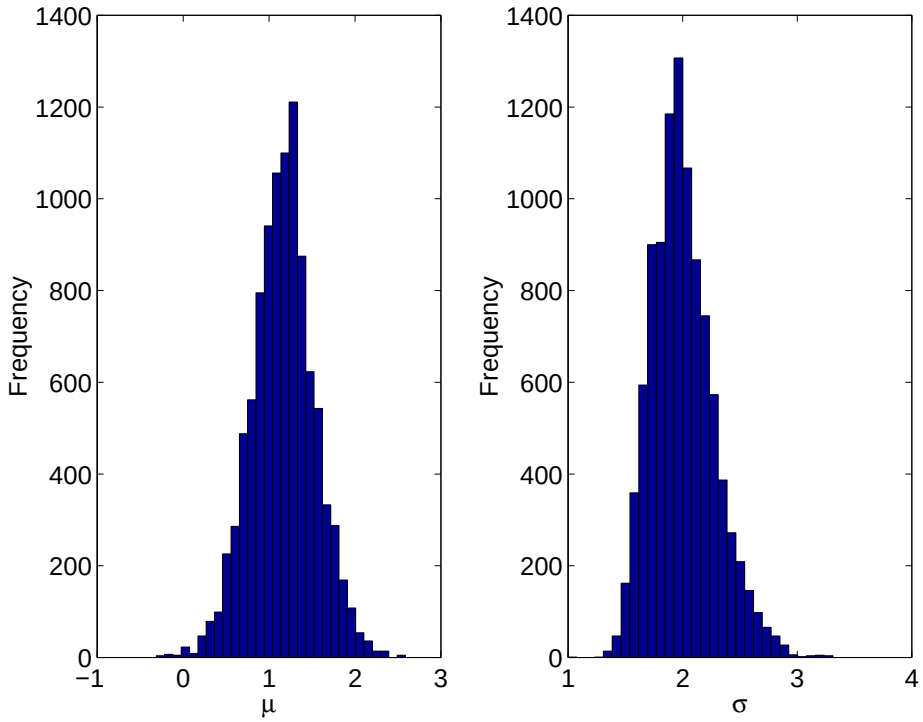


Figure 7.6: Marginal posterior histograms for mean and standard deviation of a Gaussian.

7.6 Estimating diffusivity D from solution of a partial differential equation

Suppose that $u \equiv u(x,t)$ is the number density of some animal diffusing in the interval $[0,L]$. The animal is killed if it leaves $[0,L]$ so that $u(0,t) = u(L,t) = 0$. Local conservation of number of animals gives the diffusion equation

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2}. \tag{7.31}$$

At time $t = 0$, the population density is

$$u(x,0) = \begin{cases} 1 & \text{for } 0.75L \le x \le 0.8L \\ 0 & \text{otherwise} \end{cases} \tag{7.32}$$

At time $t = T$, the population density is measured at points $x_1, x_2, \ldots, x_K$. Since the measurements are inexact, the data vector is $\mathbf{y} = (y_i)$ where

$$y_i = u(x_i, T) + \varepsilon_i \tag{7.33}$$

and ε_i are normally distributed with standard deviation s. From these measurements, we wish to sample from the posterior distribution of D, assuming that the prior distribution of D is uniform on $D \ge 0$.

By Bayes' theorem,

$$p(D|\mathbf{y}) \propto p(\mathbf{y}|D)\, p(D) \tag{7.34}$$

The likelihood function is determined by the noise process. Given the diffusivity is D and given the initial conditions, we may solve the partial differential equation and obtain the expected number density at the locations x_i after a time T, namely $u(x_i, T; D)$. The probability that we measure the data vector $\mathbf{y}$ is

$$p(\mathbf{y}|D) = \prod_{i=1}^{K} p(\varepsilon_i = y_i - u(x_i, T; D)) \tag{7.35}$$

$$\propto \exp\left[-\frac{1}{2}\sum_{i=1}^{K}\left(\frac{y_i - u(x_i,T;D)}{s}\right)^2\right] \tag{7.36}$$

where we have absorbed into the proportionality quantities which do not depend on D.

For the prior, we may choose

$$p(D) \propto \begin{cases} 1 & \text{if } D \ge 0 \\ 0 & \text{otherwise} \end{cases} \tag{7.37}$$

As before, this is improper (i.e., not normalizable), but could be made proper without affecting the analysis, for all practical purposes by imposing a conservative upper bound $D_{\max}$ on D.

The posterior probability density is thus given by

$$p(D|\mathbf{y}) \propto \begin{cases} \exp\left[-\frac{1}{2}\sum_{i=1}^{K}\left(\frac{y_i - u(x_i,T;D)}{s}\right)^2\right] & \text{if } D \ge 0 \\ 0 & \text{otherwise} \end{cases} \tag{7.38}$$

We can estimate D given y using sample-based inference.

Let $X_n = D$, X_{n+1} is given in the following way:

1. Let w be a positive constant. Draw r from a uniform distribution on $[0,1]$ and set $D' = D + w\,(2r-1)$

2. With probability

$$\alpha\left(D'|D\right) = \min\left\{1, \frac{\exp\left[-\frac{1}{2}\sum_{i=1}^{K}\left(\frac{y_i - u(x_i,T;D')}{s}\right)^2\right]}{\exp\left[-\frac{1}{2}\sum_{i=1}^{K}\left(\frac{y_i - u(x_i,T;D)}{s}\right)^2\right]}\right\}$$

set $X_{n+1} = D'$, otherwise set $X_{n+1} = D$.

Notice that at each step of the MCMC algorithm, we must compute $u\,(x_i, T; D)$, (i.e., solve the boundary value problem for a trial value of D) to work out what the solution would have looked like at T if the true diffusivity were given.

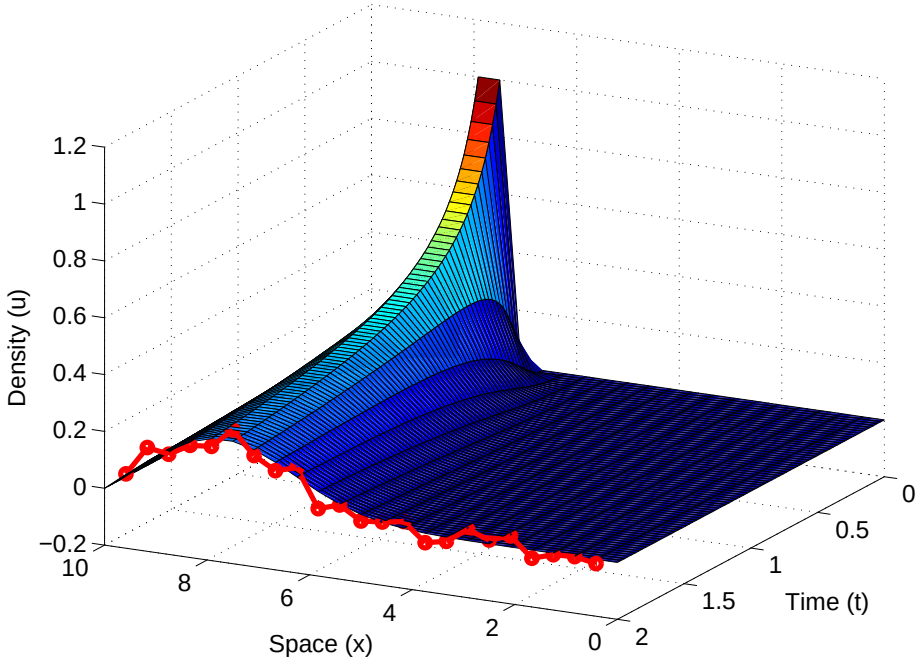


Figure 7.7: Surface is $u\,(x,t)$ with the true value of D and the points show the data measured at T.

