NPL
National Physical Laboratory

Report

NPL Report DEM-ES 018

Software Support for Metrology Best Practice Guide No. 4

Discrete Modelling and Experimental Data Analysis

R M Barker, M G Cox, A B Forbes and P M Harris

March 2007

National Physical Laboratory | Hampton Road | Teddington | Middlesex | United Kingdom | TW11 0LW

Switchboard 020 8977 3222 | **NPL Helpline** 020 8943 6880 | **Fax** 020 8614 0446 | **www.npl.co.uk**

NPL Report DEM-ES 018

Software Support for Metrology
Best Practice Guide No. 4

Discrete Modelling and Experimental Data Analysis

R M Barker, M G Cox, A B Forbes and P M Harris
Mathematics and Scientific Computing Group

March 2007

ABSTRACT

Metrology, the science of measurement, involves the determination from experiment of estimates of the values of physical quantities, along with the associated uncertainties. In this endeavour, a mathematical model of the measurement system is required in order to extract information from the experimental data. Modelling involves *model building*: developing a mathematical model of the measurement system in terms of equations involving parameters that describe all the relevant aspects of the system, and *model solving*: determining estimates of the model parameters from the measured data by solving the equations constructed as part of the model.

This best-practice guide covers all the main stages in experimental data analysis: construction of candidate models, model parameterisation, uncertainty structure in the data, uncertainty of measurements, choice of parameter estimation algorithms and their implementation in software.

Version 3.0

NPL Report DEM-ES 018

ISSN 1754–2960

National Physical Laboratory,
Hampton Road, Teddington, Middlesex, United Kingdom TW11 0LW

We gratefully acknowledge the financial support of the UK Department for Innovation, Universities and Skills (National Measurement System Directorate)

Approved on behalf of the Managing Director, NPL by Jonathan Williams,
Knowledge Leader for the Optical Technology and Scientific Computing team

Contents

Chapter 1

Introduction

1.1 Mathematical modelling in metrology

Metrology, the science of measurement, involves the determination of quantitative estimates of physical quantities from experiment, along with the associated uncertainties. This process involves the following components:

Model building. Developing a mathematical model of the experimental system in terms of mathematical equations involving parameters that describe all the relevant aspects of the system. The model will need to specify how the system is expected to respond to input data and the nature of the uncertainties associated with the data.

Model solving. Determining estimates of the model parameters from the measured data by solving the mathematical equations constructed as part of the model. In general, this involves developing an algorithm that will determine the values for the parameters that best explain the data. These algorithms are often referred to as *estimators*. The estimation process also must evaluate uncertainties associated with the fitted parameters.

Software implementation of solution algorithms. Practically all calculations of fitted parameters are performed by software.

Model validation. Determining whether the results produced are consistent with the input data, theoretical results, reference data, etc. All stages need to be examined. Does the model adequately encapsulate what is known about the system? Does the method of solution produce unbiased estimates of the parameters and valid uncertainties? If information about the model is determined by software, then it is important that the software is valid to ensure that conclusions are based on reliable calculations.

Generally, these steps are revisited as the model is refined and the experimental design is evolved, resulting in a better explanation of the observed behaviour and more dependable uncertainties associated with the quantities of interest.

[0]This document: http://www.npl.co.uk/ssfm/download/bpg.html#ssfmbpg4

1.2 Scope and structure of this Guide

It is useful to classify the types of data arising in metrology into two categories: i) *discrete* and ii) *continuous*.

Example: the distribution of heat in a rectangular plate

Modelling discrete data. In a measurement experiment, the temperatures T_i are measured simultaneously at a fixed number m of locations (x_i, y_i) on a rectangular plate in a steady state. The data can be represented in a finite array whose ith row is (x_i, y_i, T_i). The temperature $t(x, y, \mathbf{a})$ is modelled as a function of location and model parameters $\mathbf{a}$. For example, $\mathbf{a}$ could be the coefficients of a bivariate polynomial surface. The data analysis problem is to find the values of the parameters $\mathbf{a}$ so that $t(x, y, \mathbf{a})$ best explains the data. For instance, a least-squares estimate of $\mathbf{a}$ is found by solving

$$\min_{\mathbf{a}} \sum_{i=1}^{m} \left(T_i - t(x_i, y_i, \mathbf{a})\right)^2 .$$

The measurement strategy is discrete in the sense that only a finite number of measurements are taken. The data analysis problem is discrete in the sense that the function to be minimised is a discrete sum based on algebraic equations. However, the model involves continuous phenomena: the temperature is modelled as a function t of location, even though the data representing the temperature are given at a finite number of points.

Modelling continuous data. Two adjacent edges of the plate are held at temperatures $g(x)$ and $h(y)$ where g and h are known functions defined at distances x and y along the edges. The data analysis problem is to determine the steady-state temperature $t(x, y)$ at each point on the plate, given the coefficient ν of heat conduction of the material. The data analysis problem will involve the solution of the heat equation, a partial differential equation, subject to the boundary conditions. The data is continuous in the sense that g and h are defined at each point along the edge, not at a finite number of points. In practice, these functions will be specified by a finite amount of information, for example, the coefficients of polynomial representations of the functions. The numerical solution will also involve a discretisation of the equations to be solved. ♯

This Guide is concerned with modelling discrete data and experimental data analysis. In chapters 2 and 3, we describe the main components of model building and model solving and are meant to give an overview of discrete modelling in metrology. Chapter 4 discusses the data analysis methods used in metrology, while chapter 5 is concerned with important empirical models used in metrology. These two chapters present tutorial material on estimation methods and model types.

The chapters on data analysis methods and model types have been further expanded in this revision of the Guide.

A summary of the main issues is given in section 6.

Revision history

Version 1.0 (March 2000). Initial publication.

Version 1.1 revision of this guide (January 2002). The main changes introduced in version 1.1 were:

- Correction of typographical errors;
- Correction to formulæ concerning Chebyshev polynomials on page 125;
- Minor changes to the text;
- Expanded index section.

Version 2.0 (April 2004). The main changes introduced in this revision were

- Incorporation of SS*f*M Best Practice Guide No. 10: *Discrete Model Validation* [11];
- Review of statistical concepts;
- More explicit description of statistical models in terms of random variables;
- Tutorial material on generalised Gauss-Markov regression, asymptotic least squares, maximum likelihood estimation, Bayesian parameter estimation;
- Tutorial material on Fourier series, asymptotic polynomials, tensor product surfaces, wavelets, radial basis functions, neural networks, and nonuniform rational B-splines;
- Additional case studies.

Version 3.0 (March 2007). The main changes introduced in this revision are

- Additional tutorial material on Bayesian formulations and data fusion;
- Removal of the case studies from printed version;
- Removal of validation sections from shortened version.

1.3 Discrete modelling resources

1.3.1 Reference books

Discrete modelling draws on a number of disciplines, including data approximation, optimisation, numerical analysis and numerical linear algebra, and statistics. Although aspects of discrete modelling are technically difficult, much of it relies on a few underlying concepts covered in standard text books; see, for example, [89, 115, 117, 143, 152, 171, 185, 189, 207, 209]. Many text books and reference books have explicit descriptions of algorithms; see e.g. [107, 117, 167, 209], and a number of books also supply software on a disk, including the *Numerical Recipes* family of books [186] which give reasonably comprehensive guidance on algorithm design and further reading.

1.3.2 Conference series

While standard textbooks are valuable for understanding the basic concepts, few are concerned with metrology directly. The main objective of the conference series *Advanced Mathematical and Computational Tools in Metrology* is to discuss how these mathematical, numerical and computational techniques can be used in metrology. Collected papers associated with the conferences are published; see [42, 43, 44, 45, 47, 183, 46]. Many of the papers present survey or tutorial material directly relevant to discrete modelling; see, for example, [19, 21, 22, 23, 37, 41, 58, 59, 60, 61, 63, 78, 84, 96, 98, 116, 149, 150, 155, 188].

The conference series *Algorithms for Approximation* [75, 146, 153, 154, 146, 138] deals with more general aspects of data approximation, many of which have direct relevance to metrology.

1.3.3 Software sources

The last four decades have been ones of great success in terms of the development of reliable algorithms for solving the most common computational problems. In the fields of numerical linear algebra – linear equations, linear least squares, eigenvalues, matrix factorisations — and optimisation — nonlinear equations, nonlinear least squares, minimisation subject to constraints, linear programming, nonlinear programming — there is now a substantial core of software modules which the metrologist can exploit.

The scientist has a range of sources for software: i) specialist software developers/collectors such as the NAG library in the UK and IMSL in the US, ii) National laboratories, for example NPL, Harwell, Argonne, Oakridge, iii) universities, iv) industrial laboratories, v) software houses and vi) instrument manufacturers. Library software, used by many scientists and continually maintained, provides perhaps the best guarantee of reliability.

Library software. Below is a list of some of the libraries which have routines relevant to the metrologist.

NAG: A large Fortran Library[1] covering most of the computational disciplines including quadrature, ordinary differential equations, partial differential equations, integral equations, interpolation, curve and surface fitting, optimisation, linear algebra (simultaneous linear equations, matrix factorisations, eigenvalues), correlation and regression analysis, analysis of variance and non-parametric statistics. [175]

IMSL: International Mathematical and Statistical Libraries, Inc. Similar to NAG but based in the US. [206]

LINPACK: A Fortran library for solving systems of linear equations, including least-squares systems, developed at Argonne National Laboratory (ANL), USA. See [83], and Netlib (below).

EISPACK: A companion library to LINPACK for solving eigenvalue problems also developed at ANL. See [194], and Netlib (below).

[1] The NAG Library is available in other languages

LAPACK: A replacement for, and further development of, LINPACK and EISPACK. LAPACK also appears as a sub-chapter of the NAG library. See [192], and Netlib (below).

Harwell: Optimisation routines including those for large and/or sparse problems. [127]

DASL: Data Approximation Subroutine Library, developed at NPL, for data interpolation and approximation with polynomial and spline curves and surfaces. [8]

MINPACK: Another Fortran Library developed at ANL for function minimisation. MINPACK contains software for solving nonlinear least-squares problems, for example. See [112], and Netlib (below).

A number of journals also publish the source codes for software. In particular the *ACM Transactions on Mathematical Software* has published over 700 algorithms for various types of computation. *Applied Statistics* publishes software for statistical computations.

Most library software has been written in Fortran 77, a language well suited to numerical computation but in other ways limited in comparison with more modern languages. The situation has changed radically with the advent of new versions of the language — Fortran 90/95 [149, 159] — which have all the features that Fortran 77 was perceived as lacking while maintaining full backwards compatibility. Using Fortran 90/95 to create dynamically linked libraries (DLLs), it is relatively straightforward to interface the numerical library software with spreadsheet packages on a PC, for example, or to software written in other languages. Many library subroutines now also appear in Fortran 90/95 implementations, e.g. [12, 176]; see also [187].

Scientific software packages. There are a number of scientific software packages, including Matlab, Scilab,Mathematica, MathCad and S-Plus that are widely used by numerical mathematicians, scientists and engineers [156, 157, 158, 168, 210, 211, 137]. The online documentation associated with these packages includes extensive tutorial material.

Netlib. A comprehensive range of mathematical software can be obtained over the Internet through *Netlib* [82]. For example, the LINPACK, EISPACK, LAPACK and MINPACK libraries are available through Netlib along with the later algorithms published in *ACM Transactions on Mathematical Software* [190]. The system is very easy to use and there are also browsing, news and literature search facilities.

Statlib. Statlib is similar to Netlib but covers algorithms and software for statistical calculations. [199]

Guide to Available Mathematical Software - GAMS. The Guide to Available Mathematical Software [172] developed and maintained by the National Institute of Standards and Technology (NIST), Gaithersburg, MD, provides a comprehensive listing of mathematical software classified into subject areas such as linear algebra, optimisation, etc. It includes the software in Netlib and the NAG and IMSL libraries. Using the search facilities the user can quickly identify modules in the public domain or in commercial libraries.

e-Handbook of Statistical Methods. NIST/SEMATECH also publishes, online, a Handbook of Statistical Methods [173].

1.3.4 SS*f*M

The resources we have listed so far relate to science in general rather than metrology in particular. Certainly, many of the problems in metrology are generic and it is sensible to apply general solutions where they are appropriate. The SS*f*M programme as a whole aims to bridge the gap between the best computational techniques and the needs of metrology with the main focus of bringing appropriate technology to the metrologist in a usable form. The SS*f*M website [169] continues to provide an access point to a range of resources in the form of software, best-practice guides, reports, etc., and has assembled a large number of documents.

1.4 General notation

See table 1.1.

$\sim$	means 'is distributed as', e.g., $X \sim N(\mu, \sigma^2)$ means the random variable X is associated with the normal distribution with mean μ and standard deviation σ.
$\in$	in a statistical model context means 'is a sample from', e.g., $\epsilon \in N(\mu, \sigma^2)$ means ϵ is a sample from a normal distribution with mean μ and standard deviation σ.
$\sharp$	denotes the end of text concerning an example.
m	number of measurements.
n	number of model parameters $\mathbf{a} = (a_1, \ldots, a_n)^{\mathrm{T}}$.
$\mathbf{a}$	vector of model parameters $\mathbf{a} = (a_1, \ldots, a_n)^{\mathrm{T}}$.
$N(\mu, \sigma^2)$	univariate Gaussian or normal distribution with mean μ and standard deviation σ.
$R(a, b)$	rectangular (uniform) distribution, constant on $[a, b]$ and zero outside this interval.
p	number of model variables $\mathbf{x} = (x_1, \ldots, x_p)^{\mathrm{T}}$.
$\mathbf{x}$	vector of model variables $\mathbf{x} = (x_1, \ldots, x_p)^{\mathrm{T}}$.
$\mathcal{R}$	the set of real numbers.
$\mathcal{R}^n$	the set of n-vectors $\mathbf{x} = (x_1, \ldots, x_n)^{\mathrm{T}}$ of real numbers.
$\{x_i\}_1^m$	set of m elements indexed by $i = 1, 2, \ldots, m$.
$\mathbf{y}$	data vector $\mathbf{y} = (y_1, \ldots, y_m)^{\mathrm{T}}$ of measured values.
$[a, b]$	set of numbers $\{x : a \leq x \leq b\}$.
$\mathbf{a}^{\mathrm{T}}$, A^{T}	transpose of a vector or matrix.
$\mathbf{a}^{\mathrm{T}}\mathbf{b}$	inner product of two vectors $\mathbf{a}^{\mathrm{T}}\mathbf{b} = a_1 b_1 + \ldots + a_n b_n$.
I	identity matrix with 1s on the diagonal and 0s elsewhere.
J	Jacobian matrix associated with a set of functions $f_i(\mathbf{a})$ of parameters: $J_{ij} = \frac{\partial f_i}{\partial a_j}$.
$\mathcal{A}(\mathbf{y})$	parameter estimate determined from data $\mathbf{y}$ by estimator $\mathcal{A}$.
$D(\boldsymbol{\alpha})$	Distribution with parameters from the vector $\boldsymbol{\alpha}$ (section 2.4.2).
$D(\mathbf{u})$	Generalised distance (section 4.3.2).
$E(\boldsymbol{X})$	expectation of the vector of random variables $\boldsymbol{X}$.
$V(\boldsymbol{X})$	variance or uncertainty matrix of the vector of random variables $\boldsymbol{X}$.
$u(x)$	standard uncertainty associated with the estimate, x, of a random variable.
X, Y, etc.,	random variables.
$\nabla_{\mathbf{a}} f$	vector of partial derivatives $(\frac{\partial f}{\partial a_1}, \ldots, \frac{\partial f}{\partial a_n})^{\mathrm{T}}$ for a function $f(\mathbf{a})$ with respect to the parameters $\mathbf{a} = (a_1, \ldots, a_n)^{\mathrm{T}}$.
$\sum_{i=1}^{m} x_i$	sum of elements: $\sum_{i=1}^m x_i = x_1 + \ldots + x_m$.
$\prod_{i=1}^{m} x_i$	product of elements: $\prod_{i=1}^m x_i = x_1 \times \ldots \times x_m$.

Table 1.1: General notation used in this Guide.

Chapter 2

Model building

2.1 Model types

Mathematical modelling, in general, involves the assignment of mathematical terms for all the relevant components of a (measurement) system and the derivation of equations giving the relationships between these mathematical entities. In these models, we can distinguish between terms that relate to quantities that are known or measured and those that are unknown or to be determined from the measurement data. We will in general call the former terms *model variables* and use $\mathbf{x} = (x_1, \ldots, x_p)^{\mathrm{T}}$, $\mathbf{y}$, etc., to denote them and call the latter *model parameters* and denote them by $\mathbf{a} = (a_1, \ldots, a_n)^{\mathrm{T}}$, $\mathbf{b}$, etc.

A *physical model* is one in which there is a theory that defines how the variables depend on each other.

An *empirical model* is one in which a relationship between the variables is expected or observed but with no supporting theory. Many models have both empirical and physical components.

An *explicit model* is one in which one or more of the variables is given as a directly computable function of the remaining variables. We write $y = \phi(\mathbf{x}, \mathbf{a})$ to show that y is a function of the model variables $\mathbf{x}$ and parameters $\mathbf{a}$. If $\mathbf{x}$ and $\mathbf{a}$ are known, then the corresponding value for y can be calculated. The variable y is known as the *response* or *dependent* variable and the variables $\mathbf{x}$ are known as the *covariates*, *stimulus* or *explanatory variables*.[1]

An *implicit model* is one in which the variables are linked through a set of equations. We write, for example, $f(\mathbf{x}, \mathbf{a}) = 0$ to show that the components of $\mathbf{x}$ are related implicitly. It is often possible to write one variable as a function of the others, e.g.,

$$x_1 = \phi_1(x_2, \ldots, x_p, \mathbf{a}).$$

[1] The term *independent variable* is sometimes used but the use of the word 'independent' can be confused with the notion of statistical independence.

Example: implicitly and explicitly defined circle

The equation for a circle centred at (a_1, a_2) with radius a_3 can be written implicitly as

$$f(\mathbf{x}, \mathbf{a}) = (x_1 - a_1)^2 + (x_2 - a_2)^2 - a_3^2 = 0.$$

We can solve for x_1 explicitly in terms of x_2,

$$x_1 = a_1 \pm \sqrt{[a_3^2 - (x_2 - a_2)^2]},$$

or for x_2 in terms of x_1,

$$x_2 = a_2 \pm \sqrt{[a_3^2 - (x_1 - a_1)^2]}.$$

We can rewrite these two equations in parametric form $\mathbf{x} = \boldsymbol{\phi}(u, \mathbf{a})$ as

$$\begin{aligned} (x_1, x_2) &= (a_1 \pm \sqrt{[a_3^2 - (u - a_2)^2]}, u), \quad \text{or} \\ (x_1, x_2) &= (u, a_2 \pm \sqrt{[a_3^2 - (u - a_1)^2]}). \end{aligned}$$

The first equation becomes problematical when $|x_2 - a_2| \approx a_3$ while the second when $|x_1 - a_1| \approx a_3$. It is often the case that when going from an implicit expression to an explicit expression there is a preferred choice (depending on the particular circumstances) and that some choices are excluded because the equations become singular in some way. Sometimes an implicit form is preferable even when an explicit form can be deduced from it because the former has better numerical stability.

Alternatively, we can express the circle parametrically $\mathbf{x} = \boldsymbol{\phi}(u, \mathbf{a})$ as

$$x_1 = a_1 + a_3 \cos u, \quad x_2 = a_2 + a_3 \sin u.$$

This form is valid for all values of u. ♯

A *linear model* is one in which the parameters $\mathbf{a}$ appear linearly. For explicit models, it takes the form

$$y = \phi(\mathbf{x}, \mathbf{a}) = a_1\phi_1(\mathbf{x}) + \ldots + a_n\phi_n(\mathbf{x}),$$

where the functions $\phi_j(\mathbf{x})$ are *basis functions* depending on the variables $\mathbf{x}$.

A *nonlinear model* is one in which one or more of the parameters $\mathbf{a}$ appear nonlinearly.

Example: exponential decay

The function

$$y = a_1 e^{-a_2 x}$$

is an example of a nonlinear (explicit) model since the parameter a_2 appears nonlinearly. ♯

Many (but by no means all) of the models that occur in practice such as polynomials (section 5.1) and splines (section 5.2) are linear. They have the advantage that when it comes to determining best estimates of the model parameters from data (chapter 3) the equations that arise are easier to solve.

2.2 Space of models

Consider an experimental set up in which a response variable y depends on a number of covariates $\mathbf{x} = (x_1, \ldots, x_p)^{\mathrm{T}}$. We make the assumption that the system is deterministic in

that the same values of the variables gives rise to the same response, i.e., if $\mathbf{x}_1 = \mathbf{x}_2$ then correspondingly $y_1 = y_2$. With this assumption, we can say that the response y is a *function* of the variables $\mathbf{x}$ and write

$$y = \phi(\mathbf{x}),$$

to denote this relationship. If we assume that the response y depends continuously on each of the variables x_k, then we can restrict the choices for ϕ to be continuous functions. Further assumptions will in turn limit the possible choices for ϕ.

The goal of the modelling process is to include enough information about the system so that the range of choices for the function ϕ is determined by specifying a finite number of additional parameters $\mathbf{a} = (a_1, \ldots, a_n)^{\mathrm{T}}$. Each set of values of these parameters determines uniquely a response function $y = \phi(\mathbf{x}, \mathbf{a})$. We call the collection of all such functions $\phi(\mathbf{x}, \mathbf{a})$ the *space of models*. Ideally, the actual response function ϕ is specified by one such function $\phi(\mathbf{a}, \mathbf{x})$, i.e., the space of models is large enough to model the actual behaviour. On the other hand we do not want the space of models to include functions that represent system behaviour that is physically impossible, i.e., the space of models should not be too large.

Example: linear response

One of the most common types of model is one in which the response variable depends linearly on a single variable x:

$$y = \phi(x, a_1, a_2) = a_1 + a_2 x,$$

specified by intercept a_1 and slope a_2. Here the space of models is the collection of linear functions $\{y = a_1 + a_2 x\}$. ♯

The term *linear response* model should not be confused with a linear model (defined in section 2.1), although linear response models are linear because $a_1 + a_2 x$ is linear in (a_1, a_2).

Example: exponential decay

Suppose the response y is an exponential decay depending on the single variable x (time, say). Then y can be modelled as

$$y = \phi(x, a_1, a_2) = a_1 e^{-a_2 x}$$

depending on two parameters a_1 and a_2. Here, the space of models is the collection of functions $\{y = a_1 e^{-a_2 x}\}$. ♯

Example: circles

In dimensional metrology, the nominal shape of the cross section of a shaft is modelled as a circle. A circle (in a given Cartesian co-ordinate system) can be specified by three parameters, its centre coordinates (a_1, a_2) and radius a_3. To each set of parameters (a_1, a_2, a_3), we associate the circle

$$\{(x, y) : (x - a_1)^2 + (y - a_2)^2 = a_3^2\}.$$

♯

Example: water density

A number of models for the density of water y as a function of temperature x have been

proposed, e.g.

$$\begin{aligned}
\frac{y}{y_0} &= \phi_1(x, a_1, \ldots, a_4) = 1 - \frac{a_2(x-a_1)^2(x+a_3)}{x+a_4}, \\
\frac{y}{y_0} &= \phi_2(x, a_1, \ldots, a_6) = 1 - \frac{a_2(x-a_1)^2(x+a_3)(x+a_5)}{(x+a_4)(x+a_6)}, \\
\frac{y}{y_0} &= \phi_3(x, a_1, \ldots, a_6) = 1 - \sum_{j=1}^{5} a_{j+1}(x-a_1)^j, \\
\frac{y}{y_0} &= \phi_4(x, a_1, \ldots, a_9) = 1 - \sum_{j=1}^{9} a_j x^{j-1},
\end{aligned}$$

where y_0 represents the maximum density. These models are empirical in that there is no theory to define exactly the space of models. Note that the number of parameters (4, 6, 6 and 9) used to specify the functions differs from model to model. This is often the case with empirical models. ♯

In some sense, the essence of model building is being able to define the right number and type of parameters that are required to characterise the behaviour of the system adequately.

2.3 Model parameterisation

Model parameterisation is concerned with how we specify members of the space of models. Given a space of models, a *parameterisation* assigns to a set of values of the parameters **a** a unique member of the space of models, e.g., a particular curve from a family of curves.

Example: straight lines

The equation

$$L_1 : (a_1, a_2) \mapsto \{y = a_1 + a_2 x\}$$

associates to the pair of parameters (a_1, a_2) the linear function $y = a_1 + a_2 x$. Consider, also,

$$L_2 : (a_1, a_2) \mapsto \{y = a_1 + a_2(x - 100)\}.$$

These two methods are mathematically equivalent in the sense that given any pair (a_1, a_2) it is possible to find a unique pair (a_1', a_2') such that L_2 assigns the same line to (a_1', a_2') as L_1 assigns to (a_1, a_2), and vice versa. From a numerical point of view, the parameterisation L_2 may be preferable if the variable x is likely to have values around 100. However, the parameterisation

$$L_3 : (a_1, a_2) \mapsto \{x = a_1 + a_2 y\}$$

is not equivalent to L_1 since there is no pair (a_1, a_2) that L_3 can assign to the line $y = 0$. Similarly, L_1 cannot represent the line $x = 0$.

Note that the parameterisation

$$L_4 : (a_1, a_2) \mapsto \{-x \sin a_1 + y \cos a_1 = a_2\}$$

can be used to represent all lines. ♯

Example: circles

The assignment

$$C_1 : (a_1, a_2, a_3) \mapsto \{(x, y) : (x - a_1)^2 + (y - a_2)^2 = a_3^2\},$$

parameterises circles in terms of their centre coordinates and radius. Consider also

$$\begin{aligned} C_2 : (a_1, a_2, a_3) &\mapsto \{(x, y) : x^2 + y^2 + a_1 x + a_2 y + a_3 = 0\}, \\ C_3 : (a_1, a_2, a_3) &\mapsto \{(x, y) : a_1(x^2 + y^2) + a_2 x + y = a_3\}. \end{aligned}$$

The parameterisations C_1 and C_2 are equivalent to each other in that they can represent exactly the same set of circles but not to C_3. The parameterisation C_3 can be used to model arcs of circle approximately parallel to the $x-$axis in a stable way. Indeed, lines (in this context, circles with infinite radius) correspond to circles with $a_1 = 0$ in this parameterisation. ♯

2.3.1 Centering and scaling

Model parameterisations that are equivalent from a mathematical point of view may have different characteristics numerically. For example, we can scale or translate the variables and parameters and still define the same model space.

Example: variable transformation for a quadratic curve

Suppose in an experiment, the response y is modelled as a quadratic function of the variable x,

$$y = a_1 + a_2 x + a_3 x^2,$$

and x is expected to lie in the range $[95,\ 105]$. Using this equation, the quadratic curves are specified by the coefficients a_1, a_2 and a_3. We can instead parameterise these curves in terms of the transformed variable z

$$y = b_1 + b_2 z + b_3 z^2,$$

where $z = (x - 100)/5$ is expected to lie in the range $[-1,\ 1]$. ♯

More generally, given a model of the form $y = \phi(\mathbf{x}, \mathbf{a})$, we can reparameterise it as $y = \phi(\mathbf{z}, \mathbf{b})$ where

$$\mathbf{z} = D(\mathbf{x} - \mathbf{x}_0), \quad \mathbf{b} = E(\mathbf{a} - \mathbf{a}_0),$$

and D and E are $p \times p$ and $n \times n$ nonsingular scaling matrices and $\mathbf{x}_0$ and $\mathbf{a}_0$ fixed p- and n-vectors. Typically, we set $\mathbf{x}_0$ to be the centroid of the data:

$$\mathbf{x}_0 = \frac{1}{m} \sum_{i=1}^{m} \mathbf{x}_i,$$

$\mathbf{a}_0$ to be middle of the expected range for the parameters $\mathbf{a}$ and set the scaling matrices such that

$$\frac{\partial \phi}{\partial z_k}, \frac{\partial \phi}{\partial b_j} \approx \pm 1 \quad \text{near} \quad \mathbf{z} = \mathbf{0}, \quad \mathbf{b} = \mathbf{0}.$$

These transformations will generally improve the numerical performance of algorithms operating with the model. Often, the improvements are very significant.

2.3.2 Choice of basis functions

Suppose we have a linear model defined in terms of the basis functions ϕ_j as

$$y = \phi(\mathbf{x}, \mathbf{a}) = a_1\phi_1(\mathbf{x}) + \ldots + a_n\phi_n(\mathbf{x}).$$

Given a nonsingular $n \times n$ matrix D whose jth column is $\mathbf{d}_j$, we can define new basis functions ψ_j according to

$$\psi_j(\mathbf{x}) = d_{1j}\phi_1(\mathbf{x}) + \ldots + d_{nj}\phi_n(\mathbf{x}),$$

and reparameterise the model as

$$y = \psi(\mathbf{x}, \mathbf{b}) = b_1\psi_1(\mathbf{x}) + \ldots + b_n\psi_n(\mathbf{x}),$$

in order to improve the stability of the model. Such considerations are particularly important for polynomial or spline models (sections 5.1, 5.2).

2.3.3 Resolving constraints

Often the natural 'parameters' used to describe a model give rise to degrees of freedom that need to be resolved.

Example: parameters describing the geometry of targets on a planar artefact

In dimensional metrology, artefacts such as a hole plate can be modelled as a set of targets lying in a plane. The location of these targets can be described by their coordinates $\mathbf{a} = (a_1, b_1, a_2, b_2, \ldots, a_n, b_n)^{\mathrm{T}}$ where $\mathbf{a}_j = (a_j, b_j)^{\mathrm{T}}$ is the location of the jth target. However, the parameters $\mathbf{a}$ do not specify the frame of reference for the targets and three constraints have to be introduced to fix the three degrees of freedom (two translational and one rotational) associated with the system.

For example, suppose there are four points nominally at the corners of a square. We can eliminate the translational degrees of freedom by constraining the centroid $(\bar{a}, \bar{b})^{\mathrm{T}}$ to be at $(0, 0)^{\mathrm{T}}$:

$$\bar{a} = \frac{1}{n}\sum_{j=1}^{n} a_j = 0, \quad \bar{b} = \frac{1}{n}\sum_{j=1}^{n} b_j = 0, \quad \text{where } n = 4 \text{ for a square}$$

Similarly, we can fix the orientation of the targets by constraining one of the targets to lie on the line $y = x$: $a_1 = b_1$, say. These three constraints can be written in the form

$$D\mathbf{a} = \mathbf{0},$$

where D is the 3×8 matrix

$$D = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 4 & -4 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

♯

2.4 Uncertainty structure in measurement data

In this section, we review briefly some of the statistical concepts used to represent our uncertainty or degree of belief in measurement data. See, for example, [69, 70, 139].

2.4.1 Probability

The probability $P(A)$ of a statement (proposition, event) A is a real number between 0 and 1, with 0 meaning the statement must be false and 1 that it must be true. The larger the probability, the more likely the statement is to be true. The probability of A and B being true is denoted by $P(A, B)$. The notation $P(A|B)$ means the probability of A given that statement B is true. There are two basic rules that define how probabilities are combined.[2] If $\bar{A}$ represents the statement 'A is false' then

$$P(A) + P(\bar{A}) = 1.$$

This is called the *sum rule*. The *product rule* states that

$$P(A, B) = P(A|B) \times P(B) = P(B|A) \times P(A),$$

in words, the probability that both A and B are true is equal to the probability that A is true given that B is true times the probability that B is true. Two statements A and B are independent if $P(A|B) = P(A)$ and $P(B|A) = P(B)$, i.e., the probability of one being true does not depend on our knowledge of the other. For independent A and B, the product rule is $P(A, B) = P(A)P(B)$.

Bayes' Theorem arises from a rearrangement of the product rule:

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}. \tag{2.1}$$

If we regard A as a statement about parameters and B as a statement about measurement data, Bayes' Theorem allows us to update our original information $P(A)$ about A in the light of the measurements B; see section 3.5.

2.4.2 Random variables and distributions

We use a random variable X to represent a quantity about which we have uncertain knowledge. The quantity may be discrete, for example, the number of pills in a bottle taken from a production line in a pharmaceutical plant, or continuous, for example, the volume of liquid in a medicine bottle from another production line. We associate to a random variable X a probability distribution which allows us to assign probabilities to statements about X.

Discrete random variables

A *discrete random variable* X is a variable that can take only a finite number of possible values. The *frequency function* $p(x)$ states the probabilities of occurrence of the possible

[2] The work of R. T. Cox [77] showed that these rules are essentially unique and that any useful theory of probability would have to obey them.

outcomes:

$$p(x) = P(X = x),$$

the probability that the outcome is x. The *distribution function* $G(x)$ gives the probability that a random variable takes a value no greater than a specified value:

$$G(x) = P(X \leq x), \qquad -\infty < x < \infty.$$

The distribution function varies from zero to one throughout its range, never decreasing.

Continuous random variables

A *continuous random variable* X is a variable that can take any value in its range (which may be infinite). For a continuous random variable X, the counterpart of the frequency function (for a discrete random variable) is the *probability density function* (PDF) $g(x)$. This function has the property that the probability that the value of X lies between a and b is

$$P(a < X < b) = \int_a^b g(x)\,dx.$$

In order that the sum rule is obeyed, PDFs must have unit area, i.e.,

$$P(-\infty < X < \infty) = \int_{-\infty}^{\infty} g(x)\,dx = 1.$$

For example, the rectangular PDF is a density function that describes the fact that the value of X is equally likely to lie anywhere in an interval $[a,\ b]$:

$$g(x) = \begin{cases} \frac{1}{b-a}, & a \leq x \leq b, \\ 0, & \text{otherwise.} \end{cases}$$

We use the notation $X \sim U(a, b)$ to indicate that X has a rectangular distribution defined on the interval $[a, b]$.

The *distribution function* $G(x)$ gives the probability that a random variable takes a value no greater than a specified value, and is defined as for a discrete random variable:

$$G(x) = P(X \leq x), \qquad -\infty < x < \infty.$$

The distribution function can be expressed in terms of the probability density function as

$$G(x) = \int_{-\infty}^{x} g(t)\,dt.$$

Conversely, $g(x) = G'(x)$, the derivative of G. A continuous probability distribution can therefore be defined in terms of either the distribution function G or the probability density function g.

A function $Y = h(X)$ of a random variable X is also random variable and its distribution is determined by h and the PDF of X.

Probability density functions used in practice are usually determined by a small number of parameters. One of the most important distributions is the *normal* or *Gaussian distribution* whose PDF is

$$g(x) = g(x|\mu, \sigma) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right\}.$$

We use the notation $X \sim N(\mu, \sigma^2)$ to indicate X is a random variable associated with a normal distribution defined by parameters μ and σ. More generally, $X \sim D(\boldsymbol{\alpha})$ means that X is associated with a probability distribution D whose PDF is defined in terms of parameters $\boldsymbol{\alpha}$.

A vector $\boldsymbol{X} = (X_1, \ldots, X_n)^{\mathrm{T}}$ of random variables has a multivariate distribution defined in terms of a nonnegative multivariate function $g(\mathbf{x})$. Two random variables $(X, Y)^{\mathrm{T}}$ are *independently distributed* if the associated PDF $g(x, y)$ can be factored as $g(x, y) = g_x(x)g_y(y)$.

A distribution is *unimodal* if its PDF $g(\mathbf{x})$ attains a maximum at a unique point $\mathbf{x}_M$, its *mode*: $g(\mathbf{x}_M) \geq g(\mathbf{x})$ for all $\mathbf{x}$ and $g(\mathbf{x}) = g(\mathbf{x}_M)$ only if $\mathbf{x} = \mathbf{x}_M$. Distributions with one than one local maximum are called *multimodal*.

2.4.3 Operations on distributions

Measures of location and dispersion

For a given distribution, it is usual to calculate, if possible, quantities that provide a useful summary of its properties. A measure of location $L(X)$ is such that $L(X + c) = L(X) + c$ and is used to determine a representative value for X. A measure of dispersion (spread) $S(X)$ is such that $S(cX) = cS(X)$ and gives an estimate of the size of the likely range of values of X.

Expectations

Summarising quantities are often derived in terms of expectations.
If $X \sim D$ has associated PDF $g(x)$ and $h(X)$ is a function of X, then the expectation $E(h(X))$ of $h(X)$ is

$$E(h(X)) = \int_{-\infty}^{\infty} h(x)g(x)\,dx.$$

(It may be that this integral is not finite in which case $E(h(X))$ is said not to exist.)

Mean, variance and standard deviation

Of particular importance are the *mean* $\mu = E(X)$,

$$\mu = \int_{-\infty}^{\infty} xg(x)\,dx,$$

and the *variance* $V(X) = E((X - E(X))^2)$:

$$V(X) = \int_{-\infty}^{\infty} (x-\mu)^2 g(x)\,dx, \quad \mu = E(X).$$

The positive square root of the variance is known as the *standard deviation* and is usually denoted by σ so that $\sigma^2 = V(X)$. The mean is a measure of location of X and the standard deviation is a measure of dispersion. We note that if $X \sim N(\mu, \sigma^2)$ then $E(X) = \mu$ and $V(X) = \sigma^2$. If X has a rectangular distribution, $X \sim R(a, b)$, then $E(X) = (a + b)/2$ and $V(X) = (b - a)^2/3$.

Expectations can also be applied to multivariate distributions. For example, the *covariance* $C(X, Y)$ of a pair (X, Y) of random variables with joint PDF $g(x, y)$ is defined to be

$$\begin{aligned} C(X,Y) &= E((X - E(X))(Y - E(Y))), \\ &= \int (x-\mu_X)(y-\mu_Y) g(x,y)\,dx\,dy, \quad \text{where} \\ \mu_X &= E(X) = \int x g(x,y)\,dx\,dy, \quad \mu_Y = E(Y) = \int y g(x,y)\,dx\,dy, \end{aligned}$$

and $V(X) = C(X, X)$. More generally, if $\boldsymbol{X} = (X_1, \ldots, X_n)^{\mathrm{T}}$ is a vector of random variables with joint PDF $g(\mathbf{x})$, $\mathbf{x} = (x_1, \ldots, x_n)^{\mathrm{T}}$, then $E(\boldsymbol{X}) = \boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)^{\mathrm{T}}$ where $\mu_j = E(X_j)$ is given by

$$\mu_j = \int x_j g(\mathbf{x})\,d\mathbf{x} = \int \cdots \int x_j g(\mathbf{x})\,dx_1 \ldots dx_n,$$

and

$$C(X_j, X_k) = \int (x_j - \mu_j)(x_k - \mu_k) g(\mathbf{x})\,d\mathbf{x}.$$

The *variance matrix* $V(\boldsymbol{X})$, also known as the *uncertainty matrix* , *covariance* or *variance-covariance matrix*, is the $n \times n$ matrix with $V_{jk} = C(X_j, X_k)$.

Example: multivariate normal (Gaussian) distribution

The multivariate normal (Gaussian) distribution for n variables $N(\boldsymbol{\mu}, V)$ is defined by its mean $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)^{\mathrm{T}}$ and $n \times n$ variance matrix V and has PDF

$$p(\mathbf{x}|\boldsymbol{\mu}, V) = \frac{1}{|2\pi V|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}} V^{-1} (\mathbf{x}-\boldsymbol{\mu})\right\}, \tag{2.2}$$

where $|V|$ denotes the determinant of V. ♯

There are simple rules for calculating means and variances of linear combinations of random variables. If X and Y are random variables and c and d are two constants, then

$$\left.\begin{aligned} E(cX + dY) &= cE(X) + dE(Y), \\ V(cX + dY) &= c^2 V(X) + d^2 V(Y) + 2cdC(X, Y) \end{aligned}\right\}. \tag{2.3}$$

Marginalisation

Given a pair of random variables X and Y with joint PDF $g(x, y)$, the marginalised distributions for X and Y alone have PDFs defined by

$$g(x) = \int_{-\infty}^{\infty} g(x, y)dy, \quad g(y) = \int_{-\infty}^{\infty} g(x, y)dx,$$

respectively. More generally, if $\boldsymbol{X} = (X_1, \ldots, X_n)^{\mathrm{T}}$, $n > 2$, is a vector of random variables with joint PDF $g(\mathbf{x})$, X_1 has PDF

$$g(x_1) = \int \cdots \int g(\mathbf{x})\, dx_2 \ldots dx_n,$$

and X_1 and X_2 have joint PDF

$$g(x_1, x_2) = \int \cdots \int g(\mathbf{x})\, dx_3 \ldots dx_n,$$

etc. The definitions of means and variances associated with multivariate distributions, section 2.4.3, can also be expressed in terms of marginalised distributions, e.g.,

$$\mu_1 = E(X_1) = \int x_1 g(\mathbf{x}) d\mathbf{x} = \int x_1 \left\{ \int \cdots \int g(\mathbf{x})\, dx_2 \cdots dx_n \right\} dx_1 = \int x_1 g(x_1)\, dx_1,$$

and

$$\begin{aligned} C(X_1, X_2) &= \int (x_1 - \mu_1)(x_2 - \mu_2) g(\mathbf{x})\, d\mathbf{x}, \\ &= \int (x_1 - \mu_1)(x_2 - \mu_2) \left\{ \int \cdots \int g(\mathbf{x})\, dx_3 \ldots dx_n \right\} dx_1 dx_2 \\ &= \int (x_1 - \mu_1)(x_2 - \mu_2) g(x_1, x_2)\, dx_1 dx_2, \end{aligned}$$

etc.

Function of a random variable

If X is a random variable associated with the PDF $g_X(x)$ and $Y = s(X)$ where $y = s(x)$ is a one-to-one function with inverse $x = t(y)$, then the PDF associated with Y is

$$g_Y(y) = g_X(t(y)) \left| \frac{dt}{dy} \right|. \tag{2.4}$$

2.4.4 Propagation of uncertainties

The law of propagation of uncertainties (LPU), see [24] and [69, chapter 6] is derived from the rules for means and variances expressed in (2.3). Suppose first that Y is a linear combination of n random variables $X = (X_1, \ldots, X_n)^{\mathrm{T}}$,

$$Y = c_1 X_1 + \ldots + c_n X_n = \mathbf{c}^{\mathrm{T}} \boldsymbol{X}, \tag{2.5}$$

where $\mathbf{c} = (c_1, \ldots, c_n)^{\mathrm{T}}$ are known constants. Suppose that the random variables X_j are associated with distributions with means x_j and standard deviations $u_j = u(x_j)$ and that the X_j are independently distributed. A simple extension of (2.3) shows that Y is associated with a distribution with mean

$$y = E(Y) = c_1 E(X_1) + \ldots + c_n E(X_n) = c_1 x_1 + \ldots + c_n x_n,$$

and variance

$$u^2(y) = V(Y) = c_1^2 V(X_1) + \ldots + c_n^2 V(X_n) = c_1^2 u_1^2 + \ldots + c_n^2 u_n^2.$$

This is true no matter the distributions for X_j (so long as their means and standard deviations are defined).

The rule can be extended to take into account covariances. If $\boldsymbol{X}$ is a vector of random variables whose joint probability distribution has mean $\boldsymbol{\mu}$ and variance matrix V and Y is the linear combination $Y = \mathbf{c}^{\mathrm{T}}\boldsymbol{X}$, then the distribution associated with Y has mean $\mathbf{c}^{\mathrm{T}}\boldsymbol{\mu}$ and variance $\mathbf{c}^{\mathrm{T}}V\mathbf{c}$:

$$E(Y) = \mathbf{c}^{\mathrm{T}}\boldsymbol{\mu}, \quad V(Y) = \mathbf{c}^{\mathrm{T}}V\mathbf{c}. \tag{2.6}$$

Example: linear combinations of normal variates

The statement about the propagation of uncertainties can be made more strongly for combinations of normal variates. If random variables $\boldsymbol{X}$ are associated with the multivariate normal distribution $\boldsymbol{X} \sim N(\boldsymbol{\mu}, V)$, and $Y = \mathbf{c}^{\mathrm{T}}\boldsymbol{X}$ then Y is associated with the univariate Gaussian distribution $Y \sim N(\mathbf{c}^{\mathrm{T}}\boldsymbol{\mu}, \mathbf{c}^{\mathrm{T}}V\mathbf{c})$. In this case, the form of the distribution associated with Y is known precisely, not just its mean and variance. In particular, if the jth diagonal element V_{jj} is σ_j^2, then $X_j \sim N(\mu_j, \sigma_j^2)$. ♯

Now suppose Y is defined as a function $Y = f(X_1, \ldots, X_n)$ with the X_j distributed as before. The random variable Y is associated with a distribution and we wish to know its mean and standard deviation. We can find an approximate answer by linearising the function Y about $y = f(\mathbf{x})$. In (2.5) the constant c_j represents the *sensitivity* of Y with respect to changes in X_j: if X_j changes by Δ_j then Y changes by $c_j\Delta_j$. For a nonlinear function f, the sensitivity of Y with respect to a change in X_j is given by the partial derivative $c_j = \partial f/\partial X_j$ evaluated at x_j. (This partial derivative is simply the slope at $X_j = x_j$ of the function f regarded as a function of X_j alone with all other variables held fixed.) The linear approximation can then be written as

$$Y - y \approx c_1(X_1 - x_1) + \ldots + c_n(X_n - x_n),$$

or

$$\tilde{Y} \approx c_1\tilde{X}_1 + \ldots + c_n\tilde{X}_n, \tag{2.7}$$

with new random variables $\tilde{Y} = Y - y$ and $\tilde{X}_j = X_j - x_j$, $j = 1, \ldots, n$.

Equation (2.7) is of the same form as (2.5) and so

$$E(Y - y) = E(Y) - y \approx c_1(E(X_1) - x_1) + \ldots + c_n(E(X_n) - x_n) = 0,$$

i.e., $E(Y) \approx y$, and

$$\begin{aligned} u_y^2 &= V(Y - y) = V(Y) \\ &\approx c_1^2 V(X - x_1) + \ldots + c_n^2 V(X_n - x_n) \\ &= c_1^2 u_1^2 + \ldots c_n^2 u_n^2. \end{aligned}$$

Here, we have used the rule $V(X - x) = V(X)$. In summary, for nonlinear functions $Y = f(\boldsymbol{X})$ we use the same rule (2.6) as for linear functions but with the sensitivities c_j calculated as partial derivatives. We must be aware, however, that the resulting estimates of the mean and standard deviation are derived from a linearisation and may be different from the actual values .

2.4.5 Measurement model

The space of models represents the mathematical relationship between the various variables and parameters. In practice, the values of the variables are inferred from measurements subject to random effects that are difficult to characterise completely. These effects are modelled as random variables, generally with expectation zero. The actual measured values are regarded as observations of the associated random variable drawn from the corresponding statistical distribution.

Suppose that the response y is modelled as a function $y = \phi(\mathbf{x}, \mathbf{a})$ depending on variables $\mathbf{x}$ and model parameters $\mathbf{a}$ and that measurements of y are subject to random effects. The measurement model is of the form

$$Y = \phi(\mathbf{x}, \mathbf{a}) + E, \qquad E(E) = 0.$$

We note that since $E(E) = 0$, $E(Y) = \phi(\mathbf{x}, \mathbf{a})$, i.e., the value of $\phi(\mathbf{x}, \mathbf{a})$ predicted by the model $\phi(\mathbf{x}, \mathbf{a})$ is equated with the expected value of the random variable Y. Suppose measurements y_i are gathered with $y_i \in Y$, i.e., y_i is an observation of the random variable Y_i where

$$Y_i = \phi(\mathbf{x}_i, \mathbf{a}) + E_i, \qquad E(E_i) = 0.$$

We can then write

$$y_i = \phi(\mathbf{x}_i, \mathbf{a}) + \epsilon_i, \qquad \epsilon_i \in E_i,$$

where $\epsilon_i = y_i - \phi(\mathbf{x}_i, \mathbf{a})$ represents the observed value of the random variable E_i and can be thought of the *deviation* between the measurement value and the model prediction. (In data approximation, we sometimes refer to ϵ_i as the *approximation* or *residual error*.)

In many situations the measurements of two or more variables are subject to significant random effects. In this case the measurement model has a more general form such as

$$\begin{aligned} &\boldsymbol{X} = \mathbf{x}^* + \boldsymbol{D}, \quad E(\boldsymbol{D}) = \mathbf{0}, \quad E(\boldsymbol{X}) = \mathbf{x}^*; \\ &Y = \phi(\mathbf{x}^*, \mathbf{a}) + E, \quad E(E) = 0. \end{aligned}$$

Measurements $(\mathbf{x}_i, y_i)$ are regarded as observations of the random variables $(\boldsymbol{X}, Y)$, $i = 1, \ldots, m$, and we write

$$y_i = \phi(\mathbf{x}^* + \boldsymbol{\delta}_i, \mathbf{a}) + \epsilon_i, \quad i = 1, \ldots, m,$$

with $\boldsymbol{\delta} \in \boldsymbol{D}$ and $\epsilon_i \in E$.

For implicit models $f(\mathbf{x}, \mathbf{a}) = 0$, the corresponding model equations are written as

$$\begin{aligned} &\boldsymbol{X} = \mathbf{x}^* + \boldsymbol{E}, \quad E(\boldsymbol{E}) = \mathbf{0}, \quad E(\boldsymbol{X}) = \mathbf{x}^*; \\ &f(\mathbf{x}^* + \boldsymbol{\epsilon}_i, \mathbf{a}) = 0, \ \boldsymbol{\epsilon}_i \in \boldsymbol{E} \quad i = 1, \ldots, m. \end{aligned}$$

Example: refractive index of air

The refractive index of air is modelled as a function of air temperature, pressure and humidity (and other variables such as carbon dioxide content) with all three subject to significant random effects. ♯

2.4.6 Statistical models for random effects

The uncertainty structure has to describe not only which measurements are subject to random effects but also the statistical nature of these effects. Measurements $\mathbf{y} = (y_1, \ldots, y_m)^{\mathrm{T}}$ are regarded as observations associated with random variables $\boldsymbol{Y} = (Y_1, \ldots, Y_m)^{\mathrm{T}}$ and the statistical model is described by information about the multivariate statistical distribution for $\boldsymbol{Y}$. Often the information about the multivariate PDF is summarised in terms of the mean $E(\boldsymbol{Y})$ and variance (uncertainty) matrix $V(\boldsymbol{Y})$ rather than specifying the complete PDF.

If measurement y_i is associated with random variable Y_i, then the *standard uncertainty* $u(y_i)$ associated with y_i is the standard deviation of Y_i, i.e.,

$$u^2(y_i) = V(Y_i) = (V(\boldsymbol{Y}))_{ii},$$

the ith diagonal element of uncertainty matrix $V(\boldsymbol{Y})$.

Example: standard experiment model

We will refer to the following model as the *standard experiment model*. A response variable y is modelled as a function $y = \phi(\mathbf{x}, \mathbf{a})$ of variables $\mathbf{x}$ and parameters $\mathbf{a}$ and a set $\{(\mathbf{x}_i, y_i)\}_1^m$ of measurements gathered with each y_i subject to independent random effects described by a normal distribution with zero mean and standard deviation σ. The model equations are

$$y_i = \phi(\mathbf{x}_i, \mathbf{a}) + \epsilon_i, \qquad i = 1, \ldots, m,$$

with $\boldsymbol{\epsilon} \in N(\mathbf{0}, \sigma^2 I)$. These equations represent a complete statement of the model of the experiment.

The standard uncertainty $u(y_i)$ associated with y_i is σ. ♯

There are common variations in this standard model. For example, the standard uncertainties may vary with the measurements, in which case $\epsilon_i \in N(0, \sigma_i^2)$. If the random variables E_i are correlated, with uncertainty matrix V, the vector $\boldsymbol{\epsilon}$ is modelled as belonging to a multinormal distribution: $\boldsymbol{\epsilon} \in N(\mathbf{0}, V)$.

There is further guidance on statistical modelling for random effects in [70].

Chapter 3

Model fitting, parameter estimation and inference

Introduction

This chapter overviews some of the main concepts associated with determining estimates of parameter values on the basis of the measurement model and the measurement data. These concepts are derived from three sources, data approximation, classical statistics and Bayesian inference. In data approximation, the main focus has been on developing algorithmic and numerical approaches that give a good fit of the model to the data where the concept of goodness of fit is defined in mathematical terms, without particular reference to a statistical model associated with the data. In classical statistics, the focus is on determining parameter estimation methods that make best use of the data, weighting each data point appropriately. In Bayesian inference, the statistical model is used to define a probability distribution that describes the knowledge about the parameters derived from the data and any prior information. Summary information about the parameters such as best estimates of the parameter values and their associated uncertainties is derived from the probability distribution. The concepts are illustrated using a simple example.

3.1 Approximation from a space of models

The space of models attempts to characterise all possible (or probable) behaviour of a particular type of system, e.g., the ways in which a response variable could vary with its covariates. Model fitting is the process of determining from data gathered from a measurement system, a particular model that adequately represents the system behaviour. Constructing the model space is concerned with defining where we should look to explain the behaviour; model fitting is concerned with selecting the best candidate from the model space.

If the members of the model space are described by parameters $\mathbf{a} = (a_1, \ldots, a_n)^{\mathrm{T}} \in \mathcal{R}^n$ and

the measurement data $\mathbf{y} = (y_1, \ldots, y_m)^{\mathrm{T}} \in \mathcal{R}^m$ is regarded as being generated by a system specified by parameters $\mathbf{a}^* \in \mathcal{R}^n$, then model solving amounts to providing an estimate of $\mathbf{a}^*$ from $\mathbf{y}$. A scheme for determining such an estimate from data we term a *point estimator* or simply an *estimator*. We use the symbols $\mathcal{A}$, $\mathcal{B}$, etc., to denote estimators; $\mathbf{a} = \mathcal{A}(\mathbf{y})$ means the estimate of the model parameters provided by estimator $\mathcal{A}$ from data $\mathbf{y}$.

In Bayesian inference, the statistical model is used to define a probability distribution with density function $p(\mathbf{a}|\mathbf{y})$ that encodes the information about the parameters $\mathbf{a}$ derived from the data $\mathbf{y}$ and any prior information. In this context, a point estimate is usually specified in terms of a well-defined property of the distribution such as its expectation (mean) or mode.

Point estimation has the following geometric interpretation. Suppose the model equations are

$$y_i = \phi(\mathbf{x}_i, \mathbf{a}^*) + \epsilon_i, \qquad i = 1, \ldots, m,$$

with $\boldsymbol{\epsilon} \in N(\mathbf{0}, \sigma^2 I)$. Given $\{\mathbf{x}_i\}_{i=1}^m$, the m-vector $\boldsymbol{\phi}(\mathbf{a}) = (\phi(\mathbf{x}_1, \mathbf{a}), \ldots, \phi(\mathbf{x}_m, \mathbf{a}))^{\mathrm{T}}$ describes an n-dimensional model surface in $\mathcal{R}^m$. As the parameter values $\mathbf{a}$ change, the point $\boldsymbol{\phi}(\mathbf{a})$ moves along the surface. If the measurement data $\mathbf{y}$ were exact, the point $\mathbf{y} = \phi(\mathbf{a}^*)$ would lie exactly on the surface. Due the random effects $\boldsymbol{\epsilon}$, $\mathbf{y}$ is perturbed away from the surface. An estimator is a method of associating to a point $\mathbf{y}$ near the surface a preferred point $\boldsymbol{\phi}(\mathbf{a})$, defined by $\mathbf{a} = \mathcal{A}(\mathbf{y})$, on the surface.

3.2 Error functions and approximation norms

In general, estimators are defined using an *error function* $F(\mathbf{a}|\mathbf{y})$ that provides some measure of how well the data $\mathbf{y}$ matches the model behaviour specified by $\mathbf{a}$. The estimate of $\mathbf{a}^*$ is provided by (the estimate of) the minimiser of $F(\mathbf{a}|\mathbf{y})$, i.e., a point at which F takes a minimum value. Different estimators are associated with different error functions.

In data approximation, error functions are usually constructed to provide an aggregate measure of goodness of fit taking into account all the measurement data. These error functions are often related to approximation norms and the least-squares estimator is one of a family of estimators derived from such norms.

Example: approximation norms

In a standard experiment with model $y = \phi(\mathbf{x}, \mathbf{a})$ and data $\mathbf{z} = \{(\mathbf{x}_i, y_i)\}_1^m$, the quantity

$$f_i = f_i(\mathbf{x}_i, \mathbf{a}) = y_i - \phi(\mathbf{x}_i, \mathbf{a})$$

is a measure of the deviation of the model specified by $\mathbf{a}$ from the data point $(\mathbf{x}_i, y_i)$. An aggregate measure of the fit is given by a norm of the vector $\mathbf{f} = (f_1, \ldots, f_m)^{\mathrm{T}}$ such as the p-norm

$$F_p(\mathbf{a}|\mathbf{z}) = \|\mathbf{f}\|_p = \left\{ \sum_{i=1}^{m} |f_i|^p \right\}^{1/p},$$

for any prescribed value of p satisfying $1 \leq p \leq \infty$. In this guide, the p-norm is denoted by L_p.

Of particular importance in data approximation are the L_1-norm

$$F_1(\mathbf{a}|\mathbf{z}) = \sum_{i=1}^{m} |f_i|,$$

the L_2-norm (least squares)

$$F_2(\mathbf{a}|\mathbf{z}) = \left\{ \sum_{i=1}^{m} f_i^2 \right\}^{1/2},$$

and the L_∞ or Chebyshev norm

$$F_\infty(\mathbf{a}|\mathbf{z}) = \max_{1 \le i \le m} |f_i|.$$

For approximation norms, the preferred point on the model surface is the point closest to $\mathbf{y}$ in the corresponding norm. ♯

3.3 Choice of estimation method

The p-norms, for example, demonstrate that there are many criteria that can be used to determine a fit of a model to data. Which, if any, is best for a particular situation? To answer this question, we need to know what we mean by best.

Suppose that an experimental system is specified by parameters $\mathbf{a}^*$, measured data $\mathbf{y}$ have been gathered, resulting in parameter estimates $\mathbf{a} = \mathcal{A}(\mathbf{y})$. If the data was gathered by an ideal measurement system, free from perturbatory effects, then we would want the estimate $\mathbf{a}$ to be the same as $\mathbf{a}^*$, assuming that we could calculate $\mathbf{a}$ precisely. In the presence of random effects, we should expect $\mathbf{a}$ to be different from $\mathbf{a}^*$. Repeating the experiment again to gather a new set of data would yield a different estimate. Regarding $\mathbf{y}$ as a set of observations of a vector of random variables $\boldsymbol{Y}$ with multivariate PDF $g_{\boldsymbol{Y}}$, then $\mathbf{a}$ is an observation of the vector of random variables $\boldsymbol{A} = \mathcal{A}(\boldsymbol{Y})$. In principle, the PDF $g_{\boldsymbol{A}}$ associated with $\boldsymbol{A}$ is determined by that for $\boldsymbol{Y}$, and has a mean $E(\boldsymbol{A})$ and variance $V(\boldsymbol{A})$. We would like $g_{\boldsymbol{A}}$ to be concentrated in a region close to $\mathbf{a}^*$ so that the probability of observing an estimate $\mathcal{A}(\mathbf{y})$ that is close to $\mathbf{a}^*$ is high. One measure of the effectiveness of an estimator is given by the *mean squared error* (MSE) defined by

$$\mathrm{MSE}(\mathcal{A}) = E((\boldsymbol{A} - \mathbf{a}^*)^2)$$

and the *root mean squared error*, $\mathrm{RMSE}(\mathcal{A}) = (\mathrm{MSE})^{1/2}$. The RMSE is a measure of the likely distance of the estimate from $\mathbf{a}^*$. An estimate $\mathcal{A}$ is *unbiased* if $E(\boldsymbol{A}) = \mathbf{a}^*$, in which case $\mathrm{MSE}(\mathcal{A}) = V(\boldsymbol{A})$. An unbiased estimator with a small variance is *statistically efficient*. Efficiency is used in a relative sense to compare estimators with each other (or with certain theoretical bounds; see e.g., [152, chapter 4]). The MSE depends on both the *bias* $E(\boldsymbol{A}) - \mathbf{a}^*$ and the variance $V(\boldsymbol{A})$. An estimator $\mathcal{A}$ is *consistent* if the more data points we take in each data set $\mathbf{y}$, the closer $\mathbf{a} = \mathcal{A}(\mathbf{y})$ gets to $\mathbf{a}^*$ (in a stochastic sense).

Note that bias and the MSE are defined in terms of $\mathbf{a}^*$ and the analysis above is concerned with the question: given $\mathbf{a}^*$, what is the likely behaviour of the estimates $\mathbf{a} = \mathcal{A}(\mathbf{y})$ due to the likely behaviour of the measurement data specified by the statistical model for $\boldsymbol{Y}$.

Thus, it is concerned with the propagation of uncertainty associated with the data through to those associated parameter estimates, for fixed $\mathbf{a}^*$.

Using measures such as the RMSE to quantify the effectiveness of an estimation method requires the calculation of $E(\boldsymbol{A})$ and $V(\boldsymbol{A})$. For many estimators, the exact calculation of these quantities is not practical. For one important class of estimators, however, the task is straightforward. A *linear estimator* is one for which the estimate $\mathbf{a} = \mathcal{A}(\mathbf{y})$ is a linear combination of the data, i.e., there exists an $n \times m$ matrix $A^\dagger$ such that

$$\mathbf{a} = A^\dagger \mathbf{y}.$$

In terms of random variables, we have $\boldsymbol{A} = A^\dagger \boldsymbol{Y}$. In this case, the law of propagation of uncertainty can be applied directly to calculate

$$E(\boldsymbol{A}) = A^\dagger E(\boldsymbol{Y}), \quad \text{and} \quad V(\boldsymbol{A}) = A^\dagger V(\boldsymbol{Y}) \left(A^\dagger\right)^{\mathrm{T}}.$$

Note that these calculations only require us to know $E(\boldsymbol{Y})$ and $V(\boldsymbol{Y})$, not the precise form of the multivariate distribution of $\boldsymbol{Y}$.

For nonlinear estimators in which $\mathbf{a} = \mathcal{A}(\mathbf{y})$ is a nonlinear function of $\mathbf{y}$, it is possible to estimate the mean and variance of $\boldsymbol{A}$ using linearisation. Given the estimate $\mathbf{a}$ of $\mathbf{a}^*$, we determine the $n \times m$ sensitivity matrix $K = K(\mathbf{a})$ where

$$K_{ji} = \frac{\partial a_j}{\partial y_i}.$$

The expectation $E(\boldsymbol{A})$ is estimated by $\mathbf{a}$ and uncertainty matrix $V(\boldsymbol{A})$ associated with the estimates $\mathbf{a}$ by

$$K(\mathbf{a}) V(\boldsymbol{Y}) K^{\mathrm{T}}(\mathbf{a}).$$

Estimates of both $E(\boldsymbol{A})$ and $V(\boldsymbol{A})$ depend on the linearisation about $\mathbf{a}$ and can be misleading. An alternative is to use Monte Carlo methods to estimate $E(\boldsymbol{A})$ and $V(\boldsymbol{A})$ [71]. Suppose the model equations are

$$y_i = \phi(\mathbf{x}_i, \mathbf{a}^*) + \epsilon_i, \quad \epsilon_i \in N(0, \sigma^2), \quad i = 1, \ldots, m.$$

On the basis of measurement data $\mathbf{y}$, estimates $\mathbf{a} = \mathcal{A}(\mathbf{y})$ of $\mathbf{a}^*$ have been obtained. For $q = 1, \ldots, M$, we generate data vectors $\mathbf{y}_q = (y_{1,q}, \ldots, y_{m,q})^{\mathrm{T}}$, where

$$y_{i,q} = \phi(\mathbf{x}_i, \mathbf{a}) + \epsilon_{i,q}, \quad \epsilon_{i,q} \in N(0, \sigma^2), \qquad i = 1, \ldots, m$$

with $\epsilon_{i,q}$ drawn at random from the normal distribution $N(0, \sigma^2)$. The estimator is applied to the data vectors $\mathbf{y}_q$ to determine estimates $\mathbf{a}_q = \mathcal{A}(\mathbf{y}_q)$, samples from the distribution associated with $\boldsymbol{A}$. Quantities such as $E(\boldsymbol{A})$ and $V(\boldsymbol{A})$ are therefore estimated by the sample mean and variance derived from $\{\mathbf{a}_q\}_{q=1}^M$. We note that, in this scheme, the variance matrix is estimated using simulations based on the observed estimate $\mathbf{a}$ rather than on $\mathbf{a}^*$ which, in general, will be unknown. The validity of the estimate of the variance matrix will depend on the closeness of the distributions at $\mathbf{a}$ and $\mathbf{a}^*$, an issue that is likewise difficult to assess.

Both types of calculation, the linearised propagation of uncertainties and the Monte Carlo simulation, depend on the estimate $\mathbf{a} = \mathcal{A}(\mathbf{y})$ provided by the estimator in order perform the linearisation or Monte Carlo simulation. Both calculations are therefore concerned with the

propagation of uncertainty associated with the data through to those associated parameter estimates, for fixed $\mathbf{a}$, and need to be interpreted in this context.

From estimates of $E(\boldsymbol{A})$ and $V(\boldsymbol{A})$ it is possible to measure the behaviour of one estimation method relative to another.

3.4 Maximising the likelihood

The calculation of $E(\boldsymbol{A})$ and $V(\boldsymbol{A})$ for estimators $\mathbf{a} = \mathcal{A}(\mathbf{y})$ provides the tools to select an estimation method from a range of options. In practice, we do not want to perform an extensive analysis of possible estimation methods before analysing a set of experimental data. Instead we can be guided by the statistical model associated with the data to define a preferred estimation method. Maximum likelihood estimation uses the fact that in a complete statement of a model, the deviations ϵ_i are modelled as belonging to statistical distributions defined in terms of probability density functions (section 2.4.6). These distributions can be used to define a likelihood function. For example, suppose the measurement model is of the form

$$Y_i = \phi(\mathbf{x}_i, \mathbf{a}) + E_i,$$

where $\boldsymbol{E} = (E_1, \ldots, E_m)^{\mathrm{T}}$ has multivariate PDF $g(\boldsymbol{\xi})$. Let

$$\boldsymbol{\phi}(\mathbf{a}) = (\phi(\mathbf{x}_1, \mathbf{a}), \ldots, \phi(\mathbf{x}_m, \mathbf{a}))^{\mathrm{T}},$$

a vector function of $\mathbf{a}$. The probability $p(\mathbf{y}|\mathbf{a})$ of observing the data $\mathbf{y} \in \boldsymbol{Y}$ given that the model is specified by parameters $\mathbf{a}$ is given by $p(\mathbf{y}|\mathbf{a}) = g(\mathbf{y} - \boldsymbol{\phi}(\mathbf{a}))$, which we can regard as a function of $\mathbf{a}$. The maximum likelihood estimate of $\mathbf{a}$ is that which maximises $p(\mathbf{y}|\mathbf{a})$, i.e., that which provides the most probable[1] explanation of the data $\mathbf{y}$. Maximum likelihood estimates enjoy favourable properties with respect to bias and statistical efficiency and usually represent an appropriate method for determining parameter estimates. Many standard parameter estimation methods can be formulated as maximum likelihood estimation for particular statistical models for the random effects.