The Matlab code to carry out this simulation is:

```
L = 10;
D = 0.5;
s = 0.03;
Tmax = 2;
xdim = 25; tdim = 75;
x = linspace(0,L,xdim);
t = linspace(0,Tmax,tdim);
```

```
dx = x(2)-x(1); dt = t(2)-t(1);
q = dt/dx^2;
r1 = 0.75*L; r2 = 0.8*L;
u0 = zeros(1,xdim);
u0(find(x>=r1 & x<=r2)) = 1;
xDat = 2:xdim-1;
tDat = tdim;
nxDat = length(xDat);
ntDat = length(tDat);
Z = heat(D,u0,q,tdim);
u = Z(tDat,xDat);
uDat = u + s*randn(ntDat,nxDat);
figure(1); surf(x,t,Z); hold on;
if ntDat>1, mesh(x(xDat),t(tDat),uDat);
else set(plot3(x(xDat),t(tDat)*ones(1,nxDat),uDat,'r-o'),'LineWidth',3);
end; hold off; drawnow
N = 10000; m = 100; XD = 1; X = zeros(1,N); X(1) = XD;
Z = heat(XD,u0,q,tdim);
u = Z(tDat,xDat);
oLLkd = sum(sum(-(u-uDat).^2))/(2*s^2);
LL = zeros(1,N); LL(1) = oLLkd;
w = 0.1;
for n = 2:N
 XDp = XD + w*(2*rand-1);
 if XDp > 0
   Z = heat(XDp,u0,q,tdim);
   u = Z(tDat,xDat);
   nLLkd = sum(sum(-(u-uDat).^2))/(2*s^2);
   alpha = exp(nLLkd - oLLkd);
   if rand < alpha
     XD = XDp;
     oLLkd = nLLkd;
     CZ = Z;
   end
 end
 X(n) = XD; LL(n) = oLLkd;
 if rem(n,m)==0
   figure(2); plot(X(1:n)); drawnow;
   figure(3); surf(x,t,CZ); hold on;
   if ntDat>1, mesh(x(xDat),t(tDat),uDat);
   else set(plot3(x(xDat),t(tDat)*ones(1,nxDat),uDat,'r-o'),'Linewidth',3);
   end; hold off; drawnow
   disp([N/m,n/m]);
 end
end
figure(2); plot(X);
```

The function for solving the diffusion equation is

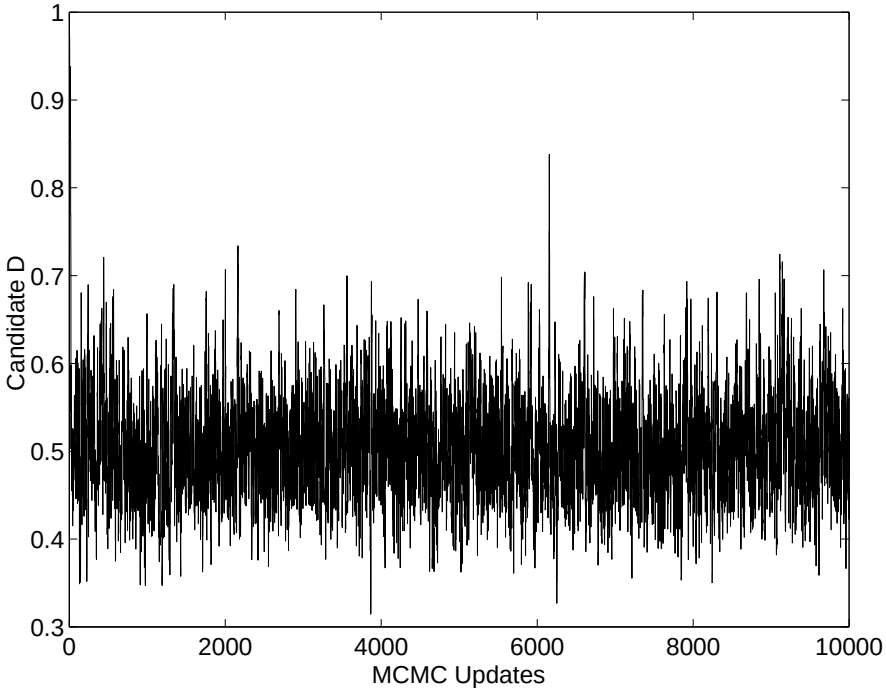


Figure 7.8: Output of Markov chain producing samples from posterior distribution of the diffusivity. True D value was 0.5.

```
function Z = heat(D,u0,q,tdim)
xdim = length(u0);
Z = zeros(tdim,xdim);
Z(1,:) = u0;
for tin = 2:tdim
tip = tin - 1;
Z(tin,2:end-1) = Z(tip,2:end-1) + ...
  D*q*(Z(tip,1:end-2)-2*Z(tip,2:end-1)+Z(tip,3:end));
end
```

7.7 Optimization using Markov chain Monte Carlo

The Markov chain Monte Carlo method may be used to find the mode of a distribution, which is an optimization problem.

7.7.1 Simulated Annealing

The mode of a distribution is the state of highest probability. In Bayesian analysis, the mode is the state of "maximum à posteriori probability" (the MAP state) and is often presented as the final "answer" in the analysis. This is generally unsatisfactory as it tells us nothing about the degree of uncertainty which our limited data admits to the reconstructed system state. Nevertheless, we are often interested in finding the mode of a function when the state space is large and the distribution is a complicated function of the state.

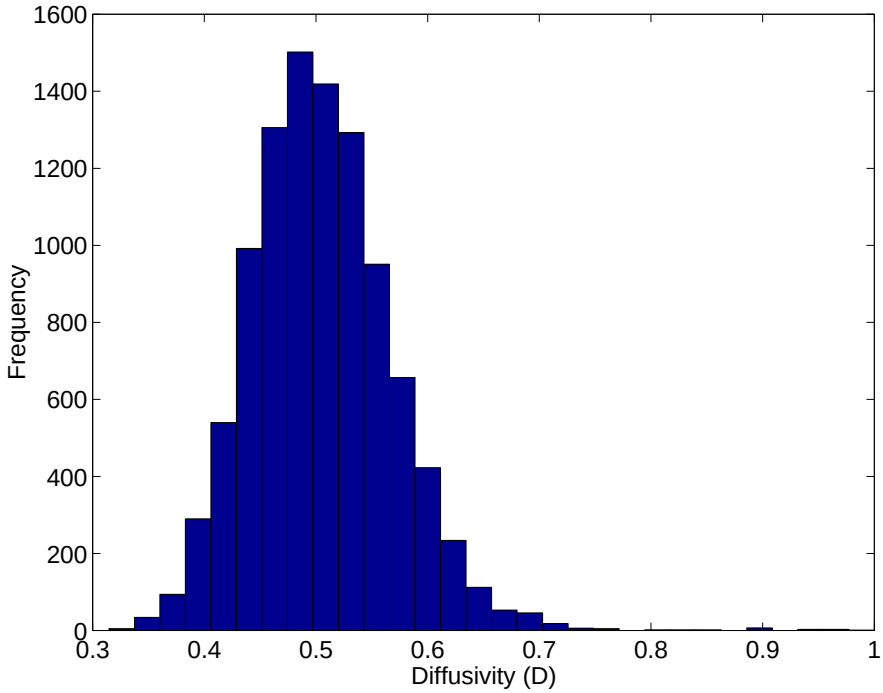


Figure 7.9: Posterior distribution of diffusivity. True D value was 0.5.