Example: standard experiment and least-squares approximation

In the standard experiment, the model equations are of the form

$$y_i = \phi(\mathbf{x}_i, \mathbf{a}) + \epsilon_i, \quad i = 1, \ldots, m, \ \boldsymbol{\epsilon} \in N(0, \sigma^2 I).$$

Regarding $f_i = y_i - \phi(\mathbf{x}_i, \mathbf{a})$ as having the probability density function specified for ϵ_i, the associated likelihood function is proportional to

$$\prod_{i=1}^{m} \exp\left\{-\frac{1}{2}\frac{f_i^2}{\sigma^2}\right\} = \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{m} f_i^2\right\},$$

so that the likelihood is maximised when

$$\sum_{i=1}^{m} f_i^2 = \sum_{i=1}^{m} (y_i - \phi(\mathbf{x}_i, \mathbf{a}))^2$$

[1] We speak of maximising the 'likelihood' rather than maximising the 'probability' because the function $p(\mathbf{y}|\mathbf{a})$ regarded as a function of $\mathbf{a}$ is not a probability density; it is only a probability density with respect to the variables $\mathbf{y}$. Sometimes the notation $l(\mathbf{a}; \mathbf{y})$ is used for $p(\mathbf{y}|\mathbf{a})$ to emphasise this distinction.

is minimised with respect to $\mathbf{a}$. ♯

The importance of least-squares estimation derives from the fact that it represents a maximum likelihood estimation for models subject to normally distributed random effects in the response variable. For linear models, it can be shown that it is unbiased and optimally efficient; see section 4.10.

Example: uniform distributions and Chebyshev approximation

Suppose, in an experiment, the model equations are of the form

$$y_i = \phi(\mathbf{x}_i, \mathbf{a}) + \epsilon_i,$$

where $\epsilon_i \in R(-S, S)$ are modelled as belonging to a rectangular distribution specified by the interval $[-S, S]$. This situation can arise, for example, if the measurements y_i are read from a digital indicating device, in which case S is half the last displayed unit. If all other random effects are negligible, a uniform distribution is appropriate. Regarding $f_i = y_i - \phi(\mathbf{x}_i, \mathbf{a})$ as having the probability density function specified for ϵ_i, the associated likelihood function is (proportional to) 1 if $|f_i| \leq S$, $i = 1, \ldots, m$, and 0 otherwise. The likelihood is maximised by any $\mathbf{a}$ such that

$$\max_i |f_i| = \max_i |y_i - \phi(\mathbf{x}_i, \mathbf{a}))| \leq S.$$

Such an $\mathbf{a}$, if it exists, can be found by solving the L_∞ (i.e., Chebyshev or *minimax*) optimisation problem

$$\min_{\mathbf{a}} \max_i |y_i - \phi(\mathbf{x}_i, \mathbf{a}))|.$$

In this way we can think of Chebyshev approximation as a maximum likelihood estimator for uniformly distributed random effects. ♯

Example: exponential power distributions and p-norms

Just as least-squares and Chebyshev approximation correspond to maximum likelihood estimation associated with Gaussian and rectangular sampling distributions, respectively, approximation in a p-norm (section 3.2) corresponds to an exponential power distribution (see e.g., [32, section 3.2.1]) with PDF

$$g(x) = \frac{K}{\alpha_3} \exp\left\{ -\frac{1}{2} \left| \frac{x - \alpha_1}{\alpha_3} \right|^{2/(1+\alpha_2)} \right\},$$

where $-\infty < \alpha_1 < \infty$, $-1 < \alpha_2 \leq 1$, is such that $p = 2/(1+\alpha_2)$, $\alpha_3 > 0$, and the normalising constant is given by

$$K^{-1} = \Gamma\left(1 + \frac{1+\alpha_2}{2}\right) 2^{1+(1+\alpha_2)/2}.$$

The parameter α_2 controls the kurtosis or 'peakedness' of the distribution. The value of $\alpha_2 = 0$ gives the normal distribution, as α_2 approaches -1 the distribution becomes more rectangular, and towards $+1$ the peak becomes narrower. ♯

3.5 Bayesian parameter estimation

3.5.1 Description

Both least-squares and maximum-likelihood methods are based on a so-called classical approach to statistical inference. In this paradigm, the parameters $\mathbf{a}$ we are trying to determine are fixed but unknown. The measured data $\mathbf{y}$ are assumed to have been generated according to a statistical model whose behaviour depends on $\mathbf{a}$. On the basis of the measurements $\mathbf{y}$ estimates $\hat{\mathbf{a}} = \mathcal{A}(\mathbf{y})$ are found using some estimation method $\mathcal{A}$. These estimates are regarded as a sample from a vector of random variables $\boldsymbol{A}$ and the uncertainty associated with the estimate $\hat{\mathbf{a}}$ is determined from the distribution $p(\hat{\mathbf{a}}|\mathbf{a})$ associated with this random vector. For nonlinear problems, the shape of this distribution will depend on $\mathbf{a}$ and since $\mathbf{a}$ is unknown, the distribution for $\boldsymbol{A}$ is also unknown but can usually be approximated on the basis of the estimate $\hat{\mathbf{a}}$.

In a Bayesian formulation [32, 113, 151, 160, 193], knowledge about $\mathbf{a}$ is encoded in a probability distribution $p(\mathbf{a}|I)$ derived from the information I we have to hand. As more information is gathered through measurement experiments, for example, this distribution is updated.

In the context of data analysis, we assume a *prior* distribution $p(\mathbf{a})$ and that data $\mathbf{y}$ has been gathered according to a sampling distribution depending on $\mathbf{a}$ from which we can calculate the probability $p(\mathbf{y}|\mathbf{a})$ of observing $\mathbf{y}$ as in maximum likelihood estimation. Bayes' Theorem (2.1) states that the *posterior* distribution $p(\mathbf{a}|\mathbf{y})$ for $\mathbf{a}$ after observing $\mathbf{y}$, is related to the likelihood and the prior distribution by

$$p(\mathbf{a}|\mathbf{y}) = Kp(\mathbf{y}|\mathbf{a})p(\mathbf{a}), \tag{3.1}$$

where the constant K is chosen to ensure that the posterior distribution integrates to unity, i.e.,

$$\int p(\mathbf{a}|\mathbf{y})\,\mathrm{d}\mathbf{a} = 1.$$

In this form, Bayes' theorem says that the posterior distribution is the likelihood weighted by the prior distribution.

If we have little prior knowledge about $\mathbf{a}$, we may take for the prior PDF the improper distribution $p(\mathbf{a}) = 1$, in which case $p(\mathbf{a}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{a})$, i.e., the posterior distribution is proportional to the likelihood. Note that $p(\mathbf{y}|\mathbf{a})$ is a PDF with respect to $\mathbf{y}$ so that $\int p(\mathbf{y}|\mathbf{a})\,\mathrm{d}\mathbf{y} = 1$, whereas $p(\mathbf{a}|\mathbf{y})$ is a PDF with respect to $\mathbf{a}$ with $\int p(\mathbf{a}|\mathbf{y})\,\mathrm{d}\mathbf{a} = 1$.

Bayes' theorem (3.1) has the following geometrical interpretation in the context of model fitting. Suppose the model equations are

$$y_i = \phi(\mathbf{x}_i, \mathbf{a}) + \epsilon_i, \quad \epsilon_i \in N(0, \sigma^2), \quad i = 1, \ldots, m, \tag{3.2}$$

and let $\mathbf{a} \mapsto \boldsymbol{\phi}(\mathbf{a}) = (\phi(\mathbf{x}_1, \mathbf{a}), \ldots, \phi(\mathbf{x}_m, \mathbf{a}))^{\mathrm{T}}$ be the model surface defined in $\mathcal{R}^m$. For accurate data, the vector $\mathbf{y}$ is a point in $\mathcal{R}^m$ close to the surface $\boldsymbol{\phi}(\mathbf{a})$. The probability $p(\mathbf{y}|\mathbf{a})$ is such that

$$p(\mathbf{y}|\mathbf{a}) \propto \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{m}(y_i - \phi(\mathbf{x}_i, \mathbf{a}))^2\right\} = \exp\left\{-\frac{1}{2\sigma^2}d^2(\mathbf{y}, \boldsymbol{\phi}(\mathbf{a}))\right\},$$

where $d(\mathbf{y}, \boldsymbol{\phi}(\mathbf{a})) = \|\mathbf{y} - \boldsymbol{\phi}(\mathbf{a})\|$ is the Euclidean distance from $\boldsymbol{\phi}(\mathbf{a})$ to $\mathbf{y}$. Therefore, for fixed $\mathbf{a}$, the probability $p(\mathbf{y}|\mathbf{a})$ is a function of the distance from $\boldsymbol{\phi}(\mathbf{a})$ to $\mathbf{y}$. If the prior distribution for $\mathbf{a}$ is constant, Bayes' theorem states that $p(\mathbf{a}|\mathbf{y})$ is proportional to $p(\mathbf{y}|\mathbf{a})$ so that, for fixed $\mathbf{y}$, the probability $p(\mathbf{a}|\mathbf{y})$ is a function of the distance from $\mathbf{y}$ to the point $\boldsymbol{\phi}(\mathbf{a})$ on the surface specified by $\mathbf{a}$. As a function of $\mathbf{y}$ with $\mathbf{a}$ fixed, $p(\mathbf{y}|\mathbf{a})$ describes a simple, multivariate normal distribution; as a function of $\mathbf{a}$, $p(\mathbf{a}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{a})$ depends on the geometry of the surface $\boldsymbol{\phi}(\mathbf{a})$. If $\boldsymbol{\phi}(\mathbf{a})$ happens to be a linear surface (a hyperplane) then $p(\mathbf{a}|\mathbf{y})$ is also a multivariate normal distribution; see section 4.1.4. For surfaces that are reasonably linear, the distance function $d^2(\mathbf{y}, \boldsymbol{\phi}(\mathbf{a}))$ will be close to a quadratic function and, consequently, $p(\mathbf{a}|\mathbf{y})$ will be approximately Gaussian.

The two distributions $p(\mathbf{y}|\mathbf{a})$ and $p(\mathbf{a}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{a})p(\mathbf{a})$ can also be related to two types of Monte Carlo calculations which we term *forward* and *inverse*. Suppose the measurement model is as in (3.2). For $q = 1, \ldots, N$, and $\mathbf{a}$ fixed, we generate data vectors $\mathbf{y}_q = (y_{1,q}, \ldots, y_{m,q})^{\mathrm{T}}$, where

$$y_{i,q} = \phi(\mathbf{x}_i, \mathbf{a}) + \epsilon_{i,q}, \quad \epsilon_{i,q} \in N(0, \sigma^2), \qquad i = 1, \ldots, m$$

with $\epsilon_{i,q}$ drawn at random from the normal distribution $N(0, \sigma_i^2)$. Then the $\mathbf{y}_q$ are samples from the distribution with PDF $p(\mathbf{y}|\mathbf{a})$. This forward type of MC calculations corresponds to repeating a set of measurements of the same artefact and noting the spread of the results. Geometrically, forward MC calculations produce a Gaussian scatter of points $\mathbf{y}_q$ centred around the point $\boldsymbol{\phi}(\mathbf{a})$.

Inverse Monte Carlo calculations determine (in an inefficient way, see section 4.11) a set of samples $\mathbf{a}_q$ from the distribution whose PDF is $p(\mathbf{a}|\mathbf{y})$. For $q = 1, \ldots, N$, prior distribution $p(\mathbf{a})$ and $\mathbf{y}$ fixed, we first sample $\mathbf{a}_q$ from the distribution $p(\mathbf{a})$. (If there is no substantive prior information then it is usually appropriate to sample $\mathbf{a}_q$ from a uniform distribution that covers all values of $\mathbf{a}$ that could occur in practice.) For each q, we generated a data vector $\mathbf{y}_q$ where

$$y_{i,q} = \phi(\mathbf{x}_i, \mathbf{a}_q) + \epsilon_{i,q}, \quad \epsilon_{i,q} \in N(0, \sigma^2), \quad i = 1, \ldots, m.$$

Note that in contrast to the forward MC calculations, here $\mathbf{a}_q$ varies from data vector to data vector. We then compare $\mathbf{y}_q$ with $\mathbf{y}$ and note the indices q from which $\mathbf{y}_q$ is close to $\mathbf{y}$, relative to some tolerance $\tau_{\mathbf{y}}$. The set $\{\mathbf{a}_q : \|\mathbf{y}_q - \mathbf{y}\| < \tau_{\mathbf{y}}\}$ is a sample from the distribution with PDF $p(\mathbf{a}|\mathbf{y})$ (in the limit as $N \to \infty$ and $\tau \to 0$). This type of MC calculations corresponds to obtaining measured data $\mathbf{y}_q$ for a range of artefacts drawn from the distribution $p(\mathbf{a})$, previously calibrated by a more accurate measurement system so that the parameter values $\mathbf{a}_q$ are known. The distribution $p(\mathbf{a}|\mathbf{y})$ is the frequency distribution for the parameter values $\mathbf{a}$ corresponding to artefacts whose measurement results $\mathbf{y}_q$ are close to $\mathbf{y}$. Geometrically, inverse MC produces a Gaussian scatter of points in a band around the surface $\boldsymbol{\phi}(\mathbf{a})$ in the region of interest The scatter of values of $\mathbf{a}$ are given by those parameter values $\mathbf{a}_q$ for which $\boldsymbol{\phi}(\mathbf{a}_q)$ was perturbed by $\boldsymbol{\epsilon}_q$ to be close to $\mathbf{y}$ (as measured by $S^2(\mathbf{y}_q, \mathbf{y})$, for example).

The conditional distributions $p(\mathbf{a}|\mathbf{y})$ and $p(\mathbf{y}|\mathbf{a})$ are linked through the joint distribution $p(\mathbf{a}, \mathbf{y})$ which can be factored in two ways

$$p(\mathbf{a}, \mathbf{y}) = p(\mathbf{y}|\mathbf{a})p(\mathbf{a}) \quad \text{and} p(\mathbf{a}, \mathbf{y}) = p(\mathbf{a}|\mathbf{y})p(\mathbf{y}).$$

Inverse Monte Carlo in fact produces a sample $\{(\mathbf{a}_q, \mathbf{y}_q)\}_{q=1}^{M}$ from this joint distribution, generated using the first factorisation. For a suitable tolerance $\tau_{\mathbf{a}}$, the subsample $\{\mathbf{y}_q :$

$\|\mathbf{a}_q - \mathbf{a}\| < \tau_{\mathbf{a}}\}$ is approximately a sample from the conditional distribution $p(\mathbf{y}|\mathbf{a})$ and $\{\mathbf{a}_q : \|\mathbf{y}_q - \mathbf{y}\| < \tau_{\mathbf{y}}\}$ is approximately a sample from $p(\mathbf{a}|\mathbf{y})$.

See section 3.10 for an example of forward and inverse MC calculations.

3.5.2 Parameter estimates and their associated uncertainties

The posterior distribution $p(\mathbf{a}|by)$ represents all the information about $\mathbf{a}$ taking into account the measurement data $\mathbf{y}$ and the prior information. In practice, summary information about this distribution is required and in metrology it is usual to provide parameter estimates along with associated uncertainties. Ideally, this would be in the form of the mean $\bar{\mathbf{a}}$ and variance V of the posterior distribution given by

$$\bar{a}_j = \int a_j p(\mathbf{a}|\mathbf{y})\,\mathrm{d}d\mathbf{a}, \quad V_{jk} = \int (a_j - \bar{a}_j)(a_k - \bar{a}_k)p(\mathbf{a}|\mathbf{y})\,\mathrm{d}\mathbf{a}.$$

However, both these quantities require integration of multivariate functions and for problems involving even a modest number of parameters, 10 say, this integration is computationally expensive. For large problems it becomes impractical.

An alternative to providing estimates that require global knowledge of the distribution is to provide an approximation to the distribution on the basis of local knowledge. The main idea is to determine a quadratic approximation to the negative logarithm $F(\mathbf{a}) = -\log p(\mathbf{a}|\mathbf{y})$ of the posterior distribution in the neighbourhood of a suitable point $\hat{\mathbf{a}}$. Almost always, $\hat{\mathbf{a}}$ is taken as the maximum likelihood estimate. The two main reasons for this are i) the ML estimate can be determined using optimisation techniques and ii) the approximation will be most valid in the region of maximum probability. For the ML estimate,

$$F(\mathbf{a}) \approx F(\hat{\mathbf{a}}) + \frac{1}{2}(\mathbf{a} - \hat{\mathbf{a}})^{\mathrm{T}} H (\mathbf{a} - \hat{\mathbf{a}}), \tag{3.3}$$

where

$$H_{jk} = -\frac{\partial^2}{\partial\alpha_j \partial\alpha_k} \log p(\hat{\mathbf{a}}|\mathbf{y})$$

is the Hessian matrix of second partial derivatives of $-\log p(\mathbf{y}|\mathbf{a})$ evaluated at the minimum $\hat{\mathbf{a}}$. (The linear term in this approximation is absent since $\partial \log p(\mathbf{a}|\mathbf{y})/\partial\alpha_j = 0$ at $\mathbf{a} = \hat{\mathbf{a}}$.) Taking exponentials of (3.3), we approximate the posterior distribution by

$$p(\mathbf{a}|\mathbf{y}) \approx K \exp\left\{-\frac{1}{2}(\mathbf{a} - \hat{\mathbf{a}})^{\mathrm{T}} H (\mathbf{a} - \hat{\mathbf{a}})\right\},$$

where K is a normalising constant. Recognising this as a multivariate normal distribution, setting $V = H^{-1}$, we have

$$p(\mathbf{a}|\mathbf{y}) \approx \frac{1}{|2\pi V|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{a} - \hat{\mathbf{a}})^{\mathrm{T}} V^{-1} (\mathbf{a} - \hat{\mathbf{a}})\right\},$$

i.e., $\mathbf{a} \sim N(\hat{\mathbf{a}}, V)$. (The notation $|V|$ denotes the determinant of V.) This approach provides parameter estimates $\hat{\mathbf{a}}$ and associated uncertainty matrix V using standard nonlinear optimisation techniques. We note that we can determine these terms without knowing the constant of proportionality in (3.1).

As with most approximating methods, this approach has to be used with some care. The multivariate normal distribution is unimodal and symmetric. If the true posterior distribution is multimodal or skewed, then the approximation could well provide poor information. (There may also be numerical difficulties in implementing the approach in these circumstances.)

3.5.3 Estimators in a Bayesian context

As stated above, the posterior distribution $p(\mathbf{a}|\mathbf{y})$ reflects all the information about $\mathbf{a}$ available from the data $\mathbf{y}$ and the prior knowledge. Once the statistical model has been specified, the likelihood $p(\mathbf{y}|\mathbf{a})$ is defined and $p(\mathbf{a}|\mathbf{y})$ can be specified up to a multiplicative constant using Bayes' theorem (3.1). Statements about $\mathbf{a}$ such as expected values, coverage intervals, etc., are all calculated using $p(\mathbf{a}|\mathbf{y})$, but depend on the constant K. For many problems an accurate calculation of K is not straightforward so we may want to rely on estimation methods such as least squares that are more simple to implement. Maximum likelihood estimation can be seen as a general and practical method of defining an approximating Gaussian distribution $\hat{p}(\mathbf{a}|\mathbf{y})$ to $p(\mathbf{a}|\mathbf{y})$ and approximate inferences about $\mathbf{a}$ can be derived from $\hat{p}(\mathbf{a}|\mathbf{y})$ rather than $p(\mathbf{a}|\mathbf{y})$. These inferences will be more or less valid to the extent that $\hat{p}$ is a good or bad approximation to p. However, we can look at estimators another way. Suppose, for a given problem, we have to hand an estimation method $\hat{\mathbf{a}} = \mathcal{A}(\mathbf{y})$. What *valid* inferences about $\mathbf{a}$ can be made given that estimates $\hat{\mathbf{a}}$ have been determined? From the statistical model, it is possible to calculate the likelihood $p(\mathbf{y}|\mathbf{a})$. Assuming that $\hat{\mathbf{a}} = \mathcal{A}(\mathbf{y})$ is a deterministic function of $\mathbf{y}$, then

$$p(\hat{\mathbf{a}}|\mathbf{a}) = \int_{\{\mathbf{y}:\hat{\mathbf{a}}=\mathcal{A}(\mathbf{y})\}} p(\mathbf{y}|\mathbf{a})\,\mathrm{d}\mathbf{y},$$

is the probability of observing an estimate $\hat{\mathbf{a}}$ given that the parameters specifying the model is $\mathbf{a}$. Applying Bayes' theorem,

$$p(\mathbf{a}|\hat{\mathbf{a}}) \propto p(\hat{\mathbf{a}}|\mathbf{a})p(\mathbf{a})$$

is the distribution for $\mathbf{a}$ given that the estimator $\mathcal{A}$ has produced estimates $\hat{\mathbf{a}}$, based on the data $\mathbf{y}$. This distribution quantifies the information about $\mathbf{a}$ from the prior knowledge $p(\mathbf{a})$ and the result of the estimation method $\hat{\mathbf{a}} = \mathcal{A}(\mathbf{y})$. Inferences about $\mathbf{a}$ based on $p(\mathbf{a}|\hat{\mathbf{a}})$ will, in general, not be as strong as those derived from $p(\mathbf{a}|\mathbf{y})$ but they will be valid as $p(\mathbf{a}|\hat{\mathbf{a}})$ represents a correct summary of the information available from the estimation method. Estimation methods can therefore be compared by the extent to which the associated posterior distribution $p(\mathbf{a}|\hat{\mathbf{a}})$ matches $p(\mathbf{a}|\mathbf{y})$; see sections 3.10 and 4.1.4.

The distributions $p(\hat{\mathbf{a}}|\mathbf{a})$ and $p(\mathbf{a}|\hat{\mathbf{a}})$ can be investigated through forward and inverse MC calculations; see section 3.10 for an example.

3.6 Parameter estimation as optimisation problems

Estimators are usually defined in terms of minimising an error function $F(\mathbf{a}|\mathbf{y})$ defined in terms of the data $\mathbf{y}$ and the parameters $\mathbf{a}$. These optimisation problems are generally solved

by determining a set of *optimality conditions* for the parameters $\mathbf{a}$ that must necessarily hold at the solution and then employing an algorithm designed to produce a solution satisfying these conditions. The following are some of the optimisation problems that are relevant to discrete modelling (in roughly decreasing order of importance in metrology) and for which mature and reliable algorithms and software implementations are available. Throughout, C is an $m \times n$ matrix, $m \geq n$, with rows $\mathbf{c}_i^{\mathrm{T}}$, $\mathbf{y} = (y_1, \ldots, y_m)^{\mathrm{T}}$ an m-vector of observations, and $\mathbf{a} = (a_1, \ldots, a_n)^{\mathrm{T}}$ a vector of optimisation parameters.

3.6.1 Linear least squares

Solve

$$\min_{\mathbf{a}} \sum_{i=1}^{m} (y_i - \mathbf{c}_i^{\mathrm{T}}\mathbf{a})^2 = \sum_{i=1}^{m} (y_i - (c_{i1}a_1 + \ldots + c_{in}a_n))^2.$$

In matrix form, this problem is written as

$$\min_{\mathbf{a}} \|\mathbf{y} - C\mathbf{a}\|_2^2.$$

The matrix C is often referred to as the *observation matrix* or *design matrix*.

3.6.2 Nonlinear least squares

Given m functions $f_i(\mathbf{a})$ of parameters $\mathbf{a}$, solve

$$\min_{\mathbf{a}} \sum_{i=1}^{m} f_i^2(\mathbf{a}),$$

where the functions f_i usually depend on $\mathbf{y}$.

3.6.3 Linear least squares subject to linear equality constraints

Given C, $\mathbf{y}$, an $p \times n$ matrix D, $p < n$, and a p-vector $\mathbf{z}$, solve

$$\min_{\mathbf{a}} \|\mathbf{y} - C\mathbf{a}\|_2^2$$

subject to the constraints

$$D\mathbf{a} = \mathbf{z}.$$

3.6.4 Nonlinear least squares subject to linear equality constraints

Given m functions $f_i(\mathbf{a})$ of parameters $\mathbf{a}$, an $p \times n$ matrix D, $p < n$ and a p-vector $\mathbf{z}$, solve

$$\min_{\mathbf{a}} \sum_{i=1}^{m} f_i^2(\mathbf{a})$$

(where the functions f_i usually depend on $\mathbf{y}$), subject to the constraints

$$D\mathbf{a} = \mathbf{z}.$$

3.6.5 Linear L_1

Given C and $\mathbf{y}$, solve

$$\min_{\mathbf{a}} \|\mathbf{y} - C\mathbf{a}\|_1 \equiv \min_{\mathbf{a}} \sum_{i=1}^{m} |y_i - \mathbf{c}_i^{\mathrm{T}}\mathbf{a}|.$$

3.6.6 Linear Chebyshev (L_∞)

Given C and $\mathbf{y}$, solve

$$\min_{\mathbf{a}} \|\mathbf{y} - C\mathbf{a}\|_\infty \equiv \min_{\mathbf{a}} \max_i |y_i - \mathbf{c}_i^{\mathrm{T}}\mathbf{a}|.$$

3.6.7 Linear programming

Given n-vectors $\mathbf{c}$ and $\mathbf{d}_i$, $i = 1, \ldots, m$, and m-vector $\mathbf{y}$, solve

$$\min_{\mathbf{a}} \ \mathbf{c}^{\mathrm{T}}\mathbf{a}$$

subject to the linear inequality constraints

$$\mathbf{d}_i^{\mathrm{T}}\mathbf{a} \geq y_i, \quad i = 1, \ldots, m.$$

3.6.8 Unconstrained minimisation

Given a function $F(\mathbf{a})$ of parameters $\mathbf{a}$, solve

$$\min_{\mathbf{a}} \ F(\mathbf{a}).$$

3.6.9 Nonlinear Chebyshev (L_∞)

Given m functions $f_i(\mathbf{a})$ of parameters $\mathbf{a}$, solve

$$\min_{\mathbf{a}} \ \max_i |f_i(\mathbf{a})|,$$

where the functions f_i usually depend on $\mathbf{y}$.

This problem can be re-formulated as

$$\min_{\mathbf{a},s} \ s$$

subject to the constraints

$$-s \leq f_i(\mathbf{a}) \leq s, \quad i = 1, \ldots, m.$$

This is a special case of the following optimisation problem.

3.6.10 Mathematical programming

Given functions $F(\mathbf{a})$ and $g_k(\mathbf{a})$, $k = 1, \ldots, K$, of parameters $\mathbf{a}$, n-vectors $\mathbf{d}_i$, $i = 1, \ldots, m$, and m-vector $\mathbf{y}$, solve

$$\min_{\mathbf{a}} \; F(\mathbf{a})$$

subject to the linear constraints

$$\mathbf{d}_i^{\mathrm{T}}\mathbf{a} \geq y_i, \quad i = 1, \ldots, m$$

and nonlinear constraints

$$g_k(\mathbf{a}) \geq 0, \quad k = 1, \ldots, K.$$

3.7 Minimisation of a function of several variables

Let $F(\mathbf{a})$ be a general (smooth) function of n variables $\mathbf{a} = (a_1, \ldots, a_n)^{\mathrm{T}}$: F is the *objective function* of the minimisation problem.

Let $\mathbf{g} = \mathbf{g}(\mathbf{a})$ be the *gradient* of F, with components $g_j = \partial F/\partial a_j$, and H the *Hessian matrix* of second partial derivatives,

$$H_{jk} = \partial^2 F/\partial a_j \partial a_k.$$

At a minimum $\mathbf{a}^*$ of F, $\mathbf{g}(\mathbf{a}^*) = \mathbf{0}$. If $\mathbf{a}$ is an approximate solution we wish to find a step $\mathbf{p}$ such that $\mathbf{g}(\mathbf{a} + \mathbf{p}) = \mathbf{0}$. To first order,

$$\mathbf{g}(\mathbf{a} + \mathbf{p}) = \mathbf{g} + H\mathbf{p},$$

suggesting that $\mathbf{p}$ should be chosen so that

$$H\mathbf{p} = -\mathbf{g}. \tag{3.4}$$

In the Newton algorithm, an estimate of the solution $\mathbf{a}$ is updated according to $\mathbf{a} := \mathbf{a} + t\mathbf{p}$, where $\mathbf{p}$ solves (3.4) and t is a step length chosen to ensure a sufficient decrease in F. Near the solution, the Newton algorithm converges quadratically, i.e., if at the kth iteration the distance of the current estimate $\mathbf{a}_k$ from the solution $\mathbf{a}^*$ is $\|\mathbf{a}_k - \mathbf{a}^*\|$, then the distance of the subsequent estimate $\mathbf{a}_{k+1}$ from the solution is $\|\mathbf{a}_{k+1} - \mathbf{a}^*\| = O(\|\mathbf{a}_k - \mathbf{a}^*\|^2)$, so that the distance to the solution is squared approximately at each iteration.

3.7.1 Nonlinear least squares

For nonlinear least-squares problems, the objective function is of the form[2]

$$F(\mathbf{a}) = \frac{1}{2}\sum_{i=1}^{m} f_i^2(\mathbf{a})$$

[2]The fraction $\frac{1}{2}$ is sometimes included to simplify related expressions.

and has gradient

$$\mathbf{g} = J^{\mathrm{T}}\mathbf{f},$$

where J is the Jacobian matrix

$$J_{ij} = \frac{\partial f_i}{\partial a_j}, \tag{3.5}$$

and Hessian matrix

$$H = J^{\mathrm{T}}J + G, \qquad G_{jk} = \sum_{i=1}^{m} f_i \frac{\partial^2 f_i}{\partial a_j \partial a_k}.$$

3.7.2 Large scale optimisation

The main computational step in the Newton algorithm is the formulation and solution of the equations (3.4) for the search direction $\mathbf{p}$ which generally takes $O(n^3)$ operations where n is the number of parameters. For very large problems, this may not be feasible (usually because too much time is required).

The conjugate gradient approach [115] is one of the main tools in general purpose large scale optimisation, particularly because it requires only a few vectors to be stored. Suppose we wish to find the minimum of $F(\mathbf{a})$, given an initial estimate $\mathbf{a}_0$. For nonlinear problems, the algorithm takes the form

I Set $k = 0$, $\mathbf{g}_0 = \nabla_{\mathbf{a}} F(\mathbf{a}_0)$.

II While $\|\mathbf{g}_k\| > \tau$ (where $\tau > 0$ is a small constant),

i Set $k = k + 1$.

ii Determine a search direction. If $k = 1$ set $\mathbf{p}_1 = -\mathbf{g}_0$. If k is a multiple of n, set $\mathbf{p}_k = -\mathbf{g}_{k-1}$. Otherwise, set

$$\beta_k = \|\mathbf{g}_{k-1}\|^2 / \|\mathbf{g}_{k-2}\|^2, \quad \mathbf{p}_k = -\mathbf{g}_{k-1} + \beta_k \mathbf{p}_{k-1}.$$

iii Determine the step length. Find α_k to minimise $F(\mathbf{a}_{k-1} + \alpha_k \mathbf{p}_k)$.

iv Update

$$\mathbf{a}_k = \mathbf{a}_{k-1} + \alpha_k \mathbf{p}_k, \quad \mathbf{g}_k = \nabla_{\mathbf{a}} F(\mathbf{a}_k).$$

III Set $\mathbf{a} = \mathbf{a}_k$ and finish.

There has been much research in developing efficient, large-scale optimisation algorithms; see e.g., [50, 163, 212]. One of the main approaches is to use a limited memory quasi-Newton algorithm [115, section 4.8]. In a quasi-Newton algorithm, the update step (3.4) is determined from an approximation to the Hessian matrix H of second partial derivatives of the objective function $F(\mathbf{a})$ or its inverse. Starting from the identity matrix, this approximation is built up from successive estimates $\mathbf{g}_k$ of the function gradients. If F is a quadratic function of n parameters, then after n steps the approximation to the Hessian is exact (in exact arithmetic). For large n, memory and computation constraints may prohibit any attempt to approximate H. Instead, the Hessian matrix is approximated by a limited number of quasi-Newton updates and can be stored by a correspondingly limited number of n-vectors.

3.8 Problem conditioning

The numerical accuracy of the solution parameters **a** will depend on the conditioning of the problem. A problem is *well-conditioned* if a small change in the data corresponds to a small change in the solution parameters, and conversely.

3.8.1 Condition of a matrix, orthogonal factorisation and the SVD

The condition of a discrete modelling problem can usually be analysed in terms of the condition of a matrix associated with the problem, for example, the observation matrix for linear least-squares problems or the Jacobian matrix for nonlinear problems.

An $m \times n$ matrix Q is *orthogonal* if $Q^{\mathrm{T}}Q = I$, the $n \times n$ identity matrix. If $m = n$ then we have in addition $QQ^{\mathrm{T}} = I$. Any two columns $\mathbf{q}_j$, $\mathbf{q}_k$, $j \neq k$, of an orthogonal matrix are at right angles to each other in the sense that $\mathbf{q}_j^{\mathrm{T}}\mathbf{q}_k = 0$. Orthogonal matrices have the property of preserving the Euclidean (2-norm) length of a vector: $\|Q\mathbf{x}\| = \|\mathbf{x}\|$.

Given two vectors $\mathbf{x} = (x_1, x_2, x_3)^{\mathrm{T}}$ and $\mathbf{y} = (y_1, y_2, y_3)^{\mathrm{T}}$ in $\mathbb{R}^3$, they can be rotated by a rotation matrix Q so that one lies along the x-axis and one lies in the xy-plane:

$$Q^{\mathrm{T}} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} \\ 0 & r_{22} \\ 0 & 0 \end{bmatrix} \text{ or } \quad \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{bmatrix} = Q \begin{bmatrix} r_{11} & r_{12} \\ 0 & r_{22} \\ 0 & 0 \end{bmatrix},$$

expressing the matrix with columns $\mathbf{x}$ and $\mathbf{y}$ as a product of an orthogonal matrix and an upper-triangular matrix. More generally, any $m \times n$ matrix C can be factored as

$$C = QR = [Q_1\ Q_2] \begin{bmatrix} R_1 \\ \mathbf{0} \end{bmatrix} = Q_1 R_1, \tag{3.6}$$

where Q is $m \times m$ orthogonal, Q_1 (Q_2) is the submatrix comprising the first n (last $m - n$) columns of Q, and R_1 is $n \times n$ upper triangular. Any $m \times n$ matrix C can also be factored as the product

$$C = USV^{\mathrm{T}} = [U_1\ U_2] \begin{bmatrix} S_1 \\ \mathbf{0} \end{bmatrix} V^{\mathrm{T}} = U_1 S_1 V^{\mathrm{T}}, \tag{3.7}$$

where U is an $m \times m$ orthogonal matrix, U_1 (U_2) is the submatrix comprising the first n (last $m - n$) columns of U, S_1 an $n \times n$ diagonal matrix with diagonal entries $s_1 \geq s_2 \geq \ldots \geq s_n \geq 0$, and V an $n \times n$ orthogonal matrix. This factorisation is known as the singular value decomposition (SVD). The columns of U (V) are the *left (right) singular vectors* and the s_j are known as the *singular values*.

The SVD shows that C maps the orthonormal vectors $\mathbf{v}_j$ onto the vectors $s_j\mathbf{u}_j$. If C has singular values all equal to one then it is an orthogonal matrix. The matrix C is full rank if and only if $s_n > 0$.

The ratio $\kappa = s_1/s_n$ of the largest singular value of a matrix to the smallest is known as the *condition number* of the matrix. There are high quality public domain software implementations of reliable algorithms to determine the SVD [83, 192].

If $C = USV^{\mathrm{T}}$ then the eigenvalue decomposition of $C^{\mathrm{T}}C$ is given by

$$C^{\mathrm{T}}C = VS^2V^{\mathrm{T}},$$

showing that the eigenvalues λ_j of $C^{\mathrm{T}}C$ are the squares of the singular values of C: $\lambda_j = s_j^2$ and the eigenvectors of $C^{\mathrm{T}}C$ are precisely the right singular vectors of C.

The singular values have a geometrical interpretation. The matrix C maps the unit sphere $\{\mathbf{x} : \|\mathbf{x}\| = 1\}$ in $\mathcal{R}^n$ into a hyper-ellipsoid in $\mathcal{R}^m$. The singular values are the lengths of the semi-axes of the ellipsoid. In particular, the largest singular value s_1 is such that

$$s_1 = \|C\mathbf{v}_1\| = \max_{|\mathbf{v}|=1} \|C\mathbf{v}\|,$$

and the smallest s_n such that

$$s_n = \|C\mathbf{v}_n\| = \min_{|\mathbf{v}|=1} \|C\mathbf{v}\|. \tag{3.8}$$

The condition number is the ratio of the length of the largest semi-axis to that of the smallest. An ill-conditioned matrix is one which maps the sphere into a long thin ellipsoid. Orthogonal matrices map the unit sphere to a unit sphere.

The unwelcome numerical consequences of working with ill-conditioned matrices are due to the fact that computation will involve relatively large numbers leading to cancellation errors. The value of orthogonal matrices is that no large numbers are introduced unnecessarily into the computations.

The conditioning of a problem depends on the parameterisation of the model. Often, the key to being able to determine accurate solution parameters is in finding an appropriate parameterisation.

Example: basis vectors for $\mathcal{R}^3$

Suppose we take as basis vectors for three dimensional space $\mathcal{R}^3$ the vectors $\mathbf{e}_1 = (1, 0, 0)^{\mathrm{T}}$, $\mathbf{e}_2 = (1, 0.001, 0)^{\mathrm{T}}$ and $\mathbf{e}_3 = (1, 0, 0.001)^{\mathrm{T}}$. Any point in $\mathbf{y}$ in $\mathcal{R}^3$ can be written as a linear combination

$$\mathbf{y} = a_1\mathbf{e}_1 + a_2\mathbf{e}_2 + a_3\mathbf{e}_3.$$

For example,

$$\begin{aligned} (0.0, 1.0, 1.0)^{\mathrm{T}} &= -2000\mathbf{e}_1 + 1000\mathbf{e}_2 + 1000\mathbf{e}_3, \\ (0.0, 1.1, 1.1)^{T} &= -2200\mathbf{e}_1 + 1100\mathbf{e}_2 + 1100\mathbf{e}_3, \end{aligned}$$

showing that a change of the order of 0.1 in the point $\mathbf{y}$ requires a change of order 100 in the parameter values $\mathbf{a}$. This type of ill-conditioning means that up to three significant figures of accuracy could be lost using these basis vectors.

If $E = [\ \mathbf{e}_1\ \mathbf{e}_2\ \mathbf{e}_3\ ]$, the orthogonal factorisation of $E = QR$ produces the standard basis vectors $\mathbf{q}_1 = (1, 0, 0)^{\mathrm{T}}$, $\mathbf{q}_2 = (0, 1, 0)$ and $\mathbf{q}_3 = (0, 0, 1)^{\mathrm{T}}$ from the columns of Q. In many situations, an analysis using QR factorisations can lead a better choice of basis vectors (or functions). ♯

3.9 Numerical stability of algorithms

One factor affecting the numerical accuracy of the parameter estimates is the conditioning of the problem. A second is the numerical stability of the algorithm used to solve the

computational problem associated with finding the parameter estimates. A numerically stable algorithm is one that introduces no unnecessary additional ill-conditioning into a problem. Many of the numerical difficulties in solving computational problems arise because the calculations introduce large numbers leading to large cancellation errors. A very simple example is the calculation of the difference of two squares $c = a^2 - b^2$. If $a = 101$ and $b = 100$, then $c = 201$; all three numbers are of the order of 100. If we calculate a^2 and b^2, we introduce numbers of the order of 10^4. If instead we calculate $a - b$ and $a + b$ and set

$$c = (a - b)(a + b),$$

all the intermediate quantities remain of the order of 100 or smaller. A floating-point error analysis shows that the latter method is numerically superior. The calculation of a^2 and b^2 can also lead to overflow problems.

Analysing the stability of an algorithm generally requires a specialist in numerical analysis. Many of the algorithms implemented in high quality library numerical software have a supporting error analysis demonstrating their favourable behaviour (which is why the algorithms appear in the library in the first place).

Issues concerning the numerical stability of algorithms are covered in the companion best-practice guide on *Numerical analysis for algorithm design in metrology* [66].

3.10 Conceptual example

In order to illustrate some of the concepts in this chapter, we will consider a simple example. The example is somewhat artificial but it is simple enough to allow a thorough analysis yet sufficiently complex to illustrate many of the important features of experimental data analysis. In particular, it has a nonlinear element that allows many issues that do not arise for linear models to be explored.

3.10.1 Measurement model

The example involves the determination of the value of a single parameter a. The measurement information about a comes from two sources. The first y_1 is a measurement of a directly, the second y_2 from a measurement of a^3. The measurement model is of the form

$$y_1 = \phi_1(a) + \epsilon_1, \quad \phi_1(a) = a, \quad y_2 = \phi_2(a) + \epsilon_2, \quad \phi_2(a) = a^3. \tag{3.9}$$

The model space $\boldsymbol{\phi}(a)$ is defined by the curve $a \mapsto (a, a^3)$ in the plane $\mathcal{R}^2$; the measurements $\mathbf{y} = (y_1, y_2)^{\mathrm{T}}$ define a point in $\mathcal{R}^2$, hopefully close to the curve. See Figure 3.1.

3.10.2 Statistical model associated with the measurement data

We model the perturbatory effects ϵ_1 and ϵ_2 as being associated with normal distributions:

$$\epsilon_k \in N(0, \sigma_k^2), \quad k = 1, 2.$$

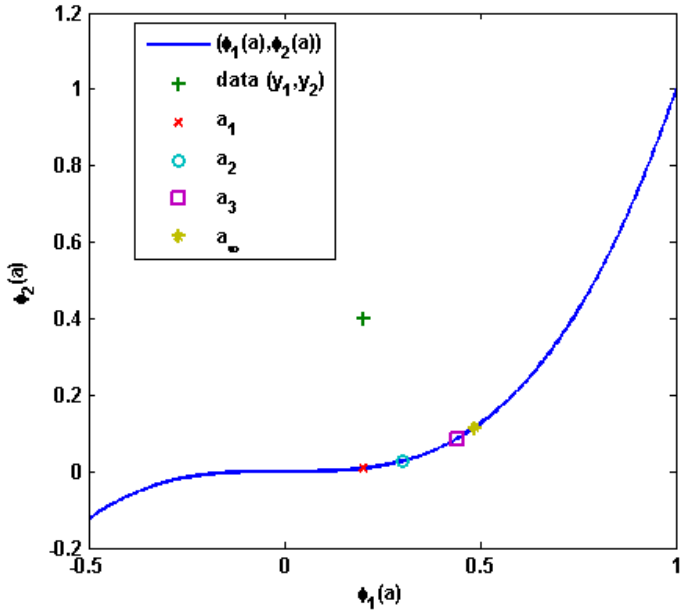

Figure 3.1: Data $\mathbf{y}$ and 4 estimates of a defined by approximation in the p-norm with $p = 1$, 2, 3 and ∞.

3.10.3 Approximation norms

Figure 3.1 shows the model curve (a, a^3) associated with the model (3.9) along with the data point $\mathbf{y} = (0.2, 0.4)^{\mathrm{T}}$, and four points on the curve determined by approximation in the p-norm, with $p = 1$, 2, 3 and ∞ (section 3.2). In each case, the estimate of a is determined by the point on the curve closest to $\mathbf{y}$ where the distance is calculated using the corresponding norm. These estimation methods do not explicitly depend on the statistical model associated with the measurement data.

Figure 3.2 graphs the corresponding functions $F_p(a)$, $p = 1$, 2, 3 and ∞ used to define the estimates. The F_1 and F_∞ functions have points of discontinuity in slope while F_2 and F_3 are smooth functions.

3.10.4 Four estimators

We define four parameter estimation methods $\hat{a}_k = \mathcal{A}_k(\mathbf{y})$ to determine estimates of a from the measurement data $\mathbf{y} = (y_1, y_2)^{\mathrm{T}}$. The first three are simple functions of $\mathbf{y}$:

$$\hat{a}_1 = y_1, \quad \hat{a}_2 = y_2^{1/3}, \quad \hat{a}_3 = (\hat{a}_1 + \hat{a}_2)/2 = (y_1 + y_2^{1/3})/2.$$

The fourth is the maximum likelihood estimator. From the statistical model, the probability of observing $\mathbf{y}$, given a, is such that

$$p(\mathbf{y}|a) \quad = \quad p(y_1|a)p(y_2|a) \propto \exp\left\{-\frac{(y_1 - a)^2}{2\sigma_1^2}\right\} \exp\left\{-\frac{(y_2 - a^3)^2}{2\sigma_2^2}\right\},$$

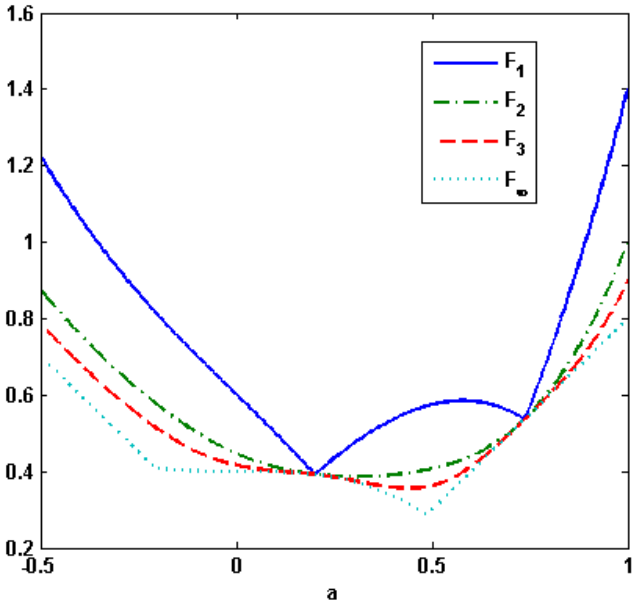

Figure 3.2: Error functions F_p for $p = 1$, 2, 3 and ∞ corresponding to data $\mathbf{y}$ in Figure 3.1.

$$= \quad \exp\left\{-\frac{1}{2}\left[\frac{(y_1-a)^2}{\sigma_1^2}+\frac{(y_2-a^3)^2}{\sigma_2^2}\right]\right\}.$$

The maximum of $p(\mathbf{y}|a)$ is attained by the $\hat{a}_4$ that minimises

$$F_4(a|\mathbf{y}) = \frac{1}{2}\left[\frac{(y_1-a)^2}{\sigma_1^2}+\frac{(y_2-a^3)^2}{\sigma_2^2}\right]. \tag{3.10}$$

At the minimum $g(a, y_1, y_2) = \partial F_4/\partial a = 0$, where

$$g(a, y_1, y_2) = \frac{a-y_1}{\sigma_1^2}+\frac{3a^2(a^3-y_2)}{\sigma_2^2}. \tag{3.11}$$

The function $g(a, y_1, y_2)$ is nonlinear in a and iterative techniques are required to find the solution (section 3.7). The Newton algorithm (with unit step length) in this case is

$$a := a - g/\dot{g}, \quad \dot{g} = \frac{\partial g}{\partial a} = \frac{1}{\sigma_1^2}+\frac{15a^4-6ay_2}{\sigma_2^2} = \frac{\sigma_2^2+\sigma_1^2(15a^4-6ay_2)}{\sigma_1^2\sigma_2^2}. \tag{3.12}$$

The first three estimators have a straightforward geometrical interpretation in terms of assigning a point on the curve (a, a^3) to the data point $\mathbf{y} = (y_1, y_2)^{\mathrm{T}}$: $\hat{a}_1$ defines the point on curve with the same x-value as y_1, $\hat{a}_2$ defines that with the same y-value as y_2 and $\hat{a}_3$ defines the point midway between $\hat{a}_1$ and $\hat{a}_2$ along the x-axis. For the case $\sigma_1 = \sigma_2$, $\hat{a}_4$ defines the point on the curve closest to $\mathbf{y}$. Figure 3.3 graphs the curve (a, a^3) along with the four points on the curve specified by the estimators $\mathcal{A}_k(\mathbf{y})$, $k = 1, 2, 3, 4$, for $\mathbf{y} = (0.2, 0.4)^{\mathrm{T}}$ and $\sigma_1 = \sigma_2$.

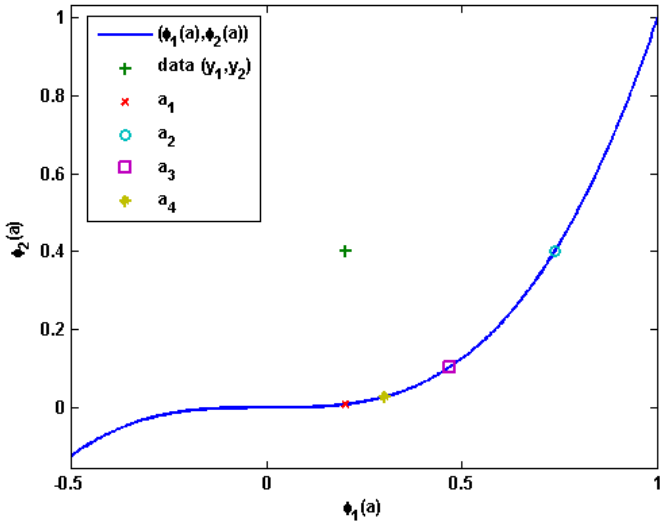

Figure 3.3: Four points on the curve specified by the estimators $\mathcal{A}_k(\mathbf{y})$, $k = 1, 2, 3, 4$, for $\mathbf{y} = (0.2, 0.4)^{\mathrm{T}}$ and $\sigma_1 = \sigma_2$.

3.10.5 Properties of the estimators

The properties of the estimators can be examined by calculating or estimating $p(\hat{a}_k|a)$, the distribution of the parameter estimates given that the parameter value is a. The distributions for the first three estimators can be determined analytically.

Details. We use the notation $p_{\hat{a}|a}(\xi)$, etc., to denote the PDF $p(\hat{a}|a)$ as function of the dummy variable ξ. Since

$$\hat{a}_1 = a + \epsilon_1, \quad \epsilon_1 \sim N(0, \sigma_1^2),$$

given a, $\hat{a}_1$ is associated with the normal distribution $\hat{a}_1 \sim N(a, \sigma_1^2)$ with PDF denoted by $p_{\hat{a}_1|a}(\xi)$. For the second estimator,

$$\hat{a}_2 = (a^3 + \epsilon_2)^{1/3}, \quad \epsilon_2 \sim N(0, \sigma_2^2).$$

This relationship defines $\hat{a}_2$ as a one-to-one function of ϵ_2. Applying rule (2.4), if $p_{\epsilon_2}(\xi)$ is the PDF for the normal distribution $N(0, \sigma_2^2)$ associated with ϵ_2, then the PDF associated with $\hat{a}_2$ is

$$p_{\hat{a}_2|a}(\xi) = 3\xi^2 p_{\epsilon_2}(\xi^3 - a^3).$$

Estimator $\hat{a}_3 = (\hat{a}_1 + \hat{a}_2)/2$ and its distribution is derived from those of $\hat{a}_1$ and $\hat{a}_2$:

$$p_{\hat{a}_1+\hat{a}_2|a}(\xi) = \int_{-\infty}^{\infty} p_{\hat{a}_1|a}(\xi - \zeta) p_{\hat{a}_2|a}(\zeta)\, d\zeta,$$

[120], and

$$p_{\hat{a}_3}(\xi) = 2p_{\hat{a}_1+\hat{a}_2|a}(2\xi) = 2\int_{-\infty}^{\infty} p_{\hat{a}_1|a}(2\xi - \zeta) p_{\hat{a}_2|a}(\zeta)\, d\zeta,$$

using (2.4).

More straightforwardly we can use (forward) Monte Carlo simulation to estimate $p(\hat{a}_3|a)$. For $q = 1, \ldots, M$, and a fixed, sample

$$y_{1,q} = a + \epsilon_{1,q}, \quad \epsilon_{1,q} \in N(0, \sigma_1^2), \quad y_{2,q} = a^3 + \epsilon_{2,q}, \quad \epsilon_{2,q} \in N(0, \sigma_2^2), \tag{3.13}$$

and set

$$\hat{a}_{1,q} = y_{1,q}, \quad \hat{a}_{2,q} = (y_{2,q})^{1/3}, \quad \hat{a}_{3,q} = (\hat{a}_{1,q} + \hat{a}_{2,q})/2.$$

Similarly, given $\mathbf{y}_q = (y_{1,q}, y_{2,q})^{\mathrm{T}}$, the maximum likelihood estimate $\hat{a}_{4,q}$ can be determined by minimising $F_4(a|\mathbf{y}_q)$ defined in (3.10).

The distribution $p(\hat{a}_4|a)$ can also be estimated by calculating the sensitivity of $\hat{a}_4 = \mathcal{A}_4(\mathbf{y})$ with respect to the data $\mathbf{y}$ (section 3.3).

Details. The equation $g(a, y_1, y_2) = 0$, where $g(a, y_1, y_2)$ is defined in (3.11), defines $\hat{a}_4 = \mathcal{A}_4(\mathbf{y})$ implicitly as a function of $\mathbf{y}$. Differentiating the equation $g(\hat{a}_4(\mathbf{y}), y_1, y_2) = 0$ with respect to y_k, we have

$$\frac{\partial g}{\partial a}\frac{\partial \hat{a}_4}{\partial y_k} + \frac{\partial g}{\partial y_k} = 0,$$

so that

$$\frac{\partial \hat{a}_4}{\partial y_k} = -\frac{1}{\dot{g}}\frac{\partial g}{\partial y_k}, \quad \dot{g} = \frac{\partial g}{\partial a}, \quad k = 1, 2,$$

gives the sensitivities of the estimate $\hat{a}_4$ with respect to the data $\mathbf{y}$. From (3.11), we have

$$\frac{\partial g}{\partial y_1} = -\frac{1}{\sigma_1^2}, \quad \frac{\partial g}{\partial y_2} = -\frac{3a^2}{\sigma_2^2}.$$

The partial derivative $\dot{g}$ of g with respect to a is given in (3.12). The uncertainty $u(\hat{a}_4)$ associated with the estimate $\hat{a}_4$, given a, is estimated by

$$\begin{aligned} u^2(\hat{a}_4) &= \frac{1}{\dot{g}^2}\left\{\left(\frac{1}{\sigma_1^2}\right)^2\sigma_1^2 + \left(\frac{3a^2}{\sigma_2^2}\right)^2\sigma_2^2\right\} \\ &= \frac{\sigma_1^2\sigma_2^2(\sigma_2^2 + 9\sigma_1^2a^4)}{(\sigma_2^2 + \sigma_1^2(15a^4 - 6ay_2))^2}, \quad a = \hat{a}_4. \end{aligned} \tag{3.14}$$

If $\hat{a}_4 = 0$, $u(\hat{a}_4) = \sigma_1$, the same as $u(\hat{a}_1)$. If $\sigma_1 = \sigma_2 = \sigma$, this expression simplifies to

$$u^2(\hat{a}_4) = \sigma^2\frac{1 + 9a^4}{(1 + 15a^4 - 6ay_2)^2}, \quad a = \hat{a}_4.$$

For accurate data, y_2 is approximately a^3. With this approximation,

$$u^2(\hat{a}_4) \doteq \frac{\sigma_1^2\sigma_2^2}{\sigma_2^2 + 9\sigma_1^2a^4}, \quad a = \hat{a}_4. \tag{3.15}$$

Note that the expression for $u(\hat{a}_4)$ in (3.14) depends on $\mathbf{y}$ while that in (3.15) does not.

We illustrate these distributions for the cases $a = 0.0$, 0.6 and 1.0 and $\sigma_1 = \sigma_2 = 0.2$. Figure 3.4 shows the model curve (a, a^3), 100 sampled data vectors $\mathbf{y}_q$, and the estimates $\hat{a}_{1,q}$ derived from $\mathbf{y}_q$, namely $\hat{a}_{1,q} = y_{1,q}$. The model estimate corresponding to a point $\mathbf{y}_q$ is represented by the point on the curve specified by $\hat{a}_{1,q} = y_{1,q}$. This estimator ignores completely the information represented by $y_{2,q}$.

Figure 3.5 gives the corresponding picture for the second estimator with $\hat{a}_{2,q} = (y_{2,q})^{1/3}$, the cube root of the second coordinate of $\mathbf{y}_q$. The estimator associates to $\mathbf{y}_q$ the point on the curve at the same y-value. This estimator ignores completely the information represented by $y_{1,q}$. The estimates are grouped in two clusters at either side of $a = 0$. The third estimator is illustrated by Figure 3.6. As expected from the definition of $\hat{a}_3$, the estimates reflect properties of both the first and second estimators. One could argue that since the measurements y_1 and y_2 are equally accurate for this case ($\sigma_1 = \sigma_2$) then averaging the estimates $\hat{a}_1$ and $\hat{a}_2$ is an appropriate way of aggregating the information.

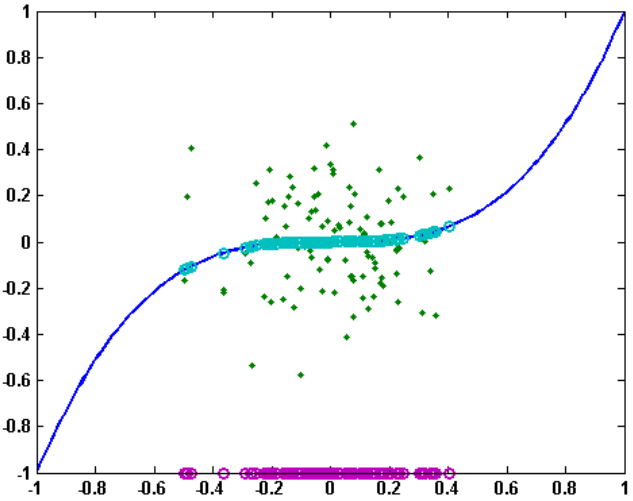

Figure 3.4: Simulated measured data $\mathbf{y}_q$ and estimates $\hat{a}_{1,q} = \mathcal{A}_1(\mathbf{y}_q)$, $q = 1, \ldots, 100$, generated according to the model (3.9), with $a = 0$ and $\sigma_1 = \sigma_2 = 0.2$.

Figure 3.7 graphs the estimates associated with the ML estimator. The parameter value associated with $\mathbf{y}_q$ is that which specifies the point on the curve closest to $\mathbf{y}_q$. (The graph's axes have slightly different scales so that the geometry portrayed is not Euclidean.) For $a = 0$, the behaviour of the ML estimates is very similar to that of the first estimator. This is because the slope of the model curve near $a = 0$ is small so that for $\mathbf{y}$ near $(0, 0)^{\mathrm{T}}$, the orthogonal projection of $\mathbf{y}$ onto the model curve is very close to the vertical projection defined by $\mathcal{A}_1$.

Figure 3.8 plots the distributions $p_{\hat{a}_k|a}$ for each of the four estimators. The distribution for the first estimator is easiest to understand as it is simply $N(0, \sigma_1^2)$. The distribution $p(\hat{a}_2|a)$ is totally different, bi-modal and with low density near zero. This behaviour was already noted in Figure 3.5. The distribution indicates that if $a = 0$, there is very little probability that the estimate $\hat{a}_2$ of a will be near zero. However, the distribution is symmetric about zero so that the expected value of $\hat{a}_2|a$ is zero, showing that the estimator is unbiased. This behaviour arises from the nonlinearity introduced by the cubic term associated with the model. (The nonlinearity itself is quite mild: the cubing function is a smooth one-to-one, strictly monotonic mapping.) The estimator $\hat{a}_3$ seems an appropriate use of the information provided by y_2. For example, if only the measurement information y_2 is available, then $\hat{a}_3$ coincides with the ML estimate of a. The distribution $p(\hat{a}_3|a)$ is also bi-modal, but has features less extreme than those for $p_{\hat{a}_2|a}$ due to the effect of averaging with $\hat{a}_1$. The distribution for $\hat{a}_4$ is essentially that for $\hat{a}_1$.

From this analysis of these estimators it would appear that if $a = 0$, estimators 1 and 4 give equivalent and most reliable estimates, and estimator 2 is the worst. Estimator 3, although it uses the information provided by measurements y_1 and y_2, performs much worse than estimator 1 which uses only y_1.

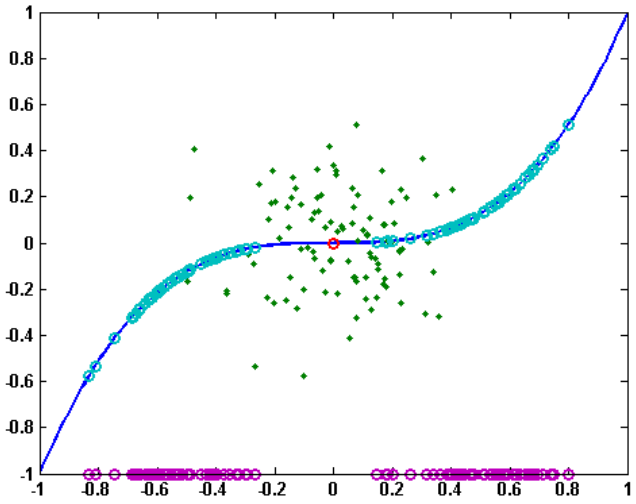

Figure 3.5: As Figure 3.4, but for estimator $\mathcal{A}_2$.

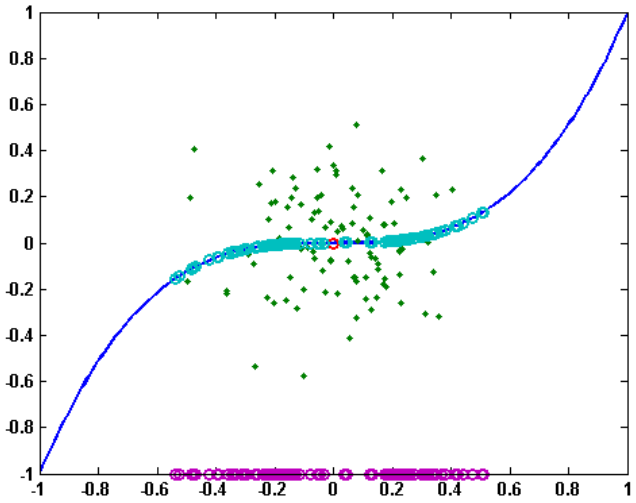

Figure 3.6: As Figure 3.4, but for estimator $\mathcal{A}_3$.

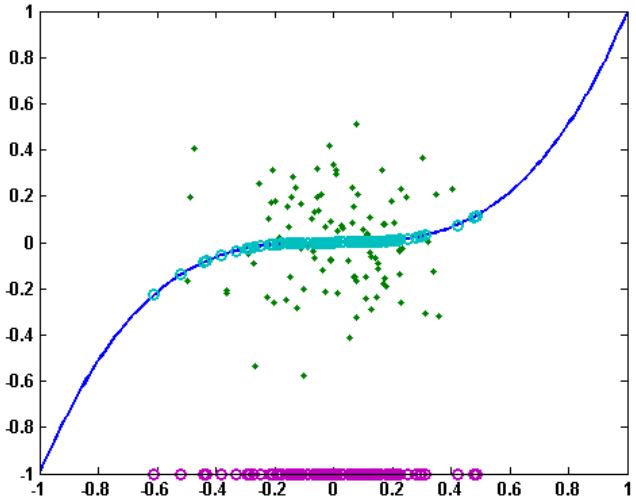

Figure 3.7: As Figure 3.4, but for estimator $\mathcal{A}_4$.

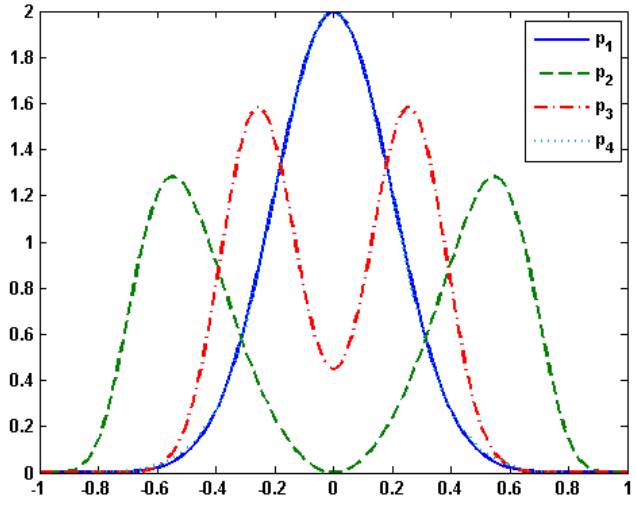

Figure 3.8: Distributions $p_k = p(\hat{a}_k|a)$ for estimators $\mathcal{A}_k$, $k = 1, 2, 3, 4$, related to the model (3.9), with $a = 0$ and $\sigma_1 = \sigma_2 = 0.2$. Distributions p_1 and p_4 are indistinguishable on this graph.

3.10.6 Inferences based on the measurements and estimates

The analysis of the estimators above concentrated in describing their behaviour in terms of the distributions $p(\hat{a}|a)$. We now look at the distributions $p(a|y_1)$, $p(a|y_2)$, $p(a|\mathbf{y})$ and $p(a|\hat{a}_k)$ that describe our knowledge about a, given that we have observed measurements $\mathbf{y}$ or estimates $\hat{a}_k$.

We assume that there is no substantive prior knowledge about a so that the improper prior distribution has 'PDF' $p(a) = 1$. (If we needed to, we could use a proper rectangular distribution on the interval $[-10, 10]$, for example.) With this assumption $p(a|y_1) \propto p(y_1|a)$, $p(a|y_2) \propto p(y_2|a)$ and $p(a|\mathbf{y}) \propto p(\mathbf{y}|a)$.

The distribution $p(a|y_1)$ is defined by the assignment

$$a \mapsto p(y_1|a) = \frac{1}{(2\pi\sigma_1^2)^{1/2}} \exp\left\{-\frac{1}{2}\left(\frac{y_1 - a}{\sigma_1}\right)^2\right\},$$

and we recognise the righthand function as the PDF associated with the normal distribution $N(y_1, \sigma_1^2)$. The fact that $p(a|y_1) = p(y_1|a)$, both regarded as functions of a, can be interpreted as follows: if the only thing we know about a parameter a comes from observing a measurement y drawn from $N(a, \sigma^2)$ with σ known, then this knowledge is captured by the statement that a is associated with the distribution $N(y, \sigma^2)$: $a \sim N(y, \sigma^2)$. The equivalence $y \sim N(a, \sigma^2) \equiv a \sim N(y, \sigma^2)$ is a result of the fact that a and y appear symmetrically through the term $(y - a)^2$ in the definition of the PDFs.

Continuing in the same way, the distribution $p(a|y_2)$ is defined by the assignment

$$a \mapsto p(y_2|a) = \frac{1}{(2\pi\sigma_2^2)^{1/2}} \exp\left\{-\frac{1}{2}\left(\frac{y_2 - a^3}{\sigma_2}\right)^2\right\}.$$

Thus, for each a, the distribution $p(y_2|a)$ is the Gaussian with mean a^3 and standard deviation σ_2 and is a function of the distance $d_2(y_2, a)$ defined by

$$d_2^2(y_2, a) = \frac{1}{\sigma_2^2}(y_2 - a^3)^2.$$

As a function of y_2, this distance in linear, but as a function of a, the nonlinear effect of the cubing function comes into play. In particular, for $|a| < 0.1$, $|a^3| < 0.001$, so that the distance is more or less constant in this region which means that $p(a|y_2)$ is flat over the same interval; see Figure 3.9.

The distribution $p(a|y_1, y_2)$ is defined by the assignment

$$a \mapsto K \exp\left\{-\frac{1}{2}\left[\left(\frac{y_1 - a}{\sigma_1}\right)^2 + \left(\frac{y_2 - a^3}{\sigma_2}\right)^2\right]\right\},$$

where K is a normalising constant. Figure 3.10 graphs $p(a|y_1, y_2)$ for $\mathbf{y}^{\mathrm{T}} = (0.0, 0.0)$, $(0.2, 0.0)$, $(0.0, 0.2)$ and $(0.2, -0.2)$. These distributions are approximately Gaussian, but with some asymmetry introduced by the nonlinear term a^3. For the case $\sigma_1 = \sigma_2$, $p(a|y_1, y_2)$ is a function of the Euclidean distance $d(\mathbf{y}, \boldsymbol{\phi}(a))$ from $\mathbf{y} = (y_1, y_2)^{\mathrm{T}}$ to $\boldsymbol{\phi}(a) = (a, a^3)^{\mathrm{T}}$. If $\mathbf{y} = (0, y_2)^{\mathrm{T}}$, then

$$d^2(\mathbf{y}, \boldsymbol{\phi}(a)) = a^2 + (y_2 - a^3)^2 = a^2 + y_2^2 - 2y_2a^3 + a^6.$$

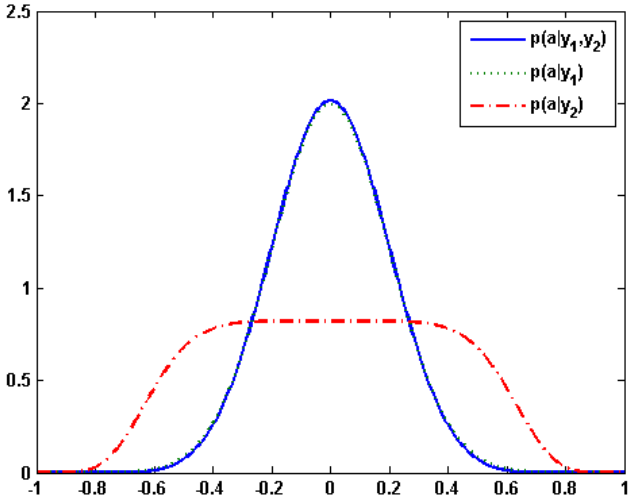

Figure 3.9: The distributions $p(a|y_1)$, $p(a|y_2)$ and $p(a|y_1, y_2)$ for $y_1 = y_2 = 0$. . Distributions $p(a|\mathbf{y})$ and $p(a|y_1)$ are indistinguishable in this graph.

Near $a = 0$, if $y_2 = 0$, then the nonlinearity of $\phi(a)$ does not come into play until a^6 becomes significant relative to a^2. On the other hand, if y_2 is significant in magnitude, the nonlinearity has an effect as soon as $2y_2a^3$ is significant relative to a^2. The asymmetry of the distribution also depends on the sign of y_2.

3.10.7 Comparison of $p(\hat{a}|a)$, $p(a|\hat{a})$ and $p(a|\mathbf{y})$

The distribution $p(\hat{a}|a)$ describes, for a fixed a, the likely variation in the estimate $\hat{a} = \mathcal{A}(\mathbf{y})$ due to the variation in $\mathbf{y}$. This behaviour can be estimated using forward Monte Carlo simulation (section 3.13). For the case of the first estimator $\hat{a}_1 = y_1$, if measurement y_1 is observed we can calculate the density $p(\hat{a}_1|a = y_1)$ which, from the above analysis, is $N(y_1, \sigma_1^2)$. Thus, if y_1 is observed, we associate with the estimate $\hat{a}_1$, the distribution $N(y_1, \sigma^2)$. This is the same distribution as $p(a|y_1)$ so that for this estimator, once a measurement y_1 has been observed, the distributions $p(\hat{a}|a)$ and $p(a|y_1)$ are the same, namely $N(y_1, \sigma^2)$. In this case, the distribution associated with the estimate $\hat{a}$ and the measurand a, given y_1 are the same.

The analysis for estimator $\hat{a}_2$ shows a totally different behaviour. Suppose that a value of $y_2 = 0$ is observed so that $\hat{a}_2 = y_1^{1/3} = 0$. Figure 3.11 graphs the probability distributions $p(a|y_2 = 0)$ and $p(\hat{a}_2|a = 0)$. The two distributions could hardly be more different. The distribution $p(a|y_2 = 0)$ accords the largest densities for a near zero while $p(\hat{a}|y_2)$ is zero at $\hat{a}_2 = 0$. It should be stressed that the differences are not the consequence of any approximations due to linearisations, etc. They are different because they are distributions for two different quantities.

The differences can be explained using inverse Monte Carlo calculations. For $q = 1, \ldots, M$,

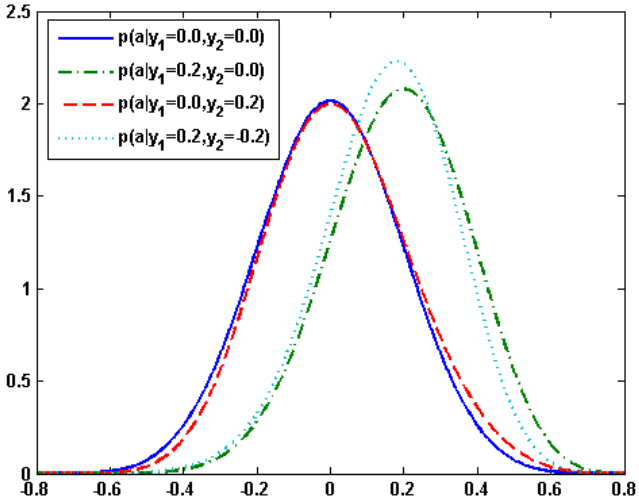

Figure 3.10: The distributions $p(a|y_1, y_2)$ for different values of y_1 and y_2.

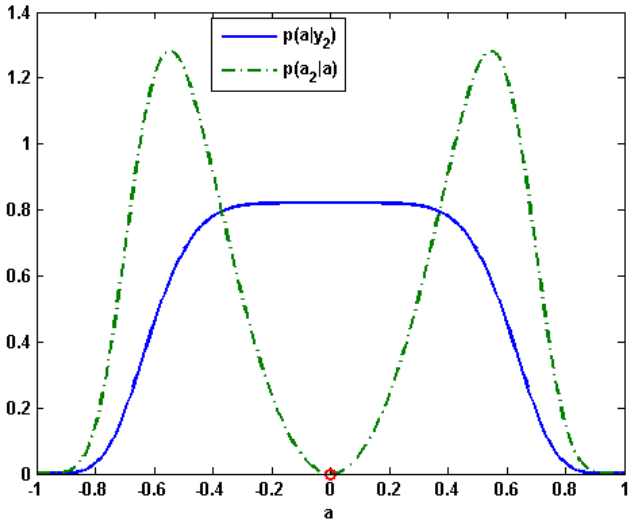

Figure 3.11: The distributions $p(a|y_2 = 0)$ and $p(\hat{a}_2|a = 0)$.

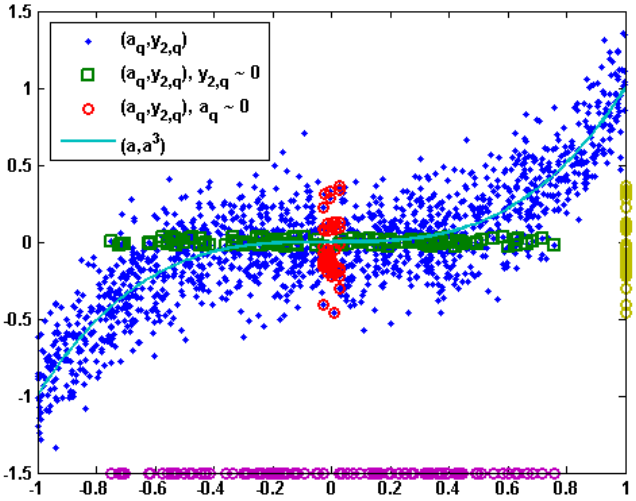

Figure 3.12: Data points $(a_q, y_{2,q})$ generated in an inverse Monte Carlo simulation (3.16).

we draw a_q from a uniform distribution defined on the interval $[-1, 1]$ and then sample

$$y_{1,q} = a_q + \epsilon_{1,q}, \quad \epsilon_{1,q} \in N(0, \sigma_1^2), \quad y_{2,q} = a_q^3 + \epsilon_{2,q}, \quad \epsilon_{2,q} \in N(0, \sigma_2^2). \tag{3.16}$$

Figure 3.12 shows 1500 points $(a_q, y_{2,q})$, generated according to this scheme with $a = 0$ and $\sigma_1 = \sigma_2 = 0.2$. The Figure shows that these data points as scattered in a band about the curve (a, a^3). The Figure also shows the data points for which a_q is close to zero and those for which $y_{q,2}$ is close to zero. Figure 3.13 is similar, only plotting $(a_q, \hat{a}_{2,q})$ with $\hat{a}_{2,q} = y_{2,q}^{1/3}$. The cube root function has the effect of pulling the data points towards the lines $y = \pm 1$, which leads to the bi-modal distribution for the estimates $\hat{a}_{2,q}$ that has already been observed. Note that in Figures 3.12 and 3.13 the distribution of the a_q for $y_{2,q}$ or $\hat{a}_{2,q}$ near zero is the same. This is to be expected since the information provided by $\hat{a}_2$ is the same as that provided by y_2 as they are related through the one-to-one function $y_2 = \hat{a}_2^3$. This means that the distribution $p(a|\hat{a}_2)$ is essentially the same as $p(a|y_2)$. Figure 3.14 plots $p(a|\hat{a}_2)$ estimated from inverse Monte Carlo simulation and $p(a|y_2)$ for the case $y_2 = 0$.[3]

While estimators $\hat{a}_1$ and $\hat{a}_2$ each use only a subset of the data (y_1 and y_2, respectively), estimators $\hat{a}_3$ and $\hat{a}_4$ use the complete set $\mathbf{y}$. Figure 3.15 plots the three distributions $p(a|y_1 = 0, y_2 = 0)$, $p(a|\hat{a}_3 = 0)$ and $p(\hat{a}_3|a = 0)$. All three are different. Firstly the distributions $p(a|\hat{a}_3 = 0)$ and $p(\hat{a}_3|a = 0)$ are different but this is to be expected at they represent two different sets of information. That $p(a|y_1 = 0, y_2 = 0)$ is different from $p(a|\hat{a}_3 = 0)$ reflects the fact that knowing $\hat{a}_3 = 0$ is not nearly as strong as knowing $y_1 = y_2 = 0$ as far as inferences about a is concerned. Figure 3.16 graphs $p(a|\hat{a}_4 = 0)$, estimated using inverse Monte Carlo simulation, and $p(a|y_1 = 0, y_2 = 0)$, and shows that for this case knowing the estimate $\hat{a}_4$ is essentially equivalent to knowing the data $\mathbf{y}$.

[3] There is technical difference in the two distributions in that while $p(a|y_2)$ is defined for $y_2 = 0$, $p(a|\hat{a}_2 = 0)$ is defined as a limiting case of $p(a|\hat{a}_2 \in [-\delta, \delta])$ as $\delta \longrightarrow 0$. Inverse Monte Carlo simulation, in fact, estimates this latter distribution. For other values, the distributions are related directly.

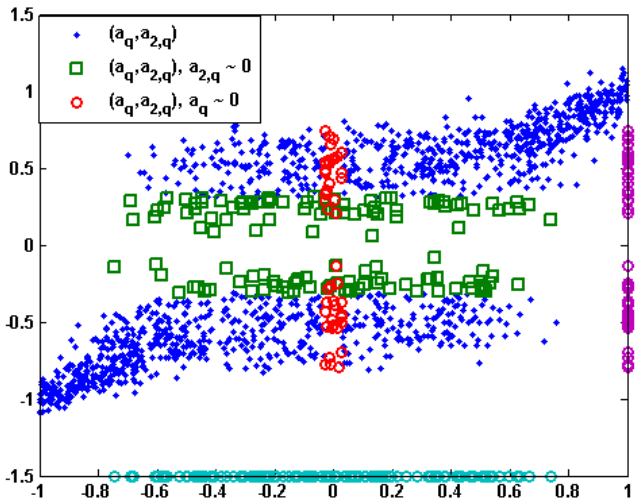

Figure 3.13: Data points $(a_q, \hat{a}_{2,q})$ generated in an inverse Monte Carlo simulation (3.16).

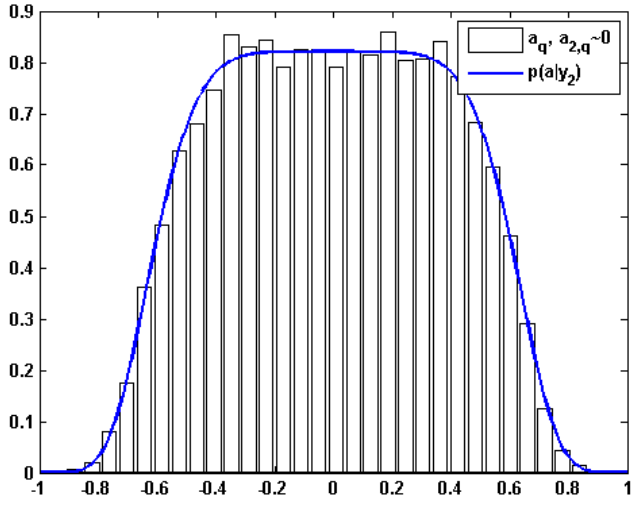

Figure 3.14: Distributions $p(a|\hat{a}_2)$ estimated from inverse Monte Carlo simulation and $p(a|y_2)$ for the case $y_2 = 0$.

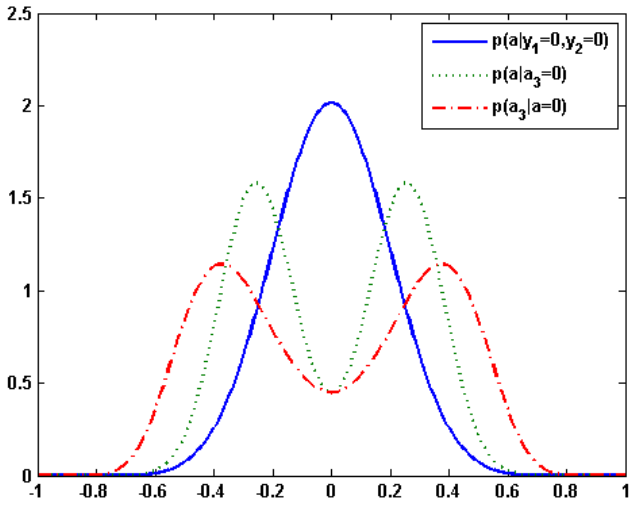

Figure 3.15: Distributions $p(a|y_1 = 0, y_2 = 0)$, $p(a|\hat{a}_3 = 0)$ and $p(\hat{a}_3|a = 0)$.

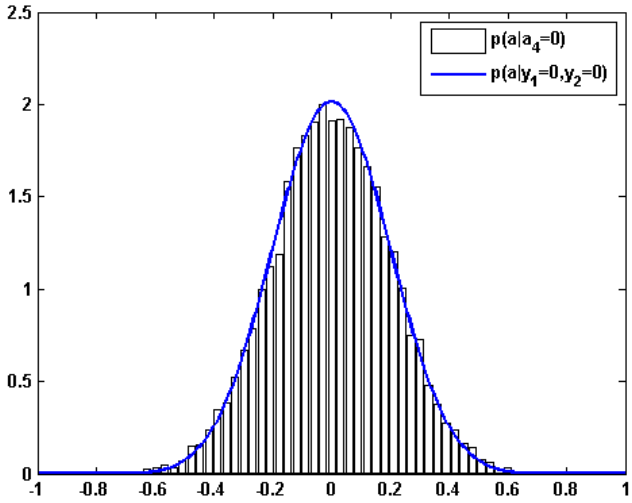

Figure 3.16: Distributions $p(a|y_1 = 0, y_2 = 0)$, $p(a|\hat{a}_4 = 0)$ and $p(\hat{a}_4|a = 0$.

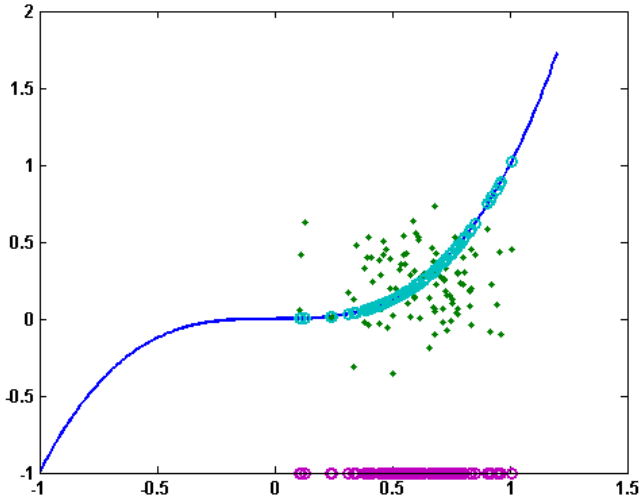

Figure 3.17: Simulated measurement data $\mathbf{y}_q$ and estimates $\hat{a}_{1,q} = \mathcal{A}_1(\mathbf{y}_q)$, $q = 1, \ldots, 100$, generated according to the model (3.9), with $a = 0.6$ and $\sigma_1 = \sigma_2 = 0.2$.

The above analysis has been related to behaviour for a near 0. We can repeat the analysis for other values. Figures 3.17–3.20 correspond to Figures 3.4–3.7 but for $a = 0.6$. In Figures 3.18 and 3.19 it is seen that estimators $\mathcal{A}_2$ and to a lesser extent $\mathcal{A}_3$ have an asymmetric bimodal character. Figure 3.21 graphs the distributions $p(\hat{a}_k|a = 0.6)$, for the estimators $\mathcal{A}_k$, $k = 1, 2, 3, 4$, while Figure 3.22 graphs the distributions $p(a|y_1 = 0.6, y_2 = 0.6^3)$, $p(a|y_1 = 0.6)$ and $p(a|y_2 = 0.6)$. For the case $a = 0$, it was seen that the information about a was derived mainly from the measurement y_2; see Figure 3.9. It is seen in Figure 3.22 that for the case $a = 0.6$, the use of both measurements y_1 and y_2 give sharper information about a.

Figure 3.23 plots the distributions $p(a|y_2 = 0.6^3)$ and $p(\hat{a}_2|a = 0.6)$. As in the case $a = 0$ (Figure 3.11), these distributions are quite different. However, as noted in the case $a = 0$, the distribution $p(a|\hat{a}_2) = p(a|y_2)$; the information derived from knowing the estimate $\hat{a}_2$ is the same as that derived from knowing y_2. Figure 3.24 graphs the distributions $p(a|y_1 = 0.6, y_2 = 0.6^3)$, $p(a|\hat{a}_3 = 0.6)$ and $p(\hat{a}_3|a = 0.6)$. The Figure shows that for this case knowing $\hat{a}_3$ is almost as good as knowing $\mathbf{y}$, so that in this situation, $\mathcal{A}_3$ is reasonably efficient. Note that the shape of $p(\hat{a}_3|a = 0.6)$ is quite different.

Figure 3.25 graphs the distributions $p(a|y_1 = 0.6, y_2 = 0.6^3)$, $p(a|\hat{a}_4 = 0.6)$ and $p(\hat{a}_4|a = 0.6)$. The graph shows that knowing the maximum likelihood estimate is essentially equivalent to knowing $\mathbf{y}$. The graph also shows the Gaussian approximant $p_N(a|0.6, \sigma_N^2)$ to $p(a|y_1 = 0.6, y_2 = 0.6^3)$ determined from a quadratic approximation to $-\log p(a|y_1 = 0.6, y_2 = 0.6^3)$ as described in section 3.5.2.

Figure 3.26 correspond to Figures 3.7 and 3.20 and gives the ML estimates $\hat{a}_4$ for the case $a = 1.0$, Figure 3.27 graphs the distributions $p(\hat{a}_k|a = 1.0)$, for the estimators $\mathcal{A}_k$, $k = 1, 2, 3, 4$, and Figure 3.28 graphs the distributions $p(a|y_1 = 1.0, y_2 = 1.0)$, $p(a|y_1 = 1.0)$ and $p(a|y_2 = 1.0)$. It is seen in Figure 3.28 the measurement y_2 is the dominant source of information about a.

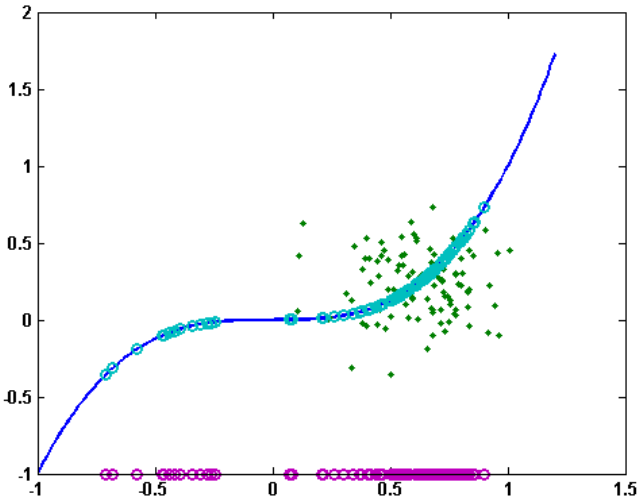

Figure 3.18: As Figure 3.17, but for estimator $\mathcal{A}_2$.

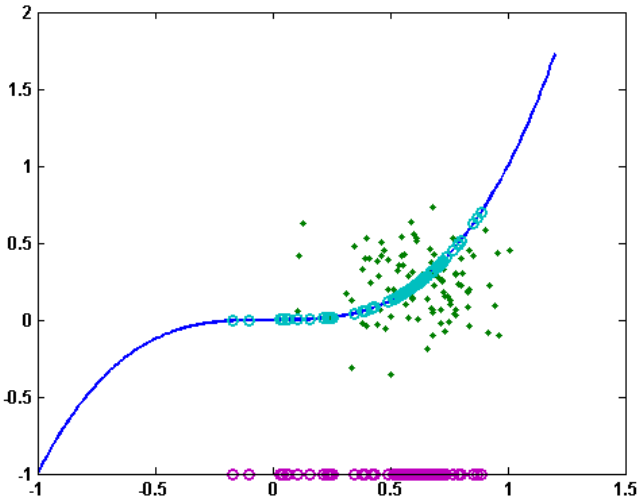

Figure 3.19: As Figure 3.17, but for estimator $\mathcal{A}_3$.

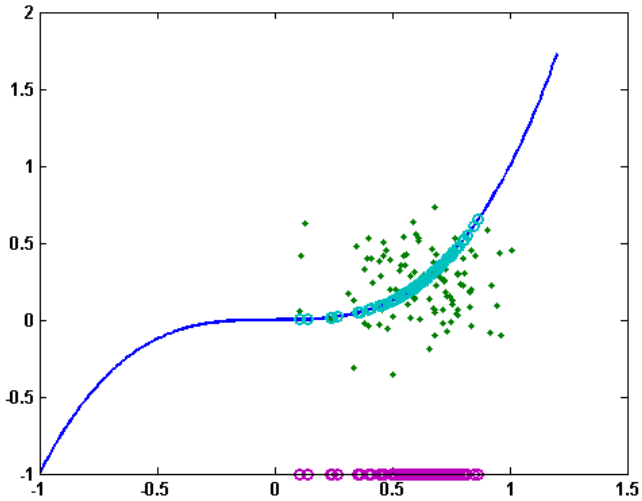

Figure 3.20: As Figure 3.17, but for estimator $\mathcal{A}_4$.

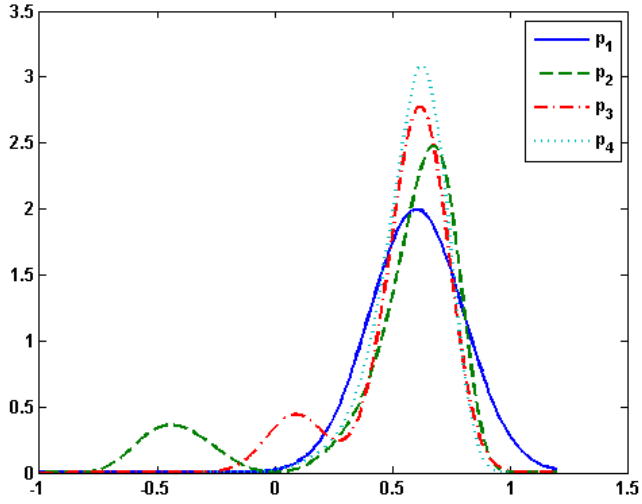

Figure 3.21: Distributions $p(\hat{a}_k|a)$ for estimators $\mathcal{A}_k$, $k = 1, 2, 3, 4$, related to the model (3.9), with $a = 0.6$ and $\sigma_1 = \sigma_2 = 0.2$.

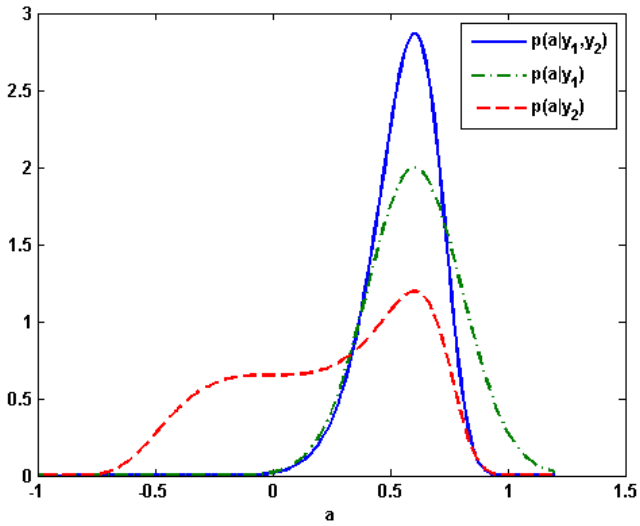

Figure 3.22: The distributions $p(a|y_1)$, $p(a|y_2)$ and $p(a|y_1, y_2)$ for $y_1 = 0.6$, $y_2 = 0.6^3$.

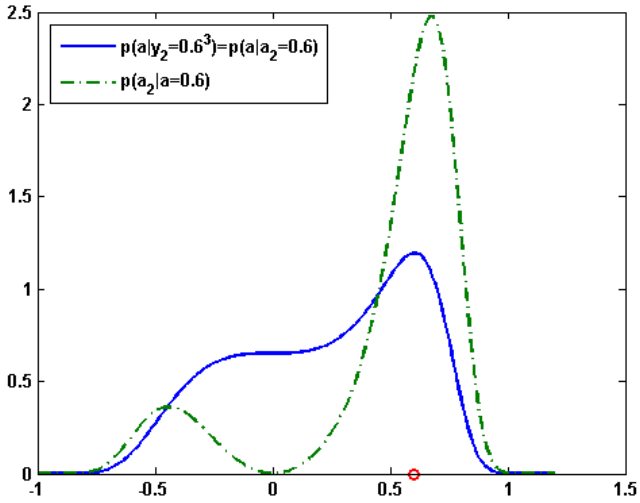

Figure 3.23: Distributions $p(a|y_2 = 0.6^3)$ and $p(\hat{a}_2|a = 0.6)$.

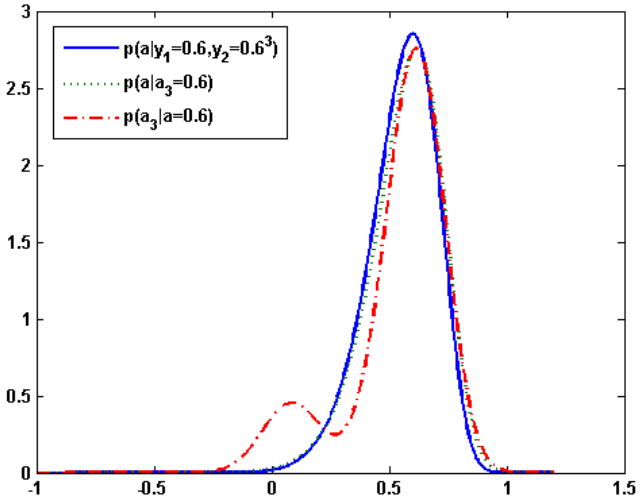

Figure 3.24: Distributions $p(a|y_1 = 0.6, y_2 = 0.6^2)$, $p(a|\hat{a}_3 = 0.6)$ and $p(\hat{a}_3|a = 0.6)$.

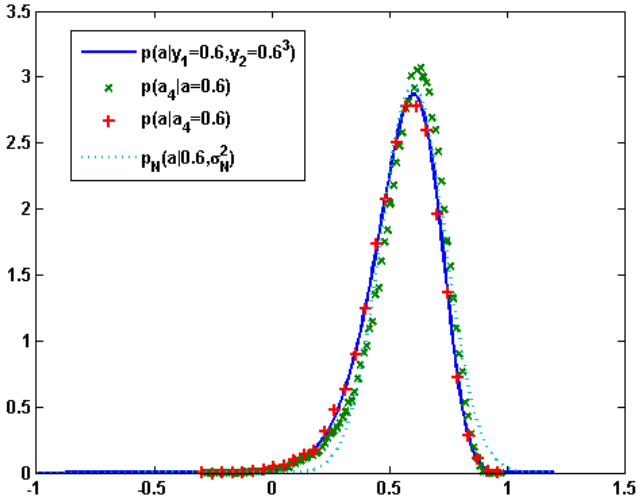

Figure 3.25: Distributions $p(a|y_1 = 0.6, y_2 = 0.6^2)$, $p(a|\hat{a}_4 = 0.6)$, $p(\hat{a}_4|a = 0.6)$ and normal distribution $p_N(a|0.6, \sigma_N^2)$.

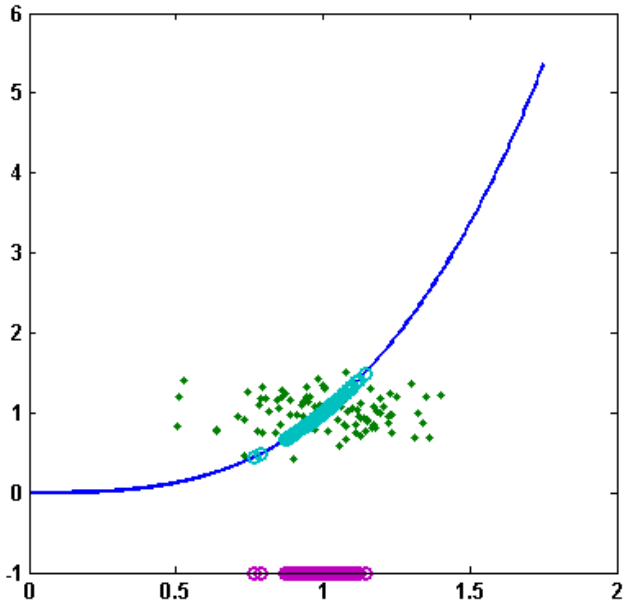

Figure 3.26: Simulation measurements $\mathbf{y}_q$ and estimates $\hat{a}_{4,q} = \mathcal{A}_4(\mathbf{y}_q)$, $q = 1, \ldots, 100$, generated according to the model (3.9), with $a = 1.0$ and $\sigma_1 = \sigma_2 = 0.2$.

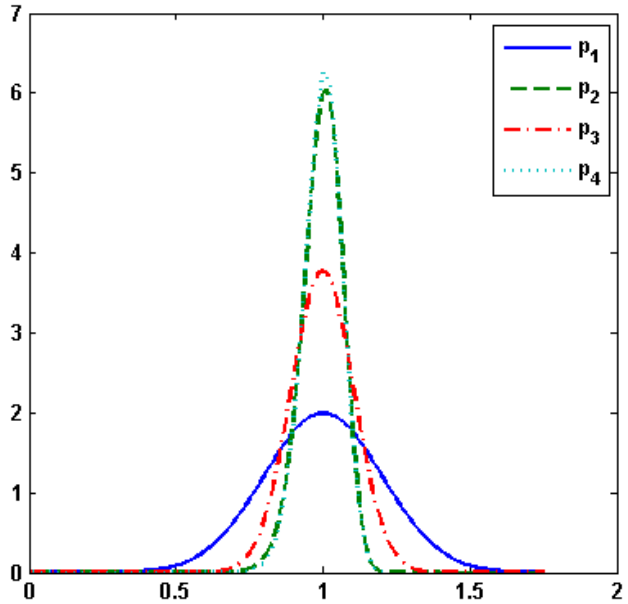

Figure 3.27: Distributions $p(\hat{a}_k|a)$ for estimators $\mathcal{A}_k$, $k = 1, 2, 3, 4$, related to the model (3.9), with $a = 1.0$ and $\sigma_1 = \sigma_2 = 0.2$.

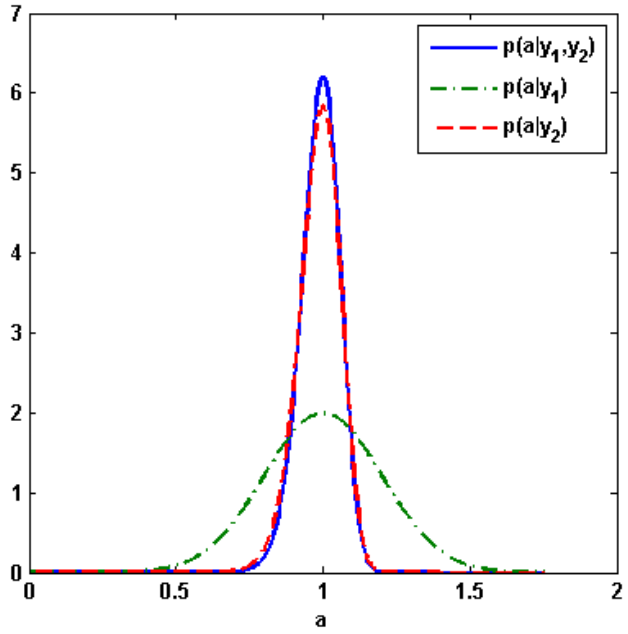

Figure 3.28: The distributions $p(a|y_1)$, $p(a|y_2)$ and $p(a|y_1, y_2)$ for $y_1 = y_2 = 1.0$.

3.10.8 Why MLE is special

We can think of point estimation, i.e., defining an estimate $\hat{\mathbf{a}} = \mathcal{A}(\mathbf{y})$, from data $\mathbf{y}$ as a method of summarising the information in $\mathbf{y}$ relevant to the model $\boldsymbol{\phi}(\mathbf{a})$. This example gives us an intuitive explanation of why maximum likelihood estimates generally are more effective than other estimation methods. For the cases illustrated $p(a|\hat{a}_4)$ is close to $p(a|\mathbf{y})$, so that an inference based on the knowledge of $\hat{a}_4$ is almost as precise as one based on $p(a|\mathbf{y})$. For the case $\sigma_1 = \sigma_2$, the ML estimate defines the points $\boldsymbol{\phi}(a)$ closest to $\mathbf{y}$.

For example, if $\hat{a}_4 = 0$, we know that $\mathbf{y} = (0, y_2)^{\mathrm{T}}$ for some y_2, and lies on the line orthogonal to the curve at $\boldsymbol{\phi}(\hat{a}_4)$. We saw earlier that if there is no prior information, $p(a|\mathbf{y})$ is proportional to $p(\mathbf{y}|a)$ and depends on the distance function

$$d^2(\mathbf{y}, \boldsymbol{\phi}(a)) = a^2 + (y_2 - a^3)^2 = a^2 + y_2^2 - 2y_2a^3 + a^6.$$

This function depends on y_2 and hence $p(a|\hat{a}_4)$ will be different from $p(a|\mathbf{y})$ since the latter depends on y_2 while the former does not. More generally, suppose the measurement model is $\mathbf{y} = \boldsymbol{\phi}(\mathbf{a}) + \boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \in N(0, \sigma^2 I)$, and let $\hat{\mathbf{a}}$ be any estimate of $\mathbf{a}$ derived from $\mathbf{y}$ and set $\hat{\mathbf{y}} = \boldsymbol{\phi}(\hat{\mathbf{a}})$. The posterior distribution for $\mathbf{a}$, $p(\mathbf{a}|\mathbf{y})$, depends on the distance function $d(\mathbf{y}, \boldsymbol{\phi}(\mathbf{a}))$. Applying the cosine rule,

$$\|\mathbf{y} - \boldsymbol{\phi}(\mathbf{a})\|^2 = \|\mathbf{y} - \hat{\mathbf{y}}\|^2 + \|\boldsymbol{\phi}(\mathbf{a}) - \hat{\mathbf{y}}\|^2 - 2\|\mathbf{y} - \hat{\mathbf{y}}\|\|\boldsymbol{\phi}(\mathbf{a}) - \hat{\mathbf{y}}\| \cos\theta$$

where θ is the angle between the vectors $\mathbf{y} - \hat{\mathbf{y}}$ and $\hat{\mathbf{y}} - \boldsymbol{\phi}(\mathbf{a})$. The first term on the right is a constant with respect to $\mathbf{a}$ and so does not contribute any information about $\mathbf{a}$. The second term is known if we know the estimate $\hat{\mathbf{a}}$. The third term involves both $\mathbf{y}$ and $\mathbf{a}$ and

represents the information about $\mathbf{a}$ available from $\mathbf{y}$ that is missing from the estimate $\hat{\mathbf{a}}$. The ML estimate is such that $\mathbf{y} - \hat{\mathbf{y}}$ is orthogonal to the surface at $\hat{\mathbf{y}}$ so that for $\mathbf{a}$ near $\hat{\mathbf{a}}$, $\cos\theta$ is near zero. In this sense, the ML estimate minimises the information lost in summarising $\mathbf{y}$ by an estimate $\hat{\mathbf{a}}$. For linear models $\cos\theta$ is identically zero and no information is lost.

3.10.9 Conceptual example: summary

The conceptual example has illustrated the following points.

- The distribution $p(\mathbf{y}|\mathbf{a})$ specifies the likely variation in the data vector $\mathbf{y}$, for a fixed set of parameter values $\mathbf{a}$.
- Inferences about $\mathbf{a}$, given that a data vector $\mathbf{y}$ has been observed, can be derived from the distribution $p(\mathbf{a}|\mathbf{y})$.
- Bayes' theorem states that $p(\mathbf{a}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{a})p(\mathbf{a})$ where $p(\mathbf{a})$ is the prior distribution for $\mathbf{a}$.
- For random effects described by multivariate normal (Gaussian) distributions and constant prior distribution, $p(\mathbf{y}|\mathbf{a})$ and $p(\mathbf{a}|\mathbf{y})$ are defined in terms of the distance $d(\mathbf{y}, \boldsymbol{\phi}(\mathbf{a}))$ from the data vector $\mathbf{y}$ to the point $\boldsymbol{\phi}(\mathbf{a})$ on the model surface. The shape of the model surface $\boldsymbol{\phi}(\mathbf{a})$ is reflected in the shape of the distribution $p(\mathbf{a}|\mathbf{y})$. If $\boldsymbol{\phi}(\mathbf{a})$ is (approximately) linear, then $p(\mathbf{a}|\mathbf{y})$ is (approximately) a multivariate Gaussian.
- Parameter estimation corresponds to finding a point $\hat{\mathbf{y}} = \boldsymbol{\phi}(\hat{\mathbf{a}})$, $\hat{\mathbf{a}} = \mathcal{A}(\mathbf{y})$, on the model surface $\boldsymbol{\phi}(\mathbf{a})$ close to the data vector $\mathbf{y}$.
- The distribution $p(\hat{\mathbf{a}}|\mathbf{a})$ specifies how variation in the data vector $\mathbf{y}$ propagates through to variation in the parameter estimates. The distribution $p(\hat{\mathbf{a}}|\mathbf{a})$ depends on the geometry of the model surface $\boldsymbol{\phi}(\mathbf{a})$ and also on the definition $\hat{\mathbf{a}} = \mathcal{A}(\mathbf{y})$ of the estimation method. A discrete representation of this distribution can be determined using forward Monte Carlo simulation.
- The distributions $p(\hat{\mathbf{a}}|\mathbf{a})$ and $p(\mathbf{a}|\mathbf{y})$ represent two different statistical quantities (related to the same model) and for nonlinear model surfaces, they will be different from each other in general.
- The distribution $p(\mathbf{a}|\hat{\mathbf{a}})$ represents the information available about $\mathbf{a}$, given that an estimate $\hat{\mathbf{a}} = \mathcal{A}(\mathbf{y})$ has been observed. In general, the distribution $p(\mathbf{a}|\hat{\mathbf{a}})$ will provide less precise information about $\mathbf{a}$ than $p(\mathbf{a}|\mathbf{y})$.
- Regarding the estimate $\hat{\mathbf{a}}$ as a summary of the data vector $\mathbf{y}$, the effectiveness (or efficiency) of this estimate is measured by the 'closeness' of $p(\mathbf{a}|\hat{\mathbf{a}})$ to $p(\mathbf{a}|\mathbf{y})$. Maximum likelihood estimates, in general, retain more information about $\mathbf{a}$ than other estimates.

Chapter 4

Parameter estimation methods

In this chapter, we describe in more detail some of the common parameter estimation methods and associated algorithms.

4.1 Linear least squares (LLS)

4.1.1 Description

Given data $\{(\mathbf{x}_i, y_i)\}_1^m$ and the linear model

$$y = \phi(\mathbf{x}, \mathbf{a}) = a_1\phi_1(\mathbf{x}) + \ldots + a_n\phi_n(\mathbf{x}), \quad n \leq m,$$

the linear least-squares estimate of the parameters $\mathbf{a}$ is the one which solves

$$\min_{\mathbf{a}} \sum_{i=1}^{m} (y_i - \mathbf{c}_i^{\mathrm{T}}\mathbf{a})^2,$$

where $\mathbf{c}_i = (\phi_1(\mathbf{x}_i), \ldots, \phi_n(\mathbf{x}_i))^{\mathrm{T}}$.

Let C be the matrix whose ith row is $\mathbf{c}_i^{\mathrm{T}}$, $\mathbf{y}$ the vector whose ith element is y_i and $\mathbf{f}(\mathbf{a}) = \mathbf{y} - C\mathbf{a}$. The problem can be reposed as

$$\min_{\mathbf{a}} F(\mathbf{a}) = \mathbf{f}^{\mathrm{T}}\mathbf{f} = \|\mathbf{y} - C\mathbf{a}\|_2^2.$$

At the solution, it is known that the partial derivatives of F with respect to the parameters are zero, i.e.,

$$\frac{\partial F}{\partial a_j} = 0, \quad j = 1, \ldots, n,$$

and this leads to the system of linear equations of order n,

$$C^{\mathrm{T}}C\mathbf{a} = C^{\mathrm{T}}\mathbf{y}, \tag{4.1}$$

known as the *normal equations*. If C is full rank, so that $C^{\mathrm{T}}C$ is invertible, the solution parameters are given (mathematically) by

$$\mathbf{a} = (C^{\mathrm{T}}C)^{-1}C^{T}\mathbf{y}. \tag{4.2}$$

Geometrical interpretation. The linear least squares estimate has the following geometrical interpretation. The columns $\mathbf{c}_j$, $j = 1, \ldots, n$ and $\mathbf{y}$ are vectors (or points) in $\mathcal{R}^m$. Linear combinations

$$C\mathbf{a} = \sum_{j=1}^{n} a_j \mathbf{c}_j = a_1\mathbf{c}_1 + \cdots + a_n\mathbf{c}_n,$$

of the vectors $\mathbf{c}_j$ define points in the n-dimensional linear subspace $\mathcal{C}$ (a hyper-plane) defined by these column vectors. The linear least squares solution defines the point $\hat{\mathbf{y}} = C\mathbf{a}$ on the linear subspace $\mathcal{C}$ closest to $\mathbf{y}$. The vector $\mathbf{y} - \hat{\mathbf{y}}$ from $\mathbf{y}$ to $C\mathbf{a}$ must be orthogonal (perpendicular) to the plane and in particular perpendicular to the vectors $\mathbf{c}_j$: $\mathbf{c}_j^{\mathrm{T}}(\mathbf{y} - C\mathbf{a}) = 0$, $j = 1, \ldots, n$. Writing these equations in matrix terms,

$$C^{\mathrm{T}}(\mathbf{y} - C^{\mathrm{T}}C\mathbf{a}) = \mathbf{0},$$

from which we derive the normal equations (4.1).

Linear least-squares estimators are the most common of the estimators used in metrology. They correspond to the maximum likelihood estimate for linear models in which the measurements of a single response variable are subject to uncorrelated normally distributed random effects:

$$y_i = a_1\phi_1(\mathbf{x}_i) + \ldots + a_n\phi_n(\mathbf{x}_i) + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2), \quad i = 1, \ldots, m \geq n.$$

They are suitable for any system for which the main random effects are associated with the response variables and these effects are symmetrically distributed about a zero mean; see section 4.1.13.

Linear least squares are less suitable for data in which more than one variable is subject to significant random effects or for data which contains outliers or rogue points or where the random effects are modelled as being governed by long tailed distributions (section 4.7).

4.1.2 Algorithms to find the linear least-squares estimate

There are two basic approaches to determining a least-squares solution to a set of over-determined equations.

Solving the normal equations. Although equation (4.2) suggests that the linear least-squares estimate is found by inverting the $n \times n$ matrix $H = C^{\mathrm{T}}C$, as in the case of practically all matrix equation problems, matrix inversion is far from the best option. If the normal equations are to be solved, the preferred approach exploits the fact that H is symmetric and, assuming it is full rank, has a Cholesky decomposition

$$H = LL^{\mathrm{T}},$$

where L is an $n \times n$ lower triangular matrix (so that $L(i, j) = 0$ if $i < j$). With this factorisation, the parameters $\mathbf{a}$ are determined by solving, in sequence, two triangular systems

$$L\mathbf{b} = C^{\mathrm{T}}\mathbf{y}, \quad L^{\mathrm{T}}\mathbf{a} = \mathbf{b}.$$

The Cholesky factorisation and the solution of the triangular systems are easily implemented in software, requiring only a few lines of code [117].

Cholesky factorisation. An $n \times n$ matrix A is symmetric if $A(i,j) = A(j,i)$, $1 \leq i,j \leq n$. A symmetric matrix is (strictly) positive definite if all its eigenvalues are (strictly) positive. If A is strictly positive definite, it can be factored as $A = LL^{\mathrm{T}}$, where L is lower triangular. The elements of L can be found by a simple step-by-step approach, as the following example indicates. If

$$\begin{bmatrix} a_{11} & a_{21} & a_{31} & a_{41} \\ a_{21} & a_{22} & a_{32} & a_{42} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & l_{31} & l_{41} \\ 0 & l_{22} & l_{32} & l_{42} \\ 0 & 0 & l_{33} & l_{34} \\ 0 & 0 & 0 & l_{44} \end{bmatrix},$$

then

$$\begin{aligned} a_{11} = l_{11}^2 &\quad\Rightarrow\quad l_{11} = \sqrt{a_{11}}, \\ a_{21} = l_{21} l_{11} &\quad\Rightarrow\quad l_{21} = a_{21}/l_{11}, \\ a_{22} = l_{21}^2 + l_{22}^2 &\quad\Rightarrow\quad l_{22} = \sqrt{[a_{22} - l_{21}^2]}, \end{aligned}$$

etc. The following algorithm computes a lower triangular matrix L such that $A = LL^{\mathrm{T}}$. The lower triangular elements $A(i,j)$, $i \geq j$ are overwritten by $L(i,j)$ [117, section 4.2.5].

```
for k = 1 : m
    A(k,k) := (A(k,k))^{1/2}
    for j = k + 1 : m
        A(j,k) := A(j,k)/A(k,k)
    end
    for j = k + 1 : m
        for l = j : m
            A(l,j) := A(l,j) − A(l,k)A(j,k)
        end
    end
end
```

The calculations can be re-organised to involve more vector-vector operations in order to improve execution speed in computer languages that support vector and array operations. For example,

```
for j = 1 : m
    if j > 1
      A(j : m, j) := A(j : m, j) − A(j : m, 1 : j − 1)A(j, 1 : j − 1)^T
    end
    A(j : m, j) := A(j : m, j)/√A(j, j)
end
```

If A has a negative eigenvalue, then (in exact arithmetic) the Cholesky factorisation will encounter having to calculate the square root of a negative number.

Solution of a lower triangular system. The importance of triangular matrices in matrix factorisation approaches is that it is straightforward to find the solution of a system of equations involving a triangular matrix. The following example gives the general approach for a lower triangular system. If

$$\begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

then

$$\begin{aligned} l_{11}x_1 = y_1 &\Rightarrow x_1 = y_1/l_{11}, \\ l_{21}x_1 + l_{22}x_2 = y_2 &\Rightarrow x_2 = (y_2 - l_{21}x_1)/l_{22}, \\ l_{31}x_1 + l_{32}x_2 + l_{33}x_3 = y_3 &\Rightarrow x_3 = (y_3 - l_{31}x_1 - l_{32}x_2)/l_{33}, \end{aligned}$$

etc. This scheme is known as *forward substitution*. If L is an $n \times n$ lower triangular matrix and $\mathbf{x} = (x_1, \ldots, x_n)^{\mathrm{T}}$ is an n-vector, the following algorithm overwrites the vector $\mathbf{x}$ with $L^{-1}\mathbf{x}$:

$x(1) := x(1)/L(1,1)$
for $j = 2 : m$
  for $k = 1 : j-1$
    $x(j) := x(j) - L(j,k)x(k)$
  end
  $x(j) = x(j)/L(j,j)$
end

Similarly, if

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ 0 & r_{22} & r_{23} & r_{24} \\ 0 & 0 & r_{33} & r_{34} \\ 0 & 0 & 0 & r_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix},$$

then

$$\begin{aligned} x_1 &= (y_1 - r_{12}x_2 - r_{13}x_3 - r_{14}x_4)/r_{11}, \\ x_2 &= (y_2 - r_{23}x_3 - r_{24}x_4)/r_{22}, \\ x_3 &= (y_3 - r_{34}x_4)/r_{33}, \\ x_4 &= y_4/r_{44}, \end{aligned},$$

working from the bottom to the top. This scheme is known as *backwards substitution*.

Orthogonal factorisation methods. If the matrix C is well conditioned, the Cholesky factorisation approach to determining the solution to the normal equations gives accurate results. However, if C is poorly conditioned, then forming the product $H = C^{\mathrm{T}}C$ is likely to lead to rounding errors and loss of numerical accuracy. It may be that the computed H fails to be strictly positive definite due to rounding errors and the calculation of the Cholesky factorisation will not be possible.

If C has orthogonal factorisation (section 3.8.1)

$$C = QR = [Q_1 \; Q_2] \begin{bmatrix} R_1 \\ 0 \end{bmatrix}, \tag{4.3}$$

then, using the fact that $\|Q\mathbf{x}\| = \|\mathbf{x}\|$, we have

$$\|\mathbf{y} - C\mathbf{a}\| = \|Q^{\mathrm{T}}\mathbf{y} - Q^{\mathrm{T}}C\mathbf{a}\| = \left\| \begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \end{bmatrix} - \begin{bmatrix} R_1 \\ 0 \end{bmatrix} \mathbf{a} \right\|,$$

where $\mathbf{t}_1$ is the first n and $\mathbf{t}_2$ the last $m - n$ elements of $\mathbf{t} = Q^{\mathrm{T}}\mathbf{y}$, i.e., $\mathbf{t}_1 = Q_1^{\mathrm{T}}\mathbf{y}$ and $\mathbf{t}_2 = Q_2^{\mathrm{T}}$. From this it is seen that $\|\mathbf{y} - C\mathbf{a}\|$ is minimised if $\mathbf{a}$ solves the upper triangular system

$$R_1\mathbf{a} = \mathbf{t}_1.$$

In practice, the orthogonalisation is applied to the augmented matrix

$$Q^{\mathrm{T}} \left[\begin{array}{cc} C & \mathbf{y} \end{array} \right] = \left[\begin{array}{cc} R_1 & \mathbf{t}_1 \\ 0 & \|\mathbf{f}\| \\ \mathbf{0} & \mathbf{0} \end{array} \right],$$

to produce simultaneously the upper triangular factor R_1, the right-hand side vector $\mathbf{t}_1$ and the norm $\|\mathbf{f}\|$ of the residuals $\mathbf{f} = \mathbf{y} - C\mathbf{a}$. As with the Cholesky factorisation, orthogonal factorisations are easy to construct [117].

Geometrical interpretation. The QR factorisation has the following geometrical interpretation. The column vectors $\mathbf{c}_j$ of C define an n dimensional subspace $\mathcal{C}$ of $\mathcal{R}^m$. The orthogonal matrix Q defines an axis system for $\mathcal{R}^m$ such that the n columns of Q_1 define an axis system for $\mathcal{C}$ and the $m - n$ columns of Q_2 define an axis system for the space of vectors $\mathcal{C}^{\perp}$ orthogonal to $\mathcal{C}$. The columns for Q_1 are constructed so that $\mathbf{q}_1$ is aligned with $\mathbf{c}_1$ so that there is an r_{11} such that $\mathbf{c}_1 = r_{11}\mathbf{q}_1$. The vector $\mathbf{q}_2$ is chosen to lie in the plane defined by $\mathbf{c}_1$ and $\mathbf{c}_2$, and so there are scalars r_{12} and r_{22} such that $\mathbf{c}_2 = r_{12}\mathbf{q}_1 + r_{22}\mathbf{q}_2$, etc. This gives the factorisation

$$[\mathbf{c}_1 \ \cdots \ \mathbf{c}_n] = [\mathbf{q}_1 \ \cdots \ \mathbf{q}_n] \left[\begin{array}{cccc} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & r_{2n} \\ & & \ddots & \vdots \\ 0 & 0 & & \\ 0 & 0 & \cdots & r_{nn} \end{array} \right],$$

i.e., in matrix notation $C = Q_1 R_1$.

The point $\hat{\mathbf{y}}$ in $\mathcal{C}$ closest to $\mathbf{y}$ can be written as a linear combination of the columns of Q_1 and is given by $\hat{\mathbf{y}} = Q_1\mathbf{t}_1$ for some $\mathbf{t}_1$. As before, $\mathbf{y} - Q_1\mathbf{t}_1$ must be orthogonal to $\mathcal{C}$ and so orthogonal to the columns of Q_1:

$$Q_1^{\mathrm{T}}(\mathbf{y} - Q_1\mathbf{t}_1) = \mathbf{0},$$

which implies

$$Q_1^{\mathrm{T}}Q_1\mathbf{t}_1 = Q_1^{\mathrm{T}}\mathbf{y}$$

or $\mathbf{t}_1 = Q_1^{\mathrm{T}}\mathbf{y}$, since $Q_1^{\mathrm{T}}Q_1 = I$, the identity matrix. Thus, the point in $\mathcal{C}$ closest to $\mathbf{y}$ is given by $\hat{\mathbf{y}} = Q_1\mathbf{t}_1$ where $\mathbf{t}_1 = Q_1^{\mathrm{T}}\mathbf{y}$: $\hat{\mathbf{y}} = Q_1\left(Q_1^{\mathrm{T}}\mathbf{y}\right)$. The point $\hat{\mathbf{y}}$ must also be a linear combination of the columns of C, so that $\hat{\mathbf{y}} = C\mathbf{a}$, for some $\mathbf{a}$. Equating $C\mathbf{a}$ with $Q_1\mathbf{t}_1$, we have

$$Q_1R_1\mathbf{a} = Q_1\mathbf{t}_1 \Rightarrow Q_1^{\mathrm{T}}Q_1R_1\mathbf{a} = Q_1^{\mathrm{T}}Q_1\mathbf{t}_1 \Rightarrow R_1\mathbf{a} = \mathbf{t}_1,$$

since $Q_1^{\mathrm{T}}Q_1 = I$. The coefficients $\mathbf{t}_1$ define $\hat{\mathbf{y}}$ as a linear combination of vectors $\mathbf{q}_j$; solving $R_1\mathbf{a} = \mathbf{t}_1$ re-defines $\hat{\mathbf{y}}$ as a linear combination of the vectors $\mathbf{c}_j$.

Since

$$\mathbf{y} = Q(Q^{\mathrm{T}}\mathbf{y}) = Q\left[\begin{array}{c} \mathbf{t}_1 \\ \mathbf{t}_2 \end{array} \right], \quad \mathbf{t}_1 = Q_1^{\mathrm{T}}\mathbf{y}, \quad \mathbf{t}_2 = Q_2^{\mathrm{T}}\mathbf{y},$$

we can express $\mathbf{y}$ as

$$\mathbf{y} = Q\left[\begin{array}{c} \mathbf{t}_1 \\ \mathbf{0} \end{array} \right] + Q\left[\begin{array}{c} \mathbf{0} \\ \mathbf{t}_2 \end{array} \right] = [C\mathbf{a} \in \mathcal{C}] + [(\mathbf{y} - C\mathbf{a}) \in \mathcal{C}^{\perp}],$$

which decomposes $\mathbf{y}$ as a sum of a vector in $\mathcal{C}$ and one in $\mathcal{C}^{\perp}$. Note that the residuals $\mathbf{y} - C\mathbf{a}$ can be computed without having to calculate $\mathbf{a}$.

The main advantage of the orthogonal factorisation method over the normal equations method is one of numerical accuracy. If, due to ill-conditioning in the matrix C, the orthogonal factorisation method potentially loses p decimal digits of accuracy, then the normal equations method potentially loses $2p$ decimal digits.

QR factorisation via Householder reflections. Given an m-vector $\mathbf{v}$, define the $m \times m$ matrix H by

$$H = I - \beta \mathbf{v}\mathbf{v}^{\mathrm{T}}, \quad \beta = 2/(\mathbf{v}^{\mathrm{T}}\mathbf{v}).$$

This type of matrix is known as a *Householder reflection*. We can calculate that

$$H^{\mathrm{T}}H = (I - \beta \mathbf{v}\mathbf{v}^{\mathrm{T}})(I - \beta \mathbf{v}\mathbf{v}^{\mathrm{T}}) = I - 2\beta \mathbf{v}\mathbf{v}^{\mathrm{T}} + \beta^2 \mathbf{v}(\mathbf{v}^{\mathrm{T}}\mathbf{v})\mathbf{v}^{\mathrm{T}} = I,$$

so that H is an orthogonal matrix and

$$H\mathbf{v} = (I - \beta \mathbf{v}\mathbf{v}^{\mathrm{T}})\mathbf{v} = \mathbf{v} - \beta(\mathbf{v}^{\mathrm{T}}\mathbf{v})\mathbf{v} = -\mathbf{v}.$$

Thus H reflects the vector $\mathbf{v}$. More generally, $H\mathbf{x}$ is the reflection of $\mathbf{x}$ in the plane orthogonal to $\mathbf{v}$, so that

$$\mathbf{x} - H\mathbf{x} = \beta(\mathbf{v}^{\mathrm{T}}\mathbf{x})\mathbf{v}$$

is a multiple of $\mathbf{v}$. This equation can be used the other way around. If we want $H\mathbf{x} = \mathbf{y}$ (where necessarily $\|\mathbf{y}\| = \|\mathbf{x}\|$), we choose the H defined by $\mathbf{v} = \mathbf{x} - \mathbf{y}$. Householder reflections can be used to perform an upper-triangularisation of a matrix. For any $\mathbf{x}$, we can define a reflection H that transforms $\mathbf{x}$ to the first co-ordinate axis, $H\mathbf{x} = \|\mathbf{x}\|\mathbf{e}_1$ where $\mathbf{e}_1 = (1, 0, \ldots, 0)^{\mathrm{T}}$, by choosing $\mathbf{v} = \mathbf{x} - \|\mathbf{x}\|\mathbf{e}_1$. Using this type of reflection, the following scheme indicates how a QR factorisation of a matrix A can be performed.

$$H_1 A = H_1 \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \\ a_{51} & a_{52} & a_{53} \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ 0 & b_{22} & b_{23} \\ 0 & b_{32} & b_{33} \\ 0 & b_{42} & b_{43} \\ 0 & b_{52} & b_{53} \end{bmatrix},$$

$$H_2 H_1 A = H_2 \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ 0 & b_{22} & b_{23} \\ 0 & b_{32} & b_{33} \\ 0 & b_{42} & b_{43} \\ 0 & b_{52} & b_{53} \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ 0 & c_{22} & c_{23} \\ 0 & 0 & c_{33} \\ 0 & 0 & c_{43} \\ 0 & 0 & c_{53} \end{bmatrix},$$

$$H_3 H_2 H_1 A = H_3 \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ 0 & c_{22} & c_{23} \\ 0 & 0 & c_{33} \\ 0 & 0 & c_{43} \\ 0 & 0 & c_{53} \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ 0 & c_{22} & c_{23} \\ 0 & 0 & d_{33} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

If an orthogonal matrix Q is expressed as a product of Householder reflections, $Q = H_1 \times \cdots \times H_n$, then Q is specified by the corresponding vectors $\mathbf{v}_k$ and scalers β_k. The matrices Q and H_k need never be calculated explicitly. To calculate $\mathbf{y} = Q\mathbf{x}$, for example, only a sequence of vector-vector operations are required.

Taking into account sparsity structure in the observation matrix. There are a number of applications in which the observation matrix has a large number of zero entries. This *sparsity structure* can be exploited to increase the efficiency of the solution process; some of these techniques are described in [55, 57, 65, 182].

4.1.3 Uncertainty associated with the fitted parameters

The uncertainty matrix $V_{\mathbf{a}}$ associated with the fitted parameters is obtained using the fact that the linear least-squares solution $\mathbf{a}$ is a linear combination of the data vector $\mathbf{y}$. If

$\mathbf{y} = (y_1, \ldots, y_m)^{\mathrm{T}}$ has associated uncertainty matrix[1] matrix $V_{\mathbf{y}}$ and $\mathbf{a}(\mathbf{y}) = G\mathbf{y}$ are n linear functions of $\mathbf{y}$, then the uncertainty matrix associated with $\mathbf{a}$ is given by[2]

$$V_{\mathbf{a}} = GV_{\mathbf{y}}G^{\mathrm{T}}.$$

The normal equations (4.1) define the linear least-squares solution (from equation (4.2)), as

$$\mathbf{a} = C^{\dagger}\mathbf{y},$$

where

$$C^{\dagger} = (C^{\mathrm{T}}C)^{-1}C^{\mathrm{T}} \tag{4.4}$$

is the *pseudo-inverse* of C [117, section 5.5.4] and is such that $CC^{\dagger}C = C$, $C^{\dagger}CC^{\dagger} = C^{\dagger}$ and $C^{\dagger}\left(C^{\dagger}\right)^{\mathrm{T}} = (C^{\mathrm{T}}C)^{-1}$. Therefore,

$$V_{\mathbf{a}} = C^{\dagger}V_{\mathbf{y}}\left(C^{\dagger}\right)^{\mathrm{T}}. \tag{4.5}$$

If $V_{\mathbf{y}} = \sigma^2 I$ (as in the case for the standard experiment), this expression simplifies to

$$V_{\mathbf{a}} = C^{\dagger}\sigma^2 I\left(C^{\dagger}\right)^{\mathrm{T}} = \sigma^2(C^{\mathrm{T}}C)^{-1}. \tag{4.6}$$

If C has orthogonal factorisation given in (4.3) then, using the fact that $Q^{\mathrm{T}}Q = I$ for an orthogonal matrix, $V_{\mathbf{a}}$ can be calculated from the triangular factor R_1 and σ:

$$V_{\mathbf{a}} = \sigma^2\left(R_1^{\mathrm{T}}R_1\right)^{-1} = \sigma^2 R_1^{-1}R_1^{-\mathrm{T}}.$$

If $h = \mathbf{h}^{\mathrm{T}}\mathbf{a}$, a linear combination of the parameters, then

$$u(h) = \sigma\|\tilde{\mathbf{h}}\|,$$

where $\tilde{\mathbf{h}}$ solves

$$R_1^{\mathrm{T}}\tilde{\mathbf{h}} = \mathbf{h}.$$

This means that the standard uncertainties associated with the fitted parameters, or linear combinations of those parameters, can be determined efficiently by solving such triangular systems.

These calculations assume that the standard deviation σ associated with the random effects[3] in the data is already known. If this is not the case, then for overdetermined systems a *posterior estimate* $\hat{\sigma}$ of σ can be determined from the vector $\mathbf{r} = \mathbf{y} - C\mathbf{a}$ of residuals:

$$\hat{\sigma} = \|\mathbf{r}\|/(m-n)^{1/2}. \tag{4.7}$$

With this posterior estimate of σ, the uncertainty matrix associated with the fitted parameters is approximated by

$$\widehat{V}_{\mathbf{a}} = \hat{\sigma}^2\left(C^{\mathrm{T}}C\right)^{-1}. \tag{4.8}$$

However, see section 4.1.5.

[1] That is, $\mathbf{y}$ is an observation of a vector of random variables $\boldsymbol{Y}$ whose multivariate distribution has variance matrix $V_{\mathbf{y}}$.

[2] That is, $\mathbf{a}$ is an observation of a vector of random variables $\boldsymbol{A}$ whose multivariate distribution has variance matrix $V_{\mathbf{a}}$.

[3] That is, $\boldsymbol{\epsilon} \in \boldsymbol{E}$ with $V(\boldsymbol{E}) = \sigma^2 I$.

Details. The estimate $\hat{\sigma}$ of σ is justified as follows. If $X_i \sim N(0,1)$, $i = 1, \ldots, m$, are independent normal variates then $\sum_{i=1}^{m} X_i^2$ has a χ^2_m distribution with mean m and variance $2m$. Let $\boldsymbol{R}$ be the random vector of residuals so that

$$\boldsymbol{R} = \boldsymbol{Y} - C\boldsymbol{A} = \boldsymbol{Y} - CC^{\dagger}\boldsymbol{Y} = (I - CC^{\dagger})\boldsymbol{Y}.$$

We assume that $\mathrm{Var}(\boldsymbol{Y}) = \sigma^2 I$ so that $(1/\sigma^2)\sum_{i=1}^{m} \sim \chi^2_m$. If $C = Q_1 R_1$ as in (4.3), then $CC^{\dagger} = Q_1 Q_1^{\mathrm{T}}$ and $I - Q_1 Q_1^{\mathrm{T}} = Q_2 Q_2^{\mathrm{T}}$, so that

$$S^2 = \boldsymbol{R}^{\mathrm{T}}\boldsymbol{R} = \left(Q_2^{\mathrm{T}}\boldsymbol{Y}\right)^{\mathrm{T}} Q_2^{\mathrm{T}}\boldsymbol{Y}.$$

Now Q is orthogonal so setting $\tilde{\boldsymbol{Y}} = Q\boldsymbol{Y}$ we have $\mathrm{Var}(\tilde{\boldsymbol{Y}}) = \sigma^2 I$ also. Therefore, $S^2/\sigma^2 = (1/\sigma^2)\sum_{i=n+1}^{m} \tilde{Y}_i^2$ is a sum of squares of $m - n$ independent, normal variates and has a χ^2_ν distribution with $\nu = m - n$ degrees of freedom, with $E(S^2/\sigma^2) = \nu$ or $E(S^2) = \sigma^2(m - n)$. From this analysis, we see that given a least-squares solution $\mathbf{a}$, a posterior estimate of σ is given $\hat{\sigma}$ in (4.7) if we equate the expected value of the sum of squared residuals with its observed value.

While this estimate is derived under the assumption that the random effects are governed by a Gaussian distribution, it is likely to be a good approximation for distributions with similar features, e.g., unimodal (that is, having one peak).