Let $Q(x)$ be a given probability density on a space Ω. Let $Q_{\max}$ be the maximum value of Q in Ω. Let $\Gamma = \{x : Q(x) = Q_{\max}, x \in \Omega\}$ be the mode set or the set of modal states, (i.e., there may be more than one). We write

$$Q(x) = \exp[-q(x)] \tag{7.39}$$

where $q(x) = -\ln Q(x)$ and introduce a "temperature" parameter T such that

$$Q_T(x) = \frac{\exp[-q(x)/T]}{\mathcal{Z}_T} \tag{7.40}$$

where $\mathcal{Z}_T$ is an unknown normalizing constant.

Consider two states x and $x' \in \Omega$ such that x is a mode (i.e., $x \in \Gamma$) but x' is not modal (i.e., $x' \notin \Gamma$). Since $Q(x') < Q(x)$, $q(x') > q(x)$ and so the ratio

$$\frac{Q_T(x')}{Q_T(x)} = \exp\left[\frac{q(x) - q(x')}{T}\right] \tag{7.41}$$

tends to zero as $T \to 0$. Hence as $T \to 0$, all probability mass in distribution Q_T is concentrated on states in Γ: we can find modal states (or state) by sampling $Q_T(x)$ at small T, using MCMC.

Roughly, the idea is to run a MCMC algorithm with equilibrium distribution $\pi_T = Q_T$. If the chain is in equilibrium at temperature T (ie $\pi^{(n)} = Q_T$) at step n and we lower the temperature to $T' < T$, we must then wait for the chain to equilibrate at the new temperature. We lower the temperature slowly, so that the chain remains in or close to equilibrium with

Q_T at each new temperature. At small T $\pi^T \sim 1/|\Gamma|$ and our MCMC returns a sample uniform on the mode states.

Suppose an ergodic Metropolis-Hastings algorithm with generation distribution $g(x'|x)$ is given. The algorithm simulates some Markov chain with states $x \in \Omega$, initializing distribution $X_0 \sim \Pr\{X_0 = x\}$ and equilibrium distribution $Q(x) = \exp(-q(x))/\mathcal{Z}$. Specify $T(n)$, a decreasing function of n called the **cooling** or **annealing schedule.** The following **Simulated Annealing Algorithm** simulates an inhomogeneous Markov Chain.

Let $X_n = x$. X_{n+1} is determined in the following way.
1. Generate candidate state x' from $g(x'|x)$
2. With probability

$$\alpha = \min\left\{1, \exp\left[-\left(q\left(x'\right) - q\left(x\right)\right)/T\left(n\right)\right]\frac{g\left(x|x'\right)}{g\left(x'|x\right)}\right\}$$

set $X_{n+1} = x'$, otherwise, set $X_{n+1} = x$.
Definition 7.1 If $\pi^{(n)}$ becomes uniform on Γ as $n \to \infty$, *ie* if $\Pr\left(x^{(n)} \in \Gamma\right) \to 1$ as $n \to \infty$, we say that the annealing "converges".
Definition 7.2 Let $Q(x)$, $x \in \Omega$ be a given distribution with mode-set Γ. A state $x \in \Omega$ communicates with Γ at height h if there is a path from x into Γ with the property that the largest value of $q = -\ln Q$ along the path is $q(x) + h$.

Theorem 7.1 Annealing theorem (Hajek, 1988)

Let $Q(x)$, $x \in \Omega$ be a given distribution with mode-set Γ. Let d^* be the smallest number such that every $x \in \Omega$ communicates with Γ at height d^*. The simulated annealing algorithm converges if and only if

$$\sum_{n=1}^{\infty} \exp\left[-d^*/T\left(n\right)\right] = \infty$$

Corollary 7.1 If $T(n) = d/\log(n)$ then simulated annealing converges iff $d \geq d^*$.

Simulated annealing is an optimization algorithm. If $q(x)$ is a cost function on the states $x \in \Omega$, simulated annealing can be used to minimize $q(x)$. Refer to the "travelling salesperson" problem.

Although the simulated annealing algorithm with schedule $T(n) = d/\log(n)$ is guaranteed to converge, it does so too slowly to be of practical value. Also, d^* is hard to estimate. More rapid cooling is usually done. The output is checked by repeating the experiment with a varying random number seed, and making sure that we get the same answer each time.

There is a physical analogy. The process simulates the physical process of "annealing", i.e., slowly cooling, a material to get special, highly ordered crystalline states.
Example 7.1 Using simulated annealing, find the maximum à posteriori state of the matrix reconstruction problem of section 7.3 with the binary Markov random field prior.

Q is our posterior $Q(x) = \Pr(D = x|g, \sigma)$. We revise the algorithm of section **??**, introducing a factor $1/T(n)$ in the power of the exponential. We take as an annealing the schedule the unreliable geometric schedule $T(n) = r^n T_0$ with $0 < r < 1$ a real constant and T_0 an initial, top temperature. In the notation of Section **??**, the algorithm is as follows.

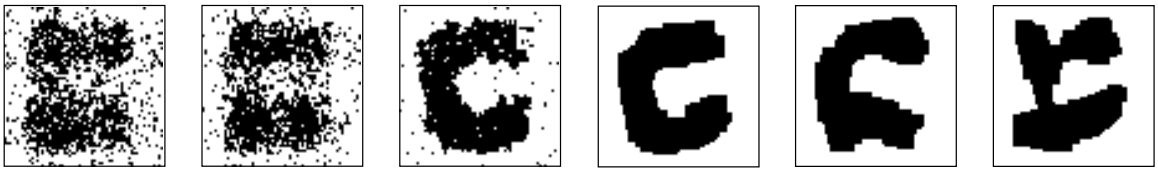
Let $X_n = x$. X_{n+1} is determined in the following way.
1. Given $x = (x_{1,1}, x_{1,2}, ..., x_{N,N})$, pick one of the N^2 pixels (m, n) at random.

Set $x' = (x_{1,1}, x_{1,2}, ... - x_{m,n}, ..., x_{N,N})$.
2. With probability

$$\alpha\left(x'|x\right) = \min\left\{1, \exp\left(-2(J\#\Delta_{mn} + \frac{x_{mn}}{\sigma^2}\left(r_m + c_n - [x]_m - [x]^n + 2x_{mn}\right))/T(n)\right)\right\}$$

set $X_{n+1} = x'$. Otherwise we set $X_{n+1} = x$.

Sample output is shown in Figure 7.10. If we compute the ratio of the posterior probabilities



(a1) T=4.4 (a2) 2.9 (a3) 1.9 (a1) T=0.2 (b) (c)

Figure 7.10: Estimating the mode of the posterior for the matrix estimation problem of Section 7.3. Sequence (a1-4) annealing with $r = 0.99999995$ and $T_0 = 10$. (b) Same cooling schedule, different random number seed. (c) annealing with $r = 0.999999$ and $T_0 = 2$.

for states 7.10(a4) and (b), $\Pr\left(D = (\text{a4})|g, \sigma\right) / \Pr\left(D = (\text{b})|g, \sigma\right)$ we obtain a value around unity. However states 7.10(a4) and (c) have a posterior ratio of around $\exp(-655)/\exp(-590) \sim 10^{-28}$. If the cooling schedule descends too rapidly, the MCMC becomes stuck in the unrepresentative, sub-optimal state Figure 7.10(c).

8

Output Analysis

8.1 Introduction

Let $X_0 = x^{(0)}, X_1 = x^{(1)} \ldots X_N = x^{(N)}$ be a realization of a homogeneous and reversible Markov chain, output by some MCMC algorithm which was set up to sample a distribution $Q(x)$ on states $x \in \Omega$. For example $\{x^{(n)}\}_{n=0}^N$ might be output from one of the following MCMC algorithm's:

1. the algorithm sampling the posterior for the matrix reconstruction problem from row and column sums,

2. the algorithm sampling the normal distribution.

Suppose $X \sim Q(.)$ is a random variable taking values in Ω. A function $f(X)$ is called a statistic. We look to the output for answers to questions like "What is the expected value of $f(X)$" — i.e., give a value for

$$\langle f(X) \rangle \equiv \sum_{x \in \Omega} f(x) \Pr(x).$$

We can estimate this from the output sample set $\{x^{(n)}\}_{n=0}^N$. The quantity

$$\bar{f}_N(\{x^{(n)}\}_{n=0}^N) \equiv \frac{1}{N} \sum_{n=1}^{N} f(x^{(n)})$$

is called an estimator of $\langle f(X) \rangle$. Since the samples $\{x^{(n)}\}_{n=0}^N$ are realizations of random variables $\{X_n\}_{n=0}^N$, our estimate, $\bar{f}_N(\{X_n\}_{n=0}^N)$, is itself a random variable. We will get a different estimate value at each "trial", that is, each time we gather a realization $\{x^{(n)}\}_{n=0}^N$ of the r.v. $\{X_n\}_{n=0}^N$.

Example 8.1 If the $\left\{x^{(n)}\right\}_{n=0}^N$ were sampled independently from a normal distribution mean μ, variance σ^2, $f(x) = x$ and $\bar{x} = \sum_n x^{(n)}/N$ is an estimator for μ. In this case $\bar{x}$ is a normally distributed random variable with mean μ and variance σ^2/N.

Example 8.2 Using the output in case (2) above, find a statistic giving the probability for the variable at pixel (j, k) to take the value $x_{jk} = -1$.

Answer If $\mathcal{A}$ and not-$\mathcal{A}$ are two possible outcomes of a trial then

$$\Pr\left(\text{event } \mathcal{A}\right) \simeq \frac{\#\text{times } \mathcal{A} \text{ occured}}{\text{number of trials}}$$

estimates the probability for the event $\mathcal{A}$. Let $f(x) = I[x_{jk} = -1]$, i.e., $f(x)$ is the indicator function, $I[\texttt{assertion}] = 0/1$ if `assertion` is true/false. Now

$$\bar{f}_N = \frac{1}{N}\sum_{n=1}^{N} I(x_{jk}^{(n)} = -1) \tag{8.1}$$

$$= \frac{\#\text{times } x_{jk} = -1 \text{ in the sampled states } \left\{x^{(n)}\right\}_{n=0}^{N}}{\text{number of samples}} \tag{8.2}$$

so our statistic $f = I[x_{jk} = -1]$ gives an estimator $\bar{f}_N$ for $\Pr(x_{jk} = -1)$

How good an estimator for $\langle f\left(X\right)\rangle$ is f_N? There are two issues:

1. Systematic error due to initialization bias — i.e., does $\bar{f}_N$ estimate $\langle f\left(X\right)\rangle$ at all? We may prove that $\pi^{(n)}$ tends to some desired unique equilibrium distribution $Q\left(x\right)$, however we have not said how large n needs to be for $\pi^{(n)} \approx Q\left(x\right)$. Multi-modality in Q and bugs in our computer program can be a problem here.

2. Autocorrelation in equilibrium — i.e., given $\pi^{(n)} = Q\left(x\right)$, for some n, how accurate is $\bar{f}_N$ as an estimator for $\langle f\left(X\right)\rangle$? Samples $x^{(n)}$, $x^{(n+1)}$ are highly correlated. How many samples should we take to achieve a given level of accuracy?

8.2 Autocorrelation in equilibrium

Recall that the variance of a quantity is a measure of the variability of the quantity about its mean. Thus if $x \sim Q(x)$, and $\mu_f \equiv \langle f(X)\rangle$,

$$\text{var}(f) = \left\langle f(X)^2 - \mu_f^2 \right\rangle$$

measures the variance of f in samples x distributed like $Q(x)$. Consider the quantity $\bar{f}_N$. $\text{var}(\bar{f}_N)$ is a measure of the variability of our estimator for μ_f.

We can quantify this statement. It is known that, under quite general conditions which we will not detail (see e.g. Geyer), for N sufficiently large,

$$\bar{f}_N \sim \text{Normal}\left(\langle f\rangle, \text{var}(\bar{f}_N)\right) \tag{8.3}$$

i.e., when N is large, our estimate $\bar{f}_N$ is normally distributed with mean $\langle f\rangle$ and some variance $\text{var}(\bar{f}_N)$. Equation 8.3 is usually written as a theorem concerning the limit

$$\lim_{N\to\infty} \sqrt{N}(\bar{f}_N - \langle f\rangle) \xrightarrow{\mathcal{D}} \text{Normal}\left(0, c\right) \tag{8.4}$$

with c some positive constant, independent of N. The limit "$\xrightarrow{\mathcal{D}}$" means that the distribution of the random variable on the left tends to the distribution on the right, i.e., normal, mean zero. Equation 8.4 is a statement of a "central limit theorem" concerning the distribution of $\bar{f}_N$.

If $\{x^{(n)}\}_{n=0}^N$ were *independent* samples and $\bar{f}_N \equiv \frac{1}{N}\sum_{n=1}^N f(x^{(n)})$ then

$$\mathrm{var}(\bar{f}_N) = \mathrm{var}(f)/N \tag{8.5}$$

However $\{x^{(n)}\}_{n=0}^N$ is a sequence of *correlated* samples from a Markov chain. We will see that

$$\mathrm{var}(\bar{f}_N) = \frac{\tau_f \mathrm{var}(f)}{N} \tag{8.6}$$

where τ_f is a number characteristic of the transition matrix of the Markov chain used to generate the sequence $\{x^{(n)}\}_{n=0}^N$. It follows from Equation 8.6 that $c = \tau_f \mathrm{var}(f)$ independent of N, in Equation 8.4.

Interpretation: In Equation 8.5 the variance of $\bar{f}_N$ goes down like $1/N$, where N is the number of independent samples. In Equation 8.5 the variance of $\bar{f}_N$ goes down like τ_f/N. Hence τ_f is the number of correlated samples with the same variance-reducing power as one independent sample. The quantity τ_f is called the **integrated autocorrelation time** (IACT, in physics literature) or **autocovariance time** (statistics literature).

Let $\sigma_{\bar{f}_N} \equiv \sqrt{\mathrm{var}(\bar{f}_N)}$ denote the standard deviation of our estimate $\bar{f}_N$. We are assuming N is large enough that the central limit theorem we quoted has set in and $\bar{f}_N$ is normally distributed. We use all of this information when we put error-bars on $\bar{f}_N$: we report something like "we measured f and obtained a mean of $\bar{f}_N \pm 2\sigma_{\bar{f}_N}$ at 95% confidence".

For a given equilibrium distribution $Q(x)$, we would like to design a chain for which τ_f is as small as possible, so that we get accurate estimates (small $\sigma_{\bar{f}_N}$) without needing large sample sizes N.