Geometric interpretation. The above calculations have a geometrical interpretation. If

$$\boldsymbol{Y} = C\mathbf{a}^* + \boldsymbol{E},$$

where E_i are independent, $E_i \sim N(0, \sigma^2)$, the vector $\boldsymbol{E}$ is the sum of two mutually orthogonal vectors

$$\boldsymbol{E} = C(\boldsymbol{A} - \mathbf{a}^*) + (\boldsymbol{Y} - C\boldsymbol{A}) \tag{4.9}$$

where $C(\boldsymbol{A} - \mathbf{a}^*)$ lies in the n-space $\mathcal{C}$ defined by the column vectors of C and $\boldsymbol{R} = \boldsymbol{Y} - C\boldsymbol{A}$ lies in the $(m-n)$-space $\mathcal{C}^{\perp}$ of vectors orthogonal to $\mathcal{C}$. Pythagoras's theorem in this context is simply $m\sigma^2 = n\sigma^2 + (m-n)\sigma^2$. When an experiment is made and data $y_i \in Y_i$ are recorded, only the residuals $r_i \in R_i$ in (4.9) are observable (since $\mathbf{a}^*$ is unknown), and the estimate of $\hat{\sigma}$ is derived from the residual vector $\mathbf{r}$.

4.1.4 Linear least squares and maximum likelihood estimation

The uncertainty matrix $V_{\mathbf{a}}$ derived above only relies on the law of propagation of uncertainty. If we make the further assumption that the observed data $\mathbf{y}$ are a sample from a multivariate normal distribution

$$\mathbf{y} \in N(C\mathbf{a}^*, V_{\mathbf{y}}),$$

then the linear least-squares solution $\mathbf{a}$ is also a sample from a multivariate normal distribution:

$$\mathbf{a} \in N(\mathbf{a}^*, V_{\mathbf{a}}).$$

If $V_{\mathbf{y}} = \sigma^2 I$, then the linear least-squares solution is also the maximum likelihood estimate. The probability $p(\mathbf{y}|\mathbf{a})$ of observing $\mathbf{y}$, given $\mathbf{a}$ is such that

$$p(\mathbf{y}|\mathbf{a}) \propto \exp\left\{-\frac{1}{2\sigma^2}(\mathbf{y} - C\mathbf{a})^{\mathrm{T}}(\mathbf{y} - C\mathbf{a})\right\},$$

and is maximised by the least squares solution.

In a Bayesian context we regard $\mathbf{a}$ as a vector of parameters, information about which is described in terms of probability distributions. If there is no substantial prior information about $\mathbf{a}$ so that the prior distribution $p(\mathbf{a})$ can be taken to be a constant, the posterior probability distribution $p(\mathbf{a}|\mathbf{y})$ is proportional to the likelihood:

$$p(\mathbf{a}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{a}).$$

If we denote by $\hat{\mathbf{a}}$ the least squares solution $\hat{\mathbf{a}} = C^{\dagger}\mathbf{y}$, then $C\hat{\mathbf{a}}$ is the point in the space $\mathcal{C}$ defined by the columns of C closest to $\mathbf{y}$. The vector $\mathbf{y} - C\mathbf{a}$ can be written as the sum of two mutually orthogonal vectors

$$\mathbf{y} - C\mathbf{a} = [\mathbf{y} - C\hat{\mathbf{a}}] + [C(\hat{\mathbf{a}} - \mathbf{a})],$$

so that from Pythagoras's Theorem

$$(\mathbf{y} - C\mathbf{a})^{\mathrm{T}}(\mathbf{y} - C\mathbf{a}) = (\mathbf{y} - C\hat{\mathbf{a}})^{\mathrm{T}}(\mathbf{y} - C\hat{\mathbf{a}}) + (C(\hat{\mathbf{a}} - \mathbf{a}))^{\mathrm{T}}(C(\hat{\mathbf{a}} - \mathbf{a})). \tag{4.10}$$

The first term on the right does not depend on $\mathbf{a}$ and so

$$p(\mathbf{a}|\mathbf{y}) \propto \exp\left\{-\frac{1}{2\sigma^2}(\mathbf{a} - \hat{\mathbf{a}})^{\mathrm{T}}C^{\mathrm{T}}C(\mathbf{a} - \hat{\mathbf{a}})\right\}. \tag{4.11}$$

Comparing the righthand side with the multivariate normal distribution, we see that

$$\mathbf{a}|\mathbf{y} \sim N(\hat{\mathbf{a}}, V_{\mathbf{a}}), \quad V_{\mathbf{a}} = \frac{1}{\sigma^2}\left(C^{\mathrm{T}}C\right)^{-1}.$$

These calculations show that the probability $p(\hat{\mathbf{a}}|\mathbf{a})$ of observing a least squares estimate $\hat{\mathbf{a}}$, given $\mathbf{a}$ is

$$\hat{\mathbf{a}}|\mathbf{a} \sim N(\mathbf{a}, V_{\mathbf{a}}).$$

On the other hand, the distribution $p(\mathbf{a}|\hat{\mathbf{a}})$ for $\mathbf{a}$ having observed a least-squares estimate $\hat{\mathbf{a}}$ is

$$\mathbf{a}|\hat{\mathbf{a}} \sim N(\hat{\mathbf{a}}, V_{\mathbf{a}}).$$

The symmetry in these two statements reflects the fact that $\mathbf{a}$ and $\hat{\mathbf{a}}$ appear symmetrically in (4.11). Furthermore, $p(\mathbf{a}|\hat{\mathbf{a}}) = p(\mathbf{a}|\mathbf{y})$, so that from this point of view, the least squares estimate does not lose any of the information that can be derived from the data $\mathbf{y}$.

4.1.5 Partial information about σ

The uncertainty matrices $V_{\mathbf{a}}$ (4.6) and $\widehat{V}_{\mathbf{a}}$ (4.8) are derived for the cases σ known and unknown, respectively. In a Bayesian context, and assuming normally distributed random effects, it is also possible to consider prior information about σ that represents degrees of belief between these two extremes. The estimate (4.8) derived from the posterior estimate $\hat{\sigma}$ of σ is the same as that calculated if the standard deviation was *known* to be $\hat{\sigma}$, but since $\hat{\sigma}$ is only estimated from the data, it seems plausible that $\widehat{V}_{\mathbf{a}}$ will underestimate the uncertainty, particularly if $m - n$ is small. The analysis below does in fact lead to different estimate of the uncertainty matrix.

We assume that σ_0 represents a prior estimate of σ and the degree of belief associated with this estimated is encoded in a parameter $m_0 \geq 0$. A large value of m_0 indicates a strong

degree of belief in σ_0, a small value, a weak degree. We can think of m_0 as the number of data points used to estimate σ from a previous experiment. The posterior distribution for $\mathbf{a}$ is a multivariate t-distribution $t_\nu(\hat{\mathbf{a}}, \bar{V})$ centered on the least squares estimate $\hat{\mathbf{a}}$, with ν degrees of freedom and scale matrix $\bar{V}$, where

$$\nu = m_0 + m - n, \quad \bar{V} = \bar{\sigma}^2 (C^{\mathrm{T}} C)^{-1}, \quad \bar{\sigma}^2 = \frac{m_0 \sigma_0^2 + m s^2}{m_0 + m - n}, \quad m s^2 = \mathbf{r}^{\mathrm{T}} \mathbf{r},$$

with $\mathbf{r} = \mathbf{y} - C\hat{\mathbf{a}}$. (Details of how the t-distribution arises are given below.) If $m_0 = 0$, corresponding to no prior information about σ,

$$\bar{V} = \frac{\mathbf{r}^{\mathrm{T}} \mathbf{r}}{m - n} (C^{\mathrm{T}} C)^{-1},$$

the same as $\widehat{V}_{\mathbf{a}}$ in (4.8). The penalty for having no knowledge about σ is that the normal distribution $N(\hat{\mathbf{a}}, \widehat{V}_{\mathbf{a}})$ is replaced by the multivariate t-distribution $t_{m-n}(\hat{\mathbf{a}}, \widehat{V}_{\mathbf{a}})$. If m_0 is large, corresponding to a high degree of belief in the prior estimate σ_0^2, then

$$\bar{V} \approx \sigma_0^2 (C^{\mathrm{T}} C)^{-1},$$

the same as that calculated in (4.6). As $m_0 \longrightarrow \infty$, the t-distribution approaches the corresponding normal distribution. In other cases, the scale matrix is defined by $\bar{\sigma}^2$ which can be regarded as an average of the prior sum of squares $m_0 \sigma_0^2$ and the sum of squares $m s^2 = \mathbf{r}^{\mathrm{T}} \mathbf{r}$, $\mathbf{r} = \mathbf{y} - C\hat{\mathbf{a}}$, arising from the data $\mathbf{y}$.

For $\nu > 2$, the variance matrix associated with $t_\nu(\boldsymbol{\mu}, V)$ is $\nu V/(\nu - 2)$. For the case $m + m_0 > n - 2$, it is appropriate to associate with the least squares estimate $\hat{\mathbf{a}}$, the uncertainty matrix

$$\check{V}_{\mathbf{a}} = \frac{m_0 + m - n}{m_0 + m - n - 2} \bar{V} = \left(\frac{m_0 \sigma_0^2 + \mathbf{r}^{\mathrm{T}} \mathbf{r}}{m_0 + m - n - 2} \right) (C^{\mathrm{T}} C)^{-1}, \quad \mathbf{r} = \mathbf{y} - C\hat{\mathbf{a}},$$

the variance matrix of the corresponding t-distribution.

Details. We first consider an appropriate distribution to characterise information about a variance parameter σ^2. Suppose $z_i \in N(0, \sigma^2)$, $i = 1, \ldots, m_0$, represent m_0 samples from a normal distribution whose standard deviation is unknown. What does the data $\mathbf{z} = (z_1, \ldots, z_{m_0})^{\mathrm{T}}$ tell us about σ^2? Let $\sigma_0^2 = \frac{1}{m_0} \sum_{i=1}^{m_0} z_i^2$. From the definition of the normal distribution, given σ, the probability of observing $\mathbf{z}$ is such that

$$p(\mathbf{z}|\sigma) \propto \sigma^{-m_0} \exp\left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^{m_0} z_i^2 \right\} = \sigma^{-m_0} \exp\left\{ -\frac{m_0 \sigma_0^2}{2\sigma^2} \right\}.$$

Since $p(\mathbf{z}|\sigma)$ only depends on $\mathbf{z}$ through σ_0^2, $p(\mathbf{z}|\sigma) = p(\sigma_0^2|\sigma)$. In fact, given σ,

$$\frac{m_0 \sigma_0^2}{\sigma^2} \sim \chi^2_{m_0}, \tag{4.12}$$

since z_i/σ is a sample from $N(0, 1)$ and the sum of squares of n standard normal variates has a χ_n^2 distribution. Writing $\eta = 1/\sigma^2$, then from Bayes' Theorem, the distribution $p(\eta|\sigma_0^2)$ for η, given σ_0^2, is such that

$$p(\eta|\sigma_0^2) \propto p(\sigma_0^2|\eta) p(\eta),$$

where $p(\eta)$ is the prior distribution for η. If we have no information about η, a suitable 'distribution' for η is $p(\eta) = 1/\eta$, $\eta > 0$. (The distribution $p(\eta) = 1/\eta$ corresponds to a uniform distribution for $\log \sigma$.) With this prior,

$$p(\eta|\sigma_0^2) = p(\eta|\mathbf{z}) \propto \eta^{m_0/2 - 1} \exp\left\{ -\frac{\eta}{2} m_0 \sigma_0^2 \right\},$$

The righthand side can be compared to the PDF for the gamma distribution $G(\alpha, \beta)$

$$p(x|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x},$$

where $\Gamma(\alpha)$ is the gamma function; for integer n, $\Gamma(n+1) = n!$. Thus, the information gained about $\eta = 1/\sigma^2$ by observing m_0 samples from $N(0, \sigma^2)$ is described by the distribution $\eta \sim G(m_0/2, m_0\sigma_0^2/2)$. In fact, the χ^2 distribution is a special case of the gamma distribution and $\eta \sim G(m_0/2, m_0\sigma_0^2/2)$ is equivalent to the scaled parameter $m_0\sigma_0^2\eta$ having the distribution $\chi^2_{m_0}$:

$$m_0\sigma_0^2\eta \sim \chi^2_{m_0}. \tag{4.13}$$

Note that in (4.12), the $\chi^2_{m_0}$ distribution relates to σ_0^2 as a parameter, while in (4.13), it relates to the parameter η. The parameter $\eta = 1/\sigma^2$ is sometimes referred to as the *precision*.

Suppose that the prior information for η is given by (4.13), and that there is no substantive prior information about the parameters $\mathbf{a}$ so that $p(\mathbf{a}) = 1$. The posterior joint distribution $p(\mathbf{a}, \eta|\mathbf{y})$ for $\mathbf{a}$ and η is such that

$$p(\mathbf{a}, \eta|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{a}, \eta)p(\eta).$$

The likelihood $p(\mathbf{y}|\mathbf{a}, \eta)$ of observing $\mathbf{y}$, given $\mathbf{a}$ and η, is derived from the multivariate normal distribution since we are assuming $\mathbf{y} \sim N(C\mathbf{a}, \eta^{-1}I)$:

$$p(\mathbf{y}|\mathbf{a}, \eta) \propto \eta^{m/2} \exp\left\{-\frac{\eta}{2}(\mathbf{y} - C\mathbf{a})^{\mathrm{T}}(\mathbf{y} - C\mathbf{a})\right\}.$$

From (4.10), if $\hat{\mathbf{a}}$ is the linear least squares solution, $\mathbf{r} = \mathbf{y} - C\hat{\mathbf{a}}$ and $s^2 = \mathbf{r}^{\mathrm{T}}\mathbf{r}/m$, the mean sum of squared residuals, the above expression can be written as

$$p(\mathbf{y}|\mathbf{a}, \eta) \propto \eta^{m/2} \exp\left\{-\frac{\eta}{2}\left[ms^2 + (\mathbf{a} - \hat{\mathbf{a}})^{\mathrm{T}}C^{\mathrm{T}}C(\mathbf{a} - \hat{\mathbf{a}})\right]\right\}.$$

Taking into account the prior distribution (4.13) for η we have

$$p(\mathbf{a}, \eta|\mathbf{y}) \propto \eta^{(m_0+m)/2-1} \exp\left\{-\frac{\eta}{2}\left[m_0\sigma_0^2 + ms^2 + (\mathbf{a} - \hat{\mathbf{a}})^{\mathrm{T}}C^{\mathrm{T}}C(\mathbf{a} - \hat{\mathbf{a}})\right]\right\}. \tag{4.14}$$

This distribution jointly describes the information about $\mathbf{a}$ and η derived from the data $\mathbf{y}$ and the prior information. If we are only interested in $\mathbf{a}$, then the posterior distribution $p(\mathbf{a}|\mathbf{y})$ is given by marginalisation (section 2.4.3):

$$p(\mathbf{a}|\mathbf{y}) = \int_0^\infty p(\mathbf{a}, \eta|\mathbf{y})\, d\eta.$$

This integration can be performed using the integration rule

$$\int_0^\infty \eta^{a-1} e^{-\eta b} d\eta = \Gamma(a) b^{-a}, \tag{4.15}$$

which leads to

$$p(\mathbf{a}|\mathbf{y}) \propto \left[m_0\sigma_0^2 + ms^2 + (\mathbf{a} - \hat{\mathbf{a}})^{\mathrm{T}}C^{\mathrm{T}}C(\mathbf{a} - \hat{\mathbf{a}})\right]^{-(m+m_0)/2}. \tag{4.16}$$

The multivariate t-distribution $t_\nu(\boldsymbol{\mu}, V)$ with mean n-vector $\boldsymbol{\mu}$, $n \times n$ scale matrix V and degrees of freedom ν has PDF

$$p_\nu(\mathbf{x}|\boldsymbol{\mu}, V) \propto \left[1 + \frac{1}{\nu}(\mathbf{x} - \boldsymbol{\mu})^T V^{-1}(\mathbf{x} - \boldsymbol{\mu})\right]^{-(\nu+n)/2}.$$

Comparing this PDF with (4.16), we see that the posterior distribution is $\mathbf{a}|\mathbf{y} \sim t_\nu(\hat{\mathbf{a}}, \bar{V})$ where

$$\nu = m_0 + m - n, \quad \bar{V} = \bar{\sigma}^2(C^{\mathrm{T}}C)^{-1}, \quad \bar{\sigma}^2 = \frac{m_0\sigma_0^2 + ms^2}{m_0 + m - n}.$$

We can also use marginalisation to determine the posterior distribution $p(\eta|\mathbf{y})$:

$$p(\eta|\mathbf{y}) = \int p(\mathbf{a}, \eta|\mathbf{y})\, d\mathbf{a}.$$

The term in (4.14) involving $\mathbf{a}$ is

$$\exp\left\{-\frac{\eta}{2}(\mathbf{a}-\hat{\mathbf{a}})^{\mathrm{T}}C^{\mathrm{T}}C(\mathbf{a}-\hat{\mathbf{a}})\right\}$$

and the integral of this function with respect to $\mathbf{a}$ defines the normalising constant

$$|2\pi\eta^{-1}(C^{\mathrm{T}}C)^{-1}|^{1/2} \propto \eta^{-n/2}$$

for the multivariate normal distribution with variance matrix $\left(\eta C^{\mathrm{T}}C\right)^{-1}$. (The term $|V|$ denotes the determinant of the square matrix V.) Thus,

$$p(\eta|\mathbf{y}) \propto \eta^{(m_0+m-n)/2-1} \exp\left\{-\frac{\eta}{2}\left[m_0\sigma_0^2 + ms^2\right]\right\}$$

which we recognise as the gamma distribution $G((m_0+m-n)/2, (m_0\sigma_0^2+ms^2)/2)$, or

$$(m_0+m-n)\bar{\sigma}^2\eta \sim \chi^2_{m_0+m-n}.$$

The posterior distribution η has the same form as the prior distribution for η.

4.1.6 Calculation of other quantities associated with the model fit

We summarise here the quantities associated with a linear least-squares fit that are often useful to calculate. It is assumed that the uncertainty matrix $V_{\mathbf{y}}$ associated with the data vector $\mathbf{y}$ is given by $V_{\mathbf{y}} = \sigma^2 I$.

- Estimates of the solution parameters $\mathbf{a} = (C^{\mathrm{T}}C)^{-1}C^{\mathrm{T}}\mathbf{y} = C^{\dagger}\mathbf{y}$.
- The model predictions $\hat{\mathbf{y}} = C\mathbf{a} = C(C^{\mathrm{T}}C)^{-1}C^{\mathrm{T}}\mathbf{y} = CC^{\dagger}\mathbf{y}$, i.e., the predicted responses $\hat{y}_i$ at values $\mathbf{x}_i$ of the covariates.
- The residual vector

  $$\mathbf{r} = \mathbf{y} - \hat{\mathbf{y}} = \mathbf{y} - C\mathbf{a} = (I - C(C^{\mathrm{T}}C)^{-1}C^{\mathrm{T}})\mathbf{y} = (I - CC^{\dagger})\mathbf{y},$$

  where I is the $m \times m$ identity matrix.
- The posterior estimate of the standard deviation of the random effects

  $$\hat{\sigma} = \|\mathbf{r}\|/(m-n)^{1/2}.$$

- The uncertainty (covariance) matrix associated with the fitted parameters. If an estimate of σ is available

  $$V_{\mathbf{a}} = \sigma^2(C^{\mathrm{T}}C)^{-1},$$

  otherwise, $V_{\mathbf{a}}$ can be obtained from

  $$V_{\mathbf{a}} = \hat{\sigma}^2(C^{\mathrm{T}}C)^{-1},$$

  where $\hat{\sigma}$ is given by (4.7).

- The standard uncertainties associated with the fitted parameters $u(a_j) = (V_{\mathbf{a}}(j,j))^{1/2}$, i.e., the square roots of the diagonal elements of the uncertainty matrix $V_{\mathbf{a}}$.
- The correlation matrix associated with the fitted parameters defined by

$$C_R(i,j) = \frac{V_{\mathbf{a}}(i,j)}{(V_{\mathbf{a}}(i,i)V_{\mathbf{a}}(j,j))^{1/2}}.$$

  Note that C_R is independent of the value of σ used to define the uncertainty matrix.
- The uncertainty (covariance) matrix $V_{\hat{\mathbf{y}}}$ associated with the model predictions $\hat{\mathbf{y}}$

$$V_{\hat{\mathbf{y}}} = CV_{\mathbf{a}}C^{\mathrm{T}} = \sigma^2 C(C^{\mathrm{T}}C)^{-1}C^{\mathrm{T}}.$$

- The standard uncertainties associated with the model predictions $u(\hat{y}_i) = (V_{\hat{\mathbf{y}}}(i,i))^{1/2}$.
- The uncertainty matrix $V_{\mathbf{r}}$ associated with the residuals

$$V_{\mathbf{r}} = \sigma^2(I - C(C^{\mathrm{T}}C)^{-1}C^{\mathrm{T}}).$$

- The standard uncertainties associated with the residual errors $u(r_i) = (V_{\mathbf{r}}(i,i))^{1/2}$. If $\mathbf{y}$ is associated with a multivariate normal distribution with variance matrix $V_{\mathbf{y}} = \sigma^2 I$, then the expected sum of the squares $\mathbf{r}^{\mathrm{T}}\mathbf{r}$ of the residuals is $(m-n)\sigma^2$. The uncertainties $u(r_i)$ are such that

$$\sum_{i=1}^{m} u^2(r_i) = (m-n)\sigma^2,$$

  and $u^2(r_i)$ is the expected value for the ith squared residual.
- If $(\mathbf{z}, w)$ represents a new data point (generated from the same model but not used in defining the model fit) then the predicted model value at $\mathbf{z}$ is

$$\hat{w} = \phi(\mathbf{z}, \mathbf{a}) = \mathbf{d}^{\mathrm{T}}\mathbf{a},$$

  where $\mathbf{d} = (d_1, \ldots, d_n)^T = (\phi_1(\mathbf{z}, \mathbf{a}), \ldots, \phi_n(\mathbf{z}, \mathbf{a}))^{\mathrm{T}}$, the standard uncertainty associated with $\hat{w}$ is

$$u(\hat{w}) = (\mathbf{d}^{\mathrm{T}}V_{\mathbf{a}}\mathbf{d})^{1/2},$$

  the predicted residual error is $t = w - \hat{w} = w - \mathbf{d}^{\mathrm{T}}\mathbf{a}$ and its variance is

$$V_t = \sigma^2 + \mathbf{d}^{\mathrm{T}}V_{\mathbf{a}}\mathbf{d}.$$

  More generally, if $Z = \{\mathbf{z}_q\}_{q=1}^{m_Z}$ is a range of values for the covariates and D is the corresponding matrix of basis functions evaluated at $\mathbf{z}_q$, i.e.,

$$D_{q,j} = \phi_j(\mathbf{z}_q),$$

  then the uncertainty matrix $V_{\mathbf{w}}$ associated with the model values $\mathbf{w} = (w_1, \ldots, w_{m_Z})^{\mathrm{T}}$, $w_q = \phi(\mathbf{z}_q, \mathbf{a})$, is

$$V_{\mathbf{w}} = DV_{\mathbf{a}}D^{\mathrm{T}},$$

  and the standard uncertainty $u(w_q)$ is

$$u(w_q) = (V_{\mathbf{w}}(q,q))^{1/2}.$$

We note that if the observation matrix has QR factorisation

$$C = QR = [Q_1\ Q_2] \begin{bmatrix} R_1 \\ \mathbf{0} \end{bmatrix} = Q_1 R_1,$$

where $Q = [Q_1\ Q_2]$ is an $m \times m$ orthogonal matrix and R_1 is an $n \times n$ upper triangular matrix and singular value decomposition (SVD) $C = U_1 S_1 V^{\mathrm{T}}$ where U_1 is an $m \times n$ orthogonal matrix, S_1 is $n \times n$ diagonal matrix and V is $n \times n$ orthogonal matrix, then

$$\begin{aligned}
C^{\mathrm{T}}C &= R_1^{\mathrm{T}} R_1 = V S_1^2 V^{\mathrm{T}}, \\
(C^{\mathrm{T}}C)^{-1} &= R_1^{-1} R_1^{-\mathrm{T}} = V S_1^{-2} V^{\mathrm{T}}, \\
(C^{\mathrm{T}}C)^{-1} C^{\mathrm{T}} &= C^{\dagger} = R_1^{-1} Q_1^{\mathrm{T}} = V S_1^{-1} U_1^{\mathrm{T}}, \quad \text{and} \\
C(C^{\mathrm{T}}C)^{-1} C^{\mathrm{T}} &= CC^{\dagger} = Q_1 Q_1^{\mathrm{T}} = U_1 U_1^{\mathrm{T}}, \\
I - C(C^{\mathrm{T}}C)^{-1} C^{\mathrm{T}} &= I - CC^{\dagger} = I - Q_1 Q_1^{\mathrm{T}} = I - U_1 U_1^{\mathrm{T}} = Q_2 Q_2^{\mathrm{T}} = U_2 U_2^{\mathrm{T}}.
\end{aligned}$$

These relations show that all the model outputs listed above can be calculated from QR factorisation or SVD of C. All the statistical information can be derived from $V_{\mathbf{a}}$.

4.1.7 Weighted linear least-squares estimator

If the random effects ϵ_i are uncorrelated but drawn from distributions with different standard deviations, e.g., $\epsilon_i \in N(0, \sigma_i^2)$ then the appropriate estimator is a weighted linear least-squares estimator which estimates $\mathbf{a}$ by solving

$$\min_{\mathbf{a}} \sum_{i=1}^{m} w_i^2 (y_i - \mathbf{c}_i^{\mathrm{T}} \mathbf{a})^2, \tag{4.17}$$

with $w_i = 1/\sigma_i$. Algorithms for the unweighted linear least squares problem can be easily adapted to deal with the weighted case by applying them to

$$\tilde{y}_i = w_i y_i, \quad \tilde{C}(i,j) = w_i C(i,j).$$

In this case the uncertainty matrix associated with the solution parameters is

$$V_{\mathbf{a}} = (\tilde{C}^{\mathrm{T}} \tilde{C})^{-1}.$$

Weighted linear least squares and MLE. Just as the linear least-squares solution is the ML estimate for the model $\mathbf{y} \in N(C\mathbf{a}, \sigma^2 I)$, the weighted linear least-squares is the ML estimate for the model $\mathbf{y} \in N(C\mathbf{a}, D)$, where D is the diagonal matrix with σ_i^2 in the ith diagonal position. Here, we assume σ_i is known, $i = 1, \ldots, m$. The likelihood of observing $\mathbf{y}$, given $\mathbf{a}$ is given by

$$p(\mathbf{y}|\mathbf{a}) \propto \exp\left\{ -\frac{1}{2} (\mathbf{y} - C\mathbf{a})^{\mathrm{T}} D^{-1} (\mathbf{y} - C\mathbf{a}) \right\},$$

which is maximised by the solution of (4.17).

4.1.8 Gauss-Markov estimator

More generally, if the vector of random effects are modelled as belonging to a multivariate distribution with uncertainty (covariance) matrix V, assumed to be full rank, the Gauss-Markov estimator which solves

$$\min_{\mathbf{a}} \ (\mathbf{y} - C\mathbf{a})^{\mathrm{T}} V^{-1} (\mathbf{y} - C\mathbf{a}), \tag{4.18}$$

is appropriate. The Gauss-Markov estimate is the ML estimate of $\mathbf{a}$ for the model $\mathbf{y} \in N(C\mathbf{a}, V)$. If V has a Cholesky decomposition $V = LL^{\mathrm{T}}$, then the Gauss-Markov estimate can be determined by applying the linear least-squares estimator to

$$\tilde{\mathbf{y}} = L^{-1}\mathbf{y}, \quad \tilde{C} = L^{-1}C.$$

Generalised QR factorisation approach. The generalised QR decomposition can be employed to solve (4.18) avoiding the calculation of the inverse of a matrix, often a cause of numerical instability [27, 68, 121, 192]. For a general (full rank) uncertainty matrix V with a factorisation $V = LL^{\mathrm{T}}$, where L is an $m \times m$ matrix, also necessarily full rank, the least-squares estimate is given by

$$\mathbf{a} = \tilde{C}^{\dagger}\tilde{\mathbf{y}}, \quad \tilde{C} = L^{-1}C, \ \tilde{y} = L^{-1}\mathbf{y}, \tag{4.19}$$

where $\tilde{C}^{\dagger}$ is the pseudo-inverse of $\tilde{C}$. For well conditioned V and L, this approach is satisfactory. However, if L is poorly conditioned the formation and use of $\tilde{C}$, etc., can be expected to introduce numerical errors. The *generalised QR factorisation* [68, 121, 181, 192] approach avoids this potential numerical instability. Suppose $V = LL^{\mathrm{T}}$, where L is $m \times p$. (Often $p = m$ but the approach applies in the more general case. Often, an uncertainty matrix is naturally expressed in factored form.) The estimate $\mathbf{a}$ can be found by solving

$$\min_{\mathbf{a},\mathbf{e}} \ \mathbf{e}^{\mathrm{T}}\mathbf{e} \quad \text{subject to constraints} \quad \mathbf{y} = C\mathbf{a} + L\mathbf{e}. \tag{4.20}$$

Note that if L is invertible,

$$\mathbf{e} = L^{-1}(\mathbf{y} - C\mathbf{a}), \quad \mathbf{e}^{\mathrm{T}}\mathbf{e} = (\mathbf{y} - C\mathbf{a})^{\mathrm{T}} V^{-1} (\mathbf{y} - C\mathbf{a}).$$

We factorise $C = QR$ and $Q^T L = TU$ where R and T are upper-triangular and Q and U are orthogonal. Multiplying the constraints by Q^{T}, we have

$$\begin{bmatrix} \tilde{\mathbf{y}}_1 \\ \tilde{\mathbf{y}}_2 \end{bmatrix} = \begin{bmatrix} R_1 \\ \mathbf{0} \end{bmatrix} \mathbf{a} + \begin{bmatrix} T_{11} & T_{12} \\ & T_{22} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{e}}_1 \\ \tilde{\mathbf{e}}_2 \end{bmatrix}, \tag{4.21}$$

where $\tilde{\mathbf{y}} = Q^{\mathrm{T}}\mathbf{y}$, and $\tilde{\mathbf{e}} = U\mathbf{e}$.

From the second set of equations, $\tilde{\mathbf{e}}_2$ must satisfy $\tilde{\mathbf{y}}_2 = T_{22}\tilde{\mathbf{e}}_2$.

Given any $\tilde{\mathbf{e}}_1$, the first set of equations is satisfied if $R_1\mathbf{a} = \tilde{\mathbf{y}}_1 - T_{11}\tilde{\mathbf{e}}_1 - T_{12}\tilde{\mathbf{e}}_2$.

We choose $\tilde{\mathbf{e}}_1 = 0$ in order to minimise

$$\mathbf{e}^{\mathrm{T}}\mathbf{e} = \tilde{\mathbf{e}}^{\mathrm{T}}\tilde{\mathbf{e}} = \tilde{\mathbf{e}}_1^{\mathrm{T}}\tilde{\mathbf{e}}_1 + \tilde{\mathbf{e}}_2^{\mathrm{T}}\tilde{\mathbf{e}}_2,$$

so that $\mathbf{a}$ solves $R_1\mathbf{a} = \tilde{\mathbf{y}}_1 - T_{12}\tilde{\mathbf{e}}_2$.

Public-domain library software for solving (4.20) and, more generally, computing generalised QR factorisations is available [192].

Uncertainty matrix associated with the Gauss-Markov estimate. If $\mathbf{a}$ is the Gauss-Markov estimate, the associated uncertainty matrix $V_{\mathbf{a}}$ is given by

$$V_{\mathbf{a}} = (C^{\mathrm{T}} V^{-1} C)^{-1}.$$

Details. In terms of the generalised QR factorisation [68],

$$V_{\mathbf{a}} = KK^{\mathrm{T}} \quad \text{where } K \text{ solves } R_1 K = T_{11}.$$

Gauss Markov estimator and MLE. If $\mathbf{y} \in N(C\mathbf{a}, V)$, then the ML estimate of $\mathbf{a}$ is given by the Gauss-Markov estimate $\hat{\mathbf{a}}$. In a Bayesian context, if the prior distribution for $p(\mathbf{a}) = 1$, then the posterior distribution $p(\mathbf{a}|\mathbf{y})$ is $\mathbf{a} \sim N(\hat{\mathbf{a}}, V_{\mathbf{a}})$. This follows from the linear least squares analysis, but applied to the matrix $\tilde{C}$ and data vector $\tilde{\mathbf{y}}$ in (4.19).

Comparison of the Gauss-Markov and linear least squares estimators. The Gauss-Markov estimator does require for its reliable implementation more technical algorithmic components that the standard linear least squares estimator. For example, the GM estimator can be implemented using the generalised QR factorisation (GQR), while the LLS estimator only requires the QR factorisation. The GQR factorisation also requires a number of steps of the order of m^3 where m is the number of data points whereas the QR factorisation can be achieved in order m steps (but see section 4.1.9 for ways to exploit structure in GM problems to make the computation order m). What is to be gained by using the GM estimator?

Suppose the model is $\mathbf{y} \in N(C\mathbf{a}, V)$ and let $\hat{\mathbf{a}}_{GM}$ and $\hat{\mathbf{a}}_{LLS}$ be the GM and LLS estimates:

$$\hat{\mathbf{a}}_{GM} = (C^{\mathrm{T}} V^{-1} C)^{-1} C^{\mathrm{T}} V^{-1} \mathbf{y}, \quad \hat{\mathbf{a}}_{LLS} = (C^{\mathrm{T}} C)^{-1} C^{\mathrm{T}} \mathbf{y}.$$

The uncertainty matrices associated with these estimates are

$$V_{GM} = (C^{\mathrm{T}} V^{-1} C)^{-1}, \quad V_{LLS} = C^{\dagger} V (C^{\dagger})^{\mathrm{T}}, \quad C^{\dagger} = (C^{\mathrm{T}} C)^{-1} C^{\mathrm{T}}.$$

Here V_{LLS} is calculated using the law of propagation of uncertainty as in (4.5); if $V = \sigma^2 I$ then $V_{LLS} = \sigma^2 \left(C^{\mathrm{T}} C\right)^{-1}$. We show below that the matrix V_{LLS} is larger than V_{GM} in the sense that $V_{LLS} - V_{GM}$ is a positive semi-definite matrix, so that for any n-vector $\mathbf{x}$,

$$\mathbf{x}^{\mathrm{T}} (V_{LLS} - V_{GM}) \mathbf{x} \geq 0.$$

This statement can interpreted as saying that, for any $\mathbf{x}$, the uncertainty associated with the linear combination $\mathbf{x}^{\mathrm{T}} \hat{\mathbf{a}}_{LLS}$ is greater than or equal to that associated with $\mathbf{x}^{\mathrm{T}} \hat{\mathbf{a}}_{GM}$.

From a Bayesian point of view, the posterior distribution $p(\mathbf{a}|\mathbf{y})$ is $\mathbf{a}|\mathbf{y} \sim N(\hat{\mathbf{a}}_{GM}, V_{GM})$. The probability of observing the GM estimate, given $\mathbf{a}$ is $\hat{\mathbf{a}}_{GM}|\mathbf{a} \sim N(\mathbf{a}, V_{GM})$. In the absence of substantive prior knowledge about $\mathbf{a}$, Bayes' Theorem tells us that

$$p(\mathbf{a}|\hat{\mathbf{a}}_{GM}) \propto p(\hat{\mathbf{a}}_{GM}|\mathbf{a}).$$

Using the symmetry with respect to $\mathbf{a}$ and $\hat{\mathbf{a}}_{GM}$ in the corresponding normal distributions, i.e., $\mathbf{a}$ and $\hat{\mathbf{a}}_{GM}$ appear in both distributions through the common term

$$(\mathbf{a} - \hat{\mathbf{a}}_{GM})^{\mathrm{T}} V_{GM}^{-1} (\mathbf{a} - \hat{\mathbf{a}}_{GM}),$$

it follows that $\mathbf{a}|\hat{\mathbf{a}}_{GM} \sim N(\hat{\mathbf{a}}_{GM}, V_{GM})$ and $p(\mathbf{a}|\hat{\mathbf{a}}_{GM}) = p(\mathbf{a}|\mathbf{y})$. In other words, the GM $\hat{\mathbf{a}}_{GM}$ estimate provides the same information about $\mathbf{a}$ as the data vector $\mathbf{y}$. The same symmetry argument shows us that

$$p(\mathbf{a}|\hat{\mathbf{a}}_{LLS}) = p(\hat{\mathbf{a}}_{LLS}|\mathbf{a}),$$

so that $\mathbf{a}|\hat{\mathbf{a}}_{LLS} \sim N(\hat{\mathbf{a}}_{LLS}, V_{LLS})$ which, in general, will be different from $p(\mathbf{a}|\mathbf{y})$; the LLS estimate in general provides less information about $\mathbf{a}$ than the data vector $\mathbf{y}$.

Details. Let C have QR factorisation $C = QR = Q_1 R_1$, where Q_1 is the matrix given by the first n columns of the orthogonal matrix Q and R_1 is the $n \times n$ upper triangle of R. Then,

$$V_{GM} = R_1^{-1} \left(Q_1^{\mathrm{T}} V^{-1} Q_1 \right)^{-1} R_1^{-\mathrm{T}}, \quad V_{LLS} = R_1^{-1} \left(Q_1^{\mathrm{T}} V Q_1 \right) R_1^{-\mathrm{T}},$$

and

$$V_{LLS} - V_{GM} = R_1^{-1} \left[\left(Q_1^{\mathrm{T}} V Q_1 \right) - \left(Q_1^{\mathrm{T}} V^{-1} Q_1 \right)^{-1} \right] R_1^{-\mathrm{T}}.$$

To show that $V_{LLS} - V_{GM}$ is positive semi-definite, it is sufficient to show that the term in the square brackets is positive semi-definite. Let $W = Q^{\mathrm{T}} V Q$, and partition and factor W as

$$W = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} = LL^{\mathrm{T}} = \begin{bmatrix} L_{11} & \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} L_{11}^{\mathrm{T}} & L_{21}^{\mathrm{T}} \\ & L_{22}^{\mathrm{T}} \end{bmatrix},$$

where W_{11} is the upper-left $n \times n$ submatrix of W, etc. Then $V = QWQ^{\mathrm{T}}$ and

$$Q_1^{\mathrm{T}} V Q_1 = W_{11} = L_{11} L_{11}^{\mathrm{T}}. \tag{4.22}$$

Similarly $V^{-1} = QW^{-1}Q^{\mathrm{T}}$ and $Q_1^{\mathrm{T}} V^{-1} Q_1$ is the upper-left $n \times n$ submatrix of W^{-1} with

$$W^{-1} = L^{-\mathrm{T}} L^{-1} = K^{\mathrm{T}} K = \begin{bmatrix} K_{11}^{\mathrm{T}} & K_{21}^{\mathrm{T}} \\ & K_{22}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} K_{11} & \\ K_{21} & K_{22} \end{bmatrix}.$$

where

$$K_{11} = L_{11}^{-1}, \quad K_{22} = L_{22}^{-1}, \quad K_{21} = -L_{22}^{-1} L_{21} L_{11}^{-1}.$$

Therefore,

$$\begin{aligned} Q_1^{\mathrm{T}} V^{-1} Q_1 = K_{11}^{\mathrm{T}} K_{11} + K_{21}^{\mathrm{T}} K_{21} &= L_{11}^{-\mathrm{T}} L_{11}^{-1} + L_{11}^{-\mathrm{T}} L_{21}^{\mathrm{T}} L_{22}^{-\mathrm{T}} L_{22}^{-1} L_{21} L_{11}^{-1}, \\ &= L_{11}^{-\mathrm{T}} \left[I + (L_{22}^{-1} L_{21})^{\mathrm{T}} (L_{22}^{-1} L_{21}) \right] L_{11}^{-1}. \end{aligned}$$

Combining this result with (4.22),

$$Q_1^{\mathrm{T}} V Q_1 - (Q_1^{\mathrm{T}} V^{-1} Q_1)^{-1} = L_{11} \left[I - (I + B^{\mathrm{T}} B)^{-1} \right] L_{11}^{\mathrm{T}}, \quad B = L_{22}^{-1} L_{21}.$$

Since

$$(I + B^{\mathrm{T}} B)^{-1} = I - B^{\mathrm{T}} (I + B B^{\mathrm{T}})^{-1} B,$$

$$Q_1^{\mathrm{T}} V Q_1 - (Q_1^{\mathrm{T}} V^{-1} Q_1)^{-1} = L_{11} \left[B^{\mathrm{T}} (I + B B^{\mathrm{T}})^{-1} B \right] L_{11}^{T},$$

and the term on the right is necessarily positive semi-definite.

4.1.9 Structured Gauss-Markov problems

While Gauss-Markov regression problems arise often in practice (although correlated effects are commonly ignored), the uncertainty matrix V can usually be specified more compactly in factored form. For example suppose the random effects associated with the measurements are modelled as

$$y_i = \mathbf{c}_i^{\mathrm{T}}\mathbf{a} + \epsilon_i + \mathbf{h}_{i,0}^{\mathrm{T}}\boldsymbol{\epsilon}_0.$$

Here, ϵ_i, represents the random effect particular to the ith measurement and $\boldsymbol{\epsilon}_0 = (\epsilon_{1,0}, \ldots, \epsilon_{k,0})^{\mathrm{T}}$ those common to all the measurements. If ϵ_i and $\boldsymbol{\epsilon}_0$ are assigned Gaussian distributions so that $\epsilon_i \in N(0, \sigma^2)$ and $\boldsymbol{\epsilon}_0 \in N(0, U_0)$, then the uncertainty matrix V associated with the data vector $\mathbf{y}$ is given by

$$V = \sigma^2 I + HU_0H^{\mathrm{T}},$$

where H is the $m \times k$ matrix whose ith row is $\mathbf{h}_{i,0}^{\mathrm{T}}$. The matrix V (and its Cholesky factor) is a full matrix by virtue of the common effects $\boldsymbol{\epsilon}_0$. Estimates of $\mathbf{a}$ are found by solving the Gauss-Markov problem,

$$\min_{\mathbf{a}}(\mathbf{y} - C\mathbf{a})V^{-1}(\mathbf{y} - C\mathbf{a}), \tag{4.23}$$

using the techniques described in section 4.1.8. However, if $D = \sigma I$, U_0 has Cholesky factorisation $U_0 = L_0L_0^{\mathrm{T}}$, $B_0 = HL_0$, then V can be factored as

$$V = BB^{\mathrm{T}}, \quad B = \begin{bmatrix} D & B_0 \end{bmatrix},$$

and (4.23) has the same solution as

$$\min_{\mathbf{a},\mathbf{e},\mathbf{e}_0} \mathbf{e}^{\mathrm{T}}\mathbf{e} + \mathbf{e}_0^{\mathrm{T}}\mathbf{e}_0 \quad \text{subject to} \quad \mathbf{y} = C\mathbf{a} + D\mathbf{e} + B_0\mathbf{e}_0. \tag{4.24}$$

Details. To see this equivalence, note that if B^{T} has QR factorisation

$$B^{\mathrm{T}} = PS = \begin{bmatrix} P_1 & P_2 \end{bmatrix} \begin{bmatrix} S_1 \\ \mathbf{0} \end{bmatrix},$$

where P is an $(m+k) \times m$ orthogonal matrix and S_1 is an $m \times m$ upper triangular matrix, then

$$V = BB^{\mathrm{T}} = S^{\mathrm{T}}P^{\mathrm{T}}PS = S_1^{\mathrm{T}}S_1,$$

so that S_1^{T} is the Cholesky factor of V. (A Cholesky factor is unique up to the sign of the columns.) Writing

$$\begin{bmatrix} \mathbf{e} \\ \mathbf{e}_0 \end{bmatrix} = P \begin{bmatrix} \tilde{\mathbf{e}} \\ \tilde{\mathbf{e}}_0 \end{bmatrix},$$

then

$$D\mathbf{e} + B_0\mathbf{e}_0 = B \begin{bmatrix} \mathbf{e} \\ \mathbf{e}_0 \end{bmatrix} = S^{\mathrm{T}}P^{\mathrm{T}}P \begin{bmatrix} \tilde{\mathbf{e}} \\ \tilde{\mathbf{e}}_0 \end{bmatrix} = S_1^{\mathrm{T}}\tilde{\mathbf{e}},$$

so that the constraint in (4.24) is equivalent to

$$\mathbf{y} = C\mathbf{a} + S_1^{\mathrm{T}}\tilde{\mathbf{e}} \quad \text{or} \quad \tilde{\mathbf{e}} = S_1^{-\mathrm{T}}(\mathbf{y} - C\mathbf{a}).$$

This means that (4.24) is equivalent to

$$\min_{\mathbf{a},\mathbf{e},\mathbf{e}_0} \mathbf{e}^{\mathrm{T}}\mathbf{e} + \mathbf{e}_0^{\mathrm{T}}\mathbf{e}_0 \ = \tilde{\mathbf{e}}^{\mathrm{T}}\tilde{\mathbf{e}} + \tilde{\mathbf{e}}_0^{\mathrm{T}}\tilde{\mathbf{e}}_0,$$

subject to $\tilde{\mathbf{e}} = S_1^{-\mathrm{T}}(\mathbf{y} - C_{\mathbf{a}})$ and is solved by the $\mathbf{a}$ that minimises

$$\tilde{\mathbf{e}}^{\mathrm{T}}\tilde{\mathbf{e}} = (\mathbf{y} - C\mathbf{a})^{\mathrm{T}}S_1^{-1}S_1^{-\mathrm{T}}(\mathbf{y} - C\mathbf{a}) = (\mathbf{y} - C\mathbf{a})^{\mathrm{T}}V^{-1}(\mathbf{y} - C\mathbf{a}),$$

(with $\tilde{\mathbf{e}}_0 = \mathbf{0}$).

The optimisation problem (4.24) can be written as

$$\min_{\mathbf{a},\mathbf{e},\mathbf{e}_0} \mathbf{e}^{\mathrm{T}}\mathbf{e} + \mathbf{e}_0^{\mathrm{T}}\mathbf{e}_0, \quad \mathbf{e} = D^{-1}\left(\mathbf{y} - C\mathbf{a} - B_0\mathbf{e}_0\right),$$

so that if

$$\tilde{C} = \begin{bmatrix} D^{-1}C & D^{-1}B_0 \\ \mathbf{0} & I \end{bmatrix}, \quad \tilde{\mathbf{y}} = \begin{bmatrix} D^{-1}\mathbf{y} \\ \mathbf{0} \end{bmatrix}, \quad \tilde{\mathbf{a}} = \begin{bmatrix} \mathbf{a} \\ \mathbf{e}_0 \end{bmatrix},$$

then (4.24), and hence (4.23), is equivalent to the standard linear least squares problem

$$\min_{\tilde{\mathbf{a}}} (\tilde{\mathbf{y}} - \tilde{C}\tilde{\mathbf{a}})^{\mathrm{T}}(\tilde{\mathbf{y}} - \tilde{C}\tilde{\mathbf{a}}).$$

By introducing the parameters $\mathbf{e}_0$ explicitly into the optimisation to explain the correlating effects, a simpler and more efficient solution method can be implemented. In the example above, $D = \sigma I$, and the approach can be extended to any D that is well-conditioned and for which it is computationally efficient to compute $D^{-1}C$, etc.

4.1.10 Linear least squares subject to linear equality constraints

Linear equality constraints of the form $D\mathbf{a} = \mathbf{d}$ where D is a $p \times n$ matrix, $p < n$, can be treated using an orthogonal factorisation approach. Such constraints arise in the application of resolving constraints to remove degrees of freedom from the model (section 2.3.3).

Suppose D^{T} is of full column rank and has the QR factorisation

$$D^{\mathrm{T}} = US = [U_1\ U_2] \begin{bmatrix} S_1 \\ 0 \end{bmatrix} = U_1 S_1, \tag{4.25}$$

where U_1 and U_2 represent the first p and last $n - p$ columns of the orthogonal factor U. If $\mathbf{a}_0$ is any solution of $D\mathbf{a} = \mathbf{d}$, then for any $(n - p)$-vector $\tilde{\mathbf{a}}$, $\mathbf{a} = \mathbf{a}_0 + U_2\tilde{\mathbf{a}}$ automatically satisfies the constraints:

$$D\mathbf{a} = D\mathbf{a}_0 + DU_2\tilde{\mathbf{a}} = \mathbf{d} + S_1^{\mathrm{T}}U_1^{\mathrm{T}}U_2\tilde{\mathbf{a}} = \mathbf{d},$$

since $U_1^{\mathrm{T}}U_2 = \mathbf{0}$. The optimisation problem

$$\min_{\mathbf{a}} \|\mathbf{y} - C\mathbf{a}\|_2^2 \quad \text{subject to} \quad D\mathbf{a} = \mathbf{d},$$

can be reformulated as the unconstrained linear least-squares problem

$$\min_{\tilde{\mathbf{a}}}\ \|\mathbf{y} - C(\mathbf{a}_0 + U_2\tilde{\mathbf{a}})\|_2^2 = \min_{\tilde{\mathbf{a}}}\ \|\tilde{\mathbf{y}} - \tilde{C}\tilde{\mathbf{a}}\|_2^2,$$

where

$$\tilde{\mathbf{y}} = \mathbf{y} - C\mathbf{a}_0, \quad \tilde{C} = CU_2.$$

This approach to treating linear equality constraints is quite general and can be applied to different types of optimisation problems. It is straightforward to show that $\mathbf{a}_0 = U_1 S_1^{-\mathrm{T}}\mathbf{d}$ satisfies the constraints i.e., $D\mathbf{a}_0 = \mathbf{d}$, and in fact is the vector of minimum norm that does so, i.e., $\mathbf{a}_0$ solves

$$\min \|\mathbf{a}_0\|_2^2 \quad \text{subject to} \quad D\mathbf{a}_0 = \mathbf{d}.$$

Uncertainty matrix associated with the linearly constrained LLS estimate. The constrained solution parameters are given by $\tilde{\mathbf{a}} = \tilde{C}^{\dagger}\tilde{\mathbf{y}}$ where $\tilde{C}^{\dagger} = (\tilde{C}^{\mathrm{T}}\tilde{C})^{-1}\tilde{C}^{\mathrm{T}}$ is the pseudo-inverse of $\tilde{C}$ (section 4.1.3). If $V_{\mathbf{y}}$ is the uncertainty matrix associated with $\mathbf{y}$, then the uncertainty matrix $V_{\tilde{\mathbf{a}}}$ associated with $\tilde{\mathbf{a}}$ is given by

$$V_{\tilde{\mathbf{a}}} = \tilde{C}^{\dagger}V_{\mathbf{y}}(\tilde{C}^{\dagger})^{\mathrm{T}}.$$

Here, we have used that fact that $V_{\tilde{\mathbf{y}}} = V_{\mathbf{y}}$, which follows from $\tilde{\mathbf{y}} = \mathbf{y} - C\mathbf{a}_0$. In particular, if $V_{\mathbf{y}} = \sigma^2 I$, then

$$V_{\tilde{\mathbf{a}}} = \sigma^2(\tilde{C}^T\tilde{C})^{-1}.$$

Since the full set of parameters are given by $\mathbf{a} = \mathbf{a}_0 + U_2\tilde{\mathbf{a}}$, the uncertainty matrix $V_{\mathbf{a}}$ associated with $\mathbf{a}$ is

$$V_{\mathbf{a}} = U_2 V_{\tilde{\mathbf{a}}} U_2^{\mathrm{T}}.$$

4.1.11 The Kalman filter

The *Kalman filter* is an efficient method for providing the solution to a structured linear least-squares problem involving a sequence of parameter vectors $\mathbf{a}_k \in \mathcal{R}^n$, $k = 1, 2, \ldots$. The parameters $\mathbf{a}_k$ often represent the state of a system at time k and it is required to determine as accurately as possible the current state, taking into account information available from the past. Information about $\mathbf{a}_k$ comes from two sources, predictive information of the form

$$\mathbf{a}_k = B_{k-1}\mathbf{a}_{k-1} + \boldsymbol{\delta}_k, \quad \boldsymbol{\delta}_k \in N(\mathbf{0}, W_k), \quad k > 1,$$

and measurement information:

$$A_k\mathbf{a}_k = \mathbf{y}_k + \boldsymbol{\epsilon}_k, \quad \boldsymbol{\epsilon}_k \in N(\mathbf{0}, V_k), \quad k > 0.$$

In the above, A_k and B_k are known observation matrices and V_k and W_k known uncertainty (variance) matrices. B_k and W_k are necessarily $n \times n$ matrices and A_k and V_k are $p_k \times n$ and $p_k \times p_k$ matrices, respectively. The role of $\boldsymbol{\delta}_k$ is to model the fact that the predictive information is not perfect and that the (actual) state at the kth stage is not determined purely by the (actual) state at the $(k-1)$th stage.

In a standard formulation of the Kalman filter, the estimate $\hat{\mathbf{a}}_k$ of $\mathbf{a}_k$ is obtained in a two-stage process. Suppose that the information about $\mathbf{a}_{k-1}$ is summarised by $\mathbf{a}_{k-1} \sim N(\hat{\mathbf{a}}_{k-1}, U_{k-1})$. If B_{k-1} is full rank then the first stage estimate $\bar{\mathbf{a}}_k$ of $\mathbf{a}_k$ is given by $\bar{\mathbf{a}}_k = B_{k-1}\mathbf{a}_k$. The uncertainty $\bar{U}_k$ associated with this estimate is

$$\bar{U}_k = B_{k-1}U_{k-1}B_{k-1}^T + W_k,$$

reflecting the uncertainty associated with $\hat{\mathbf{a}}_k$ and that associated with the prediction. The measurement information $\mathbf{y}_k$ is used to update this estimate. Setting

$$C_k = \begin{bmatrix} I \\ A_k \end{bmatrix}, \quad \mathbf{z}_k = \begin{bmatrix} \bar{\mathbf{a}}_k \\ \mathbf{y}_k \end{bmatrix}, \quad V_{\mathbf{z}_k} = \begin{bmatrix} \bar{U}_k & \\ & V_k \end{bmatrix}, \tag{4.26}$$

estimates $\hat{\mathbf{a}}_k$ are found by solving the Gauss-Markov problem associated with the model $\mathbf{z}_k \in N(C_k\mathbf{a}_k, V_{\mathbf{z}_k})$ with

$$\hat{\mathbf{a}}_k = \left(C_k^{\mathrm{T}}V_{\mathbf{z}_k}^{-1}C_k\right)^{-1} C_k^{\mathrm{T}}V_{\mathbf{z}_k}^{-1}\mathbf{z}_k. \tag{4.27}$$

The uncertainty matrix associated with this estimate is

$$U_k = \left(C_k^{\mathrm{T}} V_{\mathbf{z}_k}^{-1} C_k\right)^{-1}.$$

The process can be repeated, now starting with $\hat{\mathbf{a}}_k$ and U_k and incorporating measurement information $\mathbf{y}_{k+1}$.

The calculations can be organised (see below for details) so that $\hat{\mathbf{a}}_k$ can be expressed as

$$\hat{\mathbf{a}}_k = \bar{\mathbf{a}}_k + K_k(\mathbf{y}_k - A_k\bar{\mathbf{a}}_k), \tag{4.28}$$

where

$$K_k = \bar{U}_k A_k^{\mathrm{T}} \left(A_k \bar{U}_k A_k^{\mathrm{T}} + V_k\right)^{-1}, \tag{4.29}$$

is the *Kalman gain*. The Kalman gain specifies how much the prediction $\bar{\mathbf{a}}_k$ needs to be modified in light of the discrepancy between the prediction $A_k\bar{\mathbf{a}}_k$ and the measured values $\mathbf{y}_k$. The uncertainty matrix U_k can also be specified in terms of the Kalman gain:

$$U_k = (I - K_k A_k)\bar{U}_k.$$

Details. The uncertaint matrix U_k can be expanded as

$$U_k = \left(C_k^{\mathrm{T}} V_{\mathbf{z}_k}^{-1} C_k\right)^{-1} = \left(\bar{U}_k^{-1} + A^{\mathrm{T}} V_k^{-1} A_k\right)^{-1}.$$

Using the identity

$$\left(A + B^{\mathrm{T}} C B\right)^{-1} = A^{-1} - A^{-1}B^{\mathrm{T}}\left(BA^{-1}B^{\mathrm{T}} + C^{-1}\right)^{-1} BA^{-1},$$

for symmetric A and C, we have

$$U_k = \bar{U}_k - \bar{U}_k A_k^{\mathrm{T}}\left(A_k\bar{U}_k A_k^{\mathrm{T}} + V_k\right)^{-1} A_k\bar{U}_k = (I - K_k A_k)\bar{U}_k.$$

Furthermore,

$$V_{\mathbf{z}_k}^{-1} C_k^{\mathrm{T}} \mathbf{z}_k = \bar{U}_k^{-1}\bar{\mathbf{a}}_k + A_k V_k^{-1}\mathbf{y}_k.$$

Comparing (4.27) with (4.28), we need to show that

$$(I - K_k A_k)\bar{U}_k\left(\bar{U}_k^{-1}\bar{\mathbf{a}}_k + A_k V_k^{-1}\mathbf{y}_k\right) = \bar{\mathbf{a}}_k + K_k(\mathbf{y}_k - A_k\bar{\mathbf{a}}),$$

or that

$$(I - K_k A_k)\bar{U}_k A_k^{\mathrm{T}} V_k^{-1} = K_k.$$

Using (4.29),

$$\begin{aligned}
(I - K_k A_k)\bar{U}_k A_k^{\mathrm{T}} V_k^{-1} - K_k &= \bar{U}_k A_k^{\mathrm{T}} V_k^{-1} - K_k\left(I + A_k\bar{U}_k A_k^{\mathrm{T}} V_k^{-1}\right), \\
&= \bar{U}_k A_k^{\mathrm{T}} V_k^{-1} - \bar{U}_k A_k^{\mathrm{T}}\left(A_k\bar{U}_k A_k^{\mathrm{T}} + V_k\right)^{-1}\left(I + A_k\bar{U}_k A_k^{\mathrm{T}} V_k^{-1}\right), \\
&= \bar{U}_k A_k^{\mathrm{T}}\left(V_k^{-1} - \left(A_k\bar{U}_k A_k^{\mathrm{T}} + V_k\right)^{-1}\left(A_k\bar{U}_k A_k^{\mathrm{T}} + V_k\right) V_k^{-1}\right), \\
&= \mathbf{0},
\end{aligned}$$

as required.

A QR implementation of the Kalman filter. The estimate $\hat{\mathbf{a}}_k$ can be determined using a QR factorisation approach. If U_{k-1} and W_k can be factored as $U_{k-1} = L_{k-1}L_{k-1}^{\mathrm{T}}$ and $W_k = J_k J_k^{\mathrm{T}}$ then the uncertainty matrix $\bar{U}_k$ can be factored as

$$\bar{U}_{k-1} = \bar{L}_k \bar{L}_k^{\mathrm{T}}, \quad \bar{L}_k = [B_k L_{k-1} \ J_k].$$

Using the QR decomposition, $\bar{L}_k^{\mathrm{T}}$ can be factored as $\bar{L}_k^{\mathrm{T}} = \bar{Q}_k \tilde{L}_k^{\mathrm{T}}$ where $\tilde{L}_k$ is an $n \times n$ lower triangular matrix and $\bar{Q}_k$ is an orthogonal matrix. This factorisation determines the Cholesky factorisation of $\bar{U}_k$ as $\bar{U}_k = \tilde{L}_k \tilde{L}_k^{\mathrm{T}}$. Following the notation of (4.26), if V_k is factored as $V_k = H_k H_k^{\mathrm{T}}$, then $\hat{\mathbf{a}}_k$ is determined by the least squares solution of

$$\tilde{C}_k \mathbf{a}_k = \tilde{\mathbf{z}}_k, \tag{4.30}$$

where

$$\tilde{C}_k = L_{\mathbf{z}_k}^{-1} C_k, \quad \tilde{\mathbf{z}}_k = L_{\mathbf{z}_k}^{-1} \mathbf{z}_k,$$

and

$$L_{\mathbf{z}_k} = \begin{bmatrix} \tilde{L}_k & \\ & H_k \end{bmatrix}$$

is the Cholesky factor of $V_{\mathbf{z}_k}$. The solution of (4.30) can be found using standard linear least-squares approaches, in particular, using a QR factorisation approach (section 4.1).

A generalised QR implementation of the Kalman filter. The above calculations can also be organised in a way that avoids inverting any matrix. Using the notation of the previous paragraph, the estimate $\hat{\mathbf{a}}_k$ of $\mathbf{a}_k$ can be found by solving

$$\min_{\mathbf{a}_k, \mathbf{e}} \mathbf{e}^{\mathrm{T}}\mathbf{e} \quad \text{subject to} \quad \mathbf{z}_k = C_k \mathbf{a}_k + \begin{bmatrix} \bar{L}_k & \\ & H_k \end{bmatrix} \mathbf{e},$$

using a generalised QR factorisation approach (section 4.1.8).

The Kalman filter as a bidiagonal system. The problem addressed by the Kalman filter can be written as

$$C\mathbf{a} = \mathbf{y} + \boldsymbol{\gamma}, \quad \boldsymbol{\gamma} \in N(\mathbf{0}, U),$$

where

$$C = \begin{bmatrix} A_1 & & & & & & \\ -B_1 & I & & & & & \\ & A_2 & & & & & \\ & -B_2 & I & & & & \\ & & & \ddots & & & \\ & & & & -B_{K-1} & I & \\ & & & & & A_K & \end{bmatrix}, \quad \mathbf{a} = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_K \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{0} \\ \mathbf{y}_2 \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{y}_K \end{bmatrix},$$

and

$$U = \begin{bmatrix} V_1 & & & & & & \\ & W_2 & & & & & \\ & & V_2 & & & & \\ & & & W_2 & & & \\ & & & & \ddots & & \\ & & & & & W_K & \\ & & & & & & V_K \end{bmatrix}.$$

The main feature of C is that it has a block bi-diagonal structure of the form

$$C = \begin{bmatrix} C_{11} & C_{12} & & & \\ & C_{22} & C_{23} & & \\ & & \ddots & & \\ & & & C_{K-1,K-1} & C_{K-1,K} \end{bmatrix}, \tag{4.31}$$

where

$$C_{kk} = \begin{bmatrix} A_k \\ -B_k \end{bmatrix}, \quad C_{k,k+1} = \begin{bmatrix} \mathbf{0} \\ I \end{bmatrix}.$$

The least-squares estimate of $\mathbf{a}$ is given by the normal equations

$$C^{\mathrm{T}}U^{-1}C\mathbf{a} = C^{\mathrm{T}}U^{-1}\mathbf{y},$$

as in the case of more general Gauss-Markov problems (section 4.1.8). If U has Cholesky factorisation $U = LL^{\mathrm{T}}$, then equivalently $\mathbf{a}$ solves the linear least-squares problem

$$\tilde{C}\mathbf{a} = \tilde{\mathbf{y}}, \quad \text{where} \quad L\tilde{C} = C, \quad L\tilde{\mathbf{y}} = \mathbf{y}.$$

Importantly, $\tilde{C}$ is also a block bi-diagonal matrix.

Solution of block bi-diagonal least squares systems. Suppose C is an $m \times n$ block bi-diagonal matrix as in (4.31) and we wish to solve the linear least squares problem $C\mathbf{a} = \mathbf{y}$. If C has QR factorisation $C = QR$ where Q is an $m \times n$ orthogonal matrix and R is an $n \times n$ upper-triangular matrix, then the solution $\mathbf{a}$ solves

$$R\mathbf{a} = \mathbf{t}, \quad \mathbf{t} = \begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \\ \vdots \\ \mathbf{t}_K \end{bmatrix} = Q^{\mathrm{T}}\mathbf{y},$$

where the triangular factor also has a block bi-diagonal structure

$$R = \begin{bmatrix} R_{11} & R_{12} & & & \\ & R_{22} & R_{23} & & \\ & & & \ddots & \\ & & & R_{K-1,K-1} & R_{K-1,K} \\ & & & & R_{KK} \end{bmatrix}.$$

The solution $\mathbf{a}_k$ can be found by backwards substitution:

$$R_{KK}\mathbf{a}_K = \mathbf{t}_K, \quad R_{k-1,k-1}\mathbf{a}_{k-1} = \mathbf{t}_{k-1} - R_{k-1,k}\mathbf{a}_k, \quad k = K, K-1, \ldots, 2.$$

The uncertainty matrix $U_{\mathbf{a}}$ associated with the parameter estimates is given by

$$U_{\mathbf{a}} = \left(R^{\mathrm{T}}R\right)^{-1} = R^{-1}R^{-\mathrm{T}}.$$

The inverse $S = R^{-1}$ is necessarily upper-triangular with

$$\begin{bmatrix} R_{11} & R_{12} & & & \\ & R_{22} & R_{23} & & \\ & & \ddots & & \\ & & R_{K-1,K-1} & R_{K-1,K} & \\ & & & R_{KK} & \end{bmatrix} \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1K} \\ & S_{22} & \cdots & S_{2K} \\ & & \ddots & \vdots \\ & & & S_{KK} \end{bmatrix} = I,$$

and the blocks S_{kj}, $j \geq k$, can also be determined sequentially with

$$R_{KK}S_{KK} = I, \quad R_{kk}S_{kj} = -R_{k,k+1}S_{k+1,j}, \quad j > k.$$

Note that the uncertainty matrix $U_{\mathbf{a}_K}$ is simply $U_{\mathbf{a}_K} = S_{KK}S_{KK}^{\mathrm{T}} = (R_{KK}^{\mathrm{T}}R_{KK})^{-1}$.