8.3 Calculating the integrated autocorrelation time

The covariance $\mathrm{cov}(f, g)$ of two quantities f and g is a measure of their correlation. Let $\mathcal{M} = \{X_n\}_{n=0}^{N-1}$ be a sequence of N stationary random variables in some given homogeneous Markov chain with equilibrium distribution Q, i.e., $X_0 \sim Q$, so that all the r.v. in the sequence are distributed according to the equilibrium distribution. In terms of some statistic $f(X)$, let

$$C_{ff}(s) \equiv \mathrm{cov}\left(f(X_n), f(X_{n+s})\right) \tag{8.7}$$

$$\equiv \langle f(X_n) f(X_{n+s})\rangle - \mu_f^2 \tag{8.8}$$

be the autocovariance function (ACF) at lag s, i.e., $C_{ff}(s)$ is the covariance between the values taken by f for two r.v. X_n and X_{n+s} in the chain separated by s updates. Since the chain is homogeneous and its distribution is stationary, $C_{ff}(s)$ depends only on s and not on n. For a Markov chain, this gets larger the closer the two states are together. If we define a normalized autocovariance function, $\rho_{ff}(s)$ via

$$\begin{aligned}\rho_{ff}(s) &= C_{ff}(s)/C_{ff}(0) \\ &= C_{ff}(s)/\mathrm{var}(f)\end{aligned}$$

then $\rho_{ff}(0) = 1$ — so $f(X_n)$ is perfectly correlated with itself! We expect $\rho_{ff}(s) \to 0$ monotonically as $s \to \infty$ (see Figure 8.1). Note that some authors refer to C_{ff} as the "autocorrelation function" although we reserve this usage for the case where the means have

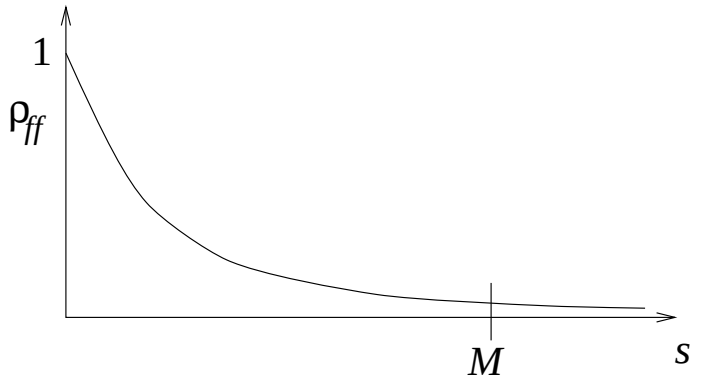


Figure 8.1: The normalised autocovariance function $\rho_{ff}(s)$.

not been subtracted. We will assume that for some M sufficiently large $\rho_{ff}(s) \simeq 0$ when $s \geq M$. We will assume also that $N \gg M$, so that the first $x^{(0)}$ and last $x^{(N)}$ samples are totally uncorrelated. We will show that under those assumptions $\mathrm{var}(\bar{f}_N) = \tau_f \mathrm{var}(f)/N$, as asserted in Equation 8.6.

$$\mathrm{var}\left(\bar{f}_N\right) = \left\langle \bar{f}_N^2 \right\rangle - \left\langle \bar{f}_N \right\rangle^2 \tag{8.9}$$

$$= \left\langle \left(\frac{1}{N} \sum_{n=1}^{N} f\left(X_n\right) \right) \left(\frac{1}{N} \sum_{m=1}^{N} f\left(X_m\right) \right) \right\rangle - \left\langle \frac{1}{N} \sum_{n=1}^{N} f\left(X_n\right) \right\rangle^2 \tag{8.10}$$

$$= \frac{1}{N^2} \sum_{n=1}^{N} \sum_{m=1}^{N} \left\langle f\left(X_m\right) f\left(X_n\right) \right\rangle - \left\langle f^2 \right\rangle \tag{8.11}$$

since $\langle f\left(X_n\right) \rangle = \langle f \rangle$, independent of n, if the chain is homogeneous and stationary. Now, if $M \ll N$,

$$\sum_{n=1}^{N} \sum_{m=1}^{N} \left\langle f\left(X_m\right) f\left(X_n\right) \right\rangle \approx \sum_{n=1}^{N} \left[\left\langle f\left(X_n\right) f\left(X_n\right) \right\rangle + 2 \sum_{s=1}^{N-M} \left\langle f\left(X_n\right) f\left(x^{(n+s)}\right) \right\rangle \right] \tag{8.12}$$

$$= \sum_{n=1}^{N} \left[C_{ff}(0) + \langle f \rangle^2 + 2 \sum_{s=1}^{N-M} C_{ff}(s) + \langle f \rangle^2 \right] \tag{8.13}$$

and so,

$$\mathrm{var}\left(\bar{f}_N\right) = \frac{1}{N^2} \sum_{n=1}^{N} \mathrm{var}(f) + 2 \sum_{s=1}^{N-M} C_{ff}(s) \tag{8.14}$$

$$\simeq \frac{\mathrm{var}(f)}{N^2} \sum_{n=1}^{N} \left[1 + 2 \sum_{s=1}^{M} \rho_{ff}(s) \right] \tag{8.15}$$

$$\simeq \frac{\mathrm{var}(f) \tau_f}{N} \tag{8.16}$$

as asserted in the previous section, with

$$\tau_f \equiv 1 + 2 \sum_{s=1}^{\infty} \rho_{ff}(s).$$

τ_f is the integral of the normalized autocovariance function $\rho_{ff}(s)$, because $\rho_{ff}(s) = \rho_{ff}(-s)$, and $\rho_{ff}(0) = 1$. To estimate τ_f from the output $\{x^{(n)}\}_{n=1}^{N}$ of an MCMC algorithm, we estimate $C_{ff}(s)$ by

$$\bar{C}_{ff}(s) = \frac{1}{N}\sum_{n=1}^{N} f\left(x^{(n)}\right) f\left(x^{(n+s)}\right) - \frac{1}{N^2}\left[\sum_{n=1}^{N} f\left(x^{(n)}\right)\right]^2$$

and then compute an estimate $\bar{\rho}_{ff}(s)$ for $\rho_{ff}(s)$ as above. A problem remains.

Since N is finite, the estimates $\bar{\rho}_{ff}(s)$ are noisy. We expect $\rho_{ff}(s) \simeq 0$ when $s \geq M$, so the signal goes to zero, and at large s, $s \geq M$ say, $\bar{\rho}_{ff}(s)$ is pure noise. If we were to form our estimate $\bar{\tau}_f$ for τ_f by summing over all s, including $s \geq M$, we would be adding noise and no signal to our estimate for τ_f. We must truncate the sum over $\bar{\rho}_{ff}(s)$ at $s = M$. We must decide from the data at what lag s noise begins to dominate. There are several approaches. The method outlined in Geyer is practical, as well as being theoretically well motivated. Well tested software exists to estimate all the quantities defined in this section. See for example code available through the R package.

Example 8.3 We will illustrate output analysis on the output of the MCMC algorithm given on page 7-23, for sampling the normal distribution with a mean $\mu = 3$ and standard deviation $\sigma = 1$.

A realization of length 10000 updates was obtained. The first 200 updates (see Figure 8.2)

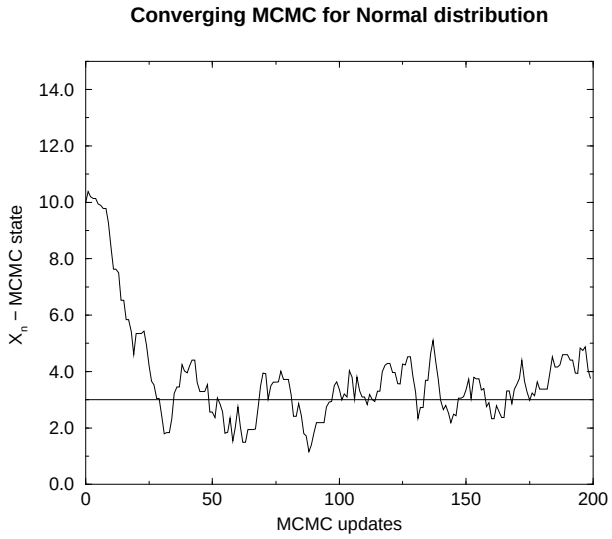


Figure 8.2: The converging sequence of MCMC updates.

were discarded, to allow time for $\pi^{(n)}$, the distribution of the Markov chain r.v. at step n, to converge to its equilibrium distribution, which, in this case, is the normal distribution. The full output is graphed in Figure 8.3. So we have a sample sequence $\{x^{(n)}\}_{n=200}^{n=10000}$ of length $N = 9800$. The mean of the $x^{(n)}$ was $\bar{x} \simeq 3.026$ (i.e., $f(x) = x$ above so that $\mu = \langle f \rangle$ and $\bar{x} \equiv \bar{f}_N$). The variance (i.e., $\sigma^2 = \mathrm{var}(x)$ above) was estimated at $\bar{\sigma}^2 \simeq 0.97$. In order to determine the variance of $\bar{x}$ the normalized ACF $\bar{\rho}_{xx}(s)$ was estimated from the same output data, as above. This estimated ACF is plotted in Figure 8.4. τ_x was calculated using the

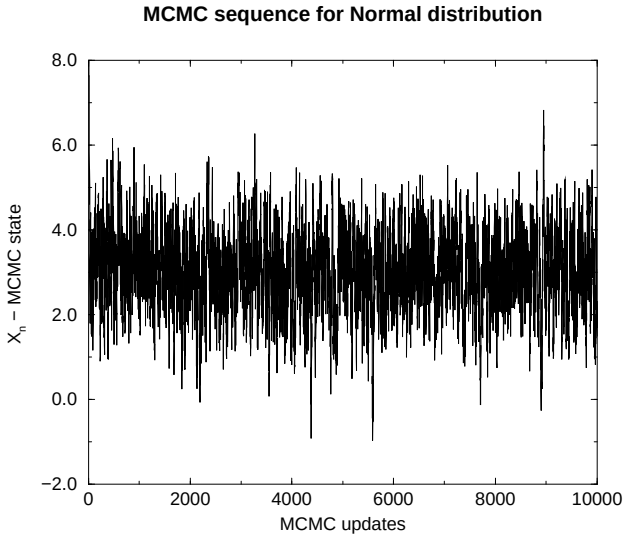


Figure 8.3: The sequence of 10000 MCMC updates output.

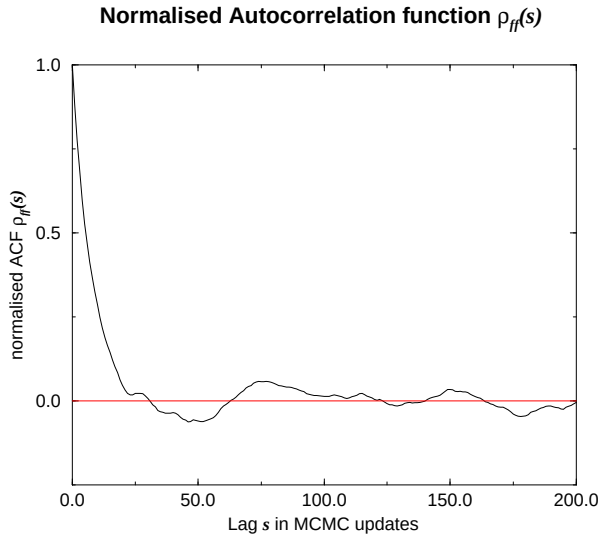


Figure 8.4: The normalised autocovariance function of the MCMC.

estimator,

$$\bar{\tau}_x = 1 + 2\sum_{s=1}^{M} \bar{\rho}_{xx}(s)$$

with $M = 23$ (see Figure 8.4). This gave an estimate of $\bar{\tau} \simeq 14.5$ for τ. This means that it takes around fifteen updates of the MCMC to give one effective independent sample. Now we estimate

$$\mathrm{var}(\bar{x}) = \frac{\bar{\sigma}^2 \bar{\tau}_x}{N} \simeq 0.0014$$

So we might reasonably report

$$\bar{x} = 3.03 \pm 0.08$$

at 95% confidence (where $0.08 \simeq 2\sqrt{0.0014}$).

Exercise 8.1 How could we check the result for $\mathrm{var}(\bar{x})$, using 10 independent runs of length 10000 ?

Answer Taking the output from each of the 10 runs we form 10 independent estimates $\bar{x}_1, \bar{x}_2 \ldots \bar{x}_{10}$. According to the CLT we quoted in Equation 8.4 these estimates should be scattered about μ with a normal distribution with variance equal around about $\mathrm{var}(\bar{x}) = 0.0014$.

Note: If its so easy to estimate the variance of $\bar{x}$ by repeating the run, why bother with the calculation of τ_f ? Because it is only by using all your data you get all the accuracy available. Also, plotting $\bar{\rho}_{ff}(s)$ is a sensible way to check for convergence.

In practice we are interested in getting $\mathrm{var}(\bar{f}_N)$ down as small as possible (i.e., getting accurate estimates). We can do this by taking more samples. We can also look at the efficiency of our algorithm. We are free to choose the candidate generation probability $g(x'|x)$, and this has a big effect on the rate at which correlations die out along the chain. Good algorithms have smaller IACT τ_f, since this leads to smaller values of $\mathrm{var}(\bar{f}_N)$ for given sample size N. Now τ_f is measured in MCMC updates. An update takes a certain amount of CPU time. So really we want τ_f small in CPU seconds. If some fancy algorithm had a small τ_f-value, but each update took a lot of time to compute, the accuracy of estimates would still only improve slowly. This is why simple algorithms (i.e., simple choices for the candidate generation probability $g(x'|x)$) are hard to beat.

Example 8.4 In the MCMC algorithm sampling the normal distribution, we generated candidate states using $G(dx'|x) = dx'/2\sigma$, i.e., the new state was chosen uniformly on the interval $[x-a, x+a]$ with $a = \sigma$. Can we reduce τ_f by using some other value of a ?