In practice, we often need the uncertainty matrix in order to calculate the uncertainty matrix $U_{\mathbf{h}}$ associated with linear combinations $\mathbf{h} = H^{\mathrm{T}}\mathbf{a}$ of the parameters. We can do this without explicitly forming $S = R^{-1}$, since

$$U_{\mathbf{h}} = H^{\mathrm{T}}(R^{\mathrm{T}}R)^{-1}H = (H^{\mathrm{T}}R^{-1})(R^{-\mathrm{T}}H),$$

so that $U_{\mathbf{h}} = \tilde{H}^{\mathrm{T}}\tilde{H}$ where $\tilde{H}$ solves $R^{\mathrm{T}}\tilde{H} = H$. This system is a lower triangular block bi-diagonal system

$$\begin{bmatrix} R_{11}^{\mathrm{T}} & & & & \\ R_{12}^{\mathrm{T}} & R_{22}^{\mathrm{T}} & & & \\ & \ddots & & & \\ & R_{K-2,K-1}^{\mathrm{T}} & R_{K-1,K-1}^{\mathrm{T}} & & \\ & & R_{K-1,K}^{\mathrm{T}} & R_{KK}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \tilde{H}_1 \\ \tilde{H}_2 \\ \vdots \\ \tilde{H}_{K-1} \\ \tilde{H}_K \end{bmatrix} = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_{K-1} \\ H_K \end{bmatrix},$$

and can be solved using forward substitution:

$$R_{11}^{\mathrm{T}}\tilde{H}_1 = H_1, \quad R_{kk}^{\mathrm{T}}\tilde{H}_k = H_k - R_{k-1,k}^{\mathrm{T}}\tilde{H}_{k-1}, \quad k = 2, \ldots, K.$$

New information can be incorporated efficiently. If the current information is summarised by the matrix R and vector $\mathbf{t}$ and new information in the form

$$C_{KK}\mathbf{a}_K + C_{K,K+1}\mathbf{a}_{K+1} = \mathbf{y}_{k+1},$$

is available, updated estimates of the augmented set of parameters are given by the linear least squares solution of

$$\begin{bmatrix} R_{11} & R_{12} & & & & \\ & R_{22} & R_{23} & & & \\ & & \ddots & & & \\ & & R_{K-1,K-1} & R_{K-1,K} & & \\ & & & R_{KK} & & \\ & & & C_{K,K} & C_{K,K+1} \end{bmatrix} \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_K \\ \mathbf{a}_{K+1} \end{bmatrix} = \begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \\ \vdots \\ \mathbf{t}_K \\ \mathbf{y}_{K+1} \end{bmatrix}.$$

The upper triangularisation of this system only involves finding an orthogonal matrix Q_{K+1}, such that

$$Q_{K+1}^{\mathrm{T}} \begin{bmatrix} R_{KK} & \\ A_{K,K} & A_{K,K+1} \end{bmatrix} = \begin{bmatrix} R_{KK} & R_{K,K+1} \\ & R_{K+1,K+1} \end{bmatrix},$$

and setting

$$\begin{bmatrix} \mathbf{t}_K \\ \mathbf{t}_{K+1} \end{bmatrix} = Q_{K+1}^{\mathrm{T}} \begin{bmatrix} \mathbf{t}_K \\ \mathbf{y}_{K+1} \end{bmatrix}.$$

If required, the new information also provides updates of the parameters $\mathbf{a}_k$ already estimated at the Kth stage, through the solution of

$$\begin{bmatrix} R_{11} & R_{12} & & & & & \\ & R_{22} & R_{23} & & & & \\ & & & \ddots & & & \\ & & & R_{K-1,K-1} & R_{K-1,K} & & \\ & & & & R_{KK} & R_{K,K+1} & \\ & & & & & R_{K+1,K+1} \end{bmatrix} \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_K \\ \mathbf{a}_{K+1} \end{bmatrix} = \begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \\ \vdots \\ \mathbf{t}_K \\ \mathbf{t}_{K+1} \end{bmatrix}.$$

The new parameter vector $\mathbf{a}_{K+1}$ is determined from $R_{K+1,K+1}\mathbf{a}_{K+1} = \mathbf{t}_{K+1}$, and an updated estimate of $\mathbf{a}_K$ is given by solving $R_{KK}\mathbf{a}_K = \mathbf{t}_K - R_{K,K+1}\mathbf{a}_{K+1}$. The matrices R_{kk} and $R_{k,k+1}$, and vectors $\mathbf{t}_k$, $k = 1, \ldots, K-1$ are unchanged from the previous step so that the new estimates of $\mathbf{a}_k$, $k = 1, \ldots, K-1$, solve $R_k\mathbf{a}_k = \mathbf{t}_k - R_{k,k+1}\mathbf{a}_{k+1}$. Writing this update as $\mathbf{a}_k := \mathbf{a}_k + \delta\mathbf{a}_k$, then

$$R_k\delta\mathbf{a}_k = -R_{k,k+1}\delta\mathbf{a}_{k+1} \quad \text{or} \quad \delta\mathbf{a}_k = -R_k^{-1}R_{k,k+1}\delta\mathbf{a}_{k+1}.$$

Uncertainty matrices involving parameters vectors $\mathbf{a}_k$, $k \leq K$, can also be updated efficiently by solving a system of the form

$$\begin{bmatrix} R_{11}^{\mathrm{T}} & & & & \\ R_{12}^{\mathrm{T}} & R_{22}^{\mathrm{T}} & & & \\ & & \ddots & & \\ & & R_{K-1,K}^{\mathrm{T}} & R_{KK}^{\mathrm{T}} & \\ & & & R_{K,K+1}^{\mathrm{T}} & R_{K+1,K+1} \end{bmatrix} \begin{bmatrix} \tilde{H}_1 \\ \tilde{H}_2 \\ \vdots \\ \tilde{H}_{K-1} \\ \tilde{H}_K \\ \tilde{H}_{K+1} \end{bmatrix} = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_{K-1} \\ H_K \\ \mathbf{0} \end{bmatrix}.$$

The new information is limited to R_{KK} (which has been updated) $R_{K,K+1}$ and $R_{K+1,K+1}$, with all other submatrices of the triangular factor remaining unchanged. This means that $\tilde{H}_k$, $k = 1, \ldots, K-1$, are the same as in the previous calculation but now

$$R_{KK}^{\mathrm{T}}\tilde{H}_K = H_K - R_{K-1,K}^{\mathrm{T}}\tilde{H}_{K-1}, \quad R_{K+1,K+1}^{\mathrm{T}}\tilde{H}_{K+1} = -R_{K,K+1}^{\mathrm{T}}\tilde{H}_K.$$

Therefore,

$$U_{\mathbf{h}} = \sum_{k=1}^{K-1} \tilde{H}_k^{\mathrm{T}}\tilde{H}_k + \tilde{H}_K^{\mathrm{T}}\tilde{H}_K + \tilde{H}_{K+1}^{\mathrm{T}}\tilde{H}_{K+1}.$$

The update can be performed if we store the two matrices $\sum_{k=1}^{K-1} \tilde{H}_k^{\mathrm{T}}\tilde{H}_k$ and $\tilde{H}_{K-1}$.

4.1.12 Using linear least-squares solvers

Software for solving linear least-squares systems is generally straightforward to use. The user has to supply the observation matrix C and the right hand side vector $\mathbf{y}$ as inputs. The software will calculate the solution parameters $\mathbf{a}$ and the residual vector $\mathbf{r} = \mathbf{y} - C\mathbf{a}$. If the software uses an orthogonal factorisation approach (as can be recommended) then the triangular factor R_1 of the observation matrix is useful output as many uncertainty calculations can be made efficiently using R_1 and $\mathbf{r}$.

4.1.13 Linear least squares: summary

Least-squares methods are the most common estimators implemented and are appropriate for many practical model fitting problems. For linear models the following *Gauss-Markov Theorem* [152, chapter 6] can be used to justify their use:

> Gauss-Markov Theorem *For models of the form*
>
> $$\mathbf{y} = C\mathbf{a} + \boldsymbol{\epsilon},$$
>
> *where C is an $m \times n$ full rank matrix, $m \geq n$, and for which the random effects modelled by $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_m)^{\mathrm{T}}$ are observations of a vector of random variables $\boldsymbol{E}$ with variance $V(\boldsymbol{E}) = \sigma^2 I$, the linear least-squares estimator*
>
> $$\mathcal{A}(\mathbf{y}) = (C^{\mathrm{T}}C)^{-1}C^{\mathrm{T}}\mathbf{y}$$
>
> *is unbiased, i.e., $\mathcal{A}(\mathbf{y})$ is an observation of a vector of random variables $\boldsymbol{A}$ with expectation $E(\boldsymbol{A}) = \mathbf{a}$, and has a smaller variance matrix $V(\boldsymbol{A})$ than that for any other linear estimator.*

From this point of view, least-squares estimation is optimal for these models.

Note that there is no assumption that the random effects are normally or even symmetrically distributed, only that they are uncorrelated and have equal variance. This generality supports the use of least-squares methods.

Assumptions about normality are usually only invoked when it is required to provide coverage intervals associated with the fitted parameters. A consequence of the Gauss-Markov theorem is that if the uncertainty matrix associated with the data is $V_{\mathbf{y}}$ then the corresponding Gauss-Markov estimator (4.18) is optimal.

If we make the further assumption that random effects are normally distributed then the linear least squares estimators correspond to maximum likelihood estimators. The different types of estimators considered above arise from different uncertainty structures associated with the measurement data. From a Bayesian point of view, if there is no substantive prior information, the posterior distribution $p(\mathbf{a}|\mathbf{y})$ is given by the multivariate normal distribution $N(\hat{\mathbf{a}}, V_{\mathbf{a}})$ where $\hat{\mathbf{a}}$ is the least squares estimate and $V_{\mathbf{a}}$ is the associated uncertainty matrix. For linear models and normally distributed effects, the distributions $p(\hat{\mathbf{a}}|\mathbf{a})$ and $p(\mathbf{a}|\hat{\mathbf{a}})$ are described by the same multivariate normal distribution. (For nonlinear models this equivalence generally does not hold.)

4.1.14 Bibliography and software sources

Algorithms for solving linear least-squares systems are described in detail in [27, 117, 143, 209]. There are linear least-squares solvers in the NAG and IMSL libraries, LINPACK, MINPACK, LAPACK, DASL and Matlab, for example [8, 83, 112, 158, 175, 192, 206]. See also [125, 182]. There is a vast literature on the Kalman filter, starting with Kalman's original paper in 1960 [141]. See also, e.g., [36, 195, 200].

4.2 Nonlinear least squares

4.2.1 Description

The nonlinear least-squares problem is: given m functions $f_i(\mathbf{a})$ of parameters $\mathbf{a} = (a_1, \ldots, a_n)$, $m \geq n$, solve

$$\min_{\mathbf{a}} \; F(\mathbf{a}) = \frac{1}{2} \sum_{i=1}^{m} \; f_i^2(\mathbf{a}). \tag{4.32}$$

(The fraction $\frac{1}{2}$ is used so that related expressions are simpler.) Necessary conditions for $\mathbf{a}$ to be a solution are that

$$\frac{\partial F}{\partial a_j} = \sum_{i=1}^{m} f_i \frac{\partial f_i}{\partial a_j} = 0, \quad j = 1, \ldots, n.$$

Defining the *Jacobian matrix* $J = J(\mathbf{a})$ by

$$J_{ij} = \frac{\partial f_i}{\partial a_j}(\mathbf{a}), \tag{4.33}$$

this condition can be written as $J^{\mathrm{T}}(\mathbf{a})\mathbf{f}(\mathbf{a}) = \mathbf{0}$.

Nonlinear least-squares estimators are used widely in metrology in situations where the response variable is modelled as a nonlinear function $y = \phi(\mathbf{x}, \mathbf{a})$ of the model parameters $\mathbf{a}$ and covariates $\mathbf{x}$. They have good bias and efficiency properties for models in which the measurements of the response variable are subject to uncorrelated random effects:

$$\begin{aligned} &y_i = \phi(\mathbf{x}_i, \mathbf{a}) + \epsilon_i, \quad i = 1, \ldots, m \geq n, \\ &\boldsymbol{\epsilon} \in \boldsymbol{E}, \quad E(\boldsymbol{E}) = \mathbf{0}, \quad V(\boldsymbol{E}) = \sigma^2 I. \end{aligned}$$

If $\boldsymbol{E} \sim N(\mathbf{0}, \sigma^2 I)$, then the nonlinear least-squares estimate is the maximum likelihood estimate of $\mathbf{a}$. Nonlinear least-squares estimators are suitable for any system for which the random effects are associated with the measurements of the response variable and these random effects are independently distributed with zero mean and approximately equal standard deviations.

Nonlinear least squares are less suitable (without modification) for data in which more than one variable is subject to significant random effects (section 4.3), data which contains outliers (section 4.7) or where there is significant correlation associated with the random effects (section 4.2.7).

4.2.2 Algorithms for nonlinear least squares

Gauss-Newton algorithm for minimising a sum of squares. The Gauss-Newton algorithm is a modification of Newton's algorithm for minimising a function. Let

$$F(\mathbf{a}) = \frac{1}{2} \sum_{i=1}^{m} f_i^2(\mathbf{a})$$

and let $J(\mathbf{a})$ be the Jacobian matrix $J = \partial f_i/\partial a_j$.

Then (in the notation of section 3.7) $\mathbf{g} = J^{\mathrm{T}}\mathbf{f}$ and $H = J^{\mathrm{T}}J + G$, where

$$G_{jk} = \sum_{i=1}^{m} f_i \frac{\partial^2 f_i}{\partial a_j \partial a_k}. \tag{4.34}$$

The Gauss-Newton (GN) algorithm follows the same approach as the Newton algorithm (section 3.7), only that in determining the update step, H is approximated by $J^{\mathrm{T}}J$, i.e., the term G is ignored and $\mathbf{p}$ is found by solving $J^{\mathrm{T}}J\mathbf{p} = -J^{\mathrm{T}}\mathbf{f}$. This corresponds to the linear least-squares problem $J\mathbf{p} = -\mathbf{f}$ and can be solved using an orthogonal factorisation approach, for example; see section 4.1. The Gauss-Newton algorithm in general converges linearly at a rate that depends on the condition of the approximation problem, the size of the residuals $\mathbf{f}$ near the solution and the curvature. If the problem is well-conditioned, the residuals are small and the summand functions f_i are nearly linear, then $J^{\mathrm{T}}J$ is a good approximation to the Hessian matrix H and convergence is fast.

Geometrical interpretation. If the model is

$$y_i = \phi(\mathbf{x}_i, \mathbf{a}) + \epsilon_i, \quad \epsilon_i \in N(0, \sigma^2), \quad f_i(\mathbf{a}) = y_i - \phi(\mathbf{x}_i, \mathbf{a}),$$

the mapping $\mathbf{a} \mapsto \boldsymbol{\phi}(\mathbf{a})$ defines an n-dimensional surface in $\mathcal{R}^m$, and we look for $\hat{\mathbf{a}}$ that defines the point on the surface closest to $\mathbf{y}$. At the solution $\boldsymbol{\phi}(\hat{\mathbf{a}})$, the vector $\mathbf{f} = \mathbf{y} - \boldsymbol{\phi}(\hat{\mathbf{a}})$ is orthogonal to the surface at $\hat{\mathbf{a}}$. The tangent plane at $\hat{\mathbf{a}}$ is

$$\boldsymbol{\phi}(\hat{\mathbf{a}} + \boldsymbol{\Delta}) \approx \boldsymbol{\phi}(\hat{\mathbf{a}}) + J\boldsymbol{\Delta},$$

and so $\mathbf{f}$ must be orthogonal to the columns of J, or in matrix terms $J^{\mathrm{T}}\mathbf{f} = \mathbf{0}$, the optimality conditions.

The Gauss-Newton algorithm has the following geometrical interpretation. If the current estimate of the parameters is $\mathbf{a}_k$, the Jacobian matrix J evaluated at $\mathbf{a}_k$ is used to construct the linear n-space $\mathcal{J}$ defined by the columns of J. The step $\mathbf{p}$ defines the point $J\mathbf{p}$ on $\mathcal{J}$ closest to $\mathbf{y} - \boldsymbol{\phi}(\mathbf{a}_k)$. Figure 4.1 illustrates one step in the Gauss-Newton algorithm.

Gauss-Newton with line search. In practice, the update step is often of the form $\mathbf{a} = \mathbf{a} + t\mathbf{p}$ where the step length parameter t is chosen using a line search strategy to ensure there is a sufficient decrease in the value of the objective function $F(\mathbf{a})$ at each iteration.

Details. If

$$\mathbf{g} = \nabla_{\mathbf{a}} F = J^{\mathrm{T}}\mathbf{f}$$

is the gradient of F at $\mathbf{a}$ and $\phi(t) = F(\mathbf{a} + t\mathbf{p})$, then $\phi'(0) = \mathbf{g}^{\mathrm{T}}\mathbf{p}$ and

$$\rho(t) = \frac{\phi(t) - \phi(0)}{t\phi'(0)} \tag{4.35}$$

is the ratio of the actual decrease to that predicted from a first order approximation $\phi(t) \approx \phi(0) + t\phi'(0)$. For smooth functions $F(\mathbf{a})$, as t increases $\phi(t)$ decreases, reaches a minimum, where $\phi'(t) = 0$, and then starts to increase, reaching a point at which $\phi(t) = \phi(0)$, i.e., $F(\mathbf{a}+t\mathbf{p}) = F(\mathbf{a})$. For the function $\rho(t)$, at 0, ρ is 1 and then decreases to zero at t such that $\phi(t) = \phi(0)$. If ϕ is a quadratic function, then ϕ reaches a minimum at $t = 1$ and with $\rho(1) = 1/2$. Figure 4.2, shows the typical

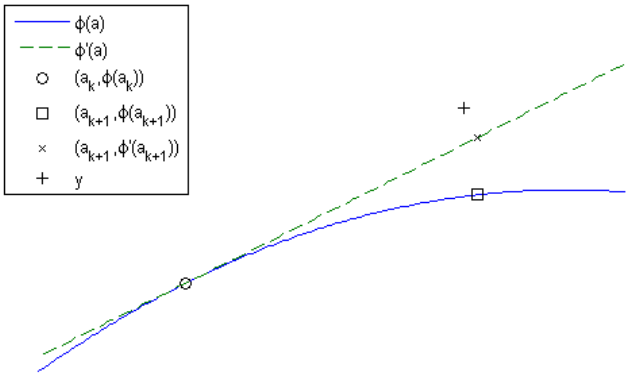

Figure 4.1: One step in the Gauss-Newton algorithm.

behaviour of $\phi(t)$ and $\rho(t)$. The graph also shows $\phi_Q(t)$, the quadratic approximation to $\phi(t)$, which takes a minimum at $t = 1$. A line search will generally look for a t such that

$$e_1 < \rho(t) < 1 - e_2, \quad 0 < e_1, e_2 < 1/2,$$

ensuring that the step is sufficiently large ($\rho(t)$ is bounded below 1) and represents a useful reduction in the function value ($\rho(t)$ is bounded above 0). Note that $\rho(t)$ only requires the evaluation of $F(\mathbf{a} + t\mathbf{p})$. Some line searches will also require that t represents a point reasonably close to a minimum of $\phi(t)$ by requiring that $|\phi'(t)|$ is a least a fixed fraction smaller than $|\phi'(0)|$:

$$|\phi'(t)| < (1 - 2e_3)|\phi'(0)|.$$

The calculation of $\phi'(t)$ involves the calculation of the gradient of $F(\mathbf{a} + t\mathbf{p})$. All three constraints can be specified by the same constant $0 < \eta = e_1 = e_2 = e_3 < 1/2$, if required.

Gauss-Newton with trust regions. The introduction of a line search is designed to improve the convergence characteristics of the Gauss-Newton algorithm. Another approach to help make the algorithm more robust is based on the concept of a trust region. In this approach, the step taken at each stage is restricted to a region in which a quadratic approximation centered at the current solution estimate to the function being minimised is judged to be valid. The size of the trust region is adjusted depending on the progress of the algorithm. See, for example, [89, 162]. A Levenberg-Marquardt trust region algorithm for nonlinear least squares is implemented in MINPACK [112].

Termination criteria. A second practical issue is concerned with convergence criteria usually involving i) the change in the objective function $\Delta F = F(\mathbf{a}) - F(\mathbf{a} + \mathbf{p})$, ii) the norm $\|\mathbf{p}\|$ of the step, and iii) the norm $\|\mathbf{g}\|$ of the gradient. Ideally, the criteria should be invariant with respect to changes of scale in the objective function and parameters.

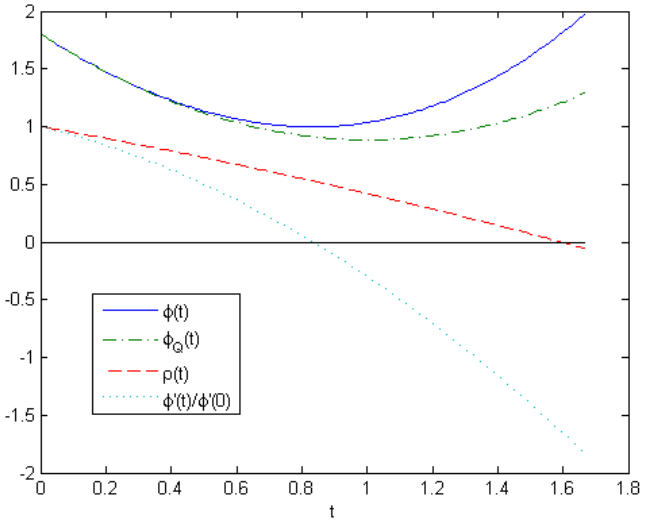

Figure 4.2: Typical behaviour of $\phi(t)$ and $\rho(t)$ used in a line search. The function $\phi_Q(t)$ is the quadratic approximation to $\phi(t)$.

A Gauss-Newton algorithm works well for problems where i) a good initial guess of the solution parameters is available, ii) the Jacobian matrix at the solution is reasonably well-conditioned, and iii) the functions f_i are not highly nonlinear. Well-designed least-squares optimisation algorithms will still work satisfactorily even if not all of these conditions apply.

Taking into account sparsity structure in the Jacobian matrix. Since the main step in the Gauss-Newton algorithm is the solution of a linear least-squares system, structured or sparse matrix techniques can be used in nonlinear least-squares problems [65].

4.2.3 Nonlinear least squares and maximum likelihood estimation

If the measurement model is $\mathbf{y} \in N(\boldsymbol{\phi}(\mathbf{a}), \sigma^2 I)$, with σ known, then the nonlinear least squares estimate is also the maximum likelihood estimate. From the definition of the multivariate normal distribution, the probability $p(\mathbf{y}|\mathbf{a})$ of observing $\mathbf{y}$, given parameter values $\mathbf{a}$, is such that

$$p(\mathbf{y}|\mathbf{a}) \propto \exp\left\{-\frac{1}{2\sigma^2}(\mathbf{y}-\boldsymbol{\phi}(\mathbf{a}))^{\mathrm{T}}(\mathbf{y}-\boldsymbol{\phi}(\mathbf{a}))\right\},$$

and is maximised by the nonlinear least squares solution.

4.2.4 Uncertainty associated with the fitted parameters

There are two approaches to estimating the uncertainty associated with a nonlinear least squares estimate. The first uses a linearised version of the law of the propagation of

uncertainty and relies only on mean and variance information. The second involves a Gaussian approximation to $p(\mathbf{a}|\mathbf{y})$, the distribution for $\mathbf{a}$, given that $\mathbf{y}$ has been observed. There are two variants to this latter approach, one based on the Hessian matrix, the second using an approximation to the Hessian matrix.

Application of the law of propagation of uncertainty. In the context of model fitting, suppose $f_i = y_i - \phi(\mathbf{x}_i, \mathbf{a})$ and that the uncertainty matrix associated with $\mathbf{y}$ is $V_{\mathbf{y}}$. Then the uncertainty matrix associated with the fitted parameters $\hat{\mathbf{a}}$, given $\mathbf{a}$, is approximated by

$$V_{\mathbf{a}} = (J^{\dagger})^{\mathrm{T}} V_{\mathbf{y}} J^{\dagger}, \quad J^{\dagger} = \left(J^{\mathrm{T}} J\right)^{-1} J^{\mathrm{T}}, \tag{4.36}$$

where J is the Jacobian matrix evaluated at the solution. If $V_{\mathbf{y}} = \sigma^2 I$, then

$$V_{\mathbf{a}} = \sigma^2 (J^{\mathrm{T}} J)^{-1}. \tag{4.37}$$

Details. Since $f_i = y_i - \phi(\mathbf{x}_i, \mathbf{a})$, we can regard $\mathbf{f} = \mathbf{f}(\mathbf{a}, \mathbf{y})$ as a function of both $\mathbf{y}$ and $\mathbf{a}$. The condition that the gradient $\mathbf{g}$ of F is zero at a minimum leads to the n equations $\mathbf{g}(\mathbf{a}, \mathbf{y}) = J^{\mathrm{T}}(\mathbf{a})\mathbf{f}(\mathbf{a}, \mathbf{y}) = \mathbf{0}$ which implicitly define $\mathbf{a} = \mathbf{a}(\mathbf{y})$ as a function of $\mathbf{y}$. In order to calculate the uncertainty matrix $V_{\mathbf{a}}$ we need to calculate the sensitivity matrix K with $K_{ji} = \partial a_j / \partial y_i$. Taking derivatives of the equation $\mathbf{g}(\mathbf{a}(\mathbf{y}), \mathbf{y}) = \mathbf{0}$ with respect to $\mathbf{y}$ yields

$$HK + J^{\mathrm{T}} = \mathbf{0}$$

so that $K = -H^{-1} J^{\mathrm{T}}$. Hence $V_{\mathbf{a}} = K V_{\mathbf{y}} K^{\mathrm{T}}$. However, this expression applies to the uncertainty in $\hat{\mathbf{a}}$ due to perturbations in the data centered around the observed data vector $\mathbf{y}$, rather than perturbations around $\boldsymbol{\phi}(\mathbf{a})$. If we evaluate J and H at $\boldsymbol{\phi}(\mathbf{a})$ then $H = J^{\mathrm{T}} J$, leading to (4.37).

If J has QR factorisation $J = Q_1 R_1$ at the solution where R_1 is an $n \times n$ upper-triangular matrix then $V_{\mathbf{a}} \approx \sigma^2 (R_1^{\mathrm{T}} R_1)^{-1}$. A posterior estimate $\hat{\sigma}$ of σ can be determined from the vector $\mathbf{f}$ of residuals at the solution according to

$$\hat{\sigma} = \frac{\|\mathbf{f}\|}{(m-n)^{1/2}},$$

but see section 4.2.5. Both (4.36) and (4.37) are based on linearisations and therefore can only provide an estimate of the variance matrix associated with the fitted parameters. For highly nonlinear models (with relatively large curvature) these estimates may be significantly different from the true variance matrix. Forward Monte Carlo simulation techniques, for example, can be used either to validate these estimates or provide alternative estimates that do not involve any linearising approximations.

However, both (4.36) and forward Monte Carlo simulations estimate, for a fixed $\mathbf{a}$, the likely variation in parameter estimates $\hat{\mathbf{a}}$ due to the likely variation of the data $\mathbf{y}$ due to the random effects associated with the measurement system, i.e., they estimate the variance of the distribution $p(\hat{\mathbf{a}}|\mathbf{a})$. In practice, $\mathbf{a}$ is unknown and only $\mathbf{y}$, and subsequently $\hat{\mathbf{a}}$, are observed and the uncertainty matrices are based on calculations with $\mathbf{a}$ set equal to the observed value. For nonlinear models, the shape of the distribution $p(\hat{\mathbf{a}}|\mathbf{a})$ depends on $\mathbf{a}$ which means that the validity of estimates of the variance matrices depends on how the distribution shape changes with respect to $\mathbf{a}$. (For linear models, $p(\hat{\mathbf{a}}|\mathbf{a})$ is independent of $\mathbf{a}$.)

Variance estimate based on a Gaussian approximation to the posterior distribution. In a Bayesian context, the posterior distribution $p(\mathbf{a}|\mathbf{y})$ is such that

$$p(\mathbf{a}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{a})p(\mathbf{a})$$

where $p(\mathbf{a})$ is the prior information and $p(\mathbf{y}|\mathbf{a})$ is the likelihood of observing $\mathbf{y}$, given $\mathbf{a}$. If there is no substantive prior information, $p(\mathbf{a}) = 1$, and if the model is $\mathbf{y} \in N(\boldsymbol{\phi}(\mathbf{a}), \sigma^2 I)$ with σ known, then

$$p(\mathbf{a}|\mathbf{y}) \propto \exp\left\{-\frac{1}{2\sigma^2}(\mathbf{y} - \boldsymbol{\phi}(\mathbf{a}))^{\mathrm{T}}(\mathbf{y} - \boldsymbol{\phi}(\mathbf{a}))\right\}. \tag{4.38}$$

The term on the right represents the kernel of a multivariate normal distribution with respect to $\mathbf{y}$. If $\boldsymbol{\phi}(\mathbf{a})$ is a linear function of $\mathbf{a}$ then it also represents the kernel of a multivariate normal distribution with the respect to $\mathbf{a}$. For nonlinear $\boldsymbol{\phi}(\mathbf{a})$, the distribution can be quite different from a multinormal. (If m is much greater than n, then asymptotic results show that it is likely to be close to a multinormal distribution.) As discussed in section 3.5.2, a Gaussian approximation to $p(\mathbf{a}|\mathbf{y})$ can be determined using a quadratic approximation to $-\log p(\mathbf{a}|\mathbf{y})$ about the mode (the point of maximum density) of the distribution. Given the nonlinear least squares solution $\hat{\mathbf{a}}$, we approximate $\mathbf{f}(\mathbf{a}) = \mathbf{y} - \boldsymbol{\phi}(\mathbf{a})$ near $\mathbf{y} - \hat{\mathbf{y}}$, $\hat{\mathbf{y}} = \boldsymbol{\phi}(\hat{\mathbf{a}})$ by a quadratic function

$$\mathbf{f}(\mathbf{a}) \approx (\mathbf{y} - \hat{\mathbf{y}}) + J(\mathbf{a} - \hat{\mathbf{a}}) + \frac{1}{2}(\mathbf{a} - \hat{\mathbf{a}})^{\mathrm{T}}\boldsymbol{G}(\mathbf{a} - \hat{\mathbf{a}}), \tag{4.39}$$

where J is the Jacobian matrix evaluated at $\hat{\mathbf{a}}$, and $\boldsymbol{G}$ is the $n \times m \times n$ array with

$$\boldsymbol{G}(j, i, k) = \frac{\partial^2 f_i}{\partial a_j \partial a_k},$$

so that $\boldsymbol{G}$ stores the $n \times n$ matrix of second partial derivatives for each of the m functions f_i, evaluated at $\mathbf{a} = \hat{\mathbf{a}}$. Then, up to quadratic terms,

$$(\mathbf{y} - \boldsymbol{\phi}(\mathbf{a}))^{\mathrm{T}}(\mathbf{y} - \boldsymbol{\phi}(\mathbf{a})) = \mathbf{f}^{\mathrm{T}}\mathbf{f},$$

is approximated by

$$ms^2 + (\mathbf{a} - \hat{\mathbf{a}})^{\mathrm{T}}\left[J^{\mathrm{T}}J + G\right](\mathbf{a} - \hat{\mathbf{a}}), \quad ms^2 = (\mathbf{y} - \hat{\mathbf{y}})^{\mathrm{T}}(\mathbf{y} - \hat{\mathbf{y}}) \tag{4.40}$$

with G defined as in (4.34). The cross term $\mathbf{f}^{\mathrm{T}}J(\mathbf{a} - \hat{\mathbf{a}})$ is zero since $\hat{\mathbf{a}}$ is the nonlinear least squares solution so that $J^{\mathrm{T}}\mathbf{f} = \mathbf{0}$. The term ms^2 on the right in (4.40) does not depend on $\mathbf{a}$, so that comparing (4.38) with the above,

$$p(\mathbf{a}|\mathbf{y}) \approx K \exp\left\{-\frac{1}{2\sigma^2}(\mathbf{a} - \hat{\mathbf{a}})^{\mathrm{T}}H(\mathbf{a} - \hat{\mathbf{a}})\right\},$$

where $H = J^{\mathrm{T}}J + G$ is the Hessian matrix of associated with

$$F(\mathbf{a}) = \frac{1}{2}\mathbf{f}^{\mathrm{T}}\mathbf{f} = \frac{1}{2}(\mathbf{y} - \boldsymbol{\phi}(\mathbf{a}))^{\mathrm{T}}(\mathbf{y} - \boldsymbol{\phi}(\mathbf{a})),$$

i.e., $p(\mathbf{a}|\mathbf{y})$ is approximated by the normal distribution $N(\hat{\mathbf{a}}, V_{\mathbf{a}}^Q)$, where

$$V_{\mathbf{a}}^Q = \sigma^2\left[J^{\mathrm{T}}J + G\right]^{-1}. \tag{4.41}$$

If H is approximated by $J^{\mathrm{T}}J$, then $V_{\mathbf{a}} = \sigma^2(J^{\mathrm{T}}J)^{-1}$, as in (4.37).

The approximation based on H is derived from a quadratic approximation of the surface $\boldsymbol{\phi}(\mathbf{a})$ at $\hat{\mathbf{a}}$ which involves the matrix G of second partial derivatives. The approximation based on $J^{\mathrm{T}}J$ is derived from a linear approximation of the surface $\boldsymbol{\phi}(\mathbf{a})$. Furthermore, the distribution $N(\hat{\mathbf{a}}, V_{\mathbf{a}})$ depends only on the estimate $\hat{\mathbf{a}}$ whereas $N(\hat{\mathbf{a}}, V_{\mathbf{a}}^Q)$ depends on $\mathbf{y}$ through its contribution to G.

Based on linearisations, the distribution for $\mathbf{a}|\mathbf{y}$ is estimated by $N(\hat{\mathbf{a}}, V_{\mathbf{a}})$ and that for $\hat{\mathbf{a}}|\mathbf{a}$ by $N(\mathbf{a}, V_{\mathbf{a}})$. It follows from these linearisations, that $p(\mathbf{a}|\hat{\mathbf{a}})$ is also estimated by $N(\hat{\mathbf{a}}, V_{\mathbf{a}})$ and that $p(\hat{\mathbf{a}}|\mathbf{a})$ and $p(\mathbf{a}|\mathbf{y})$, to a linear approximation, are represented by the same distribution. The more nonlinear the model, the less good are these linear approximations and the more disparate these two distributions can become. The distribution $p(\mathbf{a}|\hat{\mathbf{a}}, V_{\mathbf{a}}^Q)$ represents the information about $\mathbf{a}$ derived from observing the nonlinear least squares estimate $\hat{\mathbf{a}}$ and the uncertainty matrix $V_{\mathbf{a}}^Q$. In general, $V_{\mathbf{a}}^Q$ will provide information additional to that which can be derived from the parameter estimate $\hat{\mathbf{a}}$ alone.

4.2.5 Partial information about σ

The uncertainty matrices

$$V_{\mathbf{a}} = \sigma^2(J^{\mathrm{T}}J)^{-1}, \quad \widehat{V}_{\mathbf{a}} = \hat{\sigma}^2(J^{\mathrm{T}}J)^{-1}, \tag{4.42}$$

correspond to $\mathbf{y} \in N(\boldsymbol{\phi}(\mathbf{a}), \sigma^2 I)$ in the cases where σ is known exactly and nothing is known about σ. As for the linear case, section 4.1.5, we can consider the situation in which partial information about σ is encoded by the prior distribution for $\eta = 1/\sigma^2$ of the form

$$m_0\sigma_0^2\eta \sim \chi^2_{m_0},$$

where σ_0 represents a prior estimate and $m_0 \geq 0$ measures our degree of belief in σ_0; the larger m_0, the more belief we have. Assuming the prior density for $\mathbf{a}$ is $p(\mathbf{a}) = 1$, the posterior density $p(\mathbf{a}, \eta|\mathbf{y})$ is such that

$$\begin{aligned} p(\mathbf{a}, \eta|\mathbf{y}) &\propto \eta^{\frac{m_0}{2}-1} \exp\left\{-\frac{\eta}{2}m_0\sigma_0^2\right\} \eta^{m/2} \exp\left\{-\frac{\eta}{2}(\mathbf{y}-\boldsymbol{\phi}(\mathbf{a}))^{\mathrm{T}}(\mathbf{y}-\boldsymbol{\phi}(\mathbf{a}))\right\}, \\ &= \eta^{\frac{m+m_0}{2}-1} \exp\left\{-\frac{\eta}{2}\left[m_0\sigma_0^2 + (\mathbf{y}-\boldsymbol{\phi}(\mathbf{a}))^{\mathrm{T}}(\mathbf{y}-\boldsymbol{\phi}(\mathbf{a}))\right]\right\}. \end{aligned}$$

Using the integration rule (4.15), the marginal density for $\mathbf{a}$ is

$$p(\mathbf{a}|\mathbf{y}) = \int_0^\infty p(\mathbf{a}, \eta|\mathbf{y})d\eta \propto \left[m_0\sigma_0^2 + (\mathbf{y}-\boldsymbol{\phi}(\mathbf{a}))^{\mathrm{T}}(\mathbf{y}-\boldsymbol{\phi}(\mathbf{a}))\right]^{-\frac{m_0+m}{2}}.$$

Thus, $p(\mathbf{a}|\mathbf{y})$ is maximised by nonlinear least squares solution $\hat{\mathbf{a}}$. Using the same approximations as in (4.39) and (4.40),

$$p(\mathbf{a}|\mathbf{y}) \approx K\left[1 + \frac{(\mathbf{a}-\hat{\mathbf{a}})^{\mathrm{T}}\left[J^{\mathrm{T}}J + G\right](\mathbf{a}-\hat{\mathbf{a}})}{m_0\sigma_0^2 + ms^2}\right]^{-\frac{m_0+m}{2}},$$

where K is a normalising constant. Comparing the righthand side with the multivariate t distribution, we see that $p(\mathbf{a}|\mathbf{y})$ is approximated by the $t_\nu(\hat{\mathbf{a}}, \bar{V})$ where $\nu = m + m_0 - n$ and

$$\bar{V} = \bar{\sigma}^2[J^{\mathrm{T}}J + G]^{-1}, \quad \bar{\sigma}^2 = \frac{m_0\sigma_0^2 + ms^2}{m_0 + m - n}$$

Using the approximation $G \approx \mathbf{0}$, this matrix can be compared with the usual estimates for the uncertainty matrix associated with nonlinear least squares parameter estimation (4.42). If $m_0 \gg m$, indicating that there is strong belief in the prior estimate of η, then

$$\bar{V} \approx \sigma_0^2 (J^{\mathrm{T}} J)^{-1},$$

and the righthand side is the estimate of the uncertainty matrix associated with $\mathbf{a}$ based on the input estimate σ_0 of the standard deviation while, for $m_0 \approx 0$, corresponding to no prior knowledge about σ,

$$\bar{V} \approx \hat{\sigma}^2 (J^{\mathrm{T}} J)^{-1}, \quad \hat{\sigma}^2 = \frac{\mathbf{f}^{\mathrm{T}} \mathbf{f}}{m-n},$$

where $\hat{\sigma}$ is the posterior estimate of σ based on the sum of squares of residuals at the solution. For the case $m + m_0 > n - 2$, it is appropriate to associate with the nonlinear least squares estimate $\hat{\mathbf{a}}$, the uncertainty matrix

$$\check{V}_{\mathbf{a}} = \check{\sigma}^2 (J^{\mathrm{T}} J)^{-1}, \quad \check{\sigma}^2 = \frac{m_0 \sigma_0^2 + \mathbf{f}^{\mathrm{T}} \mathbf{f} + m_0}{m_0 + m - n - 2}, \qquad \mathbf{f} = \mathbf{y} - \boldsymbol{\phi}(\hat{\mathbf{a}}).$$

4.2.6 Weighted nonlinear least-squares estimator

If the functions f_i relate to random effects ϵ_i with differing variances σ_i^2, then the appropriate estimator is a weighted nonlinear least-squares estimator which estimates $\mathbf{a}$ by solving

$$\min_{\mathbf{a}} \sum_{i=1}^{m} w_i^2 f_i^2(\mathbf{a}),$$

with $w_i = 1/\sigma_i$. Algorithms for the unweighted nonlinear least squares can be easily adapted to deal with the weighted case by applying them to $\tilde{f}_i = w_i f_i$.

4.2.7 Nonlinear Gauss-Markov estimator

If the covariance matrix associated with $\boldsymbol{\epsilon}$ is V, assumed to be full rank, the appropriate estimate of the model parameters is the one that solves

$$\min_{\mathbf{a}} \ \mathbf{f}^{\mathrm{T}}(\mathbf{a}) V^{-1} \mathbf{f}(\mathbf{a}). \tag{4.43}$$

As in the linear case, we can use the Cholesky decomposition $V = LL^{\mathrm{T}}$ to convert this problem to a standard nonlinear least-squares problem applied to

$$\tilde{\mathbf{f}} = L^{-1} \mathbf{f}.$$

As for the case of linear least squares, if V and hence L is poorly conditioned the formation and use of L^{-1} could lead to numerical instability. The Gauss-Newton algorithm can be adapted so that at each iteration the Gauss-Newton step is found by solving

$$\min_{\mathbf{a},\mathbf{e}} \ \mathbf{e}^{\mathrm{T}} \mathbf{e} \quad \text{subject to constraints} \quad \mathbf{y} = -J\mathbf{p} + L\mathbf{e},$$

using, for example, the generalised QR decomposition (section 4.1.8). More generally, if V is given in factored form as $V = BB^{\mathrm{T}}$, then B can replace L in the above. There is no requirement for B to be a square matrix.

4.2.8 Structured nonlinear Gauss-Markov problems

As discussed in section 4.1.9, the uncertainty matrix V can often be specified more compactly in factored form. The same approach for the linear case, described in section 4.1.9, also applies in the nonlinear case. Suppose

$$V = \sigma^2 I + H U_0 H^{\mathrm{T}},$$

$D = \sigma I$, U_0 has Cholesky factorisation $U_0 = L_0 L_0^{\mathrm{T}}$, and $B_0 = HL_0$, then V can be factored as

$$V = BB^{\mathrm{T}}, \quad B = \begin{bmatrix} D & B_0 \end{bmatrix},$$

and (4.43) has the same solution as

$$\min_{\mathbf{a}} \mathbf{e}^{\mathrm{T}}\mathbf{e} + \mathbf{e}_0^{\mathrm{T}}\mathbf{e}_0 \quad \text{subject to} \quad \mathbf{f}(\mathbf{a}) = D\mathbf{e} + B_0\mathbf{e}_0. \tag{4.44}$$

Setting

$$\tilde{\mathbf{a}} = \begin{bmatrix} \mathbf{a} \\ \mathbf{e}_0 \end{bmatrix}, \quad \tilde{\mathbf{f}}(\tilde{\mathbf{a}}) = \begin{bmatrix} D^{-1}(\mathbf{f}(\mathbf{a}) - B_0\mathbf{e}_0) \\ \mathbf{e}_0 \end{bmatrix},$$

the nonlinear Gauss-Markov problem (4.43) is equivalent to

$$\min_{\tilde{\mathbf{a}}} \tilde{\mathbf{f}}^{\mathrm{T}}\tilde{\mathbf{f}},$$

a standard nonlinear least-squares problem involving an augmented set of parameters $\tilde{\mathbf{a}}$.

4.2.9 Nonlinear least squares subject to linear constraints

Algorithms for nonlinear least squares can also be adapted to problems with p linear constraints $D\mathbf{a} = \mathbf{d}$ on the parameters, $p < n$. As described in section 4.1.10, the optimisation problem can be reformulated as an unconstrained problem of the form

$$\min_{\tilde{\mathbf{a}}} \sum_{i=1}^{m} \tilde{f}_i^2(\tilde{\mathbf{a}}) \tag{4.45}$$

where $\tilde{f}_i(\tilde{\mathbf{a}}) = f_i(\mathbf{a}_0 + U_2\tilde{\mathbf{a}})$. Here $\mathbf{a}_0$ is any set of parameters satisfying the constraints, i.e., $D\mathbf{a}_0 = \mathbf{d}$, $\tilde{\mathbf{a}}$ represents the reduced set of $(n-p)$ parameters and U_2 is an $n \times (n-p)$ orthogonal matrix (derived from the QR factorisation of D^{T} in (4.25)) such that $DU_2 = \mathbf{0}$. Note that if J is the Jacobian matrix of partial derivatives $J_{ij} = \frac{\partial f_i}{\partial a_j}$, then the Jacobian matrix associated with (4.45) is given by $\tilde{J} = JU_2$. As described in section 4.1.10, the vector $\mathbf{a}_0 = U_1 S_1^{-\mathrm{T}}\mathbf{d}$ satisfies the constraints. In some situations, given an estimate of the parameters $\mathbf{a}$ it is necessary to find the nearest estimate of the parameters $\tilde{\mathbf{a}}$ that satisfy the constraints, i.e., we wish to solve

$$\min_{\mathbf{a}_2} \|\mathbf{a} - (U_1 S_1^{-\mathrm{T}}\mathbf{d} - U_2\mathbf{a}_2)\|.$$

Since U_2 is orthogonal, the solution is given by $\mathbf{a}_2 = U_2^{\mathrm{T}}(\mathbf{a} - U_1 S_1^{-\mathrm{T}}\mathbf{b}) = U_2^{\mathrm{T}}\mathbf{a}$, showing that

$$\mathbf{a}_0 = U_1 S_1^{-\mathrm{T}}\mathbf{d} + U_2 U_2^{\mathrm{T}}\mathbf{a}$$

is the closest vector to $\mathbf{a}$ that satisfies the constraints.

4.2.10 Using nonlinear least-squares solvers

Software for solving nonlinear least-squares systems is in principle straightforward to use. The user has to supply a software module to calculate the vector of function values $\mathbf{f}$ and the Jacobian matrix J of partial derivatives for a given value of the optimisation parameters $\mathbf{a}$. For complicated models, the correct calculation of these derivatives can involve a lot of effort both in deriving the correct formulæ and in their subsequent implementation in software. For this reason, many optimisation packages offer versions of the algorithms for which only function values are required and use finite difference approximations of the form

$$\frac{\partial f}{\partial a_j}(\mathbf{a}) \approx \frac{f(a_1, \ldots, a_j + \Delta_j, a_{j+1}, \ldots, a_n) - f(a_1, \ldots, a_n)}{\Delta_j}$$

to estimate the derivatives. This is done at the cost of accuracy of the solution and usually efficiency of the underlying algorithm. There is much current research on finding better ways of estimating derivatives. Automatic differentiation techniques, including forward and reverse accumulation, and the complex step method and their use in metrology are described in [30]. The complex step method is particularly easy to implement in languages such as Matlab or Fortran 90/95 that support complex arithmetic.

The user has also to supply an initial estimate of the optimisation parameters. For most metrology applications, this is not a usually a problem but there are situations where this is a major difficulty.

The optimisation software will calculate the solution parameters $\mathbf{a}$ and the vector of function values $\mathbf{f}$ at the solution. If the software uses an orthogonal factorisation approach in the iterative step then the triangular factor R_1 of the Jacobian matrix at the solution is useful output as all necessary statistics can be determined efficiently using R_1 and $\mathbf{f}$.

4.2.11 Bibliography and software sources

There are a number of nonlinear least-squares solvers in MINPACK and the NAG and IMSL libraries [112, 175, 206]. Nonlinear least-squares algorithms are described in [89, 115], for example. See also [163]. For more on automatic differentiation, see for example, [20, 30, 119, 198].

4.3 Generalised distance regression (GDR)

4.3.1 Description

Linear and nonlinear least-squares estimators are appropriate if only one measured variable is subject to significant random effects. However, in many metrological situations, there is significant uncertainty associated with more than one of the measured variables and it is important to take this into account in determining parameter estimates that are free from significant bias.

In a generalised distance regression (GDR) formulation, it is assumed that each set of measurements $\mathbf{x}_i$ is subject to random effects so that $\mathbf{x}_i = \mathbf{x}_i^* + \boldsymbol{\epsilon}_i$, where $\mathbf{x}_i^*$ satisfies the

model constraints $f(\mathbf{x}_i^*, \mathbf{a}) = 0$ for some unknown $\mathbf{a}$. The set of measurements $\mathbf{x}$ subsumes both the stimulus variables and the response variable (y). In this formulation, y is treated on the same footing as the other components of $\mathbf{x}$.

It is assumed that the effects modelled by $\boldsymbol{\epsilon}_i$ associated with the components of $\mathbf{x}_i$ can be correlated with each other, but that the ith and jth sets are uncorrelated, $i \neq j$. (More general uncertainty structures are considered in [95], for example.) If V_i is the uncertainty (covariance) matrix associated with $\boldsymbol{\epsilon}_i$[4] (assumed to be full rank), then maximum likelihood estimates of the model parameters $\mathbf{a}$ can be found by solving

$$\min_{\mathbf{a},\{\mathbf{x}_i^*\}} \sum_{i=1}^{m} (\mathbf{x}_i - \mathbf{x}_i^*)^{\mathrm{T}} V_i^{-1} (\mathbf{x}_i - \mathbf{x}_i^*) \tag{4.46}$$

subject to the model constraints $f(\mathbf{x}_i^*, \mathbf{a}) = 0$. This is an implicit formulation of the problem. If the surface $f(\mathbf{x}, \mathbf{a}) = 0$ can be represented explicitly (i.e., parametrically) as $\mathbf{x} = \boldsymbol{\phi}(\mathbf{u}, \mathbf{a})$, where $\boldsymbol{\phi} : \mathcal{R}^{p-1} \times \mathcal{R}^n \longrightarrow \mathcal{R}^p$, then (4.46) can be reformulated as

$$\min_{\mathbf{a},\{\mathbf{u}_i^*\}} \sum_{i=1}^{m} (\mathbf{x}_i - \boldsymbol{\phi}(\mathbf{u}_i^*, \mathbf{a}))^{\mathrm{T}} V_i^{-1} (\mathbf{x}_i - \boldsymbol{\phi}(\mathbf{u}_i^*, \mathbf{a})), \tag{4.47}$$

an unconstrained optimisation problem. If each $V_i = I$, the identity matrix, the GDR problem is known as *orthogonal regression*. Orthogonal regression for linear models is sometimes termed *total least squares*.

Generalised distance regression methods have not been used extensively until recent years. A typical situation for which they are appropriate is where the response $y = \phi(x, \mathbf{a})$ is modelled as a function of the variable x and parameters $\mathbf{a}$, and both y and x are measured subject to random effects, giving rise to observation equations of the form

$$x_i = u_i^* + \delta_i, y_i = \phi(x_i^*) + \epsilon_i, \quad \delta_i \in N(0, \sigma_x^2), \epsilon_i \in N(0, \sigma_y^2).$$

The maximum likelihood estimate of the parameters is found by solving

$$\min_{\mathbf{a},\{u_i^*\}} \sum_{i=1}^{m} \left\{ \left(\frac{x_i - u_i^*}{\sigma_x} \right)^2 + \left(\frac{y_i - \phi(u_i^*, \mathbf{a})}{\sigma_y} \right)^2 \right\}.$$

Orthogonal regression is used extensively in co-ordinate metrology.

4.3.2 Algorithms for generalised distance regression

Separation of variables approaches. At first sight, both generalised regression formulations (4.46) and (4.47) represent significantly more challenging optimisation problems than standard nonlinear least-squares problems as they have to take into account the additional parameters, etc. However, using a separation-of-variables approach, it is possible to convert them to standard nonlinear least-squares problems in the parameters $\mathbf{a}$. We consider the explicit case (4.47) first.

We assume that V is a symmetric, strictly positive definite matrix. Denote by $\mathbf{u}_i^* = \mathbf{u}_i^*(\mathbf{a})$ the solution of the footpoint problem

$$\min_{\mathbf{u}} D(\mathbf{u}) = (\mathbf{x}_i - \boldsymbol{\phi}(\mathbf{u}, \mathbf{a}))^{\mathrm{T}} V^{-1} (\mathbf{x}_i - \boldsymbol{\phi}(\mathbf{u}, \mathbf{a})). \tag{4.48}$$

[4] That is, $\boldsymbol{\epsilon}_i \in \boldsymbol{E}_i$ and $V(\boldsymbol{E}_i) = V_i$.

Let $\mathbf{n}_i$ be any vector orthogonal to the surface at $\mathbf{x}_i^* = \boldsymbol{\phi}(\mathbf{u}_i^*, \mathbf{a})$. The conditions for $\mathbf{u}_i^*$ to be a solution of (4.48) is that the vector $V^{-1}(\mathbf{x}_i - \mathbf{x}_i^*)$ is a scalar multiple of $\mathbf{n}_i$. From this, it is straightforward to show that if we define the *generalised distance* $d_i = d_i(\mathbf{a})$ by

$$d_i = \frac{1}{s_i}\mathbf{n}_i^{\mathrm{T}}(\mathbf{x}_i - \mathbf{x}_i^*), \quad s_i = (\mathbf{n}_i^{\mathrm{T}} V \mathbf{n}_i)^{1/2}, \tag{4.49}$$

then $d_i^2 = D(\mathbf{u}_i^*)$, and

$$\frac{\partial d_i}{\partial a_j} = -\frac{1}{s_i}\mathbf{n}_i^{\mathrm{T}}\frac{\partial \boldsymbol{\phi}}{\partial a_j}. \tag{4.50}$$

In this way, the explicit generalised distance regression problem can be posed as a standard nonlinear least-squares problem $\min_{\mathbf{a}} \sum_i d_i^2(\mathbf{a})$ where each function and its gradient are calculated as in (4.49) and (4.50) with all quantities evaluated at the solution $\mathbf{u}_i^*$ of the appropriate footpoint problem. Note that both d_i and its derivatives are defined in terms of V, through s_i, rather than V^{-1}. If V can be factored as $V = BB^{\mathrm{T}}$, then the footpoint problem can be posed as

$$\min_{\mathbf{u}_i, \mathbf{e}} \ \mathbf{e}^{\mathrm{T}}\mathbf{e} \quad \text{subject to} \quad \mathbf{x}_i = \boldsymbol{\phi}(\mathbf{u}_i, \mathbf{a}) + B\mathbf{e}, \tag{4.51}$$

again avoiding the formation of V^{-1}. There is no requirement in implementing the separation of variables approach that V is full rank, only that $\mathbf{n}^{\mathrm{T}} V \mathbf{n}$ is nonzero, where $\mathbf{n}$ is normal to the surface.

Example: simple GDR for parametric curves

The simple GDR problem for parametric curves can be stated as: given data points $\{(x_i, y_i)\}_1^m$ and strictly positive weights $\{(\alpha_i, \beta_i)\}_1^m$, minimise

$$\sum_{i=1}^{m} \ \left\{\alpha_i^2(x_i - \phi(u_i, \mathbf{a}))^2 + \beta_i^2(y_i - \psi(u_i, \mathbf{a}))^2\right\}$$

with respect to $\mathbf{a}$ and $\{u_i\}_1^m$ where $(\phi, \psi) = (\phi(u, \mathbf{a}), \psi(u, \mathbf{a}))$ is a parametric curve in $\mathcal{R}^2$. The theory above shows that this can be reformulated as:

$$\min_{\mathbf{a}} \sum_{i=1}^{m} d_i^2(\mathbf{a})$$

with

$$\begin{aligned} d_i &= \frac{1}{s_i}\left(-(x_i - \phi_i^*)\dot{\psi}_i + (y_i - \psi_i^*)\dot{\phi}_i\right), \\ \frac{\partial d_i}{\partial a_j} &= \frac{1}{s_i}\left(\frac{\partial \phi_i}{\partial a_j}\dot{\psi}_i - \frac{\partial \psi_i}{\partial a_j}\dot{\phi}_i\right), \end{aligned}$$

where

$$\begin{aligned} \dot{\phi}_i &= \frac{\partial \phi_i}{\partial u}, \quad \text{etc.}, \\ s_i &= \left(\frac{\dot{\psi}_i^2}{\alpha_i^2} + \frac{\dot{\phi}_i^2}{\beta_i^2}\right)^{1/2}, \end{aligned}$$

with all expressions evaluated at the solution u_i^* of the corresponding footpoint problem:

$$\min_u \ \left\{\alpha_i^2(x_i - \phi(u,\mathbf{a}))^2 + \beta_i^2(y_i - \psi(u,\mathbf{a}))^2\right\}.$$

If $\alpha_i = 1/\sigma_{x,i}$ and $\beta_i = 1/\sigma_{y,i}$, then

$$s_i = \left(\sigma_{x,i}^2\dot{\psi}_i^2 + \sigma_{y,i}^2\dot{\phi}_i^2\right)^{1/2},$$

and the footpoint problem can be posed as

$$\min_u e_x^2 + e_y^2 \quad \text{subject to} \quad \begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} \phi(u,\mathbf{a}) \\ \psi(u,\mathbf{a}) \end{bmatrix} + \begin{bmatrix} \sigma_{x,i}e_x \\ \sigma_{y,i}e_y \end{bmatrix}.$$

In this way the GDR problem can be solved if one (but not both) of $\sigma_{x,i}$ and $\sigma_{y,i}$ is zero. ♯

For the implicit case (4.46), denote by $\mathbf{x}_i^* = \mathbf{x}_i^*(\mathbf{a})$ the solution of the implicit footpoint problem

$$\min_{\mathbf{x}} \ D(\mathbf{x}) = (\mathbf{x}_i - \mathbf{x})^{\mathrm{T}}V^{-1}(\mathbf{x}_i - \mathbf{x}) \quad \text{subject to} \quad f(\mathbf{x},\mathbf{a}) = 0. \tag{4.52}$$

Then the generalised distance $d_i(\mathbf{a})$ is given by

$$d_i = \frac{1}{s_i}(\mathbf{x}_i - \mathbf{x}_i^*)^{\mathrm{T}}\nabla_{\mathbf{x}}f, \quad s_i = \left((\nabla_{\mathbf{a}}f)^{\mathrm{T}}V\nabla_{\mathbf{a}}f\right)^{1/2}, \text{with} \quad \frac{\partial d_i}{\partial a_j} = \frac{1}{s_i}\frac{\partial f}{\partial a_j}, \tag{4.53}$$

evaluated at $\mathbf{x} = \mathbf{x}_i^*$. Thus, the implicit generalised distance regression problem can also be posed as a standard nonlinear least-squares problem where each function evaluation involves the calculation of the optimal footpoints. If V can be factored as $V = BB^{\mathrm{T}}$, the footpoint problem (4.52) can be posed as

$$\min_{\mathbf{x}^*} \mathbf{e}^{\mathrm{T}}\mathbf{e} \quad \text{subject to} \quad f(\mathbf{x}^*,\mathbf{a}) = 0 \quad \text{and} \quad \mathbf{x} = \mathbf{x}^* + B\mathbf{e}.$$

In this way the implicit GDR problem can be solved in a numerically stable way for poorly conditioned or rank deficient uncertainty matrices V.

Example: simple GDR for implicit curves

The simple GDR problem for implicit curves can be stated as: given data points $\{(x_i,y_i)\}_1^m$ and strictly positive weights $\{(\alpha_i,\beta_i)\}_1^n$, minimise

$$\sum_{i=1}^m \alpha_i^2(x_i - x_i^*)^2 + \beta_i^2(y_i - y_i^*)^2$$

with respect to $\mathbf{a}$ and $\{(x_i^*,y_i^*)\}_1^m$ subject to the constraints $f(x_i^*,y_i^*,\mathbf{a}) = 0$, $i = 1,\ldots,m$. The theory above shows that this can be reformulated as:

$$\min_{\mathbf{a}} \sum_{i=1}^m d_i^2(\mathbf{a})$$

with

$$\begin{aligned} d_i &= \frac{1}{s_i}\left((x_i - x_i^*)f_x + (y_i - y_i^*)f_y\right), \\ \frac{\partial d_i}{\partial a_j} &= \frac{1}{s_i}\frac{\partial f}{\partial a_j}, \end{aligned}$$

where

$$\begin{aligned} f_x &= \frac{\partial f}{\partial x}, \quad \text{etc.}, \\ s_i &= \left(\frac{f_x^2}{\alpha_i^2} + \frac{f_y^2}{\beta_i^2} \right)^{1/2}, \end{aligned}$$

with all expressions evaluated at the solution (x_i^*, y_i^*) of the corresponding footpoint problem. If $\alpha_i = 1/\sigma_{x,i}$ and $\beta_i = 1/\sigma_{y,i}$, the above scheme can be written in terms of $\sigma_{x,i}$ and $\sigma_{y,i}$, with

$$s_i = \left(\sigma_{x,i}^2 f_x^2 + \sigma_{y,i}^2 f_y^2 \right)^{1/2}.$$

The footpoint problem can be written as

$$\min_{x^*, y^*} e_x^2 + e_y^2 \quad \text{subject to} \quad f(x^* y^*, \mathbf{a}) = 0, \quad x_i = x^* + \sigma_{x,i} e_x, \quad \text{and} \quad y_i = y^* + \sigma_{y,i} e_y.$$

In this formulation, the GDR problem can be solved if one (but not both) of $\sigma_{x,i}$ and $\sigma_{y,i}$ is zero. ♯

Structured least-squares approaches for explicit models. The GDR problem for explicit models (4.47) can be solved directly if inefficiently using standard nonlinear least-squares algorithms. However, the fact that $p-1$ parameters $\mathbf{u}_i^*$ only appear in p equations means that the associated Jacobian matrix of partial derivatives has a block-angular structure with the diagonal blocks corresponding to the parameters $\mathbf{u}_i^*$:

$$J = \begin{bmatrix} K_1 & & & & J_1 \\ & K_2 & & & J_2 \\ & & \ddots & & \vdots \\ & & & K_m & J_m \end{bmatrix}, \tag{4.54}$$

where K_i is the matrix of derivatives of the ith set of observation equations with respect to the parameters $\mathbf{u}_i^*$, and the border blocks J_i store their derivatives with respect to $\mathbf{a}$. The form of J is illustrated in figure 4.3.

The upper-triangular factor R of the Jacobian matrix also has a block-angular structure:

$$R = \begin{bmatrix} R_1 & & & & B_1 \\ & R_2 & & & B_2 \\ & & \ddots & & \vdots \\ & & & R_m & B_m \\ & & & & R_0 \end{bmatrix}, \tag{4.55}$$

where R_1^m, are $(p-1) \times (p-1)$ upper-triangular, $\{B_i\}_1^m$ are $(p-1) \times n$ border blocks and R_0 is the $n \times n$ upper-triangular factor corresponding to the parameters $\mathbf{a}$.

The use of structure exploiting algorithms for model fitting in metrology is discussed in [57, 65, 93, 99].

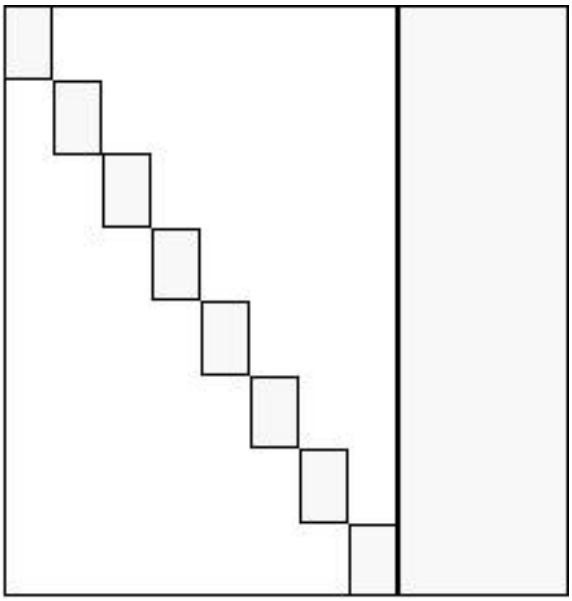

Figure 4.3: A block-angular Jacobian matrix J.

4.3.3 Approximate estimators for implicit models

We can find an approximate estimate of the solution parameters for the implicit GDR problem (4.46) by solving the least-squares problem

$$\min_{\mathbf{a}} \sum_{i=1}^{m} w_i^2 f(\mathbf{x}_i, \mathbf{a})^2,$$

where w_i are suitably chosen weights. Depending on the nature of the model and the uncertainty structure, this estimate may be fit for purpose or be used as an initial estimate in determining a refined estimate.

4.3.4 Orthogonal distance regression with linear surfaces

A linear surface in $\mathbb{R}^n$ (e.g., line in two dimensions, plane in three dimensions) is defined implicitly by an equation of the form

$$(\mathbf{x} - \mathbf{x}_0)^{\mathrm{T}} \mathbf{n} = 0,$$

where the n-vector $\mathbf{x}_0$ is a point lying in the surface and the n-vector $\mathbf{n}$ is a vector normal (orthogonal) to the surface. (Note that linear surfaces are not generally parameterised by this specification since the relationship is not one-to-one; for example any point $\mathbf{x}_0$ lying in the surface could be chosen.) The ODR problem for linear surfaces is: given data points $\{\mathbf{x}_i\}_1^m$ determine the linear surface which minimises $\sum_i d_i^2$ where $d_i = (\mathbf{x}_i - \mathbf{x}_0)^{\mathrm{T}} \mathbf{n}$ is the

distance from $\mathbf{x}_i$ to the surface. It is straightforward to show that the best-fit surface passes through the centroid $\bar{\mathbf{x}}$

$$\bar{\mathbf{x}} = \frac{1}{m}\sum_{i=1}^{m}\mathbf{x}_i$$

of the data so its equation is of the form $(\mathbf{x}-\bar{\mathbf{x}})^{\mathrm{T}}\mathbf{n} = 0$. The normal vector $\mathbf{n}$ can be determined by solving

$$\min_{\mathbf{n},\|\mathbf{n}\|=1}\sum_{i=1}^{m}((\mathbf{x}_i-\bar{\mathbf{x}})^{\mathrm{T}}\mathbf{n})^2.$$

If $\bar{X}$ is the centred data matrix with ith row equal to $(\mathbf{x}_i-\bar{\mathbf{x}})^{\mathrm{T}}$, this problem can be posed as

$$\min_{\mathbf{n},\|\mathbf{n}\|=1}\|\bar{X}\mathbf{n}\|.$$

In other words, the solution $\mathbf{n}$ is the unit vector for which the norm of $\bar{X}\mathbf{n}$ takes its minimum value. From the definition of the singular value decomposition of a matrix (section 3.8.1), we see that the solution $\mathbf{n}$ is the right singular vector of $\bar{X}$ corresponding to the smallest singular value (equation (3.8)). Thus, if $\bar{X} = USV^{\mathrm{T}}$ is the singular value decomposition of $\bar{X}$ then $\mathbf{n} = \mathbf{v}_n$ specifies the normal vector to the ODR best-fit linear surface to the data points.

4.3.5 Bibliography and software sources

The case of orthogonal distance regression is considered in [4, 28, 38, 117, 129, 130, 132, 203, 204, 208], for example. The software package ODRPACK [29] provides a fairly comprehensive facility. Generalised distance regression is considered in [1, 19, 65, 68, 91, 93, 95, 99, 101, 102, 100, 104, 105, 131]. The component XGENLINE for polynomial generalised distance regression is available for downloading from EUROMETROS [9, 87].

4.4 Generalised Gauss-Markov regression

4.4.1 Description

Generalised Gauss-Markov regression combines generalised distance regression with non-diagonal uncertainty matrices. We consider the case of a parametrically defined surface $\boldsymbol{\phi}(\mathbf{u},\mathbf{a})$, $\boldsymbol{\phi}:\mathcal{R}^{p-1}\times\mathcal{R}^n\longrightarrow\mathcal{R}^p$, and data points $\{\mathbf{x}_i\}_{i=1}^m$ nominally lying on such a surface subject to random effects characterised by an $mp\times mp$ uncertainty matrix V. We assume that V is full rank. Let $\mathbf{x}$, $\mathbf{x}^*$ and $\mathbf{f}$ be mp-vectors defined by

$$\mathbf{x} = \begin{bmatrix}\mathbf{x}_1\\ \vdots \\ \mathbf{x}_m\end{bmatrix},\quad \mathbf{x}^* = \begin{bmatrix}\mathbf{x}_1^*\\ \vdots \\ \mathbf{x}_m^*\end{bmatrix},\quad \mathbf{x}_i^* = \boldsymbol{\phi}(\mathbf{u}_i^*,\mathbf{a}),\quad \mathbf{f} = \begin{bmatrix}\mathbf{f}_1\\ \vdots \\ \mathbf{f}_m\end{bmatrix},\quad \mathbf{f}_i = \mathbf{x}_i - \mathbf{x}_i^*.$$

The *generalised Gauss-Markov regression problem* [68] is

$$\min_{\mathbf{a},\{\mathbf{u}_i^*\}}\mathbf{f}^{\mathrm{T}}V^{-1}\mathbf{f}. \tag{4.56}$$

4.4.2 Algorithms for generalised Gauss-Markov regression

The generalised Gauss-Markov problem is a type of nonlinear Gauss-Markov problem and can be solved using nonlinear least-squares algorithms (section 4.2.7) using the Cholesky factorisation of V possibly in conjunction with a generalised QR factorisation. The Jacobian matrix associated with $\mathbf{f}$ is the $mp \times (m(p-1)+n)$ matrix J with the same block-angular structure as in (4.54), with K_i representing the $p \times (p-1)$ matrix of the derivatives of $\mathbf{f}_i$ with respect to $\mathbf{u}_i^*$ and J_i representing the $p \times n$ matrix of derivatives of $\mathbf{f}_i$ with respect to $\mathbf{a}$. Since the number of computational steps for solving the generalised Gauss-Markov problem is generally of the order of m^3, for large data sets this approach is computationally expensive. See section 4.4.3, however.