The core loop of our sampler (page 7-23) is essentially unchanged

```
.
.
a=input('enter jumpsize ');                    %enter ``a'' used in generation step
for k=1:N
   xp=x+(2*a*rand(1,1)-a);                     %generate candidate with jumpsize ``a''
   ratio=exp(-((xp-mu)^2-(x-mu)^2)/(2*sigma^2));
   if rand(1,1)<ratio
      x=xp;
```

```
   end;
   Xn(k)=x;
end;
.
.
```

This MCMC algorithm simulates a Markov chain with the normal distribution as its equilibrium distribution for a any non-zero real number. Suppose the mean $\mu = 3$ and standard deviation $\sigma = 1$ as before. For each of a sequence of a-values we run the MCMC and analyze the output, computing $\tau_x(a)$. In Figure 8.5, τ_x is plotted against a. The IACT is minimized

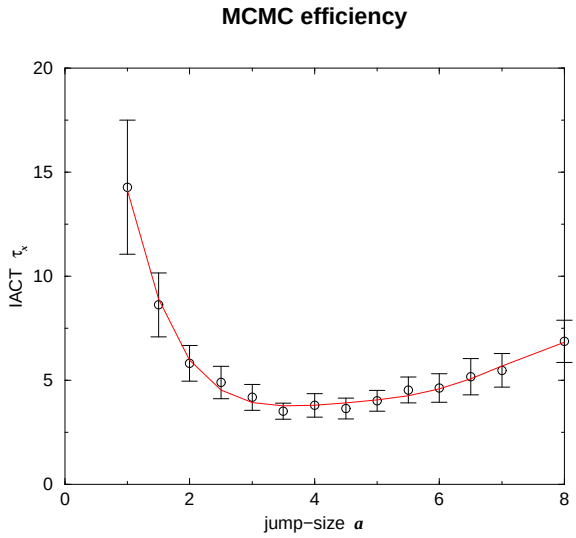


Figure 8.5: The normalised autocovariance function $\rho_{ff}(s)$.

by a value of jump size of around $a = 3.5$. This tells us that the chain outputs the greatest number of effectively independent samples per N updates when the jump size is around 3.5.

Interpretation Referring to Figure 8.6, when $a \ll \sigma$, the candidate state $x' \simeq x$, and in the acceptance ratio $\exp(-(x'-\mu)^2/2\sigma^2 + (x-\mu)^2/2\sigma^2) \simeq 1$, and hence $\alpha \simeq 1$. Most updates are accepted, but since the jump from x to x' is small the chain moves slowly through Ω and the r.v. X_n and X_{n+1} are highly correlated. On the other hand when $a \gg \sigma$, x' will often be selected well outside the interval $[\mu - 3\sigma, \mu + 3\sigma]$ where all the probability mass is concentrated. The ratio of the new and old probability densities will be small, and the acceptance probability α will be small. Although the chain makes large hops when it does move, there are too many rejections: X_n and X_{n+1} are highly correlated, because they are often equal !

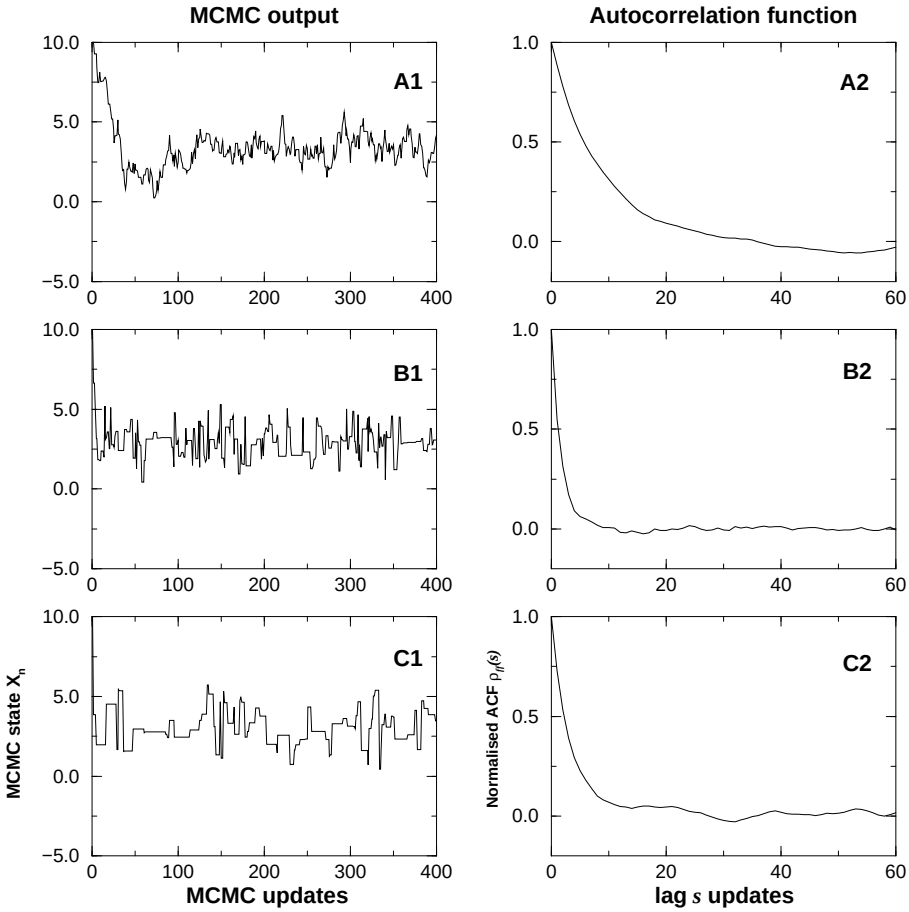


Figure 8.6: (A1) MCMC output for $a = 1$ (A2) normalised autocovariance function $\rho_{xx}(s)$ for $a = 1$. Although there is a high probability of acceptance, the sample path shows strong correlation from one update to the next. (B1) MCMC output for $a = 3.5$ (B2) $\rho_{xx}(s)$ for $a = 3.5$. The normalized autocovariance $\rho_{xx}(s)$ between X_n and X_{n+s} now falls off rapidly with increasing lag s. (C1) output for $a = 8$ The stepped appearance of the output reflects the low acceptance rate (C2) $\rho_{xx}(s)$ for $a = 8$, the correlation between states in the chain dies off slowly.

8.4 Initialization bias

Suppose we have an MCMC algorithm with intended equilibrium distribution $Q(x)$. We have seen (Figure 8.2) that the initial samples generated by MCMC are not representative of the equilibrium distribution. How long should we simulate before beginning to take samples? There is no general solution to the problem. A novel technique called "Exact simulation" offers a complete solution for certain MCMC algorithms and certain equilibrium distributions (Propp and Wilson 1995), including several of importance in statistical mechanics.

The usual approach is to monitor the behavior of some statistic (for example, $q(x^{(n)})$, where $Q(x) = \exp(-q(x))/\mathcal{Z}$) and drop samples from the part of the run where $q(x^{(n)})$ is converging to its equilibrium range. Referring to the MCMC output in Figure 8.7, our output sample would consist of $x^{(n)}$ for $n \geq k$ only. We are assuming that, for $n \geq k$ $X_n \sim Q$, i.e., $\pi^{(n)} = Q$

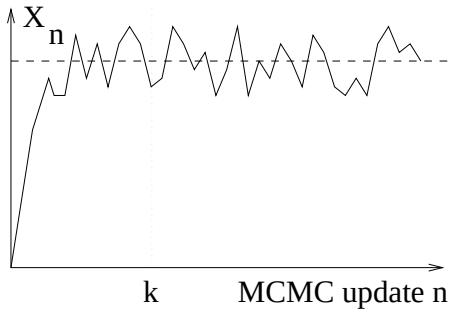


Figure 8.7: A MCMC algorithm started at an unrepresentative state converges to its equilibrium range.

for $n \geq k$. If the total run length $N \gg k$ this has a negligible effect on estimates, i.e.,

$$\bar{f}_{N-k} = \frac{1}{N-k} \sum_{n=k}^{N} f(x^{(n)})$$

when the first k samples are dropped.