4.4.3 Structured generalised Gauss-Markov problems

As with the nonlinear Gauss-Markov problem, the uncertainty matrix V often has a structure that allows the generalised nonlinear Gauss-Markov problem to be solved more efficiently. Suppose the measurement model is

$$\mathbf{x}_i = \boldsymbol{\phi}(\mathbf{u}_i, \mathbf{a}) + \boldsymbol{\epsilon}_i + H_i\boldsymbol{\epsilon}_0, \quad \boldsymbol{\epsilon}_i \in N(\mathbf{0}, U_i), \quad \boldsymbol{\epsilon}_0 \in N(\mathbf{0}, U_0)$$

where $\boldsymbol{\phi} : \mathcal{R}^{p-1} \times \mathcal{R}^n \longrightarrow \mathcal{R}^p$ is a parametric surface, $\boldsymbol{\epsilon}_i$ represents random effects specific to the ith data point $\mathbf{x}_i$ and $\boldsymbol{\epsilon}_0$ represents random effects common to all the measurements. For example, $\boldsymbol{\epsilon}_0$ could represent temperature or scale effects that influence all the measurements. The matrix H_i represents the sensitivity of the ith measurement to these effects. If U_i has factorisation $U_i = B_i B_i^T$ and U_0 has factorisation $U_0 = B_0 B_0^{\mathrm{T}}$, then the uncertainty matrix V associated with measurements $\{\mathbf{x}_i\}_{i=1}^m$ is given by

$$V = BB^{\mathrm{T}}, \qquad B = \begin{bmatrix} B_1 & & & B_{0,1} \\ & \ddots & & \vdots \\ & & B_m & B_{0,m} \end{bmatrix}, \quad B_{0,i} = H_i B_0, \tag{4.57}$$

and (4.56) can be written as

$$\min_{\mathbf{a},\{\mathbf{u}_i^*\},\mathbf{e}_0} \sum_{i=0}^{m} \mathbf{e}_i^{\mathrm{T}}\mathbf{e}_i \quad \text{subject to} \quad \mathbf{x}_i = \boldsymbol{\phi}(\mathbf{u}_i^*, \mathbf{a}) + B_i\mathbf{e}_i + B_{0,i}\mathbf{e}_0, \quad i = 1, \ldots, m. \tag{4.58}$$

Holding $\mathbf{a}$ and $\mathbf{e}_0$ fixed, it is seen that the optimal $\mathbf{u}_i^*$ must solve the footpoint problem (4.51) but for the surface

$$\bar{\boldsymbol{\phi}}_i(\mathbf{u}_i^*, \tilde{\mathbf{a}}) = \boldsymbol{\phi}(\mathbf{u}_i^*, \mathbf{a}) + B_{0,i}\mathbf{e}_0, \quad \tilde{\mathbf{a}} = \begin{bmatrix} \mathbf{a} \\ \mathbf{e}_0 \end{bmatrix}.$$

Following the same approach as described in section 4.3, we define the generalised distance $d_i(\tilde{\mathbf{a}})$ as a function of $\tilde{\mathbf{a}}$ evaluated at the solution of the ith footpoint. Then (4.58) is equivalent to

$$\min_{\tilde{\mathbf{a}}} \left\{ \mathbf{e_0}^{\mathrm{T}}\mathbf{e_0} \quad + \quad \sum_{i=1}^{m} d_i^2(\tilde{\mathbf{a}}) \right\}, \tag{4.59}$$

and can be solved using standard nonlinear least squares algorithms. This results in an algorithm that requires a number of steps linear in the number m of data points [100].

4.5 Linear Chebyshev (L_∞) estimator

4.5.1 Description

Given data $\{(\mathbf{x}_i, y_i)\}_1^m$ and the linear model

$$y = a_1\phi_1(\mathbf{x}) + \ldots + a_n\phi_n(\mathbf{x}),$$

$n \leq m$, the Chebyshev estimate of the parameters $\mathbf{a}$ is the one which solves

$$\min_{\mathbf{a}} \; F(\mathbf{a}) = \max_i |y_i - \mathbf{c}_i^{\mathrm{T}}\mathbf{a}|,$$

where $\mathbf{c}_i = (\phi_1(\mathbf{x}_i), \ldots, \phi_n(\mathbf{x}_i))^{\mathrm{T}}$. If s is the minimum value of $F(\mathbf{a})$, at least $n+1$ of the terms $|y_i - \mathbf{c}_i^{\mathrm{T}}\mathbf{a}|$ will be equal to s [185, 207]. Chebyshev estimates minimise the maximum approximation error rather than an error aggregated over all the data (as in least squares).

Chebyshev estimation is used widely in approximation where it is required to fit a curve or data set uniformly well across the range. In particular Chebyshev estimation can be regarded as a maximum likelihood estimator for linear models in which the measurements of a single response variable is subject to uncorrelated uniformly distributed random effects:

$$y_i = a_1\phi_1(\mathbf{x}_i) + \ldots + a_n\phi_n(\mathbf{x}_i) + \epsilon_i, \quad \epsilon_i \in R(-S, S), i = 1, \ldots, m \geq n.$$

Chebyshev approximation (usually nonlinear) is used in dimensional metrology to estimate the maximum departure of an artefact/manufactured part from its nominal shape.

Linear Chebyshev estimators are less suitable for data in which more than one variable is subject to significant random effects and should not be used for data which contains outliers or rogue points.

Example: averaging

In the simple case of fitting a constant to a set of values, the Chebyshev solution is the midrange, i.e., the average of the maximum and minimum values. ♯

4.5.2 Algorithms for linear Chebyshev approximation

The Chebyshev approximation problem can be reformulated as

$$\min_{\mathbf{a},s} \; s$$

subject to the linear inequality constraints

$$-s \leq y_i - \mathbf{c}_i^{\mathrm{T}}\mathbf{a} \leq s, \quad i = 1, \ldots, m.$$

This is a linear programming problem and can be solved by the simplex algorithm of Dantzig [79] (not to be confused with the simplex method of Nelder and Mead [170] for unconstrained minimisation). At the solution, at least $n+1$ of the inequalities hold as equalities so the solution can be found by determining the correct subset of $n+1$ constraints. From an initial choice of $n+1$ constraints, the simplex algorithm systematically updates this selection until the solution is found.

4.5.3 Bibliography and software sources

Linear Chebshev approximation is considered in [18, 185, 207], linear programming in [89, 115], for example. The algorithm of Barrodale and Philips [13] is widely used. There is a linear Chebyshev solver in the Matlab Optimisation Toolbox and the NAG library [158, 175] and linear programming software in the IMSL and NAG libraries [175, 206]; see also [163]. The use of Chebyshev approximation in coordinate metrology is discussed in [5, 6, 39, 40, 94, 132].

4.6 Linear L_1 estimation

4.6.1 Description

Given data $\{(\mathbf{x}_i, y_i)\}_1^m$ and the linear model

$$y = a_1\phi_1(\mathbf{x}) + \ldots + a_n\phi_n(\mathbf{x}),$$

$n \leq m$, the L_1 estimate of the parameters $\mathbf{a}$ is the one which solves

$$\min_{\mathbf{a}} \; F(\mathbf{a}) = \sum_{i=1}^{m} |y_i - \mathbf{c}_i^{\mathrm{T}}\mathbf{a}|,$$

where $\mathbf{c}_i = (\phi_1(\mathbf{x}_i), \ldots, \phi_n(\mathbf{x}_i))^{\mathrm{T}}$. At the solution, at least n of the terms $|y_i - \mathbf{c}_i^{\mathrm{T}}\mathbf{a}|$ will be zero and the L_1 estimate approximately balances the number and distribution of the vectors $\mathbf{c}_i$ associated with a positive residual with those associated with a negative residual [207]. Importantly, the magnitudes of the residuals are not important. For this reason, L_1 estimates are not particularly influenced by outliers or rogue points in the data.

Linear L_1 approximation methods are not commonly used in metrology. However, their ability to produce a good fit to the majority of the data in the presence of outliers can be very useful for systems that have normally distributed random effects in general but in which large, sporadic errors can occur, for example in measuring a surface in which there are a small number of cracks. For normally distributed random effects, the L_1 estimate can be expected to be reasonably close to a least-squares estimate.

Example: averaging

In the simple case of fitting a constant to a set of values, the L_1 solution is the median. ♯

Example: Comparing least-squares and L_1 line fits.

Figure 4.4 shows the least-squares and L_1 line fits to 12 data points with two 'outliers'. The L_1 fit (dotted line) completely ignores the large errors associated with points 3 and 11, well approximating the body of the data. In contrast, the least-squares fit is skewed towards the outliers. ♯

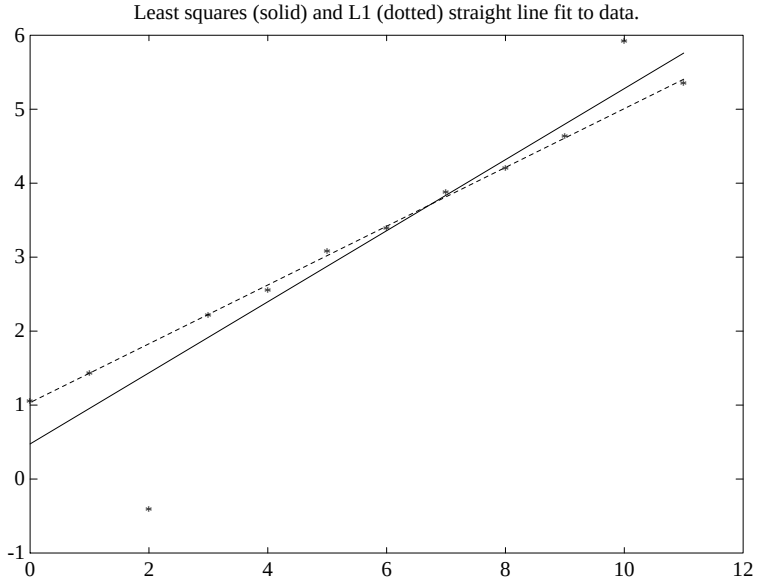

Figure 4.4: Least-squares and L_1 line fits to data with two outliers.

4.6.2 Algorithms for linear L_1 approximation

The L_1 approximation problem can be reformulated as

$$\min_{\mathbf{a},\{s_i\}} \sum_{i=1}^{m} s_i$$

subject to the linear inequality constraints

$$-s_i \leq y_i - \mathbf{c}_i^{\mathrm{T}}\mathbf{a} \leq s_i, \quad i = 1, \ldots, m.$$

At the solution $s_i = |y_i - \mathbf{c}_i^{\mathrm{T}}\mathbf{a}|$.

This is a linear programming problem and, as in the case of linear Chebyshev approximation (section 4.5), can be solved by the simplex algorithm [79]. The introduction of the potentially large number of parameters s_i means that a straightforward application of this algorithm would be inefficient. However, with modification the L_1 approximation problem can be solved effectively using a simplex-type method.

4.6.3 Bibliography and software sources

Linear L_1 approximation is considered in [14, 17, 144, 145, 185, 207], for example. The algorithms of Barrodale and Philips [15] and Bartels and Conn [16] are widely used.

4.7 Asymptotic least squares (ALS)

4.7.1 Description

Asymptotic least squares (ALS) is a form of nonlinear least-squares approximation in which a nonlinear transformation is applied in order to reduce the effect of large approximation errors associated with outliers or rogue data points. The terms *robust* and *transformed* least squares are also used. An asymptotic least-squares estimate minimises an objective function of the form

$$\tilde{F}(\mathbf{a}) = \frac{1}{2}\sum_{i=1}^{m} \tilde{f}_i(\mathbf{a})^2, \qquad \tilde{f}_i = \tau(f_i), \tag{4.60}$$

where $\tau(x)$ a transformation function having the following properties: i) τ has continuous second derivatives so that minimising $\tilde{F}$ is a smooth optimisation problem, ii) $\tau(0) = 0$, $\tau'(0) = 1$ and $\tau''(0) = 0$ so that for small f_i, $\tilde{F}$ has similar behaviour to a standard least-squares objective function, and iii) $\lim_{|x|\to\infty} \tau'(x) = 0$ so that increasing an already large approximation error will have a marginal effect on $\tilde{F}$. A simple function satisfying these criteria is

$$\tau(x) = x/(1 + c^2x^2)^{1/2}; \tag{4.61}$$

see figure 4.5. We note that $\lim_{x\to\pm\infty} \tau(x) = \pm 1/c$ and that $\tau(x)$ has the correct asymptotic behaviour.

Asymptotic least squares is appropriate for models of the form

$$y_i = \phi(\mathbf{x}_i, \mathbf{a}) + \epsilon_i + \omega_i, \quad \boldsymbol{\epsilon} \in \boldsymbol{E}, \quad E(\boldsymbol{E}) = \mathbf{0}, \quad V(\boldsymbol{E}) = \sigma^2 I,$$

and $\omega_i = 0$ for most of the measurements but there is a possibility that for some of the data points ω_i could be large relative to σ. For this model, an appropriate form of τ is

$$\tau(x) = (x/\sigma)/(1 + c^2(x/\sigma)^2)^{1/2}. \tag{4.62}$$

The parameter c in (4.62) controls the level of ϵ at which the transform takes effect (figure 4.5). If $\boldsymbol{E} \sim N(\mathbf{0}, \sigma^2 I)$, we would expect approximately 95% of the deviations $y_i - \phi(\mathbf{x}_i, \mathbf{a})$ to lie in the interval$[-2\sigma, 2\sigma]$. In this region, we want τ to make a small change, suggesting a value of c in the region of $c = 1/4$.

4.7.2 Algorithms for asymptotic least squares

Even if f_i is linear in the parameters $\mathbf{a}$ the introduction of the nonlinear τ function makes the minimisation of $\tilde{F}$ a nonlinear least-squares problem.

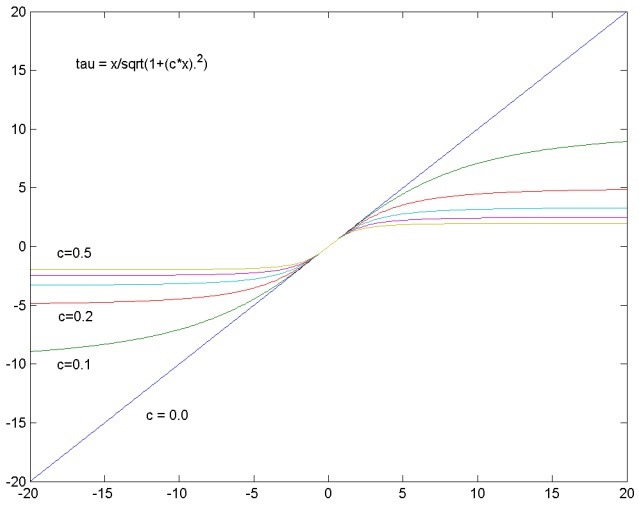

Figure 4.5: Graph of τ defined in (4.61) for different values of c.

To employ a Newton-type algorithm to minimise $\tilde{F}(\mathbf{a})$, we need to calculate

$$\mathbf{g} = \tilde{J}^{\mathrm{T}}\tilde{\mathbf{f}}, \quad \tilde{J}_{ij} = \dot{\tau}_i \frac{\partial f_i}{\partial a_j}, \quad \dot{\tau}_i = \frac{d\tau}{dx}(f_i),$$

and

$$\tilde{H} = \tilde{J}^{\mathrm{T}}\tilde{J} + \tilde{G}, \quad \tilde{G}_{jk} = \sum_i \tilde{f}_i \frac{\partial^2 \tilde{f}_i}{\partial a_j \partial a_k}.$$

We note that

$$\frac{\partial^2 \tilde{f}_i}{\partial a_j \partial a_k} = \ddot{\tau}_i \frac{\partial f_i}{\partial a_j}\frac{\partial f_i}{\partial a_k} + \dot{\tau}_i \frac{\partial^2 f_i}{\partial a_j \partial a_k}, \qquad \ddot{\tau}_i = \frac{d^2\tau}{d^2 x}(f_i).$$

The first term on the right is the contribution due to the curvature in τ, the second, due to that in F. Even if the second term is small, the first term is likely to be significant. This means that in practice the Gauss-Newton algorithm implemented for ALS will have significantly slower convergence than a Newton algorithm. However, if f is linear with $\mathbf{f} = \mathbf{y} - C\mathbf{a}$, the second term is zero and a Newton algorithm can be implemented easily with $\tilde{J}$ and $\tilde{G}$ calculated using the following identities:

$$\tilde{J}_{ij} = -c_{ij}\dot{\tau}_i, \quad \tilde{G}_{jk} = \sum_i \tau_i \ddot{\tau}_i c_{ij} c_{ik}.$$

4.7.3 Uncertainty associated with the fitted parameters

Since the ALS method is a form of nonlinear least squares the approach given in section 4.2.4 is applicable. Since the τ function is likely to introduce significant curvature, $V_{\mathbf{a}}$ evaluated using the Hessian matrix (4.36), rather than its approximation (4.37), is recommended. As with all nonlinear estimation problems, the resulting $V_{\mathbf{a}}$ is based on a linearisation and could

be significantly different from the true value. Monte Carlo techniques can be used to validate these estimates.

Example: assessment of aspheric surfaces

In determining the shape of high quality optical surfaces using measurements gathered by a coordinate measuring machine, care must be taken to ensure that the optical surface is not damaged by the contacting probe. However, using a low-force probing scheme, the presence of particles of dust on the artefact's surface introduces sporadic, large non-random effects into the measurement data. Figure 4.6 shows the residuals associated with an ALS fit of a hyperboloid surface to measurements of an aspheric mirror, a component in an earth observation camera. The spikes are due to particles of dust on the mirror or on the spherical probe. It is judged that 9 of the 401 measurements (i.e., approximately 2%) have been contaminated. Because the dust particles must necessarily have a positive diameter an asymmetric transform function τ was used in which only large, positive approximation errors are transformed. The standard noise associated with the measurements is of the order of 0.000 2 mm while the diameter of the dust particles is of the order of 0.002 mm. The difference between the ALS fitted surface and that generated using a standard (nonlinear) approach was of the order of 0.000 4 mm, and is seen to be significant relative to the standard noise. ♯

4.7.4 Bibliography and software sources

The ALS approach is described more fully in [103, 140]. Nonlinear least-squares software can be used directly to provide ALS estimates (section 4.2.11).

4.8 Robust estimators

Because of there ability to cope with outliers, the L_1 and ALS estimators are termed *robust estimators*. There are other estimation algorithms designed to cope with outliers, including the Huber M-estimator [134, 135], which behaves like a least-squares estimator for small residuals and like L_1 for outliers. In fact the Huber M-estimator can be implemented as a form of asymptotic least squares [103]. Aspects of robust estimation are considered in [62, 76, 191, 205, 208]. See also [163].

4.9 Nonlinear Chebyshev and L_1 approximation

The nonlinear Chebyshev optimisation problem is: given m functions $f_i(\mathbf{a})$, $\mathbf{a} = (a_1, \ldots, a_n)^{\mathrm{T}}$, $n \leq m$, solve

$$\min_{\mathbf{a}} \; F(\mathbf{a}) = \max_i |f_i(\mathbf{a})|. \tag{4.63}$$

The Chebyshev optimisation problem arises in data approximation with nonlinear models. Given data $\{(\mathbf{x}_i, y_i)\}_1^m$ and the nonlinear model

$$y = \phi(\mathbf{x}, \mathbf{a}),$$

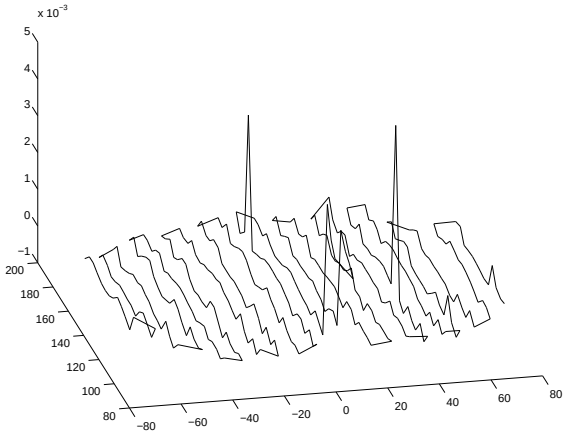

Figure 4.6: Residuals associated with an ALS fit of a hyperboloid surface to measurements of an aspheric mirror. The spikes are due to particles of dust on the mirror or on the spherical probe. The units for each axis are millimetres.

the nonlinear Chebyshev estimate of the parameters $\mathbf{a}$ is the one that solves the optimisation problem (4.63) with $f_i(\mathbf{a}) = y_i - \phi(\mathbf{x}_i, \mathbf{a})$. Chebyshev approximation problems occur frequently in dimensional metrology in which a geometric element is fitted to co-ordinate data according to the Chebyshev or related criteria.

The problem can be reformulated as

$$\min_{\mathbf{a},s} \ s$$

subject to the nonlinear constraints

$$-s \leq f_i(\mathbf{a}) \leq s, \quad i = 1, \ldots, m.$$

General purpose optimisation software can be used to solve this problem effectively.

The nonlinear L_1 optimisation problem is: given m functions $f_i(\mathbf{a}),\ n \leq m$, solve

$$\min_{\mathbf{a}} \ F(\mathbf{a}) = \sum_{i=1}^{m} |f_i(\mathbf{a})|. \tag{4.64}$$

The L_1 optimisation problem arises in data approximation involving nonlinear models with $f_i(\mathbf{a}) = y_i - \phi(\mathbf{x}_i, \mathbf{a})$.

The problem (4.64) can be formulated as

$$\min_{\mathbf{a},\{s_i\}} \sum_{i=1}^{m} \ s_i$$

subject to the constraints

$$-s_i \leq f_i(\mathbf{a}) \leq s_i, \quad i = 1, \ldots, m,$$

and solved using general purpose optimisation software. However, unlike the nonlinear Chebyshev problem, this is not a very efficient approach due to the introduction of the extra parameters s_i. An approach designed to overcome this disadvantage is described in [166].

A simpler approach to these nonlinear approximation problems is to use a Gauss-Newton strategy (section 4.2) in which at each major iteration a linear Chebyshev or L_1 problem is solved [179, 180, 207]. These algorithms can work work well on some problems, but can exhibit slow convergence on others.

4.9.1 Bibliography and software sources

Nonlinear Chebyshev and L_1 approximation are considered in [165, 166, 179, 180, 207]. There are Chebyshev (minimax) optimisation modules in the Matlab Optimisation Toolbox and the NAG library [175]. There are general purpose optimisation modules that can be applied to these problems in the NAG and IMSL libraries [175, 206]. Chebyshev approximation with geometric elements is considered in [5, 6, 39, 94].

4.10 Maximum likelihood estimation (MLE)

4.10.1 Description

Suppose $Y_i \sim D_i(\mathbf{a})$, $i = 1, \ldots, m$, are m independently distributed random variables associated with distributions $D_i(\mathbf{a})$ with PDFs $p_i(y|\mathbf{a})$ depending on n parameters $\mathbf{a} = (a_1, \ldots, a_n)^{\mathrm{T}}$, and that $\mathbf{y}$ is a set of observations of $\boldsymbol{Y}$ (denoted $\mathbf{y} \in \boldsymbol{Y}$). The likelihood function defined as $p(\mathbf{y}|\mathbf{a})$, regarded as a function of $\mathbf{a}$, is given by

$$p(\mathbf{y}|\mathbf{a}) = \prod_{i=1}^{m} p_i(y_i|\mathbf{a}).$$

The maximum likelihood estimate $\hat{\mathbf{a}}$ of $\mathbf{a}$ maximises the likelihood function $p(\mathbf{y}|\mathbf{a})$ with respect to $\mathbf{a}$. More generally, if $\boldsymbol{Y}$ has multivariate probability density $p(\mathbf{y}|\mathbf{a})$ depending on parameters $\mathbf{a}$, given an observation $\mathbf{y}$ of $\boldsymbol{Y}$, the maximum likelihood estimate maximises $p(\mathbf{y}|\mathbf{a})$ as a function of $\mathbf{a}$.

MLE is a very general parameter estimation tool used widely across science. It requires that the PDFs $p_i(y|\mathbf{a})$ are fully specified. For normally distributed random variables with known uncertainty matrices, the MLE is the same as the least-squares estimate. One of the important uses of MLE in metrology is in estimating noise parameters.

4.10.2 Algorithms for maximum likelihood estimation

Most maximum likelihood estimation algorithms determine an estimate by minimising the negative log likelihood function

$$F(\mathbf{a}|\mathbf{y}) = -\log p(\mathbf{y}|\mathbf{a}) = -\sum_{i=1}^{m} \log p_i(y_i|\mathbf{a}),$$

using a version of Newton's algorithm for function minimisation (section 3.7).

4.10.3 Uncertainty associated with the fitted parameters

The uncertainty associated with a ML estimate can be analysed in a number of ways.

Asymptotic results. Let $\mathbf{y}$ be an observation of random variables $\boldsymbol{Y}$ with associated multivariate PDF $p(\mathbf{y}|\mathbf{a})$. Let $\mathbf{y} \mapsto \mathcal{M}(\mathbf{y})$ be the maximum likelihood estimate associated with data $\mathbf{y}$. We regard the ML estimate $\hat{\mathbf{a}} = \mathcal{M}(\mathbf{y})$ as an observation of a vector of random variables $\hat{\boldsymbol{A}} = \mathcal{M}(\boldsymbol{Y})$ and the uncertainty matrix associated with $\hat{\mathbf{a}}$ is the variance matrix associated with $\hat{\boldsymbol{A}}$.

Asymptotic results (i.e., variants of the Central Limit Theorem [189]) can be used to show that if various regularity assumptions hold (to permit the interchange of integration and differentiation, for example, and ensure that various integrals are finite), then as the number of data points increases the distribution of $\hat{\boldsymbol{A}}$ approaches $N(\mathbf{a}, I^{-1}(\mathbf{a}))$ where $I(\mathbf{a})$, the

Fisher information matrix, is the expectation of the Hessian matrix $H(\mathbf{a}|\mathbf{y})$ of second partial derivatives of $F(\mathbf{a}|\mathbf{y}) = -\log p(\mathbf{y}|\mathbf{a})$:

$$I(\mathbf{a}) = \int H(\mathbf{a}|\mathbf{y}) p(\mathbf{y}|\mathbf{a})\, d\mathbf{y}, \quad H(\mathbf{a}|\mathbf{y}) = \frac{\partial^2 F}{\partial a_j \partial a_k}.$$

This matrix can be approximated by the *observed Fisher information matrix*

$$\hat{I} = H = \frac{\partial^2 F}{\partial a_j \partial a_k}(\hat{\mathbf{a}}|\mathbf{y}).$$

We therefore take as an estimate of the uncertainty matrix $V_{\mathbf{a}}$ associated with the estimates $\hat{\mathbf{a}}$

$$V_{\mathbf{a}} = \hat{I}^{-1} = H^{-1}. \tag{4.65}$$

The asymptotic results show that as the number of measurements (information) increases the estimates $\hat{\mathbf{a}}$ approach $\mathbf{a}$, so that MLE is asymptotically unbiased. The inverse of the Fisher information matrix $I^{-1}(\mathbf{a})$ represents a lower bound on the variance of any unbiased estimator and the ML estimates attains this lower bound asymptotically. This means that as the number of measurements increases, the variance matrix associated with a maximum likelihood estimate will become at least as small as that for any other unbiased estimator.

For a large number data points, the distribution of the ML estimate $\hat{\mathbf{a}}$ given $\mathbf{a}$ is approximately normal

$$\hat{\mathbf{a}}|\mathbf{a} \sim N(\mathbf{a}|V), \quad V = H^{-1}.$$

In a Bayesian context, we assume $\mathbf{a}$ is a parameter vector rather than a fixed unknown. If there is no substantive prior information about $\mathbf{a}$, and $\hat{\mathbf{a}}|\mathbf{a} \sim N(\mathbf{a}, V)$, the posterior distribution for $\mathbf{a}$, given that we have observed the ML estimate $\hat{\mathbf{a}}$ is

$$\mathbf{a}|\hat{\mathbf{a}} \sim N(\hat{\mathbf{a}}, V).$$

The symmetry associated with these distributions arises from the fact that $\hat{\mathbf{a}}$ and $\mathbf{a}$ appear symmetrically through the term

$$(\mathbf{a} - \hat{\mathbf{a}})^{\mathrm{T}} V^{-1} (\mathbf{a} - \hat{\mathbf{a}})$$

in the two distributions.

Propagation of uncertainty. The estimate in (4.65) is based on the asymptotic behaviour of the ML estimator as the number of measurements increases. We can instead use linearisation to provide an estimate of the uncertainty matrix associated with the ML estimates. At the minimum of $F(\mathbf{a}|\mathbf{y})$, the gradient $\mathbf{g}(\mathbf{a}|\mathbf{y}) = \nabla_{\mathbf{a}} F = \mathbf{0}$ and these n equations define $\mathbf{a} = \mathbf{a}(\mathbf{y})$ as functions of $\mathbf{y}$. If K is the sensitivity matrix

$$K_{ji} = \frac{\partial a_j}{\partial y_i}$$

and $V_{\mathbf{y}}$ is the uncertainty matrix associated with $\mathbf{y}$, i.e., the variance matrix associated with $\boldsymbol{Y}$, then

$$V_{\mathbf{a}} \approx K V_{\mathbf{y}} K^{\mathrm{T}}.$$

Taking differentials of the equation $\mathbf{g}(\mathbf{a}(\mathbf{y}), \mathbf{y}) = \mathbf{0}$, we have

$$HK + G_{\mathbf{y}} = 0, \quad G_{\mathbf{y}}(j,i) = \frac{\partial^2 F}{\partial a_j \partial y_i},$$

so that

$$K = -H^{-1} G_{\mathbf{y}},$$

and

$$V_{\mathbf{a}} \approx H^{-1} G_{\mathbf{y}} V_{\mathbf{y}} G_{\mathbf{y}}^{\mathrm{T}} H^{-1}.$$

The sensitivity matrix K is evaluated at the expected value of $\mathbf{y}$. In the context of model fitting, the expected value of $\mathbf{y}$, given $\mathbf{a}$, is $\boldsymbol{\phi}(\mathbf{a})$.

Example: linear models with Gaussian random effects

If the model equations are

$$\mathbf{y} \sim N(C\mathbf{a}, V_{\mathbf{y}}),$$

then

$$F(\mathbf{a}|\mathbf{y}) = \frac{1}{2}(\mathbf{y} - C\mathbf{a})^{\mathrm{T}} V_{\mathbf{y}}^{-1} (\mathbf{y} - C\mathbf{a}),$$

and

$$\mathbf{g} = -C^{\mathrm{T}} V_{\mathbf{y}}^{-1} (\mathbf{y} - C\mathbf{a}), \quad H = C^{\mathrm{T}} V_{\mathbf{y}}^{-1} C, \quad G_{\mathbf{y}} = -C^{\mathrm{T}} V_{\mathbf{y}}^{-1},$$

so that

$$\begin{aligned} V_{\mathbf{a}} = H^{-1} G_{\mathbf{y}} V_{\mathbf{y}} G_{\mathbf{y}}^{\mathrm{T}} H^{-1} &= (C^{\mathrm{T}} V_{\mathbf{y}}^{-1} C)^{-1} C^{\mathrm{T}} V_{\mathbf{y}}^{-1} V_{\mathbf{y}} V_{\mathbf{y}}^{-1} C (C^{\mathrm{T}} V_{\mathbf{y}}^{-1} C)^{-1} \\ &= (C^{\mathrm{T}} V_{\mathbf{y}}^{-1} C)^{-1} = H^{-1}. \end{aligned}$$

In this case the propagation of uncertainty estimate is the same as that derived from the observed Fisher information matrix. ♯

Gaussian approximation to the posterior distribution. In the Bayesian context (section 4.10), in the absence of substantive prior information for $\mathbf{a}$, the Gaussian approximation to the posterior distribution $p(\mathbf{a}|\mathbf{y})$ for $\mathbf{a}$ given $\mathbf{y}$ is $N(\hat{\mathbf{a}}, V)$, $V = H^{-1}$, where $\hat{\mathbf{a}}$ is the ML estimate and H is the Hessian matrix of second partial derivatives of $-\log p(\mathbf{y}|\mathbf{a})$ evaluated at $\hat{\mathbf{a}}$, so that H is the observed Fisher information matrix. As the number of data points increases, the asymptotic results show that the posterior distribution will become more like a Gaussian distribution so that the Gaussian approximation becomes a better representation.

4.10.4 Maximum likelihood estimation for multiple noise parameters

While maximum likelihood estimation (MLE) has broad application and applies to quite arbitrary distributions, in metrology a common application is in taking into account multiple random effects associated with a measurement system. The following example will illustrate the concepts involved. Suppose a measurement system is characterised by

$$y_i = (1 + \delta_i)\phi(\mathbf{x}_i, \mathbf{a}) + \epsilon_i, \quad \delta_i \in N(0, \sigma_S^2), \quad \epsilon_i \in N(0, \sigma_A^2), \tag{4.66}$$

where δ_i represents a random effect applying to the measurement scale and ϵ_i is an effect independent of scale. The probability $p(y_i|\mathbf{a}, \sigma_A, \sigma_S)$ of observing y_i given that $\mathbf{a}$, σ_A and σ_S (and $\mathbf{x}_i$) are known is such that

$$p(y_i|\mathbf{a}, \sigma_A, \sigma_S) \propto \frac{1}{\sigma_i} \exp\left\{-\frac{1}{2\sigma_i^2}(y_i - \phi(\mathbf{x}_i, \mathbf{a}))^2\right\}, \quad \sigma_i^2 = \sigma_A^2 + \sigma_S^2\phi^2(\mathbf{x}_i, \mathbf{a}),$$

so that the probability of observing a data vector $\mathbf{y} = (y_1, \ldots, y_m)^{\mathrm{T}}$ is such that

$$p(\mathbf{y}|\mathbf{a}, \sigma_A, \sigma_S) \propto \left(\prod_{i=1}^{m} \frac{1}{\sigma_i}\right) \exp\left\{-\frac{1}{2}\sum_{i=1}^{m} f_i^2\right\}, \quad f_i = f_i(\mathbf{a}, \sigma_A, \sigma_S) = \frac{y_i - \phi(\mathbf{x}_i, \mathbf{a})}{\sigma_i}.$$

The ML estimates of the parameters is found by minimising

$$F(\mathbf{a}, \sigma_A, \sigma_S) = \sum_{i=1}^{m} \log \sigma_i + \frac{1}{2}\sum_{i=1}^{m} f_i^2.$$

Note that even if σ_A and σ_S are regarded as known, this function does not represent a sum of squares since the first term involves $\mathbf{a}$ through σ_i. A simplifying approximation is to set $\hat{\sigma}_i^2 = \sigma_A^2 + \sigma_S^2 y_i$, so that the unknown $\phi(\mathbf{x}_i, \mathbf{a})$ is approximated by the measured response y_i. With this approximation, for the case σ_A and σ_S known, F above simplifies to a sum of squares.

This example can be generalised to cover the case

$$\mathbf{y} \sim N(\boldsymbol{\phi}(\mathbf{a}), V(\mathbf{a}, \boldsymbol{\sigma})), \quad \phi_i(\mathbf{a}) = \phi(\mathbf{x}_i, \mathbf{a}),$$

where the uncertainty matrix V depends potentially on $\mathbf{a}$ and noise parameters $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_{n_K})^{\mathrm{T}}$. Letting $\mathbf{b}$ denote the complete set of parameters $\mathbf{b}^{\mathrm{T}} = (\mathbf{a}^{\mathrm{T}}, \boldsymbol{\sigma}^{\mathrm{T}})$, the ML estimates of the parameters are found by minimising

$$F(\mathbf{b}) = \frac{1}{2}\log|(V(\mathbf{b})| + \frac{1}{2}\mathbf{f}^{\mathrm{T}}V^{-1}(\mathbf{b})\mathbf{f}, \quad f_i = y_i - \phi(\mathbf{x}_i, \mathbf{a}), \tag{4.67}$$

where $|V|$ denotes the determinant of V. In performing this optimisation, the derivatives of $|V(\mathbf{b})|$ and $V^{-1}(\mathbf{b})$ need to be calculated. For the first of these, we can use the formula (derived from *Jacobi's formula* for the derivative of a determinant)

$$\frac{\partial}{\partial b_j}\log|V| = \mathrm{Tr}\left(V^{-1}\frac{\partial V}{\partial b_j}\right),$$

where $\mathrm{Tr}(A)$ denotes the *trace* of a matrix, the sum of its diagonal elements. For the second, we can use the formula

$$\frac{\partial V^{-1}}{\partial b_j} = -V^{-1}\left(\frac{\partial V}{\partial b_j}\right)V^{-1}.$$

However, we can work instead with a factored form of V. For example, if $V = LL^{\mathrm{T}}$ has Cholesky factor $L = L(\mathbf{b})$, then $\log|V| = \sum_{i=1}^{m}\log l_{ii}^2$. The elements of the derivative matrix $\dot{L} = \partial L/\partial b_j$ is defined by the relationship

$$\dot{L}L^{\mathrm{T}} + L\dot{L}^{\mathrm{T}} = \dot{V},$$

from which an algorithm to compute the elements of $\dot{L}$ can be developed. We first assign $\dot{L}$ to be the lower triangle of $\dot{V} = \partial V/\partial b_j$. Then the steps

```
for j = 1 : m
    if j > 1
      L̇(j, j) := [L̇(j, j) − 2L(j, 1 : j − 1)L̇(j, 1 : j − 1)^T]/(2L(j, j))
    else
      L̇(j, j) := L̇(j, j)/(2L(j, j))
    end
    for k = j + 1 : m
       L̇(k, j) = [L̇(k, j) − L(k, 1 : j)L̇(j, 1 : j)^T − L̇(k, 1 : j − 1)L(j, 1 : j − 1)^T]/L(j, j)
    end
end
```

completes the calculation of $\dot{L}$ from L and $\dot{V}$. If $\tilde{\mathbf{f}} = L^{-1}\mathbf{f}$, then $\mathbf{f}V^{-1}\mathbf{f} = \tilde{\mathbf{f}}^{\mathrm{T}}\tilde{\mathbf{f}}$ and

$$\frac{\partial \tilde{\mathbf{f}}}{\partial b_j} = L^{-1}\left(\frac{\partial \mathbf{f}}{\partial b_j} - L^{-1}\dot{L}\tilde{\mathbf{f}}\right),$$

involving the solution of equations involving the lower triangular matrix L. However, the minimisation of $F(\mathbf{b})$ in (4.67) can also be posed as

$$\min_{\mathbf{b}} \sum \log l_{ii}(\mathbf{b}) + \frac{1}{2}\mathbf{e}^{\mathrm{T}}\mathbf{e} \quad \text{subject to} \quad \mathbf{y} = \boldsymbol{\phi}(\mathbf{a}) + L(\mathbf{b})\mathbf{e},$$

a constrained optimisation problem, but one which avoids potential problems with the calculation of the inverse of V or L [31].

4.10.5 Partially characterised noise parameters

The approach described above is quite general. However, the optimisation problems that arise will only be well posed if the data contains enough information from which estimates of the noise parameters can be derived. For example, in the model (4.66), if the observed responses y_i are all approximately at the same level, then there is no information to discriminate between the additive and scale effects. As discussed in relation to linear and nonlinear least squares, sections 4.1.5 and 4.2.5, often we have prior information about the variance associated with random effects which can help resolve such ambiguities. If the prior distribution for $\boldsymbol{\sigma}$ is $p(\boldsymbol{\sigma})$, then corresponding to (4.67), estimates of the parameters $\mathbf{b}$ are found by minimising

$$F(\mathbf{b}) = \frac{1}{2}\log|V(\mathbf{b})| + \frac{1}{2}\mathbf{f}^{\mathrm{T}}V^{-1}(\mathbf{b})\mathbf{f} - \log p(\boldsymbol{\sigma}).$$

As before, prior information about σ_k could be expressed as

$$m_{0,k}\sigma_{0,k}^2\eta_k \sim \chi^2_{m_{0,k}}, \eta_k = 1/\sigma_k^2,$$

where $\sigma_{0,k}^2$ is the prior estimate of σ_k^2 and $m_{0,k}$ specifies the degree of belief in that estimate. Setting $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_k)^{\mathrm{T}}$,

$$-\log p(\boldsymbol{\eta}) = \sum_k \left\{\frac{\eta_k}{2}m_{0,k}\sigma_{0,k}^2 - (m_{0,k}/2 - 1)\log\eta_k\right\},$$

up to an additive constant. We note that for $m_{0,k} > 2$, minimum of this function is given by $\sigma_k^2 = 1/\eta_k = m_{0,k}\sigma_{0,k}^2/(m_{0,k} - 2)$. The factor $m_{0,k}/(m_{0,k} - 2)$ arises from the fact that the χ^2_ν has expected value ν but mode $\nu - 2$, $\nu > 2$.

4.10.6 Marginalising noise parameters

The posterior distribution $p(\mathbf{a}, \boldsymbol{\sigma}|\mathbf{y})$ for $\mathbf{a}$ and $\boldsymbol{\sigma}$, given data $\mathbf{y}$, is such that

$$p(\mathbf{a}, \boldsymbol{\sigma}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{a}, \boldsymbol{\sigma})p(\boldsymbol{\sigma}).$$

Estimates of $\mathbf{a}$ and $\boldsymbol{\sigma}$ can be obtained by maximising this posterior distribution. If we are not particularly interested in the values of $\boldsymbol{\sigma}$ we might want instead to maximise the marginalised posterior distribution

$$p(\mathbf{a}|\mathbf{y}) = \int p(\mathbf{a}, \boldsymbol{\sigma}|\mathbf{y}) \, \mathrm{d}\boldsymbol{\sigma},$$

particularly if this marginalised distribution can be calculated straightforwardly, as in the following example.

Example: fusing data from a number of sensors

Suppose the observations are associated with a number of sensors, and that the measurement model for the kth sensor is

$$y_i = \eta_i(\boldsymbol{\alpha}) + \epsilon_i, \qquad \epsilon_i \in N(0, \sigma_k^2), \qquad i \in I_k.$$

Here, I_k is the set of m_k indices corresponding to the measurements taken by the kth sensor, $k = 1, \ldots, n_K$. This example covers the case of generalised distance regression

$$y_i = \phi_i(\mathbf{x}_i^*, \boldsymbol{\alpha}) + \epsilon_i, \quad \mathbf{x}_i = \mathbf{x}_i^* + \boldsymbol{\delta}_i, \quad x_{k,i} = x_{k,i}^* + \delta_{k,i}, \quad \delta_{k,i} \in N(0, \sigma_k^2), \quad \epsilon_i \in N(0, \sigma_{n_K}^2),$$

where both the response y and covariates $\mathbf{x}$ are measured subject to random effects. We assume that the partial information about $\eta_k = 1/\sigma_k$ is specified as

$$m_{0,k}\sigma_{0,k}^2\eta_k \sim \chi_{m_{0,k}}^2.$$

The posterior probability distribution $p_k(\mathbf{a}, \eta_k|\mathbf{y}_k)$ given the kth sensor results $\mathbf{y}_k$ is such that

$$p_k(\mathbf{a}, \eta_k|\mathbf{y}_k) \propto \eta_k^{\frac{m_k+m_{0,k}}{2}-1} \exp\left\{-\frac{\eta_k}{2}\left[m_{0,k}\sigma_{0,k}^2 + F_k(\mathbf{a})\right]\right\}, \qquad F_k(\mathbf{a}) = \sum_{i \in I_k}(y_i - \phi_i(\mathbf{a}))^2.$$

If $\mathbf{b} = (\mathbf{a}^{\mathrm{T}}, \boldsymbol{\eta}^{\mathrm{T}})^{\mathrm{T}}$, the posterior density $p(\mathbf{b}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{b})p(\boldsymbol{\eta})$ is maximised at the minimum of

$$F(\mathbf{b}) = \sum_{k=1}^{n_K}\left\{\eta_k[m_{0,k}\sigma_{0,k}^2 + F_k(\mathbf{a})] - (m_{0,k} + m_k - 2)\log\eta_k\right\}. \tag{4.68}$$

At the minimum, the solution $\eta_k = 1/\sigma_k^2$ satisfies

$$\sigma_k^2 = \frac{m_{0,k}\sigma_{0,k}^2 + F_k(\mathbf{a})}{m_{0,k} + m_k - 2}.$$

The objective function in (4.68) can be compared with $F_M(\mathbf{b})$ derived from the likelihood $p(\mathbf{y}|\mathbf{b})$ which has the form

$$F_M(\mathbf{b}) = \sum_{k=1}^{n_K}\left\{\eta_k F_k(\mathbf{a}) - m_k \log\eta_k\right\},$$

which provides solution estimates

$$\sigma_k^2 = \frac{F_k(\mathbf{a})}{m_k}.$$

The case $m_{0,k} = 0$ in (4.68) is not equivalent to the maximum likelihood solution since the posterior density $p(\mathbf{b}|\mathbf{y})$ involves the non-informative priors $p(\eta_k) = 1/\eta_k$.

Since the parameter η_k appears only in $p_k(\boldsymbol{\alpha}, \eta_k|\mathbf{y}_k)$, the marginalised posterior distribution is

$$p(\mathbf{a}|\mathbf{y}) = \int p(\mathbf{a}, \boldsymbol{\eta}|\mathbf{y}) \mathrm{d}\boldsymbol{\eta} = \prod_k \int p_k(\mathbf{a}, \eta_k|\mathbf{y}_k) d\eta_k = \prod_k p_k(\mathbf{a}|\mathbf{y}_k),$$

the product of the marginalised distributions. Using the integration rule (4.15),

$$p(\mathbf{a}|\mathbf{y}) \propto \prod_{k=1}^{n_K} \left[m_{0,k}\sigma_{0,k}^2 + F_k(\mathbf{a})\right]^{-\frac{m_{0,k}+m_k}{2}},$$

and estimates of the parameters can be found by minimising

$$F(\mathbf{a}) = \sum_{k=1}^{n_K} (m_{0,k} + m_k) \log \left[m_{0,k}\sigma_{0,k}^2 + F_k(\mathbf{a})\right].$$

♯

4.11 Sampling from posterior distributions

Bayes's theorem states that the posterior distribution $p(\mathbf{a}|\mathbf{y})$ for parameters $\mathbf{a}$, given data $\mathbf{y}$ is such that

$$p(\mathbf{a}|\mathbf{y}) = K^{-1} p(\mathbf{y}|\mathbf{a}) p(\mathbf{a}), \quad K = \int p(\mathbf{y}|\mathbf{a}) p(\mathbf{a}) \, \mathrm{d}\mathbf{a}. \tag{4.69}$$

For all but simple problems, the key difficulty in working with the posterior distribution is in determining the constant of integration K. Maximum likelihood estimation gets round this difficulty by approximating the posterior distribution by a multivariate normal distribution $\hat{p}(\mathbf{a})$, $\mathbf{a} \sim N(\hat{\mathbf{a}}, V)$, derived from a quadratic approximation to $\log p(\mathbf{a}|\mathbf{y})$. This approximation can be used to determine parameter estimates and associated uncertainties. However, there is no guarantee that this approximation will be adequate, especially for nonlinear models and a small number of data points.

An alternative approach is to use Markov chain Monte Carlo (MCMC) simulation methods to create a set of points $\{\mathbf{a}_q\}$ sampled from the posterior distribution $p(\mathbf{a}|\mathbf{y})$ and then base estimates, uncertainties and coverage intervals on information derived straightforwardly from $\{\mathbf{a}_q\}$, as in standard Monte Carlo methods [71]. The term Markov chain is used in these sampling methods because the distribution for the $(q+1)$th term in the chain depends only on $\mathbf{a}_q$, not on any previous step: $p(\mathbf{a}_{q+1}|\mathbf{a}_q) = p(\mathbf{a}_{q+1}|\mathbf{a}_q, \mathbf{a}_{q-1}, \ldots, \mathbf{a}_1)$. MCMC methods have general application and can in theory be used to sample from any distribution $p(\mathbf{a})$. MCMC methods can be thought of as applying an iterative operation to $\mathbf{a}_q$ to obtain the next estimate $\mathbf{a}_{q+1}$. As the chain progresses, the behaviour of the chain is determined by the asymptotic properties of the iterative operation. The situation is similar to the behaviour of $\mathbf{x}_{q+1} = A(\mathbf{x}_q/\|\mathbf{x}_q\|)$, the repeated application of a symmetric matrix to a vector. If the

eigenvalues λ_i of A are such that $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$, then, independent of the starting point, the sequence $\mathbf{x}_q$ converges to the eigenvector $\mathbf{v}_1$ corresponding to the unique largest eigenvalue λ_1. (This is the basis of the power method for finding eigenvectors of a matrix [117, section 8.2].) In MCMC, the limiting eigenvector corresponds to a limiting probability distribution from which the $\mathbf{a}_q$ are eventually sampled.

In order to apply this approach, it is necessary to design the chain such that the limiting distribution coincides with $p(\mathbf{a})$. Surprisingly, there are a number of straightforward ways to achieve this. One approach is the Metropolis-Hastings MCMC algorithm. Suppose we wish to sample $\{\mathbf{a}_q\}$ from a distribution $p(\mathbf{a})$. Given a draw $\mathbf{a}_{q-1}$, a proposed new draw $\mathbf{a}^*$ for the next member of the sequence is drawn at random from a *proposal distribution* $q(\mathbf{a}|\mathbf{a}_{q-1})$. Then $\mathbf{a}_q$ is set to $\mathbf{a}^*$ with acceptance probability

$$p_q = \min\{1, r_q\} \quad r_q = \frac{p(\mathbf{a}^*)q(\mathbf{a}_{q-1}|\mathbf{a}^*)}{p(\mathbf{a}_{q-1})q(\mathbf{a}^*|\mathbf{a}_{q-1})}. \tag{4.70}$$

The simplest way to implement the acceptance step is to draw u_q from the uniform distribution $R(0,1)$ and if $u_q < r_q$, set $\mathbf{a}_q = \mathbf{a}^*$, otherwise set $\mathbf{a}_q = \mathbf{a}_{q-1}$. The role of this acceptance probability is to ensure that the probability of going from $\mathbf{a}$ to $\mathbf{b}$ is the same as that of going from $\mathbf{b}$ to $\mathbf{a}$. This *reversibility* property leads to $p(\mathbf{a})$ being the limiting distribution of the chain. The important practical feature of this acceptance probability is that $p(\mathbf{a})$ and $q(\mathbf{a}^*|\mathbf{a})$ need only be known up a constant since $p(\mathbf{a})$ appears as a ratio $p(\mathbf{a}^*)/p(\mathbf{a}_{q-1})$, etc. If $p(\mathbf{a}|\mathbf{y})$ is a posterior distribution as in (4.69), this ratio can be calculated in terms of $p(\mathbf{y}|\mathbf{a})p(\mathbf{a})$ without the need to calculate the constant of integration K.

After a number of iterations that allow the Markov chain to converge, the sampled $\{\mathbf{a}_q\}$ are drawn from the target distribution. The number of iterations necessary to ensure convergence is difficult to predict and most implementations perform a number of repeat simulations with different initial samples to gauge if the chains have converged to the target distribution. One such scheme is given below.

To implement the algorithm it is necessary to generate the random draw from $q(\mathbf{a}|\mathbf{a}_{q-1})$ and evaluate the acceptance probability p_q. If $\hat{p}(\mathbf{a})$ is a distribution that approximates $p(\mathbf{a})$ then we can set $q(\mathbf{a}|\mathbf{a}_{q-1})$ to be $\hat{p}(\mathbf{a})$ (so the draw $\mathbf{a}^*$ is independent of the current step $\mathbf{a}_{q-1}$), in which case

$$r_q = \frac{p(\mathbf{a}^*)\hat{p}(\mathbf{a}_{q-1})}{p(\mathbf{a}_{q-1})\hat{p}(\mathbf{a}^*)}.$$

If $\hat{p}(\mathbf{a}) = p(\mathbf{a})$, then $r_q = 1$ and the proposed $\mathbf{a}^*$ is always accepted. For $\hat{p}(\mathbf{a})$ different from $p(\mathbf{a})$, the role of p_q is to modify the draws from the proposal distribution so that they become draws from the target distribution. The approximating distribution needs to strike a balance between making a proposal that stands a reasonable chance of being accepted while ensuring that all the areas of significant density are sampled. In particular, if the proposal distribution has a smaller variance than the target distribution, then the chain may take many steps to form a representative sample from the target distribution.

In parameter estimation, a natural choice for the approximating distribution is that associated with the multivariate Gaussian $N(\hat{\mathbf{a}}, \eta^{-1}V)$ where the extra scale parameter η can be used to adjust the variance of the approximating distribution to that of the target distribution. The parameter η can be tuned in an initial phase so that the acceptance rate is of the order 0.20 to 0.4, and then fixed to generate the samples $\mathbf{a}_q$ [113].

The simulation scheme (for $\eta = 1$) can be implemented as follows.

I Minimise $F(\mathbf{a}) = -\log p(\mathbf{a}|\mathbf{y})$ to determine estimate $\hat{\mathbf{a}}$ and Hessian matrix H of second order partial derivatives of F evaluated at $\hat{\mathbf{a}}$. (It is sufficient to evaluate $F(\mathbf{a})$ as $F(\mathbf{a}) = -\log p(\mathbf{y}|\mathbf{a}) - \log p(\mathbf{a})$.)

II Calculate the Cholesky factorization $H = LL^{\mathrm{T}}$ of H and set $B = L^{-\mathrm{T}}$. The variance matrix for the Gaussian approximant is $V = H^{-1} = BB^{\mathrm{T}}$.

III Draw $\mathbf{e}_0 \in N(0, I)$, so that $\mathbf{e}_0$ is an n-vector of independent, normally distributed random numbers and set $\mathbf{a}_0 = \hat{\mathbf{a}} + B\mathbf{e}_0$, $F_0 = F(\mathbf{a}_0)$ and $\hat{F}_0 = \mathbf{e}_0^{\mathrm{T}}\mathbf{e}_0$.

IV For $q = 1, \ldots, M$,

i Draw $\mathbf{e}^* \in N(0, I)$ and set $\mathbf{a} = \hat{\mathbf{a}} + B\mathbf{e}^*$, $F^* = F(\mathbf{a}^*)$ and $\hat{F}^* = (\mathbf{e}^*)^{\mathrm{T}}\mathbf{e}^*$.

ii Evaluate the ratio

$$r_q = \exp\{F_{q-1} - F^* + \hat{F}^* - \hat{F}_{q-1}\}.$$

iii Draw $u \in R(0, 1)$. If $u < r_q$, set

$$\mathbf{a}_q = \mathbf{a}^*, \quad F_q = F^*, \quad \hat{F}_q = \hat{F}^*.$$

Otherwise, set

$$\mathbf{a}_q = \mathbf{a}_{q-1}, \quad F_q = F_{q-1}, \quad \hat{F}_q = \hat{F}_{q-1}.$$

At steps III and IVii, $\mathbf{a}_0$ and $\mathbf{a}^*$ are draws from $N(\hat{\mathbf{a}}, V)$. At steps III and IVi, $\hat{F}_0 = \log \hat{p}(\mathbf{a}_0|\hat{\mathbf{a}}, V)$ and $\hat{F}^* = \log \hat{p}(\mathbf{a}^*|\hat{\mathbf{a}}, V)$, up to the same additive constant. At step IViii, the test on u drawn from the uniform distribution defined on the interval [0,1] ensures that $\mathbf{a}^*$ is accepted with probability $p_q = \min\{1, r_q\}$.

Test on convergence. The following scheme can be used to check the convergence of a chain by comparing the behaviour of chains of the same length generated using different starting points [113, section 11.6]. Suppose that we have samples $\mathbf{a}_{q,r}$, $q = 1, 2, \ldots, N$, $r = 1, \ldots, M$ from M chains of length N. The length N will typically be of the order of 5,000 – 10,000 and M may be of the order of 10. For each parameter $a = a_j$, we make the following calculations:

$$\bar{a}_{.r} = \frac{1}{N}\sum_{q=1}^{N} a_{q,r}, \quad \bar{a}_{..} = \frac{1}{M}\sum_{r=1}^{M} \bar{a}_{.r}, \quad B = \frac{N}{M-1}\sum_{r=1}^{M}(\bar{a}_{.r} - \bar{a}_{..})^2,$$

and

$$s_r^2 = \frac{1}{N-1}\sum_{q=1}^{N}(a_{q,r} - \bar{a}_{.r})^2, \quad W = \frac{1}{M}\sum_{r=1}^{M} s_r^2.$$

The quantity B represents the variance between the chains, and W the variance within the chains. The variance of the distribution associated with $a|\mathbf{y}$ is estimated by

$$V^+ = \frac{M-1}{M}W + \frac{1}{M}B.$$

If the variance for the proposal $\hat{p}(\mathbf{a})$ distribution is greater than the target distribution (as recommended to ensure that the whole of $p(\mathbf{a})$ is sampled), then this estimate will represent

an overestimate, but is unbiased in the limit as $N \to \infty$. On the other hand, the within variance $V^- = W$ can be expected to represent an underestimate because, for finite N, each chain will not have had an opportunity to range over all the target distribution. As $N \to \infty$, we expect the ratio

$$R = \left(\frac{V^+}{V^-}\right)^{1/2},$$

to approach 1 from above. This ratio represents the potential reduction in the estimate of the standard deviation of the distribution for $a|\mathbf{y}$ as $N \to \infty$. If R is less than 1.05, the expected improvement in the estimate of the standard deviation by letting the chains run longer will be no more that 5 %.

Figure 4.7 shows the discrete approximation derived from MCMC simulations to the distribution $p(a|y_2)$ discussed in section 3.10; see Figure 3.22. The proposal distribution in this case was simply a uniform distribution defined on the interval [-1,1.5]. Here is the Matlab code used to generate the MCMC samples.

```
A = zeros(N,M);

for r = 1:M                          % M chains

    a = 2.5*rand(1,1)-1;
    F = (y-a^3)^2/(2*sigmay^2);

    for q = 1:N                      % Chains of length N

        as = 2.5*rand(1,1)-1;        % Uniform proposal
        Fs = (y-as^3)^2/(2*sigmay^2);
        ratio = exp(F-Fs);
        u = rand(1,1);

        if u < ratio                 % Accept proposal
            a = as; F = Fs;
        end

        A(q,r) = a;

    end
end
```

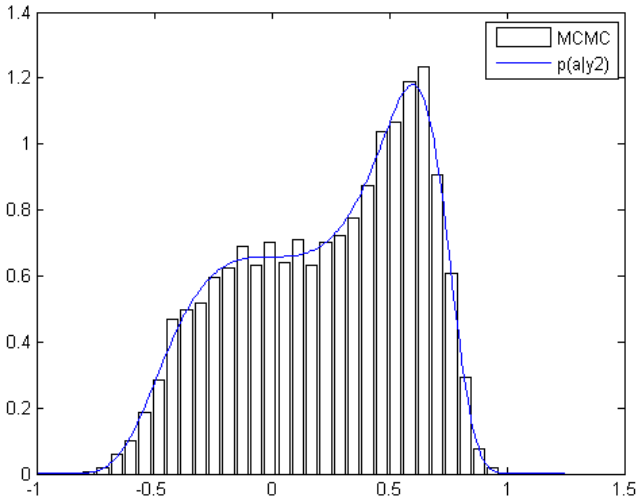

Figure 4.7: Distribution $p(a|y_2)$ and its discrete approximation determined by MCMC sampling.

Chapter 5

Discrete models in metrology

In this chapter we describe some common models used in metrology.

5.1 Polynomial curves

5.1.1 Description

Polynomials provide a class of linear models that are used extensively as empirical models for experimental data. A polynomial of degree n can be written as

$$f_n(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n = \sum_{j=0}^{n} a_j x^j = \sum_{j=0}^{n} a_j \phi_j(x),$$

where $\phi_j(x) = x^j$ are the *monomial basis* functions. (The indexing starts at zero so that the index matches the exponent.) A polynomial of degree 1 is a straight line, degree 2 a quadratic curve, etc. The immediate appeal of polynomials is that computation with polynomials requires only addition and multiplication.

5.1.2 Advantages and disadvantages

Polynomials are good for:

- Representing a smooth curve $y = \phi(x)$ or data generated from a smooth curve over a fixed interval $[x_{\min}, x_{\max}]$. They are extremely flexible and from the mathematical point of view can be used to approximate any smooth curve to a given accuracy by choosing a high enough degree. They are used, for example, to represent calibration curves of sensors.

Polynomials are not good for:

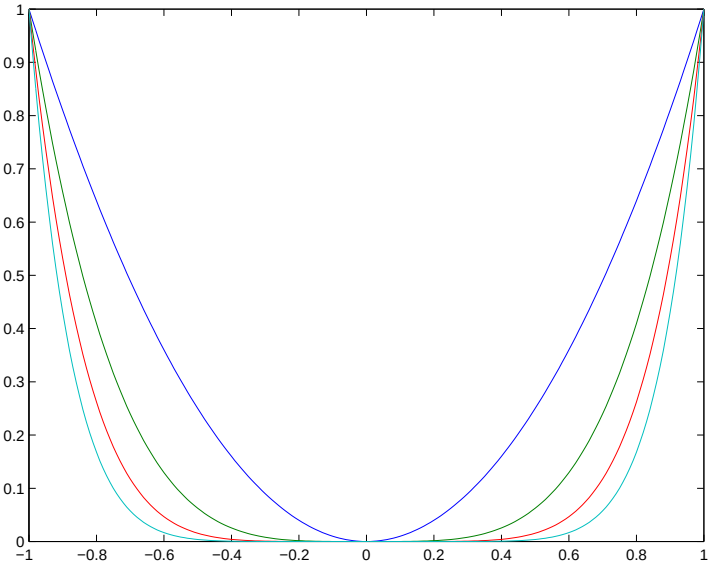

Figure 5.1: Graphs of x^{2j}, $j = 1, 2, 3, 4$, on the interval $[-1,\ 1]$.

- Representing curves or data with sharp discontinuities in value or slope.
- Describing asymptotic behaviour where the curve approaches a straight line as the variable x gets larger in magnitude (section 4.7).

5.1.3 Working with polynomials

While the description of polynomials in terms of the monomial basis functions makes clear the nature of polynomial functions, the use of the monomial basis in numerical computation leads to severe numerical difficulties. A first difficulty is that for values of the variable x significantly greater than one in absolute value, the terms x^j become very large as j increases. This problem is solved by working with a normalised variable z. If x varies within the range $[x_{\min}, x_{\max}] = \{x : x_{\min} \leq x \leq x_{\max}\}$, then

$$z = \frac{(x - x_{\min}) - (x_{\max} - x)}{x_{\max} - x_{\min}} = \frac{x - (x_{\max} + x_{\min})/2}{(x_{\max} - x_{\min})/2}, \tag{5.1}$$

and all its powers lie in the range $[-1, 1]$. (The first expression for evaluating z above has better numerical properties [66].) For small degree polynomials ($n \leq 4$, say), this normalisation is sufficient to remove most numerical difficulties.

The second difficulty arises from the fact that, especially for large j, the basis function ϕ_j looks very similar to ϕ_{j+2} in the range $[-1, 1]$. Figure 5.1 presents the graphs of $\phi_{2j} = x^{2j}$ $j = 1, 2, 3, 4$. We can regard polynomial functions defined on $[-1, 1]$ as members of a vector space of functions. In this vector space, the angle between two polynomials $p(x)$ and $q(x)$

n	$[-1,1]$	$[0,2]$	$[4,6]$	$[19,21]$
5	2	4	9	15
10	4	9	16	24
20	10	18	*	*

Table 5.1: Estimates of the number of decimal digits lost using the monomial basis functions for different degrees and intervals. An entry $*$ indicates the system was too ill-conditioned for the calculation to be made.

can be determined in terms of integrals involving their product, e.g.,

$$\int_{-1}^{1} p(x)q(x)w(x)dx,$$

where $w(x)$ is a weighting function. With this definition of angle, it is straightforward to show that the monomial basis functions ϕ_j and ϕ_{j+2} point in the roughly the same direction (in the sense that the angle between them is small), leading to ill-conditioning. This ill-conditioning worsens rapidly as the degree increases and the variable values move further from zero. Table 5.1 gives an estimate of the number of decimal digits lost using the monomial basis functions generated by 31 values $\{x_i\}_1^{31}$ randomly distributed in the interval $[-1,1]$ and subsequently translated to the intervals $[0,2]$, $[4,6]$, $[19,21]$. From the table, it is easy to see why polynomials are sometimes thought to be of very limited use because of numerical stability problems. In fact, it is their representation (i.e., parameterisation) in terms of the monomial basis functions which leads to instability, not polynomials *per se*.

Alternative representations can be derived by finding basis functions with better properties.

The *Chebyshev* polynomials $T_j(x)$ are one such set of basis functions and have the property that they are orthogonal to each other on the interval $[-1,\ 1]$ with respect to the weighting function $w(x) = 1/(1+x^2)^{1/2}$. They are defined by

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_j(x) = 2xT_{j-1}(x) - T_{j-2}(x), \quad j \geq 2.$$

Chebyshev polynomials can also be defined using the trigonometrical relationship

$$T_j(\cos\theta) = \cos j\theta, \quad \cos\theta = x.$$

Figure 5.2 presents the graphs of T_2 to T_5. Conventionally, T_0 is replaced by $T_0/2$ in the basis, so that

$$f_n(x) = \frac{1}{2}a_0T_0(x) + a_1T_1(x) + \ldots + a_nT_n(x) = {\sum_{j=1}^{n}}' a_jT_j(x);$$

the notation $\sum'$ indicates that the first term is halved.

Using orthogonal polynomials in conjunction with the variable transformation formula (5.1) it is possible to use high degree polynomial models over any interval in a numerically stable way [106, 201]. Algorithms based on Chebyshev polynomials have been implemented in NPL's Data Approximation Subroutine Library — DASL [8] — (and other libraries) and used successfully for many years. It is disappointing that there are still many polynomial

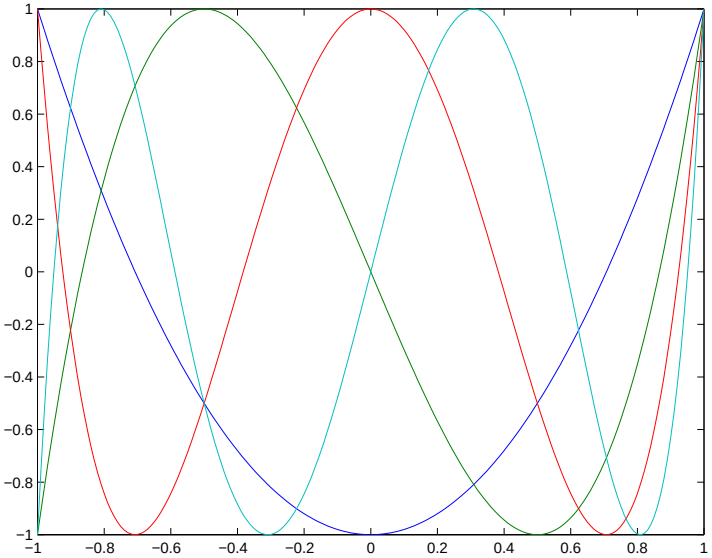

Figure 5.2: Chebyshev polynomials T_i, $i = 2, \ldots, 5$.

regression packages available for PCs that implement algorithms based on the standard monomial representation and are therefore prone to produce unreliable results. It should be emphasised that operations with a Chebyshev representation are, in essence, no more complicated than those using a monomial basis.

Example: evaluating a polynomial from a Chebyshev representation

A Chebyshev representation of a polynomial $p = p(x, \mathbf{a})$ of degree n ($n > 0$) is given in terms of the Chebyshev parameters (coefficients) $\mathbf{a} = (a_0, \ldots, a_n)^{\mathrm{T}}$ and constants $x_{\min}$ and $x_{\max}$ giving the range. The following scheme can be used to evaluate p at x.

I Calculate the normalised variable

$$z = \frac{(x - x_{\min}) - (x_{\max} - x)}{x_{\max} - x_{\min}}.$$

II Set $p = a_0/2 + a_1 z$, $t_0 = 1$, $t_1 = z$.

III for $j = 2 : n$

$$\begin{aligned} t_j &= 2zt_{j-1} - t_{j-2}, \\ p &= p + a_j t_j. \end{aligned}$$

DASL uses Clenshaw's recurrence to evaluate a polynomial from its Chebyshev representation: it requires fewer multiplications and has slightly superior floating-point error properties. [48, 53, 74, 114] ♯

Example: least-squares regression with polynomials using a Chebyshev representation

The following steps determine the least-squares best-fit polynomial of degree n ($n > 0$) to data $\{(x_i, y_i)\}_{i=1}^m$ using a Chebyshev representation. It follows the same approach as the general method described in section 4.1 for fitting a linear model to data, forming the observation matrix C whose jth column is the jth basis function evaluated at x_i, i.e., in this case, $C(i,j) = T_{j+1}(x_i)$.

I Calculate $x_{\min} = \min_i x_i$ and $x_{\max} = \max_i x_i$.

II Calculate the normalised variables

$$z_i = \frac{(x_i - x_{\min}) - (x_{\max} - x_i)}{x_{\max} - x_{\min}}, \quad i = 1, \ldots, m.$$

III Calculate the $m \times (n+1)$ observation matrix C, column by column using the recurrence relationship. For each i:

III.1 $C(i,1) = 1$, $C(i,2) = z_i$,

III.2 for $j = 3 : n+1$, $C(i,j) = 2z_i C(i,j-1) - C(i,j-2)$.

III.3 Adjust the first column: $C(i,1) = C(i,1)/2$.

IV Solve in the least-squares sense

$$C\mathbf{a} = \mathbf{y}.$$

If the linear least-squares problem is solved using a QR factorisation of the augmented matrix $[C \ \mathbf{y}]$ as described in section 4.1.2, it is possible to determine from the same orthogonal factorisation the least-squares polynomials of all degrees up to n (and the norms of the corresponding residual error vectors). This makes it very efficient to determine a range of polynomial fits to the data from which to select a best fit and is extremely useful in model validation; see, for example, [59, 60]. ♯

Other operations such as calculating the derivative of a polynomial are straightforward using a Chebyshev representation.

Example: derivative of a polynomial using a Chebyshev representation

If p is an n-degree polynomial with Chebyshev coefficients $\mathbf{a} = (a_0, \ldots, a_n)^{\mathrm{T}}$ defined on the range $[x_{\min}, x_{\max}]$ then its derivative $p' = \partial p / \partial x$ is a degree $n-1$ polynomial on the same range and can therefore be represented in terms of Chebyshev coefficients $\mathbf{b} = (b_0, \ldots, b_{n-1})^{\mathrm{T}}$. The coefficients $\mathbf{b}$ are calculated directly from $\mathbf{a}$ and $x_{\min}$ and $x_{\min}$:

I Set $b_{n+1} = b_n = 0$.

II for $j = n, n-1, \ldots, 2, 1$,

$$b_{j-1} = b_{j+1} + \frac{4ja_j}{x_{\max} - x_{\min}}.$$

♯

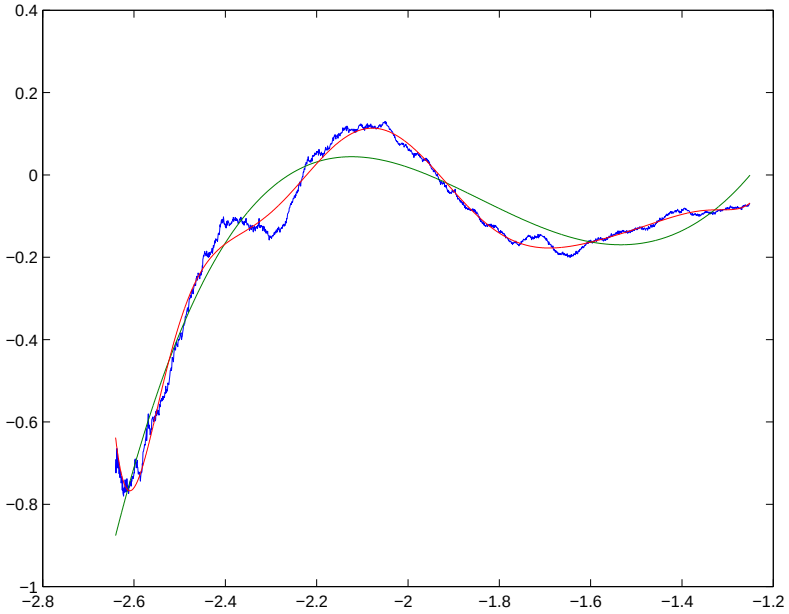

Figure 5.3: Least-squares polynomials of degrees 4 and 10 to 2000 data points.

Example: polynomial fits to data

As an example of polynomial fits, figure 5.3 shows the least-squares polynomials of degrees 4 and 10 to 2000 data points, while figure 5.4 shows the least-squares polynomial of degree 18. ♯

There are other numerical approaches to polynomial regression. Given data $\{(x_i, y_i)\}_1^m$ and weights $\{w_i\}_1^m$ the Forsythe method implicitly determines a set of basis functions ϕ_j that are orthogonal with respect to the inner product defined by

$$\sum_{i=1}^{m} w_i f(x_i) g(x_i).$$

The method of solution exploits this orthogonality, using the fact that the observation matrix C that is generated is orthogonal, so that $C^{\mathrm{T}}C$ is a diagonal matrix and the normal equations can thus be solved very simply. The use of the normal equations is numerically safe since C is perfectly well conditioned. The set of orthogonal polynomial are generated specifically for the data $\{x_i\}$ and $\{w_i\}$. By contrast, the Chebyshev polynomials are much more versatile since they are defined in the same way for all data sets.

5.1.4 Bibliography and software sources

Approximation with polynomials is one of the main topics in data and function approximation. See, for example, [49, 106, 108, 122, 185, 201, 207]. Software for polynomial approx-

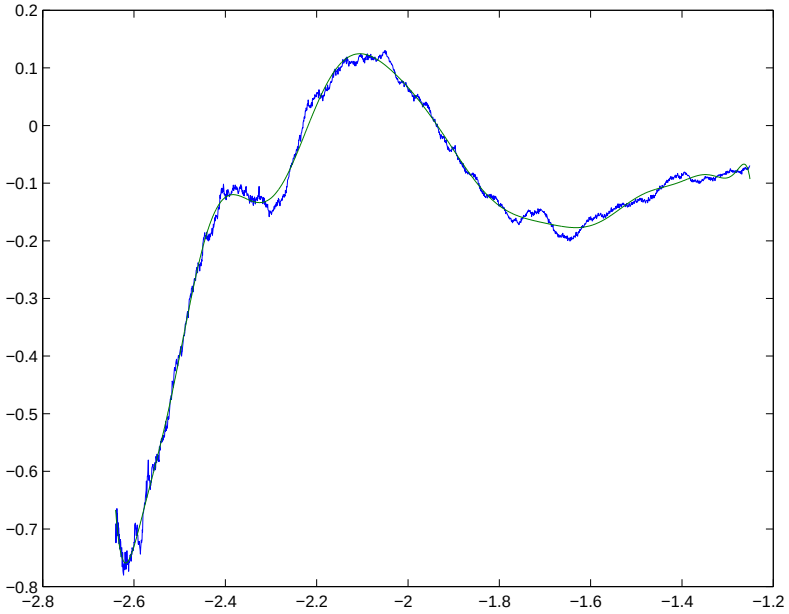

Figure 5.4: Least-squares polynomial of degree 18 to data.

imation appears in the NAG and IMSL libraries [175, 206] and there are a large number of software routines associated with polynomials available through Netlib [82]. NPL's Data Approximation Subroutine Library (DASL) and NPLFit package have extensive facilities for polynomial approximation [8, 174]. NPLFit, in particular, is aimed at metrological applications and has easy-to-use facilities for determining polynomial fits and associated uncertainties. NPLFit available as a package for downloading from EUROMETROS [9, 87].

5.2 Polynomial spline curves

5.2.1 Description

Like polynomials, polynomial spline curves — splines for short — are a class of linear models widely used for modelling discrete data. A spline $s(x)$ of order n defined over an interval $[x_{\min}, x_{\max}]$ is composed of sections of polynomial curves $p_k(x)$ of degree $n-1$ joined together at fixed points $\{\lambda_k\}_1^N$ in the interval.

Consider the case where there is one knot, at λ:

$$x_{\min} < \lambda < x_{\max},$$

and suppose we wish to build a continuous curve using two cubic polynomial curves

$$s(x) \quad = \quad p_1(x, \mathbf{a}) = a + a_1 x + a_2 x^2 + a_3 x^3, \quad x \in [x_{\min}, \lambda],$$

$$= \quad p_2(x, \mathbf{b}) = b + b_1 x + b_2 x^2 + b_3 x^3, \quad x \in [\lambda, x_{\max}].$$

We impose smoothness constraints by insisting that the function values for both curves are equal at λ and so are the first and second derivatives. (If, in addition, we were to insist that the third derivatives are equal we would force $\mathbf{a} = \mathbf{b}$.) We can show that if s satisfies these three continuity constraints, it can be written in the form

$$s(x, \mathbf{a}, c) = p_1(x, \mathbf{a}) + c(x - \lambda)_+^3,$$

where $(x - \lambda)_+ = x - \lambda$ if $x > \lambda$ and 0 otherwise.

In general, if s is a spline of order n with continuity up to the $(n-2)$nd derivative on a set of N knots $\{\lambda_k\}_1^N$ with

$$x_{\min} < \lambda_1 < \lambda_2 < \ldots < \lambda_N < x_{\max}$$

then s can be written uniquely as

$$s(x, \mathbf{a}, \mathbf{c}) = p(x, \mathbf{a}) + \sum_{k=1}^{N} c_k (x - \lambda_k)_+^{n-1}, \tag{5.2}$$

where $p(x, \mathbf{a})$ is a polynomial of degree $n-1$. The number of parameters required to define s is $n + N$ (order + number of interior knots) and s is a linear combination of the polynomial basis functions and the *truncated power functions*

$$\phi_k(x) = (x - \lambda_k)_+^{n-1}.$$

B-spline basis functions. The representation (5.2) can be used to define an explicit method of constructing a polynomial spline. In practice, using this representation can give rise to severe numerical problems (because of ill-conditioning) and, in addition, has major efficiency drawbacks. Practically all calculations using spline functions are performed using a B-spline representation of the form

$$s(x, \mathbf{a}) = \sum_{j=1}^{n+N} a_j N_{n,j}(x, \boldsymbol{\lambda}), \tag{5.3}$$

where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_N)^{\mathrm{T}}$ is the interior knot set satisfying

$$x_{\min} = \lambda_0 < \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N < \lambda_{N+1} = x_{\max}, \tag{5.4}$$

and $N_{n,j}(x, \boldsymbol{\lambda})$ are the B-spline basis functions of order n (i.e., degree $n-1$). The basis functions $N_{n,j}(x, \boldsymbol{\lambda})$ are specified by the interior knot set $\boldsymbol{\lambda} = \{\lambda_k\}_1^N$, range limits

$$x_{\min} = \lambda_0, \text{and} \quad x_{\max} = \lambda_{N+1},$$

and the additional exterior knots, λ_j, $j = 1-n, \ldots, -1$ and $j = N+2, \ldots, N+n$. These exterior knots are usually assigned to be

$$\lambda_j = \begin{cases} x_{\min}, & j < 0, \\ x_{\max}, & j > N+1. \end{cases}$$

With this choice, the basis functions are defined by the interior knots $\boldsymbol{\lambda}$ and the range constants $x_{\min}$ and $x_{\max}$. The use of coincident knots with $\lambda_j = \ldots = \lambda_{j+k}$ allows us a

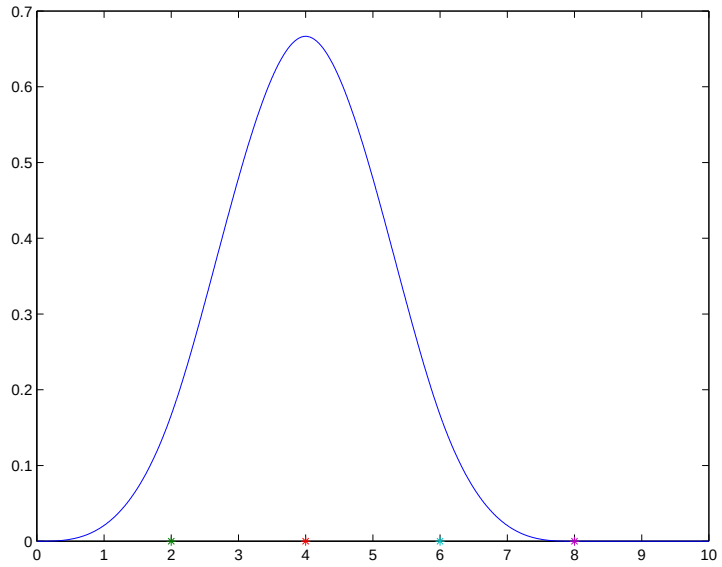

Figure 5.5: B-spline basis function $N_{4,4}(x, \boldsymbol{\lambda})$ defined on the interval $[0, 10]$ with knot set $\boldsymbol{\lambda} = (2, 4, 6, 8)^{\mathrm{T}}$.

greater degree of discontinuity at λ_j. We use $q = n + N$ to denote the number of basis functions.

A common choice of order is $n = 4$, splines constructed from cubic polynomials — *cubic splines* — because they give sufficient smoothness for most metrology applications. Figure 5.5 graphs a B-spline basis function for a cubic spline defined on the interval $[0, 10]$ with knot set $\boldsymbol{\lambda} = (2, 4, 6, 8)^{\mathrm{T}}$. Figure 5.6 graphs all eight $(= n + N)$ basis functions for this knot set.

The B-spline basis functions have a number of valuable properties including:

$$\begin{aligned} N_{n,j}(x) &\geq 0, \\ N_{n,j}(x) &= 0, x \notin [\lambda_{j-n}, \lambda_j] \quad \text{(compact support)}, \\ \sum_j N_{n,j}(x) &\equiv 1, x \in [x_{\min}, x_{\max}]. \end{aligned} \tag{5.5}$$

Using a B-spline basis, calculations with splines can be performed in a numerically stable way.

5.2.2 Typical uses

Splines are used in much the same way as polynomials, but have additional capabilities. Splines are good for:

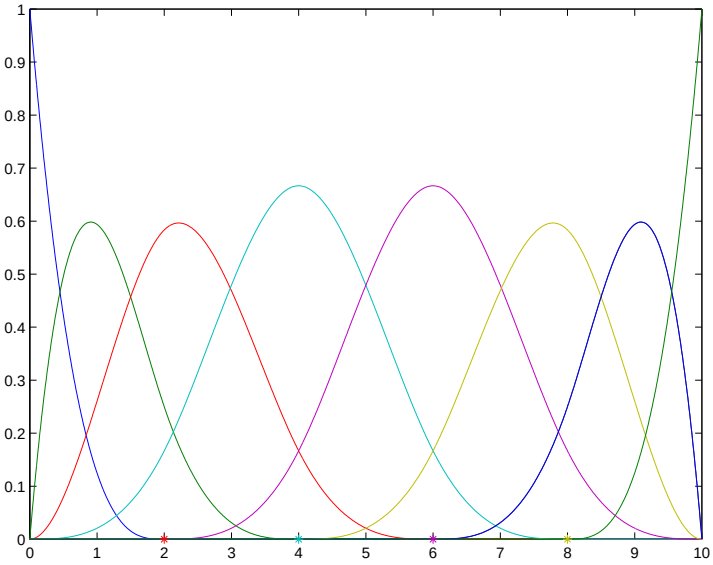

Figure 5.6: B-spline basis functions $N_{4,j}(x, \boldsymbol{\lambda})$ defined on the interval $[0, 10]$ with knot set $\boldsymbol{\lambda} = (2, 4, 6, 8)^{\mathrm{T}}$.

- Representing a smooth curve $y = \phi(x)$ or data generated from a smooth curve over a fixed interval $x \in [x_{\min}, x_{\max}]$. They are extremely flexible and from the mathematical point of view can be used to approximate any smooth curve to a given accuracy by choosing sufficient number of knots or a high enough order (degree). They are used, for example, to represent calibration curves of sensors.
- Because spline approximation can be made computationally very efficient, splines are used to represent very large sets of data.
- Splines can be use to represent curves with varying characteristics and sharp changes in shape or discontinuities, provided a suitable set of knots is used.

Splines are less good for:

- Describing asymptotic behaviour where the curve approaches a straight line as the variable x gets larger in magnitude.

Because of their flexibility, splines are used in many applications areas of mathematical modelling.

5.2.3 Working with splines

As with polynomials, it is essential to use an appropriate set of basis functions. The representation using B-splines (equation (5.3), above) is strongly recommended. Since, for

a specified set of knots, splines form a linear model, calculations involving splines centre around evaluating the basis functions $N_{n,j}(x, \boldsymbol{\lambda})$. Like Chebyshev polynomials, the basis function $N_{n,j}$ can be evaluated using a three-term recurrence relationship. The first order B-spline basis functions $N_{1,j}$ $j = 1, \ldots, N+1$ are defined by

$$\begin{aligned} N_{1,j}(x) &= \left\{ \begin{array}{ll} 1, & x \in [\lambda_{j-1}, \lambda_j), \\ 0, & \text{otherwise}, \end{array} \right\} j = 1, \ldots, N, \\ N_{1,N+1}(x) &= \left\{ \begin{array}{ll} 1, & x \in [\lambda_N, \lambda_{N+1}], \\ 0, & \text{otherwise}, \end{array} \right. \end{aligned}$$

and, for $n > 1$,

$$N_{n,j}(x) = \begin{cases} \frac{\lambda_j - x}{\lambda_j - \lambda_{j-n+1}} N_{n-1,j}(x), & j = 1, \\ \frac{x - \lambda_{j-n}}{\lambda_{j-1} - \lambda_{j-n}} N_{n-1,j-1}(x) + \frac{\lambda_j - x}{\lambda_j - \lambda_{j-n+1}} N_{n-1,j}(x), & 1 < j < N+n, \\ \frac{x - \lambda_{j-n}}{\lambda_{j-1} - \lambda_{j-n}} N_{n-1,j-1}(x), & j = N+n. \end{cases}$$

The first order B-spline basis functions equal one on a knot interval $[\lambda_{j-1}, \lambda_j)$ and zero elsewhere. An order n B-spline basis function is the weighted convex combination of two "adjacent" order $n-1$ B-spline basis functions.

Once the basis functions have been defined, spline evaluation and data fitting with splines can be performed following the general scheme for linear models.

Example: evaluating a spline in terms of its B-spline basis

A spline $s = s(x, \mathbf{a})$ of order n can be defined in terms of the B-spline coefficients (parameters) $\mathbf{a} = (a_1, \ldots, a_q)$, the interior knot set $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_N)^{\mathrm{T}}$ and constants $x_{\min}$ and $x_{\max}$ giving the range. The following scheme can be used to evaluate s at x.

I Evaluate the B-spline basis functions $N_{n,j}(x)$, $j = 1, \ldots, q = n + N$, using the recurrence relations.

II Set

$$s(x) = \sum_{j=1}^{q} a_j N_{n,j}(x). \tag{5.6}$$

s is usually evaluated by a recurrence involving the a_j, see [54]. ♯

Example: least-squares regression with splines using a B-spline representation

The following steps determine the least-squares best-fit spline of order n with a given knot set $\boldsymbol{\lambda}$ and range $[x_{\min}, x_{\max}]$ to data $\{(x_i, y_i)\}_{i=1}^m$ using a B-spline representation. It is assumed that the knots satisfy

$$x_{\min} < \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N < x_{\max},$$

and that $x_{\min} \leq x_i \leq x_{\max}$, $i = 1, \ldots, m$.

I Evaluate the B-spline basis functions $N_{n,j}(x_i)$, $j = 1, \ldots, q = n + N$, $i = 1, \ldots, m$, using the recurrence relations.

II Evaluate the $m \times q$ observation matrix C defined by $C(i,j) = N_{n,j}(x_i)$.

III Solve in the least-squares sense

$$C\mathbf{a} = \mathbf{y}.$$

♯

Other operations such as calculating the derivative of a spline are equally straightforward using a B-spline representation.

Example: derivative of a spline using a B-spline representation

Let $s = s(x, \mathbf{a})$ be a spline of order n defined in terms of the B-spline coefficients (parameters) $\mathbf{a} = (a_1, \ldots, a_q)^{\mathrm{T}}$, $q = n + N$, the interior knot set $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_N)^{\mathrm{T}}$ and range $[x_{\min}, x_{\max}]$. Its derivative $s' = \partial s/\partial x$ is an $(n-1)$th order spline defined by coefficients $\mathbf{b} = (b_1, \ldots, b_{q-1})^{\mathrm{T}}$, with

$$b_j = \begin{cases} (n-1)\frac{a_{j+1}-a_j}{\lambda_j - \lambda_{j-n+1}}, & \lambda_j > \lambda_{j-n+1}, \\ a_{j+1} - a_j, & \lambda_j = \lambda_{j-n+1}, \end{cases} \qquad j = 1, \ldots, q-1.$$

♯

Two features arise in working with splines that do not appear in approximation with general linear models. The first is the banded structure in the observation matrix and the second is the choice of knot set.

Banded structure in the observation matrix. The compact support property (equation (5.5)) of the B-spline basis functions means that for any $x \in [x_{\min}, x_{\max}]$ at most n of the basis functions $N_{n,j}(x)$ will be nonzero at x. More specifically, if $x \in [\lambda_{j-1}, \lambda_j)$, then only $N_{n,j}$, $N_{n,j+1}, \ldots, N_{n,j+n-1}$ can be nonzero. Thus, to evaluate an order n spline at any given point, only n basis functions need to be evaluated (and the inner product step (5.6) involves at most n nonzero contributions.) More importantly, any row of the observation matrix C has at most n nonzero elements appearing contiguously, i.e., adjacent to each other along the row, giving the observation matrix a *banded structure*. Figure 5.7 shows schematically (a) the structure of the observation matrix C for fitting a cubic spline (i.e., $n = 4$) with four (i.e., $N = 4$) interior knots to 11 ordered data points $(x_i, y_i)_1^{11}$, $x_i \leq x_{i+1}$ and (b) the structure of the triangular factor R determined from a QR factorisation of C (section 4.1).

The banded structure can be exploited effectively in solving the linear least squares system that arises using an orthogonal factorisation approach. The main consequence of this is that the fitting procedure can be accomplished in $O(mn^2)$ steps (i.e., in a number of steps proportional to mn^2) rather than $O(m(N+n)^2)$ if a general, full matrix approach is used. In other words, for a fixed order of spline ($n = 4$ a common choice), the computation time using a structure-exploiting approach is essentially proportional to the number m of data points and independent of the number of knots N. Using a full-matrix approach, the computation time is approximately proportional to mN^2 for a large number of knots. This efficiency saving is significant, particularly for large knot sets and is one of the reasons why splines are so popular and effective in modelling data.

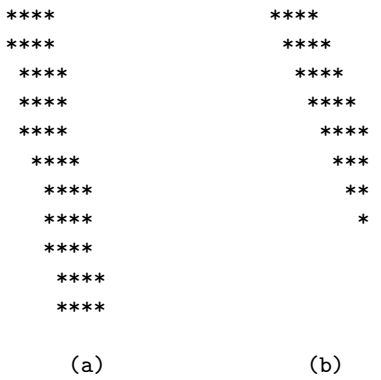

Figure 5.7: Schematic representation of (a) the structure of the observation matrix C for fitting a cube spline ($n = 4$) with four ($N = 4$) interior knots to 11 ordered data points $(x_i, y_i)_1^{11}$, $x_i \leq x_{i+1}$ and (b) the structure of the triangular factor R determined from a QR factorisation of C.

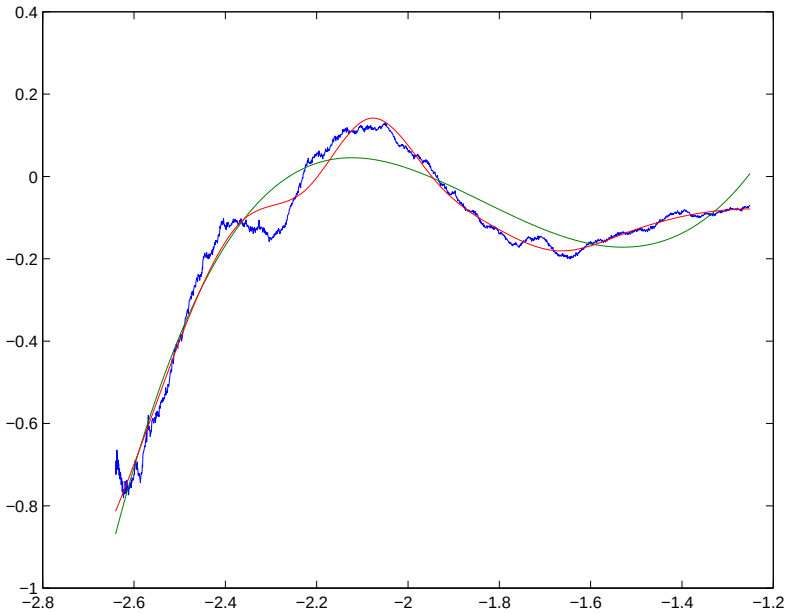

Figure 5.8: Least-squares cubic splines ($n = 4$) with one and seven interior knots to 2000 data points.

Choice of knot set. In approximation using polynomials, the main choice that a user has is fixing the degree of the polynomial. In spline approximation, the user has to fix the order (usually set at a small number with four the most common choice) and also has the much greater flexibility in fixing the number and location of the interior knots $\boldsymbol{\lambda}$ (subject to the constraints on ordering (5.4)). The knot placement can have a considerable effect on the quality of the fit, but there is no usable set of criteria that can be used to determine an optimal placement strategy (although there is much research in addressing aspects of this problem). However, there are a number of guidelines that help the user to arrive at a good set of knots. We assume that we wish to fit an nth order spline to m data points $\{(x_i, y_i)\}_1^m$.

- The number of knots N must be less than or equal to $m - n$ (i.e. $q = n + N \leq m$) in order to be able to determine all the coefficients (otherwise the observation matrix C would be rank deficient). Generally, we are looking for the smallest number of knots that provides a good fit.
- The knots λ_j should be interspersed with the abscissae $\{x_i\}$. One set of conditions (Schoenberg-Whitney) state that there should be a subset $\{t_1, \ldots, t_q\} \subset \{x_1, \ldots, x_m\}$ such that

$$t_j < \lambda_j < t_{j+n}, \quad j = 1, \ldots, N.$$

- More knots are needed in regions where the curve underlying the data is rapidly changing, fewer knots where the curve is relatively smooth.

The goodness of fit is, naturally, a qualitative attribute often assessed from a visual examination of the fit to the data. If the fit does not follow the data adequately in a region, more knots should be added, perhaps adjusting nearby knots. If the fit seems to be following the noise in the data in some regions, then knots should be removed from those regions and the remaining knots possibly adjusted. After say three or four passes, a satisfactory fit can often be attained.

Example: spline fit to data

As an example of spline fits, figure 5.8 shows the least-squares cubic splines ($n = 4$) with one and seven interior knots to 2000 data points, while figure 5.9 shows the cubic spline least-squares fit with 15 interior knots. In figure 5.10, we can compare this latter fit with a polynomial fit of degree 18 to the same data. Note that both the polynomial and spline are defined by 19 basis functions. The spline is seen to be more flexible and able to follow the shape of the data more closely. ♯

5.2.4 Bibliography and software sources

Algorithms for working with splines in terms of their B-spline representation are given in [52, 54, 55, 56, 81]. Software for spline interpolation and approximation appear in the NAG and IMSL libraries [175, 206], the Matlab spline toolbox [158], and various spline packages available through Netlib [82]. Algorithms for knot placement are described in [73, 72, 148].

Because of the computational efficiency gains to be made using structured solvers, it is recommended that special purpose spline approximation packages are used rather that standard

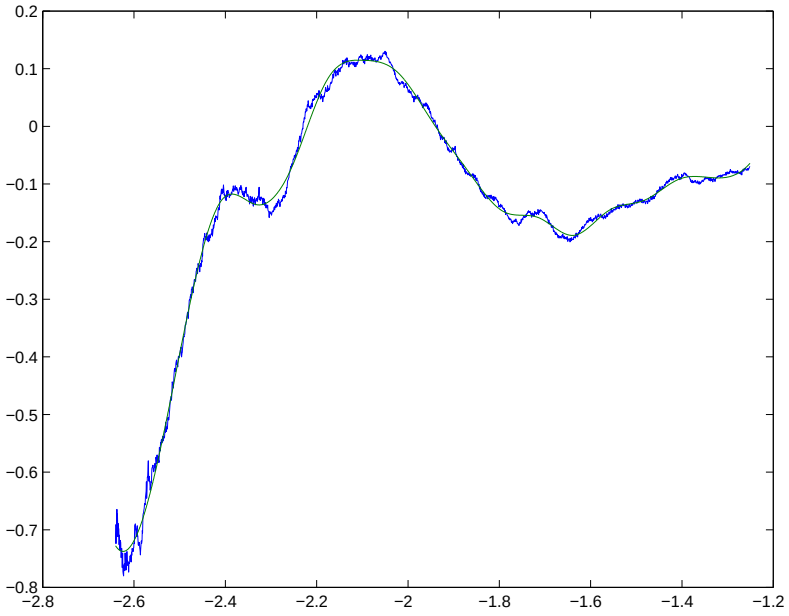

Figure 5.9: Least-squares cubic spline ($n = 4$) with 15 interior knots to data.

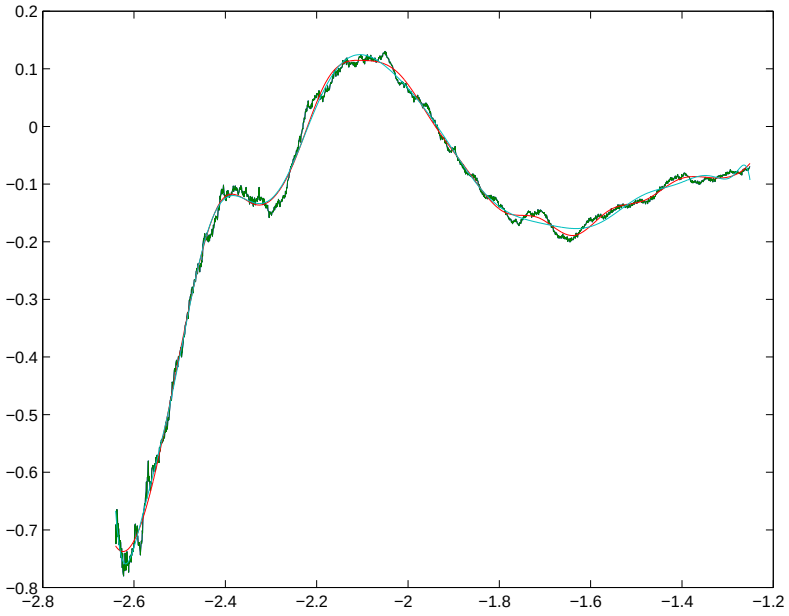

Figure 5.10: Least-squares cubic spline ($n = 4$) with 15 interior knots and the degree 18 least-squares polynomial to data.

optimisation software. DASL and the NPLFit package have extensive facilities for spline approximation [8, 174]. NPLFit, in particular is aimed at metrological applications and has easy-to-use facilities for calculating spline fits, knot choice, and associated uncertainties. NPLFit is available as a package for downloading from EUROMETROS [9, 87].

5.3 Fourier series

5.3.1 Description

A *Fourier series* of degree n is generally written as

$$\phi(x, \mathbf{a}) = a_0 + \sum_{j=1}^{n} a_j \cos jx + \sum_{j=1}^{n} b_j \sin jx,$$

where $\mathbf{a} = (a_0, a_1, \ldots, a_n, b_1, \ldots, b_n)^{\mathrm{T}}$. We note that $\phi(x + 2\pi, \mathbf{a}) = \phi(x, \mathbf{a})$. To model functions with period $2L$, we modify the above to

$$\phi(x, \mathbf{a}|L) = a_0 + \sum_{j=1}^{n} a_j \cos j\pi x/L + \sum_{j=1}^{n} b_j \sin j\pi x/L.$$

Since

$$\int_{-\pi}^{\pi} \cos jx \cos kx \, dx = \int_{-\pi}^{\pi} \sin jx \sin kx \, dx = 0, \quad j \neq k,$$

and

$$\int_{-\pi}^{\pi} \cos jx \sin kx \, dx = \int_{-\pi}^{\pi} \cos jx \, dx = \int_{-\pi}^{\pi} \sin jx \, dx = 0,$$

the basis functions 1, $\cos jx$ and $\sin jx$ are orthogonal with respect to the unit weighting function over any interval of length 2π.

If $f(x)$ is a periodic function with $f(x + 2\pi) = f(x)$ then its representation as a Fourier series is given by

$$f(x) = a_0 + \sum_{j=1}^{\infty} (a_j \cos jx + b_j \sin jx),$$

where

$$a_0 = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) \, dx,$$

and

$$a_j = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos jx \, dx, \quad b_j = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin jx \, dx, j = 1, 2, \ldots$$

Fourier series are used to model periodic functions and to analyse the frequency component or spectral characteristics of data. The Fourier transform and its inverse are important in signal processing and filtering. Fourier series are less successful in analysing data arising from responses $y(x)$ where the frequency component of y changes with location x (see section 5.6).

5.3.2 Working with Fourier series

For fixed period L, $\phi(x, \mathbf{a})$ is a linear model and fitting a Fourier series to data follows the same general scheme for fitting linear models to data $\{(x_i, y_i)\}_{i=1}^{m}$:

I Fix period L and degree n with $2n+1 \leq m$.

II Form $m \times (2n+1)$ observation matrix C. For $i = 1, \ldots, m$, set $C(i,1) = 1$, and for $j = 1, \ldots, n$, $C(i, 2j) = \cos(2\pi j/L)$ and $C(i, 2j+1) = \sin(2\pi j/L)$.

III Solve the linear least-squares system

$$\min_{\mathbf{a}} \|\mathbf{y} - C\mathbf{a}\|^2,$$

for parameters $\mathbf{a}$.

Uncertainties associated with the fitted parameters can be estimated using the general approach described in section 4.1.

It has been assumed that the period L is known. If this is not the case then we can regard L as an unknown, in which case the observation matrix $C = C(L)$ is now a nonlinear function of L[1] and the fitting problem becomes

$$\min_{\mathbf{a},L} \|\mathbf{y} - C(L)\mathbf{a}\|^2,$$

a nonlinear least-squares problem (section 4.2). This problem can be solved using the Gauss-Newton algorithm for example. Alternatively, let $\mathbf{a}(L)$ solve the linear least-squares problem

$$\min_{\mathbf{a}} \|\mathbf{y} - C(L)\mathbf{a}\|^2,$$

and set $\mathbf{r}(L) = \mathbf{y} - C(L)\mathbf{a}(L)$ and $F(L) = \|\mathbf{r}(L)\|$, the norm of the residuals for period L. A univariate minimisation algorithm can be applied to $F(L)$ to find an optimal or at least better estimate of the period.

5.3.3 Fast Fourier Transform (FFT)

For data $(x_j, y_j)_{j=1}^m$ where the abscissae $\{x_j\}$ are uniformly spaced in an interval of length one period, e.g.,

$$x_j = j2L/m,$$

the coefficients $\mathbf{a} = (a_0, a_1, \ldots, a_n, b_1, \ldots, b_n)^{\mathrm{T}}$ for the best-fit Fourier series can be calculated using the discrete Fourier transform (DFT). For any integer $m > 0$ the explicit discrete Fourier transform matrix F is the complex valued matrix defined by

$$F_{jk} = \exp\{-2\pi i(j-1)(k-1)/m\},$$

where $i = \sqrt{-1}$. Its inverse is given by

$$F^{-1}_{jk} = \frac{1}{m}\exp\{2\pi i(j-1)(k-1)/m\}.$$

The DFT of an m-vector $\mathbf{y}$ is simply $\mathbf{w} = F\mathbf{y}$. Since F is complex valued, $\mathbf{w}$ is also. The coefficients a_0, $\mathbf{a}$ and $\mathbf{b}$ of the degree n Fourier series approximation to $\mathbf{y}$ is found from $\mathbf{w}$ as follows

$$a_0 = w_1/m, \quad a_j = 2\Re(w_j)/m, \quad b_j = 2\Im(w_j)/m, \quad j = 1, \ldots, n,$$

[1] Or we could work with K=1/L instead.

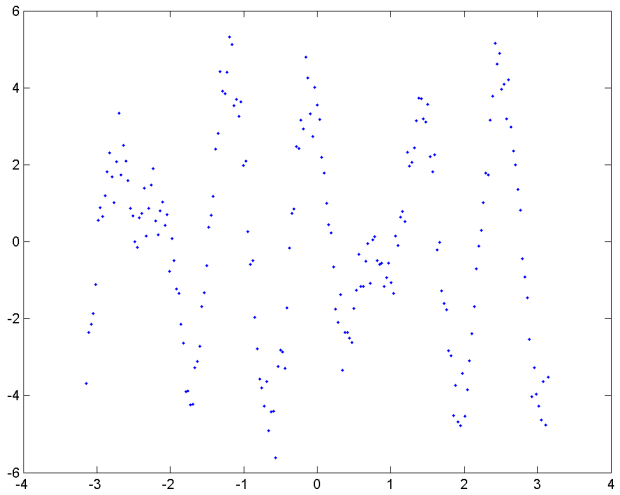

Figure 5.11: Data generated according to the model (5.7).

where $\Re(w_j)$ and $\Im(w_j)$ are the real and imaginary parts of the jth element of $\mathbf{w}$, respectively. The fitted values $\hat{\mathbf{y}}$ can be determined using the inverse DFT:

$$\hat{\mathbf{y}} = \Re\left(F^{-1} \begin{bmatrix} \mathbf{w}(1:n+1) \\ \mathbf{0} \\ \mathbf{w}(m-n+1:m) \end{bmatrix} \right).$$

Instead of working with the explicit transform matrices, the fast Fourier transform uses matrix factorisation techniques to recursively divide the calculations into smaller subproblems and attains a computational efficiency of $O(m \log m)$ rather than $O(m^2)$.

Example: fitting data generated from three Fourier components

Figure 5.11 plots data generated according to the model

$$y_j = 3\cos 5x - 2\sin 7x + 0.5\cos 9x + \epsilon_j, \qquad \boldsymbol{\epsilon} \in N(\mathbf{0}, 0.25I). \tag{5.7}$$

For this data $L = \pi = 3.1416$. Figure 5.12 graphs best-fit Fourier series of degree $n = 10$ with the estimate $\hat{L} = 3.1569$ of L found by a univariate minimisation algorithm. ♯

5.3.4 Bibliography and software sources

Fourier series and transforms are discussed in [33, 34, 80, 142, 164], for example. The fast Fourier transform was developed by Cooley and Tukey [51]. Further developments include [109], for example.

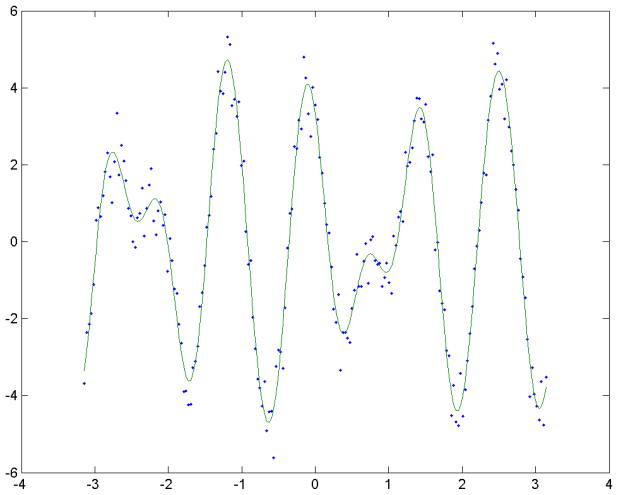

Figure 5.12: Fitted Fourier series of degree $n = 10$ to data in figure 5.11.

5.4 Asymptotic polynomials

Asymptotic behaviour associated with physical systems is quite common. For example, a response may decay to a constant as time passes. However empirical models such as polynomials, splines and Fourier series do not lend themselves to modelling asymptotic behaviour. In this section we describe a simple class of modified polynomial basis functions that can be used to model a range of asymptotic behaviour.

5.4.1 Description

Let $\{\phi_j(x)\}_{j=0}^n$ be a set of polynomial basis functions defined on $[-1, 1]$, such as Chebyshev polynomials (section 5.1). Define

$$w(x) = w(x|x_0, c, k) = \frac{1}{(1 + c^2(x - x_0)^2)^{k/2}}, \quad c > 0.$$

$w(x)$ is smooth and, for c large, $w(x)$ behaves like $|x|^{-k}$ as $|x| \to \infty$.
Defining

$$\tilde{\phi}_j = w(x)\phi_j(x),$$

then

$$\tilde{\phi}(x, \mathbf{a}) = \sum_{j=0}^{n} a_j \tilde{\phi}_j(\mathbf{x})$$

behaves like x^{n-k} as $|x| \to \infty$ and c gets large. In particular, if $k = n$, then ϕ can model asymptotic behaviour of approaching a constant. The constant c controls the degree to which asymptotic behaviour is imposed on the model.

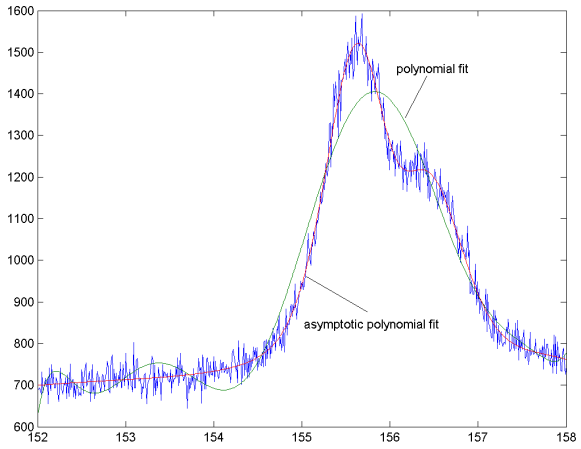

Figure 5.13: Asymptotic and standard polynomial fits of degree 9 to measurements of material properties (for aluminium).

The weighting function w can be modified to provide different asymptotic behaviour as x approaches ∞ and $-\infty$:

$$\begin{aligned} w(x) = w(x|x_0, c, k, l) &= \frac{1}{(1 + c^2(x - x_0)^2)^{k/2}}, \quad x \geq x_0, \\ &= \frac{1}{(1 + c^2(x - x_0)^2)^{l/2}}, \quad x < x_0. \end{aligned}$$

5.4.2 Working with asymptotic polynomials

With x_0 and c fixed, the function $\tilde{\phi}$ is a linear combination of basis functions and so the general approach to model fitting can be adopted:

I Fix x_0, c, k and degree n.

II Form $m \times (n + 1)$ observation matrix C for $\{\phi_j\}$: for $i = 1, \ldots, m$ and $j = 1, \ldots, n$, $C(i, j) = \phi_j(x_i)$ and weight vector $w_i = w(x_i|x_0, c, k)$. Normalise weight vector $w_i := w_i/M$ where $M = \max_i |w_i|$.

III Form modified observation matrix $\tilde{C}_{ij} = w_i C_{ij}$.

IV Solve the linear least-squares system

$$\min_{\mathbf{a}} \|\mathbf{y} - \tilde{C}\mathbf{a}\|^2$$

for parameters $\mathbf{a}$.

Uncertainties associated with the fitted parameters can be estimated using the general approach described in section 4.1. Using the Forsythe method [106], the modified basis

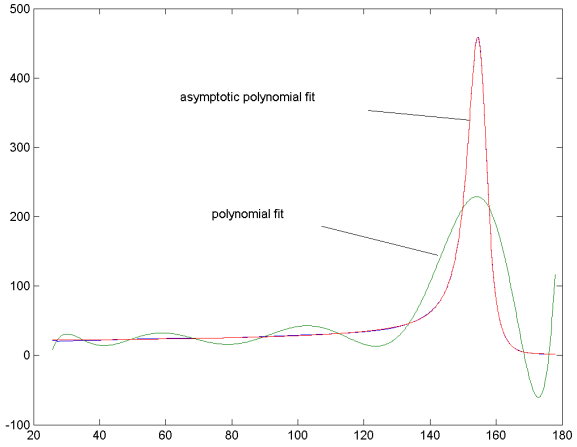

Figure 5.14: Asymptotic and standard polynomial fits of degree 9
to measurements of thermo-physical properties.

functions $\tilde{\phi}_j$ can be determined so that the observation matrix $\tilde{C}$ is orthogonal, leading to better numerical properties.

It has been assumed above that constants x_0 and c are fixed. However, we can regard one or both as additional parameters to be determined in which case the observation matrix $\tilde{C} = \tilde{C}(x_0, c)$ is now a nonlinear function of x_0 and c and the fitting problem becomes

$$\min_{\mathbf{a},x0,c} \|\mathbf{y} - \tilde{C}(x_0, c)\mathbf{a}\|^2,$$

a nonlinear least-squares problem (section 4.2). This problem can be solved using the Gauss-Newton algorithm for example. Note that at each iteration only $\tilde{C}$ has to be formed from C; there is no need to recalculate C.

Alternatively, let $\mathbf{a}(x_0, c)$ solve the linear least-squares problem

$$\min_{\mathbf{a}} \|\mathbf{y} - \tilde{C}(x_0, c)\mathbf{a}\|^2,$$

and set $\mathbf{r}(x_0, c) = \mathbf{y} - \tilde{C}(x_0, c)\mathbf{a}(x_0, c)$ and $F(x_0, c) = \|\mathbf{r}(x_0, c)\|$, the norm of the residuals. A multivariate minimisation algorithm can be applied to $F(x_0, c)$ to find an optimal or at least better estimate of these parameters.

Example: asymptotic polynomial and (standard) polynomial fits compared

In figures 5.13–5.16, asymptotic polynomial and standard polynomial fits of the same degree have been fitted to data portraying asymptotic behaviour. In each case, the asymptotic polynomial fit gives a better representation of the data. In figures 5.14 and 5.16 the asymptotic polynomial fit is barely distinguishable from the data. ♯

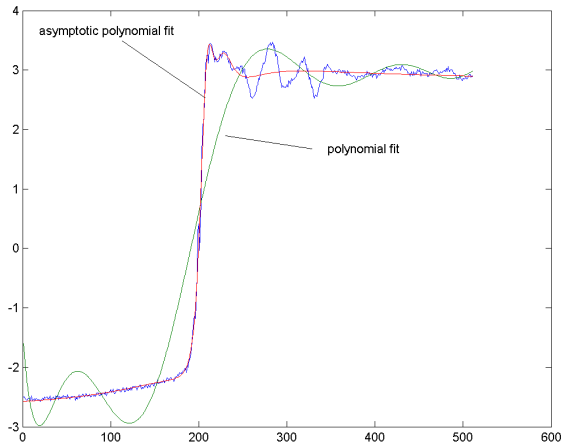

Figure 5.15: Asymptotic and standard polynomial fits of degree 9 to oscilloscope response measurements.

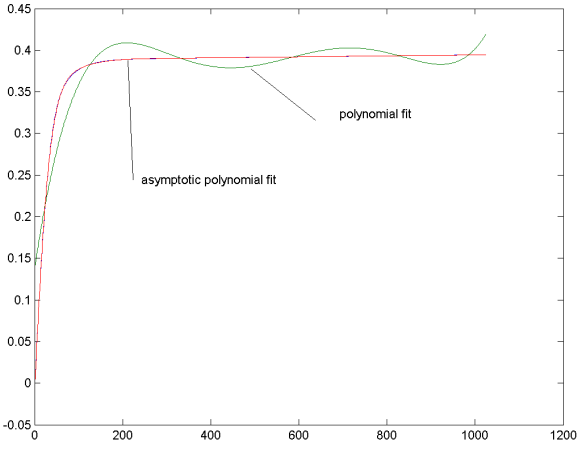

Figure 5.16: Asymptotic and standard polynomial fits of degree 5 to photodiode response measurements.

5.5 Tensor product surfaces

5.5.1 Description

The simplest way to generate linear empirical models for surfaces is to construct them from linear empirical models for curves. Suppose

$$\begin{aligned} \phi(x, \mathbf{a}) &= a_1\phi_1(x) + \ldots + a_{n_x}\phi_{n_x}(x) \quad \text{and} \\ \psi(y, \mathbf{b}) &= b_1\psi_1(y) + \ldots + b_{n_y}\psi_{n_y}(y) \end{aligned}$$

are two linear models for curves. Then the functions $\gamma_{k\ell}(x, y) = \phi_k(x)\psi_\ell(y)$, $k = 1, \ldots, n_x$, $\ell = 1, \ldots, n_y$, form the *tensor product* set of basis functions for defining linear models for representing surfaces of the form

$$z = \gamma(x, y, \mathbf{a}) = \sum_{k=1}^{n_x}\sum_{\ell=1}^{n_y} a_{jk}\gamma_{k\ell}(x, y). \tag{5.8}$$

In particular, tensor products of Chebyshev polynomials and B-spline basis functions are used extensively: see below.

Tensor products are particularly useful representations for data (x_i, y_i, z_i) in which the behaviour of the surface is similar across the domain. They are less efficient in representing generally bland surfaces with local areas of large variations. A second (and related) disadvantage is that the number of basis functions is $n_x \times n_y$, so that to capture variation in both x and y a large number of basis functions can be required. On the positive side, if the data points (x_i, y_i) lie on or near a rectangular grid, the computations can be performed very efficiently [3]: see below.

Tensor product surfaces have been proposed [67] for modelling the kinematic behaviour of coordinate measuring machines (CMMs). An empirical model is used to describe the motion of the probe stylus assembly of the CMM (its location and orientation) in terms of three functions specifying a positional correction and three a rotational correction. Each correction is a function of three independent variables, the scale readings returned by the CMM, and is represented by a tensor product of polynomial spline curves.

Tensor product spline surfaces have also been used in the modelling of a photodiode response [126], in which the independent variables are time and active layer thickness. A spline surface approximation is used to smooth measurements of the response, represent concisely the very large quantities of measurements that are made, and permit effective manipulation of the underlying function including obtaining derivatives and evaluating convolutions.

5.5.2 Working with tensor products

Orthogonality of tensor products

If $\{\phi_k\}$ and $\{\psi_l\}$ are orthonormal[2] with respect to inner products

$$\langle p, q\rangle_u = \int_a^b p(x)q(x)u(x)\,dx, \quad \langle p, q\rangle_v = \int_c^d p(x)q(x)v(x)\,dx,$$

[2]That is, for the appropriate inner product, $\langle p_k, p_l\rangle = 1$ if $k = l$, 0 otherwise.

respectively, then $\{\gamma_{kl}(x,y) = \phi_k(x)\psi_l(y)\}$ are orthonormal with respect to the inner product

$$\langle p, q \rangle_w = \int_a^b \int_c^d p(x,y)q(x,y)w(x,y)\,dy\,dx,$$

where $w(x,y) = u(x)v(y)$.

Data approximation using tensor product surfaces

Given data points $(x_i, y_i, z_i),\ i = 1, \ldots, m$, the least-squares best-fit tensor product surface is found by solving

$$\min_{\mathbf{a}} \quad \sum_{i=1}^{m} (z_i - \gamma(x_i, y_i, \mathbf{a}))^2,$$

with $\gamma(x, y, \mathbf{a})$ defined by (5.8). In matrix terms, we solve

$$\min_{\mathbf{a}} \|\mathbf{z} - \Gamma\mathbf{a}\|^2,$$

where $\mathbf{z} = (z_1, \ldots, z_m)^{\mathrm{T}}$, Γ is an $m \times n_x n_y$ matrix of elements $\gamma_{k\ell}(x_i, y_i)$, and $\mathbf{a}$ is an $n_x n_y \times 1$ vector of elements $a_{k\ell}$. In this formulation, the order of the elements $a_{k\ell}$ in $\mathbf{a}$ (and the order of the corresponding columns of Γ) comes from a choice of ordering of the $n_x n_y$ basis functions $\gamma_{k\ell}(x, y)$.

In the case that the data points relate to measurements on a *grid* in the xy-domain, an alternative linear algebraic formulation is possible that exploits *separability* of the tensor product basis functions and leads to a problem that can be solved significantly faster. Let the data points be $(x_i, y_j, z_{ij}),\ i = 1, \ldots, m_x,\ j = 1, \ldots, m_y$, and let matrices Φ, Ψ, A and Z be defined by

$$\begin{aligned} (\Phi)_{ik} &= \phi_k(x_i), \qquad i = 1, \ldots, m_x, \qquad k = 1, \ldots, n_x, \\ (\Psi)_{j\ell} &= \psi_\ell(y_j), \qquad j = 1, \ldots, m_y, \qquad \ell = 1, \ldots, n_y, \end{aligned}$$

and

$$\begin{aligned} (Z)_{ij} &= z_{ij}, \qquad i = 1, \ldots, m_x, \qquad j = 1, \ldots, m_y, \\ (A)_{k\ell} &= a_{k\ell}, \qquad k = 1, \ldots, n_x, \qquad \ell = 1, \ldots, n_y. \end{aligned}$$

Then, the surface approximation problem is to solve

$$\min_{A} \|Z - \Phi A \Psi^{\mathrm{T}}\|^2, \tag{5.9}$$

the solution to which is given (formally) by

$$\left(\Phi^{\mathrm{T}}\Phi\right) A \left(\Psi^{\mathrm{T}}\Psi\right) = \Phi^{\mathrm{T}} Z \Psi. \tag{5.10}$$

The solution to (5.10) may obtained in two stages: by solving

$$\left(\Phi^{\mathrm{T}}\Phi\right) \tilde{A} = \Phi^{\mathrm{T}} Z$$

for $\tilde{A}$, followed by solving

$$A \left(\Psi^{\mathrm{T}}\Psi\right) = \tilde{A}\Psi$$

for A. These relate, respectively, to least-squares solutions of

$$\min_{\tilde{A}} \|Z - \Phi\tilde{A}\|^2, \tag{5.11}$$

and

$$\min_{A} \|\tilde{A} - A\Psi^{\mathrm{T}}\|^2. \tag{5.12}$$

Consequently, the *surface* approximation problem (5.9) is solved by considering *curve* approximation problems (5.11) and (5.12) as follows. First, for each $j = 1, \ldots, m_y$, find the least-squares best-fit curve

$$f_j(x) = \sum_{k=1}^{n_x} \tilde{a}_{kj}\phi_k(x)$$

to the data $(x_i, z_{ij}),\ i = 1, \ldots, m_x$. Second, for each $i = 1, \ldots, n_x$, find the least-squares best-fit curve

$$f_i(y) = \sum_{\ell=1}^{n_y} a_{i\ell}\psi_\ell(y)$$

to the data $(y_j, \tilde{a}_{ij}),\ j = 1, \ldots, m_y$.

The least-squares best-fit surface is therefore obtained in $O(m_x m_y n_x^2 + m_y n_x n_y^2)$ operations compared with $O(m_x m_y n_x^2 n_y^2)$ that would apply if separability of the basis functions is ignored. For instance, if $m_x = m_y = 1000$ and $n_x = n_y = 100$, the number of operations differ by a factor of $O(10^4)$.

5.5.3 Chebyshev polynomial surfaces

We recall from section 5.1, that a polynomial curve $p_n(x)$ of degree n on the interval $x \in [x_{\min}, x_{\max}]$ has the representation[3]

$$p_n(x) = \frac{1}{2}a_0T_0(\hat{x}) + a_1\mathrm{T}_1(\hat{x}) + \ldots + a_nT_n(\hat{x}) = \sum_{k=0}{}' a_kT_k(\hat{x}),$$

where $\hat{x} \in [-1, +1]$ is related to x by

$$\hat{x} = \frac{(x - x_{\min}) - (x_{\max} - x)}{x_{\max} - x_{\min}}$$

and $T_j(\hat{x}),\ j = 0, \ldots, n$, are Chebyshev polynomials. A tensor product polynomial surface $p_{n_x n_y}(x, y)$ of degree n_x in x and n_y in y on the rectangular domain $(x, y) \in [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$ is therefore represented by

$$p_{n_x n_y}(x, y) = \sum_{k=0}^{n_x}{}' \sum_{\ell=0}^{n_y}{}' a_{k\ell}T_k(\hat{x})T_\ell(\hat{y}), \tag{5.13}$$

where $\hat{x}$ and $\hat{y}$ are each normalised to lie in the interval $[-1, +1]$. We apply, here, the standard convention that coefficients in the above representation which have either k or ℓ zero are written as $a_{k\ell}/2$, and the coefficient with both k and ℓ zero is written as $a_{00}/4$.

[3]The notation $\sum'$ indicates that the first term in the sum is halved.
The normalised variable z, in section 5.1, has been replace by $\hat{x}$.

The polynomial surface (5.13) has *total degree* $n_x + n_y$, the highest combined power of x and y of a basis function. Another way of representing a polynomial surface is to require that the *total degree* of the tensor product basis functions is specified as n. Such a polynomial surface has the representation

$$p_n(x, y) = \sum_{k=0,\ell=0}^{k+\ell \le n} a_{k\ell} T_k(\hat{x}) T_\ell(\hat{y}).$$

Advantages

- For data on regular grids, the solution algorithms are efficient and, with the use of orthogonal basis functions, numerically stable.
- Given polynomial approximation software components for one dimension (evaluation of Chebyshev basis functions, etc.) the implementation of algorithms for approximation with tensor product polynomials is straightforward, especially for data on regular grids.
- For data representing similar qualitative behaviour over the domain of interest, it is usually possible to determine good approximations.
- The order of the polynomials can be used to generate nested sequences of spaces from which to approximate the data.

Disadvantages

- For data representing different types of behaviour in different regions, a tensor product representation can be inefficient.
- For scattered data there is no easily tested criterion to determine *a priori* whether or not approximation with a particular order of polynomial will be well-posed.

5.5.4 Spline surfaces

Recalling section 5.2, a tensor product spline surface $s(x, y)$ of order n_x in x with knots $\boldsymbol{\lambda}$ and order n_y in y with knots $\boldsymbol{\mu}$ on the rectangular domain $(x, y) \in [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$ is represented by

$$s(x, y) = s(x, y, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \sum_{k=1}^{n_x+N_x} \sum_{\ell=1}^{n_y+N_y} c_{k\ell} N_{n_x,k}(x, \boldsymbol{\lambda}) N_{n_y,\ell}(y, \boldsymbol{\mu}), \tag{5.14}$$

where the knot vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ satisfy, respectively,

$$x_{\min} = \lambda_0 < \lambda_1 \le \lambda_2 \le \ldots \le \lambda_{N_x-1} \le \lambda_{N_x} < \lambda_{N_x+1} = x_{\max}$$

and

$$y_{\min} = \mu_0 < \mu_1 \le \mu_2 \le \ldots \le \mu_{N_y-1} \le \mu_{N_y} < \mu_{N_y+1} = y_{\max}.$$

The spline surface (5.14) is a piecewise bivariate polynomial of order n_x in x and n_y in y on $(\lambda_i, \lambda_{i+1}) \times (\mu_j, \mu_{j+1})$, $\quad i = 0, \ldots, N_x, \quad j = 0, \ldots, N_y$. The spline is $(n_x - k - 1)$-times

continuously differentiable along the knot-line $x = \lambda_i$ if $\#(\lambda_\ell = \lambda_i, \ell \in \{1, \ldots, N_x\}) = k$ (and similarly for the knot-line $y = \mu_j$). So, for example, a spline surface of order four in x and y for which the λ_i and μ_j are distinct is a piecewise bicubic polynomial, that is twice continuously differentiable along the lines $x = \lambda_i$ and $y = \mu_j$.

Advantages

- For data on regular grids, the solution algorithms are extremely efficient and numerically stable. For scattered data, it is still possible to exploit sparsity structure in the observation matrix but the gain in efficiency is much less than that for the case of one dimension.
- Given spline approximation software components for one dimension (evaluation of B-spline basis functions, etc.) the implementation of algorithms for approximation with tensor product polynomials is straightforward for data on regular grids.
- For data representing similar qualitative behaviour over the domain of interest, it is usually possible to determine good approximations.
- The knot vectors can be chosen to generate nested sequence of spaces from which to approximate the data.
- For data on a rectangular grid, it is easy to check *a priori* whether a particular choice of knots will lead to a well-posed approximation problem.

Disadvantages

- Splines require the knot vectors to be chosen, for the problems to be linear. If the data or surface exhibits different behaviour in different regions, the choice of knots can affect significantly the quality of the spline representation [73].
- For data representing different types of behaviour in different regions, a tensor product representation can be inefficient.
- For scattered data, there is no easily tested criterion to determine *a priori* whether or not approximation with splines defined by a pair of knot sets will be well posed.

5.6 Wavelets

5.6.1 Description

Wavelets are now an important tool in data analysis and a survey of their application to metrology is given in [147].

In one dimension, wavelets are often associated with a multiresolution analysis (MRA). In outline, let $L^2(\mathbb{R})$ be the space of square integrable functions $f : \mathbb{R} \longrightarrow \mathbb{R}$ so that

$$\int_{-\infty}^{\infty} f^2(x)\, dx < \infty.$$

If $f, g \in L^2(\mathbb{R})$ we define

$$\langle f, g \rangle = \int_{-\infty}^{\infty} f(x)g(x)\,dx,$$

and $\|f\|^2 = \langle f, f \rangle$. This inner-product is used to define orthogonality for functions in $L^2(\mathbb{R})$.

A starting point for MRA is a function $\psi(x)$, the *mother wavelet*. From ψ we define a double sequence of functions

$$\psi_{j,k} = \frac{1}{2^{j/2}}\psi(2^{-j}x - k),$$

using translations and dilations. The mother wavelet is chosen so that $\{\psi_{j,k}\}$ forms an orthonormal basis for $L^2(\mathbb{R})$. Any $f \in L^2(\mathbb{R})$ can be expressed as

$$f(x) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \langle f, \psi_{j,k} \rangle \psi_{j,k}(x).$$

The functions $\{\psi_{j,k}\}$, $k \in \mathbb{Z}$, form an orthonormal basis for a subspace W_j of $L^2(\mathbb{R})$ and these subspaces are used to define a nested sequence of subspaces

$$\ldots \supset V_{j-1} \supset V_j \supset V_{j+1} \supset \ldots$$

where

$$V_{j-1} = V_j \oplus W_j,$$

i.e., any function $f_{j-1} \in V_{j-1}$ can be uniquely expressed as $f_{j-1} = f_j + g_j$, with $f_j \in V_j$ and $g_j \in W_j$. We regard f_j as a smoother approximation to f_{j-1} (since $f(x) \in V_{j-1}$ if and only if $f(2x) \in V_j$) while g_j represents the difference in detail between f_{j-1} and f_j.

The orthogonality properties mean that computations using wavelets can be made very efficiently. In particular, the discrete wavelet transform is used to decompose a uniformly spaced finite set of discrete data points (j, f_j) into component functions at different frequencies (or scales). A major feature of a wavelet analysis is that (unlike Fourier analysis) it can describe different frequency behaviour at different locations.

Wavelets can also be used to analyse signals in higher dimensions. From the orthonormal wavelet basis for $L^2(\mathbb{R})$,

$$\{(\psi_{j,k}(x), j, k \in \mathbb{Z}\}$$

an orthonormal basis for $L^2(\mathbb{R}^2)$ is obtained by taking the tensor products (section 5.5) of two one-dimensional bases functions

$$\psi_{j_1,k_1,j_2,k_2}(x, y) = \psi_{j_1,k_1}(x)\psi_{j_2,k_2}(y).$$

and these functions can be used for MRA in two dimensions.

Advantages

- Wavelets are able to represent different types of behaviour in different regions.
- For data lying on a regular grid, algorithm implementations are efficient and numerically stable.
- Wavelets provide a nested sequence of spaces from which to approximate the data.

- Wavelets are important tools in filtering and data compression.
- Wavelets do not require the specification of subsidiary parameters (but a choice of mother wavelet is required).
- Many wavelet software packages are available.

Disadvantages

- Most wavelet implementations are concerned with data on a regular grid.
- The relationship between the choice of wavelet and the effectiveness of resulting analysis is not obvious.

5.7 Bivariate polynomials

5.7.1 Description

Tensor product surfaces (section 5.5) are especially computationally effective for approximating data where the xy-coordinates (x_i, y_i) are situated on a regular grid. If the locations of (x_i, y_i) are scattered, the tensor product approach is much less efficient. In the case of one dimension, given a set of data $\{(x_i, y_i)\}_{i=1}^m$, the Forsythe method generates, implicitly, a set of orthogonal polynomials $\phi_j(x)$ such that

$$\langle \phi_j, \phi_k \rangle = \sum_{i=1}^{m} \phi_j(x_i)\phi_k(x_i) = 0, \quad j \neq k.$$

Furthermore if there are at least n distinct x_i, then approximating the data with an order n (degree $n-1$) polynomial is a well-posed problem – the associated observation matrix has full rank. In two (or higher) dimensions conditions to guarantee a well conditioned approximation problem are much more complex. For example, if the data points (x_i, y_i, z_i) are such that (x_i, y_i) lie on a circle then the basis vectors corresponding to the basis functions x^2, y^2, x, y and 1 will be linearly dependent. More generally, if (x_i, y_i) lie on (or near to) an algebraic curve (i.e., one defined as the zeros of a polynomial), then the associated observation matrix will be rank deficient (or poorly conditioned).