8.5 Sticking and multimodality

We do not know when the chain has reached equilibrium. Considering Figure 8.8 it is clear

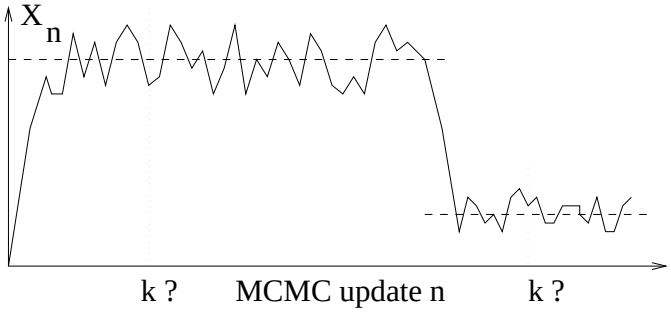


Figure 8.8: The chain may appear to be in equilibrium, when it is in fact in a metastable state.

that any judgement based on output may be premature. It may be that the generation probability $g(x'|x)$ can take the chain to any state in the state space Ω, but sections of Ω communicate only through states of very low acceptance probability. The chain may be stuck in the vicinity of one mode if $Q(x)$ is a multimodal distribution.

Example 8.5 Suppose $Q(X \in dx)$ is the sum of two normal distributions with means μ_1 and μ_2 ($\mu_2 > \mu_1$) and each having the same standard deviation σ. So, if $Q(X \in dx) = q(x)dx$, the density is

$$q(x) = \frac{1}{2\sqrt{2\pi\sigma^2}} \left(e^{-(x-\mu_1)/2\sigma^2} + e^{-(x-\mu_2)/2\sigma^2} \right).$$

We will suppose that the two normal distributions are "separated" in the sense that $\mu_2 - \mu_1 \gg \sigma$. See Figure 8.9. Now, for our MCMC algorithm with equilibrium $Q(x)$ we might choose

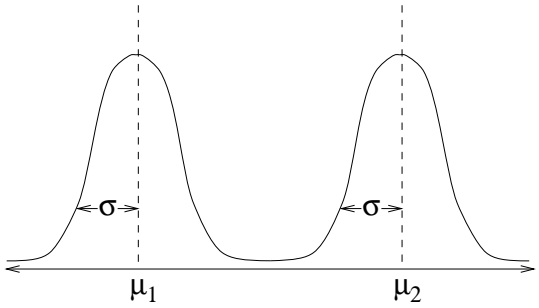


Figure 8.9: A distribution in which two regions of higher probability are separated by a region of low probability.

the following:

Let $X_n = x$. X_{n+1} is determined in the following way.
1. choose x' uniformly in an interval $[x - a, x + a]$ centered at x. Thus $x' \sim dx'/2a$ and the density $1/2a$ is (trivially) symmetric in x and x'.

2. With probability

$$\alpha = \min\left\{1,\ \frac{\exp\left(-(x'-\mu_1)/2\sigma^2\right) + \exp(-(x'-\mu_2)/2\sigma^2)}{\exp\left(-(x-\mu_1)/2\sigma^2\right) + \exp(-(x-\mu_2)/2\sigma^2)}\right\}$$

set $X_{n+1} = x'$. Otherwise we set $X_{n+1} = x$.

This algorithm has a straightforward Matlab implementation. The choices $\mu_1 = -4, \mu_2 = 4$ and $\sigma = 1$ give a wide region of low probability between lumps. Sample output for these parameter values and a range of jump sizes a is shown in Figure 8.10, along with autocovariance functions. Ergodicity is effectively lost for a less than around 4. The very slowly falling autocovariance function for $a = 4$, middle row right, indicates this. However at $a = 1$ the output is giving the illusion of an ergodicity that is not present - there is no sign of states at values $x = 4$ even though they hold half the probability mass. The MCMC is stuck in the mode around $x = -4$.

8.6 Good habits

We have seen that there is no guaranteed adequate output analysis. In particular, practical convergence cannot be demonstrated in general. The following procedure is however often sufficient.

1. Take a single long run, of length N, dropping states from the start of the run, and estimate τ_f, for $f(X)$ a statistic you care about.

2. Check that $N \gg \tau_f$. This is the only meaningful definition of a "long" run.

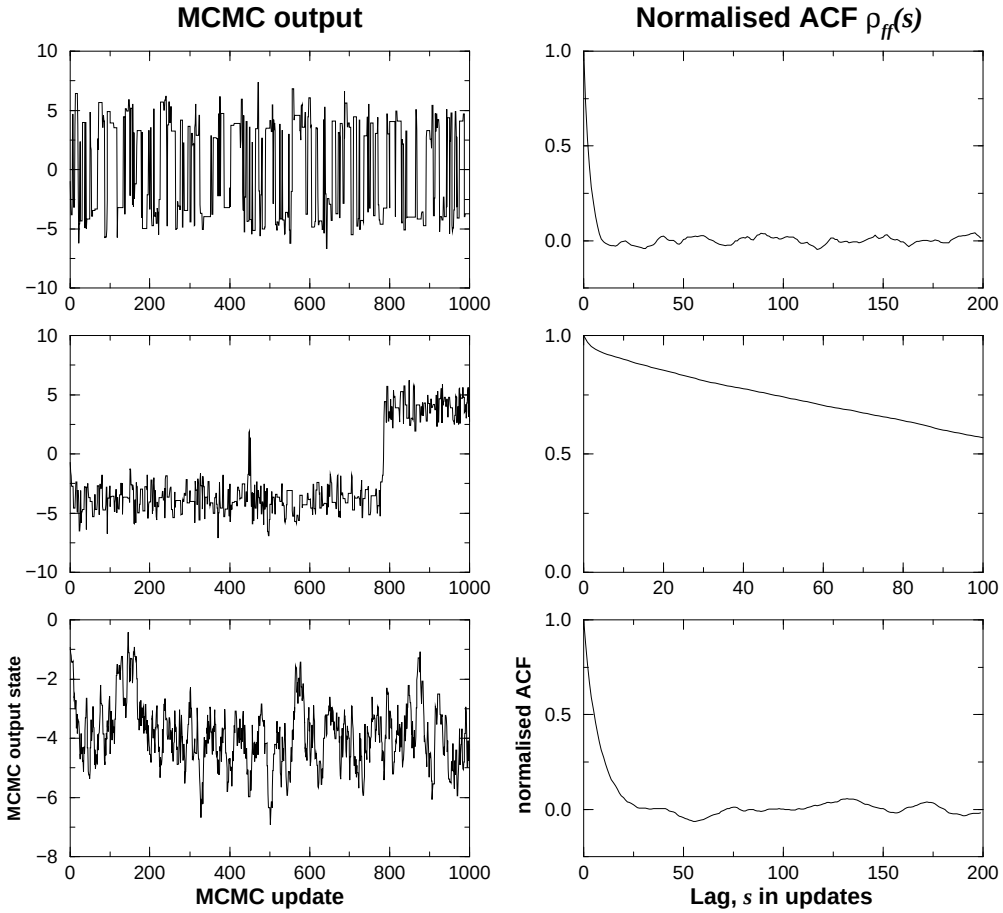


Figure 8.10: Illustration of initialisation bias and sticking: Output for MCMC sampling the multi-modal distribution of Figure 8.9. Top row, $a = 8$ (ergodic), middle row $a = 4$ (very slow mixing, strong initialisation bias), bottom row $a = 1$ (stuck).

3. Plot the output $x^{(n)}$, or $f\left(x^{(n)}\right)$ against n. Does it show any obvious trend ? Does it seem to be "in equilibrium" ?

4. Plot the normalized autocovariance function $\rho_{ff}(s)$ against lag s. It should fall off smoothly to zero, and then be distributed more or less evenly, with noise, about the x-axis.

5. Read Geyer "MCMC in practice", Statistical Science article, for some checks on the asymptotic variance of the autocorrelation function as $s \rightarrow$large.