In a paper by Huhtanen and Larsen [136], an algorithm is presented for generating bivariate polynomials that are orthogonal with respect to a discrete inner product. It is straightforward to implement and includes provision for the possibility of linear independency amongst the basis vectors. The algorithm also provides a recursive scheme to evaluate the polynomial where the length of the recursion is at most $2k+1$ where k is the degree of the polynomial. We illustrate the use of this algorithm in fitting data generated on the surface

$$z = x^4 - y^4 + xy^3 - x^3y + 2. \tag{5.15}$$

We have generated 101 data points (x_i^*, y_i^*) uniformly distributed around the circle $x^2+y^2 = 1$ and calculated z_i^* according to (5.15) so that (x_i^*, y_i^*, z_i^*) lie exactly on the surface; see figure 5.17. We have then added random perturbations to generate data points (x_i, y_i, z_i):

$$x_i = x_i^* + e_i, \quad y_i = y_i^* + f_i, \quad z_i = z_i^* + g_i, \quad e_i, f_i, g_i \in N(0, \sigma^2).$$

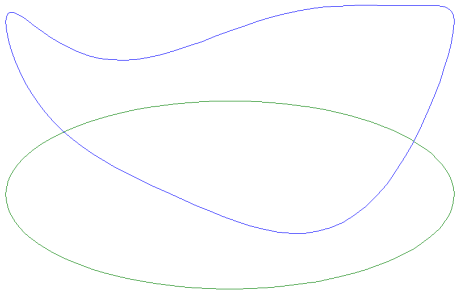

Figure 5.17: Curve defined by the quartic surface (5.15) intersected with the cylinder $x^2 + y^2 = 1$.

There are 15 basis functions associated with a bivariate polynomial of total degree 4. For the data points $\{(x_i^*, y_i^*)\}$ and degree $k = 4$ the algorithm generates 10 orthogonal vectors out of a possible 15, the remaining five being linear combinations of the other basis vectors. The maximum computed element $|(Q^*)^{\mathrm{T}}Q^* - I|$ was 1.5543×10^{-15}. For the data points, $\{(x_i, y_i)\}$, the random perturbations are enough to ensure that the basis functions are linearly independent and the algorithm produces all 15 orthogonal vectors. The maximum computed element of $|Q^{\mathrm{T}}Q - I|$ was 5.0774×10^{-14}.

This algorithm is certainly of interest for those who wish to approximate multivariate data with polynomials and it is likely there will be further developments. Multivariate orthogonal polynomials is an area of considerable research activity (see, e.g., [85]).

Advantages

- The Huhtanen and Larsen (HL) algorithm provides a method of approximating scattered data by bivariate polynomials.
- The algorithm is efficient compared to a full matrix approach and has favourable numerical properties.
- The algorithm copes with possible rank deficiency in the basis functions.
- The HL algorithm is reasonably straightforward to implement.
- The same approach can be applied in higher dimensions.
- The total order of the polynomial can be chosen to generate a nested sequence of spaces from which to chose an approximant.

Disadvantages

- Standard numerical tools for its implementation are not yet widely available.

5.7.2 Bibliography

Multivariate polynomials are discussed in [85, 136], for example.

5.8 RBFs: radial basis functions

5.8.1 Description

Let $\Lambda = \{\boldsymbol{\lambda}_j\}$, $j = 1, \ldots, n$, be a set of points in $\mathbb{R}^p$, and $\rho : \mathbb{R} \longrightarrow [0, \infty)$ a fixed function. A *radial basis function* (RBF) with centres Λ has the form

$$\phi(\mathbf{x}, \mathbf{a}) = \phi(\mathbf{x}, \mathbf{a}, \Lambda) = \sum_{j=1}^{m} a_j \rho(\|\mathbf{x} - \boldsymbol{\lambda}_j\|),$$

where $\|\mathbf{x}\| = (\mathbf{x}^{\mathrm{T}}\mathbf{x})^{1/2}$ is the Euclidean norm of a vector. Defining

$$\phi_j(\mathbf{x}) = \rho(\|\mathbf{x} - \boldsymbol{\lambda}_j\|),$$

then ϕ is seen to be a linear combination of basis functions. Therefore, approximation with RBFs follows the same general approach as with other empirical models defined in terms of basis functions. Given a set of data points $X = \{(\mathbf{x}_i, y_i) \in \mathbb{R}^p \times \mathbb{R}\}$, $i = 1, \ldots, m$, the associated observation matrix has

$$C_{ij} = \rho(\|\mathbf{x}_i - \boldsymbol{\lambda}_j\|).$$

In least-squares approximation, estimates of the parameters $\mathbf{a}$ are found by solving

$$\min_{\mathbf{a}} \|\mathbf{y} - C\mathbf{a}\|^2.$$

Common choices for the function ρ are i) $\rho(r) = r^3$, *cubic*, ii) $\rho(r) = e^{-r^2}$, *Gaussian*, iii) $\rho(r) = r^2 \log r$, *thin plate spline*, iv) $\rho(r) = (r^2 + \lambda^2)^{1/2}$, *multiquadric*, and v) $\rho(r) = (r^2 + \lambda^2)^{-1/2}$, *inverse multiquadric*. In practice, a scaling parameter μ_0 is required so that the RBF has the form

$$\phi(\mathbf{x}, \mathbf{a}|\mu_0, \Lambda) = \sum_{j=1}^{m} a_j \rho(\mu_0 \|\mathbf{x} - \boldsymbol{\lambda}_j\|).$$

If necessary, μ_0 can be regarded as a parameter to be determined as part of the fitting process, in which case the observation matrix $C = C(\mu_0)$ is now a nonlinear function of μ_0 and the optimisation problem becomes

$$\min_{\mathbf{a}, \mu_0} \|\mathbf{y} - C(\mu_0)\mathbf{a}\|^2,$$

a nonlinear least-squares problem (section 4.2). This problem can be solved using the Gauss-Newton algorithm for example. Alternatively, let $\mathbf{a}(\mu_0)$ solve the linear least-squares problem

$$\min_{\mathbf{a}} \|\mathbf{y} - C(\mu_0)\mathbf{a}\|^2,$$

and set $\mathbf{r}(\mu_0) = \mathbf{y} - C(\mu_0)\mathbf{a}(\mu_0)$ and $F(\mu_0) = \|\mathbf{r}(\mu_0)\|$, the norm of the residuals scaling parameter μ_0. A univariate minimisation algorithm can be applied to $F(\mu_0)$ to find an optimal estimate.

Advantages

- RBFs apply to scattered data.
- RBFs apply to multivariate data in any dimension. The computational cost is $O(mn(n+p)$, where m is the number of data points, n the number of centres and p the dimension.
- RBFs can represent different types of behaviour in different regions.
- It is generally possible to chose centres so that the data approximation problem is well-posed, i.e., there is no rank deficiency.
- RBF algorithms are easy to implement, involving only elementary operations and standard numerical linear algebra.
- By choosing the set of centres Λ appropriately, it is possible to generate a nested sequence of spaces from which to choose an approximant.

Disadvantages

- RBF basis functions have no natural orthogonality and can often lead to poorly conditioned observation matrices.
- RBFs give rise to full observation matrices with no obvious way of increasing computational efficiency.
- RBFs require the choice of subsidiary parameters, i.e., the centres and scaling parameter(s).

5.9 Neural networks

5.9.1 Description

Neural networks (NNs), see, e.g., [25, 26, 128], represent a broad class of empirical multivariate models. We present here two common types of network.

Multilayer perceptron

In a multilayer perceptron (MLP) [128, 161], a vector of inputs $\mathbf{x}$ is transformed to a vector of outputs $\mathbf{z}$ through a sequence of matrix-vector operations combined with the application of nonlinear *activation functions*. Often a network has three layers of nodes – input, hidden and output – and two transformations $\mathbb{R}^m \longrightarrow \mathbb{R}^l \longrightarrow \mathbb{R}^n$, $\mathbf{x} \longrightarrow \mathbf{y} \longrightarrow \mathbf{z}$ with

$$y_j = \psi(\mathbf{a}_j^{\mathrm{T}}\mathbf{x} + b_j), \quad z_k = \phi(\mathbf{c}_k^{\mathrm{T}}\mathbf{y} + d_k),$$

or, in matrix terms,

$$\mathbf{y} = \boldsymbol{\psi}(A\mathbf{x} + \mathbf{b}), \quad \mathbf{z} = \boldsymbol{\phi}(C\mathbf{y} + \mathbf{d}) = M(\mathbf{x}, A, \mathbf{b}, C, \mathbf{d}),$$

where A is an $l \times m$ matrix, C an $n \times l$ matrix, and $\mathbf{b}$ and $\mathbf{d}$ are l- and n-vectors, respectively. The activation function is often chosen to be the logistic sigmoid function $1/(1+e^{-x})$ or a hyperbolic tangent function $\tanh(x)$. These functions have unit gradient at zero and approach 1 as $x \to \infty$ and 0 or -1 as $x \to -\infty$. For classification problems, the network is designed to work as follows. The value of y_j indicates whether a feature specified by $\mathbf{a}_j$ is present ($y_j \approx 1$) or absent ($y_j \approx 0$ or -1) in the input $\mathbf{x}$. The output $\mathbf{z}$ completes the classification of the input according to the features identified in the hidden layer $\mathbf{y}$: the input is assigned to the qth class if $z_q \approx 1$ and $z_r \approx 0$ or -1, $r \neq q$. For empirical modelling, the second activation function is usually chosen to be the identity function $\phi(x) = x$, so that all values of output are possible, and

$$\mathbf{z} = M(\mathbf{x}, A, \mathbf{b}, C, \mathbf{d}) = C\boldsymbol{\psi}(A\mathbf{x} + \mathbf{b}) + \mathbf{d}, \tag{5.16}$$

a flexible multivariate function $M : \mathbb{R}^m \longrightarrow \mathbb{R}^n$.

Given training data comprising sets of inputs and required outputs $\{(\mathbf{x}_q, \mathbf{z}_q)\}$, an iterative optimisation process – the back-propagation algorithm – can be used to adjust the weighting matrices A and C and bias vectors $\mathbf{b}$ and $\mathbf{d}$ so that $M(\mathbf{x}_q, A, \mathbf{b}, C, \mathbf{d}) \approx \mathbf{z}_q$. Alternatively, standard large-scale optimisation techniques [65, 115, 118, 212] such as conjugate gradient methods can be employed. However, the optimisation problems are likely to be poorly conditioned or rank deficient and the optimisation algorithms need to cope with this possibility. Many algorithms therefore employ large-scale techniques combined with regularisation techniques [123, 124, 202].

MLP models are extremely flexible. Many of the problems associated with implementing them for a particular application are in deciding how to reduce the flexibility in order to produce a compact model while at the same time retaining enough flexibility in order to represent adequately the system being modelled.

RBF networks

Radial basis function (RBF) networks [35, 177, 178] have a similar design to multilayer perceptrons (MLPs) but the activation function is a radial basis function. Typically, we have

$$y_j = \rho_j(\|\mathbf{x} - \boldsymbol{\lambda}_j\|), \qquad \mathbf{z} = C\mathbf{y} + \mathbf{d},$$

where ρ_j is a Gaussian function, $\rho_j(x) = \exp\{-x^2/(2\sigma_j^2)\}$, for example. More generally, we can have

$$y_j = \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\lambda})^{\mathrm{T}} M_j (\mathbf{x} - \boldsymbol{\lambda})\right\},$$

where M_j is a symmetric, semi-positive definite matrix.

Advantages

- NNs can be used to approximate any continuous function $\mathbf{f} : \mathbb{R}^m \longrightarrow \mathbb{R}^n$ [110, 133].
- NNs can be used to perform nonlinear classification, in which data points belonging to different classes are separated by nonlinear hyper-surfaces.
- NN models are straightforward to evaluate and back-propagation algorithms, for example, are easy to implement.

Disadvantages

- The determination of optimal weights and biases is a nonlinear optimisation problem.
- The back-propagation algorithm can converge slowly to one of possibly many local minima.
- The behaviour of the model on training data can be a poor guide to its behaviour on similar data.
- The evaluation of the uncertainty associated with the fitted parameters is difficult.
- The effectiveness of the network can depend critically on its design (number and size of hidden layers).

5.10 Geometric elements

In this section we consider a class of models that have characteristics in many ways different from empirical models such as polynomials and splines. The most common geometric elements are lines in two and three dimensions, planes, circles in two and three dimensions, spheres, cylinders and cones. Less common but important in some fields are ellipses and ellipsoids, tori, aspherical surfaces and surfaces of revolution; see also section 5.11. Geometric elements generally can be defined in terms of two sets of parameters $\mathbf{a} = (\mathbf{s}^{\mathrm{T}}, \mathbf{t}^{\mathrm{T}})^{\mathrm{T}}$, those $\mathbf{s}$ defining their size and shape – *shape parameters* – and those $\mathbf{t}$ defining their location and orientation - *position parameters*. For example, a circle in the plane can be specified by one shape parameter describing its radius and two position parameters describing the location of its centre. In other parameterisations, there may be no such clear distinction.

Geometric elements are important in dimensional metrology, particularly co-ordinate metrology and in manufacturing and other engineering disciplines. They are used to represent the shape of manufactured parts and engineering components. They arise in many systems for which a geometrical description is required.

5.10.1 Working with geometrical elements

Most calculations with geometric elements involve the calculation of the distance $d(\mathbf{x}, \mathbf{a})$ from a data point $\mathbf{x}$ (in two or three dimensions, depending on the element) to the profile or surface of the element in terms of its shape and position parameters $\mathbf{a}$. For example the least squares best-fit element to data $X = \{\mathbf{x}_i\}_1^m$ is found by solving

$$\min_{\mathbf{a}} \sum_{i=1}^{m} d^2(\mathbf{x}_i, \mathbf{a}). \tag{5.17}$$

This type of regression is known as *orthogonal regression* since the error of fit at $\mathbf{x}_i$ is taken to be the smallest distance to the curve or surface rather that the distance calculated in a specific direction (such as parallel to the z-axis). This type of estimation is considered in section 4.3. The use of orthogonal regression is justified on the basis of maximum likelihood principles and/or on the basis of rotational invariance, since the properties of an artefact's

shape determined from measurements should not be dependent on the orientation in which the artefact is measured, with respect to the co-ordinate system used.

Example: least-squares orthogonal regression with circles, implicit version

We model a circle implicitly as $f(\mathbf{x}, \mathbf{a}) = (x - a_1)^2 + (y - a_2)^2 - a_3^2 = 0$. Suppose the data points $\mathbf{x}_i = (x_i, y_i)^{\mathrm{T}}$ are generated by a co-ordinate measuring system with random effects modelled as

$$\mathbf{x}_i = \mathbf{x}_i^* + \boldsymbol{\epsilon}_i,$$

where $\mathbf{x}_i^* = (x_i^*, y_i^*)^{\mathrm{T}}$ is the data point lying on the circle $f(\mathbf{x}, \mathbf{a}) = 0$ and $\boldsymbol{\epsilon}_i$ represents a random effect. It is assumed that the components of $\boldsymbol{\epsilon}_i = (\epsilon_i, \delta_i)^{\mathrm{T}}$ are uncorrelated and drawn from a normal distribution $N(0, \sigma^2)$. The maximum likelihood estimate of the circle parameters $\mathbf{a}$ is found by minimising

$$\min_{\mathbf{a}, \{\boldsymbol{\epsilon}_i\}} \sum_{i=1}^{m} (\epsilon_i^2 + \delta_i^2) \quad = \quad \sum_{i=1}^{m} (x_i - x_i^*)^2 + (y_i - y_i^*)^2$$

subject to the constraints $f(\mathbf{x}_i^*, \mathbf{a}) = 0$. Given any $\mathbf{a}$, this sum is minimised by setting $\mathbf{x}_i^*$ equal to the point on the circle $f(\mathbf{x}, \mathbf{a}) = 0$ nearest $\mathbf{x}_i$:

$$\begin{aligned} x_i^* &= a_1 + a_3 \frac{x_i - a_1}{r_i}, \\ y_i^* &= a_2 + a_3 \frac{y_i - a_2}{r_i}, \quad \text{where} \\ r_i &= \{(x_i - a_1)^2 + (y_i - a_2)^2\}^{1/2}. \end{aligned}$$

For this $\mathbf{x}_i^*$,

$$\{(x_i - x_i^*)^2 + (y_i - y_i^*)^2\}^{1/2} = d(\mathbf{x}_i, \mathbf{a}) = r_i - a_3,$$

and the optimisation problem reduces to (5.17). ♯

Example: least-squares orthogonal regression with circles, parametric version

Alternatively, we model a circle parametrically as

$$x^* = a_1 + a_3 \cos u, \quad y_i^* = a_2 + a_3 \sin u.$$

The maximum likelihood estimation problem can then be posed as

$$\min_{\mathbf{a}, \{u_i\}} \sum_{i=1}^{m} (\epsilon_i^2 + \delta_i^2) \;=\; \sum_{i=1}^{m} (x_i - a_1 - a_3 \cos u_i)^2 + (y_i - a_2 - a_3 \sin u_i)^2.$$

Given any $\mathbf{a}$, this sum is minimised by setting u_i according to

$$\begin{aligned} \cos u_i &= \frac{x_i - a_1}{r_i}, \\ \sin u_i &= \frac{y_i - a_2}{r_i}, \end{aligned}$$

so that the optimisation problem again reduces to (5.17). ♯

For the simpler geometric elements specified by parameters $\mathbf{a}$, the distance $d(\mathbf{x}, \mathbf{a})$ from a point $\mathbf{x}$ to the element can be calculated as an explicit function of $\mathbf{x}$ and $\mathbf{a}$. For more complicated elements, a numerical approach is required to solve the associated *foot point problems*; see section 4.3.

Rotations and translations. Often the position parameters are defined in terms of rotations and translations. Let

$$R(\boldsymbol{\alpha}) = R_z(\gamma)R_y(\beta)R_x(\alpha)$$

be the composition of three plane rotations defined by

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}, \quad R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

and

$$R_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

A roto-translation can be written in the form

$$\hat{\mathbf{x}} = T(\mathbf{x}, \mathbf{t}) = R(\boldsymbol{\alpha})R_0(\mathbf{x} - \mathbf{x}_0),$$

and is specified by parameters $\mathbf{t} = (\mathbf{x}_0^{\mathrm{T}}, \boldsymbol{\alpha}^{\mathrm{T}})^{\mathrm{T}}$ and fixed rotation R_0. The inverse transformation T^{-1} is

$$\mathbf{x} = \mathbf{x}_0 + R_0^{\mathrm{T}} R^{\mathrm{T}}(\boldsymbol{\alpha})\hat{\mathbf{x}}.$$

Example: orthogonal regression with cylinders I

Suppose we wish to fit a cylinder to data points $\{\mathbf{x}_i\}_1^m$. A cylinder is specified by a point on its axis $\mathbf{x}_0$, an axis direction vector $\mathbf{n}$ and its radius. If the cylinder axis is approximately coincident with the z-axis, we can parameterise the cylinder as follows:

$$\mathbf{x}_0(\mathbf{a}) = \begin{bmatrix} a_1 \\ a_2 \\ 0 \end{bmatrix}, \quad \mathbf{n}(\mathbf{a}) = R_y^{\mathrm{T}}(a_4)R_x^{\mathrm{T}}(a_3)\mathbf{e}_z, \quad \mathbf{e}_z = (0, 0, 1)^{\mathrm{T}},$$

and radius a_5, five parameters in all. This parameterisation becomes less stable and eventually breaks down as the angle the cylinder axis makes with the z-axis approaches a right angle. A family of parameterisations generated from this parameterisation can used to describe cylinders in a general orientation and location. Let $\mathbf{n}_0$ be the approximate axis direction and R_0 a fixed rotation matrix such that $R_0^{\mathrm{T}}\mathbf{n}_0 = \mathbf{e}_z$. Similarly, let $\mathbf{z}_0$ be a point on the nominal axis. Then the cylinder is parameterised in terms of $\mathbf{x}_0(\mathbf{a})$, $\mathbf{n}(\mathbf{a})$ and its radius, where

$$\mathbf{x}_0(\mathbf{a}) = \mathbf{z}_0 + R_0^{\mathrm{T}} \begin{bmatrix} a_1 \\ a_2 \\ 0 \end{bmatrix}, \quad \mathbf{n}(\mathbf{a}) = R_0^{\mathrm{T}} R_y^{\mathrm{T}}(a_4)R_x^{\mathrm{T}}(a_3)\mathbf{e}_z.$$

Members of this family of parameterisations are specified by the extra constants determining the fixed translation vector and rotation matrix. In order to select an appropriate member of the family, an initial indication of the axis is required.

The distance $d(\mathbf{x}, \mathbf{a})$ to a cylinder parameterised in this way is given by

$$d(\mathbf{x}, \mathbf{a}) = \|(\mathbf{x} - \mathbf{x}_0(\mathbf{a})) \times \mathbf{n}(\mathbf{a})\| - a_5, \tag{5.18}$$

where $\mathbf{c} \times \mathbf{d}$ denotes the cross product of vectors. ♯

Example: orthogonal regression with cylinders II

We consider again orthogonal regression with cylinders, using a slightly different approach so that the position and shape parameters are separated. In the first approach described above, we think of moving and shaping the cylinder so that it lies as close as possible to the data. In this second approach we think of moving the data so that it is as close to possible to the cylinder.

A cylinder in *standard position* has its axis coincident with the z-axis. A cylinder has one shape parameter, its radius, and a cylinder in standard position is given by the equation

$$f(\mathbf{x}, s) = f(x, y, z, s) = x^2 + y^2 - s^2 = 0.$$

The distance from a point $\mathbf{x} = (x, y, z)^{\mathrm{T}}$ to a cylinder in standard position is given by $d(\mathbf{x}, s) = (x^2 + y^2)^{1/2} - s$.

Suppose, as before, we wish to fit a cylinder to data points $\{\mathbf{x}_i\}_1^m$. We assume that the data has been transformed by an initial translation and rotated so that the data approximately lies in the surface of the cylinder in standard position. Let T be the roto-translation defined by $\mathbf{t} = (a_1, a_2, a_3, a_4)^{\mathrm{T}}$, where

$$\hat{\mathbf{x}}(\mathbf{t}) = \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} = R_y(a_4) R_x(a_3) \left(\begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} a_1 \\ a_2 \\ 0 \end{bmatrix} \right).$$

The distance from a point $\mathbf{x}$ to the cylinder is given in terms of the position parameters $\mathbf{t}$ and shape parameters s by

$$d(\mathbf{x}, \mathbf{a}) = d(\hat{\mathbf{x}}(\mathbf{t}), s) = (\hat{x}^2 + \hat{y}^2)^{1/2} - s. \tag{5.19}$$

The advantages of this approach are firstly, the calculation of the distance and its derivatives is simpler (compare (5.19) with (5.18)) and, secondly and more importantly, the calculations involving the transformation parameters are separated from the shape parameters and are largely generic, independent of the geometric element. ♯

5.10.2 Bibliography and software sources

Least-squares and Chebyshev regression with geometric elements and related form and tolerance assessment problems are considered in [5, 6, 7, 39, 40, 64, 90, 91, 92, 93, 97, 105, 111, 196, 213]. The package LSGE — least squares geometric elements — is available for download from EUROMETROS [9, 87].

5.11 NURBS: nonuniform rational B-splines

A nonuniform rational B-splines curve of order k is defined as a parametric curve $\boldsymbol{C} : \mathbb{R} \longrightarrow \mathbb{R}^2$ with

$$\boldsymbol{C}(u) = \frac{\sum_{j=0}^{n} N_{k,j}(u|\boldsymbol{\lambda}) w_j \boldsymbol{P}_j}{\sum_{j=0}^{n} N_{k,j}(u|\boldsymbol{\lambda}) w_j},$$

where $\boldsymbol{P}_j \in \mathbb{R}^2$ are the control points, w_j weights and $N_{k,j}(u|\boldsymbol{\lambda})$ B-spline basis functions defined on a knot set $\boldsymbol{\lambda}$ (section 5.2).

NURBS surfaces $\boldsymbol{S} : \mathbb{R}^2 \longrightarrow \mathbb{R}^3$ are generated using tensor products (section 5.5) of B-spline basis functions:

$$\boldsymbol{S}(u,v) = \frac{\sum_{j=0}^{n}\sum_{q=0}^{m} N_{k,j}(u|\boldsymbol{\lambda})N_{l,q}(v|\boldsymbol{\mu})w_{jq}\boldsymbol{P}_{jq}}{\sum_{j=0}^{n}\sum_{q=0}^{m} N_{k,j}(u|\boldsymbol{\lambda})N_{l,q}(v|\boldsymbol{\mu})w_{jq}},$$

where $N_{k,j}(u|\boldsymbol{\lambda})$ and $N_{l,q}(v|\boldsymbol{\mu})$ are the B-spline basis functions, $\boldsymbol{P}_{jq} \in \mathbb{R}^3$ are control points, and w_{jq} weights.

Nonuniform rational B-splines (NURBS) are used for computer graphics and extensively in computer-aid design for defining complex curves and surfaces and are therefore important in co-ordinate metrology.

Advantages

- NURBS can be used to model and modify highly complex curves and surfaces.
- The shape of the curve or surface is easily determined and modified by the location of the control points. NURBS provide local control, so that shifting one control point only affects the surface shape near that control point.
- NURBS are invariant under scaling, translation, shear, and rotation,
- NURBS can be used to define quadric surfaces, such as spheres and ellipsoids, commonly used in CAD exactly. Parametric B-spline surfaces can only approximate such surfaces and in doing so require many more control points.

Disadvantages

Although NURBS are in principle straightforward to implement, efficient and numerically stable approaches require appropriate use of the recurrence formulae associated with B-splines.

- Data approximation with NURBS (fitting a cloud of points with a NURBS curve or surface) is likely to give rise to rank deficient or poorly conditioned problems. However there are a number of ways of approaching approximation with parametric curves and surfaces, some of which give rise to well conditioned problems (see, e.g., [38, 98]).

5.11.1 Bibliography and software sources

Curve and surface representation in computer-aided design is described in [88, 184], for example. A number of software packages for NURBS are available for download including, for example, [197].

Chapter 6

Best practice in discrete modelling and experimental data analysis: a summary

We summarise the main issues that need to be addressed in discrete modelling and in metrological data analysis.

Functional model consists of:

- Problem variables representing all the quantities that are known or measured.
- Problem parameters representing the quantities that have to be determined from the measurement experiment. The problem parameters describe the possible behaviour of the system.
- The functional relationship between the variables and parameters.

Statistical model for the measurements consists of:

- The uncertainty structure describing which variables are known accurately and which are subject to significant random effects.
- The description of how the random effects are expected to behave, usually in terms means, variances (standard deviations) or probability density functions.

Estimator. An estimator is a method of extracting estimates of the problem parameters from the measurement data. Good estimators are unbiased, efficient and consistent.

The behaviour of an estimator can be analysed from maximum likelihood principles or using Monte Carlo simulations.

Estimator algorithm. An estimator requires the solution of a computational problem. An algorithm describes how this can be achieved.

Good algorithms determine an estimate of the solution that is close to the true solution of the computational problem and is efficient in terms of computational speed and memory requirements.

Problem conditioning and numerical stability. The effectiveness of an algorithm will depend on the conditioning the computational problem. For well conditioned problems, a small change in the data corresponds to a small change in the solution parameters, and conversely.

The conditioning of a problem depends on the parameterisation of the model. Often, the key to being able to determine accurate solution parameters is in finding the appropriate parameterisation.

A numerically stable algorithm is one that introduces no unnecessary ill-conditioning in the problem.

Software implementation and reuse. Calculations with a model should be split up into model key functions such as calculating function values and partial derivatives.

Optimisation software in the form of key solver functions can be used in implementing estimators that work with a wide range of model key functions.

For some models, special purpose solvers that exploit special features in the model are useful or necessary.

Many calculations required in discrete modelling can be performed using standard library/public domain software.

EUROMETROS. The Metrology Software environment developed under the Software Support for Metrology Programme aims to bridge the gap between library software and the metrologists needs, promoting and developing re-usable software performing the main calculations required by metrologists.

Bibliography

[1] S. J. Ahn, E. Westkämper, and Rauh. W. Orthogonal distance fitting of parametric curves and surfaces. In J. Levesley, I. J. Anderson, and J. C. Mason, editors, *Algorithms for Approximation IV*, pages 122–129. University of Huddersfield, 2002. 101

[2] AMCTM, www.amctm.org. *Advanced Mathematical and Computation Tools in Metrology*.

[3] I. A. Anderson, M. G. Cox, and J. C. Mason. Tensor-product spline interpolation to data on or near a family of lines. *Numerical Algorithms*, 5:193–204, 1993. 145

[4] I. J. Anderson, M. G. Cox, A. B. Forbes, J. C. Mason, and D. A. Turner. An efficient and robust algorithm for solving the footpoint problem. In M. Daehlen, T. Lyche, and L. L. Schumaker, editors, *Mathematical Methods for Curves and Surfaces II*, pages 9–16, Nashville, TN, 1998. Vanderbilt University Press. 101

[5] G. T. Anthony, H. M. Anthony, B. Bittner, B. P. Butler, M. G. Cox, R. Drieschner, R. Elligsen, A. B. Forbes, H. Groß, S. A. Hannaby, P. M. Harris, and J. Kok. Chebyshev best-fit geometric elements. Technical Report DITC 221/93, National Physical Laboratory, Teddington, 1993. 104, 110, 159

[6] G. T. Anthony, H. M. Anthony, B. Bittner, B. P. Butler, M. G. Cox, R. Drieschner, R. Elligsen, A. B. Forbes, H. Groß, S. A. Hannaby, P. M. Harris, and J. Kok. Reference software for finding Chebyshev best-fit geometric elements. *Precision Engineering*, 19:28 – 36, 1996. 104, 110, 159

[7] G. T. Anthony, H. M. Anthony, M. G. Cox, and A. B. Forbes. The parametrization of fundamental geometric form. Technical Report EUR 13517 EN, Commission of the European Communities (BCR Information), Luxembourg, 1991. 159

[8] G. T. Anthony and M. G. Cox. The National Physical Laboratory's Data Approximation Subroutine Library. In J. C. Mason and M. G. Cox, editors, *Algorithms for Approximation*, pages 669 – 687, Oxford, 1987. Clarendon Press. 5, 85, 124, 128, 137

[9] R. M. Barker. Software Support for Metrology Good Practice Guide No. 5: Guide to EUROMETROS: a manual for users, contributors and testers. Technical report, National Physical Laboratory, Teddington, 2004. `http://www.npl.co.uk/ssfm/download/bpg.html#ssfmgpg5`. 101, 128, 137, 159

[10] R. M. Barker, M. G. Cox, P. M. Harris, and I. M. Smith. Testing algorithms in Standards and MetroS. Technical Report CMSC 18/03, National Physical Laboratory, March 2003. http://www.npl.co.uk/ssfm/download/#cmsc18_03.

[11] R. M. Barker and A. B. Forbes. Software Support for Metrology Best Practice Guide No. 10: Discrete Model Validation. Technical report, National Physical Laboratory, Teddington, March 2001. 3

[12] V. A. Barker, L. S. Blackford, J. L. Dongarra, J. Du Croz, S. Hammarling, M. Marinova, J. Wasniewski, and P. Yalamov. *The LAPACK95 User's Guide*. SIAM, Philadelphia, 2001. 5

[13] I. Barrodale and C. Phillips. Algorithm 495: Solution of an overdetermined system of linear equations in the Chebyshev norm. *ACM Transactions of Mathematical Software*, pages 264 – 270, 1975. 104

[14] I. Barrodale and F. D. K. Roberts. An efficient algorithm for discrete l_1 linear approximation with linear constraints. *SIAM Journal of Numerical Analysis*, 15:603 – 611, 1978. 106

[15] I. Barrodale and F. D. K. Roberts. Solution of the constrained ℓ_1 linear approximation problem. *ACM Trans. Math. Soft.*, 6(2):231 –235, 1980. 106

[16] R. Bartels and A. R Conn. A programm for linearly constrained discrete ℓ_1 problems. *ACM Trans. Math. Soft.*, 6(4):609–614, 1980. 106

[17] R. Bartels, A. R. Conn, and J. W. Sinclair. Minimization techniques for piecewise differentiable functions: The ℓ_1 solution to an overdetermined linear system. *SIAM Journal of Numerical Analysis*, 15:224–241, 1978. 106

[18] R. Bartels and G. H. Golub. Chebyshev solution to an overdetermined linear system. *Comm. ACM*, 11(6):428–430, 1968. 104

[19] M. Bartholomew-Biggs, B. P. Butler, and A. B. Forbes. Optimisation algorithms for generalised regression on metrology. In P. Ciarlini, A. B. Forbes, F. Pavese, and D. Richter, editors, *Advanced Mathematical and Computational Tools in Metrology IV*, pages 21–31, Singapore, 2000. World Scientific. 4, 101

[20] M. C. Bartholomew-Biggs, S. Brown, B. Christianson, and L. Dixon. Automatic differentiation of algorithms. *J. Comp. App. Math.*, 124:171–190, 2000. 95

[21] N. Bellomo and L. Preziosi. Mathematical problems in metrology: modelling and solution methods. In P. Ciarlini, M. G. Cox, R. Monaco, and F. Pavese, editors, *Advanced Mathematical Tools for Metrology,*, pages 23–36. World Scientific, 1994. 4

[22] W. Bich. The ISO guide to the expression of uncertainty in measurement: A bridge between statistics and metrology. In P. Ciarlini, M. G. Cox, F. Pavese, and D. Richter, editors, *Advanced Mathematical Tools in Metrology, III*, pages 1–11, Singapore, 1997. World Scientific. 4

[23] W. Bich and P. Tavella. Calibrations by comparison in metrology: a survey. In P. Ciarlini, M. G. Cox, R. Monaco, and F. Pavese, editors, *Advanced Mathematical Tools in Metrology,*, pages 155–166, Singapore, 1994. World Scientific. 4

[24] BIPM, IEC, IFCC, ISO, IUPAC, IUPAP, and OIML. *Guide to the Expression of Uncertainty in Measurement*. Geneva, second edition, 1995. 18

[25] C. M. Bishop. *Neural networks and pattern recognition*. Oxford Univeristy Press, 1995. 154

[26] C. M. Bishop, editor. *Neural networks and Machine Learning*. Springer, 1998. 1997 NATO Advanced Study Institute. 154

[27] A. Björck. *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia, 1996. 74, 85

[28] P. T. Boggs, R. H. Byrd, and R. B. Schnabel. A stable and efficient algorithm for nonlinear orthogonal distance regression. *SIAM Journal of Scientific and Statistical Computing*, 8(6):1052–1078, 1987. 101

[29] P. T. Boggs, J. R. Donaldson, R. H. Byrd, and R. B. Schnabel. ODRPACK: software for weighted orthogonal distance regression. *ACM Trans. Math. Soft.*, 15(4):348–364, 1989. 101

[30] R. Boudjemaa, M. G. Cox, A. B. Forbes, and P. M. Harris. Automatic differentiation and its applications to metrology. Technical Report CMSC 26/03, National Physical Laboratory, June 2003. 95

[31] R. Boudjemaa and A. B. Forbes. Parameter estimation methods for data fusion. Technical Report CMSC 38/04, National Physical Laboratory, February 2004. 115

[32] G. E. P. Box and G. C. Tiao. *Bayesian inference in statistical analysis*. Wiley, New York, Wiley Classics Library Edition 1992 edition, 1973. 27, 28

[33] R. Bracewell. *The Fourier Transform and Its Applications*. McGraw-Hill, New York, 3rd edition, 1999. 140

[34] E. O. Brigham. *The Fast Fourier Transform and Applications*. Prentice Hall, Englewood Cliffs, NJ, 1988. 140

[35] D. S. Broomhead and D. Lowe. Multivariate functional interpolation and adaptive networks. *Complex Systems*, 2:321–355, 1988. 155

[36] R. G. Brown and P. Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. Wiley, New York, 3rd edition, 1997. 85

[37] B. P. Butler. A framework for model validation and software testing in regression. In P. Ciarlini, M. G. Cox, F. Pavese, and D. Richter, editors, *Advanced Mathematical Tools in Metrology, III*, pages 158–164, Singapore, 1997. World Scientific. 4

[38] B. P. Butler, M. G. Cox, and A. B. Forbes. The reconstruction of workpiece surfaces from probe coordinate data. In R. B. Fisher, editor, *Design and Application of Curves and Surfaces*, pages 99–116. Oxford University Press, 1994. IMA Conference Series. 101, 160

[39] B. P. Butler, A. B. Forbes, and P. M. Harris. Algorithms for geometric tolerance assessment. Technical Report DITC 228/94, National Physical Laboratory, Teddington, 1994. 104, 110, 159

[40] B. P. Butler, A. B. Forbes, and P. M. Harris. Geometric tolerance assessment problems. In P. Ciarlini, M. G. Cox, R. Monaco, and F. Pavese, editors, *Advanced Mathematical Tools in Metrology*, pages 95–104, Singapore, 1994. World Scientific. 104, 159

[41] P. Ciarlini. Bootstrap algorithms and applications. In P. Ciarlini, M. G. Cox, F. Pavese, and D. Richter, editors, *Advanced Mathematical Tools in Metrology, III*, pages 12–23, Singapore, 1997. World Scientific. 4

[42] P. Ciarlini, M. G. Cox, E. Filipe, F. Pavese, and D. Richter, editors. *Advanced Mathematical and Computational Tools in Metrology, V*, Singapore, 2001. World Scientific. 4

[43] P. Ciarlini, M. G. Cox, R. Monaco, and F. Pavese, editors. *Advanced Mathematical Tools in Metrology*, Singapore, 1994. World Scientific. 4

[44] P. Ciarlini, M. G. Cox, F. Pavese, and D. Richter, editors. *Advanced Mathematical Tools in Metrology, II*, Singapore, 1996. World Scientific. 4

[45] P. Ciarlini, M. G. Cox, F. Pavese, and D. Richter, editors. *Advanced Mathematical Tools in Metrology, III*, Singapore, 1997. World Scientific. 4

[46] P. Ciarlini, E. Felipe, A. B. Forbes, F. Pavese, C. Perruchet, and B. Siebert, editors. *Advanced Mathematical and Computational Tools in Metrology VII*. World Scientific, Singapore, 2006. 4

[47] P. Ciarlini, A. B. Forbes, F. Pavese, and D. Richter, editors. *Advanced Mathematical and Computational Tools in Metrology IV*. World Scientific, Singapore, 2000. 4

[48] C. W. Clenshaw. A note on the summation of Chebyshev series. *Math. Tab. Wash.*, 9:118–120, 1955. 125

[49] C. W. Clenshaw. A comparison of"best" polynomial approximations with truncated Chebyshev series expansions. *SIAM J. Num. Anal.*, 1:26–37, 1964. 127

[50] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *LANCELOT: a Fortran package for large-scale nonlinear optimization, release A*. Springer-Verlag, Berlin, 1992. 35

[51] J. W. Cooley and O. W. Tukey. An algorithm for the machine calculation of complex fourier series. *Math. Comput.*, 19:297–301, 1965. 140

[52] M. G. Cox. The numerical evaluation of B-splines. *Journal of the Institute of Mathematics and its Applications*, 8:36–52, 1972. 135

[53] M. G. Cox. Cubic-spline fitting with convexity and concavity constraints. Technical Report NAC 23, National Physical Laboratory, Teddington, UK, 1973. 125

[54] M. G. Cox. The numerical evaluation of a spline from its B-spline representation. *Journal of the Institute of Mathematics and its Applications*, 21:135–143, 1978. 132, 135

[55] M. G. Cox. The least squares solution of overdetermined linear equations having band or augmented band structure. *IMA J. Numer. Anal.*, 1:3 – 22, 1981. 65, 135

[56] M. G. Cox. Practical spline approximation. In P. R. Turner, editor, *Lecture Notes in Mathematics 965: Topics in Numerical Analysis*, pages 79–112, Berlin, 1982. Springer-Verlag. 135

[57] M. G. Cox. Linear algebra support modules for approximation and other software. In J. C. Mason and M. G. Cox, editors, *Scientific Software Systems*, pages 21–29, London, 1990. Chapman & Hall. 65, 99

[58] M. G. Cox. A classification of mathematical software for metrology. In P. Ciarlini, M. G. Cox, R. Monaco, and F. Pavese, editors, *Advanced Mathematical Tools for Metrology*, pages 239–246. World Scientific, 1994. 4

[59] M. G. Cox. Survey of numerical methods and metrology applications: discrete processes. In P. Ciarlini, M. G. Cox, R. Monaco, and F. Pavese, editors, *Advanced Mathematical Tools for Metrology*, pages 1–22. World Scientific, 1994. 4, 126

[60] M. G. Cox. Constructing and solving mathematical models of measurement. In P. Ciarlini, M. G. Cox, F. Pavese, and D. Richter, editors, *Advanced Mathematical Tools in Metrology II*, pages 7–21, Singapore, 1996. World Scientific. 4, 126

[61] M. G. Cox. Graded reference data sets and performance profiles for testing software used in metrogy. In P. Ciarlini, M. G. Cox, F. Pavese, and D. Richter, editors, *Advanced Mathematical Tools in Metrology, III*, pages 43–55, Singapore, 1997. World Scientific. 4

[62] M. G. Cox. A discussion of approaches for determining a reference value in the analysis of key-comparison data. In P. Ciarlini, A. B. Forbes, F. Pavese, and D. Richter, editors, *Advanced Mathematical Tools in Metrology, IV*, pages 45–65, Singapore, 2000. World Scientific. 108

[63] M. G. Cox, M. P. Dainton, A. B. Forbes, P. M. Harris, H. Schwenke, B. R. L. Siebert, and W. Woeger. Use of Monte Carlo simulation for uncertainty evaluation in metrology. In P. Ciarlini, M. G. Cox, E. Filipe, F. Pavese, and D. Richter, editors, *Advanced Mathematical and Computational Tools in Metrology V*, pages 94–106, Singapore, 2001. World Scientific. 4

[64] M. G. Cox and A. B. Forbes. Strategies for testing form assessment software. Technical Report DITC 211/92, National Physical Laboratory, Teddington, December 1992. 159

[65] M. G. Cox, A. B. Forbes, P. M. Fossati, P. M. Harris, and I. M. Smith. Techniques for the efficient solution of large scale calibration problems. Technical Report CMSC 25/03, National Physical Laboratory, Teddington, May 2003. 65, 89, 99, 101, 155

[66] M. G. Cox, A. B. Forbes, and P. M. Harris. Software Support for Metrology Best Practice Guide No. 11: Numerical analysis for algorithm design in metrology. Technical report, National Physical Laboratory, Teddington, 2004. `http://www.npl.co.uk/ssfm/download/bpg.html#ssfmbpg11`. 38, 123

[67] M. G. Cox, A. B. Forbes, P. M. Harris, and G. N. Peggs. Determining CMM behaviour from measurements of standard artefacts. Technical Report CISE 15/98, National Physical Laboratory, Teddington, March 1998. 145

[68] M. G. Cox, A. B. Forbes, P. M. Harris, and I. M. Smith. Classification and solution of regression problems for calibration. Technical Report CMSC 24/03, National Physical Laboratory, May 2003. 74, 75, 101

[69] M. G. Cox and P. M. Harris. Software Support for Metrology Best Practice Guide No. 6: Uncertainty evaluation. Technical report, National Physical Laboratory, Teddington, 2004. 14, 18

[70] M. G. Cox and P. M. Harris. Statistical error modelling. NPL report CMSC 45/04, National Physical Laboratory, Teddington, 2004. `http://www.npl.co.uk/ssfm/download/#cmsc45_04`. 14, 21

[71] M. G. Cox and P. M. Harris. SS*f*M Best Practice Guide No. 6, Uncertainty evaluation. Technical Report DEM-ES-011, National Physical Laboratory, Teddington, UK, 2006. 25, 117

[72] M. G. Cox, P. M. Harris, and P. D. Kenward. Data approximation by polynomial splines. In J. Levesley, I. J. Anderson, and J. C. Mason, editors, *Algorithms for Approximation IV*, pages 331–345. University of Huddersfield, 2002. 135

[73] M. G. Cox, P. M. Harris, and P. D. Kenward. Fixed- and free-knot least-squares univariate data approximation by polynomial splines. NPL report CMSC 13/02, National Physical Laboratory, Teddington, 2002. `http://www.npl.co.uk/ssfm/download/#cmsc13_02`. 135, 149

[74] M. G. Cox and J. G. Hayes. Curve fitting: a guide and suite of algorithms for the non-specialist user. Technical Report NAC 26, National Physical Laboratory, Teddington, UK, 1973. 125

[75] M. G. Cox and J. C. Mason, editors. *Algorithms for Approximation III*, Basel, November 1993. J. C. Baltzer AG. Special issue of *Numerical Algorithms*, volume **5**, nos. 1-4. 4

[76] M. G. Cox and E. Pardo-Igúzquiza. The total median and its uncertainty. In P. Ciarlini, M. G. Cox, E. Filipe, F. Pavese, and D. Richter, editors, *Advanced Mathematical and Computational Tools in Metrology V*, pages 106–117, Singapore, 2001. World Scientific. 108

[77] R. T. Cox. Probability, frequency, and reasonable expectation. *Amer. J. Phys.*, 4:1–13, 1946. 14

[78] A. Crampton and J. C. Mason. Surface approximation of curved data using separable radial basis functions. In P. Ciarlini, M. G. Cox, E. Filipe, F. Pavese, and D. Richter, editors, *Advanced Mathematical and Computational Tools in Metrology V*, pages 298–306, Singapore, 2001. World Scientific. 4

[79] G. B. Dantzig. *Linear programming and extensions.* Princeton University Press, Princeton, N.J., 1963. 103, 105

[80] H. F. Davis. *Fourier Series and Orthogonal Functions.* Dover, New York, 1963. 140

[81] C. de Boor. On calculating with B-splines. *J. Approx. Theory*, 6:50–62, 1972. 135

[82] J. J. Dongarra and E. Grosse. Distribution of mathematical software via electronic mail. *Communications of the ACM.*, pages 403–407, 1987. *http://www.netlib.org.* 5, 128, 135

[83] J. J. Dongarra, C. B. Moler, J. R. Bunch, and G. W. Stewart. *LINPACK Users' Guide.* Society for Industrial and Applied Mathematics, Philadelphia, 1979. 4, 36, 85

[84] J. Du Croz. Relevant general-purpose mathematical and statistical software. In P. Ciarlini, M. G. Cox, F. Pavese, and D. Richter, editors, *Advanced Mathematical Tools in Metrology, II*, pages 22–28, Singapore, 1996. World Scientific. 4

[85] C. F. Dunkl and Y. Xu. *Orthogonal polynomials of several variables.* Cambride University Press, 2001. 152, 153

[86] EUROMET, www.euromet.org.

[87] *EUROMETROS — EUROMET Repository of Software.* `http://www.eurometros.org`. 101, 128, 137, 159

[88] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design.* Academic Press, 1992. 160

[89] R. Fletcher. *Practical Methods of Optimization.* John Wiley and Sons, Chichester, second edition, 1987. 3, 88, 95, 104

[90] A. B. Forbes. Fitting an ellipse to data. Technical Report DITC 95/87, National Physical Laboratory, Teddington, 1987. 159

[91] A. B. Forbes. Least-squares best-fit geometric elements. Technical Report DITC 140/89, National Physical Laboratory, Teddington, 1989. 101, 159

[92] A. B. Forbes. Robust circle and sphere fitting by least squares. Technical Report DITC 153/89, National Physical Laboratory, Teddington, 1989. 159

[93] A. B. Forbes. Least squares best fit geometric elements. In J. C. Mason and M. G. Cox, editors, *Algorithms for Approximation II*, pages 311–319, London, 1990. Chapman & Hall. 99, 101, 159

[94] A. B. Forbes. Geometric tolerance assessment. Technical Report DITC 210/92, National Physical Laboratory, Teddington, October 1992. 104, 110

[95] A. B. Forbes. Generalised regression problems in metrology. *Numerical Algorithms*, 5:523–533, 1993. 96, 101

[96] A. B. Forbes. Mathematical software for metrology – meeting the metrologist's needs. In P. Ciarlini, M. G. Cox, R. Monaco, and F. Pavese, editors, *Advanced Mathematical Tools in Metrology*, pages 247–254, Singapore, 1994. World Scientific. 4

[97] A. B. Forbes. Validation of assessment software in dimensional metrology. Technical Report DITC 225/94, National Physical Laboratory, February 1994. 159

[98] A. B Forbes. Model parametrization. In P. Ciarlini, M. G. Cox, F. Pavese, and D. Richter, editors, *Advanced Mathematical Tools for Metrology*, pages 29–47, Singapore, 1996. World Scientific. 4, 160

[99] A. B. Forbes. Efficient algorithms for structured self-calibration problems. In J. Levesley, I. Anderson, and J. C. Mason, editors, *Algorithms for Approximation IV*, pages 146–153. University of Huddersfield, 2002. 99, 101

[100] A. B. Forbes. Structured nonlinear Gauss-Markov problems. In A. Iske and J. Levesley, editors, *Algorithms for Approximation V*, pages 167–186, Berlin, 2006. Springer. 101, 102

[101] A. B. Forbes. Surface fitting taking into account uncertainty structure in coordinate data. *Measurement Science and Technology*, 17:553–558, 2006. 101

[102] A. B. Forbes. Uncertainty evaluation associated with fitting geometric surfaces to coordinate data. *Metrologia*, 43(4):S282–S290, August 2006. 101

[103] A. B. Forbes. Least squares approaches to maximum likelihood estimation. Technical Report DEM-ES-019, National Physical Laboratory, March 2007. 108

[104] A. B. Forbes, P. M. Harris, and I. M. Smith. Generalised Gauss-Markov Regression. In J. Levesley, I. Anderson, and J. C. Mason, editors, *Algorithms for Approximation IV*, pages 270–277. University of Huddersfield, 2002. 101

[105] A. B. Forbes, P. M. Harris, and I. M. Smith. Correctness of free form surface fitting software. In D. G. Ford, editor, *Laser Metrology and Machine Performance VI*, pages 263–272, Southampton, 2003. WIT Press. 101, 159

[106] G. E. Forsythe. Generation and use of orthogonal polynomials for data fitting with a digital computer. *SIAM Journal*, 5:74–88, 1957. 124, 127, 142

[107] G. E. Forsythe, M. A. Malcolm, and C. B. Moler. *Computer Methods for Mathematical Computation*. Prentice-Hall, Englewood Cliffs, 1977. 3

[108] L. Fox and I. B. Parker. *Chebyshev polynomials in numerical analysis*. Oxford University Press, 1968. 127

[109] M. Frigo and S. G. Johnson. FFTW: An adaptive software architecture for the FFT. In *Proc. 1998 IEEE Intl. Conf. Acoustics Speech and Signal Processing*, volume 3, pages 1381–1384. IEEE, 1998. 140

[110] K. Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2(3):845–848, 1989. 155

[111] W. Gander, G. H. Golub, and R. Strebel. Least squares fitting of circles and ellipses. *BIT*, 34, 1994. 159

[112] B. S. Garbow, K. E. Hillstrom, and J. J. Moré. User's guide for MINPACK-1. Technical Report ANL-80-74, Argonne National Laboratory, Argonne, IL, 1980. 5, 85, 88, 95

[113] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. Chapman & Hall/CRC, Boca Raton, Fl., second edition, 2004. 28, 118, 119

[114] W. M. Gentleman. An error analysis of Goertzel's (Watt's) method for computing Fourier coefficients. *Comput. J.*, 12:160–165, 1969. 125

[115] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press, London, 1981. 3, 35, 95, 104, 155

[116] G. H. Golub. The singular value decomposition with applications. In P. Ciarlini, M. G. Cox, F. Pavese, and D. Richter, editors, *Advanced Mathematical Tools in Metrology, II*, pages 48–55, Singapore, 1996. World Scientific. 4

[117] G. H. Golub and C. F. Van Loan. *Matrix Computations*. John Hopkins University Press, Baltimore, third edition, 1996. 3, 61, 62, 64, 66, 85, 101, 118

[118] A. Greenbaum. *Iterative methods for solving linear systems*. SIAM, Philadelphia, 1997. 155

[119] A. Griewank and G. F. Corliss, editors. *Automatic Differentiation of Algorithms: Theory, Implementation and Applications*, Philadelphia, 1991. Society for Industrial and Applied Mathematics. 95

[120] G. R. Grimmett and D. R. Stirzaker. *Probability and Random Processes*. Oxford University Press, third edition, 2001. 41

[121] S. Hammarling. The numerical solution of the general Gauss-Markov linear model. Technical Report TR2/85, Numerical Algorithms Group, Oxford, 1985. 74

[122] D. C. Handscombe and J. C. Mason. *Chebyshev Polynomials*. Chapman&Hall/CRC Press, London, 2003. 127

[123] P. Hansen. Analysis of discrete ill-posed problems by means of the L-curve. *SIAM J. Sci. Stat. Comp.*, 34(4):561–580, 1992. 155

[124] P. Hansen. Regularization tools: a Matlab package for analysis and solution of discrete ill-posed problems. *Num. Alg.*, 6:1–35, 1994. 155

[125] R. J. Hanson and K. H. Haskell. Algorithm 587: two algorithms for linearly constrained least squares problems. *ACM Trans. Math. Soft.*, 8(3):323–333, 1982. 85

[126] P. M. Harris. The use of splines in the modelling of a photodiode response. Technical Report DITC 88/87, National Physical Laboratory, Teddington, UK, 1987. 145

[127] Harwell subroutine library: a catalogue of subroutines. Technical Report AERE-R-9185, Computer Science and Systems Division, Harwell Laboratory, Harwell Laboratory, England. 5

[128] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Macmillan, New York, second edition, 1999. 154

[129] H.-P. Helfrich and D. Zwick. A trust region method for implicit orthogonal distance regression. *Numerical Algorithms*, 5:535 – 544, 1993. 101

[130] H.-P. Helfrich and D. Zwick. Trust region algorithms for the nonlinear distance problem. *Num. Alg.*, 9:171 – 179, 1995. 101

[131] H.-P. Helfrich and D Zwick. A trust region algorithm for parametric curve and surface fitting. *J. Comp. Appl. Math.*, 73:119–134, 1996. 101

[132] H.-P. Helfrich and D. Zwick. ℓ_1 and ℓ_∞ fitting of geometric elements. pages 162–169, 2002. 101, 104

[133] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. 155

[134] P. J. Huber. Robust estimation of a location parameter. *Ann. Math. Stat.*, 35:73–101, 1964. 108

[135] P. J. Huber. *Robust Statistics*. Wiley, New York, 1980. 108

[136] M. Huhtanen and R. M Larsen. On generating discrete orthogonal bivariate polynomials. *BIT*, 42:393–407, 2002. 151, 153

[137] INRIA, Domaine de Voluceau, Rocquencourt, France. *Scilab*. `www.scilab.org`. 5

[138] A. Iske and J. Levesley, editors. *Algorithms for Approximation V*, Berlin, 2006. Springer. 4

[139] ISO. ISO 3534 statistics – vocabulary and symbols – part 1: probability and general statistical terms. Technical report, International Standards Organisation, Geneva, 1993. 14

[140] D. P. Jenkinson, J. C. Mason, A. Crampton, M. G. Cox, A. B. Forbes, and R. Boudjemaa. Parameterized approximation estimators for mixed noise distributions. In P. Ciarlini, M. G. Cox, F. Pavese, and G. B. Rossi, editors, *Advanced Mathematical and Computational Tools in Metrology VI*, pages 67–81, Singapore, 2004. World Scientific. 108

[141] R. E. Kalman. A new approach to linear filtering and prediction problems. *Trans. ASME – J. Basic Engr.*, pages 35–45, 1960. 85

[142] E Kreyszig. *Advanced Engineering Mathematics*. John Wiley and Sons, eighth edition, 1999. 140

[143] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems*. Prentice-Hall, Englewood Cliffs, 1974. 3, 85

[144] D. Lei, I. J. Anderson, and M. G. Cox. An improve algorithm for approximating data in the ℓ_1 norm. In P. Ciarlini, M. G. Cox, E. Filipe, F. Pavese, and D. Richter, editors, *Advanced Mathematical and Computational Tools for Metrology V*, pages 247–250, Singapore, 2001. World Scientific. 106

[145] D. Lei, I. J. Anderson, and M. G. Cox. A robust algorithm for least absolute deviation curve fitting. In J. Levesley, I. J. Anderson, and J. C. Mason, editors, *Algorithms for Approximation IV*, pages 470–477. University of Huddersfield, 2002. 106

[146] J. Levesley, I. J. Anderson, and J. C. Mason, editors. *Algorithms for Approximation IV*. University of Huddersfield, 2002. 4

[147] G. L. Lord, E. Pardo-Igúzquiza, and I. M. Smith. A practical guide to wavelets for metrology. Technical Report NPL Report CMSC 02/00, National Physical Laboratory, Teddington, June 2000. 149

[148] T. Lyche and K. Mørken. A discrete approach to knot removal and degree reduction for splines. In J. C. Mason and M. G. Cox, editors, *Algorithms for Approximation*, pages 67–82, Oxford, 1987. Clarenden Press. 135

[149] Z. A. Maany. Building numerical libraries using Fortran 90/95. In P. Ciarlini, A. B. Forbes, F. Pavese, and D. Richter, editors, *Advanced Mathematical and Computational Tools in Metrology IV*, pages 143–156. World Scientific, 2000. 4, 5

[150] P. Maas. Wavelet methods in signal processing. In P. Ciarlini, M. G. Cox, F. Pavese, and D. Richter, editors, *Advanced Mathematical Tools in Metrology, III*, pages 43–55, Singapore, 1997. World Scientific. 4

[151] D. J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003. 28

[152] K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis*. Academic Press, London, 1979. 3, 24, 85

[153] J. C. Mason and M. G. Cox, editors. *Algorithms for Approximation*, Oxford, 1987. Clarenden Press. 4

[154] J. C. Mason and M. G. Cox, editors. *Algorithms for Approximation II*, London, 1990. Chapman & Hall. 4

[155] J. C. Mason and D. A. Turner. Applications of support vector machine regression in metrology and data fusion. In P. Ciarlini, M. G. Cox, E. Filipe, F. Pavese, and D. Richter, editors, *Advanced Mathematical and Computation Tools in Metrology V*, Singapore, 2001. World Scientific. 4

[156] MathSoft, Inc., Cambridge, MA. *MathCad 2000. http://www.mathsoft.com.* 5

[157] MathSoft, Inc, Seattle, WA. *S-PLUS 2000 Guide to Statistics, Volumes 1 and 2*, 1999. *http://www.mathsoft.com.* 5

[158] MathWorks, Inc., Natick, Mass. *Using Matlab*, 2002. *http://www.mathworks.com.* 5, 85, 104, 135

[159] M. Metcalf and J. Reid. *Fortran 90/95 Explained*. Oxford University Press, 1996. 5

[160] H. S. Migon and D. Gamerman. *Statistical Inference: an Integrated Approach*. Arnold, London, 1999. 28

[161] M. L. Minsky and S. Papert. *Perceptrons*. MIT Press, Cambridge, MA, 1969. 154

[162] J. J. Moré. The Levenberg-Marquardt algorithm: implementation and theory. In G. A. Watson, editor, *Lecture Notes in Mathematics 630*, pages 105–116, Berlin, 1977. Springer-Verlag. 88

[163] J. J. Moré and S. J. Wright. *Optimization Software Guide*. SIAM, Philadelphia, 1993. 35, 95, 104, 108

[164] N. Morrison. *Introduction to Fourier Analysis*. Wiley, New York, 1994. 140

[165] W. Murray and M. L Overton. A projected Lagrangian algorithm for nonlinear minimax optimization. *SIAM Journal for Scientific and Statistical Computing*, 1(3):345–370, 1980. 110

[166] W. Murray and M. L. Overton. A projected Lagrangian algorithm for nonlinear ℓ_1 optimization. *SIAM Journal for Scientific and Statistical Computing*, 2:207–224, 1981. 110

[167] J. C. Nash. *Compact Numerical Methods for Computers: Linear Algebra and Function Minimisation, Second Edition*. Adam Hilger, Bristol & American Institute of Physics, New York, 1990. 3

[168] National Instruments, Corp., Austin, TX. *LabVIEW*. *http://www.ni.com./*. 5

[169] National Physical Laboratory, *http://www.npl.co.uk/ssfm/index.html*. *Software Support for Metrology Programme*. 6

[170] J. A. Nelder and R. Mead. A simplex method for function minimization. *Comp. J.*, 7:308–313, 1965. 103

[171] G. L. Nemhhauser, A. H. G. Rinnooy Kan, and M. J. Todd, editors. *Handbooks in Operations Research and Management Science, Volume 1: Optimization*, Amsterdam, 1989. North-Holland. 3

[172] NIST, gams.nist.gov. *GAMS: guide to available mathematical software*. 5

[173] NIST/SEMATECH, http://www.itl.nist.gov/div898/handbook/. *e-Handbook of Statistical Methods*. 6

[174] *NPLFit — Software for fitting polynomials and polynomial splines to experimental data*. `http://www.eurometros.org/packages/#nplfitlib`. 128, 137

[175] The Numerical Algorithms Group Limited, Wilkinson House, Jordan Hill Road, Oxford, OX2 8DR. *The NAG Fortran Library, Mark 20, Introductory Guide*, 2002. *http://www.nag.co.uk/*. 4, 85, 95, 104, 110, 128, 135

[176] The Numerical Algorithms Group Limited, Wilkinson House, Jordan Hill Road, Oxford, OX2 8DR. *The NAG Fortran 90 Library*, 2004. *http://www.nag.co.uk/*. 5

[177] M. J. L. Orr. Introduction to radial basis function networks. Technical report, Centre for Cognitive Science, University of Edinburgh, April 1996. 155

[178] M. J. L. Orr. Recent advances in radial basis function networks. Technical report, Institute for Adaptive and Neural Computation, University of Edinburgh, June 1999. 155

[179] M. R. Osborne and G. A. Watson. An algorithm for minimax approximation in the nonlinear case. *Computer Journal*, 12:63–68, 1969. 110

[180] M. R. Osborne and G. A. Watson. On an algorithm for non-linear l_1 approximation. *Computer Journal*, 14:184–188, 1971. 110

[181] C. C. Paige. Fast numerically stable computations for generalized least squares problems. *SIAM J. Numer. Anal.*, 16:165–171, 1979. 74

[182] C. C. Paige and M. A. Saunders. LSQR: and algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software*, 8(1), 1982. 65, 85

[183] F. Pavese et al., editors. *Advanced Mathematical and Computational Tools in Metrology, VI*, Singapore. World Scientific. Turin, 8-12th September , 2003. 4

[184] L. Piegl and W. Tiller. *The NURBS Book*. Springer-Verlag, New York, NY, 2nd edition, 1996. 160

[185] M. J. D. Powell. *Approximation Theory and Methods*. Cambridge University Press, Cambridge, 1981. 3, 103, 104, 106, 127

[186] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes: the Art of Scientific Computing*. Cambridge University Press, Cambridge, 1989. www.nr.com. 3

[187] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in Fortran 90*. Cambridge University Press, Cambridge, 1996. 5

[188] D. Rayner and R. M. Barker. METROS – a website for algorithms for metrology and associated guidance. In P. Ciarlini, M. G. Cox, E. Filipe, F. Pavese, and D. Richter, editors, *Advanced Mathematical and Computational Tools in Metrology V*, pages 298–306, Singapore, 2001. World Scientific. 4

[189] J. A. Rice. *Mathematical Statistics and Data Analysis*. Duxbury Press, Belmont, CA, second edition, 1995. 3, 111

[190] J. R. Rice and R. J. Hanson. References and keywords for Collected Algorithms from ACM. *ACM Trans. Math. Softw.*, 10(4):359–360, December 1984. 5

[191] C. Ross, I. J. Anderson, J. C. Mason, and D. A. Turner. Approximating coordinate data that has outliers. In P. Ciarlini, A. B. Forbes, F. Pavese, and D. Richter, editors, *Advanced Mathematical and Computational Tools in Metrology IV*, pages 210–219. World Scientific, 2000. 108

[192] SIAM, Philadelphia. *The LAPACK User's Guide*, third edition, 1999. 5, 36, 74, 85

[193] D. S. Sivia. *Data Analysis: a Bayesian Tutorial*. Clarendon Press, Oxford, 1996. 28

[194] B. T. Smith, J. M. Boyle, J. J. Dongarra, B. S. Garbow, Y. Ikebe, V. C. Klema, and C. B. Moler. *Matrix Eigensystems Routines - EISPACK Guide*. Springer-Verlag, New York, 1977. Lecture Notes in Computer Science, Vol. 51. 4

[195] W Sorenson, H, editor. 85

[196] D. Sourlier and W. Gander. A new method and software tool for the exact solution of complex dimensional measurement problems. In P. Ciarlini, M. G. Cox, F. Pavese, and D. Richter, editors, *Advanced Mathematical Tools in Metrology, II*, pages 224–237, Singapore, 1996. World Scientific. 159

[197] Mark Spink. *NURBS toolbox for Matlab*. http://www.aria.uklinux.net/nurbs.php3, 2000. 160

[198] W. Squire and G. Trapp. Using complex variables to estimate derivatives of real functions. *SIAM Rev.*, 40:110–112, 1998. 95

[199] StatLib, Statistics Department, Carnegie-Mellon University, http://lib.stat.cmu.edu/. 5

[200] R. F. Stengal. *Optimal Control and Estimation*. Dover, New York, 1994. 85

[201] G. Szego. *Orthogonal Polynomials*. American Mathematical Society, New York, 1959. 124, 127

[202] A. N. Tikhonov and V. Y. Arsenin. *Solutions to Ill–Posed Problems*. Winston and Sons, Washington D. C., 1977. 155

[203] S van Huffel, editor. *Recent Advances in Total Least Squares and Errors-in-Variables Techniques*, Philadelphia, 1997. SIAM. 101

[204] S. van Huffel and J. Vandewalle. *The Total Least Squares Problem: Computational Aspects and Analysis*. SIAM, Philadelphia, 1991. 101

[205] D. Vecchia and J. D. Splett. Outlier-resistant methods for estimation and model fitting. In P. Ciarlini, M. G. Cox, R. Monaco, and F. Pavese, editors, *Advanced Mathematical Tools in Metrology*, pages 143–154. World Scientific, 1994. 108

[206] Visual Numerics, Inc., 12657 Alcosta Boulevard, Suite 450, San Ramon, CA 94583 USA. *IMSL Fortran numerical library, version 5.0. http://www.vni.com/.* 4, 85, 95, 104, 110, 128, 135

[207] G. A. Watson. *Approximation Theory and Numerical Methods*. John Wiley & Sons, Chichester, 1980. 3, 103, 104, 106, 110, 127

[208] G. A. Watson. Some robust methods for fitting parametrically defined curves or surfaces to measured data. In P. Ciarlini, A. B. Forbes, F. Pavese, and D. Richter, editors, *Advanced Mathematical and Computational Tools in Metrology IV*, pages 256–272. World Scientific, 2000. 101, 108

[209] J. H. Wilkinson and C. Reinsch. *Handbook of Automatic Computation Volume II: Linear Algebra*. Springer-Verlag, Berlin, 1971. 3, 85

[210] S. Wolfram. *The Mathematica Book*. Cambridge University Press, Cambridge, third edition. 5

[211] Wolfram Research, Inc., 100 Trade Center Drive, Champaign, IL 61820-7237, USA. *http://www.wolfram.com/mathematica/.* 5

[212] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *Trans. Math. Soft*, 23(4), 1997. 35, 155

[213] D. Zwick. Algorithnms for orthogonal fitting of lines and planes: a survey. In P. Ciarlini, M. G. Cox, F. Pavese, and D. Richter, editors, *Advanced Mathematical Tools in Metrology, II*, pages 272–283, Singapore, 1996. World Scientific. 159